ENHANCING MULTI-MODAL REASONING VIA SELF-REVISITING IMAGE EVIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust visual reasoning in Vision–language models (VLMs) often requires re-grounding intermediate steps in the underlying visual evidence. Recent approaches typically rely on external image operations such as zooming or cropping to re-access fine-grained details during inference, which requires additional image re-encoding and can disrupt the reasoning trajectory. We argue that VLMs already provide strong internal signals for identifying and reusing visual evidence, and that these signals can be directly leveraged to support image-grounded reasoning. Motivated by this insight, we propose an end-to-end self-revisit framework, **SIEVE**, that teaches models to re-engage image evidence through internal representations. SIEVE extracts embeddings of salient image regions and injects them into the reasoning chain when additional grounding is needed, enabling later steps to condition on relevant visual cues without external tool calls or re-encoding. We use reinforcement learning to teach the model when to trigger visual revisiting and which region embeddings to retrieve and insert during the reasoning process. Experiments across multiple visual reasoning benchmarks and evaluations of perception, reasoning, and hallucination show that SIEVE consistently improves performance, achieving an average gain of 8% across settings.

1 INTRODUCTION

Vision–Language Models (VLMs) have become increasingly capable of multimodal question answering, often relying on long chain-of-thought (CoT) reasoning (Team et al., 2025a; Guo et al., 2025; Bai et al., 2025a; Team et al., 2025b). However, their reasoning pipeline remains largely text-centric. In a standard VLM inference pipeline, the image is encoded into a fixed set of visual tokens that serve as static context, while reasoning unfolds autoregressively in text. As generation proceeds, the model's conditioning gradually shifts toward the growing history of generated text tokens, reducing the relative influence of visual evidence (Li et al., 2025e; Fang et al., 2025). Consequently, the model rarely revisits the image in a targeted, step-dependent way, and visual information is often underutilized in long-horizon reasoning. To address this limitation, recent work has begun to explicitly integrate visual evidence into the reasoning trajectory, drawing inspiration from human cognition (Shao et al., 2024a). Following the release of OpenAI's o3 model (OpenAI, 2025), a common approach is to use external visual operations such as zooming and cropping to obtain more focused views of relevant regions (Huang et al., 2025a; Bai et al., 2025b). Beyond predefined operations, other approaches allow VLMs to generate executable code for more flexible, programmatic image manipulation (Lee et al., 2025; Mallis et al., 2025).

These methods can be effective, but they rely on external tools or agents in ways that introduce two practical limitations. (1) **Disrupted Reasoning Continuity**: Existing methods often disrupt the continuity of the reasoning chain during multi-turn interactions. Specifically, they typically rely on an external module to generate a new visual view, which is appended to the original input rather than being directly integrated into the CoT reasoning process. As a result, the extracted visual information cannot be inserted at the corresponding reasoning step in the CoT, but instead appears as additional image input alongside the preceding images and text. (2) **Complex Training Pipeline**: Enabling a VLM to repeatedly invoke external tools requires constructing large-scale tool-use training data and designing sophisticated training pipelines, substantially increasing the complexity of equipping the model with such capabilities. Some work even argue whether tool call are actually useful. (Shao et al., 2026)

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

In this paper, we ask a simple question: *Do we truly need to generate new image views through external operations to let model revist the image information during inference?* Our hypothesis is that the original visual embeddings contain useful localized information that can be reused during reasoning, and that the main bottleneck is the model's limited ability to selectively reuse relevant visual evidence at the right time. Instead of cropping, zooming, and re-encoding additional images, we propose to directly extract task-relevant visual embeddings and insert them into the reasoning chain. To validate this idea, we manually identify salient regions and directly injecting their visual embeddings into intermediate reasoning. This simple intervention consistently improves performance on the V* benchmark without requiring any additional training.

Inspired by this observation, we introduce **SIEVE**, a framework that enables VLMs to revisit visual evidence without external tools or agentic image operations. When the model signals a need for additional visual grounding during inference, SIEVE retrieves embeddings for the relevant image regions and inserts them into the current reasoning chain, rather than localizing regions with an external tool and re-encoding new views. By reusing already encoded region features, SIEVE preserves access to fine-grained, localized visual cues for grounded multi-step reasoning while avoiding redundant vision re-encoding. We further develop a visually grounded RL training pipeline that enables the model to learn when and how to effectively retrieve and insert region embeddings. This RL training is highly data-efficient: SIEVE requires only a small dataset (approximately 1.5k samples) to acquire the capability. We evaluate SIEVE on Qwen3-based VLMs (4B and 8B) (Yang et al., 2025a) across multiple benchmarks. SIEVE consistently improves over inference-time tool-augmented baselines, showing that relevant visual evidence can be revisited by reusing internal region embeddings rather than generating new image views. Our main contributions are as follows:

- We show that the original visual embeddings already retain task-relevant fine-grained information, making it possible to revisit visual evidence without generating new image views through external crop or zoom tools. Building on this observation, we propose **SIEVE**, which retrieves and reinserts region-level visual embeddings from the original encoding during reasoning.
- We develop a **visually grounded RL training pipeline** that teaches the model when to revisit fine-grained visual evidence during reasoning. To provide fine-grained supervision for RL, we introduce a saliency-based mechanism that identifies visual embeddings critical to the current reasoning semantics. Notably, SIEVE requires only 1.5k training examples, enabling effective visual grounding with lightweight training data.
- We extensively evaluate SIEVE across multiple model scales and multimodal benchmarks. SIEVE consistently improves grounded multimodal reasoning performance, achieving up to an 8% average improvement across several benchmarks.

2 Related Work

2.1 Tool-augmented Multi-modal Reasoning

Recent work extends VLM inference beyond a single visual pass by introducing explicit visual requery mechanisms during reasoning. A prominent line equips models with visual tools (e.g., cropping, zooming, object detection) that produce targeted observations and feed them back as additional inputs (Huang et al., 2025b; Su et al., 2025b; Fan et al., 2025; Chen et al., 2025; Hu et al.). Later approaches learn tool-use policies with reinforcement learning, rewarding effective re-query trajectories and verification behaviors, often adopting coarse-to-fine strategies that start from a global view and refine under higher-resolution observations (Zheng et al., 2025; Zhong et al., 2025). Related methods further encourage pixel-space exploration, either via instruction tuning to broaden search coverage (Su et al., 2025a) or RL to strengthen perceptual competence and steer attention to task-relevant regions (Yu et al., 2025a). Despite strong performance, these systems typically incur inference-time overhead due to repeated view generation and re-encoding.

2.2 Multi-modal Reasoning in Latent Space

In parallel, several lines reduce reliance on explicit image operations by operating in representation space or by selectively using visual tokens. Generation-based methods synthesize auxiliary visual representations to support inference (Xu et al., 2025; Chern et al., 2025; Li et al., 2025c). More recently, latent thinking-with-images paradigms introduce learnable latent visual tokens and

embedding-level manipulation to internalize certain visual operations and enable mode switching during inference (Li et al., 2025a; Zhang et al., 2025a; Yang et al., 2025b). A complementary efficiency literature observes that dense patch sequences are redundant and studies how to select, prune, or merge visual tokens while preserving representational fidelity (Chen et al., 2024; Cao et al., 2024; Wang et al., 2025; Bolya et al., 2022; Zeng et al., 2025; Zhang et al., 2024a; Huang et al., 2024; Li et al., 2025b; Hu et al., 2025; Yu et al., 2025b; Song et al., 2024). These approaches share a common premise: they construct reasoning processes within a latent visual space and train models to perform reasoning in that space. However, this paradigm requires substantial effort to enable the model to learn reasoning over newly introduced visual latents that differ from the native textual representation space. Motivated by this limitation, we propose a framework that directly leverages the embeddings of images within the textual reasoning space, rather than constructing and training reasoning in a separate latent space.

