Lang2Act: Fine-Grained Visual Reasoning through Self-Emergent Linguistic Toolchains

Anonymous ACL submission

Abstract

Visual Retrieval-Augmented Generation (VRAG) enhances Vision-Language Models (VLMs) by incorporating external visual documents to address a given query. Existing VRAG frameworks usually depend on rigid, pre-defined external tools to extend the perceptual capabilities of VLMs, typically by explicitly separating visual perception from subsequent reasoning processes. However, this decoupled design can lead to unnecessary loss of visual information, particularly when image-based operations such as cropping are applied. In this paper, we propose **Lang2Act**, which enables fine-grained visual perception and reasoning through self-emergent linguistic toolchains. Rather than invoking fixed external engines, Lang2Act collects self-emergent actions as linguistic tools and leverages them to enhance the visual perception capabilities of VLMs. To support this mechanism, we design a two-stage Reinforcement Learning (RL)-based training framework. Specifically, the first stage optimizes VLMs to self-explore high-quality actions for constructing a reusable linguistic toolbox, and the second stage further optimizes VLMs to exploit these linguistic tools for downstream reasoning effectively. Experimental results demonstrate the effectiveness of Lang2Act in substantially enhancing the visual perception capabilities of VLMs, achieving performance improvements of over 4%. All code and data will be released on GitHub.

1 Introduction

Retrieval-Augmented Generation (RAG) has been established as a foundational framework for enhancing Large Language Models (LLMs) (Lewis et al., 2020; Liu et al., 2025) by retrieving query-related text documents and then feeding these documents as contextual knowledge to support generation (Ram et al., 2023). To extend the benefits of RAG to visual documents, existing methods (Yu et al., 2024; Faysse et al., 2024) typically adopt an end-to-end Visual Retrieval-Augmented Generation (VRAG) modeling paradigm. This design avoids the error propagation introduced by text-based RAG pipelines that rely on optical character recognition, instead directly leveraging the strong visual understanding capabilities of Vision-Language Models (VLMs). Specifically, VRAG models treat entire document-page snapshots as retrieval units and generate answers directly based on the retrieved visual pages. Such systems effectively preserve global visual context and layout structures, which are often lost in text-only representations of visually rich documents.

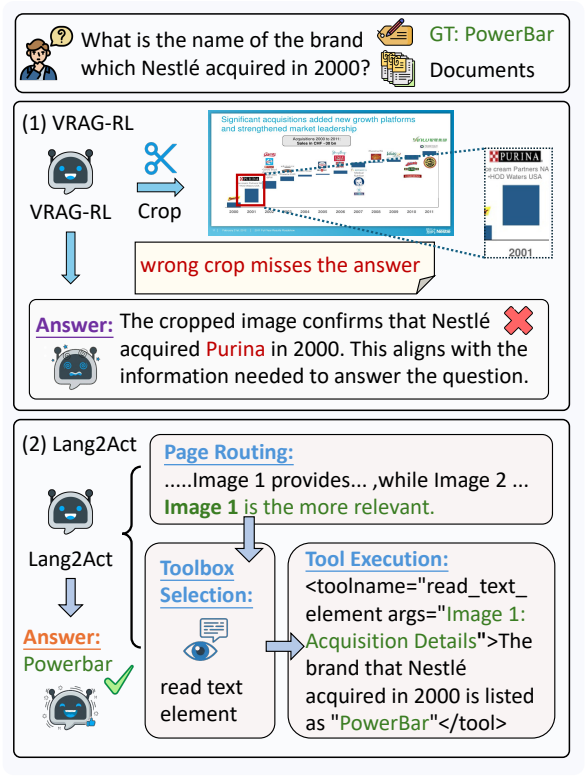


Figure 1: Comparison between VRAG-RL and the Lang2Act framework.

To further empower VLMs in handling complex

queries, some works have focused on exploiting the reasoning capabilities of VLMs to produce more accurate answers grounded in retrieved document pages. Some of them (Wu et al., 2025; Peng et al., 2025; Sun et al., 2025) employ Reinforcement Learning (RL) (Shao et al., 2024; Yu et al., 2025) to optimize VLMs for generating more effective Chain-of-Thought (CoT) (Wei et al., 2022) reasoning to derive more accurate answers. This enables VLMs to more effectively select query-relevant document pages and extract critical evidence from visual documents. By incentivizing VLMs (Bai et al., 2025; Yao et al., 2024; Peng et al., 2025) to plan search steps and integrate information across multiple sources autonomously, these approaches facilitate more effective retrieval and reasoning behaviors, thereby improving the collection and utilization of relevant external knowledge.

Instead of focusing on the denoising and utilization of retrieved documents, recent research (Wang et al., 2025c,a) aims to empower VLMs with finer-grained perceptual capabilities through interaction with explicit image tools. By leveraging external APIs to actively crop, zoom, or select specific image regions, these models can selectively attend to subtle visual details that are often overlooked by holistic processing. Through this simulation of active observation, tool-enhanced VLMs (Wang et al., 2025c,a) seek to bridge the gap between the intrinsic reasoning abilities of VLMs and low-level pixel inspection. However, such systems typically rely on rigid, pre-defined image tools that explicitly decouple visual perception from logical reasoning, constraining models to fixed mechanical operations and potentially leading to the loss of critical visual information during tool application.

To bridge this gap and unify perception with reasoning, we propose Lang2Act, a framework that enables fine-grained visual reasoning via self-emergent linguistic toolchains. As shown in Figure 1, unlike traditional methods that rely on detached external engines, Lang2Act curates a set of linguistic tools to enhance visual perception, such as `read_text_elem`, and implicitly internalizes visual operations through autoregressive generation. When a linguistic tool is decoded, the VLM dynamically modulates its attention according to the tool instruction, thereby enabling more fine-grained visual perception. To optimize how VLMs curate and utilize linguistic tools, we adopt a two-stage RL-based optimization mechanism (Yu et al., 2025), in which the model first discovers effective visual actions for problem solving through self-exploration, and subsequently strengthens the visual perception capability of VLMs by leveraging the curated linguistic toolbox.

Experimental results on multiple visual question answering benchmarks demonstrate the effectiveness of Lang2Act, which outperforms all baselines by more than 4%. Further analysis shows that, by leveraging intrinsic linguistic toolchains, Lang2Act not only facilitates accurate localization of the ground-truth image regions, but also achieves higher answer accuracy when attending to golden regions. These results indicate that Lang2Act enhances visual perception, more effectively exploits visual evidence, and alleviates hallucination. Notably, Lang2Act mitigates unexpected information loss typically introduced by image tools, relying instead on linguistic tools.

2 Related Work

Retrieval-Augmented Generation (RAG) has been established as a foundational framework for enhancing Large Language Models (LLMs) by grounding generation in external knowledge (Lewis et al., 2020; Gu et al., 2021; Liu et al., 2025). While effective for plain text, traditional RAG often discards critical layout cues and spatial structures when processing visually rich documents (Zhang et al., 2025). To address this, Visual Retrieval-Augmented Generation (VRAG) (Suri et al., 2024; Cho et al., 2024) has emerged as a significant advancement. By utilizing whole document-page snapshots as retrieval units, systems like ColPali (Faysse et al., 2024) and VisRAG (Yu et al., 2024) effectively preserve the global visual context. However, despite retaining layout information, these methods typically rely on implicit holistic processing, mapping high-dimensional visual features directly to answers without the resolution required to inspect fine-grained visual details.

To further empower RAG models in handling complex and knowledge-intensive queries, recent works leverage Reinforcement Learning (RL) (Shao et al., 2024; Yu et al., 2025) to enhance VLMs for both retrieval-augmented reasoning and search planning. For intrinsic reasoning, R1-Onevision (Yang et al., 2025), Vision-R1 (Huang et al., 2025), and ThinkLite-VL (Wang et al., 2025d) employ RL to reward Chain-of-Thought trajectories that can accurately deduce the golden answers. For information seeking, MM-

Search-R1 (Wu et al., 2025) and R1-Router (Peng et al., 2025) learn to autonomously conduct search planning, while EVisRAG (Sun et al., 2025) guides VLMs in visual evidence extraction and integration. Despite their success in accurate visual evidence exploitation, these methods fail to consider the role of RL in capturing richer visual cues for finer-grained visual understanding.

To enable finer-grained visual perception, recent works have augmented VLMs (Bai et al., 2025; Yao et al., 2024) with external visual tools. For example, Pixel-Reasoner (Wang et al., 2025a) enables pixel-level inspection by introducing explicit operations, such as zoom and select, to attend to specific image coordinates. Likewise, VRAG-RL (Wang et al., 2025c) formulates the retrieval–reasoning loop as a sequential decision process, allowing an agent to iteratively crop ambiguous regions for more detailed examination. Although these methods achieve improvements in image perception, they rely on rigid, tool-based interfaces, leading to fragmented reasoning processes and inevitable inference latency. More importantly, discrete visual operations can cause irreversible information loss due to imperfect cropping or suboptimal region selection.

