

# VQA-G Annotator: A Self-Improving Agentic Framework for High-Fidelity Grounded Dataset Synthesis

Rongsheng Hu, Yuan Liu, Runwei Guan, Yicheng Di, Jiayu Bao, Yi Jin, Hongjian Shi, Yuan Hong, Lin Yu, Ruhui Ma<sup>(✉)</sup> and Rajkumar Buyya<sup>(✉)</sup>

© The Author(s)

**Abstract** As artificial intelligence systems increasingly rely on multi-source perception and cross-modal learning for human-like visual understanding, their outputs should be both semantically valid and spatially grounded. However, large-scale vision-language benchmarks with precise grounding remain costly to construct, while existing synthetic pipelines often suffer from distribution drift, error propagation, and weak spatial verification. We introduce VQA-G Annotator, a self-improving agentic framework that combines distribution-aware planning, multi-agent orchestration, and spatial reasoning verification in a closed-loop generation, evaluation, and refinement process. Experiments across four benchmark sources show that VQA-G Annotator achieves an average VQAScore of 0.88 for semantic alignment and an Acc@0.5 of 0.57 for spatial grounding, comparing favorably with strong automatic baselines. Downstream and human evaluations further support the utility of the synthesized data for scalable and trustworthy grounded visual understanding.

**Keywords** Visual Question Answering, Visual Grounding, Synthetic Data Generation, Agentic Framework, Spatial Reasoning

- Rongsheng Hu, Yuan Liu, Yicheng Di, Jiayu Bao and Yi Jin are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, 214122, China. E-mail: hu.rongsheng@stu.jiangnan.edu.cn, lyuan1800@jiangnan.edu.cn, diyicheng1@stu.jiangnan.edu.cn, 7243115008@stu.jiangnan.edu.cn, 7253115003@stu.jiangnan.edu.cn.
  - Runwei Guan is with the Thrust of Artificial Intelligence, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, 511453, China. E-mail: runwayrguan@hkust-gz.edu.cn.
  - Hongjian Shi and Ruhui Ma are with the School of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China. E-mail: shhjwu5@sjtu.edu.cn, ruhuima@sjtu.edu.cn.
  - Yuan Hong is with the UGO-AI Intelligent Technology (Shanghai) Co., Ltd, Shanghai, 201206, China. E-mail: yuan.hong@ugo-ai.com
  - Lin Yu is with the Shanghai Jinqiao Intelligent Connected Vehicle Development Co., Ltd, China (Shanghai) Pilot Free Trade Zone, 200131, China. Email: Linyu.00@126.com
  - Rajkumar Buyya is with the University of Melbourne, Victoria, 3010, Australia. E-mail: rbuyya@unimelb.edu.au.
- Manuscript received: 20xx-xx-xx; revised: 20xx-11-xx; accepted: 20xx-00-xx

## 1 Introduction

Modern intelligent systems currently process visual information from heterogeneous sources, including RGB imagery, depth cameras, thermal detectors, and multi-spectral devices [1, 2], reflecting a shift toward contextual reasoning beyond basic pattern recognition. This evolution has catalyzed transformative applications from medical diagnosis [3] to autonomous driving [4]. Central to this are Vision-Language Models (VLMs), which bridge visual perception and natural language understanding, enabling systems to answer questions about images and engage in multimodal reasoning [5–7].

Nevertheless, VLMs often lack reliable spatial anchoring, resulting in plausible but ungrounded answers. This hallucination phenomenon poses significant risks in safety-critical applications where trustworthiness is non-negotiable [8, 9]. Bridging this gap requires models that actively ground their reasoning in verifiable spatial regions, which represents a capability at the intersection of perception, localization, and cognitive inference.

A principled approach is training VLMs with grounded supervision via Visual Question Answering with Grounding (VQA-G), which couples question answering with region-level grounding [10–12]. Grounded supervision makes reasoning verifiable by exposing visual evidence and provides dense training signals that improve compositional reasoning while reducing spurious correlations [13].

Scaling VQA-G data, however, is bottlenecked by the high cost and complexity of manual annotation, which demands linguistically diverse QA pairs with precise spatial grounding. Recent work has explored synthetic data generation using Large Language Models (LLMs) and VLMs [5, 14], and successes in this paradigm [15] demonstrate the potential, yet multimodal grounding introduces unique challenges, simply combining generation and grounding tools is insufficient. Three key obstacles persist: **(1) Distribution drift and mode collapse**. Unconstrained VLMs over-produce easy, high-frequency patterns while under-sampling cognitively demanding queries, drifting from realistic statistics.

This hurts diversity and long-tail coverage critical for robust training [16–18]. (2) **Error propagation in open-loop pipelines.** Linear pipelines are brittle, as incorrect grounding renders samples unusable, yet restarting discards partially correct results [19]. This wastes computation, particularly problematic given GPU resource constraints [20–23]. (3) **Unreliable spatial verification.** “Foundation-model-as-a-judge” paradigms lack coordinate-aware spatial reasoning, often accepting semantically plausible but geometrically inaccurate boxes [24, 25].

To this end, we propose **VQA-G Annotator**, a self-improving agentic framework that synthesizes high-fidelity VQA-G annotations through closed-loop generate, evaluate, and refine. It consists of three components: (1) a **data planner** that uses KL-guided residuals to align synthetic data with reference distributions over question types, object frequencies, and question lengths; (2) a **hub-and-spoke annotate orchestrator** coordinating specialized agents with constraint monitoring and targeted re-execution; and (3) a **Spatial Reasoning Verifier (SR-Verifier)**, a distilled VLM with geometry-based negative augmentation that produces structured critiques for semantic and grounding verification.

Experiments across Visual7W, VizWiz, GQA, and RefCOCOg demonstrate consistent improvements over strong automatic annotation baselines in semantic alignment and spatial grounding. Ablation studies verify the effectiveness of each component, while downstream fine-tuning on synthesized data achieves competitive performance against same-scale human-annotated subsets, demonstrating the practical value of the generated supervision.

Our contributions are as follows:

- We introduce **VQA-G Annotator**, a closed-loop agentic framework for scalable VQA-G dataset synthesis with minimal human intervention.
- We propose a **distribution-aligned data planner** that mitigates mode collapse by tracking KL-guided residuals over reference statistics.
- We design a **hub-and-spoke orchestrator** with constraint monitoring and **targeted re-execution** for robust and efficient annotation.
- We develop a **Spatial Reasoning Verifier** trained via multi-stage distillation for coordinate-aware verification and structured refinement feedback.
- We validate the framework through extensive experiments, ablations, human evaluation, and downstream fine-tuning, demonstrating improvements in semantic quality, spatial grounding, and downstream utility.

The rest of the paper is organized as follows. Section 2

reviews related work. Section 3 presents the proposed framework. Section 4 describes the experimental setup. Section 5 reports the results, including ablations, human evaluation, and downstream performance. Section 6 discusses limitations, and Section 7 concludes the paper.

## 2 Related Work

### 2.1 Grounded VQA and Synthetic Data Generation

VQA-G extends image-level question answering by requiring each answer to be supported by localized visual evidence. Visual7W introduces object-level grounding through “which” questions [12], while GQA uses scene-graph annotations to emphasize compositional reasoning [13]. Referring-expression resources such as RefCOCO, RefCOCOg and Flickr30k Entities further connect natural language to image regions [26–28]. Recent benchmarks broaden grounding to long human-centric expressions, gigapixel scenes, and document VQA [29–31]. Yet these resources still depend on manual region annotations, motivating scalable automatic annotation.

Synthetic supervision reduces this cost and has become central to vision-language instruction tuning [5, 14]. LLaVA synthesizes multimodal conversations from image captions [5], InstructBLIP converts existing datasets into instruction formats [32], and recent pipelines scale synthesis with synthetic pointing or web-mined multimodal content [33, 34]. However, these pipelines mainly optimize linguistic diversity and answer plausibility. Reliable VQA-G data additionally requires region annotations that are semantically faithful to the question and answer, which motivates our coordinated generation and spatial verification framework.

### 2.2 Spatial Verification and Agentic Systems

Open-vocabulary grounding tools provide the basic interface for attaching regions to language. GroundingDINO supports text-conditioned open-set detection [35], while SAM provides promptable segmentation [36]. Although effective as proposal generators, these tools are optimized for generic detection or segmentation and do not directly capture the query-specific semantics of VQA-G, so a plausible box may still fail to support the answer, relation, or evidence required by a generated QA pair.

Recent spatial verification studies therefore make localized evidence explicit. Contrastive Region Guidance tests whether visual prompts causally affect VLM responses [37], while zero-shot REC via True or False verification evaluates candidate boxes independently without enforcing a single best match [38]. R-Bench further shows that LVLMS often

hallucinate inter-object relations and rely on language priors instead of instance-level evidence [39]. Grounded Chain-of-Thought and ViGoRL anchor reasoning steps to coordinates, exposing inconsistencies between answers and supporting regions [40, 41]. Together, these works indicate that spatial verification should jointly assess target identity, geometric fit, and relation consistency, going beyond textual plausibility alone.

Agentic frameworks complement this verification view by organizing generation, tool use, and refinement. ReAct interleaves reasoning with actions [42], while Reflexion and Self-Refine use feedback to revise previous outputs [43, 44]. General “LLM-as-a-judge” evaluation enables scalable assessment [45], but text-centric judges are weak at coordinate-aware auditing. Our work combines agentic generation with a specialized SR-Verifier so that flawed samples can be repaired or rejected according to both semantic and spatial evidence.

### 3 Methodology

#### 3.1 Overview

As shown in Fig. 1, VQA-G Annotator converts an unlabeled image pool  $\mathcal{U}$  into a growing grounded VQA set  $\mathcal{L}$  through a closed plan, generate, verify, and refine loop. For each image, the data planner selects distribution-aware constraints for question type, target object, and question length; the annotate orchestrator instantiates these constraints with specialized captioning, VQA generation, and grounding agents; and SR-Verifier audits the resulting triplet for semantic validity and spatial alignment, accepting reliable samples while returning structured critiques for targeted re-execution. The following subsections detail the planner, orchestrator, and verifier.

#### 3.2 Data Planner

During unsupervised generation, VLMs often exhibit strong preference bias [18], repeatedly producing a narrow set of easy, high-frequency patterns while under-sampling cognitively demanding or long-tail concepts. This behavior is a typical form of mode collapse in generative data synthesis, where the empirical dataset distribution drifts away from a realistic target distribution and becomes overly concentrated on a few modes, yielding reduced diversity and skewed long-tail statistics [16, 17]. To address this issue, we propose a distribution-alignment sampling strategy as the data planner. Instead of treating planning as “filling missing categories”, the planner uses KL divergence as a global alignment objective and implements it through a feasible residual-weight heuristic.

##### 3.2.1 Global Target Distribution Learning

Given a reference dataset  $\mathcal{D}_{\text{ref}}$ , we extract three target distributions that characterize the global statistics of realistic VQA-G data. (1) We categorize questions into seven root types (*What*, *Where*, *When*, *Who*, *Why*, *How*, and *Which*), denoted as  $\mathcal{T}_{\text{root}}$ , and estimate the corresponding distribution  $P_{\text{ref}}^T(q)$  for each type  $q \in \mathcal{T}_{\text{root}}$ . (2) We estimate the object frequency distribution  $P_{\text{ref}}^O(o)$  over a predefined vocabulary  $\mathcal{O}$ . (3) We model question complexity using the token-length distribution  $P_{\text{ref}}^L(\ell)$ , where  $\ell$  denotes the number of tokens in a question and provides a simple proxy for its structural complexity. Together, these distributions define the desired global statistics of the synthesized data.