3 METHODOLOGY

Our motivation is that the visual embeddings produced by a VLM already contain rich information for visual reasoning, as long as the model can access the relevant local evidence when needed. We first test this idea with a controlled study on the V* dataset examining whether region-level visual features can directly enhance multimodal inference. Concretely, we manually identify task-relevant regions, extract their corresponding embeddings, and augment inference by inserting these region embeddings with the model's original visual embeddings, all without any additional training. We evaluate this intervention against the standard setting that relies solely on global image. As shown in Figure 1, region-augmented inference yields a 3% improvement, confirming that localized embedding evidence provides a compact yet effective signal that the model can immediately leverage. This finding aligns with prior work demonstrating that LLM hidden states encode rich semantic structure (Skean et al., 2025; Schiekiera et al., 2026; Liu et al., 2024).

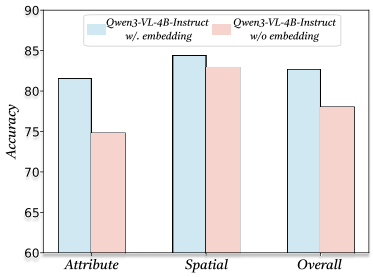


Figure 1: Performance with and without region embeddings.

Motivated by this observation, we introduce **SIEVE**, a framework that treats task-relevant region embeddings as reusable visual evidence and learns to incorporate them within RL policy optimization. Specifically, SIEVE (i) extracts and caches region embeddings as compact evidence units and (ii) jointly optimizes how such evidence is selected and integrated into the model's reasoning process throughout reinforcement learning. Section 3.1 presents an overview of the training pipeline. Section 3.2 details our automatic evidence discovery procedure. Section 3.3 introduces a visually grounded RL formulation that trains the model to retrieve and insert cached region embeddings on demand, enabling systematic evidence-aware reasoning beyond global visual representations.

3.1 OVERVIEW OF SIEVE

Figure 5 illustrates the training workflow of SIEVE. Prior to training-time rollouts, we construct embeddings of visually salient regions through a two-stage process. On the textual side, we compute token-level saliency scores to identify the tokens that exert the greatest influence on subsequent generation. These salient tokens serve as semantic queries. On the visual side, we compute cross-modal similarity between each query token and the image patch representations to localize the most relevant spatial regions.

During each rollout, the model first assesses whether supplementary visual information is required at the current reasoning step. When additional visual information is warranted, the previously cached evidence embeddings are injected into the generation process. Otherwise, the model proceeds directly toward producing an answer. This selection decision can be invoked repeatedly across multiple reasoning steps until the model either produces a final answer or reaches a predefined maximum reasoning horizon. Upon termination, a reward is computed from the final output. When the model fails despite utilizing the cached evidence, we interpret this as an indication that the stored embeddings are misaligned or insufficient. In such cases, we re-identify task-relevant regions under the current model parameters, re-extract their region embeddings, and update the evidence cache ac-

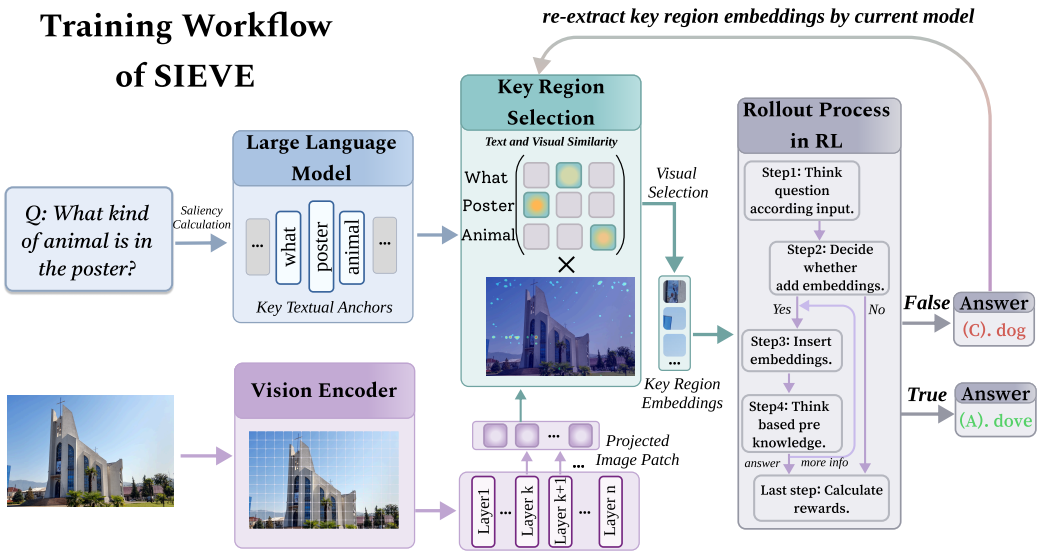


Figure 2: Training workflow of SIEVE. For each question, the embeddings of image patches aligned with key textual anchors are extracted and cached as visual evidence. During RL rollouts, the policy learns when to insert this evidence into the reasoning stream, with rewards computed from the final answer. Embeddings of visual evidence are periodically re-extracted using the updated model to keep the evidence aligned with the evolving policy.

cordingly. The refreshed embeddings are employed in subsequent rollouts, enabling the evidence to co-evolve with the policy and progressively improve visual grounding throughout training.

Algorithm 1 Self-Guided Visual Evidence Discovery

Require: Multimodal model $\mathcal{M}$; image I; token embeddings $\{\mathbf{h}_i\}_{i=1}^{L}$; u, v row and column coordinates of the patch.
Ensure: Evidence snapshot set $\mathcal{E}$

