

# SlideAgent: Hierarchical Agentic Framework for Multi-Page Visual Document Understanding

Yiqiao Jin<sup>1\*</sup>, Rachneet Kaur<sup>2</sup>, Zhen Zeng<sup>2</sup>, Sumitra Ganesh<sup>2</sup>, and Srijan Kumar<sup>1</sup>

<sup>1</sup>Georgia Institute of Technology, <sup>2</sup>J.P. Morgan AI Research

{yjin328,srijan}@gatech.edu, zhen.zeng@jpmchase.com,

{rachneet.kaur,sumitra.ganesh}@jpmorgan.com

<https://SlideAgent.github.io/>

## Abstract

Multi-page visual documents such as manuals, brochures, presentations, and posters convey key information through layout, colors, icons, and cross-slide references. While large language models (LLMs) offer opportunities in document understanding, current systems struggle with complex, multi-page visual documents, particularly in fine-grained reasoning over elements and pages. We introduce SlideAgent, a versatile agentic framework for understanding *multi-modal*, *multi-page*, and *multi-layout* documents, especially slide decks. SlideAgent employs specialized agents and decomposes reasoning into three specialized levels—global, page, and element—to construct a structured, *query-agnostic* representation that captures both overarching themes and detailed visual or textual cues. During inference, SlideAgent selectively activates specialized agents for multi-level reasoning and integrates their outputs into coherent, context-aware answers. Extensive experiments show that SlideAgent achieves significant improvement over both proprietary (+7.9) and open-source models (+9.8).

## 1 Introduction

Visual documents—from earnings reports and academic lectures to business strategy presentations—are ubiquitous, conveying ideas not only from text, but also from the intricate interplay of layout, icons, visual hierarchy, and cross-page relationships. These documents are central to high-stakes domains such as finance, science, and technology. Accurately interpreting them thus remains a pressing challenge.

**Challenges.** Accurately interpreting these *multi-page*, *multi-modal* artifacts remains challenging. Recent advances in multimodal large language models (MLLMs) have accelerated document understanding (Verma et al., 2024), yet three gaps

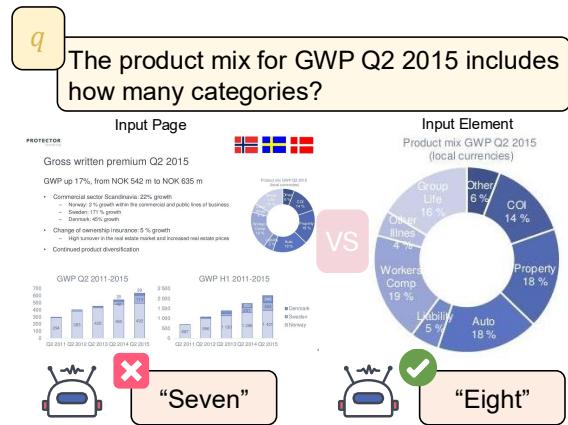


Figure 1: When given the full page, the LLM miscounts the number of product mix categories. After isolating the chart, it correctly identifies all eight categories, highlighting the importance of accurate element parsing.

remain: **1) Scalable Fine-Grained Reasoning.**

State-of-the-art MLLMs process a limited number of images at once (Liu et al., 2023) and tend to treat each page holistically, missing fine-grained, element-level cues required for user queries (Faysse et al., 2024; Tanaka et al., 2025). In Figure 1, GPT-4o miscounts chart segments on a cluttered page of a slide deck, but correctly identifies all segments once the relevant chart is cropped—highlighting the importance of element-level parsing for latent reasoning abilities. **2) Domain-Specific Visual Semantics.**

Most MLLMs are pre-trained on natural images (Wu et al., 2024), lacking exposure to domain-specific diagrams, financial charts, or scientific plots. Consequently, they struggle with the specialized language of visual documents (Cho et al., 2024). For example, logos appear on every page, reinforcing brand identities but do not offer additional content. Color schemes encode categorical information (red for losses and green for gains in financial reports). Icons convey abstract concepts (lightbulbs for innovation, arrows for causal relationships); and spatial positioning signals importance (centered elements typically

\*Work done as an intern at J.P. Morgan AI Research.

matter more than corner annotations). falter in spatial reasoning (Wang et al., 2024b,a), failing to locate visual elements (Sharma and Vats, 2025; Bhattacharyya et al., 2025; Polak and Morgan, 2025). Low-resolution visual encoders in MLLMs (Liu et al., 2023) further miss details such as footnotes or superscripts. **3) Metadata-Free Integration.** Many systems (Singer-Vine, 2025; huridocs, 2025; Rausch et al., 2021) rely on clean metadata—figure locations, hierarchy tags, embedded text layers—that can be unavailable or corrupted in real-world PDF. Users may take screenshots of PDF documents, scan copies of physical documents, export slides or documents as flattened PDFs, upload or share PDFs generated from software that strips out or does not preserve document structure. Recent metadata-free methods (Yu et al., 2025; Tanaka et al., 2025) address these by parsing only visual images without relying on the metadata, despite a performance gap.

**This Work.** We present SlideAgent, an LLM-based agentic framework for **fine-grained understanding of multi-page, varying-size visual documents**. Inspired by the human information processing model (Lang, 2000; Naysmith et al., 2021), SlideAgent employs a hierarchical architecture with specialized agents at three levels: *global* (document-wide topics), *page* (page-specific features and cross-page relations), and *element* (fine-grained components such as charts, figures, and text blocks). During *knowledge construction* stage, SlideAgent parses layout and generates *query-agnostic* knowledge at each level. At *inference* stage, SlideAgent retrieves *query-specific* knowledge, enabling scalable fine-grained reasoning over relevant pages and elements. We benchmark SlideAgent and baseline models, demonstrating significant performance on both open-source and proprietary models. SlideAgent surpasses its LLM counterpart by 7.9 (proprietary) and 9.8 (open-source), respectively. We also show that SlideAgent enhances spatial reasoning, visual counting, and cross-element understanding, with results that are highly interpretable.

## 2 Method

**Problem Formulation** Given a multi-page visual document  $\mathcal{P} = \{p_1, \dots, p_{|\mathcal{P}|}\}$  with  $|\mathcal{P}|$  pages and a query  $q$ , the goal is to generate a natural language answer  $a = f(q, \mathcal{P})$  by reasoning over relevant visual and textual elements.

**Overview** As shown in Figure 2, SlideAgent operates in two stages: 1) *Knowledge Construction*: Build a hierarchical, *query-agnostic* knowledge base  $\mathcal{K} = \{\mathcal{K}_g, \mathcal{K}_p, \mathcal{K}_e\}$  capturing global, page, and element knowledge; 2) *Retrieval and Question-Answering*: Using multi-level retrieval to retrieve *query-specific* content from  $\mathcal{K}$  and synthesize the answer  $a$ , ensuring both broad contextual understanding and fine-grained reasoning.

### 2.1 Knowledge Construction Stage

Given a multi-page document, SlideAgent constructs hierarchical knowledge at three levels using specialized agents in a top-down manner.

**Global Agent** The global agent  $\mathcal{M}_g$  generates document-level knowledge  $\mathcal{K}_g = \mathcal{M}_g(\mathcal{P})$ , capturing the overall summary, objectives, and narrative flow of the document. This layer establishes overarching themes to support high-level reasoning about the document’s purpose. Since visual documents are often large and LLMs have a limited capacity for processing visuals, SlideAgent samples the first three pages to generate  $\mathcal{K}_g$  (example in Appendix Figure 7).

**Page Agent** For each page  $p_i \in \mathcal{P}$ , the page agent generates page-level knowledge  $\mathcal{K}_p$  in a sequential manner, conditioned on the page’s visual content  $v_i$ , the global knowledge  $\mathcal{K}_g$ , and the knowledge from the preceding page  $\mathcal{K}_p^{i-1}$ :

$$\mathcal{K}_p^i = \mathcal{M}_p(v_i, \mathcal{K}_g, \mathcal{K}_p^{i-1}), i \in [1, |\mathcal{P}|]. \quad \mathcal{K}_p^0 = \emptyset \quad (1)$$

The complete page-level knowledge  $\mathcal{K}_p = \bigcup_{i=1}^{|\mathcal{P}|} \mathcal{K}_p^i$  provides an intermediate representation that captures page-specific content while linking them to the global context (sample in Figure 8). To ensure comprehensive understanding of all pages,  $\mathcal{K}_p$  is subsequently used to refine  $\mathcal{K}_g$ .

**Element Agent** LLMs often struggle with spatial reasoning over visual documents (Wang et al., 2024b,a), failing to locate visual elements (Sharma and Vats, 2025; Bhattacharyya et al., 2025; Polak and Morgan, 2025). To address this, our element agent integrates external tools to explicitly capture the spatial and structural information of each page.

At the finest granularity, the element agent decomposes each page  $p_i$  into a set of elements using a layout parsing pipeline  $f : v_i \rightarrow \{(i, e_j, b_j, t_j)\}_{j=1}^{M_i}$ , where each element is represented by its page index  $i$ , verbatim text  $e_j$ , bounding box coordinates  $b_j$ , and element type  $t_j$ .  $f$