##### 3.2.2 Distribution-Alignment Objective

Let  $\mathcal{L}_t$  denote the accepted labeled set at the beginning of iteration  $t$ . We compute its empirical question-type, object-frequency, and question-length distributions, denoted by  $\hat{P}_t^T$ ,  $\hat{P}_t^O$ , and  $\hat{P}_t^L$ , respectively.

For compactness, let

$$\mathcal{D} = \{T, O, L\}, \quad (1)$$

with the corresponding category spaces

$$\mathcal{X}_T = \mathcal{T}_{\text{root}}, \quad \mathcal{X}_O = \mathcal{O}, \quad \mathcal{X}_L = [\ell_{\min}, \ell_{\max}] \cap \mathbb{Z}. \quad (2)$$

Here,  $\ell_{\min}$  and  $\ell_{\max}$  denote the minimum and maximum token lengths retained in the length prior. Before computing the divergence, we apply additive smoothing:

$$\tilde{P}_t^d(x) = \frac{c_t^d(x) + \epsilon}{\sum_{x' \in \mathcal{X}_d} c_t^d(x') + \epsilon |\mathcal{X}_d|}, \quad d \in \mathcal{D}, \quad x \in \mathcal{X}_d, \quad (3)$$

where  $c_t^d(x)$  is the count of category  $x$  in  $\mathcal{L}_t$  and  $\epsilon = 10^{-6}$ . This smoothing ensures that the KL divergence remains well-defined when some categories have not yet appeared.

The planner seeks to align the current distributions with the reference distributions by reducing

$$\mathcal{J}(t) = \sum_{d \in \mathcal{D}} \text{KL}(P_{\text{ref}}^d \parallel \tilde{P}_t^d), \quad (4)$$

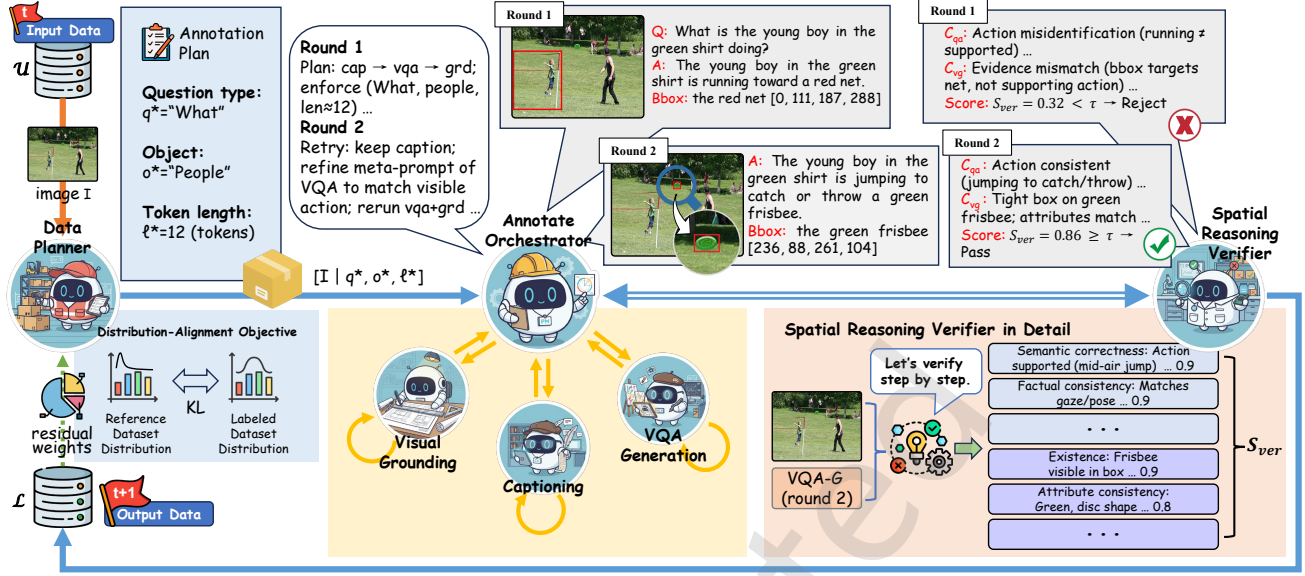
where

$$\text{KL}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}. \quad (5)$$

To implement this objective with discrete sampling decisions, we define the positive distribution deficit

$$r_x^d(t) = [P_{\text{ref}}^d(x) - \tilde{P}_t^d(x)]_+, \quad d \in \mathcal{D}, \quad x \in \mathcal{X}_d, \quad (6)$$

where  $[z]_+ = \max(z, 0)$ . The residuals serve as sampling importance scores: categories with larger deficits receive higher generation priority. As newly accepted samples update



**Fig. 1** Overview of the VQA-G Annotator framework. **Left:** Given an image  $I$  sampled from the unlabeled pool  $\mathcal{U}$ , the data planner uses distribution-alignment residuals to produce an annotation plan ( $q^*, o^*, \ell^*$ ) specifying the target question type, object, and token length. **Center:** The annotate orchestrator executes the plan with three sub-agents, including captioning, VQA generation, and visual grounding, with constraint monitoring and targeted re-execution. **Right:** The Spatial Reasoning Verifier evaluates the generated VQA-G triplet, yielding critiques ( $C_{qa}, C_{vg}$ ) and a score  $S_{ver}$ ; its evaluation protocol is illustrated in the bottom-right inset, while training details are deferred to a later section. Samples with  $S_{ver} < \tau$  are refined, and those with  $S_{ver} \geq \tau$  are added to  $\mathcal{L}$ .

$\mathcal{L}_t$ , their residuals decrease, causing the planner to reallocate sampling effort toward the remaining under-represented categories.

### 3.2.3 Image-Level Feasibility Analysis

Not every question type or object is compatible with every image. Imposing incompatible constraints, such as generating a *Who* question for an image without people, may produce unanswerable questions. Therefore, for each candidate image  $I$  sampled from the unlabeled pool  $\mathcal{U}$ , the planner first identifies its salient objects and normalizes open-vocabulary object mentions to the predefined vocabulary  $\mathcal{O}$ :

$$\text{Obj}(I) = \{o_1, o_2, \dots, o_m\} \subseteq \mathcal{O}, \quad (7)$$

and determines the applicable question types:

$$\mathcal{F}(I) = \{q \in \mathcal{T}_{\text{root}} : \text{IsApplicable}(q, I)\}. \quad (8)$$

Together,  $\text{Obj}(I)$  and  $\mathcal{F}(I)$  define the image-specific feasible space for distribution-aligned planning.

### 3.2.4 Greedy Residual-Guided Planning

Given the residual weights and the feasible space of image  $I$ , the planner uses the residual deficits as a discrete proxy for the distribution-alignment objective. When positive feasible residuals exist, it selects

$$q^* \in \arg \max_{q \in \mathcal{F}(I)} r_q^T(t), \quad (9)$$

$$o^* \in \arg \max_{o \in \text{Obj}(I)} r_o^O(t). \quad (10)$$

If multiple candidates attain the same maximum, one is sampled uniformly using a fixed random seed. If all feasible residuals are zero, the planner instead samples from the corresponding reference distribution restricted and renormalized over the feasible candidates.

The target question length is sampled according to

$$\ell^* \sim P_{\text{res}}^L(\ell | t), \quad (11)$$

where

$$P_{\text{res}}^L(\ell | t) = \begin{cases} \frac{r_{\ell}^L(t)}{\sum_{\ell' \in \mathcal{X}_L} r_{\ell'}^L(t)}, & \sum_{\ell' \in \mathcal{X}_L} r_{\ell'}^L(t) > 0, \\ P_{\text{ref}}^L(\ell), & \text{otherwise.} \end{cases} \quad (12)$$

Because image-level feasibility may exclude globally deficient categories, each step prioritizes the largest feasible residual but does not guarantee a strict decrease in  $\mathcal{J}(t)$ . If either  $\mathcal{F}(I)$  or  $\text{Obj}(I)$  is empty, the image is skipped and another candidate is sampled. If a restricted reference distribution has zero total mass, a uniform distribution over the feasible candidates is used.

The resulting plan

$$\mathcal{P}(I, t) = (q^*, o^*, \ell^*) \quad (13)$$

is passed to the downstream annotation orchestrator as generation constraints. Repeatedly prioritizing under-represented feasible categories reallocates generation effort toward deficient modes, thereby mitigating mode collapse and encour-

aging alignment with the reference statistics.

### 3.3 Annotate Orchestrator

Traditional linear pipelines suffer from error propagation without localized recovery, forcing complete restarts even when earlier stages produced valid outputs [19]. To address this, we design the annotate orchestrator as a centralized executive in a hub-and-spoke topology. It coordinates three specialized sub-agents, namely captioning ( $\mathcal{A}_{\text{cap}}$ ), VQA generation ( $\mathcal{A}_{\text{vqa}}$ ), and visual grounding ( $\mathcal{A}_{\text{grd}}$ ), and supports constraint monitoring with granular rollback, so that only the failing sub-agent is re-executed while valid upstream results are preserved.

#### 3.3.1 Dynamic Execution Workflow

The orchestrator performs Chain-of-Thought (CoT) reasoning [46] to decompose the plan into three sequential stages, each followed by inter-stage constraint verification.

**Captioning.** Given plan  $\mathcal{P} = (q^*, o^*, \ell^*)$ , the orchestrator constructs meta-prompt  $\pi_{\text{cap}}$  for  $\mathcal{A}_{\text{cap}}$  to generate:

$$C_{\text{img}} = \mathcal{A}_{\text{cap}}(I; \pi_{\text{cap}}). \quad (14)$$

The orchestrator validates the caption via a constraint verification function:

$$v_{\text{cap}} = \mathbb{V}_{\text{cap}}(C_{\text{img}}, o^*) \in \{\text{True}, \text{False}\}, \quad (15)$$

which checks whether  $o^*$  is mentioned with sufficient descriptive detail. If  $v_{\text{cap}} = \text{False}$ , the orchestrator constructs an augmented prompt:

$$\pi'_{\text{cap}} = \pi_{\text{cap}} \oplus \mathcal{F}_{\text{aug}}(C_{\text{img}}, o^*, v_{\text{cap}}), \quad (16)$$

where  $\mathcal{F}_{\text{aug}}(\cdot)$  is a feedback synthesis function that generates constraint-aware refinement instructions, and re-invokes  $\mathcal{A}_{\text{cap}}$  to obtain  $C'_{\text{img}}$ .

**VQA Generation.** Using the validated caption  $\tilde{C}_{\text{img}}$  and constraints  $(q^*, \ell^*)$ , the orchestrator prompts  $\mathcal{A}_{\text{vqa}}$  to generate:

$$(q, a) = \mathcal{A}_{\text{vqa}}(I, \tilde{C}_{\text{img}}; \pi_{\text{vqa}}). \quad (17)$$

The generated QA pair undergoes multi-constraint verification:

$$v_{\text{vqa}} = \mathbb{V}_{\text{vqa}}(q, a, q^*, o^*, \ell^*) \in \{\text{True}, \text{False}\}, \quad (18)$$

which jointly checks question type alignment, token length, and object reference. If  $v_{\text{vqa}} = \text{False}$ , the orchestrator determines whether to refine  $\mathcal{A}_{\text{vqa}}$  directly or cascade back to  $\mathcal{A}_{\text{cap}}$  based on error analysis, constructing appropriate augmented prompts via  $\mathcal{F}_{\text{aug}}(\cdot)$ .

