Look Before You Reason: Implicit Visual Thinking for Efficient Multimodal Reasoning

Zerui Chen, Changrui Chen, Fei Ni, and Jiankang Deng
MVP Lab, Imperial College London
https://zerchen.github.io/projects/lore.html

Abstract

Vision-language models (VLMs) have advanced rapidly in multimodal reasoning, yet they remain limited on tasks requiring fine-grained understanding of small or easily overlooked image regions. Recent *thinking with images* approaches mitigate this limitation by allowing models to inspect images through operations such as zooming, cropping, or code execution. While effective, their multi-turn interactions with VLMs introduce redundant token computation and slow inference, and separate region inspection from the reasoning step that depends on it. To address these issues, we propose **Lo**ok Before You **Re**ason (LoRe), an implicit visual thinking framework for efficient visual reasoning within a single-turn VLM interaction. Given an image and a question, LoRe predicts the most relevant image region, aligns it with the corresponding image patch features, and strengthens the associated key-value (KV) cache entries, enabling subsequent decoding to directly exploit the selected visual cues without additional visual inputs or explicit tool calls. We further employ reinforcement learning to guide the VLM toward identifying informative image regions and leveraging them during reasoning. To support fine-grained evaluation of multimodal reasoning, we introduce COCO-ObjQA, a large-scale benchmark for object-centric multimodal reasoning. Experiments on three challenging multimodal benchmarks demonstrate that LoRe consistently improves VLM reasoning performance with low additional inference overhead.

1 Introduction

Recent vision-language models (VLMs) have made rapid progress in multimodal reasoning, yet they continue to struggle when answers depend on small, localized, or easily overlooked image regions. To address this limitation, the emerging *thinking with images* paradigm [20, 35, 36, 45, 48, 54, 58] treats visual input not as static context, but as evidence to be actively processed through operations such as zooming, cropping, or the use of other visual tools. By leveraging multimodal chain-of-thought (CoT) [46] elicited through supervised fine-tuning (SFT) or reinforcement learning (RL) [11, 38], these approaches have demonstrated promising performance on diverse multimodal reasoning tasks.

Despite their improved reasoning capabilities, existing *thinking with images* approaches remain far from efficient. At certain interaction steps with the VLM, these approaches explicitly issue a tool call to request a visual operation such as zooming or cropping, then wait for the operation to complete and receive the processed visual observation before resuming reasoning. Such back-and-forth between the VLM and external visual tools repeats across multiple turns until a final answer is reached.

This paradigm suffers from two fundamental limitations. First, it is computationally expensive: intermediate visual observations and reasoning chains must be tokenized and processed by the VLM in subsequent turns, leading to substantial token consumption and high inference latency. Second, it introduces a temporal mismatch between visual inspection and reasoning. When the model decides to examine a specific image region, the resulting visual evidence only becomes available in the next

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

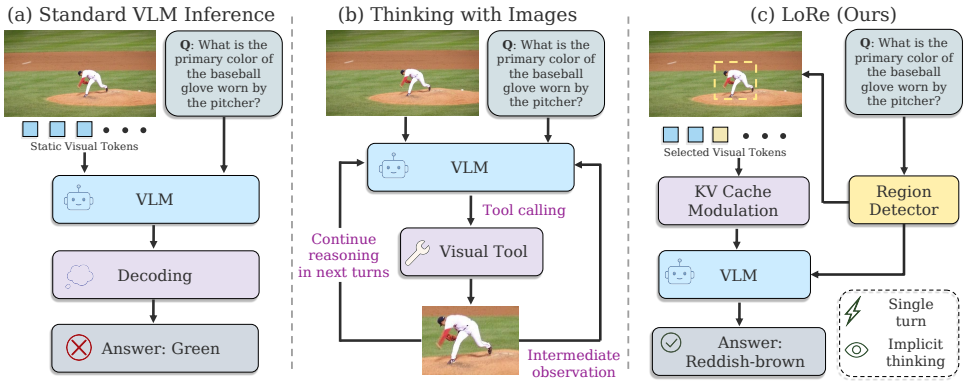


Figure 1: **Comparison of visual reasoning paradigms.** (a) Conventional VLMs rely on static visual tokens, which limits reasoning ability, particularly for small regions. (b) *thinking with images* approaches depend on costly multi-turn visual manipulation, incurring significant inference overhead. (c) LoRe achieves efficient single-turn implicit visual reasoning via lightweight KV cache modulation.

interaction turn, rather than in the current reasoning step that prompted the inspection. Consequently, critical visual evidence is incorporated too late to support efficient multimodal reasoning.

To alleviate these limitations, as shown in Figure 1(c), we introduce **Lo**ok Before You **Re**ason (LoRe), a novel implicit thinking algorithm for efficient reasoning requiring only a single interaction turn with VLMs. Given an image and question, LoRe first identifies the most relevant region and strengthens its visual evidence during subsequent token decoding. Unlike prior methods relying on external tools and multiple interaction turns to generate intermediate visual inputs, LoRe leverages VLMs' inherent capabilities to unify visual manipulation and reasoning within a single inference process.

Specifically, LoRe performs implicit thinking through lightweight key-value (KV) cache modulation. After identifying the most relevant region, LoRe maps it to corresponding image patch features and amplifies their entries in the KV cache. It allows subsequent decoding to focus more on the selected visual evidence. In this way, crucial visual evidence can directly influence the current reasoning step, without modifying the input image or introducing additional visual tokens. We further train VLMs with RL to encourage the model to identify the region that supports the final answer, enabling it to learn when and where to look while immediately leveraging the discovered evidence for reasoning.

To comprehensively evaluate LoRe's reasoning ability, we construct COCO-ObjQA, an object-centric benchmark derived from object-level annotations in the COCO dataset [24]. Existing benchmarks such as V* [49] and HR-Bench [42] remain limited in scale and coverage. In contrast, COCO-ObjQA contains more than 8,000 evaluation samples, enabling controlled evaluation of a model's ability to reason over visual targets of varying sizes. It complements existing benchmarks and more reliably exposes failure cases, particularly for small objects. Extensive experiments demonstrate that LoRe consistently improves VLM reasoning across three challenging benchmarks with low inference overhead. Our contributions are summarized as follows:

- We propose LoRe, an implicit thinking framework that unifies visual manipulation and reasoning within a single inference turn. By amplifying informative image tokens through KV cache modulation, LoRe enables efficient reasoning without explicit tool calls or additional visual tokens.
- We employ reinforcement learning to train VLMs to identify the most task-relevant image region and immediately leverage the discovered visual evidence within the same reasoning turn, enabling grounded and efficient multimodal reasoning without temporal mismatch.
- We curate COCO-ObjQA, a large-scale benchmark to enable controlled evaluation of fine-grained visual reasoning. Extensive experiments demonstrate that LoRe consistently improves VLM reasoning abilities across three challenging multimodal benchmarks with low inference overhead.

2 Related Work

Large Vision-language Models. Recent years have witnessed remarkable progress in large vision-language models [3, 18, 25, 43] (VLMs). State-of-the-art VLMs typically consist of a vision encoder [13, 14, 40, 50] and a large language model [2, 10, 39] (LLM). Early VLMs, such as

Flamingo [1] and BLIP [21, 22], adopt the classic encoder-decoder architecture [41], where visual features are first encoded and textual representations cross-attend to them at each token-generation step, leading to relatively slow inference. By contrast, recent VLMs [3, 18, 25, 43] typically employ a simpler decoder-only architecture: visual features are projected into the language embedding space and processed jointly with text tokens via self-attention for autoregressive generation. This architectural paradigm is compatible with highly optimized LLM inference engines [17, 57], thereby substantially improving inference efficiency. Despite impressive progress, existing VLMs still exhibit limited reasoning ability [7, 9], as they largely treat visual content as static context for the language model [34]. As a result, the model reasons over compressed visual tokens without actively revisiting or emphasizing task relevant regions during inference.

Reasoning in Vision-language Models. To improve the reasoning ability of VLMs, recent studies have explored multimodal chain-of-thought reasoning [46], often elicited through supervised fine-tuning or reinforcement learning [11, 38]. Earlier efforts mainly focus on eliciting step-by-step textual rationales or decomposing complex visual questions into intermediate reasoning steps, thereby encouraging VLMs to perform more deliberate inference over visual inputs [44, 56]. Orthogonal to prompting-based reasoning, another line of work improves visual reasoning by enhancing fine-grained perception, for example through high-resolution encoding, adaptive image partitioning, or region-level visual grounding [27, 42, 49]. Inspired by OpenAI o3 [29], the emerging *thinking with images* paradigm treats visual inputs as dynamic evidence to be actively inspected during reasoning. Existing methods typically rely on external tools [35, 47, 58], executable programs [15, 37, 55], or generative imagination [8, 19, 52] to manipulate visual information. These methods share the intuition that visual reasoning should not be performed solely over a fixed set of compressed image tokens, but should instead allow the model to seek additional visual evidence when needed. While effective, these explicit *thinking with images* approaches usually require multi-turn interactions, where intermediate visual observations and reasoning traces are repeatedly generated, tokenized, and processed by VLMs. This leads to substantial token overhead, high inference latency, and a temporal mismatch between visual inspection and reasoning. LoRe instead predicts a region and amplifies its existing value-cache entries within the same generation request. This targets evidence selection within an encoded image rather than the acquisition of new visual observations.

