SlideBot: A Multi-Agent Framework for Generating Informative, Reliable, Multi-Modal Presentations

Eric Xie[1], **Danielle Waterfield**[1], **Michael Kennedy**[1], **Aidong Zhang**[1]
[1]University of Virginia
{jrg4wx, dla9ck, mjk3p, aidong}@virginia.edu

Abstract

Large Language Models (LLMs) have shown immense potential in education, automating tasks like quiz generation and content summarization. However, generating effective presentation slides introduces unique challenges due to the complexity of multimodal content creation and the need for precise, domain-specific information. Existing LLM-based solutions often fail to produce reliable and informative outputs, limiting their educational value. To address these limitations, we introduce SlideBot - a modular, multi-agent slide generation framework that integrates LLMs with retrieval, structured planning, and code generation. SlideBot is organized around three pillars: **informativeness**, ensuring deep and contextually grounded content; **reliability**, achieved by incorporating external sources through retrieval; and **practicality**, which enables customization and iterative feedback through instructor collaboration. It incorporates evidence-based instructional design principles from Cognitive Load Theory (CLT) and the Cognitive Theory of Multimedia Learning (CTML), using structured planning to manage intrinsic load and consistent visual macros to reduce extraneous load and enhance dual-channel learning. Within the system, specialized agents collaboratively retrieve information, summarize content, generate figures, and format slides using LaTeX, aligning outputs with instructor preferences through interactive refinement. Evaluations from domain experts and students in AI and biomedical education show that SlideBot consistently enhances conceptual accuracy, clarity, and instructional value. These findings demonstrate SlideBot's potential to streamline slide preparation while ensuring accuracy, relevance, and adaptability in higher education.

Extended version — https://arxiv.org/abs/2511.09804

1 Introduction

Artificial intelligence (AI) has transformed education by automating tasks such as quiz generation and content summarization, reducing instructor workloads, and improving student learning experiences (Elkins et al. 2024; Agrawal et al. 2024; Fagbohun et al. 2024). Among these advancements, large language models (LLMs) stand out due to their ability to follow user instructions (Ouyang et al. 2022) and effectively process contextual information, allowing them to perform complex tasks in diverse domains (Achiam et al. 2023; Touvron et al. 2023). Their proficiency in understanding and generating text unlocks significant potential for automating tasks in education.

Slide presentations are a core instructional medium that facilitates structured and engaging multimedia content that supports deeper student learning (Alley and Neeley 2005; Bartsch and Cobern 2003; Mayer 2005b). However, generating slides is time-consuming and demands coordination across multiple modalities, text, visuals, and layout, a task that challenges existing LLM-based solutions. Without access to external information, LLMs rely on their parametric knowledge, the internal representations learned from their training data. In other words, the models draw on the learned statistical patterns and associations to generate content. This increases the risk of generating outdated or hallucinated content, statements that seem credible but lack factual accuracy, raising reliability concerns in educational contexts (Flier 2023; Cremin, Dash, and Huang 2022; Bender et al. 2021). These limitations are especially problematic in evolving fields such as biomedicine or AI, where domain-specific knowledge is complex and rapidly updated (Ahmad, Yaramis, and Roy 2023).

In addition to using factual domain-specific knowledge, pedagogical knowledge is key when generating slides, as effective instructional materials must align with how people learn and how teaching practices are mediated by tools and contexts. The importance of pedagogical knowledge is evidenced by cognitive load theory (CLT) and the cognitive theory of multimedia learning (CTML) (Chandler and Sweller 1991; Mayer 2005a). CLT posits that one's working memory has an overall limited capacity for processing new information, which is divided into three types: intrinsic (complexity of new information), extraneous (distractions from new information), and germane (effort used to process and integrate new information) (Chandler and Sweller 1991). Effectively designed slides optimize overall cognitive load by reducing or eliminating distractions to support germane load and promote schema construction (Paas and Sweller 2014). Automating slide generation with LLMs helps reduce extraneous load by embedding evidence-based design principles such as eliminating redundant text and aligning visuals with key points that ultimately minimize unnecessary processing demands. While CLT emphasizes managing overall cognitive resources, CTML extends this by focusing on how learn-

ers process information across channels as well as how well-designed multimedia materials can optimize this processing to support meaningful learning (Mayer 2002, 2005b). CTML posits that learners process information through dual visual and auditory channels with limited working memory and meaningful learning occurs when materials support the selection, organization, and integration of information (Mayer 2002, 2005b). Poorly designed slides can allow cognitive overload to occur. As such, automating slide creation with LLMs offers a way to embed CTML principles such as coherence (removing unnecessary information), signaling (highlighting important information), and spatial contiguity (aligning text and visuals), directly into the design process while aiming to avoid cognitive overload.

