

SeeingEye: Agentic Information Flow Unlocks Multimodal Reasoning in Text-Only LLMs

Anonymous ACL submission

Abstract

Recent text-only large language models (LLMs), such as DeepSeek-R1, show strong reasoning ability but remain fragile or unusable on multimodal tasks. Existing methods often convert images into a single caption, which lacks the diversity and task adaptivity needed for Visual Question Answering (VQA) and provides no principled channel for fine-grained visual evidence. We introduce **SeeingEye**, a modular framework that unlocks multimodal reasoning in text-only LLMs through a small VLM translator agent. The translator acts as a perception agent: it invokes specialized tools (*e.g.*, OCR and crop) and iteratively distills multimodal inputs into structured intermediate representations (SIRs) tailored to the question. These SIRs are passed to a text-only LLM reasoner, and multi-round feedback lets the reasoner request targeted visual details before answering. Experiments on knowledge-intensive VQA benchmarks, including MMMU-Pro, MMMU, OCR-BenchV2, and MIA-Bench, show that SeeingEye reduces inference cost while outperforming strong end-to-end VLMs on most reported benchmarks. For example, a 3B vision translator paired with an 8B language reasoner outperforms a monolithic 32B VLM on challenging knowledge-based questions.

1 Introduction

Recent text-only LLM reasoners, such as DeepSeek-R1, have demonstrated remarkable text-only reasoning, pushing the frontiers of artificial intelligence in tasks from code generation to complex problem-solving (Brown et al., 2020; Guo et al., 2025). Compared to multimodal reasoners, they enjoy a wide adoption and cost efficiency, but lack multimodal reasoning capabilities. This raises a central research question: can we bridge text-only LLM reasoners with multimodal reasoning capabilities that are effective and more cost-efficient than multimodal reasoning models?

A common paradigm to answer this question has centered on converting the visual input into text. Early approaches relied on generating static, single-form captions, from general descriptions to more query-focused variants (Khademi et al., 2023; Özdemir and Akagündüz, 2024; Ma et al., 2024). However, these non-interactive descriptions lack the adaptability for diverse VQA tasks and create a fixed information bottleneck. Recognizing this, more recent works introduce dynamic primitives like tool use (Wu et al., 2023) or integrated active perception (Wu and Xie, 2024). While a significant step forward, these methods present new limitations: the information flow is often an unstructured conversational history of tool calls, or the perception and reasoning modules are tightly coupled within monolithic VLMs. Such architectures are difficult to scale and cannot easily leverage the distinct, rapidly advancing power of state-of-the-art text-only reasoners. Consequently, even advanced systems still lack a formal, structured medium for information exchange, namely an efficient channel that allows a powerful, text-only reasoning agent to iteratively query and comprehend visual information.

Motivated by these limitations, we argue the key to unlocking multimodal reasoning in text-only LLMs is not to simply describe, but to actively *translate*. We introduce **SeeingEye**, a novel, modular framework that reconceptualizes the vision component as an agent-based translator rather than a passive descriptor. The Translator interacts with the visual input by invoking specialized tools, such as OCR for text extraction or cropping for targeted inspection, to iteratively distill the complex scene into a novel Structured Intermediate Representation (SIR) (See Fig. 3) to preserve as much valuable information as possible across modalities. Then, based on the input question, the Translator automatically selects appropriate tools and dynamically adjusts its execution steps, ultimately gen-

erating the SIR in various forms tailored to the problem-solving process. Crucially, the process is not unidirectional; the reasoning agent can provide feedback to the Translator, requesting clarifications that prompt further tool use to refine the SIR. This multi-round interaction creates a targeted information flow that extracts the precise visual evidence needed to arrive at a confident answer.

Through comprehensive experiments on knowledge-intensive VQA benchmarks including MMMU-Pro, MMMU, OCR-BenchV2, and MIA-Bench, we demonstrate that our agent-based, modular system (*e.g.*, a 3B VLM translator + 8B LLM reasoner) reduces inference costs and surpasses much larger, monolithic end-to-end VLMs on most reported benchmarks.

Our core contributions are as follows:

- We propose **SeeingEye**, a novel, **plug-and-play framework** that unlocks the multimodal reasoning capabilities of powerful, pre-existing text-only LLMs without requiring any modification to their architecture.
- We introduce the **Structured Intermediate Representation (SIR)**, a targeted information channel that delivers precise visual evidence to the text-only reasoner.
- We design a novel **Agentic Information Flow**, where a translator agent autonomously selects tools based on the VQA task. This multi-round interaction generates and refines the SIR, improving both effectiveness and cost-efficiency.

Overall, our results highlight a scalable pathway to advanced multimodal reasoning, liberating strong text-only LLMs to fully leverage their powerful reasoning capabilities on visual data.

2 Related Work

The Evolving Landscape of Visual Question Answering. The VQA landscape has rapidly evolved from simple recognition towards multifaceted reasoning. While benchmarks like GQA (Hudson and Manning, 2019) introduced critical challenges in compositional and spatial reasoning, a recent wave of datasets tests more specialized, expert-level capabilities. These include reasoning with college-level knowledge (MMMU (Yue et al., 2024a)), across multilingual contexts (M3Exam (Zhang et al., 2023)), through complex, layered

instructions (MIA-Bench (Qian et al., 2024)), and within specific domains like chart analysis (EncQA (Mukherjee et al., 2025)). This task diversity exposes the limitations of monolithic models that use a fixed visual encoding strategy (Agrawal et al., 2023; Ke et al., 2025). Such static approaches create an information bottleneck, failing to distill the precise visual details required for each unique challenge and thus motivating our adaptive, agent-based translation framework.

Structured Representations for Multimodal Reasoning. A critical bottleneck in existing multimodal systems is the conversion of rich visual scenes into coarse textual summaries, such as generic captions or unordered OCR text (Alayrac et al., 2022; Li et al., 2023), which discards vital semantic and spatial relationships. Insights from the text-only domain have shown that structured inputs, including JSON schemas, key-value pairs, and knowledge graphs (Cheng et al., 2024; Sun et al., 2023), significantly enhance an LLM’s reasoning capabilities by making relationships explicit and reducing ambiguity. Our work operationalizes this principle for the visual domain. We propose the Structured Intermediate Representation (SIR) as a rich, deliberate communication channel that bridges the gap between visual perception and high-fidelity reasoning.

Agentic Frameworks and Visual Chain-of-Thought. LLM-based agents have demonstrated powerful abilities in planning, tool use, and interactive reasoning (Yao et al., 2023; Shinn et al., 2023). This agentic paradigm, often augmented by Chain-of-Thought (CoT) prompting (Wei et al., 2022) to improve reasoning transparency, has been extended to the multimodal domain. In Visual CoT, models generate step-by-step textual rationales to ground their reasoning in visual evidence (Zhang and Zhang, 2023; Rose et al., 2023). Our work advances this concept from generating linear, unstructured rationales to a more sophisticated, agent-driven process. The translator agent in our *SeeingEye* framework engages in a multi-round *Agentic Information Flow*, dynamically constructing and refining a *structured* representation (our SIR) through a feedback loop with the reasoner, enabling a more targeted and adaptive problem-solving strategy.