Attention and representation steering. PASTA [53] reweights selected LLM attention heads toward user-specified text. ClearSight [51] enhances visual signals during middle-layer modality fusion to reduce object hallucination, while ICT [6] applies image- and object-level activation shifts to selected head outputs. LoRe shares the goal of strengthening relevant evidence but predicts a question-conditioned spatial region online and updates its live value-cache entries. It does not directly reweight attention probabilities or require user-marked regions. Decoding resumes in the same request with native optimized attention kernels.

3 Methodology

LoRe combines question-conditioned region selection with value-cache modulation within a single generation request. Section 3.1 describes this mechanism, Section 3.2 introduces its RL training procedure, and Section 3.3 presents COCO-ObjQA for evaluation across target-object sizes.

3.1 Implicit visual thinking

Preliminaries for VLM decoding. Figure 2 gives an overview of LoRe. The VLM prediction process generally consists of two phases: prefilling and autoregressive generation. During prefilling, the VLM tokenizes the multimodal prompt into a single sequence of visual and text tokens, and each transformer layer stores the key-value representations of all prompt tokens in its KV cache. During autoregressive generation, the model forwards only the newly generated token at each step, attends to the cached prefix, and appends the new token's key-value representations to the cache. For a prompt of length m, the layer-ℓ cache after prefilling and after generating t tokens can be written as:

$$C_\ell^{(0)} = \{(k_{\ell,i}, v_{\ell,i})\}_{i=1}^{m}, \qquad C_\ell^{(t)} = C_\ell^{(t-1)} \cup \{(k_{\ell,m+t}, v_{\ell,m+t})\}.$$

This conventional mechanism makes decoding efficient, but it also fixes the visual evidence after prefilling. Subsequent tokens can attend to the static visual features, and the model has no native operations for amplifying the influence of a region that becomes salient during reasoning.

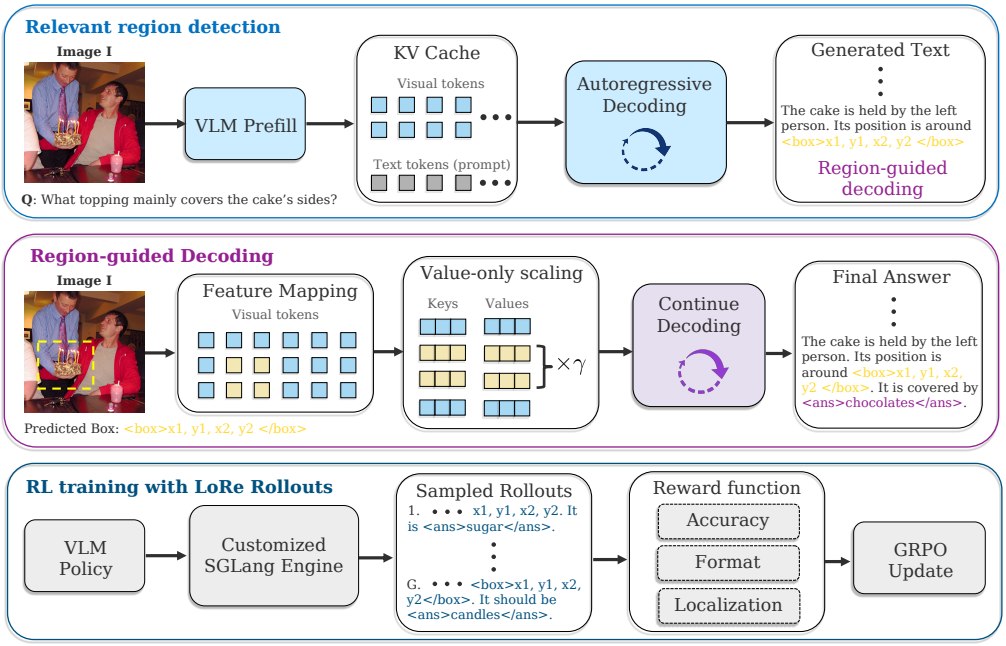


Figure 2: **Overview of LoRe.** The VLM emits a region as coordinate tokens. The region detector denotes this generation-and-parsing process, not a separate detection network. LoRe maps the box to visual-token positions and scales only their value-cache entries before decoding continues in the same request. Keys, position IDs, and token order remain unchanged. No crop, additional visual tokens, or second VLM call is required. RL optimizes these trajectories with answer, format, and localization rewards.

Relevant region prediction. Given an image I and a question Q, we append an instruction asking the VLM to first localize the relevant region and then answer. The VLM itself generates ordinary text tokens in the format `<box>x1, y1, x2, y2</box>`, followed by reasoning and an `<ans>...</ans>` answer span. There is no separate detection head or external detector. Qwen2.5-VL uses image-pixel coordinates, while Qwen3-VL uses its native $[0, 1000]$ coordinate convention. Thus, the region control signal is explicit, whereas visual focusing is performed internally on the cache.

After prefilling, a lightweight processor checks the generated sequence every four tokens, searching the latest 128 tokens for a complete box span. It parses the coordinates as $R_q = (x_1, y_1, x_2, y_2)$, converts them to image coordinates, and clamps them to the image boundary. The default configuration applies one cache update when a usable region is found, then resumes the same request.

If the span is missing or malformed, or the clamped box cannot be mapped to visual tokens, the cache remains unchanged and standard decoding continues without retrying or discarding the sample. All reported accuracies include these fallback cases. A valid but inaccurate box still triggers modulation and can amplify irrelevant evidence.

Region-guided decoding. Once the relevant region R_q is detected, LoRe uses it to guide the remaining decoding process without altering the image input or starting a new VLM call. The key step is to convert the predicted box into the visual-token positions that have already been stored in the KV cache during prefilling. Let $\mathcal{G} = (H, W)$ be the effective visual-token grid after patch merging, (W_I, H_I) be the image size, and p_i be the sequence position of the i-th visual token. For $R_q = (x_1, y_1, x_2, y_2)$, LoRe first converts normalized coordinates to the image coordinate system if needed, clamps the box to the image boundary, and selects the covered grid cells:

$$\mathcal{P}(R_q) = \{p_{vW+u} \mid u \in U_q,\ v \in V_q\}.$$

Here $U_q = [\lfloor x_1 W/W_I \rfloor, \lceil x_2 W/W_I \rceil)$ and $V_q = [\lfloor y_1 H/H_I \rfloor, \lceil y_2 H/H_I \rceil)$ denote the horizontal and vertical grid ranges, with zero-based grid indices. We read `image_grid_thw` after dynamic-resolution preprocessing, account for spatial patch merging, and verify that the effective grid matches the visual-token count before mapping cells to row-major sequence positions. The mapping therefore adapts to image size and aspect ratio rather than assuming a fixed grid. Very small regions remain subject to patch quantization.

Given $\mathcal{P}(R_q)$, LoRe scales the selected value-cache entries once in every decoder layer $\ell \in \mathcal{L}$, leaving the keys, position IDs, token order, and attention masks unchanged:

$$v'_{\ell,j} = \begin{cases} \gamma v_{\ell,j}, & j \in \mathcal{P}(R_q), \\ v_{\ell,j}, & j \notin \mathcal{P}(R_q), \end{cases} \qquad \gamma \geq 1.$$

γ is the amplification factor, with $\gamma = 1$ disabling modulation. For a fixed query q, let $\alpha_j = \mathrm{softmax}_j(q^\top k_j/\sqrt{d})$ and $o = \sum_j \alpha_j v_j$, omitting layer and head indices. The intervention gives

$$o' = o + (\gamma - 1) \sum_{j \in \mathcal{P}(R_q)} \alpha_j v_j.$$

Thus, value scaling amplifies attended evidence without directly changing α_j or rotary-encoded keys. Later attention can change indirectly as the modified residual state affects subsequent queries. In the pre-normalized decoder, subsequent normalization constrains magnitude but does not undo changes in residual direction or composition. All unselected tokens remain available, although excessive scaling can overwhelm global context. This is evidence amplification, not a recovery of details lost during vision encoding. Tokens with negligible query–key affinity remain weak contributors.

Value scaling is a linear intervention compatible with native optimized attention kernels. Scaling keys would also change logits in a query-dependent manner, increasing positive logits while making negative ones more negative. An additive logit bias offers direct attention reweighting but requires a different attention implementation. LoRe uses value-only scaling by default to preserve the existing attention computation and keep overhead low.

3.2 Reinforcement learning for LoRe

Policy rollouts with LoRe. The goal of RL training is to make the model learn a coupled behavior: it should predict a region that supports the question, and it should benefit from that region when generating the final answer. Therefore, rollouts must use the same inference procedure described in Section 3.1, rather than treating localization as an auxiliary output that is detached from decoding. Each training sample contains an image-question pair $x = (I, Q)$, a ground-truth answer a, and metadata such as answer type τ and problem type z. The policy first produces a region span and then continues generation under LoRe's region-guided cache modulation, so the final response is evaluated after the model has actually used the detected visual evidence.

To implement this rollout process efficiently, we extend the SGLang engine [57] with a dynamic value-boost processor. During generation, the processor monitors the token stream, parses completed box spans, maps the detected region to visual-token positions, applies the cache modulation inside the inference backend, and then resumes decoding in the same request. For each prompt, we sample G trajectories from the VLM policy $\pi_{\theta_{\mathrm{old}}}$, which provides the training data for RL.