**Visual Grounding.** The orchestrator feeds the validated QA pair  $(\tilde{q}, \tilde{a})$  to  $\mathcal{A}_{\text{grd}}$ , which can invoke external expert models as tools. Depending on the task requirement, the grounding tool can be either bounding box detectors [35] or

segmentation models [36]. In this work, we focus on bounding box outputs:

$$b = \mathcal{A}_{\text{grd}}(I, \tilde{q}, \tilde{a}; \pi_{\text{grd}}), \quad (19)$$

where  $b = [x_1, y_1, x_2, y_2]$  denotes the predicted bounding box coordinates. A preliminary sanity check is performed:

$$v_{\text{grd}} = \mathbb{V}_{\text{grd}}(b, I) \in \{\text{True}, \text{False}\}, \quad (20)$$

verifying coordinate validity and minimum box area. This tool-augmented design enhances spatial localization precision while maintaining framework extensibility.

The final output is a complete VQA-G triplet:  $\mathcal{T} = (I, \tilde{q}, \tilde{a}, \tilde{b})$ , where tilde notation indicates validated components.

### 3.4 Spatial Reasoning Verifier

To ensure the reliability of synthesized data, we introduce the **SR-Verifier**, a specialized VLM trained to audit both the semantic correctness of question-answer pairs and the spatial alignment of grounding annotations. While general-purpose VLMs excel at semantic understanding, they often struggle with precise coordinate reasoning and are prone to spatial hallucinations [24, 25]. Our SR-Verifier addresses this through a three-stage training pipeline that distills coordinate-aware reasoning from a teacher model.

#### 3.4.1 Training Pipeline

**Stage 1: Geometry-Based Augmentation.** Given a seed dataset  $\mathcal{D}_{\text{seed}}$  with human-annotated VQA-G triplets  $(I, q, a, b^{\text{GT}})$ , we construct an augmented verification dataset by generating negative samples via controlled spatial distortions. For each positive sample, we synthesize  $\lambda_{\text{neg}}$  negative samples using four geometric operators: (1) *Box Jitter* adds Gaussian noise  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  to simulate annotation errors; (2) *Box Scaling* uniformly expands or shrinks the box by factor  $s \sim \mathcal{U}(s_{\text{min}}, s_{\text{max}})$  to model incomplete coverage; (3) *Aspect Ratio Distortion* alters width-height ratio  $r \sim \mathcal{U}(r_{\text{min}}, r_{\text{max}})$  to capture shape mismatches; (4) *Boundary Shift* translates the box toward image edges to model localization bias. The augmented dataset is:

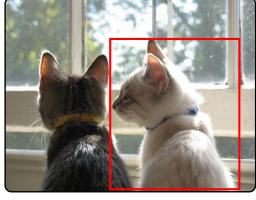
$$\mathcal{D}_{\text{aug}} = \{(I, q, a, b^{\text{GT}}, y)\} \cup \{(I, q, a, b_{\text{neg}}, y)\}, \quad (21)$$

where  $b_{\text{neg}} \in \mathcal{N}_{\text{geo}}$  denote geometry-perturbed negative boxes, and  $y \in \{0, 1\}$  indicates correctness.

**Stage 2: Spatial Reasoning Distillation.** We employ a large VLM as the teacher to generate process-supervised annotations for both positive and negative samples. A key insight is that directly prompting models to output a scalar verification score often leads to score distribution collapse,

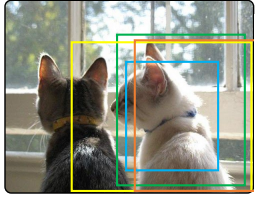
### Step1: Grounding Augmentation

Ground Truth



**Question:**  
Who has the blue necklace?  
**Answer:**  
The white cat.  
**Boxes:**  
white cat [213,81,462,374]

Geometry-Based Augmentation



- **Box Jitter**  
[225, 73, 477, 364]
- **Box Scaling**  
[244, 118, 431, 338]
- **Aspect Ratio distortion**  
[138, 81, 500, 374]
- **Boundary shift**  
[263, 81, 500, 374]

### Step2: Spatial Reasoning Distillation

[Meta-Info]

1. Ground Truth Box: [213, 81, 462, 374]; 2. Proposed Box: [263, 81, 500, 374];
3. Error Type: Boundary Shift; 4.  $S_{\text{target}} = \text{IoU}(b^{\text{GT}}, b_{\text{proposed}}): 0.693$

Teacher VLM  
(Qwen3-VL)

**C<sub>qa</sub>:** ① Semantic✓(1.0) ② Factual✓(1.0) ③ Complete✓(1.0)  
④ Relevant✓(1.0) →  $S_{\text{qa}}=1.0$

**C<sub>vg</sub>:** ① Existence: Object present, shifted right (0.8)  
② Attributes: White fur, blue necklace visible (0.8)  
③ Coverage: Misses left head, excess right background (0.5)  
④ Boundary: Left edge cuts face, right overextends (0.5)  
→  $S_{\text{vg}}=0.65$

$S_{\text{teacher}} = 0.5 \times 1.0 + 0.5 \times 0.65 = 0.825$

### Step3: Student Fine-tuning

Student VLM  
(MiniCPM-o)

**C<sub>qa</sub>:** ①(1.0) ②(1.0) ③(1.0) ④(1.0) →  $S_{\text{qa}}=1.0$

**C<sub>vg</sub>:** ① Existence: Cat present, off-center (0.75)  
② Attributes: Key features captured (0.85)  
③ Coverage: Left portion missing, right excess (0.55)  
④ Boundary: Imprecise edges (0.55) →  $S_{\text{vg}}=0.68$

$S_{\text{pred}} = 0.84$

**Fig. 2 Three-Stage Training Pipeline of Spatial Reasoning Verifier.** *Stage 1:* Geometry-based augmentation generates positive/negative samples via four spatial distortion operators. *Stage 2:* Teacher model (Qwen3-VL) performs dual verification with privileged meta-information  $\mathcal{M}$ , producing structured CoT critiques and fine-grained scores for VQA and visual grounding dimensions. *Stage 3:* Student model (MiniCPM-o) is trained to replicate teacher's reasoning and scoring without access to  $\mathcal{M}$ , learning to generate dual critiques  $\{C_{\text{qa}}, C_{\text{vg}}\}$  and predict aggregated score  $S_{\text{teacher}}$  via supervised fine-tuning.

where models tend to produce a few fixed values instead of calibrated assessments. To address this, we adopt a multi-step scoring paradigm in which the teacher first performs explicit spatial reasoning via CoT [46], then provides fine-grained scores for each reasoning step, which are aggregated into the final score. This design enhances both interpretability and scoring accuracy by decomposing verification into verifiable sub-criteria.

To ensure accurate teacher reasoning, we provide privileged meta-information  $\mathcal{M} = \{b^{\text{GT}}, \text{ErrorType}, S_{\text{target}}\}$  as calibration signal, where  $S_{\text{target}} = \text{IoU}(b^{\text{GT}}, b_{\text{proposed}})$ . Here,  $b_{\text{proposed}} \in \{b^{\text{GT}}\} \cup \mathcal{N}_{\text{geo}}$  denotes the box under evaluation. Critically,  $\mathcal{M}$  is used only to guide the teacher's internal reasoning and does not appear in the generated outputs, ensuring the distilled reasoning chains rely purely on observable visual-textual evidence.

The teacher performs dual verification through structured CoT. For VQA verification, it generates a four-step critique  $C_{\text{qa}} = \{c_1^{\text{qa}}, \dots, c_4^{\text{qa}}\}$ , where each step  $c_k^{\text{qa}}$  contains: (1) natural language reasoning about one dimension (semantic correctness, factual consistency, answer completeness, or question-answer relevance), and (2) a corresponding score  $s_k^{\text{qa}} \in [0, 1]$ . The VQA score is computed as:

$$S_{\text{qa}} = \frac{1}{4} \sum_{k=1}^4 s_k^{\text{qa}}. \quad (22)$$

Similarly, for visual grounding verification, the teacher

produces  $C_{\text{vg}} = \{c_1^{\text{vg}}, \dots, c_4^{\text{vg}}\}$  through four spatial reasoning steps: (1) existence check, which reasons whether the answer-described object exists in the proposed region  $b_{\text{proposed}}$  and produces score  $s_1^{\text{vg}}$ ; (2) attribute consistency, which analyzes the alignment of visual attributes (color, shape, material, state) between the answer and the boxed region, yielding score  $s_2^{\text{vg}}$ ; (3) spatial coverage, which evaluates whether the box tightly encloses the target object without excessive background, producing  $s_3^{\text{vg}}$ ; (4) boundary alignment, which assesses edge precision and completeness, resulting in  $s_4^{\text{vg}}$ . The grounding score aggregates these sub-scores:

$$S_{\text{vg}} = \frac{1}{4} \sum_{k=1}^4 s_k^{\text{vg}}. \quad (23)$$

The final teacher score combines both aspects via a balancing hyperparameter  $\beta \in [0, 1]$ :

$$S_{\text{teacher}} = \beta \cdot S_{\text{qa}} + (1 - \beta) \cdot S_{\text{vg}}. \quad (24)$$

The resulting distillation dataset comprises input triplets, dual critiques with step-wise reasoning and scores, and final aggregated scores:

$$\mathcal{D}_{\text{distill}} = \{I, q, a, b_{\text{proposed}}, \mathcal{C}, S_{\text{teacher}}\}, \quad (25)$$

where  $\mathcal{C} = \{C_{\text{qa}}, C_{\text{vg}}\}$ .

**Stage 3: Student Fine-Tuning.** The student SR-Verifier is fine-tuned on  $\mathcal{D}_{\text{distill}}$  via supervised learning. Given input  $(I, q, a, b_{\text{proposed}})$ , the model is trained to generate dual critiques  $C_{\text{qa}}$  and  $C_{\text{vg}}$  and predict the final score. The training

objective combines two losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CoT}} + \lambda \cdot \mathcal{L}_{\text{score}}, \quad (26)$$

where  $\mathcal{L}_{\text{CoT}}$  is the autoregressive loss for generating reasoning chains,  $\mathcal{L}_{\text{score}} = (S_{\text{pred}} - S_{\text{teacher}})^2$  is the score regression loss, and  $\lambda$  balances the two objectives.

### 3.4.2 Inference and Feedback Loop

During inference, the SR-Verifier evaluates each candidate triplet  $(I, q, a, b)$  by producing dual critiques and a verification score. The final score  $S_{\text{ver}}$  is computed in the same way as Equation 24, using the same  $\beta$  weighting, and the sample is accepted when  $S_{\text{ver}} \geq \tau$  and otherwise rejected. For accepted samples, the triplet is added to the labeled dataset  $\mathcal{L}$ . For rejected samples, the structured feedback  $\mathcal{C}$  together with  $S_{\text{ver}}$  is returned to the annotation orchestrator, which analyzes the critiques, identifies the failure source, and refines the meta-prompts for re-invoking the relevant sub-agents.