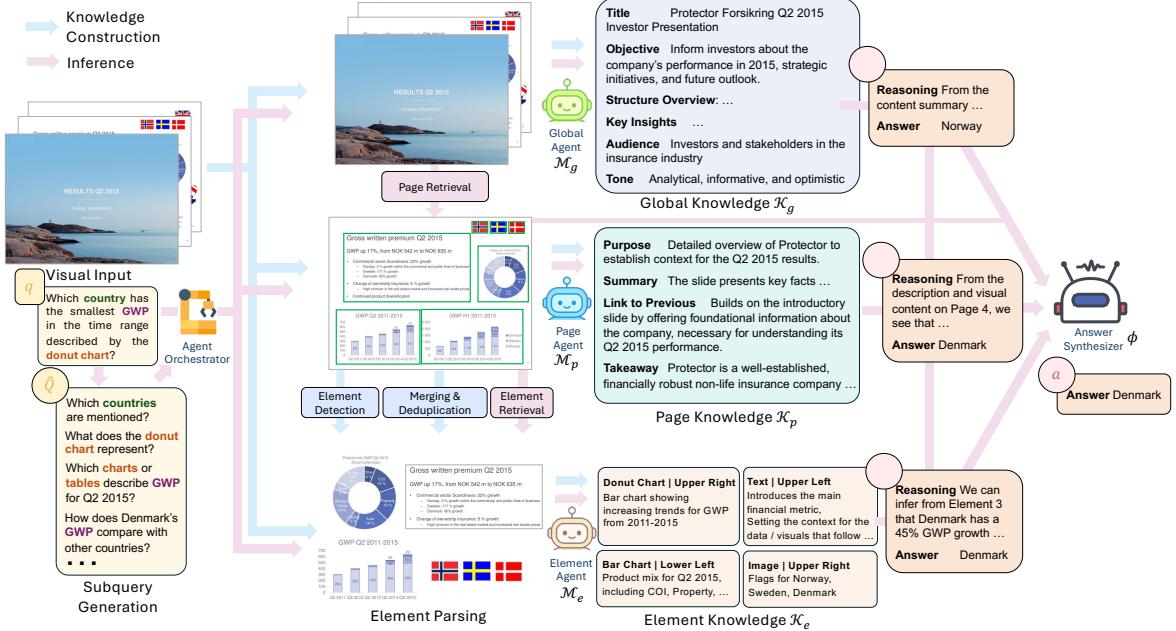


Figure 2: SlideAgent generates knowledge about input slide decks in a hierarchical manner at 3 levels: *global*, *page*, and *element*. At each level, specialized agents generate query-agnostic knowledge during *knowledge construction*, then retrieve and reason over query-specific textual & visual knowledge during *inference* stage. Sample knowledge  $\mathcal{K}$  generated by SlideAgent is in Appendix Figure 7,8,9 and answers generated by the agents are in Figure 5.

integrates text detection, layout detection, and element classification, followed by post-processing to merge fragmented elements (Appendix A.1).

For each detected element  $e_j$ , the agent consumes the annotated visuals and metadata to generate element-level knowledge:

$$\mathcal{K}_e^j = \mathcal{M}_e(i, v_i, e_j, b_j, t_j, \mathcal{K}_g, \mathcal{K}_p^i). \quad (2)$$

$\mathcal{K}_e^j$  consists of the element’s semantic role, functional purpose, and its relation to the slide page (example in Figure 9). This design allows SlideAgent to reason consistently across diverse visual content while preserving spatial relationships for document understanding.

## 2.2 Inference Stage

**Query Classification** Different queries require different perspectives to answer effectively. For instance, queries about global understanding, like “What is the overall theme of this presentation?” requires a broad overview from a macroscopic perspective and activates only the global agent. In contrast, a fact-based query, like “What is the revenue on slide 3?” requires detailed, slide-specific information and requires both the page and element agents. Leveraging too many agents may increase computation or introduce noise. Thus, the agent orchestrator first attempts to classify each query  $q$  into one of 4 predefined categories, such as global understanding, fact-based direct queries, multi-hop

reasoning, and layout & visual relationships. Each type corresponds to a question-answering strategy and a targeted set of agents (Appendix Table 7). For instance, global understanding queries activate only the global agent, as they require a broad overview, while fact-based queries trigger both the page and element agents to retrieve detailed, slide-specific information. If none of the predefined categories apply, the agent defaults to the “unknown” category, which activates all agents.

**Subquery Generation and retrieval** The original query  $q$  is usually short and can lead to noisy retrieval. Using  $q$ , SlideAgent generates subqueries  $\hat{Q}$  targeting key entities in the query. For example, for the query ‘Which country has the smallest GWP in the time range described by the donut chart,’ the model generates subqueries related to keywords such as *country*, *donut chart*, and *GWP*.  $q$  and  $\hat{Q}$  are concatenated to jointly retrieve the top- $k_\ell$  pages  $\hat{\mathcal{P}}$  and the top elements  $\hat{\mathcal{E}}$  along with their page / element knowledge  $\mathcal{K}_p$  and  $\mathcal{K}_e$ . Note that the retriever can include simple sparse retrievers such as BM25 (Robertson et al., 2004), dense retriever such as SFR (Meng et al., 2024), GTE (Li et al., 2023) and Linq (Kim et al., 2024), and multimodal retriever such as COLPALI (Faysse et al., 2024) and VisRAG (Yu et al., 2025).

**Answer Generation and Synthesis** The system guides structured reasoning through hierarchical

context understanding to generate  $h_g, h_p, h_e$ , which contain both the agent’s answer and its reasoning. The global context is processed to generate  $h_g = f_g(\mathcal{K}_g, \mathcal{P}_s, q)$ , which captures the overall document-level context. The page-level agent then derives from the retrieved pages and corresponding knowledge:  $h_p = f_p(\mathcal{K}_p, \mathcal{R}_p(\mathcal{P}, \{q\} \cup \hat{\mathcal{Q}}), h_g)$ . At the element-level, the visual input is annotated with bounding boxes  $\{b_i\}$ , which are then processed by the element agent to generate  $h_e = f_e(\mathcal{R}'_e(\mathcal{E}, \{q\} \cup \hat{\mathcal{Q}}), \text{Annot}(\hat{\mathcal{V}}))$ , capturing the detailed visual and textual cues.

If all agents agree according to answer matching (Appendix B.2), or only one agent is activated, the answer is taken directly from the activated agent. Otherwise, the answer synthesizer  $\phi(\cdot)$  combines the reasoning from all agents and visuals from the retrieved pages to generate the final answer:

$$a = \phi(h_g, h_p, h_e, \{v_i : p_i \in \mathcal{R}(\hat{\mathcal{P}})\}). \quad (3)$$

### 3 Experiments

**Datasets** We evaluate SlideAgent and baselines on two tasks: 1) *multi-page understanding*, using datasets such as SlideVQA (Tanaka et al., 2023), TechSlides, and FinSlides (Wasserman et al., 2025); and 2) single-page understanding, using InfoVQA (Mathew et al., 2022). Details of the datasets are in Appendix C.1.

**Models.** We benchmark SlideAgent against 3 types of baselines: 1) **Multimodal LLMs**: 15 LLMs from 8 model families, including proprietary models (GPT-4o (OpenAI, 2025), Gemini (Anil et al., 2023), Claude (Anthropic, 2025)) and open-source models (Llama-3.2 (Grattafiori et al., 2024), InternVL3 (Chen et al., 2024), Phi-3 (Abdin et al., 2024), Qwen2.5-VL (Bai et al., 2025)); 2) **Multimodal RAG Methods**: VisRAG (Yu et al., 2025), VDocRAG (Tanaka et al., 2025), and COLPALI (Faysse et al., 2024); 3) **Multi-agent Systems**: ViDoRAG (Wang et al., 2025a). We evaluate SlideAgent using two backbone LLMs, including both proprietary models (GPT-4o (OpenAI, 2025)) and open-source models (InternVL3-8B (Chen et al., 2024)). These models are chosen for their widespread use in state-of-the-art QA systems (Yu et al., 2025; Jin et al., 2025; Cho et al., 2024). The parameter sizes, knowledge cutoff dates, and release dates are in Appendix Table 8. We use the text-based retriever SFR (Meng et al., 2024) due to its strong efficiency-performance (Cheng et al., 2024; Jin et al., 2025). For models restricted to

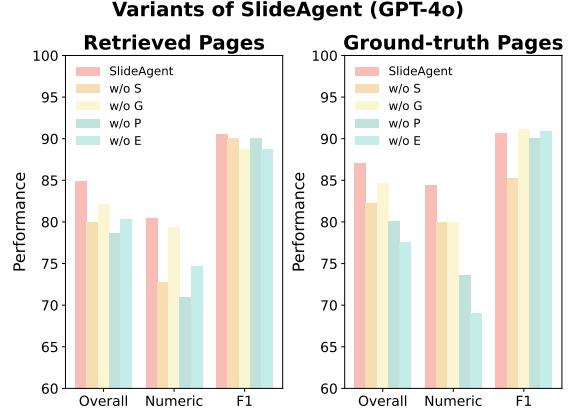


Figure 3: Performance comparison among variants of SlideAgent with base model GPT-4o.

single-image input (e.g. LLaVA-v1.5 (Liu et al., 2023)), we concatenate the top 3 retrieved images into one following previous work (Yu et al., 2025). In settings where ground-truth pages are available, we provide them as input to all models. Otherwise, each model receives as many retrieved pages up to its input capacity.

**Metrics.** We evaluate the performance of SlideAgent in end-to-end question answering. For questions asking about *numeric* values, we extract, standardize, and compare the prediction and the ground-truth in various formats, including percentages, decimals, integers, and word-based representations (e.g., “three”, “thousand”, “million”). Numbers are normalized to a unified format (e.g., ‘17k’ → ‘17000’, ‘2.5 million’ → ‘2500000’, ‘97%’ → ‘0.97’). Otherwise, we use F1-score to evaluate the *lexical overlap* between predicted and ground-truth answers. Both answers are normalized, tokenized, and preprocessed by removing stopwords and punctuation before metric calculation. For ranking, we use MRR, Hit@k, and nDCG@k (Appendix C.2).