3 Methodology

This section presents Lang2Act, a unified framework for fine-grained visual reasoning with Vision-Language Models (VLMs), as illustrated in Fig. 2. Given a question q and retrieved document pages $\mathcal{P} = \{p_1, \ldots, p_K\}$, the objective of the VLM is to answer the query q based on retrieved documents $\mathcal{P}$. We first introduce a linguistic tool curation process that extracts tools emerging from model reasoning trajectories (Sec. 3.1). Building upon this curated toolset, Lang2Act continuously optimizes the visual understanding capability of VLMs by incorporating linguistic tools as prompts during inference and training (Sec. 3.2).

3.1 Enhancing Visual Reasoning Trajectories for Linguistic Tool Curation

To enable VLMs to perform fine-grained and interpretable visual reasoning, Lang2Act first optimizes VLMs to generate higher-quality visual reasoning trajectories, thereby encouraging the emergence of more effective visual understanding actions. The linguistic tools distilled from successful reasoning trajectories are then collected to form the linguistic tool pool $\mathcal{T}_{\text{pool}}$.

Linguistic Action Emergence via RL Training. Existing visual understanding approaches (Sun et al., 2025) typically adopt Chain-of-Thought (CoT) prompting to facilitate visual reasoning in VLMs, including explicit visual actions such as "read the title, list related information". To collect higher-quality visual actions from reasoning trajectories, we train the VLM (π_θ) using the Decoupled Clip and Dynamic sampling Policy Optimization (DAPO) method (Yu et al., 2025) to encourage self-exploration of reasoning trajectories and retains these trajectories that can lead to correct answers.

For each training sample $(q, \mathcal{P}, a^*)$, where a^* denotes the ground-truth answer to the question q, the VLM policy π_θ generates a complete linguistic reasoning trajectory τ along with a final answer a:

$$(\tau, a) \sim \pi_\theta(\cdot \mid q, \mathcal{P}). \tag{1}$$

The model parameters θ are then optimized to maximize the expected answer reward:

$$\max_\theta \mathbb{E}_{\tau \sim \pi_\theta(\cdot|q,\mathcal{P})}[r_{\text{ans}}(\tau, a)], \tag{2}$$

where the answer reward $r_{\text{ans}}(\tau, a) \in \{0, 1\}$ is provided by an automatic reward model that evaluates the predicted answer against the reference answer.

Tool Curation. To organize and summarize the self-explored linguistic actions, we process the reasoning trajectories sequentially while maintaining a global tool pool $\mathcal{T}_{\text{pool}}$. These linguistic tools that appear most frequently are retained in $\mathcal{T}_{\text{box}}$, and each tool t in $\mathcal{T}_{\text{pool}}$ is represented as a standardized triplet consisting of a tool name, a textual description, and a parameter specification.

After obtaining the optimized exploration policy π_{θ^*}, we sample reasoning trajectories for all training instances $(q_i, \mathcal{P}_i, a_i^*)$ in the dataset. Specifically, for each instance, we sample a trajectory:

$$(\tau_i, a_i) \sim \pi_{\theta^*}(\cdot \mid q_i, \mathcal{P}_i), \tag{3}$$

and denote the resulting trajectory set as $\{\tau_i\}_{i=1}^n$. Each trajectory τ_i contains a sequence of self-explored linguistic actions executed over the retrieved document pages. When processing a trajectory τ_i, we use the optimized VLM π_{θ^*} to abstract these actions into a set of utilized tools S_i:

$$S_i \sim \pi_{\theta^*}(\tau_i, \mathcal{T}_{\text{pool}}^{(i-1)}). \tag{4}$$

The π_{θ^*} parses the actions in τ_i and aligns them to maintain a consistent format with the existing tool set $\mathcal{T}_{\text{pool}}^{(i-1)}$. And then the tool pool is updated using the produced tool set S_i:

$$\mathcal{T}_{\text{pool}}^{(i)} = \mathcal{T}_{\text{pool}}^{(i-1)} \cup S_i, \tag{5}$$

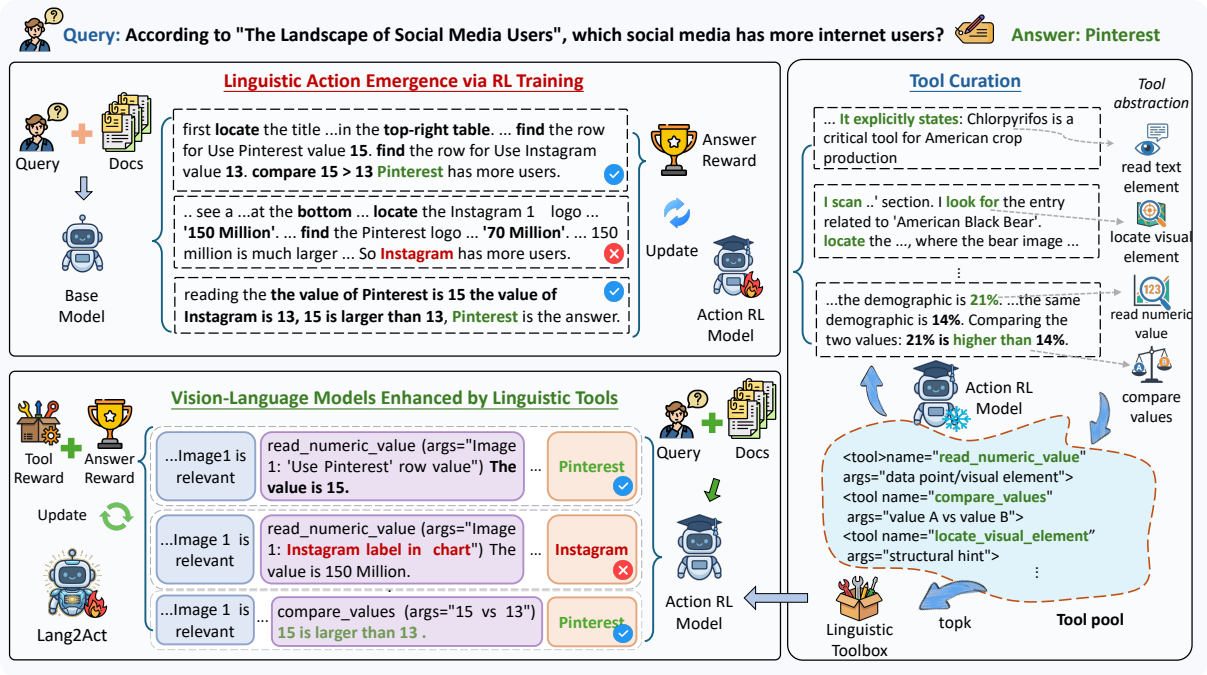


Figure 2: The overview architecture of Lang2Act.

where $\mathcal{T}_{\text{pool}}^{(i)}$ denotes the tool pool after processing trajectory τ_i, and the pool is initialized as $\mathcal{T}_{\text{pool}}^{(0)} = \emptyset$.

After processing all n trajectories, we get the final tool set $\mathcal{T}_{\text{pool}}^{(n)}$ and select the top-K most frequently used tools from the set to construct the linguistic toolbox:

$$\mathcal{T}_{\text{box}} = \text{Top}_K\left(\{(t_1, f(t_1)), \ldots, (t_m, f(t_m))\}\right), \quad (6)$$

where $t_j \in \mathcal{T}_{\text{pool}}^{(n)}$ and the ranking is determined by the usage frequency $f(t_j)$, computed as:

$$f(t_j) = \sum_{i=1}^{n} \mathbf{1}[t_j \in S_i]. \quad (7)$$

3.2 Optimizing Vision-Language Models through Linguistic Tool-Based Prompting

With the curated linguistic toolbox $\mathcal{T}_{\text{box}}$ obtained in Sec. 3.1, we further optimize the VLM parameters θ^* via reinforcement learning by augmenting the optimized VLM with linguistic tools, thereby fully leveraging the potential of these linguistic tools in visual reasoning.

Linguistic Tool Grounded Visual Reasoning. We internalize visual perception tools as verbalized functions within the curated toolbox $\mathcal{T}_{\text{box}}$, enabling the VLM to autoregressively generate a structured linguistic toolchain z together with the final answer a in a unified generation process:

$$(z, a) \sim \pi_{\theta^*}(\cdot \mid q, \mathcal{P}, \mathcal{T}_{\text{box}}). \quad (8)$$

To facilitate grounded and interpretable tool usage, the linguistic toolchain z is explicitly organized into two consecutive components:

$$z = \{z_{\text{route}}, z_{\text{exec}}\}, \quad (9)$$

where z_{route} corresponds to a visual routing stage that performs the page selection over the retrieved pages $\mathcal{P}$ and identifies a subset of relevant pages, denoted as $\mathcal{P}_{\text{sub}} \subseteq \mathcal{P}$, for subsequent reasoning. Conditioned on the selected subset $\mathcal{P}_{\text{sub}}$, the second reasoning component z_{exec} represents a linguistic tool execution stage that generates a sequence of language-level tool actions along with their corresponding execution observation:

$$z_{\text{exec}} = \{(u_1, o_1), (u_2, o_2), \ldots, (u_l, o_l)\}, \quad (10)$$

where u denotes an implicit invocation of a linguistic tool from $\mathcal{T}_{\text{box}}$, implemented via autoregressive decoding. This invocation is applied to guide VLMs to focus on specific regions of the target document in $\mathcal{P}_{\text{sub}}$ by modulating their internal visual attention. The corresponding execution observation o is generated internally by the model through language-driven reasoning over the target pages. These execution observations serve as fine-grained intermediate evidence supporting the final answer generation.