Fig. 1 provides a concrete example of this self-improving loop. In round 1, the orchestrator runs the captioning, VQA generation, and visual grounding pipeline to produce a candidate triplet. The SR-Verifier assigns  $S_{\text{ver}} = 0.32 < \tau$  and returns structured critiques, where  $\mathcal{C}_{\text{qa}}$  identifies an action misidentification and  $\mathcal{C}_{\text{vg}}$  flags a mismatch between the answer and the bounding box evidence. In round 2, the orchestrator retains the validated caption, synthesizes constraint-aware meta-prompts from the critiques, and selectively re-runs  $\mathcal{A}_{\text{vqa}}$  and  $\mathcal{A}_{\text{grd}}$ . The corrected annotation reaches  $S_{\text{ver}} = 0.86 \geq \tau$ , showing that closed-loop re-execution can recover from localized errors without discarding valid upstream outputs.

## 4 Experimental Setup

### 4.1 Datasets and Data Usage

We use four public vision-language datasets that cover complementary image domains and grounding formats. Visual7W [12] provides COCO-based images with human QA pairs and object-level grounding annotations. GQA (Balanced) [13] provides Visual Genome-based images with scene-graph-derived compositional questions and object annotations. VizWiz [47] contributes images captured by visually impaired users under challenging real-world conditions such as blur, occlusion, and poor lighting. RefCOCOg [27] provides referring-expression localization examples. For spatial evaluation on VizWiz and RefCOCOg, available region masks or boundary annotations are converted to enclosing axis-aligned boxes when needed.

These datasets are assigned to distribution estimation, verifier training, and benchmark evaluation according to their

splits and annotation types. The data planner estimates linguistic priors from Visual7W training questions and object priors from the GQA training scene graphs associated with images referenced by the GQA balanced training questions. Since these statistics are aggregate priors, they do not expose image-level QA or box annotations from the evaluation pool to the generator. For benchmark evaluation, we construct a shared image pool  $\mathcal{U}$  from Visual7W, GQA, VizWiz, and RefCOCOg, using 5,000 unique images from each source subset. Each automatic method, including all baselines and VQA-G Annotator, generates annotations on this same fixed pool under identical conditions. Original annotations are excluded from generation and used only post-hoc for scoring or reference rows. The Visual7W and GQA image IDs used in  $\mathcal{U}$  are not used as SR-Verifier training, validation, or test examples, nor as downstream fine-tuning examples. Exact image IDs, including COCO-derived overlaps between Visual7W and RefCOCOg, are deduplicated at the image level.

### 4.2 Baseline Methods

We compare four configurations, all following the same three-stage generation protocol, namely caption generation, QA generation, and bounding box generation:

(1) **GPT-4o** [7]: API-based generation with in-house bbox prediction (gpt-4o-2024-08-06,  $T = 0.7$ ); malformed or out-of-bound boxes are discarded.

(2) **GPT-4o (w/ G-DINO)**: GPT-4o for caption or QA generation, with Grounding-DINO [35] for bbox prediction.

(3) **InternVL3** [48]: Open-source 8B VLM with dynamic high-resolution encoding, zero-shot ( $T = 0.7$ ,  $p = 0.9$ ).

(4) **InternVL3 (w/ G-DINO)**: InternVL3 for caption or QA generation, with Grounding-DINO for bbox prediction.

These four baselines and VQA-G Annotator differ only in annotation strategy, namely single-pass plain VLMs, single-pass with Grounding-DINO grounding, and iterative generation with distribution planning and SR-Verifier feedback. This system-level comparison isolates the contribution of the proposed iterative framework to end-to-end annotation quality. Human annotations from the original datasets serve as reference, but are unavailable for RefCOCOg VQA metrics, since that benchmark provides referring expressions and box annotations, not human-annotated question-answer pairs.

### 4.3 Implementation Details

**Data planner.** We assemble the reference distribution  $P_{\text{ref}}$  from two benchmarks across three dimensions. We use the Visual7W training split [12] for linguistic priors, estimating the question-type prior  $P_{\text{ref}}^T$  from 163,630 training QA pairs

whose root types  $\mathcal{T}_{\text{root}}$  are What, Where, When, Who, Why, How, and Which. We estimate the token-length prior  $P_{\text{ref}}^L(\ell)$  from the 149,913 Visual7W training questions whose token lengths fall in the experimental range  $\ell \in [5, 25]$ . For object ontology priors, we define the predefined vocabulary  $\mathcal{O}$  as the 1,702 object categories appearing in the GQA training scene graphs of the 72,140 training images referenced by the GQA balanced training questions. Counting their 1,213,534 object instances over  $\mathcal{O}$  yields  $P_{\text{ref}}^O$ . The planner is instantiated using Qwen3-VL-8B-Instruct<sup>①</sup> with temperature  $T = 0.7$  for extracting image-level feasible sets  $\mathcal{F}(I)$  and object candidates  $\text{Obj}(I) \subseteq \mathcal{O}$  via zero-shot prompting and vocabulary normalization.

**Annotate Orchestrator.** The orchestrator and three sub-agents ( $\mathcal{A}_{\text{cap}}$ ,  $\mathcal{A}_{\text{vqa}}$ ,  $\mathcal{A}_{\text{grd}}$ ) are all based on Qwen3-VL-8B-Instruct. For  $\mathcal{A}_{\text{grd}}$ , we integrate Grounding-DINO (SwinT-OGC) [35] as the external tool, producing bounding boxes  $b = [x_1, y_1, x_2, y_2]$  in absolute coordinates. The orchestrator employs CoT prompting [46] with max context length 2048 tokens, enabling inter-stage feedback synthesis via  $\mathcal{F}_{\text{aug}}(\cdot)$ .

**SR-Verifier.** The SR-Verifier is built on MiniCPM-o 2.6<sup>②</sup> and trained via the three-stage pipeline. *Stage 1:* We construct  $\mathcal{D}_{\text{aug}}$  from 50,000 human-annotated triplets sampled from Visual7W and GQA training images by generating  $\lambda_{\text{neg}} = 3$  negative samples per positive via geometric operators (box jitter with  $\sigma = 0.05$ , scaling factor  $s \sim \mathcal{U}(0.6, 1.4)$ , aspect ratio  $r \sim \mathcal{U}(0.5, 2.0)$ , boundary shift  $\delta = 0.1$ ), yielding 200,000 augmented samples with a 1:3 positive-to-negative ratio. *Stage 2:* We use Qwen3-VL-235B-A22B-Instruct<sup>③</sup> (accessed via API calls) as the teacher model to process  $\mathcal{D}_{\text{aug}}$  and generate dual critiques  $\mathcal{C} = \{\mathcal{C}_{\text{qa}}, \mathcal{C}_{\text{vg}}\}$  and scores  $S_{\text{teacher}}$  using privileged ground-truth IoU as calibration signal, with balancing hyperparameter  $\beta = 0.5$  (Equation 24). *Stage 3:* Student fine-tuning for 3 epochs with batch size 32, learning rate  $2 \times 10^{-5}$ , AdamW optimizer, and loss weight  $\lambda = 1.0$ .

**System Configuration.** We set global verification threshold  $\tau = 0.85$  and maximum refinement iterations  $T_{\text{max}} = 5$ .  $\tau$  is listed here as a system-level deployment parameter; it is tuned on a validation set (see Section 5.2).

## 4.4 Evaluation Metrics

**Automatic Semantic Proxies.** We report three automatic semantic-alignment proxies, specifically VQAScore [49], TIFA [9], and CLIPScore [50]. These metrics provide a consistent basis for comparing generated annotations under a

shared evaluation protocol, and we use them as complementary indicators of semantic alignment, particularly for datasets whose original annotations are not human-written QA pairs.

**Visual Grounding.** Since ground-truth boxes are unavailable for synthetic data, we conduct human evaluation by having 10 expert annotators independently re-annotate 7,200 stratified samples (1,800 per dataset) following standardized protocols. We report the mean IoU (mIoU), averaging  $\text{IoU}(b_{\text{pred}}, b_{\text{human}})$  across all samples, and  $\text{Acc}@0.5$ , the fraction with  $\text{IoU} \geq 0.5$ , a standard localization threshold. VG metrics are reported as  $\text{mean} \pm 95\%$  confidence interval, estimated over the stratified human-reference evaluation set.

**Distributional Alignment.** To validate the data planner’s mitigation of mode collapse, we estimate the final smoothed empirical distributions  $\tilde{P}^d$  from the generated dataset  $\mathcal{L}$  and compute the aggregate KL objective  $\mathcal{J}(t)$  (Equation 4) against the reference statistics  $P_{\text{ref}}$ . Since  $\mathcal{J}(t)$  is the sum of the question-type, object, and length KL terms, we report both the total divergence and its three components. We also report Coverage@100, defined as the fraction of the 100 rarest object categories in  $P_{\text{ref}}^O$  that are covered at least once in the generated set.

**Composite Score.** The primary ranking metric is defined as:

$$\text{VQA-G Score} = 2 \cdot \frac{\text{VQAScore} \cdot \text{Acc}@0.5}{\text{VQAScore} + \text{Acc}@0.5}, \quad (27)$$

which harmonically balances semantic and spatial quality. We propagate the  $\text{Acc}@0.5$  confidence interval through this harmonic-mean function to estimate the VQA-G Score confidence interval, and combine dataset-level intervals in quadrature for the average rows.

## 5 Results and Analysis

In this section, we evaluate VQA-G Annotator from complementary perspectives. We first compare automatic annotation systems across the four benchmark sources (§5.1), followed by ablations to assess each component (§5.2). We then evaluate SR-Verifier as a standalone quality classifier (§5.3), examine downstream transfer through VLM fine-tuning (§5.4), and conclude with human evaluation (§5.5).

### 5.1 Main Results: Comparative Analysis

All automatic methods are evaluated under the same generation and scoring protocol. Source-specific rows in the following tables are computed using the corresponding benchmark source.