**Settings** We evaluate SlideAgent under two realistic settings: 1) **End-to-End Performance**. The model must first retrieve relevant pages before answering the query, reflecting real workflows where users query long visual documents –analysts navigating 100-page earnings presentation, students reviewing lecture slides, or engineers inspecting multi-page technical specifications. This setting measures the end-to-end capability in retrieval, spatial reasoning, and layout understanding. 2) **Performance with Ground-truth Pages**. The model is directly given the page(s) containing the answer, isolating reasoning from retrieval. This mirrors cases where context is known—e.g., a product man-

Model	SlideVQA			TechSlides			FinSlides		
	Overall	Num	F1	Overall	Num	F1	Overall	Num	F1
<i>Multimodal LLMs (Type 1)</i>									
Gemini 2.0	75.0	71.3	79.8	50.4	67.6	41.6	70.8	70.6	77.8
Gemini 2.5	<u>83.8</u>	<u>78.3</u>	<b>91.8</b>	51.1	71.4	41.2	76.2	75.8	<b>100.0</b>
Gemini 2.5-lite	71.2	60.8	87.0	47.3	58.1	41.9	57.0	56.6	68.3
Claude 4.1	78.4	74.3	82.3	61.0	<u>81.4</u>	52.3	56.5	54.8	73.3
Claude 3.5	62.5	68.3	54.6	52.5	80.2	39.5	48.5	49.5	29.6
GPT-4o	77.0	72.1	84.0	63.4	78.3	53.9	80.0	80.8	62.1
<i>Multimodal RAG (Type 2) and Agenetic Methods (Type 3)</i>									
COLPALI	78.8	73.7	83.4	64.1	73.2	54.5	80.9	81.5	62.7
VisRAG	78.2	73.1	85.4	64.7	72.6	54.7	79.2	81.1	75.8
VDocRAG	80.0	75.0	87.8	67.0	80.5	57.0	<u>83.5</u>	<u>83.8</u>	64.2
ViDoRAG	81.1	76.4	88.1	<u>68.7</u>	78.2	<u>59.4</u>	82.2	83.3	65.1
SlideAgent	<b>84.9</b>	<b>80.4</b>	<u>90.5</u>	<b>70.9</b>	<b>82.5</b>	<b>66.2</b>	<b>85.5</b>	<b>85.9</b>	<u>79.6</u>
Impr.	+7.9	+8.3	+6.5	+7.5	+4.2	+12.3	+5.5	+5.0	+17.5

Table 1: Performance comparison of proprietary models on SlideVQA, TechSlides, and FinSlides. All baseline methods use GPT-4o for answer generation. Improvements over the base model GPT-4o is shown in green. The best and second best performance are highlighted in bold and underlined. SlideAgent outperforms all baseline methods (Type 2/3) sharing the same base model (GPT-4o), and even outperforms stronger raw LLMs (e.g. Gemini-2.5).

ager reviewing a linked design slide—and primarily focuses on reasoning and layout comprehension.

Model	Overall	Num	F1
<i>Raw LLMs</i>			
Gemini 2.0	72.3	66.1	81.3
Gemini 2.5	<b>86.9</b>	<b>81.1</b>	<b>95.4</b>
Gemini 2.5 lite	71.7	62.6	87.0
Claude 4.1	36.3	35.4	38.9
Claude 3.5	31.5	30.8	37.0
GPT-4o	69.0	59.3	90.5
<i>Multimodal RAG and Agenetic Methods</i>			
ViDoRAG	71.2	60.5	90.7
SlideAgent	<u>79.6</u>	<u>69.9</u>	<u>94.1</u>
Impr.	+10.6	+10.5	+3.6

Table 2: Performance comparison among Gemini, Claude, GPT-4o, ViDoRAG, and SlideAgent models.

### 3.1 End-to-End Performance

Tables 1/2/3 and Appendix Table 5 demonstrate the end-to-end performance of SlideAgent across proprietary and open-source models.

**Consistent Improvements across Architectures**  
For both proprietary and open-source models, SlideAgent consistently outperforms all baseline methods (Type 2/3) and the base models across all metrics. For proprietary models (Table 1),

SlideAgent improves overall accuracy by +7.9, numeric reasoning by +8.3, and lexical overlap by +6.5 on SlideVQA. Similar trends are observed for TechSlides (+7.5 overall; +12.3 F1) and FinSlides (+5.5 overall; +17.5 F1), indicating robust performance across diverse domains. For open-source models, the advantage remains pronounced: SlideAgent outperforms InternVL3-8B by +9.8 overall (+11.7 numeric), showing that our hierarchical, multi-agent design generalizes well across LLMs.

### Comparison with Multimodal LLMs

SlideAgent consistently achieves the best or second-best performance among proprietary models (Tables 1 and 11). Notably, although the base GPT-4o model slightly lags behind stronger LLMs such as Gemini 2.5 in raw capability (e.g. 83.8 for Gemini 2.5 vs. 77.0 for GPT-4o in Table 1), SlideAgent’s structured reasoning pipeline fills this gap (84.9 overall for SlideAgent). For open-source models, SlideAgent achieves better performance across all models except for the Qwen2.5 family. While strong base models such as Gemini-2.5 and Qwen2.5 already exhibit advanced multimodal comprehension, SlideAgent is model-agnostic in nature and can be directly applied to these models to further enhance performance.

**Performance across Query Types** Figure 4 presents the performance breakdown across query

Model	SlideVQA			TechSlides			FinSlides		
	Overall	Num	F1	Overall	Num	F1	Overall	Num	F1
<i>Multimodal LLMs (Type 1)</i>									
Llama 3.2 11B	42.9	43.3	42.3	41.4	52.5	36.2	23.3	23.2	26.2
Phi3	72.3	61.8	90.6	59.4	60.0	59.1	48.8	48.5	64.3
Qwen2.5 7B	<b>79.5</b>	<u>70.5</u>	<b>94.3</b>	59.3	52.5	<b>65.9</b>	53.6	52.5	<u>85.7</u>
Qwen2.5 32B	<u>79.2</u>	<b>71.1</b>	<u>92.2</u>	<b>67.5</b>	<b>87.5</b>	60.6	<u>57.4</u>	<u>56.6</u>	<b>87.5</b>
LLaVA 1.5 7B	36.8	22.4	79.0	23.3	12.5	38.7	10.7	10.1	16.6
LLaVA 1.5 13B	44.9	25.1	81.8	28.1	17.5	45.0	20.6	16.5	36.7
LLaVA 1.6 7B	50.9	37.3	82.6	34.4	37.5	32.2	12.2	12.1	17.8
LLaVA 1.6 13B	16.7	10.2	81.5	45.2	40.0	49.1	32.0	31.3	64.3
InternVL3 8B	63.0	56.5	74.1	55.4	57.5	54.4	49.8	49.5	64.3
<i>Multimodal RAG (Type 2) and Agentic Method (Type 3)</i>									
COLPALI	63.4	56.7	73.8	57.1	60.9	55.2	50.4	49.3	65.7
VisRAG	63.6	56.5	75.5	56.8	57.7	55.4	51.1	49.6	65.2
VDocRAG	65.2	59.7	77.0	59.2	60.7	58.3	51.8	50.1	65.9
ViDoRAG	68.8	61.9	77.3	61.4	61.9	59.3	52.7	55.4	66.6
SlideAgent	72.7	68.2	79.4	<u>63.1</u>	<u>78.0</u>	<u>61.7</u>	<b>63.3</b>	<b>62.8</b>	68.3
Impr.	+9.8	+11.7	+5.4	+7.7	+20.5	+2.3	+13.5	+13.3	+4.0

Table 3: Performance comparison of various models on SlideVQA, TechSlides, and FinSlides datasets. SlideAgent outperforms all baseline methods (Type 2/3) sharing the same base model (GPT-4o).

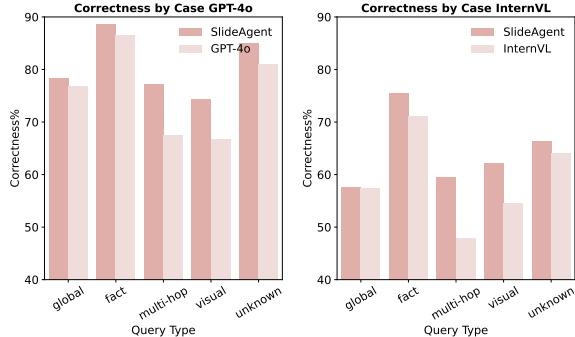


Figure 4: Accuracy of SlideAgent and base models (GPT-4o / InternVL3-8B) on different query types.

types, as defined in Table 7. SlideAgent improves question-answering across diverse query types, especially in multi-hop reasoning and visual/layout questions. The greatest improvement occurs on case 3 (multi-hop reasoning), with a 9.8 point improvement (67.4 to 77.2). This means explicitly guiding the model’s reasoning using generated knowledge ( $\mathcal{K}_g, \mathcal{K}_p, \mathcal{K}_e$ ) significantly improves reasoning capabilities. A notable 7.7-point improvement (66.7 → 74.4) is also observed in visual/layout reasoning, demonstrating the benefit of fine-grained element-level reasoning and retrieval that multimodal LLMs hardly achieve. Both SlideAgent and its counterpart perform well on Case 2 (Fact-based Direct Queries), with only a modest 2.1-point improvement, reflecting the

model’s ability to handle page-level reasoning with little space for further enhancement.

### 3.2 Performance with Ground-truth Pages

When ground-truth pages are provided (Table 11/12), the performance gap narrows. As all models receive the exact pages with the answers, noise introduced in retrieval is eliminated. Under this oracle setting, SlideAgent still improves over GPT-4o (+7.7 overall and +12.5 numeric on SlideVQA), demonstrating the effectiveness of element-level retrieval.

### 3.3 Effectiveness of Knowledge Construction

We evaluate whether hierarchical knowledge representations ( $\mathcal{K}$ ) improve retrieval beyond end-to-end QA. Specifically, we test page-level retrieval using generated subqueries  $\hat{Q}$  and page knowledge  $\mathcal{K}_p$  from SlideAgent. Table 4 shows consistent gains across both text-based retrievers (BM25 (Robertson et al., 2004), BGE (Xiao et al., 2023), SFR (Meng et al., 2024)) and multimodal retrievers (COLPALI (Faysse et al., 2024), VisRAG (Yu et al., 2025), SigLIP2 (Tschanne et al., 2025)).