Tool-Enhanced Reasoning Optimization. To synergistically optimize the ability of π_{θ^*} to utilize the curated tools, execute them intrinsically,

and integrate visual information, we also employ DAPO (Yu et al., 2025) by maximizing the expected reward:

$$\max_{\theta} \mathbb{E}_{(z,a)\sim\pi_{\theta^*}(\cdot|q,\mathcal{P},\mathcal{T}_{\text{box}})}\big[R(z,a)\big], \tag{11}$$

where the reward $R(z,a)$ is defined as a weighted combination to balance final answer correctness and intrinsic toolchain validity:

$$R(z,a) = \alpha \cdot r_{\text{ans}}(z,a) + \beta \cdot r_{\text{tool}}(z), \tag{12}$$

where both α and β are hyperparameters. The answer reward follows Eq. 2, and the reward $r_{\text{tool}}(z)$ is a tool-usage regularization term that encourages linguistic tool usage to occur within the tool execution stage z_{exec}:

$$r_{\text{tool}}(z) = \mathbb{I}_{\text{valid}}(z \mid \mathcal{T}_{\text{box}}), \tag{13}$$

where $\mathbb{I}_{\text{valid}}(z) = 1$ if the linguistic toolchain z satisfies the prescribed structural constraints on tool usage, and 0 otherwise.

4 Experimental Methodology

This section describes the dataset, evaluation metrics, baselines, and implementation details.

Datasets. We evaluate Lang2Act on three document visual question answering benchmarks: SlideVQA (Tanaka et al., 2023), ViDoSeek (Wang et al., 2025b), and MMLongBench-Doc (Ma et al., 2024), which respectively cover presentation slides, visually rich documents, and long-context multimodal documents. Additional dataset statistics and benchmark details are provided in Appendix A.2.

Evaluation Metrics. Following Wang et al. (2025c), we adopt the LLM-as-judge method and calculate the accuracy as the primary evaluation metric. Specifically, we employ Qwen2.5-72B-Instruct (Bai et al., 2025) as an automatic evaluator to compare the predictions of models with the corresponding ground-truth answers.

Baselines. To comprehensively validate the effectiveness of Lang2Act, we compare our method with four categories of approaches: `Prompting Methods`, Vision Language Reasoning Models (`VLRMs`), Multimodal Retrieval-Augmented Generation models (`MRAGs`), and `Tool-Enhanced VLMs`.

First, the `Prompting Methods` category includes Vanilla model (Bai et al., 2025), TOT (Yao et al., 2023), and GOT (Besta et al., 2024), which serve to evaluate the native reasoning capabilities of the backbone model under different chain-of-thought prompting strategies (Wei et al., 2022). Second, the `VLRMs` category consists of R1-Onevision (Yang et al., 2025), Vision-R1 (Huang et al., 2025), ThinkLite-VL (Wang et al., 2025d), OpenVL-Thinker (Deng et al., 2025), and VisionMasters (Li et al., 2025). These methods primarily focus on enhancing reasoning capabilities through reinforcement learning or specific fine-tuning. While they possess strong logical deduction skills, they typically lack explicit tool-use mechanisms to acquire fine-grained information through tool interaction. Third, the `MRAGs` category includes VisDom (Suri et al., 2024), MM-Search-R1 (Wu et al., 2025), and EVisRAG (Sun et al., 2025). These models are explicitly optimized with multimodal retrieval-augmented generation objectives, aiming to effectively process and integrate information from retrieved document contexts. Finally, the `Tool-Enhanced VLMs` are evaluated in our experiments and serve as our primary baselines: Pixel-Reasoner (Wang et al., 2025a) and VRAG-RL (Wang et al., 2025c). These methods incorporate explicit tool execution or action-based mechanisms to support complex visual reasoning, representing the current mainstream approach of using external aids for visual tasks.

Implementation Details. We employ Qwen2.5-VL-7B-Instruct (Bai et al., 2025) as the backbone model and conduct reinforcement learning using the EasyR1 framework (Yaowei et al., 2025). Training is performed with the DAPO (Yu et al., 2025) algorithm, utilizing a group size of 8. For retrieval, we use ColQwen2 (Faysse et al., 2024) to extract the top-3 document pages per query as visual evidence. Regarding the training data, we collect approximately 11K samples from the training splits of OpenDocVQA (Tanaka et al., 2025) and SlideVQA (Tanaka et al., 2023), applying a rigorous filtering strategy to exclude low-quality or overly simple instances. Additional details on hyperparameters, training environments, and data filtering criteria are provided in Appendices A.3 and A.4.

5 Evaluation Result

This section first presents a comprehensive evaluation of Lang2Act on three challenging document-based VQA benchmarks. We then conduct ablation studies to validate the effectiveness of individual components of Lang2Act and provide an in-depth analysis of the role of linguistic tools in enhancing visual perception capabilities.

Methods	In Domain			Out of Domain									Avg.
	SlideVQA			MMLongBench-Doc						ViDoSeek			
	Single	Multi	Overall	TXT	TAB	CHA	FIG	LAY	Overall	Single	Multi	Overall	
Prompting Methods													
Direct	71.36	52.56	66.55	31.21	23.50	27.53	23.59	21.19	27.59	64.50	66.20	65.24	53.12
TOT	66.02	39.68	59.28	27.18	22.12	21.91	27.57	25.42	27.82	57.36	69.82	62.78	49.96
GOT	69.11	45.68	63.12	29.19	21.66	22.47	24.25	22.03	27.12	55.66	62.37	58.58	49.60
Vision-Language Reasoning Models (VLRMs)													
R1-Onevision	71.78	51.32	66.55	30.54	26.73	27.53	27.57	25.42	30.50	62.48	68.41	65.06	54.03
Vision-R1	78.09	53.79	71.87	32.55	27.65	26.40	29.90	30.51	32.25	60.31	68.01	63.66	55.92
ThinkLite-VL	76.88	58.02	72.05	33.22	24.88	26.97	29.90	27.12	31.66	63.88	71.43	67.16	56.96
OpenVLThinker	78.94	57.50	73.45	35.23	28.57	26.97	30.23	27.97	32.71	65.43	74.45	69.35	58.50
VisionMatters	74.76	56.08	69.98	35.91	23.96	29.21	27.91	28.81	31.66	61.40	72.23	66.11	55.91
Multimodal Retrieval-Augmented Generation Models (MRAGs)													
VisDom	73.79	50.09	67.72	33.89	21.20	21.35	18.94	21.19	25.96	62.95	70.62	66.29	53.32
MM-Search-R1	76.64	53.44	70.70	31.54	25.35	27.53	26.91	22.88	29.92	65.58	72.23	68.48	56.36
EVisRAG	79.55	54.67	73.18	32.55	27.19	25.28	27.57	27.12	30.97	66.51	73.84	69.70	57.95
Tool-Enhanced VLMs													
Pixel-Reasoner	78.13	57.50	72.84	37.37	27.19	24.72	**32.89**	27.97	33.22	67.96	70.1	68.89	58.31
VRAG-RL	81.74	49.21	73.41	31.54	24.88	30.34	30.23	20.34	31.55	64.65	71.63	67.69	57.55
Lang2Act	**83.62**	**64.55**	**78.74**	**38.59**	**31.80**	**31.46**	32.23	**33.05**	**36.55**	**74.25**	**75.35**	**74.87**	**63.39**

Table 1: Overall performance. The best results are highlighted in **bold**, and the second-best are underlined.

5.1 Overall Performance

As shown in Table 1, we compare Lang2Act against several baseline models, including prompting-based models, VLRMs, MRAG systems, and Tool-enhanced VLMs.

Overall, Lang2Act consistently outperforms all baseline models by achieving improvements of over 4% in different testing scenarios, which demonstrates its effectiveness in addressing a wide range of visual QA tasks. Compared with prompting-based models, Lang2Act achieves more than 10% gains, indicating that our training method enables the backbone model to perform more effective reasoning trajectories for visual understanding and answer questions more accurately. Based on reinforcement learning, VLRMs such as ThinkLite-VL (Wang et al., 2025d) and OpenVLThinker (Deng et al., 2025) also learn to conduct more effective reasoning for answering given questions. Even using the same reinforcement learning strategy, Lang2Act shows substantial improvements, demonstrating that leveraging carefully designed linguistic tools allows the VLM to converge to more effective reasoning trajectories after training. While MRAG models shift the focus from enhancing visual reasoning capabilities to evidence denoising and extraction, Lang2Act outperforms them, highlighting its effectiveness in query-related evidence selection and extraction. Finally, compared with Tool-enhanced baseline models that utilize image-based tools for visual perception, Lang2Act achieves over 5% improvements. This demonstrates the advantage of our self-emergent linguistic toolchain, which enables flexible, context-aware, and fine-grained visual operations, rather than relying on rigid external APIs, thereby better facilitating visual perception.