#### 5.1.1 Performance Analysis

Table 1 reports a system-level comparison, with results broken down by source subset. VQA-G Annotator obtains consis-

① <https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct>

② <https://huggingface.co/openbmb/MiniCPM-o-2.6>

③ <https://huggingface.co/Qwen/Qwen3-VL-235B-A22B-Instruct>

**Table 1** Performance by benchmark source. We report three semantic-alignment proxies (CLIPScore, TIFA, and VQAScore), two visual grounding metrics (mIoU and Acc@0.5), and the VQA-G Score, defined as the harmonic mean of VQAScore and Acc@0.5. VG metrics and VQA-G Scores are reported as mean $\pm$ 95% confidence interval, with VQA-G intervals propagated from Acc@0.5. Bold values indicate the best automatic method. Grounding variants sharing the same VLM backbone have identical semantic scores, shown using merged cells.

| Dataset                       | Method                        | Semantic Proxies        |                    |                        | VG Metrics                        |                                   | Avg VQA-G Score( $\uparrow$ )     |
|-------------------------------|-------------------------------|-------------------------|--------------------|------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
|                               |                               | CLIPScore( $\uparrow$ ) | TIFA( $\uparrow$ ) | VQAScore( $\uparrow$ ) | mIoU( $\uparrow$ )                | Acc@0.5( $\uparrow$ )             |                                   |
| Visual7W                      | Human Annotation              | 0.651                   | 0.865              | 0.890                  | 0.507 $\pm$ 0.036                 | 0.581 $\pm$ 0.057                 | 0.703 $\pm$ 0.043                 |
|                               | GPT-4o                        | <b>0.777</b>            | <b>0.919</b>       | 0.920                  | 0.354 $\pm$ 0.034                 | 0.282 $\pm$ 0.051                 | 0.432 $\pm$ 0.062                 |
|                               | GPT-4o (w/ G-DINO)            |                         |                    |                        | 0.491 $\pm$ 0.036                 | 0.500 $\pm$ 0.057                 | 0.648 $\pm$ 0.050                 |
|                               | InternVL3                     | 0.718                   | 0.889              | 0.909                  | 0.318 $\pm$ 0.033                 | 0.228 $\pm$ 0.047                 | 0.365 $\pm$ 0.063                 |
|                               | InternVL3 (w/ G-DINO)         |                         |                    |                        | 0.438 $\pm$ 0.035                 | 0.492 $\pm$ 0.057                 | 0.638 $\pm$ 0.050                 |
|                               | <b>VQA-G Annotator (Ours)</b> | 0.742                   | 0.912              | <b>0.928</b>           | <b>0.573<math>\pm</math>0.037</b> | <b>0.629<math>\pm</math>0.055</b> | <b>0.750<math>\pm</math>0.041</b> |
| VizWiz                        | Human Annotation              | 0.724                   | 0.794              | 0.762                  | 0.698 $\pm$ 0.035                 | 0.765 $\pm$ 0.049                 | 0.763 $\pm$ 0.025                 |
|                               | GPT-4o                        | 0.749                   | 0.845              | <b>0.952</b>           | 0.358 $\pm$ 0.033                 | 0.365 $\pm$ 0.054                 | 0.528 $\pm$ 0.059                 |
|                               | GPT-4o (w/ G-DINO)            |                         |                    |                        | 0.427 $\pm$ 0.034                 | 0.516 $\pm$ 0.057                 | 0.669 $\pm$ 0.050                 |
|                               | InternVL3                     | 0.739                   | 0.831              | 0.892                  | 0.338 $\pm$ 0.032                 | 0.325 $\pm$ 0.053                 | 0.476 $\pm$ 0.060                 |
|                               | InternVL3 (w/ G-DINO)         |                         |                    |                        | 0.456 $\pm$ 0.035                 | 0.508 $\pm$ 0.057                 | 0.647 $\pm$ 0.048                 |
|                               | <b>VQA-G Annotator (Ours)</b> | <b>0.758</b>            | <b>0.855</b>       | 0.912                  | <b>0.488<math>\pm</math>0.036</b> | <b>0.542<math>\pm</math>0.056</b> | <b>0.680<math>\pm</math>0.046</b> |
| GQA                           | Human Annotation              | 0.695                   | 0.883              | 0.912                  | 0.550 $\pm$ 0.032                 | 0.620 $\pm$ 0.056                 | 0.738 $\pm$ 0.041                 |
|                               | GPT-4o                        | 0.724                   | 0.898              | 0.908                  | 0.395 $\pm$ 0.031                 | 0.310 $\pm$ 0.052                 | 0.462 $\pm$ 0.060                 |
|                               | GPT-4o (w/ G-DINO)            |                         |                    |                        | 0.502 $\pm$ 0.032                 | 0.548 $\pm$ 0.056                 | 0.683 $\pm$ 0.045                 |
|                               | InternVL3                     | 0.710                   | 0.884              | 0.899                  | 0.378 $\pm$ 0.030                 | 0.295 $\pm$ 0.052                 | 0.444 $\pm$ 0.062                 |
|                               | InternVL3 (w/ G-DINO)         |                         |                    |                        | 0.485 $\pm$ 0.032                 | 0.532 $\pm$ 0.056                 | 0.668 $\pm$ 0.046                 |
|                               | <b>VQA-G Annotator (Ours)</b> | <b>0.730</b>            | <b>0.906</b>       | <b>0.916</b>           | <b>0.518<math>\pm</math>0.033</b> | <b>0.565<math>\pm</math>0.056</b> | <b>0.699<math>\pm</math>0.045</b> |
| RefCOCOg                      | Human Annotation              | –                       | –                  | –                      | 0.816 $\pm$ 0.017                 | 0.962 $\pm$ 0.023                 | –                                 |
|                               | GPT-4o                        | 0.684                   | 0.784              | 0.769                  | 0.348 $\pm$ 0.027                 | 0.285 $\pm$ 0.051                 | 0.416 $\pm$ 0.057                 |
|                               | GPT-4o (w/ G-DINO)            |                         |                    |                        | 0.468 $\pm$ 0.030                 | 0.512 $\pm$ 0.057                 | 0.615 $\pm$ 0.043                 |
|                               | InternVL3                     | 0.672                   | 0.770              | 0.757                  | 0.332 $\pm$ 0.026                 | 0.270 $\pm$ 0.050                 | 0.398 $\pm$ 0.057                 |
|                               | InternVL3 (w/ G-DINO)         |                         |                    |                        | 0.452 $\pm$ 0.029                 | 0.495 $\pm$ 0.057                 | 0.599 $\pm$ 0.044                 |
|                               | <b>VQA-G Annotator (Ours)</b> | <b>0.692</b>            | <b>0.795</b>       | <b>0.780</b>           | <b>0.485<math>\pm</math>0.030</b> | <b>0.528<math>\pm</math>0.056</b> | <b>0.630<math>\pm</math>0.042</b> |
| <b>Average (All Datasets)</b> |                               |                         |                    |                        |                                   |                                   |                                   |
|                               | GPT-4o                        | <b>0.733</b>            | 0.861              | <b>0.887</b>           | 0.364 $\pm$ 0.016                 | 0.311 $\pm$ 0.026                 | 0.460 $\pm$ 0.030                 |
|                               | GPT-4o (w/ G-DINO)            |                         |                    |                        | 0.472 $\pm$ 0.017                 | 0.519 $\pm$ 0.028                 | 0.655 $\pm$ 0.024                 |
|                               | InternVL3                     | 0.709                   | 0.843              | 0.864                  | 0.342 $\pm$ 0.015                 | 0.280 $\pm$ 0.025                 | 0.422 $\pm$ 0.030                 |
|                               | InternVL3 (w/ G-DINO)         |                         |                    |                        | 0.458 $\pm$ 0.017                 | 0.507 $\pm$ 0.028                 | 0.639 $\pm$ 0.024                 |
|                               | <b>VQA-G Annotator (Ours)</b> | 0.731                   | <b>0.867</b>       | 0.884                  | <b>0.516<math>\pm</math>0.017</b> | <b>0.566<math>\pm</math>0.028</b> | <b>0.690<math>\pm</math>0.022</b> |

tently higher VQA-G Scores than the automatic annotation baselines. For semantic alignment, GPT-4o achieves the best average CLIPScore and VQAScore, while our method attains the best average TIFA and remains competitive across all three semantic proxies.

The main difference comes from the semantic–spatial trade-off. Plain VLM baselines produce plausible questions and answers, but their boxes are less reliable. Adding Grounding-DINO improves localization for both GPT-4o and InternVL3, indicating the value of an explicit grounding component. The remaining gap, however, suggests that grounding quality also depends on how generation, box selection, and verification are coordinated. The closed-loop pipeline therefore appears to improve annotation quality not by increasing semantic-proxy scores alone, but by reducing inconsistencies between the generated QA pair and the localized region.

### 5.1.2 Distributional Alignment

Table 2 evaluates distributional alignment on the GQA image pool as an independent diagnostic of planner behavior. Relative to unguided generation, VQA-G Annotator yields lower divergence across question type, object frequency, and question length, together with broader rare-object coverage. This pattern suggests that the planner is not only matching a single marginal statistic, but also helping maintain coverage over less frequent objects.

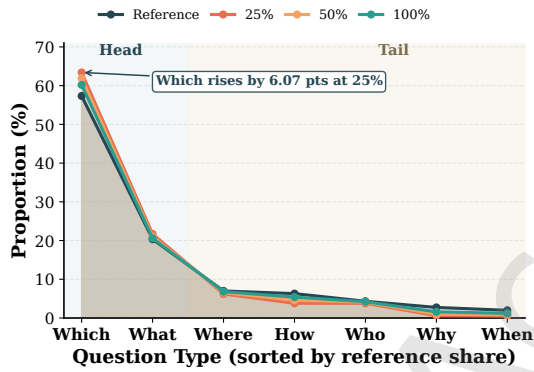
Fig. 3 shows how this effect develops during generation. Early outputs are biased toward frequent question types, and the planner gradually reduces these deviations as more samples are synthesized. By the final stage, the generated distribution moves closer to the reference distribution, with most remaining discrepancies concentrated in low-frequency categories.

**Table 2** Distribution alignment on the GQA image pool. Let  $KL_d = KL(P_{ref}^d \| \tilde{P}^d)$  for  $d \in \{T, O, L\}$ , where  $T$ ,  $O$ , and  $L$  denote question type, object, and length.  $KL_{sum} = KL_T + KL_O + KL_L$  matches the objective in Equation 4. Coverage@100 is the fraction of the 100 rarest object categories in  $P_{ref}^O$  covered at least once.

| Method          | $KL_{sum}(\downarrow)$ | $KL_T(\downarrow)$ | $KL_O(\downarrow)$ | $KL_L(\downarrow)$ | Coverage@100( $\uparrow$ ) |
|-----------------|------------------------|--------------------|--------------------|--------------------|----------------------------|
| InternVL3       | 9.4705                 | 3.1255             | 5.1362             | 1.2088             | 0.442                      |
| GPT-4o          | 9.1349                 | 2.9595             | 5.0460             | 1.1294             | 0.469                      |
| VQA-G Annotator | <b>3.0969</b>          | <b>0.7212</b>      | <b>1.3021</b>      | <b>1.0736</b>      | <b>0.667</b>               |

**Table 3** Ablation study on the GQA image pool. Each row removes or replaces one design choice while keeping the remaining generation pipeline unchanged. Acc@0.5 and VQA-G Score are reported as mean $\pm$ 95% confidence interval.

| Variant                             | VQAScore( $\uparrow$ ) | Acc@0.5( $\uparrow$ ) | VQA-G Score( $\uparrow$ ) | Tokens/Sample |
|-------------------------------------|------------------------|-----------------------|---------------------------|---------------|
| Full System                         | 0.916                  | 0.565 $\pm$ 0.056     | 0.699 $\pm$ 0.045         | 2,850         |
| w/o KL Planner (Uniform)            | 0.842                  | 0.521 $\pm$ 0.057     | 0.644 $\pm$ 0.045         | 2,820         |
| w/o SR-Verifier (Qwen3-VL-8B Judge) | 0.878                  | 0.384 $\pm$ 0.055     | 0.534 $\pm$ 0.056         | 2,780         |
| w/o Targeted Re-exec                | 0.897                  | 0.542 $\pm$ 0.056     | 0.676 $\pm$ 0.046         | 4,200         |
| w/o Orchestrator                    | 0.795                  | 0.438 $\pm$ 0.056     | 0.565 $\pm$ 0.049         | 1,650         |



**Fig. 3** Evolution of question-type distribution run on the GQA image pool. The reference distribution is compared against intermediate generations at 25%, 50%, and 100% completion. The planner progressively reduces the early over-representation of “Which” and improves tail categories such as “When”, “Why”, and “How”, yielding a closer match to  $P_{ref}^T$  by the final stage.