3 Method

Our proposed framework, **SeeingEye**, unlocks the multimodal reasoning capabilities of text-only

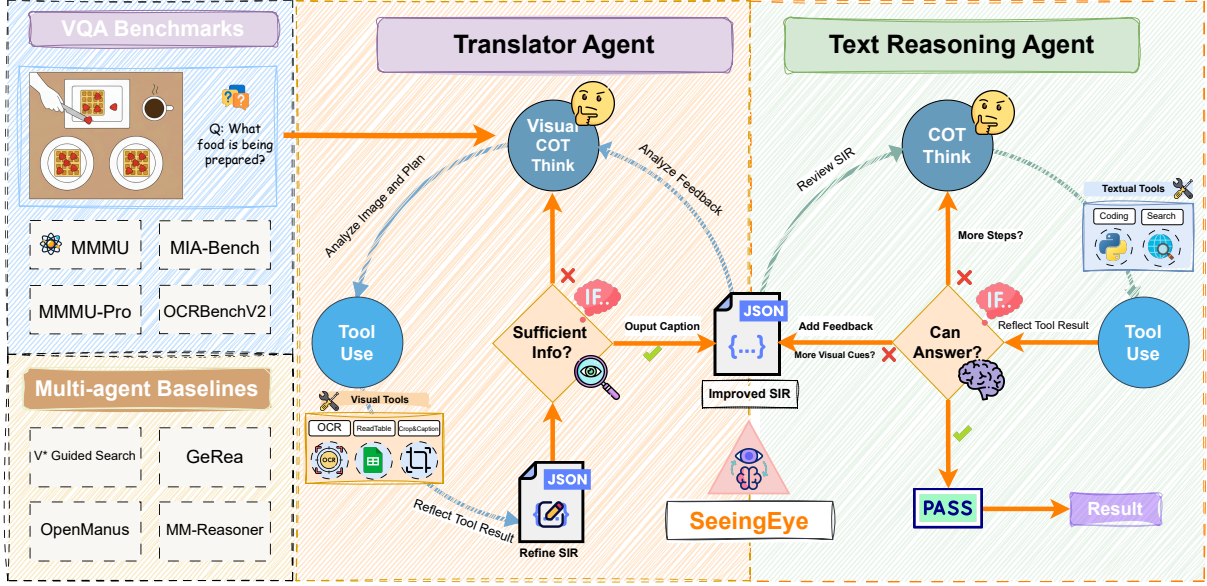


Figure 1: The Agentic Information Flow of our SeeingEye framework. The process begins with the **Translator Agent** (left), which takes the previous state’s SIR and external feedback to perform a Visual Chain-of-Thought (VCoT) analysis. It uses tools to gather new visual evidence, reflects on the results, and iteratively refines the SIR. If the information is deemed sufficient (**PASS**), the improved SIR is passed to the **Text Reasoning Agent** (right). The Reasoner performs its own CoT-driven analysis and tool use. Based on its confidence, it either produces a final answer (**PASS**) or generates targeted feedback (**FAIL**), initiating a new outer loop iteration for the Translator to gather more specific visual cues.

LLMs by introducing a decoupled, two-agent system. This system comprises a **Translator Agent** (\mathcal{A}_T) and a **Reasoning Agent** (\mathcal{A}_R). The core of our method lies in a novel **Agentic Information Flow**, orchestrated through a nested loop structure, where the agents collaborate by iteratively improving a central **Structured Intermediate Representation (SIR)** (see details in Fig. 3 and Sec. A.1). Figure 1 provides a high-level illustration, while Algorithm 1 presents the formal specification of this interactive process.

3.1 The Translator Agent: Grounded Visual Analysis

The Translator Agent, \mathcal{A}_T , is a lightweight VLM responsible for converting raw pixel data into a rich, structured, and query-relevant format. Its state at the start of outer-loop iteration i is defined by the input image I , the question Q , the SIR from the previous iteration S_{i-1} , and feedback from the Reasoner F_{i-1} . The agent’s goal is to produce an improved SIR, S_i , through a multi-step inner loop.

Visual Chain-of-Thought (VCoT) Analysis. At each inner step j , the agent generates a *Visual Chain-of-Thought* (VCoT) (Chen et al., 2024), a textual thought process $c_T^{(j)}$ describing its direct

visual observations and its reasoning for the next action.

$$c_T^{(j)} = \text{VCoT}(I, Q, S_{i-1}, F_{i-1}, h_T^{(j-1)}) \quad (1)$$

where $h_T^{(j-1)}$ is the history of actions within the current inner loop.

Adaptive Tool Selection and Execution. Guided by its VCoT, \mathcal{A}_T selects a tool $o_T^{(j)}$ from its toolset \mathcal{O}_T via its policy π_T , and executes it to yield a result $r_T^{(j)}$.

$$o_T^{(j)} \sim \pi_T(c_T^{(j)}, I, Q, S_{i-1}, F_{i-1}, h_T^{(j-1)}) \quad (2)$$

Among the tools in \mathcal{O}_T , **SmartGridCaption** is a specialized sub-routine designed for complex spatial queries that require targeted analysis. As illustrated in our case study (Figure 2), the tool is invoked when a direct visual observation proves insufficient. Initially, the agent generates a global SIR describing a “church building” but cannot identify the “animal in the poster” from this coarse view (Step 1).

To resolve this ambiguity, the tool first discretizes the image into a 4×4 grid. It then leverages a vision-LLM to interpret the query and select the most informative patches, in this instance, the rectangular region [9, 9] containing the poster (Step 2).

Algorithm 1 SeeingEye: Agentic Information Flow

```

1: Input: question  $Q$ , options  $O$ , image  $I$ 
2: Parameters: MAX_ITERS, MAX_STEPS
3: Initialize:  $S_0 \leftarrow \text{null}$ ,  $F_0 \leftarrow \text{null}$ 
4: for  $i = 1 \rightarrow \text{MAX\_ITERS}$  do
5:      $\triangleright$  Translator Agent inner loop
6:      $S_{\text{current}} \leftarrow S_{i-1}$ 
7:     for  $j = 1 \rightarrow \text{MAX\_STEPS}$  do
8:          $a_T \leftarrow$ 
           TranslatorPolicy(VCoT( $I, Q, S_{\text{current}}, F_{i-1}$ ))
9:         if  $a_T$  is ToolCall( $o_T$ , args) then
10:             $r_T \leftarrow \text{ExecuteTool}(o_T, \text{args})$ 
11:             $S_{\text{current}} \leftarrow \text{RefineSIR}(S_{\text{current}}, r_T)$ 
12:         else if  $a_T$  is TerminateSIR then
13:             break
14:          $S_i \leftarrow S_{\text{current}}$ 
15:          $\triangleright$  Reasoning Agent inner loop
16:         for  $k = 1 \rightarrow \text{MAX\_STEPS}$  do
17:              $a_R \leftarrow$ 
               ReasonerPolicy(CoT( $S_i, Q, O$ ))
18:             if  $a_R$  is ToolCall( $o_R$ , args) then
19:                  $r_R \leftarrow \text{ExecuteTool}(o_R, \text{args})$ 
20:             else if  $a_R$  is TerminateAnswer( $A$ )
21:                 return  $A$ 
22:             else if  $a_R$  is TerminateFeedback( $F$ )
23:                 then
24:                      $F_i \leftarrow F$ 
25:                 break
26:         return ForceAnswer( $S_{\text{MAX\_ITERS}}, Q, O$ )

```

232 A detailed caption is then generated for this specific
233 crop, yielding the critical observation: “Poster fea-
234 turing a person holding a dove” (Step 3). Crucially,
235 this new, fine-grained detail is not treated in isola-
236 tion. As shown in the “Integration Process”, it is
237 strategically integrated with the previous global de-
238 scription to create an updated, more comprehensive
239 SIR. The SIR thus evolves from a generic overview
240 to a targeted representation containing the precise
241 fact needed for the query. This procedure of tar-
242 geted refinement effectively transforms a vague
243 spatial query into a high-confidence textual fact,
244 enabling the Reasoning Agent to deduce the final
245 answer with ease (Step 4).