Methods	SlideVQA	ViDoSeek	MMBench	Avg.
Qwen2.5-VL-3B-Instruct				
Vanilla VLM	60.18	63.80	25.61	49.86
Vanilla DAPO	67.72	67.42	29.10	54.74
Lang2Act	**71.02**	**71.98**	**31.20**	**58.06**
w/o Action RL	69.03	69.18	30.27	56.22
w/o Tool-based RL	68.35	68.30	29.68	55.47
Qwen2.5-VL-7B-Instruct				
Vanilla VLM	66.55	65.24	27.59	53.12
Vanilla DAPO	76.79	70.23	35.86	60.96
Lang2Act	**78.74**	**74.87**	**36.55**	**63.39**
w/o Action RL	77.25	73.23	35.39	61.95
w/o Tool-based RL	76.03	72.68	35.86	61.52

Table 2: Ablation results comparing different training strategies used by Lang2Act.

5.2 Ablation Study

These experiments comprehensively validate the effectiveness of our two training strategies, namely action RL and tool-based RL, which respectively

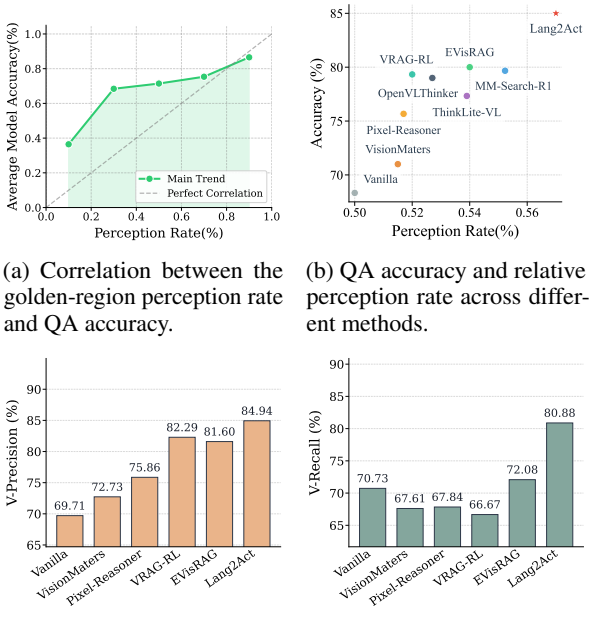
encourage the VLM to perform more visual understanding actions and to fully exploit the provided linguistic tools. To further evaluate the generalization capability of Lang2Act, we conduct experiments on both Qwen2.5-VL-3B and 7B backbones (Bai et al., 2025).

As shown in Table 2, compared with vanilla LLMs, Lang2Act yields consistent improvements of over 8% across different parameter scales (3B and 7B), demonstrating both its effectiveness and strong generalization capability. These consistent gains further indicate that our linguistic toolchain provides a generalizable solution for enhancing the visual perception capability of VLMs, largely independent of model capacity. When removing either the action RL or the tool-based RL phase, the performance of Lang2Act drops by more than 1%, indicating that both training strategies play a critical role in enabling fine-grained visual reasoning. While Vanilla DAPO improves the performance of vanilla LLMs by around 5% through enhancing the standard `think-then-answer` chain-of-thought reasoning paradigm via RL (Yu et al., 2025), it is still outperformed by our Lang2Act w/o Tool-based RL variant. This comparison highlights that the self-driven action exploration in the first stage can spontaneously discover visual grounding patterns that are more effective than generic reasoning thoughts. Furthermore, although directly optimizing LLMs to ground the linguistic tools (Lang2Act w/o Action RL) yields competitive results, it still underperforms the full Lang2Act framework. This remaining gap confirms that the initial exploration stage not only provides essential reasoning priors for curating higher-quality linguistic tools but also acts as an effective initialization that maximizes the effectiveness of subsequent linguistic tool-enhanced optimization.

5.3 Effectiveness of Linguistic Tools in Visual Document Perception

To investigate the underlying mechanisms behind the performance gains of Lang2Act, we conduct a comprehensive quantitative analysis, as shown in Figure 3, to examine how linguistic tools enhance the visual perception capability of backbone models through curated linguistic supervision.

First, we employ a vanilla LLM to analyze the relationship between perception rate and QA accuracy. As illustrated in Figure 3(a), the average accuracy of the vanilla model exhibits a strong positive correlation with the attention hit rate. This



(a) Correlation between the golden-region perception rate and QA accuracy.

(b) QA accuracy and relative perception rate across different methods.

(c) QA accuracy when models successfully perceive the golden regions.

(d) Golden-region perception ratio for correctly answered questions.

Figure 3: Quantitative analysis of image perception quality in relation to QA accuracy. We compute V-Precision (Figure 3(c)) and V-Recall (Figure 3(d)) for analysis by leveraging the perception rate on the golden region together with QA accuracy. The perception rate is calculated according to whether the model internal attention hit the golden region.

empirical observation highlights the critical role of an accurate visual perception in capturing key information from the given document pages. Motivated by this finding, we further plot the perception rate together with QA accuracy of different models in Figure 3(b). The results consistently show that QA accuracy improves as the perception rate increases. Notably, Lang2Act demonstrates its effectiveness by achieving the best performance in terms of both QA accuracy and perception rate. In contrast, the tool-enhanced VLMs, Pixel-Reasoner and VRAG-RL, exhibit nearly identical perception rates while attaining different QA accuracies. This suggests that image-tool-based methods may struggle to precisely capture key visual regions when relying on raw image operations, such as clipping.

We then conduct deeper analyses to further validate the effectiveness of different models. As shown in Figure 3(c), we report the QA accuracy on queries where models successfully attend to the golden regions of visual documents. Higher accuracy indicates a stronger ability to exploit visual evidence for question answering. Lang2Act achieves the highest score, demonstrating its superiority in

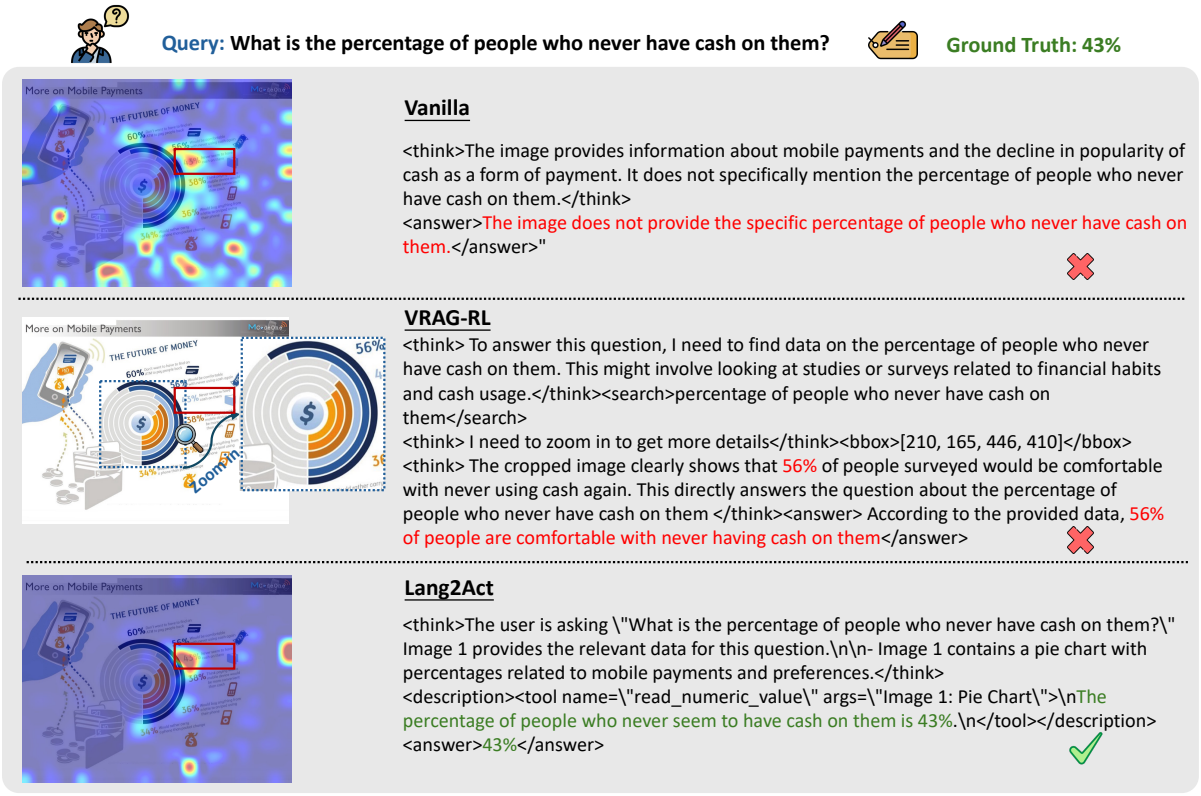


Figure 4: Case Study on SlideVQA. The red box indicates the ground truth region of the given image.

fully leveraging visual clues in the given image to generate more accurate answers. Furthermore, Figure 3(d) presents the perception rate for queries that are answered correctly. A lower rate suggests that the answer generation relies more heavily on memorized knowledge, which may increase the risk of hallucination. Lang2Act again achieves the highest perception rate, indicating its potential to alleviate knowledge conflicts and encourage stronger reliance on external visual information.