To address challenges faced by educators in generating slides, we present SlideBot: an agentic slide generation framework that embeds the theoretical aspects of CLT and CTML and leverages large language models in combination with retrieval, structured planning, and modular code generation. SlideBot decomposes the generation process across multiple specialized agents - including retrievers, planners, figure creators, and coders - to produce well-formatted slides from retrieved content and pedagogical prompts. The slides are made using LaTeX[1]-based slides using the Beamer[2] package, combining precise formatting with flexible customization. This structured architecture enables the consistent generation of context-grounded, university-level presentations tailored for instructional use.

We design SlideBot around three core pillars: **informativeness**, ensuring deep, domain-specific coverage; **reliability**, grounding outputs in high-quality external sources; and **practicality**, supporting usability and instructor customization. We validate our approach through both student surveys and expert reviews, demonstrating consistent gains in explanation quality, conceptual accuracy, and overall suitability over Microsoft Copilot, a state-of-the-art presentation generation tool, as well as a direct prompt baseline, where the model produces slides directly from a single prompt without retrieval, planning, or other agentic assistance. By incorporating credible sources and structuring outputs around best practices in multimedia design, our system upholds the pillars by mitigating hallucinations, improving explanation clarity, and providing an adaptable pipeline that instructors can customize to their content, formatting, and pedagogical needs. This interactive, modular approach establishes a new paradigm for reliable and instructor-friendly AI-assisted presentation generation.

2 Related Work

Artificial Intelligence in Education

The integration of artificial intelligence (AI) into education has revolutionized modern classrooms, offering tools that enhance both teaching and learning experiences. AI-powered systems in education can be broadly categorized based on their intended end user: student-centered AI, which focuses on improving student learning, and educator-centered AI, which streamlines teaching workflows through automation (Wang et al. 2024).

Student-Centered Learning Support. Language models have been shown to provide detailed explanations and adapt feedback to individual learning needs. For instance, GPT-based systems can rival or exceed the explanatory capabilities of students or teaching assistants in specific domains. Balse et al. (2023) found that GPT-3.5's explanations for programming errors matched the quality of TA-generated feedback. Similarly, Leinonen et al. (2023) showed that GPT-3 produced clearer code explanations than students. However, other studies reveal limitations in factual accuracy and pedagogical depth: Prihar et al. (2023) observed that GPT-3 explanations for middle school math problems fell short of those written by teachers, reinforcing concerns about perpetuating misconceptions (Kunz and Kuhlmann 2024). To improve interactivity, recent systems adopt dialogic strategies such as Socratic prompting and recontextualization to align explanations with student interests (Shridhar et al. 2022; Yadav, Tseng, and Ni 2023). AI-powered chatbots like EduChat (Dan et al. 2023) and the Taiwan Adaptive Learning Platform (Kuo, Chang, and Bai 2023) integrate real-time feedback and dynamic questioning, enhancing student engagement through continual adaptation.

Educator-Centered Support and Content Generation. Educator-facing tools have focused on reducing instructional overhead by automating tasks such as quiz creation, grading, and feedback generation (Alsafari et al. 2024; Kasneci et al. 2023). For example, CyberQ (Agrawal et al. 2024) uses LLMs with knowledge graph augmentation to generate targeted cybersecurity assessments. Other systems demonstrate that LLM-generated quiz questions can rival those written by instructors in clarity and pedagogical quality (Elkins et al. 2024; Doughty et al. 2024; Xiao et al. 2023). Automated feedback systems further show that LLMs can produce high-quality evaluative comments across disciplines (Xiao et al. 2024; Li et al. 2024).

Automated Slide Development

Automating slide generation has gained significant interest, particularly for scientific and technical presentations. Early work in this area focused on extractive methods that identify and rank important sentences from documents to form slide content. For example, Sefid et al. (2019) proposed a method based on the SummaRuNNer model (Nallapati, Zhai, and Zhou 2017), which uses a windowed labeling ranking system to combine semantic and lexical features within a sentence window, measuring the importance and novelty of sentences for slide construction.

Other researchers explored alternative approaches to identify relevant information. Hu and Wan (2015) developed PPSGen, a framework that uses Support Vector Regressors and Integer Linear Programming (ILP) to rank and select key sentences. Wang, Wan, and Du (2017) proposed a phrase-based approach that extracts key phrases and learns hierarchical relationships to structure bullet points for slides. Over time, these strategies evolved to incorporate deep

[1]https://www.latex-project.org

[2]https://ctan.org/pkg/beamer

learning methods. For example, Sefid et al. (2021) extended their earlier work by incorporating contextual information and deep neural network approaches to enhance sentence scoring and summary construction. Their approach combines feature-based and neural methods to improve coherence and relevance.