246 **Iterative SIR Refinement and Termination.** A
247 core feature of our framework is the iterative refine-
248 ment of the SIR within the Translator’s inner loop.
249 After each tool use, the agent reflects on its VCoT

and the tool result to update the SIR. Let $S^{(j-1)}$ be
the SIR at the beginning of the step; the refinement
process is:

$$S^{(j)} = \text{RefineSIR}(S^{(j-1)}, c_s^{(j)}, r_T^{(j)}) \quad (3)$$

Following this refinement, the agent quanti-
tatively assesses the completeness of the up-
dated SIR, producing a confidence score $c_s^{(j)} =$
AssessSufficiency($S^{(j)}$). This score is compared
against a predetermined sufficiency threshold,
 τ_T . The inner loop terminates by invoking the
TerminateAndOutputSIR tool if this confidence
exceeds the threshold ($c_s^{(j)} \geq \tau_T$) or if the maxi-
mum step limit N_T is reached. Upon termination,
the agent outputs the final SIR for the outer loop,
 $S_i = S^{(j)}$, along with a categorical confidence
level $C_T \in \{\text{low, mid, high}\}$ derived from the fi-
nal score. If confidence is below the threshold and
steps remain, the agent continues its inner loop to
gather more visual information.

3.2 The Reasoning Agent: High-Level Cognition and Decision-Making

The Reasoning Agent, \mathcal{A}_R , is a powerful text-only
LLM whose state for iteration i consists of the
SIR S_i , the question Q , and a short-term memory
 $M_{R,i-1}$ summarizing its prior actions. It leverages
the SIR for high-level reasoning and to decide on a
terminal action.

SIR-Grounded Analysis and Tool Use. The
Reasoner initiates its own inner loop, generating
a chain-of-thought $c_R^{(k)}$ and using its policy π_R to
select a tool $o_R^{(k)}$ from its distinct toolset \mathcal{O}_R (e.g.,
CodeInterpreter, Search).

$$c_R^{(k)} = \text{CoT}(S_i, Q, M_{R,i-1}, h_R^{(k-1)}) \quad (4)$$

$$o_R^{(k)} \sim \pi_R(c_R^{(k)}, S_i, Q, M_{R,i-1}, h_R^{(k-1)}) \quad (5)$$

The tool result $r_R^{(k)}$ is appended to its inner-loop
history $h_R^{(k)}$.

Terminal Decision Policy. The Reasoning
Agent’s decision-making is also governed by a
confidence-threshold mechanism. After each inner-
loop step k (which includes its own CoT and op-
tional tool use), the agent assesses its ability to
answer the question based on its current reason-
ing history $h_R^{(k)}$, yielding a confidence score $c_a^{(k)}$.
This score is compared against a high-confidence
answering threshold, τ_R .

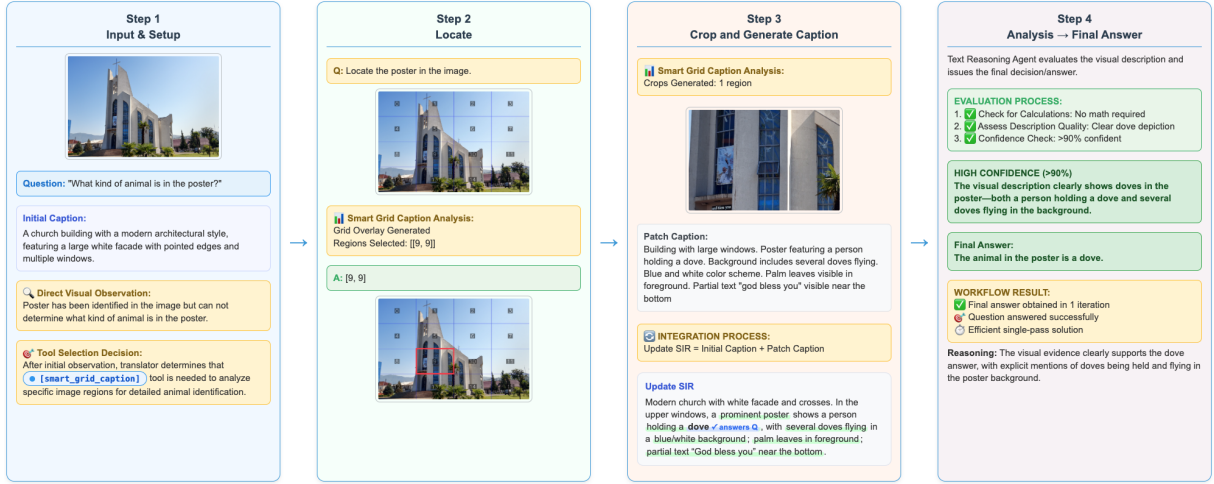


Figure 2: A detailed case study of the **SmartGridCaption** tool. (1) An initial global SIR fails to identify the animal in the poster. (2) The tool grids the image and locates the relevant patch [9, 9]. (3) A fine-grained patch caption is generated and integrated, updating the SIR with the crucial detail of a “dove”. (4) This refined SIR enables the Reasoning Agent to provide a high-confidence final answer in a single iteration.

- **If confidence is high** ($c_a^{(k)} \geq \tau_R$) or if the final outer iteration ($i = \text{MAX_ITERS}$) is reached, the agent is compelled to execute the `TerminateAndAnswer` action. It generates the final answer A , and the process terminates.
- **If confidence is low** ($c_a^{(k)} < \tau_R$), the agent’s decision policy π_D makes an autonomous choice. It can either continue its inner reasoning loop (if $k < N_R$) to further analyze the SIR or use more textual tools, or it can execute the `TerminateAndAskTranslator` action. This choice is formalized as:

$$a_{\text{final}} \sim \pi_D(S_i, Q, h_R^{(k)}) \quad (6)$$

where $a_{\text{final}} \in \{\text{ContinueReasoning}, \text{TerminateAndAskTranslator}\}$. Choosing the latter synthesizes a feedback query F_i specifying the missing visual information, which is passed back to the Translator Agent to initiate a new outer loop iteration.

4 Experiments

We conduct a series of experiments to evaluate the effectiveness of our SeeingEye framework. Our evaluation is designed to answer a central research question: How does our proposed Translator-based Agentic Information Flow, designed to unlock the reasoning capabilities of text-only LLMs, compare in terms of performance and efficiency against state-of-the-art monolithic VLMs and other advanced agent-based approaches?