5.4 Case Study

To empirically demonstrate the effectiveness of Lang2Act, we randomly select one representative case from SlideVQA for qualitative analysis. We compare Lang2Act with a vanilla VLM and VRAG-RL. VRAG-RL relies on explicit image tools for image processing, whereas Lang2Act leverages linguistic tools to modulate visual attention, enabling more fine-grained perception.

As illustrated in Figure 4, the user queries the specific percentage of people who never carry cash. The vanilla VLM model attends to the ground-truth region but distributes its attention across multiple irrelevant regions in an attempt to gather additional visual evidence, which ultimately misleads the model and prevents it from accurately answering the question. In contrast, Lang2Act concentrates the VLM's attention on the ground-truth region and accurately generates the golden answer, demonstrating its effectiveness in enhancing the perceptual capability of VLMs. On the other hand, VRAG-RL (Wang et al., 2025c) attempts to resolve the ambiguity of visual understanding through active cropping and answers the question based on salient evidence retained after image tool execution. However, in this case, VRAG-RL performs an incorrect action by cropping out the crucial information "43%", leading to an erroneous answer of "56%". This example illustrates that enhancing visual perception through explicit image tools may introduce the risk of incorrect image operations, resulting in the loss of critical visual information.

6 Conclusion

This paper proposes Lang2Act to leverage self-emergent linguistic toolchains for fine-grained visual perception. Experimental results demonstrate the effectiveness of Lang2Act, which further internalizes visual actions to bridge the gap between reasoning and fine-grained visual perception.

Limitations

Lang2Act demonstrates superior effectiveness and efficiency compared to existing tool-enhanced VLMs, particularly in enabling fine-grained visual perception with an intrinsic linguistic toolbox. Our approach successfully concentrates the model's visual attention onto informative regions through intrinsic linguistic tools, a behavior that empirically aligns with improved answer accuracy. However, fully disentangling the strict causal dynamics between these attentional shifts and the final generation outcomes remains a complex challenge, primarily due to the inherent black-box nature of neural networks. While our current experimental analysis establishes a robust positive correlation between attention concentration and reasoning correctness, the theoretical formalization of this causality presents an open avenue for further exploration.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *ArXiv preprint*, abs/2502.13923.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multimodal retrieval is what you need for multi-page multi-document understanding. *ArXiv preprint*, abs/2411.04952.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *ArXiv preprint*, abs/2503.17352.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *ArXiv preprint*, abs/2407.01449.

Jiuxiang Gu, Jason Kuen, Vlad I. Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. 2021. Unidoc: Unified pretraining framework for document understanding. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 39–50.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *ArXiv preprint*, abs/2503.06749.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yuting Li, Lai Wei, Kaipeng Zheng, Jingyuan Huang, Linghe Kong, Lichao Sun, and Weiran Huang. 2025. Vision matters: Simple visual perturbations can boost multimodal math reasoning. *ArXiv preprint*, abs/2506.09736.

Zhenghao Liu, Pengcheng Huang, Zhipeng Xu, Xinze Li, Shuliang Liu, Chunyi Peng, Haidong Xin, Yukun Yan, Shuo Wang, Xu Han, and 1 others. 2025. Knowledge intensive agents. *Available at SSRN 5459034*.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLONGBENCH-DOC: benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Chunyi Peng, Zhipeng Xu, Zhenghao Liu, Yishan Li, Yukun Yan, Shuo Wang, Zhiyuan Liu, Yu Gu, Minghe Yu, Ge Yu, and 1 others. 2025. Learning to route queries across knowledge bases for step-wise retrieval-augmented reasoning. *ArXiv preprint*, abs/2505.22095.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv preprint*, abs/2402.03300.

Yubo Sun, Chunyi Peng, Yukun Yan, Shi Yu, Zhenghao Liu, Chi Chen, Zhiyuan Liu, and Maosong Sun. 2025. Visrag 2.0: Evidence-guided multi-image reasoning in visual retrieval-augmented generation. *ArXiv preprint*, abs/2510.09733.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A Rossi, and Dinesh Manocha. 2024. Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation. *ArXiv preprint*, abs/2412.10704.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. 2025. Vdocrag: Retrieval-augmented generation over visually-rich documents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24827–24837.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13636–13645. AAAI Press.

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

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025b. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. *ArXiv preprint*, abs/2502.18017.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025c. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. *ArXiv preprint*, abs/2505.22019.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. Mmsearch-r1: Incentivizing lmms to search. *ArXiv preprint*, abs/2506.20670.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, and 1 others. 2025. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *ArXiv preprint*, abs/2503.10615.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *ArXiv preprint*, abs/2408.01800.

Zheng Yaowei, Lu Junting, Wang Shenzhi, Feng Zhangchi, Kuang Dongdong, and Xiong Yuwen. 2025. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv preprint*, abs/2503.14476.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and 1 others. 2024. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *ArXiv preprint*, abs/2410.10594.

Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. 2025. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17443–17453.

A Appendix

A.1 License

We summarize the licenses and usage terms of the datasets used in this work. ViDoseek and MMLongBench-Doc are released under the Apache 2.0 license. SlideVQA and OpenDocVQA are provided under the NTT Software Evaluation License, which permits non-commercial academic use for research and evaluation purposes. We strictly follow the original licensing terms of all datasets and do not redistribute any third-party raw data.

A.2 Additional Details of Datasets

To comprehensively evaluate Lang2Act's ability to maintain a continuous reasoning context and mitigate visual hallucinations, we conduct experiments on three representative benchmarks covering diverse document scenarios. We first employ SlideVQA (Tanaka et al., 2023) to assess cross-slide reasoning over interconnected text and diagrams, serving as a rigorous testbed for aggregating fragmented visual evidence without the context loss often induced by rigid cropping. To further validate fine-grained perception in dense layouts, we utilize ViDoSeek (Wang et al., 2025b), which challenges the model to precisely capture specific attributes in visually rich documents where raw image operations often fail. Additionally, we incorporate MMLongBench-Doc (Ma et al., 2024) to examine performance on long multimodal documents, ensuring our method sustains accurate visual grounding over extended horizons and effectively alleviates the risk of hallucination caused by error accumulation. Table 3 summarizes the query counts and dataset characteristics.

A.3 Experimental Details of Data Filtering

In the Tool-Based Optimization training, we employ the model obtained from the Action Exploration as the initialization. For each training sample, the model generates eight candidate completions through stochastic sampling. Samples for which all eight completions are correct are removed, as they provide a limited learning signal for further optimization. After filtering, approximately 5,700 samples are retained to form the training set for the second-stage RL, focusing the optimization on more challenging instances. Figure 5 illustrates the distribution of sample difficulty before and after filtering.

Dataset	#Queries
SlideVQA	2215
ViDoSeek	1142
MMLongBench-Doc	859

Table 3: Dataset statistics.

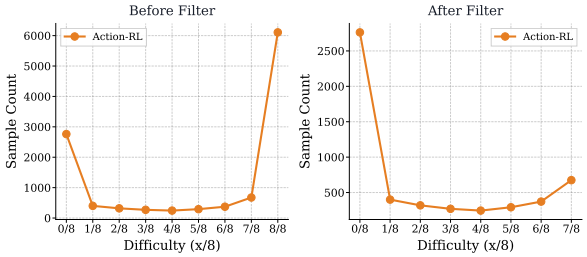


Figure 5: Distribution of sample difficulty before and after filtering in the Tool-Based Optimization training.

A.4 Additional Implementation Details

All experiments are conducted on NVIDIA A800 GPUs. The detailed hyperparameters we use during the training period of Action RL and Tool-based RL are shown in Table 4 and Table 5.

Reward Function for DAPO Training. We follow the reward formulation defined in Eq. 12. In all experiments, we set $\alpha = 0.8$ and $\beta = 0.2$. The answer reward $r_{\text{ans}}(z, a)$ evaluates whether the predicted answer is correct. Following Eq. 2, we adopt an automatic evaluator to compare the generated answer a , which is extracted from the <answer> block, with the ground-truth answer a^*, and assign a binary score:

$$r_{\text{ans}}(z, a) = \begin{cases} 1, & \text{if the generated answer is correct .} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Beyond the answer-based reward, we also incorporate a tool reward that enforces structured reasoning by requiring the model to generate outputs following a fixed tag order, namely <think>, <description>, and <answer>. In particular, linguistic tool invocations are constrained to appear only within the <description> block and must conform to the curated toolbox, thereby encouraging disciplined and well-structured tool usage during visual reasoning.