#### Text-based Retrievers Show Largest Gains.

Structured agent outputs substantially enhance text-based retrievers. SFR achieves the largest gains (+6.4 MRR, +8.5 nDCG@1), showing that page-

Text-based Retrievers	MRR	Recall@1	nDCG@1	Recall@3	Hit@3	nDCG@3
BM25 (Robertson et al., 2004)	59.0	51.5	52.0	63.0	65.3	57.7
w/ SA	63.9 <small>+4.9</small>	54.9 <small>+3.4</small>	56.6 <small>+4.6</small>	67.1 <small>+4.1</small>	68.8 <small>+3.5</small>	62.6 <small>+4.9</small>
BGE (Xiao et al., 2023)	70.1	56.1	60.9	75.8	78.1	69.2
w/ SA	72.3 <small>+2.2</small>	58.4 <small>+2.3</small>	63.2 <small>+2.3</small>	77.4 <small>+1.6</small>	80.3 <small>+2.2</small>	71.3 <small>+2.1</small>
SFR (Meng et al., 2024)	70.1	56.1	60.9	75.8	78.1	69.2
w/ SA	76.5 <small>+6.4</small>	59.9 <small>+3.8</small>	69.4 <small>+8.5</small>	77.3 <small>+1.5</small>	81.8 <small>+3.7</small>	73.2 <small>+4.0</small>
Multimodal Retrievers	MRR	Recall@1	nDCG@1	Recall@3	Hit@3	nDCG@3
SigLIP2 (Tschannen et al., 2025)	26.7	15.9	18.0	31.0	34.0	25.0
w/ SA	28.0 <small>+1.3</small>	16.3 <small>+0.4</small>	18.0 <small>+0.0</small>	32.3 <small>+1.3</small>	35.5 <small>+1.5</small>	26.1 <small>+1.1</small>
COLPALI (Faysse et al., 2024)	82.1	68.2	75.5	78.9	84.0	77.4
w/ SA	82.9 <small>+0.8</small>	70.4 <small>+2.2</small>	76.2 <small>+0.7</small>	88.6 <small>+9.7</small>	90.1 <small>+6.1</small>	83.1 <small>+5.7</small>
VisRAG (Yu et al., 2025)	76.0	63.3	68.6	82.3	84.1	76.0
w/ SA	79.7 <small>+3.7</small>	66.3 <small>+3.0</small>	71.6 <small>+3.0</small>	85.5 <small>+3.2</small>	87.7 <small>+3.6</small>	79.4 <small>+3.4</small>

Table 4: Artifacts generated by SlideAgent, particularly  $\hat{Q}$  and  $\mathcal{M}_p$ , improves *page-level* retrieval performance, especially for text-based retrievers.

level knowledge  $\mathcal{K}_p$  provides richer semantic signals than raw OCR, especially useful in the absence of the vision modality. Even sparse retrievers like BM25 improves (+4.9 MRR), as lexical matching benefits from structured page descriptions, which integrates multimodal cues that pure text extraction often miss. Notably, despite much lower computational costs, text-based retrievers already rival multimodal LLM-based methods, justifying our choice of SFR as the base retriever.

**Multimodal Retrievers Exhibit Smaller but Consistent Gains.** COLPALI (Faysse et al., 2024) improves slightly in ranking quality (+0.8 MRR) but substantially in coverage (+9.7 Recall@3), indicating that structured subqueries help it surface more relevant pages even if they are not ranked at the very top. VisRAG (Yu et al., 2025) achieves a larger ranking boost (+3.7 MRR), suggesting that multimodal LLMs still benefit from textual guidance in aligning queries to page content. SigLIP2 shows minimal gains (+1.3 MRR), likely because it is optimized for natural image domains and transfers less effectively to document-style inputs.

### 3.4 Ablation Studies

We analyze each design choice by removing global agent (w/o G), page agent (w/o P), element agent (w/o E), and subquery generation (w/o S). Figures 3/6 summarize results across proprietary (GPT-4o) and open-source (InternVL3-8B) backbones.

**Page-level Reasoning is Critical** Removing the page agent causes the steepest degradation—GPT-4o

drops –6.3 overall (–9.5 numeric) and InternVL3-8B –8.8 overall. The page agent integrates global themes  $\mathcal{K}_g$  and sequential context  $\mathcal{K}_p^{i-1}$  (Eq. 1), enabling cross-slide coherence and multi-hop reasoning. Its absence breaks this structural bridge between global context and page details. Results for SlideAgent w/o E, w/o G, and w/o S are in Appendix D.1.

### 3.5 Qualitative Answer Analysis

Figure 5 shows a qualitative example of answer generated by SlideAgent to illustrate how SlideAgent leverages hierarchical knowledge for reasoning.

Given the slide deck, SlideAgent builds hierarchical knowledge  $\mathcal{K} = \{\mathcal{K}_g, \mathcal{K}_p, \mathcal{K}_e\}$  (Figures 7–9). Because the query lacks global/page cues (e.g. “on page 4”), the agent orchestrator activates all agents. The global agent performs deck-level triage and nominates the pages directly following page 2’s summary as related to “cause/effect region.” The page agent pinpoints *Page 4: Wealth Management–The Cause*, whose description states that listed problems leads to **Business Under-Performance**. **Element** ( $\mathcal{K}_e$ ) grounds the answer by parsing the flowchart on Page 4, extracting the node whose verbatim text matches the query and following its directed edge to the target node labeled **Business Under-Performance**. The answer synthesizer fuses these signals to return **Business under-performance** with explicit provenance (Page 4, flowchart edge from the queried node). This layering is essential: without global triage, search drifts; without page focus, we may match the phrase but

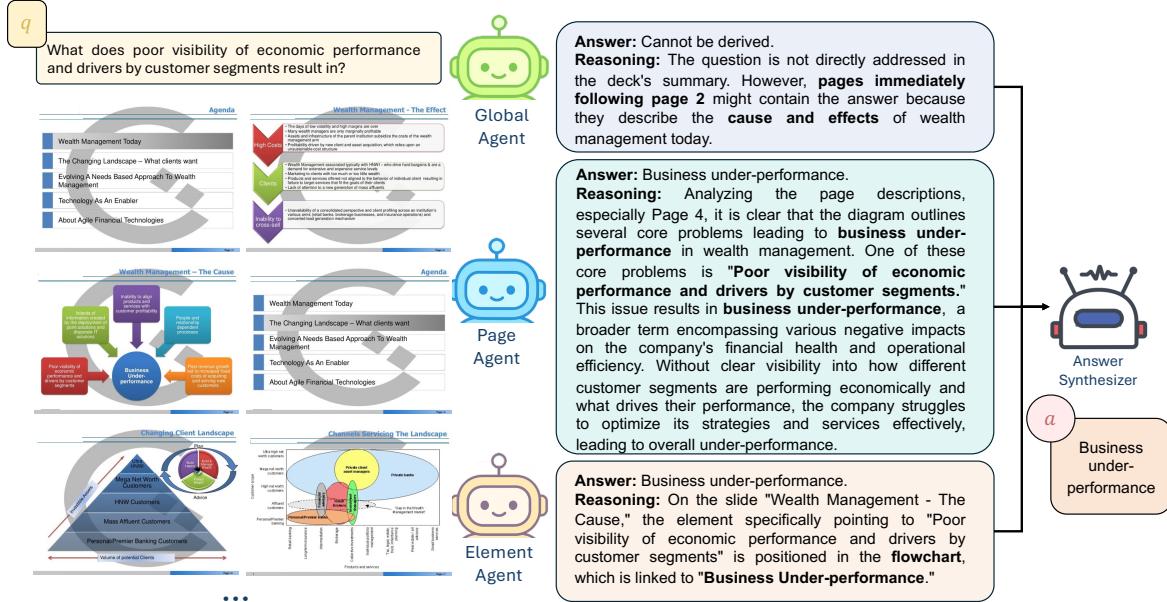


Figure 5: Example answers generated by SlideAgent. Agents at different levels work together to provide comprehensive responses.

miss causality; without element grounding we rely on summaries and risk speculation. Combining all three yields robust reasoning: the global agent provides thematic scope, the page agent narrows candidates, and the element agent confirms answers with visual grounding. This layered verification produces accurate, confident responses.

## 4 Related Work

In this section, we discuss relevant works in visual document understanding and Multimodal LLMs. A more detailed description is in Appendix B.

**Visual Document Understanding** relied on computer vision pipelines combining OCR, layout parsing, and heuristics to extract semantics (Shilman et al., 2005; Bhowmik, 2023; Gao et al., 2019; Kieninger, 1998; Smith, 2007). Though effective, these methods were brittle to noisy and visually rich inputs. Recent models such as BERT-grid (Denk and Reisswig, 2019), LayoutLM (Xu et al., 2020b,a), LayoutT5 (Tanaka et al., 2021), and TILT (Powalski et al., 2021) jointly encode texts, layouts, and visuals through multimodal pretraining—laying the foundation for LLM document understanding. **Visual Question Answering based on Multi-page Document** requires layout comprehension and long-context reasoning. Earlier studies focused on segmentation (Haurilet et al., 2019) and generation (Sun et al., 2021; Fu et al., 2022) perspectives, while recent work focus on multi-hop, numerical, or commonsense reasoning capabilities (Tanaka et al., 2023; Mathew et al., 2022; Ma et al., 2024). Retrieval pipelines such as

ColPali (Faysse et al., 2024), VisRAG (Yu et al., 2025), and VDocRAG (Tanaka et al., 2025) combine retrieval and reasoning for multi-page comprehension, emphasizing the need for fine-grained element-level reasoning.

**General-Purpose Multimodal LLMs** such as GPT-4/4o (Achiam et al., 2023; Hurst et al., 2024), Gemini (Anil et al., 2023), LLaVA (Liu et al., 2023), and InternVL3 (Zhu et al., 2025) have advanced visual reasoning (Shao et al., 2024; Yang et al., 2023). Yet, trained largely on natural images (Wu et al., 2024), these models struggle with visual documents such as slides, which require precise grounding of heterogeneous elements (charts, tables, icons).

## 5 Conclusion

We present SlideAgent, a hierarchical agentic framework that leverages specialized agents at multiple levels for fine-grained infographics understanding, particularly for slide decks. SlideAgent demonstrates significant performance gains and strong generalizability across both proprietary and open-source models.

## 6 Limitations

We did not evaluate element-level retrieval due to the lack of corresponding annotations in existing datasets and the difficulty of defining precise boundaries among elements. Benchmarks in visual document understanding could include such annotations to enable finer-grained analysis.

**Retrieval Method.** Our framework supports both textual and multimodal retrievers. We primarily adopt text-based retrieval for efficiency, though future work could explore multimodal retrievers or domain-specific strategies tailored to slide decks.

## 7 Ethical Considerations

**Data Privacy, Consent, and Intellectual Property.** Visual documents such as those processed by SlideAgent may contain sensitive business or personal data. Ensuring compliance with privacy regulations (e.g., GDPR, CCPA) and obtaining appropriate consent are essential. Organizations should also respect intellectual property rights and establish policies that balance knowledge sharing with fair use.

**Content Reliability and User Responsibility.** Like other LLM-based systems, SlideAgent may inherit biases or produce inaccurate content. While it enhances question answering, outputs should be verified by human judgment, particularly in sensitive or high-stakes scenarios. Clear user guidance and validation practices can help ensure responsible use.

## Disclaimer

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("J.P. Morgan") and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

## References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone, 2024. *arXiv:2404.14219*, 2:6.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*.

Anthropic. 2025. *Claude*.

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-v1 technical report. *arXiv:2502.13923*.

Aniket Bhattacharyya, Anurag Tripathi, Ujjal Das, Archan Karmakar, Amit Pathak, and Maneesh Gupta. 2025. Information extraction from visually rich documents using l1m-based organization of documents into independent textual segments. In *ACL*.