4.1 Experimental Settings

Benchmarks. We evaluate our framework on a suite of challenging, reasoning-centric Visual Question Answering (VQA) benchmarks that require deep understanding of visual details, text, and domain-specific knowledge.

- **MMMU** (Yue et al., 2024a): A massive, multi-discipline multimodal benchmark featuring questions from college-level exams across six core disciplines, requiring expert-level knowledge and reasoning. We report on the validation set.
- **MMMU-Pro** (Yue et al., 2024b): A more challenging successor to MMMU, curated by human experts to feature more complex reasoning chains and reduce annotation artifacts. We evaluate on both the Standard and Vision subsets.
- **OCR-BenchV2** (Fu et al., 2024): A comprehensive benchmark for evaluating OCR capabilities in the wild, testing the model’s ability to read and interpret text from diverse and complex scenes.
- **MIA-Bench** (Qian et al., 2024): A Multimodal Instruction-following and Analysis benchmark designed to assess a model’s ability to follow complex instructions that require comparing, calculating, and reasoning over multiple image regions.

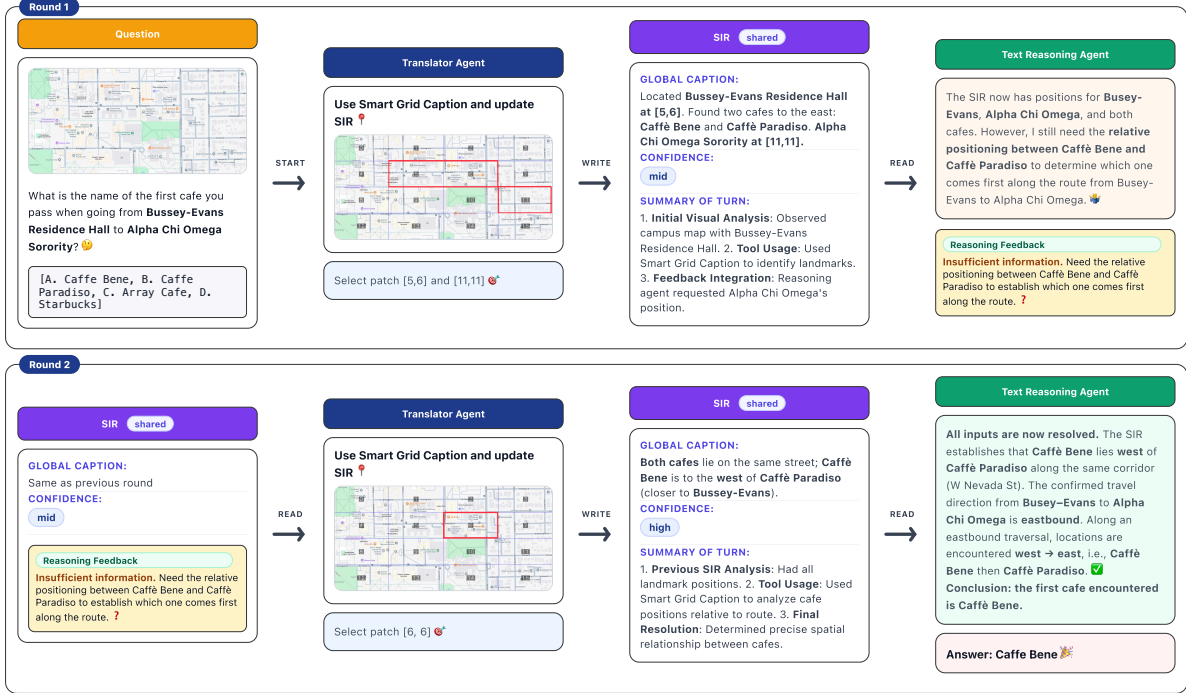


Figure 3: A case study of the **Structured Intermediate Representation (SIR)**. The Translator Agent converts visual evidence into a structured textual state that preserves task-relevant scene details, enabling the Reasoning Agent to solve the question through text-only reasoning.

Baselines. We compare our method against leading end-to-end VLMs and an existing agentic multimodal method:

- **End-to-End VLMs:** These are monolithic, state-of-the-art models that process image and text inputs in a unified architecture. We include **LLaVA-1.5 (7B)** (Liu et al., 2024), a widely-used open-source VLM; several variants of **Qwen2.5-VL** (Bai et al., 2025), a powerful series of VLMs; and **GPT-4o-mini**, a highly capable multimodal model from OpenAI (OpenAI, 2024). These models represent the dominant paradigm and serve as a direct point of comparison for overall performance.
- **Agentic Multimodal Method:** We include **OpenManus** (Liang et al., 2025) with GPT-4o-mini as a standalone comparison method for agentic multimodal reasoning.

Implementation Details Our model, referred to as **SeeingEye**, is instantiated using a 3B parameter Vision Language Model (Qwen2.5-VL) as the Translator Agent (\mathcal{A}_T) and an 8B parameter text-only Large Language Model (Qwen3) as the Text Reasoning Agent (\mathcal{A}_R). For our experiments, the inner loops for both agents (N_T, N_R) and the

outer feedback loop (N_{outer}) are each capped at a maximum of 3 iterations to ensure computational tractability.

4.2 Main Results

The main results of our experiments are presented in Table 1. Our SeeingEye framework, instantiated with a compact 3B VLM translator and an 8B text-only LLM reasoner, demonstrates strong performance across a suite of challenging benchmarks.

Performance on Reasoning-Centric Benchmarks. Our results reveal a clear trend: on complex, knowledge-intensive reasoning benchmarks such as MMMU and MMMU-Pro, our modular framework substantially outperforms larger monolithic VLMs. It also achieves the best score on MIA-Bench, while GPT-4o-mini remains strongest on OCR-BenchV2. This outcome suggests that the reasoning capabilities of powerful text-only LLMs are a distinct and valuable asset that is not fully leveraged in monolithic designs. By decoupling perception from reasoning, our framework allows the text-only agent to operate in its native domain, processing rich, structured textual information rather than latent visual features. This architec-

Table 1: Performance comparison on various knowledge-intensive multimodal benchmarks. Our method, SeeingEye, uses a compact 3B VLM translator with an 8B text-only LLM reasoner. Scores are reported using each benchmark’s official metric.

Benchmark	SeeingEye	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen2.5-VL-32B	LLaVA-1.5-7B	GPT-4o-mini
MMMU-Pro Vision (200)	45.00	24.61	23.26	28.77	12.13	31.72
MMMU-Pro Standard (10 option, 200)	47.00	25.82	23.60	32.93	14.21	38.99
MMMU (500)	63.00	48.33	51.11	51.56	32.47	59.40
OCR-BenchV2 (score, 200)	52.50	52.00	53.96	52.22	19.21	55.08
MIA-Bench (score, 200)	90.73	76.90	79.90	89.60	69.80	88.58

tural choice proves to be a more parameter-efficient and effective pathway to unlocking advanced multimodal reasoning.

Effectiveness of the Agentic Information Flow.