Retrieval Implementation Details. We use ColPali (Faysse et al., 2024) as the visual embedding model to encode document pages and queries for retrieval. We build and query the retrieval index with LlamaIndex using similarity search. Unless otherwise specified, we retrieve the top-3 pages for each query and provide the same retrieved evidence

Epochs	1
Rollout batch size	48
Global batch size	8
Max grad norm	1.0
Data type	bf16
Learning rate	1.0e−6
Weight decay	1.0e−2
KL coefficient	1.0e−2
Rollout temperature	1.0
Epsilon	0.2
Epsilon High	0.28
Max prompt length	8192
Max response length	2048
Image max pixels	401408

Table 4: Hyperparameters for Action RL.

Epochs	3
Rollout batch size	128
Global batch size	64
Max grad norm	1.0
Data type	bf16
Learning rate	1.0e−6
Weight decay	1.0e−2
Rollout temperature	1.0
Epsilon	0.2
Epsilon High	0.28
Max prompt length	8192
Max response length	2048
Image max pixels	401408

Table 5: Hyperparameters for Tool-based RL.

to all methods for evaluation. Table 6 reports the retrieval performance of our retriever across the evaluated benchmarks.

Baselines and Comparison Setup. We compare our method with a diverse set of strong baselines covering vision-language reasoning models and retrieval-augmented generation approaches.

R1-Onevision-7B (Yang et al., 2025) proposes a unified multimodal reasoning framework by formally aligning visual and textual representations. It leverages reinforcement learning to improve cross-modal reasoning without relying on task-specific heuristics.

Vision-R1-7B (Huang et al., 2025) further extends reinforcement learning to multimodal large language models by introducing vision-guided reward signals. This approach incentivizes step-by-step reasoning grounded in visual perception, eliminating the need for human-annotated preference data.

Dataset	Recall@1	Recall@3	Recall@5	MRR@5
ViDoSeek	75.40	89.70	95.10	83.30
SlideVQA	92.91	98.10	98.96	95.53
MMLongBench-Doc	49.00	61.70	66.50	55.70
Avg.	72.44	83.17	86.85	78.18

Table 6: Retrieval performance of ColQwen2 on three datasets. Metrics are Recall@k and MRR@5 (%).

OpenVLThinker-7B (Deng et al., 2025) investigates iterative self-improvement for LVLM reasoning by alternating SFT and GRPO-style RL, showing that SFT can surface useful reasoning behaviors and narrow the search space for subsequent RL, leading to stronger multi-step visual reasoning.

ThinkLite-VL (Wang et al., 2025d) adopts a lightweight reasoning-oriented training strategy that emphasizes efficient self-improvement. It alternates between supervised fine-tuning and reinforcement learning, enabling the model to progressively refine its multimodal reasoning capability with reduced computational overhead.

VisionMatters (Li et al., 2025) revisits multimodal reasoning from the perspective of image perturbation. Systematically analyzing how visual variations affect model predictions, it enhances robustness and visual sensitivity through targeted fine-tuning.

Pixel-Reasoner (Wang et al., 2025a) explicitly encourages fine-grained pixel-space reasoning via curiosity-driven reinforcement learning.

MM-Search-R1 (Wu et al., 2025) equips multimodal models with explicit search capabilities, enabling iterative retrieval and reasoning over external visual evidence.

EVisRAG (Sun et al., 2025) addresses the challenge of multi-image integration in VRAG systems. It proposes an evidence-guided paradigm trained via Reward-Scoped GRPO (RS-GRPO), which incentivizes the model to explicitly extract evidence from individual images before synthesizing the final answer

VRAG-RL (Wang et al., 2025c) further integrates reinforcement learning into the RAG paradigm, optimizing vision-perception-driven retrieval and reasoning through iterative policy improvement.

A.5 Additional Baseline Comparison Results

In addition to the LLM-based evaluation reported in the main paper, we provide supplementary re-

Methods	In Domain			Out of Domain									Avg.
	SlideVQA			MMLongBench-Doc						ViDoSeek			
	Single	Multi	Overall	TXT	TAB	CHA	FIG	LAY	Overall	Single	Multi	Overall	
Prompting Methods													
Direct	59.16	49.74	56.75	35.91	25.35	33.15	29.90	30.51	32.36	22.64	55.33	36.87	41.96
TOT	55.04	35.45	50.02	34.23	23.96	30.90	33.89	36.44	33.88	21.24	59.15	37.74	40.56
GOT	60.92	51.32	58.47	37.58	25.81	32.02	33.89	35.59	35.04	24.50	55.13	37.83	43.78
Vision-Language Reasoning Models (VLRMs)													
R1-Onevision	59.89	47.80	56.79	37.58	27.65	32.58	36.88	33.05	36.09	16.43	57.95	34.50	42.46
Vision-R1	72.63	56.61	68.53	41.28	28.11	38.20	40.20	**42.37**	40.28	32.71	60.56	44.83	51.21
ThinkLite-VL	65.84	53.97	62.80	37.25	27.65	30.34	37.21	33.05	35.74	26.36	61.77	41.77	46.77
OpenVLThinker	69.24	52.38	64.92	41.61	28.57	33.71	35.88	38.14	37.49	29.77	61.97	43.78	48.73
VisionMatters	61.47	51.85	59.01	37.25	24.88	33.15	33.22	33.90	34.58	24.34	61.37	40.46	44.68
Multimodal Retrieval-Augmented Generation Models (MRAGs)													
VisDom	58.92	49.38	56.48	29.87	20.28	24.72	21.59	27.12	26.78	18.76	57.14	35.46	39.57
MM-Search-R1	65.47	50.09	61.53	35.57	23.96	34.27	33.89	33.05	33.99	25.12	61.37	40.89	45.47
EVisRAG	73.85	52.20	68.31	37.92	27.19	32.02	33.89	38.14	35.86	42.33	61.67	50.53	51.56
Tool-Enhanced VLMs													
Pixel-Reasoner	70.33	54.14	66.19	42.09	26.73	36.52	40.20	38.14	39.28	36.55	63.12	48.51	51.32
VRAG-RL	64.93	54.67	62.30	36.58	**33.90**	**39.33**	25.81	33.55	35.97	24.81	61.57	40.81	46.36
Lang2Act	**76.78**	**61.73**	**72.78**	**43.96**	27.19	36.52	**40.86**	38.14	**40.51**	**44.34**	**65.39**	**53.50**	**55.60**

Table 7: Overall performance by using accuracy score for evaluation. The best results are highlighted in **bold**, and the second-best are underlined.

sults using an automatic accuracy metric in Table 7. This evaluation directly compares predicted answers with ground-truth responses to compute exact-match accuracy, without relying on an external judge model.

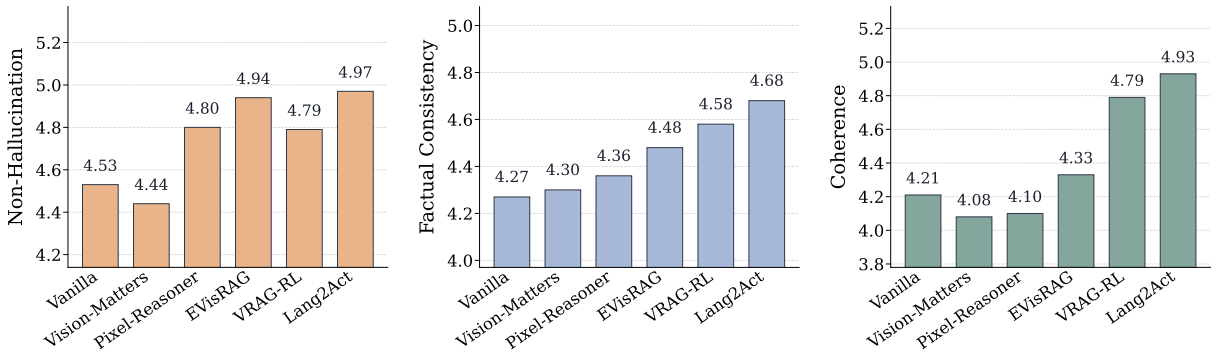
The overall performance trends of accuracy remain consistent with those evaluated in LLM-as-judge in Table 1, demonstrating that the improvements of Lang2Act are not sensitive to the choice of evaluation protocol.