Run $\mathcal{M}(I, \{\mathbf{h}_i\})$ to obtain logits $\mathbf{z}_L$ and hidden states $\{\mathbf{H}^{(\ell)}\}$.
$\hat{v} = \arg\max_{v \in \mathcal{V}} \mathbf{z}_L[v],\ s = \mathbf{z}_L[\hat{v}]$ // choose prediction target
$\mathrm{Sal}(i) = \|\nabla_{\mathbf{h}_i} s \odot \mathbf{h}_i\|_2,\ \mathcal{A} = \mathrm{Filter}(\mathrm{Sal})$ // identify salient textual anchors
$\bar{\mathbf{H}} = \frac{1}{|\mathcal{L}_{\mathrm{mid}}|} \sum_{\ell \in \mathcal{L}_{\mathrm{mid}}} \mathbf{H}^{(\ell)}$ // stabilize cross-modal representations
Extract patch tokens $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^{N}$ and anchor token reps $\{\mathbf{q}_i\}_{i \in \mathcal{A}}$ from $\bar{\mathbf{H}}$.
Normalize $\hat{\mathbf{x}}_j = \mathbf{x}_j / \|\mathbf{x}_j\|_2, \hat{\mathbf{q}}_i = \mathbf{q}_i / \|\mathbf{q}_i\|_2$; initialize $\mathcal{E} \leftarrow \emptyset$.
for $i \in \mathcal{A}$ **do**
  $s_{ij} = \cos(\hat{\mathbf{q}}_i, \hat{\mathbf{x}}_j),\ w_{ij} = \frac{\exp(s_{ij}/\tau)}{\sum_{u=1}^{N} \exp(s_{iu}/\tau)}$ // anchor–patch affinity
  $S(\mathcal{B}_i) = \max_{p_{u,v} \in \mathcal{B}_i} s_{u,v},\ \mathcal{B}^* = \mathrm{TopK}(\{S(\mathcal{B}_i)\})$ // score blocks and select top-k
  $\mathcal{R}_i = \mathrm{BBox}\big(\bigcup_{\mathcal{B} \in \mathcal{B}^*} \mathcal{B}\big)$ // merge selected blocks into a region
  $E_i = \underset{j \in \mathcal{R}_i}{\mathrm{Concat}}\ \mathbf{x}_j$ // concatenate all patch embeddings in region $\mathcal{R}_i$
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{E_i\}$
end for
return $\mathcal{E}$

3.2 Self-Guided Visual Evidence Identification

A central challenge in constructing visual evidence lies in determining what to store: the embeddings must capture precisely the visual information that the model would need to revisit during reasoning, without relying on manual annotation or task-specific heuristics. We address this through a two-stage self-guided visual evidence identification pipeline, which is illustrated in Algorithm 1. First, the model introspects on its own predictive process to surface the most prediction-critical tokens as

textual anchors. Subsequently, we ground these anchors onto spatially coherent image regions via cross-modal matching within the model's internal representation space.

3.2.1 Discovering Textual Anchors via Gradient Saliency

Rather than relying on external concept taggers or handcrafted keyword lists, we derive anchors directly from the model's own predictive dynamics. Our primary signal is token-level gradient saliency: if a token is critical to the model's next-step prediction, the output logit will exhibit high sensitivity to perturbations of that token's embedding, manifesting as a large gradient magnitude. This yields an importance landscape over input tokens, from which we select the most influential ones as anchors. Formally, let the multimodal model produce logits $\mathbf{z}_L \in \mathbb{R}^{|\mathcal{V}|}$ at the last input position L, where $\mathcal{V}$ is the vocabulary. Let $\hat{v} = \arg\max_{v \in \mathcal{V}} \mathbf{z}_L[v]$ denote the predicted next token, and define the scalar target $s = \mathbf{z}_L[\hat{v}]$. We compute a saliency score for each input token embedding $\mathbf{h}_i \in \mathbb{R}^d$ as

$$\mathrm{Sal}(i) = \left\| \nabla_{\mathbf{h}_i} s \odot \mathbf{h}_i \right\|_2, \tag{1}$$

where $\odot$ denotes element-wise multiplication. This gradient–input formulation captures both the sensitivity of the prediction (via the gradient) and the magnitude of the representation (via $\mathbf{h}_i$), ensuring that high saliency reflects genuine dependence of the model's output on token i. Since raw saliency scores often assign non-trivial weight to function words (e.g., *the*, *is*) that carry limited semantic content, we apply a stop-word filter and retain only content-bearing tokens whose saliency exceeds a predefined threshold. The surviving tokens constitute our textual anchors, i.e., the semantics that the model implicitly treats as pivotal to its reasoning (e.g., objects, attributes, or spatial relations). These anchors subsequently serve as queries for visual grounding.

3.2.2 Identifying Visual Evidence with Textual Anchors

Given the textual anchors, we localize their corresponding visual regions by matching the internal hidden representations of text tokens and image patch tokens within the model's joint multimodal space, where both modalities reside in the same representation space and can therefore be directly compared. Our approach operates on intermediate-layer representations, where cross-modal semantics exhibit more explicit alignment.

Let $\mathbf{H}^{(\ell)} \in \mathbb{R}^{L \times d}$ denote the hidden states at layer ℓ, for $\ell = 1, \ldots, \mathcal{L}$. Prior work has shown that middle layers tend to capture richer semantic representations than either early or later layers (Skean et al., 2024; 2025). We accordingly compute a stabilized representation by averaging middle layer's hidden states with: $\bar{\mathbf{H}} = \frac{1}{|\mathcal{L}_{\text{mid}}|} \sum_{\ell \in \mathcal{L}_{\text{mid}}} \mathbf{H}^{(\ell)}$. Both modalities are obtained from the same $\bar{\mathbf{H}}$ by indexing the corresponding token positions. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote the patch-token representations and $\mathbf{q} \in \mathbb{R}^d$ the representation of a textual anchor token.

To ensure robust similarity computation, we apply mean-centering and ℓ_2 normalization to both the patch tokens and the anchor, yielding normalized vectors $\{\hat{\mathbf{x}}_i\}_{i=1}^N$ and $\hat{\mathbf{q}}$. We then compute anchor–patch affinity via cosine similarity: $s_i = \cos(\hat{\mathbf{x}}_i, \hat{\mathbf{q}}) \quad i = 1, \ldots, N$, and convert the affinities into a temperature-scaled softmax distribution:

$$w_i = \frac{\exp(s_i/\tau)}{\sum_{j=1}^N \exp(s_j/\tau)}. \tag{2}$$

We leverage this distribution to extract a spatially coherent region. Specifically, we map patch tokens onto the $H \times W$ patch grid and compute each block's score as the maximum similarity within the block: $s_b = \max_{j \in \mathcal{B}_b} w_{ij}$,where $\mathcal{B}_b$ denotes the set of patches in block b. We then select the top-K scoring blocks $\mathcal{B}_i = \mathrm{TopK}(\{s_b\})$, and expand each selected block by computing the bounding rectangle of the selected patches. We set k to 1, and further discussion it in the 8. We evaluate the similarity of these patches with their corresponding textual anchors and retain the highest-scoring one as the expanded block $\mathcal{R}_i = \mathrm{Expand}(\mathcal{B}_i)$. The embeddings of patches within each region are then aggregated to form a region-level snapshot: $E_i = \underset{j \in \mathcal{R}_i}{\mathrm{Concat}} \mathbf{x}_j$. The resulting region embeddings are cached as evidence snapshots and stored alongside each training sample, serving as reusable visual evidence during rollout. When needed, these embeddings are appended to the input embedding sequence during the prefill stage and processed as continuous context embeddings, with

their hidden states pre-computed into the KV cache so that subsequent tokens can condition on the injected visual evidence throughout generation.