The mixed OCR-BenchV2 result also clarifies the role of our framework: SeeingEye is not simply a stronger OCR engine, but a structured perception-to-reasoning interface. Its largest gains appear when visual evidence must be selected, organized, and reasoned over rather than merely transcribed. In contrast to passing unstructured captions or conversational history, our **Agentic Information Flow** is mediated by the **Structured Intermediate Representation (SIR)**. The SIR is a stateful, query-conditioned data object: the Translator populates it with structured visual evidence, and the Reasoner can issue precise feedback to refine the representation. This transforms the interaction from a stateless exchange of text blobs into a high-fidelity collaborative process.

5 Discussion

5.1 Plug-and-Play Reasoning Agents

To validate the plug-and-play nature of our framework, we conduct an ablation study by fixing the Translator Agent while varying the text-only Reasoning Agent (\mathcal{A}_R). As shown in Table 2, our framework is model-agnostic, successfully integrating with various open-source and proprietary reasoners.

Crucially, the overall system performance scales with the reasoning capability of the text-only model, improving from 52.67% with Qwen3-8B to 54.67% with the larger Qwen3-14B on the MMMU_{dev} set. This result supports our core hypothesis: the SeeingEye architecture effectively isolates and leverages the reasoning power of the text-only agent, making the choice of reasoner an important driver of the final performance.

Table 2: Performance on the MMMU_{dev} set when varying the text-only Reasoning Agent. The SeeingEye is kept fixed. Results show that system performance scales with the reasoner’s capability.

Benchmark	Qwen3-8B	Qwen3-14B	GPT-4o-mini (text-only)
MMMU _{dev} (%)	52.67	54.67	54.29

5.2 Cost-Efficiency and Scalability

A primary motivation for our decoupled, agent-based design is to create a scalable and cost-effective pathway to multimodal reasoning. Monolithic VLMs incur substantial computational costs by processing high-dimensional images through a massive, unified architecture. Our framework strategically mitigates this expense through an efficient division of labor: we use a small, low-cost VLM (our 3B Translator Agent) to perform the pixel-to-concept translation. This agent distills the query-relevant visual information into a compact, text-based Structured Intermediate Representation (SIR). Consequently, the powerful but expensive text-only LLM (our 8B Reasoning Agent) only processes this low-dimensional and inexpensive textual SIR, never the raw image.

This architectural efficiency is validated in Table 3. We compare the cost-performance trade-off against a strong Qwen2.5-VL-32B baseline on the MMMU_{val} set. To provide a strong repeated-sampling comparison, we report the Qwen2.5-VL-32B’s *best* accuracy over three runs and account for the corresponding generation cost. Our method achieves a superior accuracy of 63.00% at a comparable, and in the median case, lower, total cost. This demonstrates that by intelligently dividing labor and minimizing interactions with the most expensive reasoning components, our framework provides a more favorable and scalable trade-off between inference cost and performance.

We use the following formula to calculate the cost. Let $p_{in}^{(m)}$ and $p_{out}^{(m)}$ be the official Qwen prices (USD) per 1000 input/output tokens for model $m \in$

Table 3: Inference cost and accuracy on the MMMU_{val} set (per question). For a fair comparison, the Qwen2.5-VL-32B reported is the *best* of three runs. Prices are in USD.

Method	Accuracy (%)	Input \$	Output \$	Total \$
Qwen2.5-VL-32B (best-of-3)	60.67	0.003198	0.009072	0.01227
Ours (avg)	63.00	0.0090	0.0026	0.0116
Ours (median)	63.00	0.0076	0.0019	0.0101

{tr, rs, vlm}, where tr is the Translator (Qwen2.5-VL-3B), rs is the Reasoner (e.g., Qwen3-8B/14B), and vlm is the monolithic baseline. Let $T_{in,i}^{(m)}$ and $T_{out,i}^{(m)}$ be the input/output tokens used by model m at outer-loop iteration $i = 1, \dots, N$.

$$C_{\text{ours}} = \sum_{i=1}^N \left(\frac{T_{in,i}^{(\text{tr})}}{1000} p_{in}^{(\text{tr})} + \frac{T_{out,i}^{(\text{tr})}}{1000} p_{out}^{(\text{tr})} + \frac{T_{in,i}^{(\text{rs})}}{1000} p_{in}^{(\text{rs})} + \frac{T_{out,i}^{(\text{rs})}}{1000} p_{out}^{(\text{rs})} \right) \quad (7)$$

5.3 The Efficacy of Multi-Round Interaction

A cornerstone of our SeeingEye framework is the multi-round interaction that allows the Reasoning Agent to provide feedback and the Translator Agent to iteratively refine the SIR. To quantify the impact of this mechanism, we conducted an ablation study on the maximum number of outer loop iterations on the challenging MMMU-Pro (Vision) benchmark.

As demonstrated in Table 4, the benefits of our iterative refinement process are substantial and consistent. In the single-iteration setting (Max Iterations = 1), where the Reasoner cannot provide feedback, the system achieves a baseline accuracy of 34.21%. This is analogous to static, one-shot captioning methods. Enabling a single round of feedback (Max Iterations = 2), where the Translator can act upon the Reasoner’s request for more specific visual information, yields a significant performance gain to 36.84%.

Most notably, allowing for up to three full iterations elevates the performance to **45.00%**, an absolute improvement of over 10% compared to the single-shot approach. This steep performance curve provides strong empirical evidence for our central hypothesis: complex multimodal reasoning is not a monolithic perception task, but an iterative

Table 4: Performance on MMMU-Pro (Vision) when varying the maximum number of outer loop iterations. The results clearly demonstrate the significant benefit of multi-round SIR refinement.

Benchmark	Max Outer Iterations		
	1	2	3
MMMU-Pro (Vision) (%)	34.21	36.84	45.00

process of inquiry and refinement. The ability for the agents to repeatedly pass and modify the SIR allows the system to progressively drill down into the most critical visual details, discard initial ambiguities, and ultimately converge on a high-fidelity representation of the scene that is precisely tailored to the reasoning needs of the query.

6 Conclusion

In this work, we addressed the challenge of unlocking multimodal reasoning in powerful, pre-existing text-only LLMs. We introduced **SeeingEye**, a novel framework that decouples perception from reasoning through a collaborative, two-agent system. Our core innovation is the **Agentic Information Flow**, where a lightweight Translator Agent iteratively generates and refines a **Structured Intermediate Representation (SIR)** to provide targeted, high-fidelity visual evidence to a text-only Reasoning Agent. Comprehensive experiments demonstrate that our modular, plug-and-play approach improves cost-efficiency and outperforms strong monolithic VLMs on several complex reasoning benchmarks while remaining competitive elsewhere. Our findings suggest that advanced multimodal AI can benefit from structured collaboration between specialized agents rather than relying solely on scaling monolithic end-to-end models.