Showmik Bhowmik. 2023. *Document layout analysis*, volume 3. Springer.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *arXiv:2405.13792*.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv:2411.04952*.

Timo I Denk and Christian Reisswig. 2019. Bertgrid: Contextualized embedding for 2d document representation and understanding. *arXiv preprint arXiv:1909.04948*.

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

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. 2022. Doc2ppt: Automatic presentation slides generation from scientific documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 634–642.

Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinquin Yan, Yu Fang, Florian Kleber, and Eva Lang. 2019. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International*

- conference on document analysis and recognition (ICDAR)*, pages 1510–1515. IEEE.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Monica Haurilet, Ziad Al-Halah, and Rainer Stiefelhagen. 2019. Spase-multi-label page segmentation for presentation slides. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 726–734. IEEE.
- huridocs. 2025. Pdf document layout analysis: A docker-powered microservice for intelligent pdf document layout analysis, ocr, and content extraction.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.
- Yiqiao Jin, Kartik Sharma, Vineeth Rakesh, Yingtong Dou, Menghai Pan, Mahashweta Das, and Srijan Kumar. 2025. Sara: Selective and adaptive retrieval-augmented generation with context compression. *arXiv:2507.05633*.
- Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. Agentreview: Exploring peer review dynamics with llm agents. In *EMNLP*, pages 1208–1226.
- Thomas G Kieninger. 1998. Table structure recognition based on robust block segmentation. In *Document recognition V*, volume 3305, pages 22–32. SPIE.
- Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy-yong Sohn, and Chanyeol Choi. 2024. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog.
- Annie Lang. 2000. The limited capacity model of mediated message processing. *Journal of communication*, 50(1):46–70.
- VI Levenshtcin. 1966. Binary coors capable or ‘correcting’ deletions, insertions, and reversals. In *Soviet physics-doklady*, volume 10.
- Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv:2308.03281*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *NeurIPS*, 36:34892–34916.
- Haowei Liu, Xi Zhang, Haiyang Xu, Yaya Shi, Chaoya Jiang, Ming Yan, Ji Zhang, Fei Huang, Chunfeng Yuan, Bing Li, and 1 others. 2024. Mibench: Evaluating multimodal large language models over multiple images. *arXiv preprint arXiv:2407.15272*.
- Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, and 1 others. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010.
- Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.
- Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *CVPR*, pages 1697–1706.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *CVPR*, pages 2200–2209.
- Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.
- Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.
- Laura F Naysmith, Veena Kumari, and Steven CR Williams. 2021. Neural mapping of prepulse-induced startle reflex modulation as indices of sensory information processing in healthy and clinical populations: A systematic review. *Human Brain Mapping*, 42(16):5495–5518.
- OpenAI. 2025. Gpt-4o.
- Maciej P Polak and Dane Morgan. 2025. Leveraging vision capabilities of multimodal llms for automated data extraction from plots. *arXiv:2503.12326*.
- Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, pages 732–747. Springer.

- Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. Docparser: Hierarchical document structure parsing from renderings. In *AAAI*, volume 35, pages 4328–4338.
- Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *CIKM*, pages 42–49.
- Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Karun Sharma and Vidushee Vats. 2025. Think to ground: Improving spatial reasoning in llms for better visual grounding. In *Workshop on Reasoning and Planning for Large Language Models*.
- Michael Shilman, Percy Liang, and Paul Viola. 2005. Learning nongenerative grammatical models for document analysis. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 962–969. IEEE.
- Jeremy Singer-Vine. 2025. [pdfplumber](#).
- Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.
- Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.
- Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy XR Wang. 2021. D2s: Document-to-slide generation via query-based text summarization. *arXiv preprint arXiv:2105.03664*.
- Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. 2025. Vdocrag: Retrieval-augmented generation over visually-rich documents. *arXiv:2504.09795*.
- Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidenvqa: A dataset for document visual question answering on multiple images. In *AAAI*, volume 37, pages 13636–13645.
- Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13878–13888.
- Deep Search Team. 2024. [Docling technical report](#). Technical report.
- Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2021. Document collection visual question answering. In *International Conference on Document Analysis and Recognition*, pages 778–792. Springer.
- Michael Tschanne, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.
- Gaurav Verma, Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Tucker Balch, and Manuela Veloso. 2024. Adaptagent: Adapting multimodal web agents with few-shot learning from human demonstrations. In *ACL*.
- Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armeneh Nourbakhsh, and Xiaomo Liu. 2024a. Docilm: A layout-aware generative language model for multimodal document understanding. In *ACL*, pages 8529–8548.
- Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. 2024b. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *NeurIPS*, 37:75392–75421.
- Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025a. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. *arXiv:2502.18017*.
- Yiyang Wang, Rishabh Goel, Sheraz Hassan, Taegen J Doscher, Shilin Wang, Lexington Allen Whalen, Aditya S Gandhi, Yaman S Sangar, Alex Cabral, Xuhai Xu, and 1 others. 2025b. Puffem: An e-cigarette sleeve for estimating user nicotine intake. In *CHASE*, pages 129–133. IEEE.
- Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. 2025. Real-mm-rag: A real-world multimodal retrieval benchmark. In *ACL*.
- Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2024. Next-gpt: Any-to-any multimodal llm. In *ICML*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muenninghoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, *arXiv:2309.07597*.
- Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, and 1 others. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200.

Yunqiu Xu, Linchao Zhu, and Yi Yang. 2024. Mc-bench: A benchmark for multi-context visual grounding in the era of mllms. *arXiv preprint arXiv:2410.12332*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chun-yuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Jun-hao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and 1 others. 2025. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *ICLR*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xue-hui Wang, Yue Cao, Yangzhou Liu, Xinguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *Preprint, arXiv:2504.10479*.

Model	Overall	Num	F1
<i>Raw LLMs</i>			
Llama 3.2 11B	55.4	49.6	67.9
Phi3	53.4	43.1	88.7
Qwen2.5 7B	<b>80.1</b>	<b>73.0</b>	<b>96.4</b>
Qwen2.5 32B	<u>72.1</u>	<u>62.0</u>	<u>94.8</u>
LLaVA 1.5 7B	25.1	10.6	83.9
LLaVA 1.5 13B	20.2	8.0	84.2
LLaVA 1.6 7B	32.6	22.6	79.9
LLaVA 1.6 13B	34.9	24.8	80.9
InternVL3 8B	66.7	57.7	85.2
<i>Multimodal RAG and Agentic Methods</i>			
ViDoRAG	67.1	59.2	85.9
SlideAgent	<u>75.4</u>	<u>66.5</u>	<u>92.1</u>
Impr.	+8.7	+8.8	+6.8

Table 5: Performance comparison among open-source multimodal models and agentic methods.

## A Details Method

### A.1 Element Merging

Raw output, especially text spans from the element processing pipeline (Section 2.1) can fragment coherent texts into multiple parts, creating challenges

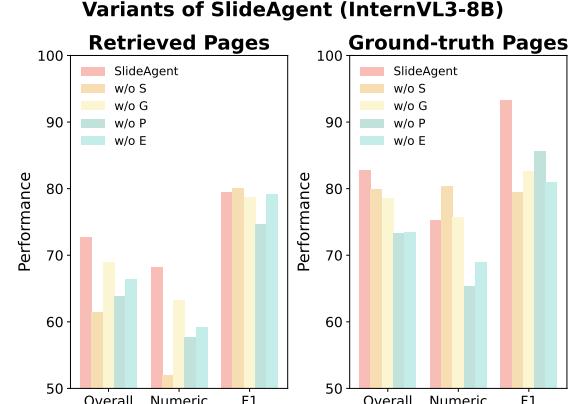


Figure 6: Performance comparison among variants of SlideAgent with base model InternVL3-8B.

for downstream understanding. We thus adopt a graph-based merging algorithm to reconstruct semantically coherent text blocks while preserving spatial layout.

**Distance-Based Adjacency** Our merging criterion is based on minimum distance between bounding boxes. For two boxes  $b_i = (x_1^i, y_1^i, x_2^i, y_2^i)$  and  $b_j = (x_1^j, y_1^j, x_2^j, y_2^j)$ , the minimum distance is computed from the horizontal and vertical distances  $d_h$  and  $d_v$ :

$$d_{\min}(b_i, b_j) = \sqrt{d_h^2 + d_v^2}$$

where  $d_h = \min(|x_2^i - x_1^j|, |x_2^j - x_1^i|)$  if boxes don't overlap horizontally, and  $d_h = 0$  otherwise. Similarly,  $d_v = \min(|y_2^i - y_1^j|, |y_2^j - y_1^i|)$  if boxes don't overlap vertically, and  $d_v = 0$  otherwise.

Two boxes are considered adjacent if  $d_{\min}(b_i, b_j) \leq \tau$  where  $\tau$  is a threshold we choose to be 15 pixels. The results are fed into the following graph-based component detection.

## B Extended Related Work

### B.1 Document and Infographics Understanding

Early research in document and infographic understanding predates multimodal LLMs and was dominated by computer vision pipelines. Classical approaches typically combined OCR, layout parsing, and heuristic rules to extract semantics from text-rich visual content such as forms, tables, and scientific figures (Shilman et al., 2005; Bhowmik, 2023; Gao et al., 2019; Kieninger, 1998; Smith, 2007). While these methods enabled structured information extraction, they were brittle to domain shifts and noisy scans.