Modern advancements in this field focus on interactive and real-time solutions using up-to-date information. For example, to address the challenge of handling longer documents, Gupta (2023) investigated the use of LLMs with extended token limits, such as Longformer-Encoder-Decoder (LED) (Beltagy, Peters, and Cohan 2020) and BIGBIRD-Pegasus (Zaheer et al. 2020). These models process full-length scientific papers and generate cohesive section-slide pairs. Microsoft Copilot (Microsoft 2023) integrates the LLM capabilities directly into PowerPoint to assist users with slide creation by generating slide outlines, suggested text, and visual content based on user prompts and document context. This approach highlights a trend toward more flexible user-driven slide generation that automates many aspects of presentation design.

Agentic and Instructor-Aligned Generation Frameworks

Recent advances in AI have shifted from single-prompt generation to structured, multi-agent pipelines in which distinct components collaborate to solve complex tasks. This agentic paradigm enables systems to decompose tasks such as code generation (Yang et al. 2024), tool use (Schick et al. 2023), and long-context reasoning (Lu et al. 2023) into subtasks handled by specialized agents. In education, such modularity can allow for a closer alignment with instructional workflows, supporting features like structured planning, instructor comments, and multimodal content insertion.

Among these agents, retrieval modules often leverage techniques from Retrieval-Augmented Generation (RAG) (Lewis et al. 2020), which grounds model responses in external documents to improve factual accuracy and reduce hallucinations. RAG has proven effective in domains requiring up-to-date, trustworthy information, including biomedical QA (Jin, Leaman, and Lu 2023; Lála et al. 2023) and literature synthesis (Jiang et al. 2023; Mialon et al. 2023).

Our proposed SlideBot builds on this paradigm by coordinating a set of specialized components under a central Moderator to handle retrieval, structured slide planning, and presentation enhancement. This modular design enables iterative refinement, ensuring that outputs remain grounded in domain knowledge while following evidence-based instructional principles to support learning.

3 Methodology

SlideBot is built around three key pillars to support effective university-level content creation:

- **Informativeness**: Combines delivery clarity, factual accuracy, and breadth of domain-relevant subtopics to provide a coherent narrative that supports student learning.
- **Reliability**: Presents factually-accurate content grounded in credible sources, maintaining consistent results across each generated presentation.
- **Practicality**: Readily usable and adaptable in real teaching environments, providing clear formatting, multimodal support, and instructor-facing features.

To fulfill these pillars, SlideBot employs a multimodal, multi-agent framework that decomposes educational slide generation into three stages: Content Retrieval, Slide Draft Generation, and Presentation Enhancement. Each stage is handled by specialized agents coordinated by a central Moderator, as shown in Figure 1.

Content Retrieval

The Content Retrieval stage is responsible for gathering relevant, domain-specific information to ground the presentation in accurate and informative content using Retrieval-Augmented Generation (RAG) (Lewis et al. 2020). RAG combines generative capabilities of LLMs with external retrieval mechanisms to reduce hallucinations and enhance factual grounding (Béchard and Ayala 2024). Our system implements a modular corpus interface, allowing retrieval from interchangeable knowledge sources, such as research papers or textbooks. This ensures adaptability across disciplines, supporting both practicality and reliability.

Once a target corpus is selected, the Retriever agent constructs a query to the appropriate source. For academic literature, we use the arXiv API[3] to identify relevant scholarly articles using keyword matching. For structured resources such as textbooks, we leverage BM-25 (Robertson, Zaragoza et al. 2009), a probabilistic ranking algorithm that prioritizes documents where uncommon query terms appear frequently. This allows instructors to incorporate cutting-edge research and curated textbook materials into their presentations.

After retrieving relevant documents or passages, the Retriever constructs a detailed summary and finds source metadata for citation (e.g., title, authors, publication date). These summaries are passed back to the Moderator, which may select a subset of the content for inclusion into the slides. This filtering step ensures that only high-quality, relevant material is discussed within the presentation.