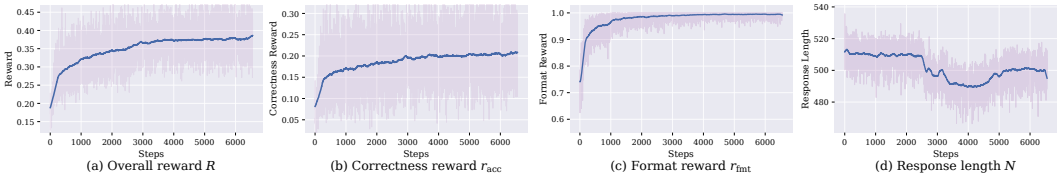
Reward function. Following recent visual RL studies that rely on verifiable rewards [47], we use a rule-based reward that prioritizes final-task correctness while providing weak supervision for relevant region detection. The main term $r_{\mathrm{acc}}(y, a, \tau)$ evaluates the final answer according to the answer type τ: multiple-choice answers are judged by the final answer, while numerical and OCR answers use their own matching rules. This term provides the dominant impact on LoRe's reasoning ability.

We add two auxiliary terms to stabilize the LoRe trajectory. The format reward r_{fmt} checks for exactly one non-empty `<ans>...</ans>` span and balanced optional reasoning delimiters. Region supervision comes from r_{loc}: it gives a box-presence bonus for visually grounded samples and adds parseability and center-proximity terms for bbox-supervised samples. Thus, the VLM learns coordinate generation through these rewards rather than a separate detector loss. The final reward is:

$$R = \min\{1,\ r_{\mathrm{acc}} + (1 - r_{\mathrm{acc}})(0.03\, r_{\mathrm{fmt}} + r_{\mathrm{loc}})\}.$$

This residual form prioritizes correctness: correct answers are capped at one while localization bonuses remain bounded, preventing them from dominating the answer reward.

GRPO update. We adopt Group Relative Policy Optimization (GRPO) [11], following recent reinforcement learning approaches for *thinking with images* [35, 47, 48, 58]. For each prompt x, we sample a group of LoRe trajectories $\{y_i\}_{i=1}^G$ from $\pi_{\theta_{\mathrm{old}}}$ and compute rewards $\{R_i\}_{i=1}^G$. The reward of each trajectory is converted into a clipped score $S_i = \mathrm{clip}(\alpha(R_i + b), -c, c)$ and then normalized



(a) Overall reward R (b) Correctness reward r_{acc} (c) Format reward r_{fmt} (d) Response length N

Figure 3: **GRPO training dynamics for LoRe with Qwen2.5-VL-7B [4].** The increasing correctness reward with stable response length indicates that RL improves the LoRe's reasoning ability.

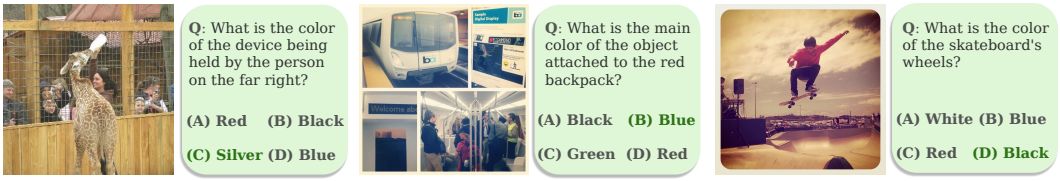


Figure 4: **Qualitative examples from COCO-ObjQA.** Boxes visualize target COCO annotations across object scales and are not part of the evaluation input. Models receive the original full image and the multiple-choice question.

within the same prompt group to obtain $A_i = (S_i - \mu_x)/(\sigma_x + \epsilon)$. This relative advantage measures whether a trajectory finds and uses visual evidence better than other candidates for the same question. With $\rho_{i,t} = \pi_\theta(y_{i,t} \mid x, y_{i,<t})/\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})$ and $\bar{\rho}_{i,t} = \text{clip}(\rho_{i,t}, 1-\epsilon_\ell, 1+\epsilon_h)$, we minimize:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}\left[\frac{1}{G}\sum_{i,t} T_i^{-1} \min(\rho_{i,t}A_i, \bar{\rho}_{i,t}A_i)\right] + \beta D_{\text{KL}}.$$

Here the expectation is over prompts and rollout groups, T_i is the number of generated tokens in trajectory y_i, and D_{KL} is the token-level regularizer against a frozen reference policy. We recompute log probabilities before each update, use asymmetric clipping bounds, and filter groups whose rewards are identical, since they provide no relative learning signal. This objective directly couples the localization decision with the final answer reward, while the value-boost rollout engine ensures that rewards are assigned to trajectories that actually use LoRe during generation.

Figure 3 shows the training curves for Qwen2.5-VL-7B [4]. The overall reward and correctness reward steadily increase, indicating that GRPO improves task-level reasoning rather than only fitting the required response format. The format reward saturates early, which suggests that the model quickly learns to produce parseable spans for the rollout engine. Meanwhile, the response length remains stable, indicating that LoRe better utilizes visual evidence and learns to reason efficiently.

3.3 COCO-ObjQA benchmark

LoRe focuses on cases where the answer is visible but requires selecting a relevant object or attribute. We construct COCO-ObjQA from MS COCO [24] instance annotations to evaluate this ability across target sizes, complementing the smaller V* benchmark [49] and resolution-focused HR-Bench [42]. We retain targets whose category appears once in the image to reduce instance ambiguity. Gemini 3 [30] receives the full image, target crop, category, and box to generate a multiple-choice question. Validation includes structural checks, independent GPT-5.4 checks of answerability and label correctness from the full image, and targeted manual inspection of high-risk samples. Confirmed failures are removed. Evaluation models receive only the original full image and question, including its answer options, without boxes, crops, or highlighted regions. COCO-ObjQA contains 8,537 QA pairs from 4,107 images across all 80 COCO categories. By target-area ratio (COCO box area divided by image area), 1,743 examples are small ($\leq$1%), 3,036 medium (1–10%), and 3,758 large ($>$10%). This is a controlled diagnostic of object-centric evidence selection, not a comprehensive test of multi-object relations or image transformations.



Figure 5: **Qualitative effects of LoRe.** LoRe localizes task-relevant visual evidence across different benchmarks before producing the final answer. Yellow boxes denote the region detection results.

4 Experiments

4.1 Experimental setting

Benchmarks and metrics. We evaluate LoRe on three held-out benchmarks that probe complementary forms of fine-grained visual reasoning. V* [49] benchmark focuses on high-resolution visual search, including attribute recognition and relative-position reasoning, while HR-Bench [42] stresses perception under 4K and 8K image resolutions. COCO-ObjQA, introduced in Section 3.3, provides the main object-level diagnostic: each question is tied to a known target instance and is answered from the original full image, allowing us to test whether a method can find and use localized evidence rather than rely on global image priors. We report exact-match accuracy by extracting the final option from each model response. For COCO-ObjQA, we also report accuracy by target-object size, with the small-object split most directly reflecting LoRe's evidence-selection ability.

Training details. We train LoRe using the GRPO objective and rollout procedure described in Section 3.2. The training data contains Ref-L4 [5], ViGoRL-SA and SAT-2 [33], PixMo-Count [12], InfographicVQA [28], and AI2D [16], which provide supervision signals for localization, spatial reasoning, counting, OCR, and diagram understanding. We initialize the policy from Qwen2.5-VL-7B [4] or Qwen3-VL-8B [3] and train for roughly 7K steps on 8 NVIDIA H200 GPUs, which takes about 65 hours. Rollout generation uses our customized SGLang [57] implementation of LoRe, while policy optimization is performed with FSDP. In our default setup, four GPUs serve rollouts, and four GPUs compute actor and reference-policy log probabilities. Unless otherwise stated, we use bfloat16 precision with a batch size of 32, and a maximum response length of 4096. We use Adam with a learning rate of 1×10^{-6}, a KL coefficient of 0.01, and the value boosting factor γ of 2.

Comparison protocol. DeepEyes, ViLaSR, and Thyme are evaluated zero-shot using their official checkpoints under our common evaluation protocol. Their training data and objectives differ from LoRe's. Table 1 therefore compares complete systems rather than isolating the cache intervention under matched training. Latency is measured on the same H200 machine with batch size one, without an external inference engine, and includes each method's visual operations and VLM interactions. For LoRe, this also includes region parsing and KV cache updates. Further details are in Appendix E.

4.2 Main results

Figure 5 shows qualitative results of LoRe across different benchmarks, while Table 1 summarizes the quantitative comparison. We compare different thinking algorithms on V* [49], HR-Bench [42], and COCO-ObjQA, and report latency to quantify the inference overhead.

Representative VLMs still have limited fine-grained reasoning ability. Table 1 first shows that direct VLM inference remains limited on fine-grained visual reasoning, even when the backbone

Table 1: **Main comparison** on fine-grained multimodal reasoning benchmarks. Explicit reasoning baselines use official checkpoints with different training recipes. Gain rows report speedup over the fastest explicit reasoning baseline and relative accuracy gains over the corresponding base VLM. Bold and underlined scores mark the best and second-best results.