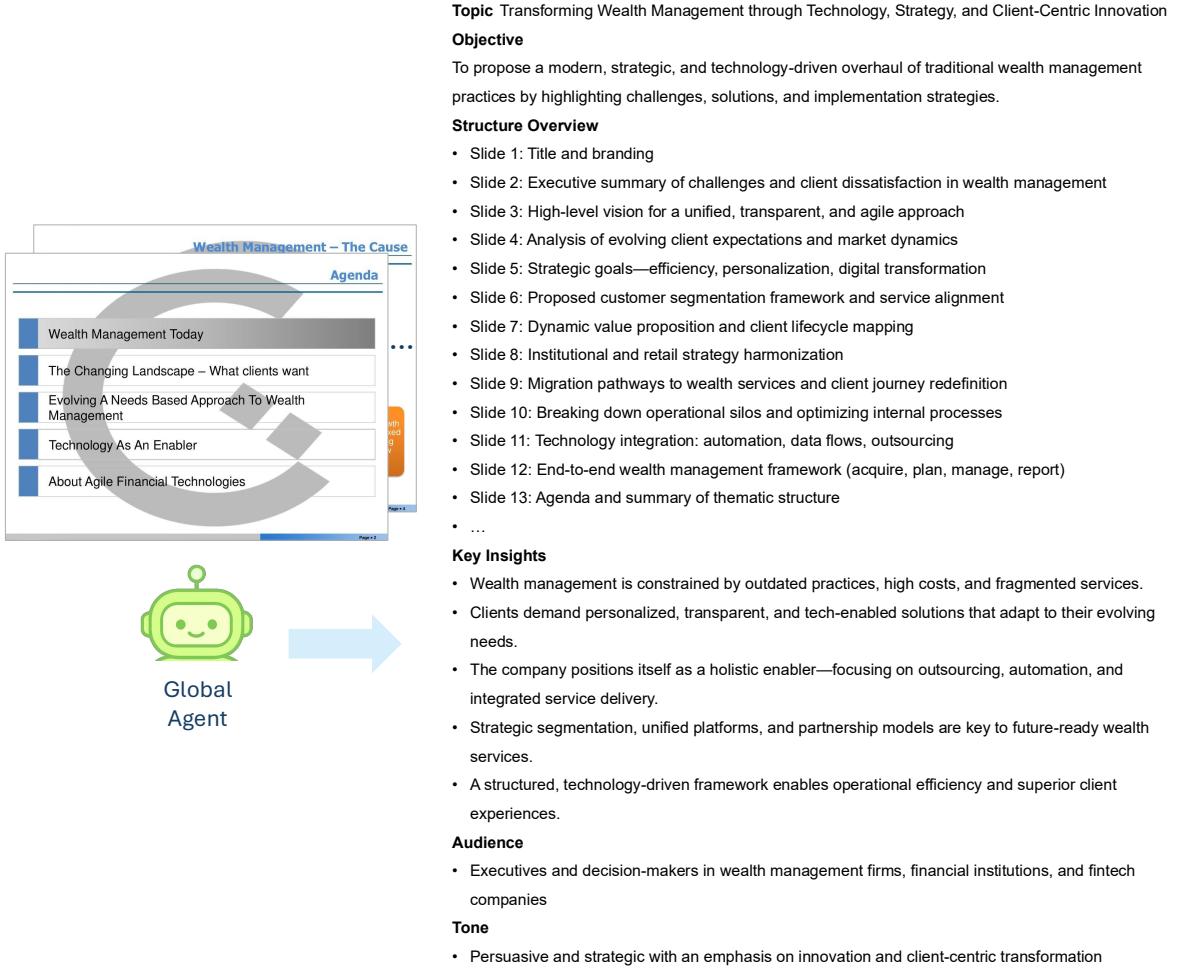


Figure 7: Sample global knowledge  $\mathcal{K}_g$  generated by the global agent, showing document-level summary, objectives, and narrative structure.

Notation	Description
$p_i, v_i$	A slide page and its visual content
$\mathcal{P} = \{p_1, p_2, \dots\}$	Set of pages in the slide deck
$q$	User query
$\hat{\mathcal{Q}}$	Set of subqueries generated from the original query $q$
$a$	The final answer generated by SlideAgent
$e_j$	Individual element in a slide, consisting of visual content, bounding box, and type
$b_j$	Bounding box coordinates of an element within a slide
$t_j$	Type of the element (e.g., text, chart, image, table)
$\mathcal{K}_g, \mathcal{K}_p, \mathcal{K}_e$	Global, page, and element knowledge
$\mathcal{M}_g, \mathcal{M}_p, \mathcal{M}_e$	Global, page, and element agent
$\mathcal{R}$	Retrieval function that fetches relevant pages or elements based on subqueries
$\phi(\cdot)$	Answer synthesizer that combines reasoning from all agents to generate the final answer $a$
$\text{Annot}(\hat{V})$	Annotated visual content used by the element agent for processing
$h_g, h_p, h_e$	Answers and reasoning from global, page, and element agents

Table 6: Mathematical notation used throughout the paper.

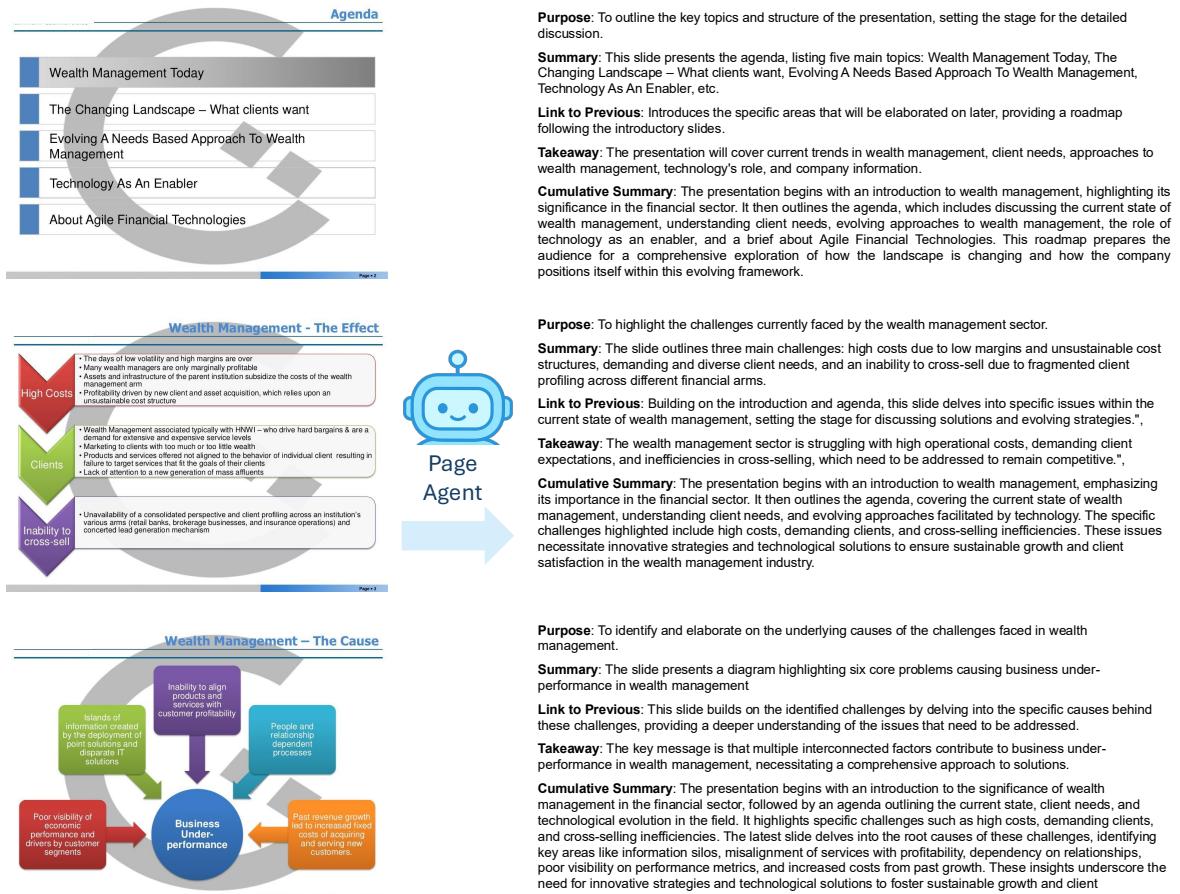


Figure 8: Sample page-level knowledge representation  $\mathcal{K}_p^i$  generated by the page agent, capturing slide-specific content and cross-slide relationships.

Case	Characteristics	Example	Agents
1-Global Understanding	Asks about the overall theme, purpose, or general summary of the entire presentation.	"What is the main topic of the presentation?", "What is this deck about?"	Global
2-Fact-based Direct Query	Asks for specific facts, data, or information from particular slides.	"What is the revenue reported on slide 7?", "Which slide shows the product roadmap?"	Page, Element
3-Multi-hop Reasoning	Requires comparing information across multiple slides or elements.	"Compare revenues in slide 5 and slide 10", "How do Sweden and Denmark compare?"	Global, Page, Element
4-Layout / Visual Relationship	Asks about visual relationships, positioning, or layout elements.	"What does the diagram below the table on slide 12 illustrate?", "Is the color red used to denote negative performance?"	Element
5-Uncertain	If the query is unclear, use all agents to answer the query.	\	Global, Page, Element

Table 7: SlideAgent's query classification that determines which hierarchical agents to activate.



**A** **Content:** Evolving A Needs Based Approach To Wealth Management  
**Semantic Role:** This element introduces the topic of evolving strategies in wealth management.  
**Functional Purpose:** It aims to inform the audience about the need to adapt wealth management practices to better meet client needs.  
**Relation to Slide:** This element is part of the agenda, outlining a key discussion point regarding innovative approaches in financial management.  
**Bbox:** [0.11, 0.48, 0.79, 0.60]

**B** **Content:** Agenda  
**Semantic Role:** This element indicates the purpose of the slide, which is to outline the agenda.  
**Functional Purpose:** It provides clarity to the audience, signaling that the slide contains the main topics to be covered in the presentation.  
**Relation to Slide:** It frames the slide's content, helping the audience understand the structure and flow of the presentation.  
**Bbox:** [0.84, 0.044, 0.98, 0.11]

**A** **Content:** Wealth Management associated typically with HNWI ...  
**Functional Purpose:** Emphasize the challenge of meeting client expectations and tailoring services appropriately.  
**Relation to Slide:** Supports the point about demanding and diverse client needs, reinforcing the complexity and high expectations from clients.  
**Bbox:** [0.20, 0.42, 0.95, 0.60]

**C** **Content:** Confidential  
**Semantic Role:** Indicate the confidentiality of the slide content.  
**Functional Purpose:** Ensure viewers are aware the information is confidential.  
**Relation to Slide:** Serves as a footer to remind viewers of the sensitive nature of the information.  
**Bbox:** [0.03, 0.98, 0.10, 1.00]

**A** **Content:** Islands of information created by the deployment of point solutions and disparate IT solutions...  
**Semantic Role:** Highlighting a specific problem related to fragmented information systems.  
**Functional Purpose:** Emphasize the issue of information silos in wealth management.  
**Relation to Slide:** Supports the slide's overall message of identifying core problems causing business under-performance.  
**Bbox:** [0.14, 0.33, 0.36, 0.51]

**B** **Content:** Past revenue growth led to increased fixed costs of acquiring and serving new customers...  
**Semantic Role:** Highlighting a specific problem related to increased costs from past growth.  
**Functional Purpose:** Emphasize the issue of rising fixed costs due to past revenue growth.  
**Relation to Slide:** Supports the slide's overall message of identifying core problems causing business under-performance.  
**Bbox:** [0.74, 0.67, 0.96, 0.83]

Figure 9: Sample element-level knowledge representation  $\mathcal{K}_e^j$  generated by the element agent.