## 5.2 Ablation Studies

To examine the role of each component, we conduct ablations on the GQA image pool, generating 10,000 samples for each variant under the same evaluation protocol.

### 5.2.1 Component-Wise Ablation

Table 3 shows a consistent decline when a core design choice is removed or replaced. For the planner ablation, we keep the same image-level feasible space but replace KL-guided residual selection with uniform sampling over feasible question types and objects. The uniform variant mainly affects semantic quality and question diversity, with manual inspection showing a concentration of generated questions in frequent templates such as “What-color” and “What-is”.

For the verifier ablation, we replace the distilled SR-Verifier with generic Qwen3-VL-8B self-evaluation. This variant produces the clearest localization loss, especially in

Acc@0.5, suggesting that generic VLM judgment is less reliable for coordinate-level spatial errors. Removing targeted re-execution has a smaller effect on final quality but increases token usage, since the system must rerun complete samples instead of repairing failed stages. The orchestrator ablation also reduces the VQA-G Score, suggesting that inter-stage coordination and selective rollback are important for converting partial intermediate outputs into reliable final annotations. Overall, the ablation results support that the gains come not from a single module but from the combined planning, verification, and refinement design.

### 5.2.2 Impact of Verification Threshold

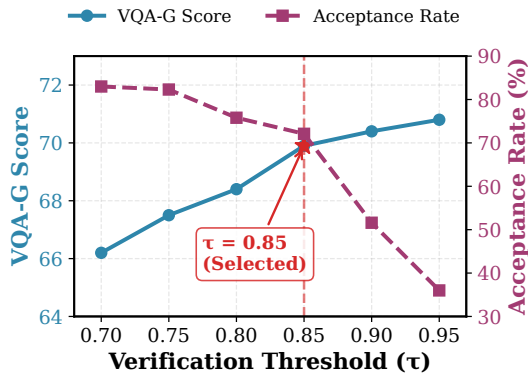
Fig. 4 shows the expected quality–throughput trade-off as the acceptance threshold varies. A high threshold improves the retained-sample score but rejects many candidates, while a low threshold increases acceptance at the cost of noisier annotations. We use  $\tau = 0.85$  as a practical operating point because it preserves most of the quality gain while keeping the acceptance rate suitable for large-scale generation.

### 5.2.3 Iterative Refinement Analysis

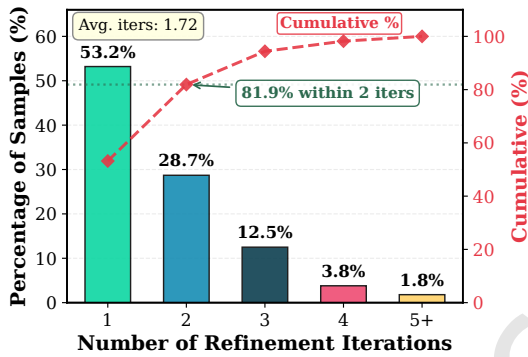
Fig. 5 summarizes how many refinement rounds are needed before acceptance. Most accepted samples pass within one refinement, and only a small fraction reaches the later iterations. The final scores of refined samples remain close to those accepted on the first attempt, suggesting that targeted refinement can recover many initially flawed candidates without simply admitting lower-quality outputs.

## 5.3 Standalone Evaluation of SR-Verifier

We evaluate SR-Verifier as a standalone binary classifier on controlled geometric corruptions and two non-geometric



**Fig. 4** Trade-off between data quality (VQA-G Score) and acceptance rate as verification threshold  $\tau$  varies. The red dashed line indicates our selected operating point  $\tau = 0.85$ .



**Fig. 5** Distribution of refinement iterations among accepted samples. Bars show per-iteration percentages, and the dashed line shows the cumulative distribution. Overall, 81.9% of samples are accepted within one refinement and 94.4% within two, indicating rapid convergence.

consistency errors. The former measures sensitivity to localization perturbations, while *Entity Error* and *Evidence Conflict* test whether the verifier can reject geometrically plausible regions that identify the wrong target or fail to support the question-answer pair.

### 5.3.1 Experimental Setup

Given a VQA-G sample  $(I, q, a, b)$ , SR-Verifier predicts whether  $b$  correctly grounds the question-answer pair, where  $y = 1$  denotes a valid grounding and  $y = 0$  an inconsistent one. Its output  $S_{\text{ver}} \in [0, 1]$  serves as the classification confidence. All evaluation samples are drawn from the validation and test splits of Visual7W, GQA, and VizWiz and are disjoint from the verifier training set  $\mathcal{D}_{\text{aug}}$ .

For geometric evaluation, we apply the four corruption operations defined in Section 3.4: box jitter, box scaling, aspect-ratio distortion, and boundary shift. The validation set contains 2,000 positive and 2,000 negative samples, with 500 negatives from each corruption type. We select  $\tau_{\text{cls}} = 0.72$  by maximizing F1 over  $\tau_{\text{cls}} \in [0.5, 0.9]$ . The mixed geometric

**Table 4** Standalone performance of SR-Verifier on geometric and non-geometric consistency tests.

| Test Set                    | Acc ( $\uparrow$ ) | F1 ( $\uparrow$ ) | AUC ( $\uparrow$ ) |
|-----------------------------|--------------------|-------------------|--------------------|
| Mixed Geometric Corruptions | 0.892              | 0.885             | 0.934              |
| Box Jitter                  | 0.812              | 0.825             | 0.833              |
| Box Scaling                 | 0.815              | 0.838             | 0.947              |
| Aspect-Ratio Distortion     | 0.852              | 0.865             | 0.882              |
| Boundary Shift              | 0.945              | 0.941             | 0.966              |
| Entity Error                | 0.884              | 0.885             | 0.913              |
| Evidence Conflict           | 0.841              | 0.857             | 0.869              |

test set contains 10,000 positives and 10,000 negatives, with the negatives sampled equally across the four types. Each corruption-specific test is constructed independently using the same 10,000 positives and 10,000 negatives of the corresponding type, ensuring balanced and directly comparable evaluation.

We further construct two non-geometric test sets. An *Entity Error* replaces the target box with a plausible distractor object or another instance of the same category. An *Evidence Conflict* uses a visually related region that does not provide the relational, causal, or contextual evidence required by the fixed question-answer pair. Each test set contains 1,000 original positives and 1,000 synthesized negatives. Samples without a suitable replacement region are skipped to avoid ambiguous labels. We report Accuracy, F1, and ROC-AUC for all settings.

### 5.3.2 Results Analysis

As shown in Table 4, SR-Verifier reliably distinguishes valid groundings from geometric corruptions. Box jitter is the most difficult case because a perturbed box may retain substantial target content, whereas boundary shifts are more readily detected. The consistently strong AUC values indicate that the verifier also ranks valid and corrupted samples effectively across different perturbations.

SR-Verifier remains effective on Entity Error and Evidence Conflict, although both retain geometrically plausible regions. Evidence Conflict is more challenging because verification requires judging whether the selected region sufficiently supports the relation or context expressed by the question-answer pair. These results show that SR-Verifier captures semantic-spatial consistency in addition to geometric localization quality.

## 5.4 Downstream Task Evaluation

To assess the practical utility of the synthesized data, we fine-tune VLMs on VQA-G Annotator outputs and evaluate them on downstream VQA and grounding benchmarks. The

goal is to test whether the generated samples provide useful supervision under a controlled same-size fine-tuning setting.

#### 5.4.1 Experimental Setup

We select two representative VLMs as fine-tuning backbones: (1) **LLaVA-1.5-7B**<sup>①</sup>, a widely adopted open-source baseline; and (2) **Qwen2-VL-7B-Instruct**<sup>②</sup>, which provides visual encoding and instruction-following capability. For each model, we perform LoRA fine-tuning [51] with rank  $r = 64$  and  $\alpha = 128$ , targeting the attention projection layers. We compare three training configurations, namely, **Baseline**: the pre-trained model without additional fine-tuning; **Human-Annotated**: LoRA fine-tuned on 10,000 human-labeled VQA-G samples drawn from the Visual7W and GQA training sets; and **Synthetic (Ours)**: LoRA fine-tuned on 10,000 samples generated by VQA-G Annotator.

All models are trained for 3 epochs with batch size 128, learning rate  $2 \times 10^{-4}$ , and AdamW optimizer. Evaluation is conducted on the test splits of Visual7W, GQA, and RefCOCOg using accuracy (Acc) for VQA and mean Intersection over Union (mIoU) for visual grounding. Given the broad coverage of public VLM pre-training data, we view this experiment primarily as a controlled comparison of fine-tuning data quality under the same model initialization.

#### 5.4.2 Results Analysis

Table 5 shows that fine-tuning on VQA-G Annotator data is competitive with fine-tuning on same-size human-annotated subsets for both backbones. The gains are modest but consistent across the evaluated VQA and grounding metrics. These findings, obtained under a controlled 10,000-sample setting with fixed model, training budget, and evaluation protocol, indicate that distribution-aligned, spatially verified samples can serve as effective supervision as opposed to merely boosting annotation scores.

### 5.5 Qualitative Analysis and Human Evaluation

Automatic metrics quantify semantic proxies and localization overlap, but they do not fully capture question naturalness, perceived usefulness, or ambiguous cases that require human judgment. We therefore conduct a blinded human evaluation of generated VQA-G triplets.

#### 5.5.1 Human Evaluation Protocol

We recruit 15 expert annotators with backgrounds in computer vision and NLP. The study covers 1,200 anonymized VQA-G triplets, with 300 stratified samples from each of Visual7W, VizWiz, GQA, and RefCOCOg. Each triplet is

rated by three annotators, and assignments are balanced across methods and datasets. Method names, dataset source labels, generation order, and SR-Verifier scores are hidden; samples are presented in randomized order. Annotators score semantic correctness, spatial precision, question naturalness, and overall usefulness on a 5-point Likert scale. We average continuous ratings for the reported scores and estimate 95% confidence intervals over the balanced method-level triplet assignments. We use majority vote for binary quality-control decisions and expert adjudication for ties when needed. The study includes 60 gold-check samples with known severe grounding or QA errors; no annotator was excluded under the 80% gold-check accuracy rule. Fleiss'  $\kappa = 0.68$  indicates reasonably consistent inter-annotator agreement.

#### 5.5.2 Comparative Human Evaluation Results

Table 6 reports the mean human scores. VQA-G Annotator receives the largest average score, with the clearest difference in spatial precision. This agrees with the automatic grounding metrics and suggests that SR-Verifier helps enforce region-level alignment that annotators can perceive. GPT-4o obtains a slightly higher semantic score, which is consistent with its language and visual reasoning capability, but its spatial score is much lower without dedicated grounding.

For naturalness and usefulness, the differences are smaller but still favor the proposed pipeline in this evaluation. We interpret these results as complementary to the automatic metrics: human ratings capture perceived annotation quality and ambiguity, while mIoU and Acc@0.5 measure overlap under a fixed box-matching criterion.

#### 5.5.3 Case Study

Fig. 6 illustrates how VQA-G Annotator associates each QA pair with question-dependent visual evidence. For direct spatial and recognition questions, the retained boxes identify targets at an appropriate level of granularity. Case 1 localizes the batter within the batter's box, Case 4 isolates the pedestrian walking on the sidewalk while holding an umbrella, and Case 7 focuses on the small Toshiba logo instead of the entire laptop. For action and causal questions, the grounding preserves the evidence needed to support the answer. In Case 2, the box covers both the cat and the contacting hand, while Case 3 highlights the disc involved in the attempted block. Case 5 further shows that the system can associate a temporal answer with the visually relevant scene content.