Method	**Latency**	**V***			**HR-Bench**			**COCO-ObjQA**			
		Dir. Attr.	Rel. Pos.	Avg.	4K	8K	Avg.	Small	Medium	Large	Avg.
Representative VLMs											
InternVL3-8B [59]	1.25s	72.1	68.3	70.6	62.7	52.4	57.6	79.8	91.7	92.2	89.5
Qwen2.5-VL-7B [4]	1.92s	75.2	68.1	72.4	69.1	62.8	66.0	72.3	90.1	91.2	86.9
Qwen3-VL-8B [3]	2.95s	82.4	80.9	81.8	<u>78.6</u>	<u>74.1</u>	<u>76.4</u>	79.2	92.3	93.1	90.0
State-of-the-art Reasoning Approaches											
ViLaSR-7B [47]	6.05s	46.3	69.7	55.6	60.8	55.7	58.3	35.4	55.1	69.7	57.5
DeepEyes-7B [58]	34.73s	**91.1**	<u>87.8</u>	**89.8**	75.2	72.9	74.1	82.6	91.4	94.0	90.7
Thyme-7B [55]	6.18s	82.7	80.1	81.7	77.9	72.2	75.1	80.2	91.2	93.0	89.7
Ours											
LoRe (Qwen2.5-VL-7B)	2.16s	86.1	77.2	82.6	76.2	71.0	73.6	<u>83.1</u>	<u>93.2</u>	<u>94.2</u>	<u>91.6</u>
Gain / speedup	2.8×	+14.5%	+13.4%	+14.1%	+10.3%	+13.1%	+11.5%	+14.9%	+3.4%	+3.3%	+5.4%
LoRe (Qwen3-VL-8B)	3.06s	<u>88.8</u>	**88.2**	<u>88.6</u>	**82.1**	**78.3**	**80.2**	**84.3**	**94.5**	**95.1**	**92.7**
Gain / speedup	2.0×	+7.8%	+9.0%	+8.3%	+4.5%	+5.7%	+5.0%	+6.4%	+2.4%	+2.1%	+3.0%

is relatively strong. Among representative VLMs, Qwen3-VL-8B performs the best, but it reaches only 81.8% average accuracy on V* and 76.4% on HR-Bench. COCO-ObjQA further makes the object-scale failure mode explicit. Although Qwen3-VL-8B reaches 90.0% average accuracy on COCO-ObjQA, it obtains only 79.2% on small objects, compared with 92.3% and 93.1% on medium and large objects. The same pattern is stronger for Qwen2.5-VL-7B, which drops to 72.3% on small objects while achieving 90.1% and 91.2% on medium and large objects.

Explicit reasoning offers a different accuracy–latency tradeoff. DeepEyes-7B achieves the highest V* average, 89.8%, at 34.73s per sample. ViLaSR-7B is faster at 6.05s but obtains 55.6% on V* and 57.5% on COCO-ObjQA under our protocol. These scores describe the transferred checkpoint, not the capability of crop-based reasoning under matched training. Thyme-7B achieves 75.1% on HR-Bench and 89.7% on COCO-ObjQA at 6.18s. These results show that explicit visual processing can improve accuracy, with benefits that depend on the checkpoint and benchmark, while requiring additional inference overhead.

LoRe improves reasoning while preserving efficient inference. LoRe provides a different route to reasoning gains by turning visual focusing into a cache-side operation within a single interaction turn. With Qwen2.5-VL-7B as the backbone, LoRe improves the V* average from 72.4% to 82.6%, HR-Bench from 66.0% to 73.6%, and COCO-ObjQA from 86.9% to 91.6%. The largest COCO-ObjQA gain appears on small objects, where accuracy increases from 72.3% to 83.1%. With Qwen3-VL-8B, LoRe further reaches the best HR-Bench and COCO-ObjQA results, achieving 80.2% and 92.7% average accuracy, respectively, while also obtaining the second-best V* average at 88.6%. Crucially, these gains do not require explicit image manipulation or an additional VLM call: LoRe (Qwen2.5-VL-7B) runs at 2.16s and is 2.8× faster than the fastest explicit reasoning baseline, while LoRe (Qwen3-VL-8B) remains 2.0× faster. These results show that LoRe improves fine-grained reasoning by reusing existing visual-token caches, avoiding the extra latency of tool-based visual reasoning.

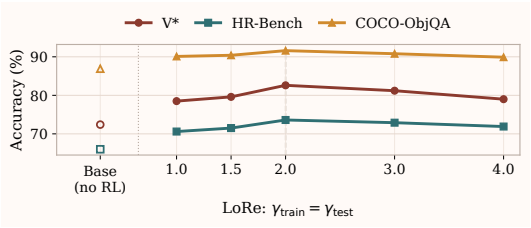
Remaining V* performance gap. On Direct Attribute, LoRe (Qwen2.5-VL-7B) scores 86.1% versus DeepEyes' 91.1%, a 5.0-point gap. The larger 10.6-point gap, 77.2% versus 87.8%, is on Relative Position. Explicit cropping and re-encoding can expose fine detail absent from the initial representation, while a single predicted box may omit a reference object, select a distractor, or inadequately cover distant objects. These limitations help explain the complementary tradeoff: LoRe improves Relative Position over its base model's 68.1%, while running approximately 16× faster than DeepEyes overall.

4.3 Ablation study

We study the training-free pipeline, subsequent RL training, reward terms, amplification strength, and number of focused regions using Qwen2.5-VL-7B and the evaluation protocol of Section 4.2.

Table 2: **Training-free LoRe, RL training, and reward ablations.** LoRe variants use $\gamma = 2$, while the base model uses no modulation. These rows do not constitute a fixed-checkpoint ablation of modulation after RL. COCO-ObjQA$_{\text{IoU}}$ is mean box IoU, with invalid predictions counted as zero.

Ablation model variant	V*$_{\text{avg}}$	HR-Bench$_{\text{avg}}$	COCO-ObjQA$_{\text{small}}$	COCO-ObjQA$_{\text{avg}}$	COCO-ObjQA$_{\text{IoU}}$
Implicit reasoning ablations					
Qwen2.5-VL-7B base	72.4	66.0	72.3	86.9	0.24
+ implicit thinking, w/o RL	77.4	70.2	78.1	89.4	0.31
+ implicit thinking and RL (LoRe)	**82.6**	**73.6**	**83.1**	**91.6**	**0.43**
Reward-term ablations					
Full reward	**82.6**	**73.6**	**83.1**	**91.6**	**0.43**
w/o answer reward r_{acc}	70.1	65.3	70.2	84.8	0.40
w/o answer-format reward r_{fmt}	80.1	72.8	81.2	90.5	0.39
w/o grounding reward r_{loc}	78.2	70.8	78.4	89.2	0.28



Number of focused regions

Setting	V*$_{\text{avg}}$	HR-Bench$_{\text{avg}}$	COCO-ObjQA$_{\text{avg}}$
Base (no RL)	72.4	66.0	86.9
$K = 1$	**82.6**	**73.6**	**91.6**
$K = 2$	81.5	73.6	91.3
$K = 3$	80.2	72.9	90.5
Oracle box	84.1	75.8	93.0

Figure 6: **KV cache modulation configurations.** Left: LoRe policies are trained and evaluated with the same indicated γ. The non-RL base reference is displayed separately. Right: one region performs best overall among learned settings. The oracle row substitutes reference boxes using the same RL checkpoint and $\gamma = 2$. This comparison highlights the potential for more accurate region localization to further improve reasoning performance of LoRe.

Training-free focusing and subsequent RL. Table 2 compares the base VLM, training-free LoRe, and LoRe after RL. The training-free pipeline improves V* from 72.4% to 77.4%, HR-Bench from 66.0% to 70.2%, and COCO-ObjQA small objects from 72.3% to 78.1%. RL raises these scores to 82.6%, 73.6%, and 83.1%, respectively, and the COCO-ObjQA average from 89.4% to 91.6%. Mean COCO-ObjQA IoU increases from 0.24 to 0.31 and then 0.43. These comparisons show gains from the training-free pipeline and from training within it, but they do not isolate test-time modulation from RL using a fixed trained checkpoint.

Contribution of each reward term. Removing r_{acc} in Table 2 causes the largest accuracy drop, reducing V* from 82.6% to 70.1% and COCO-ObjQA from 91.6% to 84.8%, supporting answer correctness as the main learning signal. Removing r_{fmt} causes smaller, consistent losses. Without r_{loc}, mean COCO-ObjQA IoU falls from 0.43 to 0.28, while accuracy drops by 4.4, 2.8, and 2.4 percentage points on V*, HR-Bench, and COCO-ObjQA. Thus, grounding supervision benefits both localization and final answers, beyond learning the answer format alone.

KV cache modulation strategies. In the left panel of Figure 6, each LoRe point uses a policy trained and evaluated with the indicated factor, $\gamma_{\text{train}} = \gamma_{\text{test}}$. The $\gamma = 1$ point uses RL without cache modulation during training or inference. The non-RL base reference is shown separately. This comparison therefore varies the training and inference configurations jointly. Among the evaluated LoRe settings, $\gamma = 2$ gives the best accuracy: 82.6% on V*, 73.6% on HR-Bench, and 91.6% on COCO-ObjQA. Larger factors reduce accuracy, consistent with excessive local amplification weakening the relative contribution of global context.

The right panel compares one, two, and three independently predicted regions, alongside the non-RL base reference. A single region gives the best learned result overall: V* accuracy decreases from 82.6% to 81.5% and 80.2% with two and three regions. Two regions match one region on HR-Bench but perform worse on COCO-ObjQA. Additional regions may introduce distractors rather than useful evidence. A single enclosing box can cover multiple objects, but cannot weight them separately and may include irrelevant context between distant objects.