A.6 Tool Frequency of the Curated Toolbox

To analyze the behavioral patterns emerging from the self-driven exploration phase, we sampled 1,500 reasoning trajectories and aggregated the usage frequency of each linguistic tool. The statistical distribution is presented in Table 8. The results reveal a clear hierarchy in visual reasoning. The most frequently employed tools are Perception-Oriented actions, specifically `read_text_element` (64.26%) and `read_numeric_value` (41.73%). This dominance indicates that the model prioritizes precise information extraction as the foundation for answering document-based queries. Following perception, reasoning-oriented tools such as `identify_entity_attribute`, `compare_values`, and `locate_visual_element` exhibit stable usage frequencies, demonstrating the model's capability to perform structural analysis and comparative reasoning after grounding the visual evidence. Based on this frequency distribution, we observe a significant long-tail effect. Tools ranked 8th and below (e.g., specific arithmetic operations like `subtract_values`) appear with negligible frequency, representing outlier cases that contribute little to generalizability. Consequently, to construct a compact and efficient linguistic toolbox $\mathcal{T}_{box}$, we selected the top-7 most frequent tools. This selection covers over 98.9% of the total tool usage observed in the sampled trajectories, ensuring that the final toolbox encapsulates the core visual reasoning primitives while filtering out sparse, task-specific noise.

Tool Name	Count	Frequency
`read_text_element`	964	64.26%
`read_numeric_value`	626	41.73%
`identify_entity_attribute`	259	17.26%
`compare_values`	259	17.26%
`locate_visual_element`	245	16.33%
`compute_percentage`	189	12.60%
`infer_missing_information`	41	2.73%
`subtract_values`	20	1.33%
`add_values`	5	0.33%
`count_matching_values`	3	0.20%

Table 8: Frequency of tool usage sorted by count.



(a) Non-hallucination performance. (b) Factual consistency scores. (c) Coherence evaluation.

Figure 6: Large language model–based evaluation of response quality under successful attention perception. All scores are computed only on samples where the model attention successfully perceives the relevant (golden) regions. We report performance from three complementary perspectives: (a) non-hallucination, (b) factual consistency, and (c) coherence, to assess the quality of model responses under reliable visual perception.

A.7 Performance Analysis under Oracle Retrieval

To rigorously assess the model's fine-grained visual reasoning capabilities and its resilience to interference, we conducted an oracle setting experiment in Table 9, where ground-truth pages containing the answer are directly provided, supplemented by distractor images to ensure a minimum input of three pages per query.

In the oracle setting, where the correct document is guaranteed, VRAG-RL (Wang et al., 2025c) exhibits competitive performance, validating that its active cropping mechanism effectively enhances perception by physically zooming into specific regions. However, Lang2Act consistently surpasses this strong baseline, particularly on detail-intensive benchmarks like ViDoSeek (Wang et al., 2025b). This performance advantage demonstrates that while mechanical cropping improves resolution, it inherently risks severing the semantic link between local details and the global layout, whereas our linguistic toolchain maintains the holistic context required for complex interpretation.

Methods	Vidoseek	SlideVQA	MMBench	Avg.
Vanilla	72.85	64.60	32.56	56.67
R1-OneVision	73.56	67.67	36.05	59.09
Vision-R1	71.28	72.19	38.89	60.78
ThinkLite-VL	75.48	71.83	39.02	62.11
OpenVLThinker	70.23	72.91	38.37	60.26
VisionMaters	73.29	70.56	37.86	60.57
Pixel-Reasoner	71.54	69.53	38.16	59.74
MM-Search-R1	77.58	71.69	37.98	62.41
EVisRAG	72.94	74.63	40.83	62.80
VRAG-RL	73.73	75.67	39.31	62.90
Lang2Act	**79.95**	**79.68**	**43.28**	**67.63**

Table 9: Reasoning performance across three benchmarks under oracle retrieval.

A.8 Analysis of Model Confidence and Response Quality

To deeply evaluate the quality of the generated reasoning chains beyond simple accuracy, we employed an advanced LLM judge to assess three critical dimensions: hallucination, factual consistency, and coherence, using the evaluation prompt illustrated in Figure 17. As shown in Figure 6, the results indicate that existing tool-enhanced approaches often struggle with coherence and consistency, primarily due to the context fragmentation caused by rigid raw image operations. By physically isolating visual regions, these methods sever the semantic connection between local details and global structures, frequently forcing the model to hallucinate information to bridge logical gaps. In contrast, Lang2Act achieves superior performance by leveraging its self-emergent linguistic toolchain to internalize visual perception into the autoregressive generation process. This design maintains a continuous reasoning flow where visual grounding is tightly coupled with logical deduction, effectively suppressing hallucinations and ensuring that the generated trajectories remain factually consistent and logically coherent.

A.9 Inference Latency of Lang2Act.

We evaluate inference latency on the SlideVQA dataset, measuring the end-to-end runtime required to generate a final answer for each query. All methods are evaluated under the same experimental settings to ensure a fair comparison as shown in Table 10.

Method	Latency (s)
Vanilla	48.77
VisionMatters	51.66
OpenVLThinker	72.93
EVisRAG	77.45
Pixel-Reasoner	94.45
VRAG-RL	123.78
Lang2Act	50.49

Table 10: Average end-to-end inference latency on SlideVQA

A.10 Prompt Examples

Below are sample prompts used for multimodal reasoning tasks. This section provides the specific prompt templates used for the baselines and our method.

Figure 7 presents the Vanilla prompt, which instructs the model to conduct internal reasoning within `<think>` tags before providing a direct answer, serving as the standard baseline. Figure 8 illustrates the Action RL prompt, requiring the model to explicitly describe the visual evidence used for reasoning, which is utilized to generate high-quality training data. Figure 9 presents the Tools Curation prompt, which deconstructs these reasoning steps into atomic, structure-aware cognitive operations to construct the tool pool.Figure 10 displays the EVisRAG (Sun et al., 2025) prompt, which enforces a strict four-step structured reasoning process: observing images, recording evidence, reasoning, and answering. Figure 11 depicts the prompt for our proposed Lang2Act framework. It defines a set of linguistic tools (e.g., numerical extraction, visual element identification) in the context and requires the model to perform fine-grained analysis and grounding of visual information using these tools within the `<description>` tag before generating the final answer.

Regarding complex reasoning strategies, Figure 12 details the Tree-of-Thoughts (ToT) prompt (Yao et al., 2023), guiding the model to deconstruct the problem into sub-problems and evaluate the validity of multiple reasoning branches. Figure 13 presents the Graph-of-Thoughts (GOT) prompt (Besta et al., 2024), asking the model to generate initial thoughts and then refine and merge them to construct a comprehensive reasoning graph. For tool-enhanced methods, Figure 14 illustrates the VRAG-RL prompt (Wang et al., 2025c), which allows the agent to query knowledge via a search engine and execute image cropping using `<bbox>` tags to acquire local details. Figure 15 shows the PixelReasoner prompt (Wang et al., 2025a), which adopts a specialized function-call format to enable the model to execute pixel-level image cropping operations based on normalized bounding boxes. Finally, Figure 16 provides the VLM judge prompt used for automatic evaluation, where an expert system validates the correctness of the model's generated answer against the ground truth.

Vanilla Prompt.

System Prompt:

Answer the given question based on the {num_images} image(s) provided. You must conduct reasoning inside <think> and </think> first. After reasoning, you should directly provide the answer inside <answer> and </answer>, without detailed illustrations.

User Prompt:

Query: {Query Description}

Images: {Retrieved Images}

Figure 7: Prompt of Vanilla.

Action RL Prompt.

System Prompt:

You are a specialized AI assistant for visual question answering based on multiple provided document images. Your task is to answer the user's question by carefully analyzing all images.

Your response must strictly follow this format:
<think>...</think>
<description>...</description>
<answer>...</answer>

Guidance:
- You have exactly {num_images} image(s). Analyze each briefly in <think>, then conclude which one(s) you used.
- In <description>, describe only the visual evidence you actually used, and clearly indicate where it appears in the image.
- In <answer>, output only the final concise answer.

User Prompt:

Query: {User Question}

Images: {Retrieved Images}

Figure 8: Prompt of Action RL.

Tools Curation Prompt.

System Prompt:

You are the Lead Architect of a Document Visual Reasoning System. Deconstruct questions into atomic, structure-aware cognitive operations.

CORE PHILOSOPHY
1) Structure Awareness: Tools must reflect layout (rows, columns, axes).
2) Atomic Data Extraction: Locate region first, then extract data.
3) Analytical Calculation: Define precise math tools (subtract, rank_values).

CURRENT TOOL POOL
{tool_pool_text}

GUIDELINES: DESIGNING DOCUMENT TOOLS
- Tables/Grids: Navigate rows/columns (e.g., locate_table_row, read_cell_value).
- Charts/Graphs: Map visuals to values (e.g., map_bar_to_axis).
- Reasoning: Define specific logic tools for calculation/comparison.