3.3 Visual-grounded Reinforcement Learning

Existing tool-augmented thinking-with-images methods (Zheng et al., 2025; Hong et al., 2025; Zhang et al., 2025b) enlarge the action space with external tool calls and require image re-encoding at every reasoning step. Since SIEVE simply reuses embeddings already produced by the vision encoder and projected into text space, it sidesteps these issues entirely and we can formalize the reasoning as shown in Equation 3, where the policy selects the next action conditioned on the original image and the full interaction history, including both generated text and any previously inserted visual evidence, accumulated up to the current step:

$$a_t \sim \pi_\theta(\cdot \mid s_t), \qquad s_t \triangleq I \,\|\, (x_1 \| E_1) \,\|\, \cdots \,\|\, (x_{t-1} \| E_{t-1}). \tag{3}$$

Here, I denotes the input image, x_t is the text generated by the model, and E_t is the embeddings of visual evidence inserted at turn t (with $E_t = \varnothing$ when no insertion occurs). Thus, s_t represents the accumulated context: the raw image followed by all preceding textual responses and inserted evidence blocks in temporal order. Conditioned on s_t, the policy samples the next action a_t, either terminating with a final answer or triggering insertion of the cached visual evidence. The rollout terminates when a final answer is produced or when a predefined maximum number of turns is reached.

Trajectory-level reward design. We design a trajectory-level reward function that holistically evaluates the quality of the complete reasoning path. The reward comprises four complementary components, each yielding a binary score of 1 (satisfied) or 0 (violated). Given a trajectory τ, the total reward is:

$$\mathcal{R}(\tau) = \lambda_1 \mathcal{R}_{\text{res}}(\tau) + \lambda_2 \mathcal{R}_{\text{format}}(\tau) + \lambda_3 \mathcal{R}_{\text{emb}}(\tau) + \lambda_4 \mathcal{R}_{\text{act}}(\tau), \tag{4}$$

where the λ values are scaling coefficients. In our experiments, we set $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $\lambda_3 = 0.5$, and $\lambda_4 = 0.2$. In Section 10, we discuss the settings of the reward parameters. Each component targets a distinct aspect of the desired behavior:

- **Format reward** ($\mathcal{R}_{\text{format}}$) promotes well-structured outputs. For single-turn trajectories, the full reward is granted only if the model produces a valid reasoning chain followed by a final answer. For multi-turn trajectories, obtaining the full reward additionally requires an explicit embedding selection during an intermediate turn. Any structural violation results in a zero format reward.
- **Result reward** ($\mathcal{R}_{\text{res}}$) evaluates the correctness of the final answer, serving as the primary learning signal for reasoning quality.
- **Embedding reward** ($\mathcal{R}_{\text{emb}}$) is activated exclusively when the model produces a correct final answer and invokes embedding insertion at least once during intermediate reasoning steps. This bonus incentivizes the model to actively leverage visual evidence when it is beneficial for task resolution, rather than bypassing the available evidence.
- **Action reward** $\mathcal{R}_{\text{act}}$ improves training stability in two ways: (i) a thought richness reward, it penalizes overly short reasoning traces that could hack the reward, and (ii) a signal reward, it provides a small positive reward for committing to an action, either retrieving an embedding or producing an answer, which discourages the policy from collapsing into "non-committal" outputs that avoid taking actions.

4 Experiments

4.1 Experimental Setup

For training, we sample 1,500 images from COCO 2017 (Lin et al., 2014) and construct the corresponding training data with pre-extracted region embeddings. For evaluation, we focus on two challenging high-resolution visual reasoning benchmarks: V*Bench (Wu & Xie, 2024) and HR-Bench (Wang et al., 2024), reporting results at both 4K and 8K resolutions. To assess generalization

Table 1: Results on V* Bench and HR-Bench. * indicates results reproduced by our implementation.

Model	V* Bench			HR-Bench 4K			HR-Bench 8K		
	Attribute	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall
Qwen3-VL-4B-Instruct									
Vanilla*	74.78	82.89	78.01	86.00	69.50	77.75	80.75	64.00	72.38
DyFo	86.09	75.00	81.68	72.75	57.25	65.00	69.75	53.50	61.62
ZoomEye	**92.17**	86.84	**90.05**	**90.00**	61.00	75.50	**88.00**	60.00	74.00
Zoom-Refine	86.09	73.47	81.17	87.00	66.00	76.50	87.00	63.00	75.00
SIEVE	81.74	**92.11**	85.86	88.00	**74.50**	**81.25**	85.00	**67.25**	**76.13**
Δ (vs. Vanilla)	+6.96	+9.22	+7.85	+2.00	+5.00	+3.50	+4.25	+3.25	+3.75
Qwen3-VL-8B-Instruct									
Vanilla*	84.35	80.26	82.72	89.00	69.00	79.00	81.00	67.50	74.25
DyFo	86.09	78.95	83.25	78.50	57.25	67.88	71.50	54.50	63.00
ZoomEye	81.74	73.68	78.35	88.00	60.00	74.00	88.00	53.00	70.50
Zoom-Refine	83.48	82.89	83.25	**91.00**	63.00	77.00	**92.00**	60.00	76.00
SIEVE	**88.70**	**86.84**	**87.96**	90.00	**73.00**	**81.50**	83.00	**73.50**	**78.25**
Δ (vs. Vanilla)	+4.35	+6.58	+5.24	+1.00	+2.75	+2.50	+2.00	+6.00	+4.00

beyond high-resolution reasoning, we additionally evaluate on perception benchmarks MME-RealLite (Zhang et al., 2024b) and RealWorldQ (xAI); multimodal reasoning benchmarks MathVista (Lu et al., 2024), LogicVista (Xiao et al., 2024), and WeMath (Qiao et al., 2025); and the hallucination benchmark Hallusion (Wu et al., 2024). We compare SIEVE against representative zoom-and-refine baselines, including DyFo (Li et al., 2025d), ZoomEye (Shen et al., 2024), and Zoom-Refine (Yu et al., 2025c), as well as a vanilla GRPO-trained model optimized solely with format and accuracy rewards. We adopt Qwen3-VL-4B-Instruct (Yang et al., 2025a) and Qwen3-VL-8B-Instruct as base models. Both are trained with GRPO (Shao et al., 2024b) for 60 rollout steps on two NVIDIA H200 GPUs. Each rollout batch contains 16 prompts with 8 rollouts per prompt. We set the KL divergence coefficient to 0.0 and the maximum response length is set as 8,192 tokens.

4.2 Main Results

In Table 1, we report the performance of SIEVE on V* and HRBench, comparing it with other models and methods at both the 4B and 8B scales. As shown in the table, SIEVE consistently outperforms all baselines on both benchmarks across both model sizes. Notably, the 4B variant of SIEVE achieves a 10.06% relative improvement over the corresponding vanilla model. This result indicates that enabling the model to reason with hidden-state embeddings can effectively enhance performance on high-resolution tasks. We further validate our approach on additional datasets. As presented in Table 2, SIEVE demonstrates consistent improvements over both the vanilla model and the GRPO-trained model across perception tasks, reasoning tasks, and hallucination benchmarks. In particular, SIEVE 4B achieves a 17.58% gain on MME-Real-Lite, while SIEVE 8B achieves improvements of 19.30% on MME-Real-Lite and 20.65% on WeMath. These results suggest that injecting hidden-state embeddings during reasoning can yield benefits across diverse task categories. We analyze the time cost between SIEVE and other method in Section 7. The effects of hyper-parameters during the identification of visual evidence are reported in Section 8. Additional results compared with DeepEyes and Pixel-Reasoner in Section 9. And we provide the visual example in Section 11.1.