With the rise of deep learning, several specialized benchmarks spurred progress in document visual question answering (VQA). Datasets such as DocVQA (Mathew et al., 2021), DocCVQA (Tito et al., 2021), FUNSD (Jaume et al., 2019), and DocBank (Li et al., 2020) emphasized reasoning over diverse document layouts, scanned forms, and large-scale document collections. ChartQA (Masry et al., 2022) further extended this line by introducing reasoning over structured figures like bar and line charts. On the modeling side, early work on text-centric VQA demonstrated that reading and interpreting embedded text was crucial (Singh et al., 2019; Mishra et al., 2019). To better represent visually rich documents, approaches such as BERTgrid (Denk and Reisswig, 2019) and LayoutLM (Xu et al., 2020b) proposed contextualized embeddings that jointly encode textual content and 2D spatial layout. This line was extended by LayoutLMv2 (Xu et al., 2020a), LayoutT5 (Tanaka et al., 2021), and TILT (Powalski et al., 2021), which integrated stronger visual features and multimodal pretraining objectives. These advances have achieved impressive results on single-image document VQA tasks by unifying textual, layout, and visual signals, laying the foundation for more recent LLM-based systems.

### B.1.1 Slide Deck Understanding

Building on document and infographic understanding, research has also explored presentation slides, which pose unique challenges due to their combination of dense text, figures, and multi-page structure. Early efforts addressed component-level analysis, such as object segmentation on slide pages (Haurilet et al., 2019), or generation tasks like creating slides from research papers (Sun et al., 2021; Fu et al., 2022). More recently, benchmarks such as MMLongBench (Ma et al., 2024) have emphasized long-context processing over lengthy multi-page PDFs, highlighting the scalability issues that arise when moving from single documents to multi-slide collections.

### B.1.2 Slide Visual Question Answering

Slide visual question answering (Slide VQA) emerges as a prominent direction within this broader area, aiming to automatically interpret presentation slides to answer natural-language queries. Early work such as SlideVQA (Tanaka et al., 2023) introduced multi-modal slide decks paired with questions requiring single-hop, multi-hop, and nu-

merical reasoning across slides. This line was extended by InfoVQA (Mathew et al., 2022), which targeted information-centric slides and documents, often involving arithmetic and commonsense reasoning. More recent benchmarks such as TechSlides and FinSlides (Wasserman et al., 2025) from the REAL-MM-RAG suite emphasize domain-specific contexts—technical and financial presentations that integrate text, figures, and tables.

At the same time, multi-image and multi-context evaluation benchmarks such as MIBench (Liu et al., 2024) and MC-Bench (Xu et al., 2024) highlight the unique reasoning challenges in multi-slide and multi-panel scenarios. Retrieval-based methods such as ColPali (Faysse et al., 2024) improve slide-grounded search and QA, while pipeline approaches like VisRAG (Yu et al., 2025) and VDocRAG (Tanaka et al., 2025) integrate retrieval with reasoning to support multi-page comprehension. Collectively, these datasets and methods underscore both the promise and complexity of slide understanding, while leaving open the question of whether element-level reasoning—beyond page-level retrieval—can substantially improve performance.

### B.1.3 General Purpose LLMs

General-purpose large language models (MLLMs) such as GPT-4 (Achiam et al., 2023), GPT-4o (Hurst et al., 2024), Gemini (Anil et al., 2023), LLaVA (Liu et al., 2023), InternVL3 (Zhu et al., 2025), and Visual CoT (Shao et al., 2024) have significantly advanced visual reasoning across a wide range of tasks. Visual prompting methods like Set-of-Marks (Yang et al., 2023) further enhance reasoning by augmenting inputs with structured visual annotations. While these open-source and closed-source MLLMs form the foundation for many comparisons, they often struggle when applied to slides, where precise grounding and fine-grained interpretation are required to parse heterogeneous elements such as charts, tables, and icons.

Yet key challenges remain. Domain-specific visual semantics are often overlooked, as LLMs trained primarily on natural images (Wu et al., 2024) struggle to capture the specialized conventions of slides and infographics—for instance, repetitive logos, color-coded encodings, or abstract symbolic icons. Metadata-dependent integration is similarly fragile: real-world slides and PDFs frequently lack clean structural annotations, and scanned or flattened exports strip away layout cues,

rendering metadata-reliant systems brittle (huri-docs, 2025; Rausch et al., 2021). Finally, scalable fine-grained reasoning capabilities remains limited (Jin et al., 2024). Current models can process only a small number of images at once (Liu et al., 2023), while retrieval typically occurs at the page level (Faysse et al., 2024; Tanaka et al., 2025), overlooking element-level reasoning.

Our work builds on these developments by introducing a metadata-free, slide-specialized agentic framework. Unlike prior page-level approaches, our method constructs hierarchical representations and employs retrieval-then-reasoning at the global (deck-level), page (slide-level), and element (fine-grained component) layers. This design enables robust comprehension across diverse slide domains and opens the door to systematic evaluation of whether element-level reasoning provides measurable improvements over page-level retrieval alone.

**Adjacency Graph Construction.** An undirected graph  $G = (V, E)$  is constructed, where vertices  $V = \{1, 2, \dots, |B|\}$  represent the bounding boxes for the textual elements, and  $E$  edges connect adjacent boxes. An edge  $(i, j) \in E$  exists if  $d_{\min}(b_i, b_j) \leq \tau$ .

**Connected Component Detection.** We identify coherent text spans using depth-first search (DFS) to find connected components. Each component  $C_k = \{i_1, i_2, \dots, i_m\}$  represents indices of boxes that should be merged into a single text block.

**Spatial Merging and Text Concatenation.** For each component  $C_k$ , we compute the unified bounding box:

$$B_k = \left( \min_{i \in C_k} x_1^i, \min_{i \in C_k} y_1^i, \max_{i \in C_k} x_2^i, \max_{i \in C_k} y_2^i \right)$$

The final text is given by concatenating text in the reading order. This is achieved by sorting boxes within each component using lexicographic ordering: primary sort by vertical position ( $y_1$ ), secondary sort by horizontal position ( $x_1$ ).

This approach effectively reduces element fragmentation while preserving spatial relationships, enabling more accurate element-level descriptions for our hierarchical framework. Algorithm 1 provides the complete implementation details.

## B.2 Answer Matching

To determine answer equivalence, i.e. if two agents agree upon their opinions, we do fuzzy string matching between each pair of answers  $a_1, a_2$  us-

ing Normalized Levenshtein Similarity (NLS):

$$\text{NLS}(a_1, a_2) = \left( 1 - \frac{\text{editdistance}(a_1, a_2)}{\max(|a_1|, |a_2|)} \right), \quad (4)$$

where  $\text{editdistance}(a_1, a_2)$  computes the Levenshtein distance (Levenshtcin, 1966) between the answers after tokenization. Two answers are considered equivalent if  $\text{NLS}(a_1, a_2) \geq 0.75$ .

## C Experimental Setup

### C.1 Dataset Descriptions

- SlideVQA (Tanaka et al., 2023) is a multi-modal, multi-image VQA dataset consisting of  $\geq 2600$  slide decks that require single-hop, multi-hop, and numerical reasoning. Each slide deck corresponds to multiple questions. The license is available at <sup>1</sup>.
- InfoVQA (Mathew et al., 2022) is a dataset that feature questions that require reasoning and arithmetic skills. It contains over 30,000 questions with over 5,400 images. The dataset is released under CC-BY license.
- TechSlides and FinSlides (Wasserman et al., 2025) are from the REAL-MM-RAG benchmark, which comprises slides and documents with text, figures, tables, and images, requiring systems to handle combined textual and visual data. The dataset is released under the CDLA-Permissive-2.0 license.

### C.2 Metrics

For ranking evaluation, we adopt MRR, Hit@k, and nDCG@k. Mean Reciprocal Rank (MRR) reflects how early the correct answer appears in the ranked list by averaging the reciprocal of its first relevant position across all queries. Hit@k measures the proportion of queries for which at least one correct answer occurs within the top- $k$  retrieved results. Normalized Discounted Cumulative Gain (nDCG@k) assesses overall ranking quality by weighting relevant items according to their positions and normalizing by the ideal ranking, thereby capturing both accuracy and ordering of retrieved answers.

<sup>1</sup>[https://github.com/nttmdlab-nlp/SlideVQA/  
blob/main/LICENSE](https://github.com/nttmdlab-nlp/SlideVQA/blob/main/LICENSE)

### C.3 Implementation Details

We used EasyOCR<sup>2</sup> and Docling (Team, 2024) for detecting textual and visual elements, respectively. Whenever applicable, the answers are generated using a temperature of 0.0 to ensure deterministic results.

---

**Algorithm 1** Graph-based depth-first search for merging fragmented elements into coherent semantic units while preserving spatial layout.