COGNITIVE CAPABILITY SPECTRUM
* Layout: (locate_row/col, find_title, find_legend, intersect_regions)
* Data: (read_text, read_numeric, extract_key_value_pair)
* Chart: (trace_line_trend, get_bar_height, map_color_to_category)
* Math & Logic: (compute_pct, subtract_values, find_max, count_rows)
* Verify: (verify_signature_presence, check_checkbox_status)

OUTPUT FORMAT (MANDATORY)
1. New Definitions: `DEFINE_TOOL: name || args || desc`
2. Applications: `<tool name="..." args="...">reasoning</tool>`
3. End: `END_OF_TOOLS`

EXAMPLE (Structure & Math)
DESC: Found 'Q3 Revenue', read value, compared to 'Q2', calculated growth.
OUTPUT:
DEFINE_TOOL: subtract_values || val1, val2 || Calculate difference.
<tool name="locate_table_row" args="row 'Q3 Revenue'">Row 4</tool>
<tool name="read_cell_value" args="Row 4, col 'Amount'">$150M</tool>
<tool name="subtract_values" args="150, 100">50</tool>
END_OF_TOOLS

User Prompt:

Analyze the reasoning steps.

DESCRIPTION: {description}

OUTPUT:

Figure 9: Prompt of Tools Curation.

EVisRAG Prompt.

System Prompt:

You are an AI Visual QA assistant. I will provide you with a question and several images. Please follow the four steps below.

Step 1: Observe the Images

First, analyze the question and consider what types of images may contain relevant information. Then, examine each image one by one, paying special attention to aspects related to the question. Identify whether each image contains any potentially relevant information.
Wrap your observations within <observe>...</observe> tags.

Step 2: Record Evidences from Images

After reviewing all images, record the evidence you find for each image within <evidence>...</evidence> tags.
If you are certain that an image contains no relevant information, record it as: [i]: no relevant information (where i denotes the index of the image).
If an image contains relevant evidence, record it as: [j]: [the evidence you find for the question] (where j is the index of the image).

Step 3: Reason Based on the Question and Evidence

Based on the recorded evidence, reason about the answer to the question.
Include your step-by-step reasoning within <think>...</think> tags.

Step 4: Answer the Question

Provide your final answer based only on the evidence you found in the images.
Wrap your answer within <answer>...</answer> tags.
Avoid adding unnecessary contents in your final answer; for example, if the question is a yes/no question, simply answer <answer>yes</answer> or <answer>no</answer>.
If none of the images contain sufficient information to answer the question, respond with <answer>insufficient to answer</answer>.

Formatting Requirements:

Use the exact tags <observe>, <evidence>, <think>, and <answer> for structured output.
It is possible that none, one, or several images contain relevant evidence.
If you find no evidence or too little evidence to answer the question, follow the instructions above for insufficient information.

User Prompt:

Query: {User Question}

Images: {Retrieved Images}

Figure 10: Prompt of EVisRAG for evidence-structured visual question answering.

Lang2Act Prompt.

System Prompt:

You are a specialized AI assistant for visual question answering. Your task is to answer the user's question by carefully analyzing all the provided images.

Your response must strictly follow this XML format:
<think>...</think>
<description>...</description>
<answer>...</answer>

Guidance:
1. In <think>, analyze all {num_images} images and state which one(s) contain relevant evidence.
2. In <description>, focus only on the selected images and describe your reasoning process using the tools below.
3. In <answer>, provide only the final, concise answer grounded in visual evidence.

Available Tools for <description>:
– <tool name="locate_visual_element" args="Image k: structural hint"> Locate specific visual elements or regions based on structural hints. </tool>
– <tool name="read_text_element" args="Image k: locator/region"> Read and transcribe visible text from the located region. </tool>
– <tool name="read_numeric_value" args="Image k: data point"> Extract specific numeric values or counts from visual elements. </tool>
– <tool name="identify_entity_attribute" args="Image k: entity"> Identify specific attributes associated with entities. </tool>
– <tool name="compare_values" args="Image k: value A vs value B"> Compare quantitative values to determine ordering or equality. </tool>
– <tool name="compute_percentage" args="part_value, total_value"> Compute the percentage based on given values. </tool>
– <tool name="infer_missing_information" args="Image k: data"> Infer missing information based on given data. </tool>

User Prompt:

Query: {User Question}

Images: {Retrieved Images}

Figure 11: Prompt of Lang2Act.

ToT Prompt.

System Prompt:

You are an AI assistant. I will provide a query and {num_images} image(s). You must use a 'Tree of Thoughts' approach to arrive at the answer.

Follow these two steps:

In the first step (within the <think> tag):
1. **Deconstruct the Problem**: Break down the main question into smaller, manageable sub-problems.
2. **Generate Multiple Thoughts**: For each sub-problem, generate at least two potential lines of reasoning or 'thoughts' on how to solve it using the provided images.
3. **Evaluate Thoughts**: Assess each thought's validity. Analyze the evidence in the images that supports or refutes each thought. State which thoughts are dead ends and which are promising.
4. **Conclude**: Based on your evaluation, synthesize the most promising thoughts to form a final, coherent reasoning chain.

In the second step (within the <answer> tag):
Provide only the final, concise answer that results from your 'Tree of Thoughts' analysis. If the question asks for a 'yes' or 'no', only provide that.

User Prompt:

Query: {User Question}

Images: {Retrieved Images}

Figure 12: Prompt of Tree-of-Thoughts (ToT).

GOT Prompt.

System Prompt:

You are an AI assistant. I will provide a query and {num_images} image(s). You must use a 'Graph of Thoughts' approach to solve the problem.

Follow these two steps:

In the first step (within the <think> tag):
1. **Generate Initial Thoughts**: Create several initial, independent thoughts or approaches to answering the question based on the images.
2. **Transform and Refine**: For each thought, consider how it can be improved, refined, or combined with others. Merge promising thoughts into a more powerful, synthesized line of reasoning. Discard thoughts that are incorrect.
3. **Structure as a Graph**: Explain your final reasoning process as a graph where thoughts are nodes. Show how you progressed from initial thoughts to the final synthesized conclusion.

In the second step (within the <answer> tag):
Provide only the final, concise answer derived from your 'Graph of Thoughts' analysis.

User Prompt:

Query: {User Question}

Images: {Retrieved Images}

Figure 13: Prompt of Graph-of-Thoughts (GOT).

VRAG-RL Prompt.

System Prompt:

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you lack knowledge, you may call a search engine via <search> query </search>. When an image is retrieved, you may crop it using <bbox>[x1, y1, x2, y2]</bbox>. Repeat as needed. If no further knowledge is needed, provide the answer within <answer> and </answer>. For example, <answer> Beijing </answer>.

User Prompt:

Query: {Query Description}

Figure 14: Prompt of VRAG-RL.

PixelReasoner Prompt.

System Prompt:

You are a helpful assistant. You may call one or more functions to assist with the user query. You are provided with function signatures within <tools> XML tags (specifically crop_image_normalized to zoom in based on normalized bbox). For each function call, return a json object with function name and arguments within <tool_call> XML tags: <tool_call> { "name": ..., "arguments": ... } </tool_call>.

User Prompt:

Query: {Query Description}

Guidelines: Understand the given visual information and the user query. Determine if it is beneficial to employ the given visual operations (tools). For a video, we can look closer by select_frames. For an image, we can look closer by crop_image_normalized. Reason with the visual information step by step, and put your final answer within \boxed{}.

Figure 15: Prompt of PixelReasoner.

VLM Judge Prompt.

System Prompt:

You are an expert evaluation system for a question answering chatbot. You will be given one evaluation item. You will see a query, a reference answer, and a generated answer. Your task is to evaluate the correctness of the generated answer. Your response MUST be exactly one line, formatted as <judge>True</judge> if the generated answer is correct, or <judge>False</judge> otherwise. Do not add any other text or explanations.

User Prompt:

Query: {Query}

Reference Answer: {Reference Answer}

Generated Answer: {Model Answer To Evaluate}

Figure 16: Prompt of the automatic judge for single-item evaluation.

LLM Judge Prompt for Reasoning Quality.

System Prompt:

You are an expert evaluator for Large Language Models. Your task is to evaluate the quality of a model's "Chain of Thought" (reasoning process) and final answer based on the user's Query and the provided Gold Answer.
Please evaluate the [Model Response] based on the following three specific dimensions. For each dimension, assign a score from 1 to 5 stars.

1. Coherence (Reasoning Logic & Fluency)

Definition: Evaluates whether the chain of thought is logically sound, structured, and easy to follow.
(1 Star: Disjointed/Confusing; 3 Stars: Readable but with leaps; 5 Stars: Perfectly smooth/Logical).

2. Non-Hallucination (Faithfulness)

Definition: Evaluates whether the response contains fabricated information.
(1 Star: Major fabrications; 3 Stars: Minor errors; 5 Stars: Entirely truthful).

3. Factual Consistency (Alignment with Gold Answer)

Definition: Evaluates whether the model's final conclusion aligns with the Gold Answer.
(1 Star: Contradictory; 3 Stars: Partially consistent; 5 Stars: Fully consistent).

Output Format:

Please strictly follow this format:
Coherence: [Score]
Non-Hallucination: [Score]
Factual Consistency: [Score]
Average: [Average Score]
Explanation: [Brief explanation]

User Prompt:

Query: {Query}

Gold Answer: {Gold Answer}

Model Response: {Model Response}

Figure 17: Prompt used for the automatic evaluation of reasoning coherence, faithfulness, and factual consistency.