For the oracle-box comparison, we keep the RL checkpoint, $\gamma = 2$, and decoding procedure fixed and replace predicted regions with COCO annotations on COCO-ObjQA or SAM2-derived [32] pseudo-ground-truth regions on V* and HR-Bench. The respective gains are 1.4, 1.5, and 2.2 percentage points. This comparison with reference regions suggests that more accurate localization could further improve LoRe's reasoning performance. It is not a strict upper bound, especially when reference boxes are approximate, and does not attribute all remaining errors to localization.

Table 3: **Controlled comparisons with Qwen2.5-VL-7B.** The two $\gamma_{\text{train}} = 2$ rows share a checkpoint. Crop-based GRPO matches LoRe's backbone, training data, recipe, and objectives. All scores are average accuracy (%).

Policy	γ_{train}	γ_{test}	**V***	**HR-Bench**	**COCO-ObjQA**
RL, no modulation	1	1	78.5	70.6	90.1
LoRe, modulation disabled	2	1	76.9	69.8	90.6
LoRe, default	2	2	**82.6**	**73.6**	91.6
Crop-based GRPO	–	–	80.9	71.8	**92.3**

Controlling for RL and training data. Table 3 separates the effects of training and test-time modulation. Training and testing at $\gamma = 1$ improve over the base VLM, confirming gains from RL alone. For the checkpoint trained at $\gamma = 2$, disabling modulation preserves the box output, prompt, and decoding procedure. Restoring modulation improves V*, HR-Bench, and COCO-ObjQA accuracy by 5.7, 3.8, and 1.0 percentage points, respectively. We also train a crop-based GRPO policy with the same backbone, data, recipe, and objectives. It predicts one box and re-encodes the crop as a second visual observation before answering. LoRe exceeds this control by 1.7 points on V* and 1.8 on HR-Bench, while trailing by 0.7 on COCO-ObjQA, without additional visual encoding. Appendix F provides further modulation, fallback, localization, and transfer analyses.

Localization reliability. RL reduces fallback rates from 11.0% to 6.8% on V*, 8.5% to 5.0% on HR-Bench, and 2.5% to 1.5% on COCO-ObjQA (Appendix F.3). At $\gamma = 2$, oracle boxes correct 20.7%, 16.3%, and 38.1% of predicted-box errors, respectively (Appendix F.4). Remaining errors indicate that accurate localization alone does not guarantee correct reasoning.

5 Conclusion

We present LoRe, a framework for single-turn visual focusing and multimodal reasoning. The VLM emits a question-relevant region, and LoRe amplifies its visual value-cache entries before generation continues in the same request. GRPO training couples region selection with answer correctness, response format, and visual grounding. COCO-ObjQA complements V* and HR-Bench by enabling object-centric evaluation across target sizes. Across these benchmarks, LoRe improves the corresponding base VLMs while adding little inference overhead, offering a useful accuracy–latency tradeoff relative to explicit visual-tool methods. Controlled comparisons further show that cache modulation contributes beyond the gains from RL alone.

Limitations and future work. LoRe amplifies encoded evidence, but it cannot recover discarded detail or replace operations that transform and re-encode an image. A single region may omit relevant objects or include distractors between distant objects, and multiple independent regions do not consistently improve the reported results. COCO-ObjQA emphasizes single-target evidence selection rather than comprehensive relational reasoning. More reliable multi-object localization and selective use of crop or zoom tools are promising extensions, but a hybrid routing mechanism is outside the present single-turn setting.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge Prof. Stefanos Zafeiriou for his valuable insights and discussions on tiny object perception with multimodal large language models. This work was supported by the Turing AI Fellowship (EP/Z534699/1). Fei Ni was supported by a MSCA Postdoctoral Fellowship. Jiankang Deng was supported by the NVIDIA Academic Grant. The authors gratefully acknowledge the computing resources provided by Isambard-AI, part of the National AI Research Resource (AIRR).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, et al. Qwen technical report. *arXiv:2309.16609*, 2023.

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

[4] 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:2502.13923*, 2025.

[5] Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S.-H. Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. In *CVPR Workshops*, 2025.

[6] Junzhe Chen, Tianshu Zhang, Shiyu Huang, Yuwei Niu, Linfeng Zhang, Lijie Wen, and Xuming Hu. ICT: Image-object cross-level trusted intervention for mitigating object hallucination in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4209–4221, 2025.

[7] Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. Measuring and improving chain-of-thought reasoning in vision-language models. In *NAACL*, 2024.

[8] Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv:2505.22525*, 2025.

[9] Yasser Dahou, Ngoc Dung Huynh, Phuc H Le-Khac, Wamiq Reyaz Para, Ankit Singh, and Sanath Narayan. Vision-language models can't see the obvious. In *ICCV*, 2025.

[10] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-V3 technical report. *arXiv:2412.19437*, 2024.

[11] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

[12] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[14] Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor G Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. In *CVPR*, 2025.

[15] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.

[16] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *SOSP*, 2023.

[18] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv:2408.03326*, 2024.

[19] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. In *ICML*, 2025.

[20] Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. DyFo: A training-free dynamic focus visual search for enhancing LMMs in fine-grained visual understanding. In *CVPR*, 2025.

[21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[23] Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Chen Wei, Konstantinos Psounis, and Kaipeng Zhang. TIR-Bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv:2511.01833*, 2025.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[26] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, 2024.

[27] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. In *ICLR*, 2025.

[28] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *WACV*, 2022.

[29] OpenAI. OpenAI o3 and o4-mini System Card, 2025.

[30] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with Gemini 3. `https://blog.google/products-and-platforms/products/gemini/gemini-3/`, 2025. Accessed: 2026-05-07.

[31] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, Yifan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

[33] Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J. Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. In *NeurIPS*, 2025.

[34] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. In *NeurIPS*, 2025.

[35] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. OpenThinkIMG: Learning to think with images via visual tool reinforcement learning. *arXiv:2505.08617*, 2025.

[36] Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv:2506.23918*, 2025.

[37] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via python execution for reasoning. In *ICCV*, 2023.

[38] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[39] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

[40] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv:2502.14786*, 2025.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[42] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *AAAI*, 2025.

[43] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv:2508.18265*, 2025.

[44] Xiyao Wang, Yuhang Zhou, Xiaoyu Liu, Hongjin Lu, Yuancheng Xu, Feihong He, Jaehong Yoon, Taixi Lu, Fuxiao Liu, Gedas Bertasius, Mohit Bansal, Huaxiu Yao, and Furong Huang. Mementos: A comprehensive benchmark for multimodal large language model reasoning over image sequences. In *ACL*, 2024.

[45] Yikun Wang, Siyin Wang, Qinyuan Cheng, Zhaoye Fei, Liang Ding, Qipeng Guo, Dacheng Tao, and Xipeng Qiu. VisuoThink: Empowering LVLM reasoning with multimodal tree search. In *ACL*, 2025.

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[47] Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *NeurIPS*, 2025.

[48] Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. VTool-R1: VLMs learn to think with images via reinforcement learning on multimodal tool use. In *ICLR*, 2026.

[49] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, 2024.

[50] Yin Xie, Kaicheng Yang, Xiang An, Kun Wu, Yongle Zhao, Weimo Deng, Zimin Ran, Yumeng Wang, Ziyong Feng, Roy Miles, Ismail Elezi, and Jiankang Deng. Region-based cluster discrimination for visual representation learning. In *ICCV*, 2025.

[51] Hao Yin, Guangzong Si, and Zilei Wang. ClearSight: Visual signal enhancement for object hallucination mitigation in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14625–14634, 2025.

[52] Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, Tieniu Tan, and Furu Wei. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in mllms. *arXiv:2510.24514*, 2025.

[53] Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for LLMs. In *International Conference on Learning Representations*, 2024.

[54] Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv:2505.15436*, 2025.

[55] Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. In *ICLR*, 2026.

[56] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024.

[57] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *NeurIPS*, 2024.

[58] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *ICLR*, 2026.

[59] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv:2504.10479*, 2025.

A LoRe Inference Details

This supplementary material provides implementation and evaluation details that support the main paper. We begin with the inference-time LoRe processor, which preserves the standard VLM input and performs visual focusing inside the same generation request. Algorithm 1 summarizes the runtime procedure from multimodal prefilling to final answer extraction.

Algorithm 1 LoRe single-turn inference algorithm. LoRe converts a standard VLM request into region-guided decoding without adding a cropped image, extra visual tokens, or a second VLM call.

Require: Image I, question Q, VLM π_θ, boost γ, interval K, window W
Ensure: Response with final answer in `<ans>...</ans>`

Prefill (I, Q) once and cache visual/text tokens as $C_\ell^{(0)}$.
Record visual-token positions in the multimodal prefix.
Decode autoregressively from $C_\ell^{(0)}$.
Every K tokens, search the latest W tokens for `<box>...</box>`.
if no valid region span is available **then**
    Continue standard VLM decoding without changing the cache.
else
    Parse $R_q = (x_1, y_1, x_2, y_2)$ and project it to $\mathcal{P}(R_q)$.
    If $\mathcal{P}(R_q)$ is non-empty, scale its value-cache entries once by γ.
    Keep keys, token order, and attention masks unchanged.
    Resume the same generation request from the modified cache.
end if
Extract the final answer using the base-model evaluator.