Slide Draft Generation

Once the Moderator receives the summaries from the retrieval stage, it constructs a detailed slide plan to guide the generation of content. This plan follows a predefined structural guide that provides a flexible outline while embedding evidence-based instructional principles from cognitive load theory (CLT) and the cognitive theory of multimedia learning (CTML). The guide begins with a high-level introduction to manage intrinsic load by establishing core concepts, progresses into deeper exploration organized by either individual papers or conceptual subtopics to support schema construction, and concludes with key takeaways that reinforce germane load. Additional structural elements such as a title slide, table of contents, and references are also included to ensure presentation flow and reduce extraneous

[3]https://info.arxiv.org/help/api/index.html

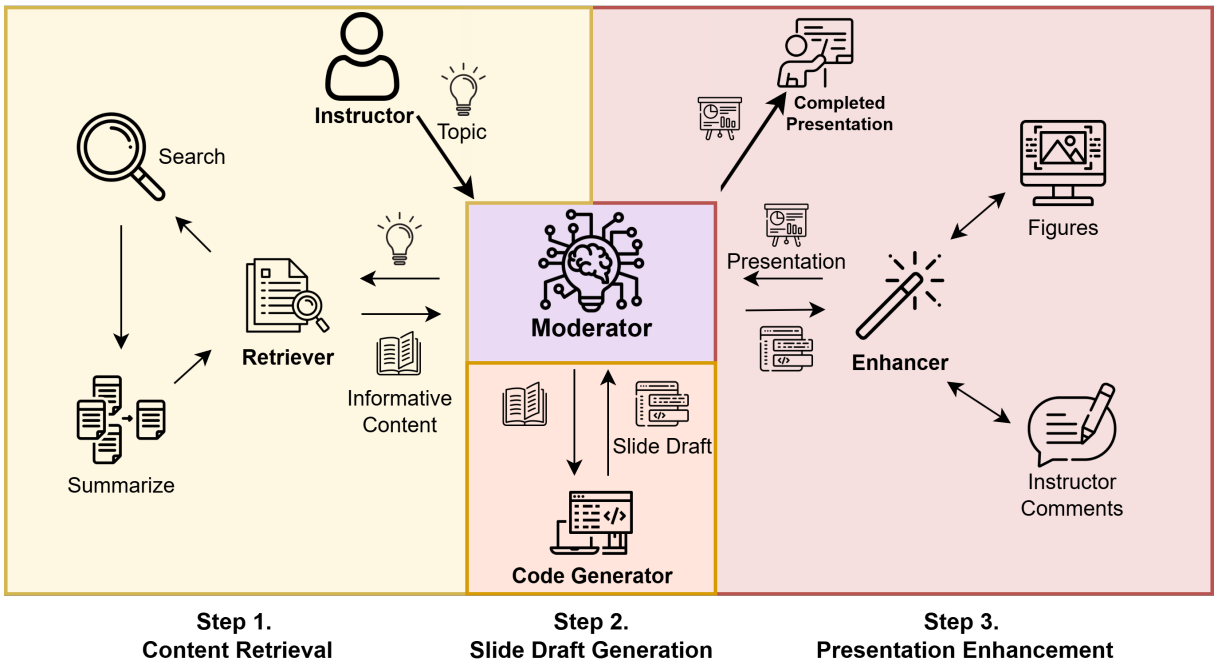

Figure 1: SlideBot's slide generation pipeline operates in three stages: Content Retrieval, where the Moderator receives a topic from the Instructor and communicates with the Retriever to gather and summarize relevant information from a user-selected or automatically designated corpus; Slide Draft Generation, where the Moderator constructs a structured slide plan and the Code Generator translates it into LaTeXBeamer code; and Presentation Enhancement, where the Enhancer inserts figures and instructional comments before returning a completed presentation to the instructor. The Moderator coordinates all agents and manages feedback loops to ensure quality, adaptability, and consistency.

load. The slide plan also incorporates CTML principles: signaling is implemented by using bolded key terms and bullet hierarchies to direct attention to main ideas, coherence is maintained by limiting each slide to concise, directly relevant points with supporting visuals, and spatial contiguity is achieved by pairing explanatory text with corresponding diagrams. Using this structural guide, the Moderator constructs a slide plan that specifies individual slide headers, key facts or explanations to present on each slide, and corresponding citations.

After the slide plan is completed, it is passed to the Code Generator agent. Due to the complexity of direct multimodal generation, where language models often struggle to simultaneously produce coherent text, layout, and visual structure, we instead leverage modern LLMs' strengths in code generation (Jiang et al. 2024). The Code Generator translates the structured blueprint into LaTeXcode that creates structured presentations when compiled. LaTeXis widely adopted within academia for its precision, flexibility, and broad collection of packages. We use the Beamer class, a LaTeXpackage specialized for slide presentations to structure the output, ensuring consistent formatting, support for mathematical notation, citations, and the integration of dynamic visual elements. After the code is compiled and validated by the Moderator, the code is returned to the Code Generator if a compilation error occurs, along with the error message and suggested revisions. This iterative loop minimizes hallucinations in code generation and ensures reliable, executable slide output.