7 Limitations

Our framework depends on the quality of the Translator Agent and its available visual tools. If the translator fails to detect a small but decisive visual detail, or if OCR and crop-based inspection return noisy outputs, the SIR may omit evidence that the Reasoning Agent cannot recover from text alone. Although multi-round feedback mitigates this issue, it also introduces additional latency compared with single-pass VLM inference. The SIR is intentionally compact and structured, which improves reasoning efficiency but may discard low-level visual

546	cues that are difficult to verbalize. Finally, our empirical study focuses on knowledge-intensive VQA benchmarks and uses model pricing available at the time of evaluation; broader deployment settings, additional multimodal tasks, and changing inference prices may affect the observed cost-performance trade-off.		
547			
548			
549			
550			
551			
552			
553	References		
554	Aishwarya Agrawal, Ivana Kajić, Emanuele Bugliarello, Elnaz Davoodi, Anita Gergely, Phil Blunsom, and Aida Nematzadeh. 2023. Reassessing evaluation practices in visual question answering: A case study on out-of-distribution generalization . In <i>Findings of the Association for Computational Linguistics: EACL 2023</i> , pages 1201–1226, Dubrovnik, Croatia. Association for Computational Linguistics.		
555			
556			
557			
558			
559			
560			
561			
562	Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. <i>Advances in neural information processing systems</i> , 35:23716–23736.		
563			
564			
565			
566			
567			
568			
569	Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-vl technical report . <i>arXiv preprint arXiv:2502.13923</i> .		
570			
571			
572			
573	Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. <i>Advances in neural information processing systems</i> , 33:1877–1901.		
574			
575			
576			
577			
578			
579	Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. 2024. Visual chain-of-thought prompting for knowledge-based visual reasoning . In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> , volume 38, pages 1254–1262.		
580			
581			
582			
583			
584			
585	Kewei Cheng, Nesreen K Ahmed, Theodore Willke, and Yizhou Sun. 2024. Structure guided prompt: Instructing large language model in multi-step reasoning by exploring graph structure of the text. <i>arXiv preprint arXiv:2402.13415</i> .		
586			
587			
588			
589			
590	Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, and 1 others. 2024. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. <i>arXiv preprint arXiv:2501.00321</i> .		
591			
592			
593			
594			
595			
596	Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in		
597			
598			
599			
		llms via reinforcement learning. <i>arXiv preprint arXiv:2501.12948</i> .	600
			601
	Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In <i>Proceedings of the IEEE/CVF conference on computer vision and pattern recognition</i> , pages 6700–6709.		602
			603
			604
			605
			606
	Fucaï Ke, Joy Hsu, Zhixi Cai, Zixian Ma, Xin Zheng, Xindi Wu, Sukai Huang, Weiqing Wang, Pari Delir Haghighi, Gholamreza Haffari, and 1 others. 2025. Explain before you answer: A survey on compositional visual reasoning. <i>arXiv preprint arXiv:2508.17298</i> .		607
			608
			609
			610
			611
			612
	Mahmoud Khademi, Ziyi Yang, Felipe Frujeri, and Chenguang Zhu. 2023. Mm-reasoner: A multimodal knowledge-aware framework for knowledge-based visual question answering. In <i>Findings of the Association for Computational Linguistics: EMNLP 2023</i> , pages 6571–6581.		613
			614
			615
			616
			617
			618
	Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In <i>International conference on machine learning</i> , pages 19730–19742. PMLR.		619
			620
			621
			622
			623
	Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, Xiao Tang, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025. Openmanus: An open-source framework for building general ai agents .		624
			625
			626
			627
	Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In <i>Proceedings of the IEEE/CVF conference on computer vision and pattern recognition</i> , pages 26296–26306.		628
			629
			630
			631
			632
	Ziyu Ma, Shutao Li, Bin Sun, Jianfei Cai, Zuxiang Long, and Fuyan Ma. 2024. Gereqa: Question-aware prompt captions for knowledge-based visual question answering. <i>arXiv preprint arXiv:2402.02503</i> .		633
			634
			635
			636
	Kushin Mukherjee, Donghao Ren, Dominik Moritz, and Yannick Assogba. 2025. Encqa: Benchmarking vision-language models on visual encodings for charts. <i>arXiv preprint arXiv:2508.04650</i> .		637
			638
			639
			640
	OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence .		641
			642
	Övgü Özdemir and Erdem Akagündüz. 2024. Enhancing visual question answering through question-driven image captions as prompts . In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops</i> , pages 1562–1571.		643
			644
			645
			646
			647
			648
	Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. 2024. Mia-bench: Towards better instruction following evaluation of multimodal llms. <i>arXiv preprint arXiv:2407.01509</i> .		649
			650
			651
			652
			653

654	Daniel Rose, Vaishnavi Himakunthala, Andy Ouyang,	Zhuosheng Zhang and Aston Zhang. 2023. You only	711
655	Ryan He, Alex Mei, Yujie Lu, Michael Saxon, Chin-	look at screens: Multimodal chain-of-action agents.	712
656	may Sonar, Diba Mirza, and William Yang Wang.	<i>arXiv preprint arXiv:2309.11436</i> .	713
657	2023. Visual chain of thought: bridging logical		
658	gaps with multimodal infillings. <i>arXiv preprint</i>	A Appendix	714
659	<i>arXiv:2305.02317</i> .		
660	Noah Shinn, Federico Cassano, Ashwin Gopinath,	A.1 SIR Structure and Prompts	715
661	Karthik Narasimhan, and Shunyu Yao. 2023. Re-	The Structured Intermediate Representation (SIR)	716
662	flexion: Language agents with verbal reinforcement	is the central data object that facilitates communi-	717
663	learning. <i>Advances in Neural Information Process-</i>	cation between the Translator Agent and the Rea-	718
664	<i>ing Systems</i> , 36:8634–8652.	soning Agent. Its schema is defined as follows, and	719
665	Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo	the subsequent prompts detail how it is managed.	720
666	Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-		
667	Yeung Shum, and Jian Guo. 2023. Think-on-	A.2 Tool Prompts	721
668	graph: Deep and responsible reasoning of large lan-	These are the critical termination tools used by the	722
669	guage model on knowledge graph. <i>arXiv preprint</i>	Reasoning Agent to conclude its inner loop, either	723
670	<i>arXiv:2307.07697</i> .	by providing a final answer or by requesting more	724
671	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	visual information from the Translator Agent.	725
672	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,		
673	and 1 others. 2022. Chain-of-thought prompting elic-	A.3 Agent Prompts	726
674	its reasoning in large language models. <i>Advances</i>		
675	<i>in neural information processing systems</i> , 35:24824–	B Generative AI Statement	727
676	24837.		
677	Chenfei Wu, Shengming Yin, Weizhen Qi, Xi-	This work utilized generative AI tools to assist with	728
678	aodong Wang, Zecheng Tang, and Nan Duan.	formatting, generating LaTeX templates, and refin-	729
679	2023. Visual chatgpt: Talking, drawing and edit-	ing word choice. The authors reviewed and verified	730
680	ing with visual foundation models. <i>arXiv preprint</i>	all AI-assisted content to ensure factual accuracy	731
681	<i>arXiv:2303.04671</i> .	and academic integrity.	732
682	Penghao Wu and Saining Xie. 2024. V*: Guided visual		
683	search as a core mechanism in multimodal llms. In		
684	<i>Proceedings of the IEEE/CVF Conference on Com-</i>		
685	<i>puter Vision and Pattern Recognition</i> , pages 13084–		
686	13094.		
687	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak		
688	Shafraan, Karthik Narasimhan, and Yuan Cao. 2023.		
689	React: Synergizing reasoning and acting in language		
690	models. In <i>International Conference on Learning</i>		
691	<i>Representations (ICLR)</i> .		
692	Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng,		
693	Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang,		
694	Weiming Ren, Yuxuan Sun, and 1 others. 2024a.		
695	Mmmu: A massive multi-discipline multimodal un-		
696	derstanding and reasoning benchmark for expert agi.		
697	In <i>Proceedings of the IEEE/CVF Conference on Com-</i>		
698	<i>puter Vision and Pattern Recognition</i> , pages 9556–		
699	9567.		
700	Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang,		
701	Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao		
702	Yu, Ge Zhang, Huan Sun, and 1 others. 2024b.		
703	Mmmu-pro: A more robust multi-discipline mul-		
704	timodal understanding benchmark. <i>arXiv preprint</i>		
705	<i>arXiv:2409.02813</i> .		
706	Wenxuan Zhang, Mahani Aljunied, Chang Gao,		
707	Yew Ken Chia, and Lidong Bing. 2023. M3exam: A		
708	multilingual, multimodal, multilevel benchmark for		
709	examining large language models. <i>Advances in Neu-</i>		
710	<i>ral Information Processing Systems</i> , 36:5484–5505.		