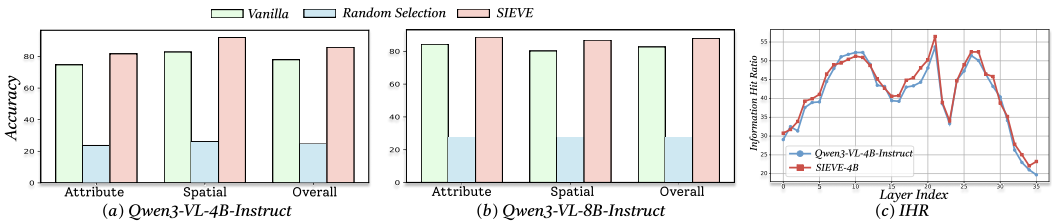
4.3 Ablation Studies

4.3.1 Role of Inserting Embedding

In this section, we empirically demonstrate that the selected region embeddings contribute positively and meaningfully to the reasoning process. To validate this claim, we construct a controlled ablation experiment in which image patch embeddings are randomly sampled and inserted following the same inference protocol as our method. Concretely, whenever the model determines that

Table 2: Benchmark comparison on perception, reasoning, and general tasks. For each model size, we report the base model (Vanilla), applying GRPO to the base model (+GRPO), and our method SIEVE. Subscripts indicate absolute gains over the corresponding Vanilla baseline. * denotes results reproduced by us.

Benchmark	Split	Qwen3-VL-4B-Instruct			Qwen3-VL-8B-Instruct		
		Vanilla*	+GRPO	SIEVE	Vanilla*	+GRPO	SIEVE
MME-Real-Lite	Perception	50.90	54.41	**55.95**$_{+5.05}$	52.95	54.58	**58.43**$_{+5.48}$
	Reasoning	38.27	40.00	**51.07**$_{+12.8}$	38.40	39.47	**53.20**$_{+14.8}$
	Overall	45.96	48.78	**54.04**$_{+8.08}$	47.26	48.67	**56.38**$_{+9.12}$
RealWorldQA	Overall	68.10	67.84	**71.11**$_{+3.01}$	69.28	**70.72**	69.28$_{+0.00}$
MathVista	Mini	67.70	68.40	**70.50**$_{+2.80}$	69.90	70.30	**71.60**$_{+1.70}$
LogicVista	Overall	43.08	46.21	**48.99**$_{+5.91}$	48.66	52.01	**53.02**$_{+4.36}$
WeMath	Overall	50.98	51.03	**53.05**$_{+2.07}$	54.71	55.40	**66.01**$_{+11.3}$
HallusionBench	qAcc	35.74	37.55	**38.99**$_{+3.25}$	36.10	37.91	**42.24**$_{+6.14}$
	fAcc	41.91	42.77	**43.64**$_{+1.73}$	44.22	45.38	**47.98**$_{+3.76}$
	aAcc	63.41	**64.98**	64.95$_{+1.54}$	66.35	67.19	**68.11**$_{+1.76}$
	Overall	47.02	48.43	**49.19**$_{+2.17}$	48.89	50.16	**52.78**$_{+3.89}$



(a) *Qwen3-VL-4B-Instruct* (b) *Qwen3-VL-8B-Instruct* (c) *IHR*

Figure 3: (a) and (b) are the comparison of w/o inserting embedding, inserting random select embedding and insert embedding chosen by our method. (c) is IHR in different Layers on Qwen3-VL-4B-Instruct and SIEVE-4B.

additional visual information is required to assist its reasoning, we inject randomly selected patch embeddings instead of the semantically aligned embeddings identified by our saliency-based selection mechanism. We evaluate this variant on V* Bench, and the results are reported in Figure 3.(a) and Figure 3.(b). As shown, replacing our selected embeddings with randomly sampled ones leads to a substantial and consistent degradation in performance. This performance drop indicates that the gains observed in our method are not simply due to the act of injecting additional visual tokens into the reasoning process. Rather, they stem from incorporating semantically relevant and contextually aligned visual embeddings that meaningfully support intermediate reasoning steps. Together, these results show that our embedding selection captures task-relevant visual information and that the gains stem from informed cross-modal grounding rather than arbitrary token augmentation.

We visualize image-token hidden states from different VLM layers and analyze the corresponding image embeddings selected via saliency-based token analysis. Specifically, after identifying salient tokens that strongly influence generation, we compute their similarity with image patch representations extracted from different transformer layers and map the selected embeddings back to their spatial locations to obtain visual grounding results.

Using *Qwen3-VL-4B-Instruct* as an example, we compute layer-wise embedding retrieval accuracy to justify selecting middle layers. Since our goal is to extract informative key-region embeddings rather than fully reconstruct object extents, we only require the predicted region to overlap with the ground-truth object region. Accordingly, we define the **Information Hit Ratio (IHR)** as



Figure 4: The comparison of metrics change during stage with and without action rewards. (a) is the rewards change. (b) is the extropy loss change, (c) is the average response length, (d) is the max response length.

IHR $= \mathbb{I}\left(|B_{\text{pred}} \cap B_{\text{gt}}| > 0\right)$, where a prediction is considered correct if the predicted bounding box overlaps with the ground-truth region.

We randomly sample 300 images from *COCO2017*, each potentially containing multiple objects. As shown in Figure 3(c), the IHR increases significantly in the middle layers, supporting our design choice of extracting features from intermediate representations. In this experiment, we set $TopK = 1$ to directly reflect the feature matching quality of each layer. We further compare the matching precision between SIEVE and the vanilla model before and after training in Section 11.2.

4.3.2 Role of Action Rewards

In Section 3.3, we incorporate action-level rewards into the total reward function, including a thought richness reward and a signal reward. These additional rewards encourage the model to produce more informative reasoning traces and issue appropriate response requests, improving both training stability and interpretability. In Figure 4, we analyze the impact of enabling or disabling these action rewards. Figures (a) and (b) illustrate the effect of the signal reward. In Figure (a), when the signal reward is removed, the training reward collapses in the later stages. Although the model performs well during early training, the overall reward eventually drops to nearly zero, indicating that without explicit signal supervision, the policy drifts and fails to maintain consistent behavior. In contrast, Figure (b) shows that introducing the signal reward significantly stabilizes training and prevents reward collapse.

Figures (c) and (d) examine the influence of the thought richness reward. With this reward enabled, the average response length remains more stable throughout training, avoiding the sudden decline observed when it is disabled. Without the thought richness reward, the model tends to exploit the reward mechanism by outputting empty reasoning tags (e.g., *<think> </think>*) without substantive thought content, obtaining rewards without genuinely engaging in reasoning. Moreover, the maximum response length becomes highly unstable, often due to repetitive or duplicated content that artificially inflates response length. These results show that action-level rewards improve training stability, prevent degenerate behaviors, and promote meaningful, structured reasoning during reinforcement learning.