Presentation Enhancement

During the final stage of the pipeline, the Moderator identifies candidate slides where visual elements would enhance signaling or coherence, such as diagrams, charts, or architectural overviews, to be inserted by the Enhancer agent. To ensure consistency, the Enhancer uses prewritten figure macros, streamlining the process so that generating figures requires only supplying the appropriate parameters. The Enhancer also inserts instructor-view-only comments that provide instructional guidance, such as elaboration prompts, warnings about potential misconceptions, or suggestions for visual aids. These enhancements support both practicality, by aligning slides with classroom needs, and reliability, ensuring consistent figure generation.

Once enhancement is complete, the finished slide deck is returned to the user. If desired, the user can send revision requests back to the Moderator, which coordinates with the appropriate agents to implement modifications. This iterative loop allows the slides to be refined multiple times, ensuring alignment with the instructor's needs.

SlideBot's modular design enables straightforward extensions, such as integrating additional specialized agents, expanding the library of figure macros, or incorporating new retrieval corpora to broaden domain coverage. Through its modular and iterative design, not only does this framework

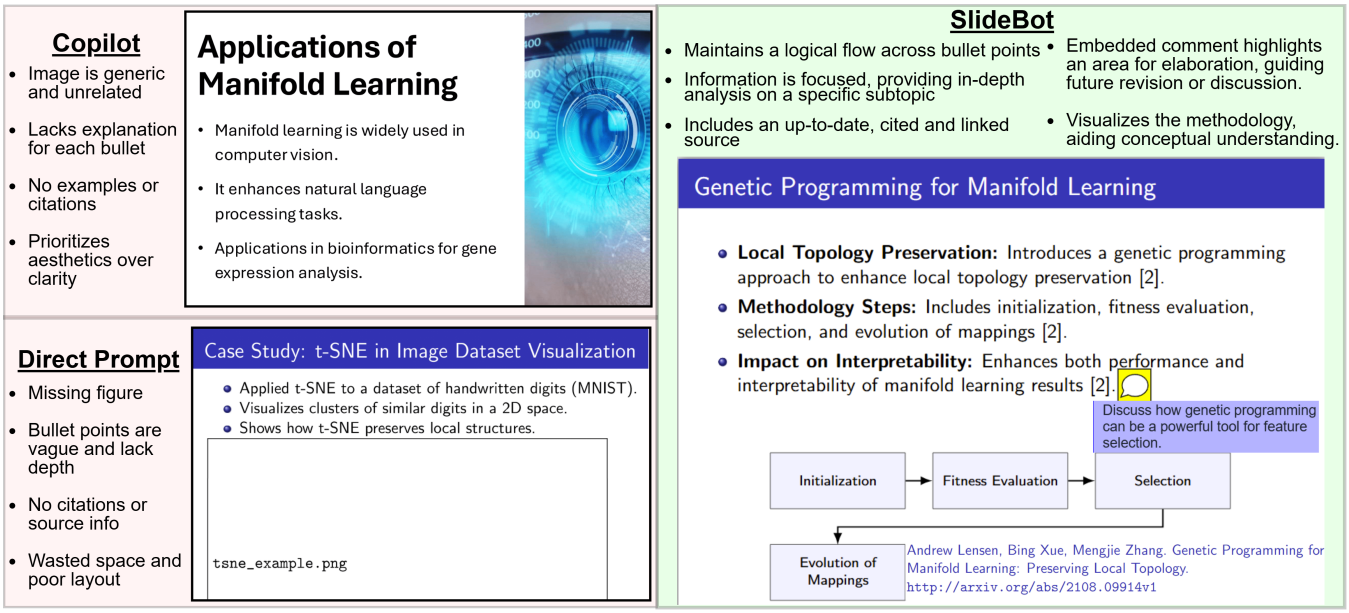

Figure 2: Qualitative comparison of slides generated by Copilot and GPT-4o Direct Prompting (left), and SlideBot (right) on the topic "Manifold Learning." Copilot and Direct Prompt outputs lack explanatory depth, meaningful visuals, and relevant citations. In contrast, SlideBot produces focused, grounded, and pedagogically useful content by retrieving information from Lensen, Xue, and Zhang (2021) via the Retriever agent, adding figures and instructor comments via the Enhancer, and compiling structured LaTeX Beamer slides through the Code Generator.

deliver accurate, pedagogically grounded content, SlideBot also remains adaptable, ready to incorporate new tools, agents, and resources as instructional needs evolve.