SIR JSON Structure

Schema Overview: The SIR schema defines the fields for visual description, confidence, and feedback. **Schema Content:**

```
{
  "global_caption": {
    "type": "string",
    "description": "A comprehensive description of ALL visual elements in sentence form or table form, including: text content, numerical values, table structures, objects, layouts, colors, spatial relationships, and any other visual information. Be factual and descriptive - do not infer anything not exists in the original image.",
  },
  "confidence": {
    "type": "string",
    "enum": ["low", "mid", "high"],
    "description": "Your confidence level in the completeness and accuracy of this global caption. 'low' = incomplete analysis or unclear image, 'mid' = good analysis with some limitations, 'high' = comprehensive and thorough analysis.",
  },
  "feedback": {
    "type": "string",
    "description": "Specific feedback about what additional visual information you need from the translator. Be precise about what's missing or unclear in the current description.",
  }
}
```

Figure 4: **SIR JSON Structure.** The `global_caption` and `confidence` fields are provided by the Translator Agent (via the `TerminateAndOutputCaption` tool). The `feedback` field is provided by the Text Reasoning Agent (via the `TERMINATE_AND_ASK_TRANSLATOR` tool).

SIR Management Prompt

Prompt Overview: This prompt is part of the Translator's instructions, detailing how to iteratively manage and format the SIR. **Prompt Content:**

SIR MANAGEMENT:

- Maintain a continuously evolving SIR throughout your analysis
- After each tool use or observation, update your SIR with new information
- Your SIR should be comprehensive and capture ALL visual elements discovered
- Always state your current SIR after each step

SIR OUTPUT FORMAT: Structure your evolving SIR using clear sections:

```
{
  "global_caption": {
    "type": "string",
    "description": "A comprehensive description of ALL visual elements in sentence form or table form, including: text content, numerical values, table structures, objects, layouts, colors, spatial relationships, and any other visual information. Be factual and descriptive - do not infer anything not exists in the original image.",
  },
  "confidence": {
    "type": "string",
    "enum": ["low", "mid", "high"],
    "description": "Your confidence level in the completeness and accuracy of this global caption. 'low' = incomplete analysis or unclear image, 'mid' = good analysis with some limitations, 'high' = comprehensive and thorough analysis.",
  },
}
```

Figure 5: **Translator Agent SIR Management Prompt.** This defines how the Translator iteratively builds the SIR and the output format it must adhere to.

Feedback Input Prompt

Prompt Overview: This prompt is dynamically given to the Translator Agent when it starts a new outer-loop iteration based on feedback from the Reasoning Agent. **Prompt Content:**

Your current SIR with reasoning feedback (iteration {iteration-1}):
{self.current_sir}

IMPROVEMENT TASK:

1. Analyze the reasoning feedback carefully to understand what visual details are needed
2. Look at the image again with this feedback in mind
3. UPDATE your current SIR to address the feedback - don't start fresh
4. Focus on visual details that help answer the question: {question}
5. Maintain objectivity - describe what you see, don't infer answers

Remember: Update your existing SIR incrementally, don't recreate it from scratch.

Figure 6: **Translator Agent Feedback Input Prompt.** This guides the Translator to refine the existing SIR in a targeted manner based on the Reasoner's feedback.

TerminateAndAnswer Tool

Prompt Overview (Tool Description): Terminate the reasoning process and provide a final answer when you have sufficient information from the SIR to confidently answer the question.

Tool Prompt:

Use this tool when:

- The SIR contains all necessary visual details to answer the question
- You can identify the correct answer from the available options
- No additional information or refinement is needed from the translator agent
- Your answer matches one of the multiple choice options (if applicable)

IMPORTANT: For multiple choice questions, ensure your answer corresponds to one of the given options (A, B, C, D).

This signals that the iterative feedback loop should end with your final answer.

Tool Raw Code:

```
class TerminateAndAnswer(BaseTool):
    name: str = "terminate_and_answer"
    description: str = _TERMINATE_AND_ANSWER_DESCRIPTION
    parameters: dict = {
        "type": "object",
        "properties": {
            "answer": {
                "type": "string",
                "description": "Your final answer to the question. Please include
                    short answer only. For multiple choice, only
                    include option",
            },
            "confidence": {
                "type": "string",
                "description": "Your confidence level in this answer.",
                "enum": ["high", "medium", "low"],
            },
            "reasoning": {
                "type": "string",
                "description": "Brief explanation of how the SIR information
                    led to this answer.",
            }
        },
        "required": ["answer", "confidence", "reasoning"],
    }
```

Figure 7: **Reasoning Agent TerminateAndAnswer Tool.** This tool allows the Reasoner to conclude the entire process with a definitive answer.

TerminateAndAskTranslator Tool

Prompt Overview (Tool Description): Terminate current reasoning step and request more specific visual observations from the translator.

Tool Prompt:

Use this tool when:

- The current SIR (visual description) is insufficient for answering the question
- You need more specific details about certain parts of the image
- Important visual elements seem to be missing from the description
- You need clarification about spatial relationships, text content, or visual elements
- The translator's description lacks crucial information needed for reasoning

This signals that you need additional visual analysis before you can provide a final answer.

Tool Raw Code:

```
class TerminateAndAskTranslator(BaseTool):
    name: str = "terminate_and_ask_translator"
    description: str = _TERMINATE_AND_ASK_TRANSLATOR_DESCRIPTION
    parameters: dict = {
        "type": "object",
        "properties": {
            "feedback": {
                "type": "string",
                "description": "Specific feedback about what additional visual
                    information you need from the translator.
                    Be precise about what's missing or unclear in
                    the current description.",
            }
        },
        "required": ["feedback"],
    }
```

Figure 8: **Reasoning Agent TerminateAndAskTranslator Tool.** This tool allows the Reasoner to request further visual refinement from the Translator, initiating a new outer loop.

Translator Agent: System Prompt

Prompt Overview: Guides the lightweight VLM to act as a Visual-Only Captioner. Its sole objective is to observe the image, use visual tools for precision, and iteratively build a structured, factual, and neutral description of visual content (the SIR).

Prompt Content:

You are "Visual-Only Captioner to capture input images".

Goal: Output a raw, neutral description of visible content only. Preserve blanks, placeholder marks, unknowns ("?"), typos, casing, punctuation, and line breaks exactly as seen. Do NOT infer, normalize, answer, or explain meaning.