Prompt format. The evaluation pipeline uses a single user message that contains the image followed by a text prompt. The text prompt starts with the benchmark question and then appends the same LoRe-specific output constraints across V*, HR-Bench, and COCO-ObjQA. In the notation used throughout the paper, the prompt has the following structure:

Prompt template
{question}
Instructions:
1. Locate the most relevant object or image region that helps answer this question.
2. Output its bounding box as `<box>`x1, y1, x2, y2`</box>`.
3. Explain why this region is relevant.
4. Give the final answer as `<ans>`X`</ans>`, where X is the predicted option.

For Qwen2.5-VL, the box coordinates are interpreted as pixel coordinates with (x_1, y_1) and (x_2, y_2) denoting the top-left and bottom-right corners. For Qwen3-VL, the same prompt uses the model's relative coordinate convention in $[0, 1000]$, where zero and 1000 correspond to the image boundary along each axis. This difference affects only the coordinate parser before token-grid projection. The single-turn message structure, region span, and answer span are otherwise unchanged.

Region parsing and fallback. LoRe uses the same image-question prompt as the base VLM, with an additional instruction to emit a region span before the final answer in `<ans>...</ans>` format. During generation, the processor scans the generated token stream for the first complete `<box>`x1, y1, x2, y2`</box>` pattern. To keep this step inexpensive, it checks the stream every four generated tokens and searches only the most recent 128 generated tokens. If no valid span is found, if the span is malformed, or if the parsed box covers no visual token, LoRe leaves the cache unchanged and continues with standard VLM decoding. This fallback path is used consistently during both training rollouts and evaluation.

Coordinate and token mapping. All cache updates are applied to visual-token positions, so the parsed region must first be aligned with the visual-token grid. The processor converts the parsed box to image pixel coordinates when the prompt uses a relative coordinate convention, clamps the box to the image boundary, and maps the covered area to the grid described in Section 3.1. For Qwen-style VLMs, LoRe reads the image grid metadata returned by the model processor. If the vision backbone applies spatial token merging, the logical grid is adjusted to match the number of visual tokens stored in the language-model prefix. The selected grid cells are then mapped to sequence positions in the prompt. When explicit visual-token positions are available from the prompt token IDs, LoRe uses

them directly. Otherwise, it falls back to the contiguous image-token span inferred from the expected visual-token count.

Cache modulation. The default implementation uses `cache_values` mode. After the target visual-token sequence positions are identified, LoRe scales the corresponding value-cache entries by γ and leaves the key cache unchanged. We use $\gamma = 2.0$ by default and apply the update from the first decoder layer through the final decoder layer. The cache entries are mutated once, after the region is detected, so subsequent decoding can continue with the model's native attention kernels. We also implemented two diagnostic variants: an eager additive attention-bias mode and a `cache_kv` mode that scales both keys and values. The main paper reports value-cache scaling because it provides a favorable accuracy-efficiency tradeoff while avoiding changes to attention-score computation.

B Training-free Parser Diagnostics

Table 4 reports a small diagnostic run for the training-free LoRe pipeline before RL. The purpose is to verify that the base VLM can usually expose a parsable region span and that the fallback path is rare under the same prompt format used by LoRe. This diagnostic is not used to compute the main benchmark accuracy. **Samples** gives the number of evaluated examples. **Parsed box** is the fraction of generations that contain a syntactically valid `<box>`x1, y1, x2, y2`</box>` span. **Cache update** is the fraction of examples for which the parsed box can be mapped to at least one visual token and therefore triggers value-cache modulation. **Fallback** is the complementary fraction in which LoRe leaves the cache unchanged because no valid update region is available. **Avg. selected** reports the average number of visual tokens selected by the parsed region among updated examples, while **Avg. visual** reports the average total number of visual tokens in the original multimodal prefix.

Table 4: **Parser and cache-update diagnostics before RL.** We run Qwen2.5-VL-7B on a 200-sample COCO-ObjQA diagnostic subset with max generation length 512, and $\gamma = 2.0$.

Model	Samples	Parsed box	Cache update	Fallback	Avg. selected	Avg. visual
Qwen2.5-VL-7B	200	97.5%	97.5%	2.5%	64.5	331.7

C Reinforcement Learning Details

Reward components. For each rollout, the reward system first computes the answer reward r_{acc} using the task's answer type. Multiple-choice tasks are evaluated by extracting the final option from `<ans>...</ans>`, while numerical, OCR, Boolean, and string-style tasks use their corresponding matching rules. The auxiliary format reward r_{fmt} is binary: it is one only when the completion contains exactly one non-empty `<ans>...</ans>` span and any optional reasoning delimiters are balanced. This term supports answer extraction, while box presence and parseability are rewarded by r_{loc}.

The localization reward r_{loc} is used as weak shaping rather than as the dominant objective. For visually grounded problem types, including grounding, spatial reasoning, and embodied tasks, it adds a small bonus when a parsable box is present. For bbox-supervised samples, it additionally rewards parseability and center proximity to the target box. Given a predicted box b and a target box $b^\star$, the center component is

$$r_{\text{center}}(b, b^\star) = \max\left(0,\ 1 - \frac{\|c(b) - c(b^\star)\|_2}{\sqrt{2}\, s}\right),$$

where $c(\cdot)$ is the box center and s is the largest absolute coordinate value, lower bounded by one. The localization term used in the experiments is

$$r_{\text{loc}} = 0.10\, r_{\text{box}} + 0.05\, r_{\text{boxfmt}} + 0.15\, r_{\text{center}},$$

where unavailable terms are set to zero. Here r_{box} is the box-presence reward for visually grounded samples, r_{boxfmt} is the parseability reward for bbox-supervised samples, and r_{center} is the center-proximity reward above. The final reward follows the residual form in the main paper:

$$R = \min\{1,\ r_{\text{acc}} + (1 - r_{\text{acc}})(0.03\, r_{\text{fmt}} + r_{\text{loc}})\}.$$

This form keeps answer correctness as the primary learning signal: auxiliary terms can improve partially correct trajectories, but they cannot push a correct answer above one or dominate r_{acc}.

GRPO hyperparameters. For each prompt, we sample $G = 16$ trajectories with temperature 1.0 and maximum response length 4096. Rewards are transformed by $S_i = \mathrm{clip}(5(R_i - 0.5), -c, c)$ before group normalization, matching the scaling and bias used in our training configuration. We use asymmetric PPO clipping with lower and upper bounds 0.2 and 0.28, respectively, and a KL coefficient of 0.01 against the frozen reference policy. The actor uses Adam with learning rate 1×10^{-6}, weight decay 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, gradient clipping at 1.0, and bfloat16 precision. Training uses 8 H200 GPUs: four GPUs serve SGLang rollouts, and four GPUs compute actor and reference-policy log probabilities with FSDP. Groups whose rewards are identical are filtered because they provide no relative advantage signal.

D COCO-ObjQA Construction Details

COCO-ObjQA is derived from MS COCO instance annotations. For each image, we enumerate annotated object instances and keep target categories that appear exactly once in the image. This filtering reduces ambiguity: when a generated question refers to the target category, there is a single annotated instance that can serve as the intended visual evidence. For each retained target, we crop the annotated object from the image and provide both the full image and the crop to Gemini 3, together with the category name and the ground-truth box. The prompt asks Gemini 3 to generate a concise multiple-choice question about visible attributes of the focused object, without referring to the crop, bounding box, or object location.

We first check that the generated text follows the expected six-line structure: one question line, four answer-option lines, and one answer line. An independent GPT-5.4 check then assesses whether the question is answerable from the original full image and whether the labeled option is correct and unambiguous. Flagged problems are removed, and the remaining samples are ranked by estimated answerability or labeling risk for targeted manual inspection. Confirmed failures are removed before the final benchmark is formed. This inspection targets high-risk samples rather than a random subset and does not establish an unbiased benchmark-wide error estimate. Generated questions may still inherit model biases.

Each benchmark record stores the full-image path, question, options, answer label, COCO box, category, and target-area ratio. Standard evaluation supplies only the original full image and multiple-choice question. Crops and boxes support construction and localization diagnostics. Only the explicitly labeled oracle-box ablation uses reference boxes to guide decoding.

The targeted manual audit covers the 319 highest-risk samples after automatic filtering. Before removal of confirmed failures, 94.7% of these questions are answerable and 90.6% have the correct ground-truth option. These rates describe the selected high-risk subset, not a random estimate of label quality across COCO-ObjQA. The final benchmark statistics below are computed after automatic and manual filtering.

The size split is computed from the target-area ratio, defined as the COCO box area divided by the full image area. Following the main paper, small, medium, and large objects correspond to target-area ratios of $\leq$1%, 1–10%, and $>$10%, respectively. COCO-ObjQA contains 8,537 QA pairs from 4,107 images across all 80 COCO categories, with 1,743 small-object examples, 3,036 medium-object examples, and 3,758 large-object examples. This stratification is used only for evaluation, and the model receives the same full-image input format across all size groups.

E Evaluation and Latency Protocol

Answer extraction. The three primary benchmarks use exact-match accuracy on the final option. The evaluator first extracts the option from the model's `<ans>...</ans>` span. If a strict answer span is unavailable, it falls back to conservative option matching near the end of the response. The same extraction logic is applied to base VLMs and LoRe variants.