4 Experiments

Experimental Settings

To evaluate SlideBot's effectiveness, we conducted two studies in distinct domains: computer science and biomedical education. Each study employed the same dual-survey evaluation framework to assess presentation quality from both learner and instructor perspectives.

Survey Design and Evaluation Metrics. We designed two complementary surveys to capture feedback from both learners and instructors. Our general student survey targeted university students across disciplines to evaluate presentation quality from a learner's perspective. It focused on surface-level aspects such as clarity, structure, and overall appeal, factors independent of domain knowledge. We surveyed 15 participants for the computer science student survey and 13 participants for the biomedicine student survey. Our expert survey was administered to domain-specific experts (graduate students and professors from the field) to reflect an instructor's perspective, emphasizing the depth, accuracy, and instructional usefulness of the content. We surveyed 7 experts in computer science and 4 in biomedicine.

Computer Science Study. We curated a diverse set of undergraduate- to graduate-level AI-related topics, including manifold learning, attention mechanisms, and graph neural networks. To isolate the pipeline's contributions from model capacity, we implemented our pipeline using GPT-4o-mini and compared it to GitHub Copilot, a state-of-the-art AI assistant linked directly to Microsoft PowerPoint.

Biomedical Study. We applied the pipeline to biomedical education by retrieving from a curated textbook corpus (Jin et al. 2021), reflecting material used regularly in practice. We conducted two studies: (1) evaluating the impact of retrieved content on informativeness, and (2) varying model size to compare the effects of model scaling. In these experiments, we compare SlideBot to a Direct Prompt baseline, where the model receives a single instruction to "Generate a graduate-level presentation on [topic]" and produces slides without retrieval, planning, or specialized agents. Direct prompting reflects how AI models are typically used in practice, relying on the model's general knowledge without further customization.

To assess system performance, we structured evaluation metrics around three core pillars:

- **Informativeness**: measured through explanation style (student), conceptual accuracy (expert), and topic coverage (expert), reflecting how clearly and accurately the material conveys a broad range of key ideas.
- **Reliability**: assessed via credibility (student) and variability in suitability across topics (student), capturing trustworthiness and output consistency.
- **Practicality**: evaluated through structure and flow (student), overall suitability (student), and instructor utility (expert), measuring real-world usability and alignment with teaching needs.

Pillar	Metric	Copilot	SlideBot (Ours)	Δ
Informativeness	Explanation Style	2.24	**3.96**	+1.71
	Conceptual Accuracy	2.71	**3.57**	+0.86
	Topic Coverage	2.67	**4.10**	+1.43
Reliability	Credibility	1.93	**4.36**	+2.42
	Variability*	1.47	**0.33**	–1.14
Practicality	Structure & Flow	3.42	**4.04**	+0.62
	Overall Suitability	2.09	**3.67**	+1.58
	Instructor Utility	2.14	**3.76**	+1.62

Table 1: Comparison of SlideBot and Copilot across three evaluation pillars: *Informativeness*, *Reliability*, and *Practicality*. Metrics are derived from both a student survey (unshaded rows) and an expert evaluation (shaded rows). All scores are averaged on a 1–5 scale. **Δ** displays the difference between SlideBot's and Copilot's score for each metric. *The "Variability" metric captures the difference in average "Overall Suitability" between the best- and worst-rated presentations for each method - lower values are desirable, indicating more consistent outputs.

In each survey, participants reviewed three presentations per generation method and scored each metric on a scale of 1-5. This dual perspective evaluates both the quality and adaptability of generated presentations across multiple domains, model sizes, and pipeline configurations.

Qualitative Analysis

To ground these evaluations in a concrete example, Figure 2 presents a comparison of slides generated by Copilot, GPT-4o-mini with a direct prompt (single instruction, no structured assistance), and our full pipeline (SlideBot). The Copilot output violates several evidence-based instructional design principles. For example, irrelevant visuals and absent citations hinder both the coherence principle (integrating only relevant information) and the signaling principle (highlighting key concepts). Vague bullet points impose unnecessary extraneous load under Cognitive Load Theory (CLT) by forcing learners to infer missing context, while poor spatial alignment between text and visuals violates the spatial contiguity principle, reducing integration between modalities. Generation from a direct prompt results in similar issues, though its failures extend beyond instructional design principles. Without agent-based coordination, functional errors begin to occur, such as missing images or improperly formatted elements that extend past slide boundaries.