DO:

- Describe only visible elements: text, shapes, colors, axes, legends, labels, numbers, layout, positions, arrows, boxes, tables, panels.
- Extract on-screen text ****verbatim**** (including blanks and "?").
- Note spatial relations ("X above Y", "arrow A->B").
- Mark unknowns/blanks exactly as they appear (e.g., "?", placeholder marks, empty cell).
- Always think step by step first before using a tool. Decide which tool is most appropriate for the current observation step.
- TOKEN LIMIT: Keep your responses concise and within 1024 tokens. Focus on the most essential visual details.

DON'T (hard ban):

- No answers, explanations, conclusions, predictions, calculations, or domain knowledge.
- Don't replace blanks/"?" with guesses. Don't add units or meanings.

Available tools:

- OCR: Extract text with high precision, useful for image that contains text
- read_table: Parse structured tabular data, useful for spreadsheets, data tables
- smart_grid_caption: Used to analyze specific image regions

SIR OUTPUT FORMAT:

```
{
  "global_caption": "A comprehensive description of ALL visual elements",
  "confidence": "low/mid/high"
}
```

Figure 9: **Translator Agent System Prompt** enforces strict visual-only captioning behavior and defines the SIR output format.

Translator Agent: First Step Prompt

Prompt Overview: Initializes the translator agent's first observation step, establishing the empty SIR and guiding initial visual analysis.

Prompt Content:

You are "Visual-Only Captioner to capture input images".

INITIAL TASK:

1. ****Direct Visual Observation****: Look at the image and identify the main visual elements
2. ****Create Initial SIR****: Start building your SIR with overall structure, layout, and prominent elements

CURRENT SIR STATUS: Empty - you are starting fresh

SIR MANAGEMENT:

- Maintain a continuously evolving SIR throughout your analysis
- After each tool use or observation, update your SIR with new information
- Your SIR should be comprehensive and capture ALL visual elements discovered
- Always state your current SIR after each step

Figure 10: **Translator Agent First Step Prompt** initiates the visual analysis process and establishes SIR management protocol.

Translator Agent: Next Step Prompt

Prompt Overview: Guides iterative refinement of the SIR based on current state and previous observations.

Prompt Content:

Based on the current state and previous memory, what's your next action?. Goal: Output a raw, neutral description of visible content only. Preserve blanks, placeholder marks, unknowns ("?"), typos, casing, punctuation, and line breaks exactly as seen. Do NOT infer, normalize, answer, or explain meaning.

Remember, you can directly observe the image content yourself without tools. So, if you haven't, start with direct visual observation of the image content. Then use tools to get detailed, accurate information.

Available tools (use to enhance visual observation):

- OCR: Extract text with high precision, useful for image that contains text
- read_table: Parse structured tabular data, useful for spreadsheets, data tables
- smart_grid_caption: Used to analyze specific image regions

If you think you have comprehensive visual details, you should use terminate_and_output_caption tool with your stored_sir containing your complete objective visual description. This tool will format your caption as proper JSON.

Figure 11: **Translator Agent Next Step Prompt** guides the iterative SIR refinement process.

Reasoning Agent: System Prompt

Prompt Overview: Guides the text-only LLM to act as a question answering expert, analyzing the SIR from the translator and determining whether to answer or request more visual information.

Prompt Content:

You are a question answering expert. You receive (1) a text caption of image from translator and (2) a question relevant to the image. Analyze the information and provide clear reasoning to answer the question. ALWAYS provide your reasoning and thoughts BEFORE using tools. Explain what you're trying to accomplish and why.

Your capabilities:

- Analyze textual descriptions of various scenarios (visual scenes, documents, data, etc.)
- Provide detailed explanations and clear reasoning when helpful
- Indicate when information is insufficient or ambiguous in the text description
- Keep responses under 1024 tokens - be concise and focus on key reasoning points.

Available tools:

- `python_execute`: Use for calculations, data analysis, mathematical operations, or any computation. ALWAYS include `print()` statements to show results.
- `terminate_and_answer`: Use ONLY when you have HIGH CONFIDENCE in your answer and it matches one of the available options (for multiple choice questions)
- `terminate_and_ask_translator`: Use when you need MORE SPECIFIC visual information to make an accurate decision

DECISION CRITERIA - BE CONSERVATIVE:

- Use `python_execute` when math/data processing clarifies the answer.
- Use `terminate_and_answer` only if text gives specific distinguishing details and confidence ≥ 0.9 , and (for MCQ) your answer matches an option.
- Otherwise use `terminate_and_ask_translator` and state exactly which visual labels/regions/relations you need, when visual cues are ambiguous or insufficient.

Figure 12: **Reasoning Agent System Prompt** defines the agent's role as an expert reasoner with conservative decision criteria.

Reasoning Agent: Next Step Prompt

Prompt Overview: Guides intermediate reasoning steps, emphasizing confidence assessment and computational verification.

Prompt Content:

Analyze the provided visual description and determine if you have SUFFICIENT SPECIFIC DETAILS to answer with HIGH CONFIDENCE.
ALWAYS provide your reasoning and thoughts BEFORE taking any action.

Consider these key questions:

- Does the problem require calculations, data analysis, or computational verification?
- Does the visual description provide specific, distinguishing details?
- Can you clearly differentiate between all options based on the description?
- Are you >90% confident in your answer AND does it match an available option (for multiple choice)?

****COMPUTATION NEEDED**** - USE `python_execute` FIRST:

- When math/data processing clarifies the answer.
- Need to verify calculations or process numerical information
- ****ALWAYS**** include `print()` statements to show your work and results

****HIGH CONFIDENCE (>90%)**** - USE `terminate_and_answer`:

- You can clearly rule out incorrect options
- ****ESPECIALLY****: After performing calculations with `python_execute` that confirm your answer
- ****MANDATORY****: Your answer matches one of the multiple choice options (A, B, C, D) if applicable
- ****IMPORTANT****: If your calculated answer doesn't match any option, use `python_execute` again to recalculate with different approach/units/interpretation
- Provide your confident answer with reasoning

****NEED MORE DETAILS**** - USE `terminate_and_ask_translator`:

- Description is too general or vague
- Missing specific visual details needed to distinguish between options
- Uncertain which option is correct
- Request SPECIFIC visual information you need (exact labels, shapes, spatial relationships, etc.)

Keep responses under 1024 tokens - be concise and focus on key reasoning points.

Figure 13: **Reasoning Agent Next Step Prompt** provides structured decision criteria for tool selection.

Translator Agent: Final Step Prompt

Prompt Overview: Forces final SIR output when maximum translation steps are reached.

Prompt Content:

****FINAL OUTPUT****

You have reached the maximum number of steps. You must now provide your final visual description using `terminate_and_output_caption` tool.

FINAL ROUND STRATEGY:

1. ****Synthesize all observations**** from your previous tool usage and direct observation
2. ****No hallucination/inference**** Output raw, neutral description of visible content. Preserve blanks, placeholder marks, unknowns ("?"), typos, casing, punctuation, and line breaks exactly as seen. Do NOT infer, normalize, answer, or explain meaning.
3. ****MANDATORY: Use `terminate_and_output_caption`**** - you cannot use other tools at this point

Figure 14: **Translator Agent Final Step Prompt** enforces termination and final SIR generation.