The figure also demonstrates instance disambiguation in crowded or cluttered scenes. In Case 6, the selected region corresponds to the person on the far right raising a paddle while excluding the other occupants of the raft. Similarly,

① <https://huggingface.co/liuhaotian/llava-v1.5-7b>

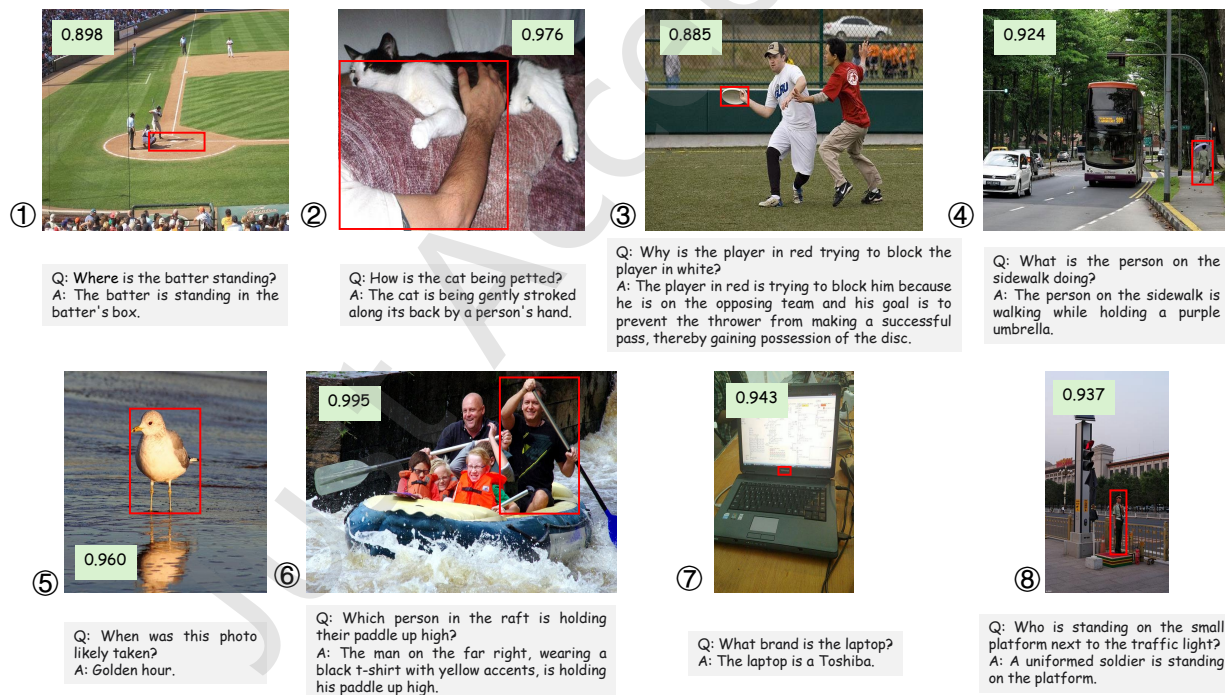
② <https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct>

**Table 5** Downstream task performance after fine-tuning on different data sources. We report accuracy (Acc) for VQA on Visual7W and GQA test sets and mean IoU (mIoU) for visual grounding. Bold entries mark the largest values among fine-tuned settings for each model.

| FT Model     | Training Data        | Visual7W     |              | GQA          |              | RefCOCOg |              | Average      |              |
|--------------|----------------------|--------------|--------------|--------------|--------------|----------|--------------|--------------|--------------|
|              |                      | Acc(↑)       | mIoU(↑)      | Acc(↑)       | mIoU(↑)      | Acc(↑)   | mIoU(↑)      | Acc(↑)       | mIoU(↑)      |
| LLaVA-1.5-7B | Baseline (zero-shot) | 0.542        | 0.398        | 0.587        | 0.421        | -        | 0.365        | 0.565        | 0.395        |
|              | Human-Annotated      | 0.628        | 0.512        | 0.665        | 0.538        | -        | 0.485        | 0.647        | 0.512        |
|              | Synthetic (Ours)     | <b>0.643</b> | <b>0.530</b> | <b>0.681</b> | <b>0.558</b> | -        | <b>0.505</b> | <b>0.662</b> | <b>0.531</b> |
| Qwen2-VL-7B  | Baseline (zero-shot) | 0.598        | 0.445        | 0.624        | 0.468        | -        | 0.412        | 0.611        | 0.442        |
|              | Human-Annotated      | 0.672        | 0.548        | 0.708        | 0.575        | -        | 0.521        | 0.690        | 0.548        |
|              | Synthetic (Ours)     | <b>0.689</b> | <b>0.568</b> | <b>0.723</b> | <b>0.595</b> | -        | <b>0.544</b> | <b>0.706</b> | <b>0.569</b> |

**Table 6** Human evaluation scores (1-5 scale) on 1,200 stratified triplets across four datasets. Annotators rate samples on semantic correctness, spatial precision, question naturalness, and overall usefulness. Scores are reported as mean $\pm$ 95% confidence interval. Bold entries mark the largest mean score in each column.

| Method                        | Semantic                        | Spatial                         | Naturalness                     | Usefulness                      | Avg.                            |
|-------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| GPT-4o                        | <b>4.52<math>\pm</math>0.16</b> | 2.65 $\pm$ 0.25                 | 4.15 $\pm$ 0.21                 | 4.22 $\pm$ 0.20                 | 3.89 $\pm$ 0.23                 |
| GPT-4o (w/ G-DINO)            | 4.28 $\pm$ 0.19                 | 3.31 $\pm$ 0.25                 | 4.38 $\pm$ 0.18                 | 4.45 $\pm$ 0.17                 | 4.11 $\pm$ 0.21                 |
| InternVL3                     | 3.45 $\pm$ 0.25                 | 2.22 $\pm$ 0.23                 | 3.58 $\pm$ 0.24                 | 3.48 $\pm$ 0.25                 | 3.18 $\pm$ 0.25                 |
| InternVL3 (w/ G-DINO)         | 3.71 $\pm$ 0.24                 | 3.76 $\pm$ 0.23                 | 3.85 $\pm$ 0.23                 | 3.78 $\pm$ 0.23                 | 3.78 $\pm$ 0.23                 |
| <b>VQA-G Annotator (Ours)</b> | 4.48 $\pm$ 0.17                 | <b>4.56<math>\pm</math>0.16</b> | <b>4.42<math>\pm</math>0.18</b> | <b>4.50<math>\pm</math>0.17</b> | <b>4.49<math>\pm</math>0.17</b> |

**Fig. 6** Representative VQA-G annotations produced by VQA-G Annotator, with accepted grounding boxes and SR-Verifier quality scores. The examples cover spatial localization, action and causal reasoning, temporal inference, fine-grained recognition, and instance disambiguation under multi-object or cluttered scenes.

Case 8 identifies the uniformed soldier standing on the small platform despite nearby traffic infrastructure and background distractors. Across the eight cases, the accepted boxes range from compact diagnostic regions to larger interaction regions, suggesting that SR-Verifier evaluates whether the grounding is answer-relevant regardless of box size. These examples com-

plement the quantitative and human evaluations by showing how the closed-loop generate-evaluate-refine process supports fine-grained localization, relation-sensitive grounding, and target disambiguation across diverse scenes.

## 6 Discussion and Limitations

VQA-G Annotator is also cost-effective for large-scale generation. The online cost is approximately \$0.003 to \$0.004 per accepted sample, excluding one-time verifier distillation, benchmark construction, and human evaluation. This is about  $12\times$  lower than our GPT-4o-based setup and  $125$  to  $300\times$  lower than typical manual annotation costs. Iterative refinement increases latency by approximately  $1.5$  to  $2\times$ , but provides a favorable trade-off when annotation quality is the primary concern.

The comparison is conducted at the annotation-pipeline level because each method places different amounts of structure around its backbone VLM. The observed gains therefore reflect the joint contribution of residual planning, iterative feedback, targeted re-execution, and specialized verification. Future work may evaluate additional iterative pipelines under matched compute budgets.

The main failures arise from complex background separation, accounting for 62% of rejected samples, and mirror or reflective scenes, accounting for 23%. These cases introduce visual ambiguity for both grounding and verification. The SR-Verifier is also limited to bounding-box supervision and does not support pixel-level segmentation or multi-object relational grounding. Moreover, downstream evaluation uses same-size fine-tuning subsets, leaving larger mixtures of human and synthetic data for future study.

Future work will extend the framework to specialized domains such as medical imaging and autonomous driving, as well as temporal and 3D data including video and point clouds. Its modular design allows individual agents to be replaced or extended while retaining closed-loop verification.

## 7 Conclusion

In this paper, we presented VQA-G Annotator, a self-improving agentic framework for high-fidelity grounded VQA data synthesis. It integrates a KL-guided data planner, a constraint-aware multi-agent orchestrator, and an SR-Verifier that provides structured semantic and spatial feedback within a closed-loop refinement process. Experiments across four benchmark sources show consistent improvements over strong automatic baselines, while standalone evaluation, downstream fine-tuning, and human assessment further support the effectiveness and utility of the generated annotations. Although the current framework is limited to bounding-box grounding and remains sensitive to visual ambiguity, its modular design can be extended to richer supervision, specialized domains, and temporal or 3D multimodal scenarios. By linking semantic interpretation with explicit spatial evidence and iterative

verification, this work supports the transition from visual perception toward trustworthy cognitive-level understanding.

## References

- [1] D. S. da Silva, R. S. Astolfi, S. K. Jagatheesaperumal, P. Selvaraj, N. S. Russel, and V. H. C. de Albuquerque, Advances in 3d fusion of multimodal medical images: 3d reconstruction of bone, muscle, and ligament structures under load from radiographs and magnetic resonance imaging, *Research on Biomedical Engineering*, vol. 41, no. 1, pp. 1–35, 2025.
- [2] J. Zhu, H. Li, and T. Zhang, Camera, lidar, and imu based multi-sensor fusion slam: A survey, *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 415–429, 2024.
- [3] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. De Albuquerque, A novel transfer learning based approach for pneumonia detection in chest x-ray images, *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.
- [4] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. De Albuquerque, Deep learning for safe autonomous driving: Current challenges and future directions, *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.
- [5] H. Liu, C. Li, Q. Wu, and Y. J. Lee, Visual instruction tuning, *arXiv preprint arXiv:2304.08485*, 2023.
- [6] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, Qwen2.5-vl technical report, 2025. [Online]. Available: <https://arxiv.org/abs/2502.13923>
- [7] OpenAI, Gpt-4o system card, 2024. [Online]. Available: <https://arxiv.org/abs/2410.21276>
- [8] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko, Object hallucination in image captioning, *arXiv preprint arXiv:1809.02156*, 2018.
- [9] Y. Hu *et al.*, Tifa: Accurate and interpretable evaluation of text-to-image models, *arXiv preprint arXiv:2303.11897*, 2023.
- [10] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, Vqa: Visual question answering, *arXiv preprint arXiv:1505.00468*, 2015.
- [11] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, Making the v in vqa matter: Elevating the role of image understanding in visual question answering, in *CVPR*, 2017.
- [12] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, Visual7w: Grounded question answering in images, in *CVPR*, 2016.
- [13] D. A. Hudson and C. D. Manning, Gqa: A new dataset for real-world visual reasoning and compositional question answering, in *CVPR*, 2019.
- [14] Y. Wang, Y. Kordi, S. Mishra *et al.*, Self-instruct: Aligning language models with self-generated instructions, *arXiv preprint arXiv:2212.10560*, 2022.