In contrast, SlideBot's output applies CTML and CLT principles more effectively. Focused bullet points with citations reduce extraneous load and support reliability. Visuals are directly tied to the described methodology, fulfilling the coherence and multimedia principles and aiding germane load through schema construction. Embedded instructor comments act as signaling devices, providing strategies to direct learner attention toward key areas for elaboration or deeper reasoning. The logical flow across slides aligns with segmenting and pre-training principles, easing the transition from introductory material to deeper exploration. These qualitative improvements reflect our three design pillars - informativeness, reliability, and practicality - and illustrate the qualitative improvements that underpin the quantitative gains reported in the following section.

Comparison with Copilot

Table 1 summarizes average scores across all metrics split into their respective pillars, drawn from both the student and expert surveys.

SlideBot outperforms Copilot in all metrics related to informativeness. Explanation style, rated by students, shows the largest improvement with a gain of +1.71. Conceptual accuracy also increases substantially (+0.86), which is particularly noteworthy in an educational setting, as it reflects not only the correctness of individual facts but also the alignment of presented material with accepted domain knowledge and learning objectives. Inaccuracies or misconceptions from hallucinations can significantly hinder understanding and lead to persistent misunderstandings, an issue SlideBot helps address. SlideBot also scores higher in Topic Coverage, reflecting the inclusion of a broader range of subtopics. While coverage alone is often a matter of preference, in combination with strong conceptual accuracy and related measures, it indicates that SlideBot can effectively address a wide scope of material.

For reliability, the credibility of SlideBot's slides was rated +2.42 higher than Copilot's. Additionally, our system achieved a lower variability score (–1.14), suggesting more consistent presentation quality across different topics. This is attributed to SlideBot's structured control over content planning, grounding in verifiable sources, and code validation loop, all of which help mitigate hallucinations and preserve factual consistency.

In terms of practicality, slides produced by our system were consistently viewed as more usable in real-world teaching settings. Improvements were observed in structure and flow (+0.62), overall suitability (+1.58), and instructor utility (+1.62). Instructor comments, consistent formatting, and figure support contributed to these gains, helping instructors better understand, adapt, and present the material.

Taken together, these results demonstrate that our pipeline substantially improves presentation quality across all dimensions, highlighting the value of a modular, retrieval-augmented, and planning-driven approach.

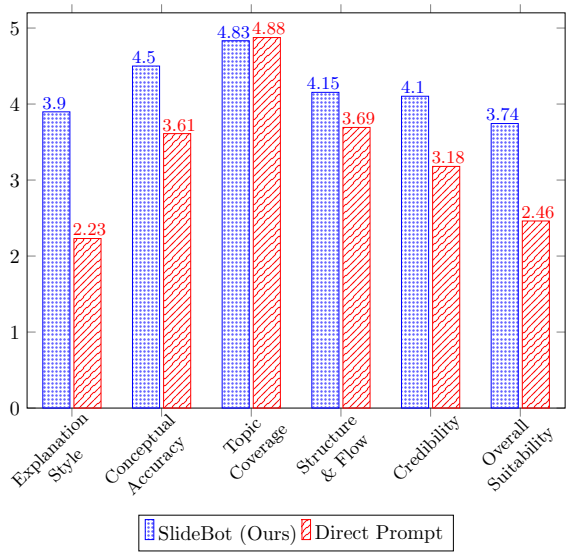

Figure 3: Comparison of SlideBot and Direct Prompt generation (both using GPT-4o) across six presentation quality metrics, with Explanation Style, Structure & Flow, Credibility, and Overall Suitability obtained from a student survey, and Conceptual Accuracy and Topic Coverage obtained from an expert survey.

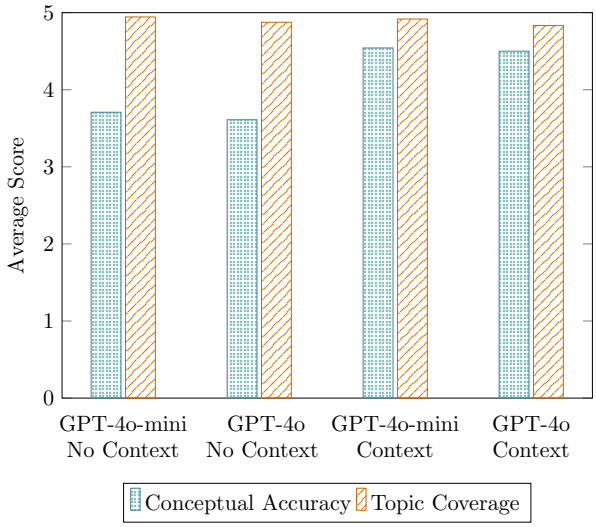

Figure 4: Comparison of informativeness metrics across model and context combinations obtained from an expert survey. Results for GPT-4o models (with and without context) are repeated from Figure 3 to enable side-by-side comparison with GPT-4o-mini variants, renamed for clarity.