5 Conclusion

In this work, we propose a training framework, SIEVE, that enables vision–language models (VLMs) to revisit and leverage image information during inference without relying on external tools. Unlike existing approaches that depend on tool invocation, retrieval systems, or additional visual processing modules, SIEVE extracts and utilizes the intrinsic signals already present within the VLM's hidden states, using them as structured hints to guide image-grounded reasoning. By exploiting these internal representations, SIEVE encourages the model to dynamically refine its understanding of visual content in a lightweight and self-contained manner. Despite its simplicity, SIEVE demonstrates substantial performance improvements over both the vanilla baseline model and training-free tool-based methods. These results suggest that effective visual revisiting does not necessarily require external tool orchestration; rather, carefully harnessing internal multimodal representations can provide a more efficient and scalable solution for improving image-grounded reasoning.

AI USE STATEMENT

In this paper, we used AI tools solely to refine grammar and improve the clarity and readability of the text.

ETHICS STATEMENT

All datasets used in this work are publicly available and contain no sensitive information. Our method enhances LLM reasoning without introducing new data collection or human interaction. While stronger reasoning ability may be misused, we emphasize that this work is intended for beneficial research and responsible applications.

REPRODUCIBILITY STATEMENT

We will release our code and data once the paper is published. The appendix includes detailed experimental setups and hyperparameters so that others can reproduce our results. We also encourage the community to follow good research practices when using our code and data, to help maintain the reliability and transparency of future work.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025a.

Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *arXiv preprint arXiv:2505.14231*, 2025b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15710–15719, 2024.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Zhangquan Chen, Ruihui Zhao, Chuwei Luo, Mingze Sun, Xinlei Yu, Yangyang Kang, and Ruqi Huang. Sifthinker: Spatially-aware image focus for visual reasoning. *arXiv preprint arXiv:2508.06259*, 2025.

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

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*, 2025.

Hao Fang, Changle Zhou, Jiawei Kong, Kuofeng Gao, Bin Chen, and Shu-Tao Xia. Grounding language with vision: A conditional mutual information calibrated decoding strategy for reducing hallucinations in lvlms. *arXiv preprint arXiv:2505.19678*, 2025.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Junshan Hu, Jialiang Mao, Zhikang Liu, Zhongpu Xia, Peng Jia, and Xianpeng Lang. Tokenflex: Unified vlm training for flexible visual tokens inference. *arXiv preprint arXiv:2504.03154*, 2025.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jiaqi Huang, Zunnan Xu, Jun Zhou, Ting Liu, Yicheng Xiao, Mingwen Ou, Bowen Ji, Xiu Li, and Kehong Yuan. Sam-r1: Leveraging sam for reward feedback in multimodal segmentation via reinforcement learning. *arXiv preprint arXiv:2505.22596*, 2025a.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European Conference on Computer Vision*, pp. 214–230. Springer, 2024.

Zeyi Huang, Yuyang Ji, Anirudh Sundara Rajan, Zefan Cai, Wen Xiao, Haohan Wang, Junjie Hu, and Yong Jae Lee. Visualtoolagent (vista): A reinforcement learning framework for visual tool selection. *arXiv preprint arXiv:2505.20289*, 2025b.

Jimin Lee, Steven-Shine Chen, and Paul Pu Liang. Interactive sketchpad: A multimodal tutoring system for collaborative, visual problem-solving. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2025.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025a.

Chaoyu Li, Yogesh Kulkarni, and Pooyan Fazli. Regate: Learning faster and better with fewer tokens in mllms. *arXiv preprint arXiv:2507.21420*, 2025b.

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. *arXiv preprint arXiv:2501.07542*, 2025c.

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, 2025d. URL https://arxiv.org/abs/2504.14920.

Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N Metaxas. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. *arXiv preprint arXiv:2502.03628*, 2025e.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14551–14558, 2024.

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 (ICLR)*, 2024.

Dimitrios Mallis, Ahmet Serda Karadeniz, Sebastian Cavada, Danila Rukhovich, Niki Foteinopoulou, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. Cad-assistant: tool-augmented vllms as generic cad task solvers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7284–7294, 2025.

OpenAI. Thinking with images. https://openai.com/index/thinking-with-images/, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, Yifan Zhang, et al. 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)*, pp. 20023–20070, 2025.

Louis Schiekiera, Max Zimmer, Christophe Roux, Sebastian Pokutta, and Fritz Günther. From associations to activations: Comparing behavioral and hidden-state semantic geometry in llms. *arXiv preprint arXiv:2602.00628*, 2026.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Jiahao Shao, Yuanbo Yang, Yiyi Liao, Yujun Shen, Ceyuan Yang, and Yinghao Xu. Thinking with tools, not with pixels: Tool calls as text scaffolds for visual reasoning. *arXiv preprint arXiv:2608.09682*, 2026.

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

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. *arXiv preprint arXiv:2411.16044*, 2024.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. Does representation matter? exploring intermediate layers in large language models. *arXiv preprint arXiv:2412.09563*, 2024.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. *arXiv preprint arXiv:2409.10994*, 2024.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025a.

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 preprint arXiv:2505.08617*, 2025b.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025a.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025b.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2408.15556.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025.

Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Xiyang Wu, Tianrui Guan, Dianqi Li, Shuaiyi Huang, Xiaoyu Liu, Xijun Wang, Ruiqi Xian, Abhinav Shrivastava, Furong Huang, Jordan Lee Boyd-Graber, Tianyi Zhou, and Dinesh Manocha. Autohallusion: Automatic generation of hallucination benchmarks for vision-language models, 2024. URL https://arxiv.org/abs/2406.10900.

xAI. Grok-1.5 vision preview. URL https://x.ai/news/grok-1.5v.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts, 2024. URL https://arxiv.org/abs/2407.04973.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025b.

En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, et al. Perception-r1: Pioneering perception policy with reinforcement learning. *arXiv preprint arXiv:2504.07954*, 2025a.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Introducing visual perception token into multimodal large language model. *arXiv preprint arXiv:2502.17425*, 2025b.

Xuan Yu, Dayan Guan, and Yanfeng Gu. Zoom-refine: Boosting high-resolution multimodal understanding via localized zoom and self-refinement. *arXiv preprint arXiv:2506.01663*, 2025c.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *arXiv preprint arXiv:2508.01548*, 2025.

Chi Zhang, Haibo Qiu, Qiming Zhang, Zhixiong Zeng, Lin Ma, and Jing Zhang. Deepsketcher: Internalizing visual manipulation for multimodal reasoning. *arXiv preprint arXiv:2509.25866*, 2025a.

Huiyong Zhang, Lichun Wang, Shuang Li, Kai Xu, and Baocai Yin. Area-keywords cross-modal alignment for referring image segmentation. *Neurocomputing*, 581:127475, 2024a.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024b.

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. *arXiv preprint arXiv:2508.11630*, 2025b.

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. *arXiv preprint arXiv:2505.14362*, 2025.

Hao Zhong, Muzhi Zhu, Zongze Du, Zheng Huang, Canyu Zhao, Mingyu Liu, Wen Wang, Hao Chen, and Chunhua Shen. Omni-r1: Reinforcement learning for omnimodal reasoning via two-system collaboration. *arXiv preprint arXiv:2505.20256*, 2025.