- [15] H. Shi, Y. Di, X. Ruan, M. Liao, Q. Zhang, R. Ma, and H. Guan, Efficient federated recommender system with adaptive model pruning and momentum-based batch adjustment, *ACM Trans. Recomm. Syst.*, vol. 3, no. 3, Mar. 2025. [Online]. Available: <https://doi.org/10.1145/3704267>
- [16] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, Ai models collapse when trained on recursively generated data, *Nature*, vol. 631, no. 8022, pp. 755–759, 2024.
- [17] Z. Hu, M. Rostami, and J. Thomason, Multi-modal synthetic data training and model collapse: Insights from vlms and diffusion models, 2025. [Online]. Available: <https://arxiv.org/abs/2505.08803>
- [18] J. Zhang, S. Yu, D. Chong, A. Sicilia, M. R. Tomz, C. D. Manning, and W. Shi, Verbalized sampling: How to mitigate mode collapse and unlock llm diversity, 2025. [Online]. Available: <https://arxiv.org/abs/2510.01171>
- [19] J. He, C. Treude, and D. Lo, Llm-based multi-agent systems for software engineering: Literature review, vision, and the road ahead, *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 5, May 2025. [Online]. Available: <https://doi.org/10.1145/3712003>
- [20] Z. Jiang, Z. Cai, H. Zhao, B. Zhang, T. Wu, Y. Qiang, R. Ma, H. Guan, and R. Buyya, HeShare: Energy-Aware and Efficient Multi-Task GPU Sharing in Heterogeneous GPU-based Computing Systems, *IEEE Transactions on Computers*, no. 01, pp. 1–12, Nov. 5555. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/TC.2025.3628924>
- [21] L. Jin, Z. Cai, H. Wang, Z. Zhang, R. Ma, H. Guan, Y. Liu, and B. Rajkumar, Ephemera: Accelerating i/o-intensive serverless workloads with a harvested in-memory file system, *ACM Trans. Archit. Code Optim.*, vol. 22, no. 3, Sep. 2025. [Online]. Available: <https://doi.org/10.1145/3747846>
- [22] J. Liu, Z. Cai, Y. Liu, H. Li, Z. Zhang, R. Ma, and R. Buyya, Smore: Enhancing gpu utilization in deep learning clusters by serverless-based co-location scheduling, *IEEE Transactions on Parallel and Distributed Systems*, vol. 36, no. 5, pp. 903–917, 2025.
- [23] L. Lin, R. Ma, Z. Wang, Z. Cai, H. Xu, B. Zhang, R. Ma, and R. Buyya, Hwdsqp: A historical weighted and dynamic scheduling quantum protocol to enhance communication reliability, *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 8, pp. 2810–2824, 2025.
- [24] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 14 455–14 465.
- [25] A. Favero, L. Zancato, M. Trager, S. Choudhary, P. Perera, A. Achille, A. Swaminathan, and S. Soatto, Multi-modal hallucination control by visual information grounding, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 303–14 312.
- [26] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, Referring to objects in photographs of natural scenes, in *EMNLP*, 2014.
- [27] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, Generation and comprehension of unambiguous object descriptions, in *CVPR*, 2016.
- [28] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.
- [29] F. Wei, J. Zhao, K. Yan, H. Zhang, and C. Xu, A large-scale human-centric benchmark for referring expression comprehension in the llm era, in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 69 566–69 587. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2024/file/80f0cd0305f7741659304f5325f3bf6d-Paper-Datasets\\_and\\_Benchmarks\\_Track.pdf](https://proceedings.neurips.cc/paper_files/paper/2024/file/80f0cd0305f7741659304f5325f3bf6d-Paper-Datasets_and_Benchmarks_Track.pdf)
- [30] T. Ma, B. Bai, H. Lin, H. Wang, Y. Wang, L. Luo, and L. Fang, When visual grounding meets gigapixel-level large-scale scenes: Benchmark and approach, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 22 119–22 128.
- [31] W. Yu, W. Chen, G. Qi, W. Li, Y. Li, L. Sha, D. Xia, and J. Huang, Bbox docvqa: A large scale bounding box grounded dataset for enhancing reasoning in document visual question answer, 2025. [Online]. Available: <https://arxiv.org/abs/2511.15090>
- [32] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. [Online]. Available: <https://arxiv.org/abs/2305.06500>
- [33] Y. Yang, A. Patel, M. Deitke, T. Gupta, L. Weihs, A. Head, M. Yatskar, C. Callison-Burch, R. Krishna, A. Kembhavi et al., Scaling text-rich image understanding via code-guided synthetic multimodal data generation, *arXiv preprint arXiv:2502.14846*, 2025.
- [34] Y. Jia, J. Li, X. Yue, B. Li, P. Nie, K. Zou, and W. Chen, Visualwebinstruct: Scaling up multimodal instruction data through web search, *arXiv preprint arXiv:2503.10582*, 2025.
- [35] S. Liu, Z. Li, T. Li et al., Grounding dino: Marrying dino with grounded pre-training for open-set object detection, in *ICCV*, 2023.
- [36] A. Kirillov, E. Mintun, N. Ravi et al., Segment anything, *arXiv preprint arXiv:2304.02643*, 2023.
- [37] D. Wan, J. Cho, E. Stengel-Eskin, and M. Bansal, Contrastive region guidance: Improving grounding in vision-language models without training, 2024. [Online]. Available: <https://arxiv.org/abs/2403.02325>
- [38] J. Liu and R. Hu, Zero-shot referring expression comprehension via vision-language true/false verification, 2025. [Online]. Available: <https://arxiv.org/abs/2509.09958>

- [39] M. Wu, J. Ji, O. Huang, J. Li, Y. Wu, X. Sun, and R. Ji, Evaluating and analyzing relationship hallucinations in large vision-language models, 2024. [Online]. Available: <https://arxiv.org/abs/2406.16449>
- [40] Q. Wu, X. Yang, Y. Zhou, C. Fang, B. Song, X. Sun, and R. Ji, Grounded chain-of-thought for multimodal large language models, 2025. [Online]. Available: <https://arxiv.org/abs/2503.12799>
- [41] G. Sarch, S. Saha, N. Khandelwal, A. Jain, M. J. Tarr, A. Kumar, and K. Fragkiadaki, Grounded reinforcement learning for visual reasoning, 2025. [Online]. Available: <https://arxiv.org/abs/2505.23678>
- [42] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, React: Synergizing reasoning and acting in language models, in *The eleventh international conference on learning representations*, 2022.
- [43] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, Reflexion: Language agents with verbal reinforcement learning, *Advances in Neural Information Processing Systems*, vol. 36, pp. 8634–8652, 2023.
- [44] A. Madaan *et al.*, Self-refine: Iterative refinement with self-feedback, *arXiv preprint arXiv:2303.17651*, 2023.
- [45] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, Judging llm-as-a-judge with mt-bench and chatbot arena, *Advances in neural information processing systems*, vol. 36, pp. 46 595–46 623, 2023.
- [46] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [47] D. Gurari, Q. Li, A. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, Vizwiz grand challenge: Answering visual questions from blind people, in *CVPR*, 2018.
- [48] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao *et al.*, Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, *arXiv preprint arXiv:2504.10479*, 2025.
- [49] Z. Lin, D. Pathak, B. Li, J. Li, X. Xia, G. Neubig, P. Zhang, and D. Ramanan, Evaluating text-to-visual generation with image-to-text generation, in *European Conference on Computer Vision*. Springer, 2024, pp. 366–384.
- [50] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, Clipscore: A reference-free evaluation metric for image captioning, in *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021, pp. 7514–7528.
- [51] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, Lora: Low-rank adaptation of large language models. *ICLR*, vol. 1, no. 2, p. 3, 2022.



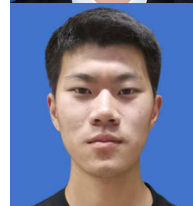
thetic data generation.



measurement, social



networks, multi-task learning, and statistical machine learning. He serves as a peer reviewer for journals including IEEE TNNLS, TITS, ITSC, TCSVT, EAAI, Information Fusion, and Information Sciences.



**Rongsheng Hu** is currently a Ph.D. student with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He received the B.S. degree from Jiangnan University. His research interests include visual question answering, multimodal learning, visual grounding, and synthetic data generation.

**Yuan Liu** is currently a Professor and Doctoral Supervisor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He is a Senior Member of CCF and serves as an editorial board member and reviewer for several journals. His research interests include network traffic measurement, social networks, and digital media.

**Runwei Guan** (Student Member, IEEE) received the M.S. degree in Data Science from the University of Southampton, U.K., in 2021, and the Ph.D. degree from the University of Liverpool. He is currently a postdoctoral researcher. His research interests include multimodal perception, lightweight neural networks, multi-task learning, and statistical machine learning. He serves as a peer reviewer for journals including IEEE TNNLS, TITS, ITSC, TCSVT, EAAI, Information Fusion, and Information Sciences.

**Yicheng Di** is currently a Ph.D. student with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. His research interests include multimodal learning and recommender systems.

**Jiayu Bao** is currently a Ph.D. student with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. His research interests include recommender systems and federated learning.

**Yi Jin** is currently a Ph.D. student with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. Her research interests include multimodal learning and machine learning.



machine learning.

**Hongjian Shi** is currently a Ph.D. student with the School of Computer Science, Shanghai Jiao Tong University, Shanghai, China. He received the B.E. degree in Electronic Engineering from Shanghai Jiao Tong University. His research interests center around the efficiency and scheduling of distributed



**Yuan Hong** is the CEO of UGO-AI Intelligent Technology (Shanghai) Co., Ltd. With over 30 years of experience in the automotive industry, he is currently mainly researching data model training under the integrated architecture of autonomous driving vehicles, road, and cloud.



application of key technologies in intelligent transportation systems.

**Lin Yu** is the General Manager of Shanghai Jinqiao Intelligent Connected Vehicle Development Co., Ltd. He received the Ph.D. degree in Traffic Information Engineering and Control from Tongji University in 2006. As a Professorate Senior Engineer, he has long been engaged in the R&D and industrial ap-



AI systems, and machine learning.

**Ruhui Ma (Corresponding Author)** is currently an Associate Professor with the School of Computer Science, Shanghai Jiao Tong University, Shanghai, China. He received the Ph.D. degree in Computer Science from Shanghai Jiao Tong University. His research interests include cloud computing systems,



engineering from Monash University, Melbourne, Australia, in 2002. He has authored more than 625 publications and seven textbooks, including *Mastering Cloud Computing*. He is one of the highly cited authors in computer science and software engineering worldwide. His research interests include cloud computing and distributed systems.

**Rajkumar Buyya (Corresponding Author)** (Fellow, IEEE) is a Redmond Barry Distinguished Professor and the Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, the University of Melbourne, Australia. He received the Ph.D. degree in Computer Science and Software En-