---

**Input:** Set of OCR bounding boxes  $B = \{b_i = (x_1^i, y_1^i, x_2^i, y_2^i, t_i)\}_{i=1}^{|B|}$ , distance threshold  $\tau = 15$   
**Output:** Merged bounding boxes  $\tilde{B}$

```

1: Initialize adjacency matrix  $A \in \{0, 1\}^{|B| \times |B|}$ 
   with zeros
2: for  $i = 1$  to  $|B|$  do
3:   for  $j = i + 1$  to  $|B|$  do
4:     if  $d_{\min}(b_i, b_j) \leq \tau$  then
5:        $A[i, j] \leftarrow 1$ ,  $A[j, i] \leftarrow 1$ 
6:     end if
7:   end for
8: end for
9: Initialize visited array  $\text{visited}[1 : |B|] \leftarrow \text{false}$ 
10: Initialize components list  $\mathcal{C} \leftarrow []$ 
11: for  $i = 1$  to  $|B|$  do
12:   if  $\text{visited}[i] = \text{false}$  then
13:     Initialize component  $C \leftarrow []$ 
14:     DFS( $i, A, \text{visited}, C$ )            $\triangleright$  Collect
        connected component
15:     Append  $C$  to  $\mathcal{C}$ 
16:   end if
17: end for
18: Initialize merged boxes  $\tilde{B} \leftarrow []$ 
19: for each component  $C \in \mathcal{C}$  do
20:   if  $|C| = 1$  then
21:     Append  $b_{C[0]}$  to  $\tilde{B}$        $\triangleright$  Single box, no
        merging
22:   else
23:      $x_1 \leftarrow \min_{i \in C} x_1^i$ ,  $y_1 \leftarrow \min_{i \in C} y_1^i$ 
24:      $x_2 \leftarrow \max_{i \in C} x_2^i$ ,  $y_2 \leftarrow \max_{i \in C} y_2^i$ 
25:     Sort  $C$  by  $(y_1^i, x_1^i)$  to get reading order
26:      $t_{\text{merged}} \leftarrow " ".join(\{t_i\}_{i \in \text{sorted}(C)})$ 
27:     Append  $(x_1, y_1, x_2, y_2, t_{\text{merged}})$  to  $\tilde{B}$ 
28:   end if
29: end for
30: return  $\tilde{B}$ 
```

---

## D Additional Experiments

### D.1 Ablation Studies (Cont’d)

**Element-level Reasoning Ensures Precision** Ablating the element agent causes a moderate yet consistent decline (−4.6 overall for GPT-4o; −6.3 for InternVL3-8B), especially numeric questions. Even with perfect retrieval, omitting fine-grained texts and layouts weakens factual grounding.

**Impact of Global Thematic Guidance** Removing the global agent yields the smallest drop (−2.8 GPT-4o; −3.7 InternVL3-8B), as lower-level agents already embed partial global context via  $\mathcal{K}_p$  and  $\mathcal{K}_e$  in knowledge construction (Eq. 1/2). However, responses become less aware of overarching themes.

**Subquery Generation Strengthens Retrieval.** Removing subqueries causes larger losses under retrieval (−5.0 GPT-4o; −11.3 InternVL3-8B) than with ground-truth pages (−2.9 to −4.9). Visual-aware subqueries markedly improve retriever accuracy by aligning textual intent with visual semantics—especially beneficial for weaker open-source encoders.

## E Future Work

**Element Parsing.** SlideAgent relies on advanced text and layout detection tools, which may struggle with visually complex or low-contrast designs (e.g., white charts on white backgrounds). Enhancing robustness in parsing could further improve overall reliability.

**Modeling Element Relations.** SlideAgent currently treats elements independently. Extending it to explicitly model inter-element relations—for instance, via graph structures linking textual and visual components—offers a promising direction for capturing richer semantics, though at higher computational cost.

**User-Centric Interaction.** While SlideAgent primarily enhances reasoning and retrieval for slide understanding, future work could make such agentic frameworks more user-centric through seamless integration into real-world user behaviors (Wang et al., 2025b). Lightweight feedback mechanisms (e.g., contextual pop-ups or in-situ clarifications) could further improve interpretability and interaction without disrupting natural workflows or requiring excessive user input.

<sup>2</sup><https://github.com/JaidedAI/EasyOCR>

<b>Family</b>	<b>Model</b>	<b>Parameters</b>	<b>Knowledge Cutoff</b>	<b>Release</b>
GPT ( <a href="#">OpenAI, 2025</a> )	gpt-4o	\	Oct 2023	May 2024
Gemini ( <a href="#">Anil et al., 2023</a> )	gemini-2.5-flash	\	Jan 2025	June 2025
	gemini-2.5-flash-lite	\	Jan 2025	June 2025
	gemini-2.0-flash	\	June 2024	Jan 2025
Claude ( <a href="#">Anthropic, 2025</a> )	claude-3-5-haiku-latest	\	April 2024	Oct 2024
	claude-opus-4-1-20250805	\	Jan 2025	Aug 2025
Llama ( <a href="#">Grattafiori et al., 2024</a> )	Llama-3.2-11B-Vision-Instruct	11B	\	Sep 2024
InternVL ( <a href="#">Chen et al., 2024</a> )	InternVL3-8B	8B	\	Apr 2025
Phi ( <a href="#">Abdin et al., 2024</a> )	Phi-3-vision-128k-instruct	3.8B	Oct 2023	July 2024
Qwen ( <a href="#">Bai et al., 2025</a> )	Qwen2.5-VL-7B-Instruct	7B	\	Feb. 2025
	Qwen2.5-VL-32B-Instruct	32B	\	Feb 2025
LLaVA ( <a href="#">Liu et al., 2023</a> )	llava-1.5-7b-hf	7B	Dec 2022	Apr. 2023
	llava-1.5-13b-hf	13B	Dec 2022	Apr. 2023
	llava-v1.6-mistral-7b-hf	7B	Dec 2023	Mar 2024
	llava-v1.6-vicuna-13b-hf	13B	Dec 2023	Mar 2024

Table 8: Overview of models used in the experiments, including their family, model name, parameter size, knowledge cutoff date, and release date.

---

You are given a complete slide deck consisting of multiple slides. Your task is to synthesize a concise, high-level summary of the overall message, structure, and purpose of the deck.

### **### Cumulative Summary of the Preceding Slides**

{Cumulative Summary}

**### Response format** Please respond with a markdown string that follows the following format:  
**Title** <Concise Explicit / Inferred Title of the Presentation>

**Objective** <What is the presentation trying to achieve? (e.g., inform, persuade, pitch, propose)>

### **Structure Overview**

- **Slide 1:** <Brief description of the slide>
- **Slide 2:** <Brief description of the slide>
- ...
- **Slide N:** <Brief description of the slide>

### **Key Insights**

- <Major takeaway 1>
- <Major takeaway 2>
- ...

**Audience** <Intended audience type (e.g., executives, investors, engineers)>

**Tone** <Overall tone: e.g., persuasive, analytical, optimistic, urgent>

---

Table 9: Prompt template for the global agent to generate comprehensive slide deck summaries, including title, objective, structure, key insights, audience, and tone.

---

**Element Type** <text | image | chart | table | icon | button | etc.>

**Position on Slide** <e.g., top-right, centered, below title>

**Verbatim Content** <if text, give the literal string; else describe the visual>

**Semantic Role** <What is the element trying to do or communicate?>

**Functional Purpose** <Its practical function within the slide (e.g., emphasize point, guide attention, show evidence, support action)>

**Relation to Slide** <How does it connect to or support the slide's overall message?>

**Inferred Importance** <how central is this element to the slide? Answer with low, medium, or high>

---

Table 10: Element-level slide description prompt format used by the element agent to generate detailed annotations for each visual component.

Model	SlideVQA			TechSlides			FinSlides		
	Overall	Num	F1	Overall	Num	F1	Overall	Num	F1
<i>Raw Models</i>									
Gemini 2.0	86.3	81.0	90.3	59.7	60.0	<u>59.6</u>	78.1	77.8	<b>88.9</b>
Gemini 2.5	<b>89.0</b>	<b>85.7</b>	<b>93.2</b>	61.5	65.0	<u>59.5</u>	76.6	76.4	83.3
Gemini 2.5-lite	81.8	75.5	90.1	56.3	55.0	57.0	78.1	78.4	66.7
Claude 4.1	85.7	82.4	89.7	58.0	77.5	48.3	52.6	52.0	60.8
Claude 3.5	58.2	64.0	50.9	55.8	<b>83.7</b>	42.5	47.8	48.0	40.3
GPT-4o	79.4	71.9	86.4	64.5	77.1	58.0	83.0	84.3	80.1
<i>Multimodal RAG and Agentic Methods</i>									
ViDoRAG	81.8	73.8	87.0	<u>65.8</u>	78.0	58.7	84.2	84.7	81.5
SlideAgent	<u>87.1</u>	<u>84.4</u>	<u>90.6</u>	<b>68.7</b>	<u>82.5</u>	<b>61.5</b>	<b>85.8</b>	<b>85.9</b>	<u>85.6</u>
Impr.	+7.7	+12.5	+4.2	+4.1	+5.4	+3.5	+2.8	+1.5	+5.5

Table 11: Performance comparison of baselines and proprietary models on SlideVQA, TechSlides, and FinSlides, assuming access to ground-truth pages containing the correct answer. All baseline methods use GPT-4o for question-answering.

Model	SlideVQA			TechSlides			FinSlides		
	Overall	Num	F1	Overall	Num	F1	Overall	Num	F1
<i>Raw Models</i>									
Llama 3.2 11B	44.6	52.1	34.6	47.1	62.8	39.3	39.1	39.2	33.5
Phi3	78.3	69.1	91.6	53.9	67.4	47.2	<u>63.8</u>	<u>63.7</u>	65.1
Qwen2.5 7B	<u>85.1</u>	<u>77.7</u>	<b>95.3</b>	57.7	69.8	51.8	52.7	52.0	<b>77.8</b>
Qwen2.5 32B	<b>87.4</b>	<b>82.6</b>	<u>93.7</u>	49.6	62.8	43.2	<b>69.5</b>	<b>69.6</b>	65.1
LLaVA 1.5 7B	42.9	27.9	79.9	24.6	15.0	39.4	14.2	14.4	20.9
LLaVA 1.5 13B	46.7	29.5	83.1	29.2	17.5	46.6	23.8	20.3	41.2
LLaVA 1.6 7B	59.0	45.8	84.2	36.2	40.0	34.0	16.0	15.2	21.3
LLaVA 1.6 13B	62.3	48.2	87.1	54.7	62.5	49.5	35.2	30.7	67.6
InternVL3 8B	73.3	65.4	85.7	58.4	72.1	51.5	56.3	55.9	65.6
<i>Baseline methods based on InternVL3-8B</i>									
ViDoRAG	76.3	68.1	89.9	<u>61.3</u>	<u>75.1</u>	<u>53.9</u>	58.4	58.7	67.3
SlideAgent	82.8	75.3	93.3	<b>64.6</b>	<b>79.5</b>	<b>58.0</b>	62.8	62.6	<u>68.1</u>
Impr.	+9.5	+9.8	+7.6	+6.2	+7.4	+6.4	+6.5	+6.7	+2.5

Table 12: Performance comparison of baselines and proprietary models on SlideVQA, TechSlides, and FinSlides, assuming access to ground-truth pages containing the correct answer. All baseline methods use InternVL3-8B for question-answering.