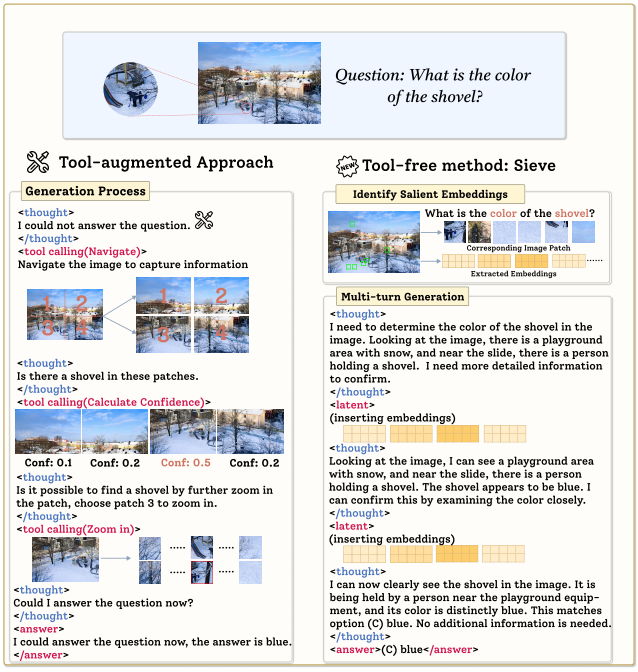


Figure 5: This figure compares tool-augmented methods with SIEVE. The left shows tool-based reasoning, where external tools are invoked for additional visual information. The right shows SIEVE, which directly retrieves and injects key region embeddings into the reasoning process.

6 ILLUSTRATION OF SIEVE

Here we use an figure to compare SIEVE with current other methods.

7 TIME COST

Table 3: Average inference time per item and accuracy on V* benchmark under different reasoning mechanisms.

Model	Vanilla		ZoomEyes		ZoomRefine		SIEVE	
	Time (s)	Acc.	Time (s)	Acc.	Time (s)	Acc.	Time (s)	Acc.
Qwen3-VL-4B-Instruct	5.4758	78.01	5.7689	90.05	6.5733	81.17	6.2811	85.86
Qwen3-VL-8B-Instruct	3.5696	82.72	6.8715	78.35	7.4327	83.25	6.4445	87.86

This section evaluates the relationship between the average response time per question and the accuracy. We run experiments on the Vstar dataset to measure the average time required to answer each question. To ensure a fair comparison across methods, we disable inference acceleration frameworks (e.g., vLLM and SGLang). Accordingly, we exclude DyFo from this analysis, since its original implementation is vLLM-based, and re-implementing it in a huggingface inference stack could introduce non-trivial performance degradation and confound the latency comparison. All experiments are conducted under the same hardware and runtime conditions. In addition, we fix the maximum generation length to 1024 tokens and enforce a unified "think-then-answer" protocol for all methods, ensuring that differences in response time are not caused by variations in output length or reasoning format. The results are presented in Table 3.

Overall, we observe that the average response time of SIEVE does not increase substantially compared to the baseline. This indicates that the additional reasoning mechanism introduced by SIEVE

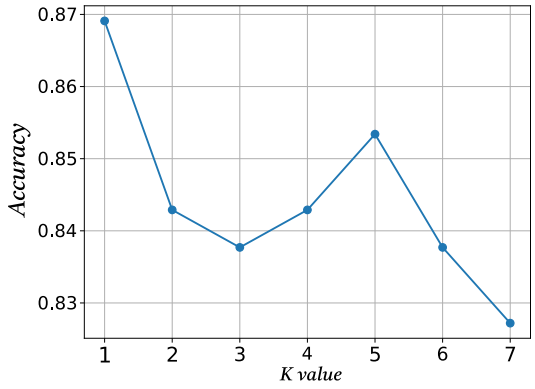


Figure 6: The relationship between the K-value and the accuracy of Vstar benchmark on Qwen3-VL-4B-Instruct.

does not impose significant computational overhead. Specifically, compared with methods that require tool invocation during inference, on Qwen3-VL-4B-Instruct, SIEVE is faster than ZoomRefine, while on Qwen3-VL-8B-Instrct it is fastest compared to tool-call methods. We attribute these differences to the varying reasoning capabilities of different models, which lead to different response behaviors across tasks and consequently different time overheads when combined with various baselines. Overall, although SIEVE introduces some additional time cost, its overhead is not the largest among the compared methods, while it consistently maintains stable performance improvements across different models and baselines. Considering both time cost and performance, SIEVE provides the best trade-off between efficiency and effectiveness.

8 HYPERPARAMETER ANALYSIS

In this section, we analyze the choice of the top-k parameter in *Identifying Visual Evidence with Textual Anchors* to justify the effectiveness of setting $k = 1$. Specifically, we conduct experiments using Qwen3-VL-4B-Instruct on the V* benchmark, varying k from 1 to 7 to evaluate how inserting different numbers of region embeddings affects model performance. The results are shown in Figure 6.

In the Figure, the model achieves the best performance when $k = 1$. As k increases, although more embeddings are inserted into the reasoning process, the performance gradually degrades. We attribute this phenomenon to the way region snapshots are constructed. In our method, patches associated with each textual anchor are aggregated to form a single region-level embedding. While this aggregation captures the key visual evidence, it inevitably introduces additional irrelevant visual information within the region. When more regions are inserted (i.e., larger k), the amount of such noisy information accumulates, which interferes with the model's reasoning process and reduces performance. These results suggest that inserting a small number of highly relevant visual embeddings is more beneficial than introducing multiple potentially noisy regions. Therefore, we adopt $k = 1$ as the default setting in our method.

9 ADDITIONAL RESULTS WITH RL-BASED METHODS

We compare SIEVE against competitive agentic RL baselines on Qwen2.5-VL-7B. As shown in the right table, SIEVE achieves performance comparable to DeepEyes* (reproduced by ourselves), and gains absolute improvement over DeepEyesV2 and Pixel-Reasoner by 3.0 and 0.5, respectively. It is important to mention that SIEVE does not call external tools and is only trained by 1.5k data samples, which is significantly fewer than tool-calling methods. In this way, SIEVE shows the potential of using internal signals instead of external tools to aid more data-efficient VLM reasoning.

Table 4: Performance of V* on serveral rl-based training methods.

Method	Acc.	Data
Vanilla	78.0	N/A
DeepEyes*	85.6	47k
DeepEyesV2	81.8	N/A
Pixel-Reasoner	84.3	7.5k+15k
SIEVE	**84.8**	**1.5k**

Table 5: Performance with different values of λ_1, λ_2, λ_3, and λ_4.

λ_1	λ_2	λ_3	λ_4	Acc.
0.6	0.3	0.2	0.2	81.68
0.6	0.3	0.6	0.2	85.34
0.6	0.3	1.0	0.2	79.58
0.6	0.3	0.5	0.2	**85.86** (Ours)

10 Discussion of the setting of rewards parameters

Following common GRPO reward design practice, we assign a smaller weight to the format reward than to the answer reward, encouraging format compliance without over-optimizing formatting.

As discussed in the ablation study, the action reward is assigned a small coefficient to stabilize policy optimization and prevent collapse. We further vary the embedding reward coefficient λ_3 and evaluate on V*. Results show stable performance around our chosen value, while large deviations cause significant degradation, demonstrating both the robustness of our parameter choice and its consistency with standard RL practice.