LoRe evaluation settings. Unless otherwise stated, LoRe evaluation uses value-cache scaling with $\gamma = 2.0$, all decoder layers enabled, a bbox check interval of four generated tokens, and a decode window of 128 generated tokens for region detection. Evaluation generation uses the same

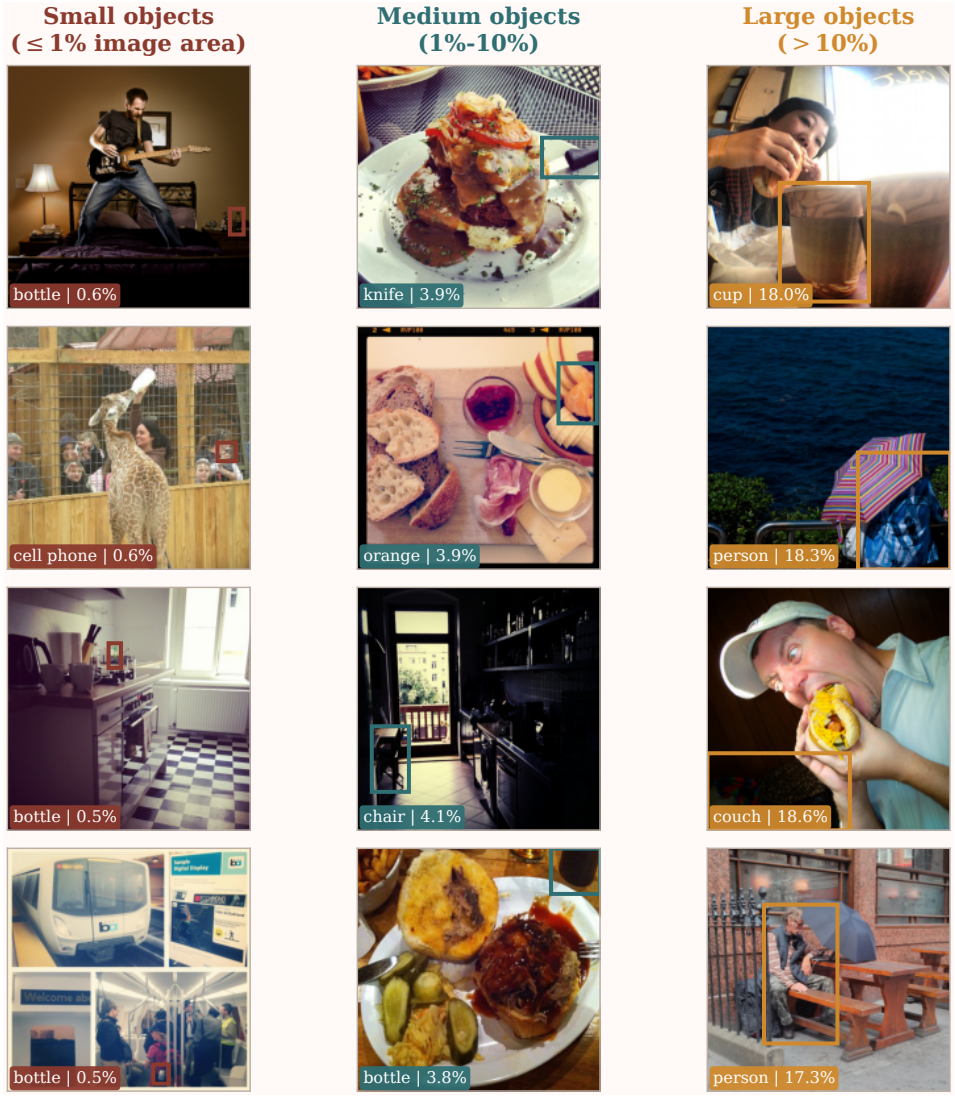


Figure 7: **COCO-ObjQA examples across target-object scales.** We visualize representative full-image samples from the small, medium, and large splits. Boxes mark the target COCO instances used to generate object-centric questions, and labels show the category and target-area ratio. The boxes are used for dataset construction and localization diagnostics, while models are evaluated on the original full image.

image resolution and prompt formatting as the corresponding base model, except for the additional instruction to emit the region span. The maximum generation length for benchmark evaluation is 2048 tokens.

Latency measurement. Latency is reported as average per-sample wall-clock generation time. All methods in Table 1 are run on the same H200 GPU machine with batch size one, and without an external inference engine such as SGLang or vLLM. For LoRe and the corresponding direct VLM baselines, timing starts immediately before model generation and stops after output tokens are produced. Dataset loading, metric aggregation, and JSON serialization are excluded. The LoRe timing includes box parsing and cache modulation, so the reported latency reflects the additional inference work introduced by the method. For explicit *thinking with images* methods, latency includes the full multi-step reasoning path, including additional VLM interactions and visual operations, when those operations are part of the method's inference procedure.

Localization diagnostics. For standard COCO-ObjQA evaluation, the ground-truth box is used only after generation to measure localization quality. COCO-ObjQA$_{\text{IoU}}$ in Table 2 averages IoU

over all evaluated examples, assigning zero when no valid box is produced. It is not used to choose the final answer. The separate oracle-box ablation substitutes COCO boxes on COCO-ObjQA and SAM2-derived [32] pseudo-ground-truth regions on V* and HR-Bench while retaining the same RL checkpoint, $\gamma = 2$, and decoding procedure. These reference regions provide a localization diagnostic rather than a strict performance bound.

F Additional Experiments and Diagnostics

F.1 Controlled RL and crop-based comparisons

Table 3 distinguishes training without modulation from disabling modulation in an already trained LoRe policy. For $\gamma_{\text{train}} = \gamma_{\text{test}} = 1$, GRPO still learns to emit a box and reasoning trace, but neither training nor evaluation modifies the cache. For the two $\gamma_{\text{train}} = 2$ rows, the checkpoint, prompt, box-generation process, and decoding procedure are identical. Only the test-time value scaling changes. Disabling scaling therefore evaluates the trained policy without the cache intervention it encountered during learning. This is distinct from training a policy without modulation from the outset.

The crop-based GRPO control uses the same Qwen2.5-VL-7B initialization, training data, recipe, and objectives as LoRe. It emits one box, receives the selected crop as a re-encoded second visual observation, and then answers. The comparison changes how the selected region is returned to the model. Crop re-encoding performs better on COCO-ObjQA, while LoRe performs better on V* and HR-Bench without adding a visual observation. These results support an accuracy–computation tradeoff rather than uniform dominance over explicit visual processing.

F.2 Cache-modulation variants

We compare three interventions on the same Qwen2.5-VL-7B base model before RL (Table 5). Additive score bias adds $\log \gamma$ to the selected visual-token logits, multiplying their attention odds by γ. The implemented eager attention path explicitly computes scores and softmax, preventing the use of Flash Attention on modified layers. Value-only scaling and joint key–value scaling instead mutate the selected cache entries once and retain the native optimized kernels. Additive bias and value-only scaling use $\gamma = 2$. Joint key–value scaling uses $\gamma = 1.01$ because stronger scaling was unstable, so this is a comparison of the evaluated configurations rather than a matched-strength sweep.

Table 5: **Training-free modulation variants.** Accuracy is reported in percent and latency in seconds. The final columns give direct changes in selected-token attention mass (percentage points) and attention entropy (nats).

Intervention	γ	V*	HR-Bench	COCO-ObjQA	Latency	Δ mass	Δ entropy
Additive score bias	2	76.8	68.2	89.3	2.71	+0.24	+0.007
Value scaling	2	**77.4**	**70.2**	**89.4**	**2.16**	0.00	0.000
Key–value scaling	1.01	74.3	68.5	89.3	2.33	+0.26	+0.005

Value-only scaling gives the highest accuracy and lowest latency among these configurations. Its zero direct change in attention mass and entropy follows from leaving queries and keys unchanged at the intervention: the attention output changes through the values, not the current attention probabilities. Subsequent attention can still change indirectly through the modified residual state. In contrast, scaling a key multiplies its logit by γ, increasing positive logits while making negative ones more negative, which explains why it is not equivalent to a uniformly positive attention bias.

F.3 Fallback behavior before and after RL

Table 6 complements the 200-sample, training-free parser diagnostic in Appendix B with benchmark-level results before and after RL. A fallback occurs when the generated box is missing, malformed, or cannot be mapped to visual tokens. The cache is left unchanged and the same policy continues decoding. These samples remain in the overall accuracy. A valid but inaccurate box is not a fallback and may still amplify irrelevant evidence.

Table 6: **Fallback frequency and answer accuracy for Qwen2.5-VL-7B.** All entries are percentages. Fallback and non-fallback accuracy are measured within their respective subsets.

Policy	Benchmark	Fallback rate	Fallback acc.	Non-fallback acc.	Overall acc.
Pre-RL	V*	11.0	61.9	79.3	77.4
Pre-RL	HR-Bench	8.5	64.7	70.7	70.2
Pre-RL	COCO-ObjQA	2.5	79.8	89.6	89.4
RL-trained	V*	6.8	69.2	83.6	82.6
RL-trained	HR-Bench	5.0	70.0	73.8	73.6
RL-trained	COCO-ObjQA	1.5	66.4	92.0	91.6

RL reduces the fallback rate on all three benchmarks, although fallback cases retain lower accuracy than non-fallback cases. In these measurements, every syntactically valid box was successfully mapped, including on HR-Bench's 4K and 8K splits. The mapping reads each image's dynamic grid metadata and spatial merge factor, so varying resolution and aspect ratio do not require a fixed patch grid. Patch quantization can still limit localization precision for very small regions.