11 Visualization Results

11.1 Visual example of our methods

In Figure 7, using the V* dataset as a case study, we demonstrate how our model retrieves bounding boxes in image coordinate space that align with the learned region embeddings. Specifically, the selected embeddings are mapped back to their corresponding spatial patch locations and aggregated to form coherent bounding regions. The resulting visualizations show that these extracted embeddings consistently correspond to semantically meaningful and task-relevant image regions, rather than arbitrary or background areas. Although minor localization drift may occur due to the patch-based segmentation mechanism in Qwen-VL where object boundaries may not perfectly align with fixed patch grids our extended patching strategy mitigates this issue. By explicitly injecting the target object's embedding as a structured guidance signal during inference, the model is encouraged to focus on spatially relevant regions and refine its reasoning accordingly. This design not only improves visual grounding fidelity but also provides an intuitive explanation for the consistent performance gains achieved by our method across benchmarks.

11.2 Visualization between different layers extraction

11.2.1 Select Embeddings in Different Layers

We further compare the matching precision between SIEVE and the vanilla model before and after training showing that SIEVE achieves higher precision in locating the corresponding object regions. As illustrated in Figure 8, grounding quality varies across layers. Early-layer representations often fail to capture the true semantic target of the salient token, producing noisy matches, while late-layer representations tend to be overly task-specialized and biased toward output prediction. In contrast, middle-layer representations yield more accurate correspondences between salient tokens and relevant image regions, producing spatially coherent and semantically aligned patches. This



Figure 7: Examples of image regions associated with the extracted embeddings. The green box denotes the matched bounding box, and the red box is the expanded region whose embedding is fed into the model. The bottom-left corner shows a zoomed view of the red box. Minor box drift may occur due to Qwen-VL's patch segmentation.



Figure 8: This diagram shows how saliency-guided embeddings for image tokens are computed across transformer layers, highlighting how key regions are identified and how layer choice affects visual grounding quality.

observation aligns with prior findings Skean et al. (2024; 2025) that intermediate transformer layers capture richer semantic information than both shallow and late layers. These results empirically justify computing similarity and extracting region embeddings from middle-layer hidden states, which better balance semantic abstraction and spatial fidelity for cross-modal alignment.

12 More Examples

Here we choose more examples in Vstar benchmark to show the generation process of SIEVE. In these examples, the content in '[ ]' is not generated by model. We use it to hint of the inserting embeddings operation. We show the example in Figure 9, Figure 10, Figure 11 and Figure 12.



Question: What is the material of the glove?
(A) rubber (B) cotton
(C) kevlar (D) leather

Multi-turn Generation

<thought>
I need to determine the material of the glove. Looking at the image, the glove is blue and has a shiny, flexible appearance, which is typical of latex or nitrile gloves commonly used in food service. Among the options, rubber (A) is the most likely candidate, as latex gloves are often referred to as rubber gloves in casual contexts. Cotton (B) would be too soft and not shiny, kevlar (C) is too rigid and not used for food service, and leather (D) would be too stiff and not commonly used for this purpose. I need more information to confirm.
</thought>
<latent>
[inserting embeddings.....]

<thought>
I need to determine the material of the glove. From examining the image, the glove is blue, shiny, and flexible, which is characteristic of latex or nitrile gloves—both of which are types of rubber. Among the given options, (A) rubber is the most appropriate choice since latex gloves are commonly referred to as rubber gloves in everyday language. The other options—cotton, kevlar, and leather—do not match the visual properties of the glove. Therefore, I can conclude the material is rubber.

No additional information is needed.
</thought>

<answer>(C) blue</answer>

Answer Region 

Figure 9: Example of SIEVE on Vstar.



Question: What is the color of the dustpan?
(A) purple (B) red
(C) blue (D) white

Multi-turn Generation

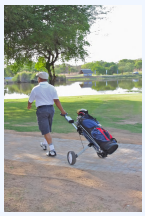
<thought>
I need to determine the color of the dustpan. Looking at the image, I can see a dustpan and brush set on the sidewalk near the storefront. The dustpan appears to be a dark color, possibly blue or black. However, considering the options provided, I should check if it matches any of them. From the image, it looks like the dustpan is blue. However, I need more information to confirm.
</thought>
<latent>
[inserting embeddings.....]
<thought>
I have sufficient information to determine the color of the dustpan from the image. It appears to be blue. Therefore, I can answer directly.

No additional information is needed.
</thought>

<answer>(C) blue</answer>

Answer Region 

Figure 10: Example of SIEVE on Vstar.



Question: What is the color of the SUV car?
(A) black (B) red
(C) silver (D) blue

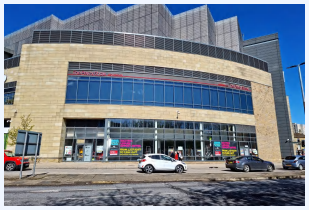
Multi-turn Generation

<thought>
I need to determine the color of the SUV car in the image. Looking at the background, there is a vehicle on the right side of the image near the trees. It appears to be a small SUV or similar vehicle. From the image, I can see that the car is a light color, possibly silver or gray, but I need to confirm. Since the image is somewhat blurry in the distance, I should rely on visible details. The car appears to have a metallic finish, which is typical of silver vehicles. I will now check if there is any definitive visual evidence to confirm the color.
</thought>
<latent>
[inserting embeddings.....]

<thought>
I need to determine the color of the SUV car. Looking at the image, there is a vehicle in the background on the right side, near the trees. It appears to be a light-colored vehicle, possibly silver or gray. The lighting in the image makes it a bit hard to tell for sure, but the reflective surface suggests a metallic finish, which is typical of silver vehicles. Given the available visual evidence, I can reasonably conclude that the car is silver.
</thought>
<latent>
[inserting embeddings.....]
<thought>
I have sufficient information to determine the color of the SUV car. From the image, the vehicle in the background appears to be silver, based on its reflective metallic finish and overall appearance. Therefore, I can now answer directly.
</thought>

<answer>(C) sliver</answer> Answer Region 

Figure 11: Example of SIEVE on Vstar.



Question: What is the color of the pink-haired woman's handbag?
(A) brown (B) white
(C) red (D) pink

Multi-turn Generation

<thought>
I need to identify the color of the pink-haired woman's handbag. First, I must locate the pink-haired woman in the image. Then, I need to examine her handbag closely to determine its color. The options are brown, white, red, or pink. Since the image shows a street scene with a building and some people, I should focus on the individuals near the building entrance, particularly those with pink hair. Once I locate her, I can observe the handbag she is carrying.
</thought>
<latent>
[inserting embeddings.....]

<thought>
I need to determine the color of the pink-haired woman's handbag. First, I must locate the pink-haired woman in the image. Looking at the image, I can see a group of people near the entrance of the building. Among them, there is a woman with pink hair. She is carrying a handbag. Now, I need to examine the color of her handbag. From the image, it appears her handbag is white. However, I need more information to confirm.
</thought>
<latent>
[inserting embeddings.....]

<thought>
I have now identified the pink-haired woman and examined her handbag. From the image, it is clear that her handbag is white. Therefore, I can directly answer the question without needing more information.
</thought>

<answer>(B) white</answer>

Answer Region 

Figure 12: Example of SIEVE on Vstar.