F.4 Oracle localization and remaining errors

We use the same RL-trained Qwen2.5-VL-7B checkpoint and decoding procedure for predicted-box and oracle-box evaluation. Reference regions are COCO annotations on COCO-ObjQA and SAM2-derived pseudo-ground-truth boxes on V* and HR-Bench, as described in Section 4.3. Table 7 varies the inference-time boost factor while holding the reference regions and checkpoint fixed. The best evaluated factor remains $\gamma = 2$, indicating that stronger amplification alone does not remove the localization gap.

Table 7: **Boost strength with reference regions.** All rows use the same RL checkpoint trained at $\gamma = 2$. Scores are average accuracy (%).

Region source	γ_{test}	V*	HR-Bench	COCO-ObjQA
Predicted box	2	82.6	73.6	91.6
Oracle box	1.5	83.5	75.0	92.5
Oracle box	2	**84.1**	**75.8**	**93.0**
Oracle box	2.5	83.8	75.4	92.7
Oracle box	3	83.2	74.7	92.2

Table 8 compares correctness per example at $\gamma = 2$. "Predicted only" denotes examples answered correctly only with the predicted box, while "Oracle only" denotes errors corrected by the reference box. Recovery is the fraction of predicted-box errors corrected by the oracle, computed as $\text{Oracle only}/(\text{Oracle only} + \text{Both wrong})$.

Table 8: **Per-example oracle comparison.** The four outcome columns partition each benchmark and sum to 100%. Recovery is conditioned on predicted-box errors.

Benchmark	Both correct	Predicted only	Oracle only	Both wrong	Recovery
V*	80.5	2.1	3.6	13.8	20.7
HR-Bench	71.5	2.1	4.3	22.1	16.3
COCO-ObjQA	89.8	1.8	3.2	5.2	38.1

Reference boxes correct more errors than they introduce, but some predictions regress and substantial errors remain with either region source. The comparison therefore identifies scope for improved localization without assigning all residual errors to it. Recovering missing visual detail or improving subsequent evidence integration may be necessary for further gains. For relative-position questions, a single box may omit one of the compared objects, select a visually similar distractor, or include irrelevant context between distant objects. The multi-region results in Figure 6 show that independently adding boxes does not reliably solve this evidence-integration problem.

F.5 Transfer beyond object-attribute questions

We evaluate the training-free LoRe pipeline on MathVista-testmini [26], We-Math [31], and TIR-Bench [23]. Both rows in Table 9 use the same Qwen2.5-VL-7B base checkpoint. LoRe applies its default value-cache update directly to the original full image, with no task-specific training, cropping, image re-encoding, or additional VLM call.

Table 9: **Training-free transfer to additional reasoning tasks.** Scores are accuracy (%).

Method	MathVista-testmini	We-Math	TIR-Bench
Qwen2.5-VL-7B	68.6	61.3	16.0
LoRe (pre-RL)	71.3	64.7	16.2

LoRe improves MathVista and We-Math by 2.7 and 3.4 percentage points, respectively, supporting the use of evidence focusing for diagrams, labels, and quantities. The gain on TIR-Bench is only 0.2 points. This limited improvement is consistent with the method's scope: amplifying a fixed visual representation cannot create the transformed observations required by operations such as rotation or flipping. These results support transfer across evidence-selection tasks, rather than generality across arbitrary image manipulations.

G Limitations

LoRe assumes that the model can emit a useful region before completing the final answer. Missing, malformed, or unmappable regions leave the cache unchanged, and decoding continues with the same policy, whereas a valid but inaccurate box can amplify unhelpful evidence. The oracle-box comparison in Figure 6 indicates scope for improved localization but does not explain every remaining error.

LoRe operates on visual tokens already present in the original prefix. It cannot recover discarded detail or implement transformations such as rotation and flipping that require a new observation. A single enclosing box may cover multiple objects, but cannot weight them independently and may include distractors between distant targets. Multiple independently predicted regions do not consistently improve the reported results. Selective crop or zoom use could complement implicit focusing, but would require a routing mechanism and additional image encoding outside the current single-turn design.

COCO-ObjQA diagnoses single-target evidence selection across object sizes, but its generated questions and targeted validation do not guarantee exhaustive label correctness or coverage of relational reasoning. The broader evaluation is limited to single-image question answering. Multi-image, video, and interactive settings would require additional handling of temporal and cross-image token positions.

H Broader Impacts

LoRe aims to improve fine-grained multimodal reasoning while reducing the extra inference cost introduced by explicit visual-tool use. This can make visual reasoning systems more practical and more compute-efficient, especially in settings where a model needs to answer questions about localized evidence without repeatedly cropping or re-encoding images. At the same time, stronger and cheaper multimodal reasoning can also inherit the broader risks of VLMs, including incorrect visual interpretations, over-reliance on model outputs, and potential misuse in automated decision-making. LoRe does not by itself provide safety guarantees, so deployment in sensitive domains should include task-specific validation, human oversight, and safeguards appropriate to the application.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's main contributions: LoRe as a single-turn implicit visual thinking framework, RL training for region-aware reasoning, and COCO-ObjQA as an object-centric benchmark. These claims are supported by the method in Section 3.1–3.2, the dataset construction in Section 3.3, and the empirical results in Section 4.2–4.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix G discusses the main limitations of LoRe, including its dependence on useful region generation, failure cases where malformed or irrelevant boxes lead to fallback or unhelpful amplification, the fact that LoRe can only modulate visual tokens already encoded by the VLM, and the current focus on single-image question answering.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not state formal theorems or lemmas. The equations in Section 3.1–3.2 define the KV-cache modulation procedure and the GRPO reward/objective rather than theoretical results requiring proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 reports the benchmarks, training data, backbone models, rollout setup, optimizer, hardware, and main generation settings. Appendix A–E further specifies the inference-time LoRe processor, parser fallback behavior, reward components, GRPO hyperparameters, COCO-ObjQA construction, answer extraction, latency protocol, and localization diagnostics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and generated benchmark assets are not released with the submission to preserve anonymity. We plan to release the LoRe implementation, the COCO-ObjQA construction scripts, and the benchmark records upon acceptance. The training datasets and V*/HR-Bench evaluation benchmarks are publicly available, and COCO-ObjQA can be reconstructed from the public MS COCO dataset using the protocol described in Section 3.3 and Appendix D.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the benchmarks, training datasets, backbone models, training duration, hardware, batch size, optimizer, learning rate, KL coefficient, maximum response length, and value boosting factor. Appendix C and Appendix E provide the GRPO rollout details, reward scaling, clipping parameters, evaluation settings, answer extraction, latency measurement, and localization diagnostics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper reports exact-match accuracy and latency over held-out benchmarks, together with ablations in Table 2 and Figure 6, but it does not report confidence intervals or multiple independent training seeds. Multiple full RL training runs are omitted due to the cost of retraining 7B/8B VLM policies on 8 H200 GPUs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Section 4.1 reports that LoRe training runs for roughly 7K steps on 8 NVIDIA H200 GPUs and takes about 65 hours, with four GPUs serving SGLang rollouts and four GPUs computing actor/reference log probabilities with FSDP. Appendix C repeats the 8-H200 training setup, and Appendix E specifies that latency is measured on the same H200 GPU machine with batch size one and without an external inference engine.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the work conforms to it. The paper studies efficient multimodal reasoning with public VLM backbones, public benchmarks/datasets, and a COCO-derived object-centric benchmark; no human-subjects data, sensitive personal data, or private scraped data is collected.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix H discusses positive impacts from more compute-efficient fine-grained visual reasoning, as well as negative impacts inherited from stronger VLMs, including incorrect visual interpretations, over-reliance on model outputs, and possible misuse in sensitive applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The submission does not release a high-risk pretrained model, image generator, or scraped dataset. The VLM backbones and external benchmarks used in the experiments are existing public assets released by their original authors.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, benchmarks, pretrained models, and software systems used in the paper are credited in text with formal citations, including MS COCO, V*, HR-Bench, Ref-L4, ViGoRL-SA/SAT-2, PixMo-Count, InfographicVQA, AI2D, Qwen2.5-VL, Qwen3-VL, and SGLang. We use each asset within its publicly released terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces COCO-ObjQA and documents its construction in Section 3.3 and Appendix D, including the COCO source annotations, single-category filtering, Gemini 3 question generation protocol, automatic validation and targeted manual inspection, stored fields, target-area-ratio split, and dataset statistics. The generated benchmark records are not released with the anonymous submission, but the asset itself is described in sufficient detail for review and planned release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or human-subjects research. COCO-ObjQA is generated from MS COCO annotations with Gemini 3 question generation, automated validation, and internal manual label checks as described in Section 3.3 and Appendix D.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human-subjects research, crowdsourced participants, or private user data, so IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The core method uses Qwen2.5-VL-7B and Qwen3-VL-8B as VLM backbones for LoRe training and evaluation, as described in Section 4.1. The COCO-ObjQA construction also uses Gemini 3 to generate multiple-choice questions from full images, target crops, category names, and boxes, as documented in Section 3.3 and Appendix D.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.