Impact of Pipeline Guidance

To isolate the impact of our full architecture on *Practicality* and *Reliability*, we compare SlideBot's output to a Direct Prompt baseline slide generation that relies on the parametric capabilities of the language model. In this baseline, slides are generated in a single prompt without retrieval, structured planning, or enhancement.

As shown in Figure 3, SlideBot significantly outperforms the Direct Prompt baseline across nearly all metrics. Explanation style improves by +1.67 points, while conceptual accuracy and credibility increase by +0.89 and +0.92, respectively. Direct Prompt achieves a similarly high Topic Coverage score, suggesting that the choice of base model may have a greater influence on topic breadth than the framework. The gap in Structure & Flow is slightly narrower; however, without additional guidance, the Direct Prompt content still lacks utility as reflected in the +1.28 advantage for SlideBot in Overall Suitability. This contrast highlights that decomposing the task across specialized agents results in strong performance across all dimensions.

Impact of Model Size

To assess whether our approach provides substantial benefits compared to model scaling, we compare GPT-4o and GPT-4o-mini with and without retrieved context, with data gathered from an expert survey. As shown in Figure 4, Topic Coverage remains consistent across all settings. Clear distinctions emerge in Conceptual Accuracy: GPT-4o-mini improves from 3.71 to 4.54 (+0.83), and GPT-4o from 3.61 to 4.50 (+0.89) when using a Retrieval agent. Interestingly, GPT-4o-mini marginally outperforms GPT-4o. We observe that because the smaller model has more limited parametric knowledge, it relies more heavily on the retrieved content and thus adheres more closely to it, whereas the larger model can default to its internal knowledge, which may be less explanatory or reliable.

Overall, the results reinforce that SlideBot's architectural design has a significantly larger impact than the size of the base model in driving conceptual understanding and content quality, indicating a cost-effective improvement measure.

5 Conclusion

While LLMs show promise in education, they often fall short in generating structured, reliable, and domain-specific presentation materials. To address this, we introduce SlideBot, a modular, agent-based slide generation framework with multimodal capabilites, that emphasizes informativeness, reliability, and practicality through the integration of Cognitive Load Theory and Cognitive Theory of Multimedia Learning principles. By combining retrieval, planning, and LaTeX-based formatting, SlideBot produces high-quality, customizable slides aligned with instructor needs in any subject area. Empirical results show significant improvements in informativeness, reliability, and practicality compared to Microsoft's Copilot, with benefits that stem from its adaptable, modular design rather than relying solely on larger models. As educational technology continues to evolve, our approach serves as a promising step toward creating impactful, reliable, AI-driven teaching tools.

Acknowledgements

This work is supported in part by the US Department of Education under grant H327S240013. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Department of Education.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agrawal, G.; Pal, K.; Deng, Y.; Liu, H.; and Chen, Y.-C. 2024. CyberQ: Generating Questions and Answers for Cybersecurity Education Using Knowledge Graph-Augmented LLMs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21).

Ahmad, M. A.; Yaramis, I.; and Roy, T. D. 2023. Creating trustworthy llms: Dealing with hallucinations in healthcare ai. *arXiv preprint arXiv:2311.01463*.

Alley, M.; and Neeley, K. A. 2005. Rethinking the design of presentation slides: A case for sentence headlines and visual evidence. *Technical communication*, 52(4): 417–426.

Alsafari, B.; Atwell, E.; Walker, A.; and Callaghan, M. 2024. Towards effective teaching assistants: From intent-based chatbots to LLM-powered teaching assistants. *Natural Language Processing Journal*, 1: 100101.

Balse, R.; Kumar, V.; Prasad, P.; and Warriem, J. M. 2023. Evaluating the Quality of LLM-Generated Explanations for Logical Errors in CS1 Student Programs. In *Proceedings of the 16th Annual ACM India Compute Conference*, 49–54.

Bartsch, R. A.; and Cobern, K. M. 2003. Effectiveness of PowerPoint presentations in lectures. *Computers & education*, 41(1): 77–86.

Béchard, P.; and Ayala, O. M. 2024. Reducing hallucination in structured outputs via Retrieval-Augmented Generation. *arXiv preprint arXiv:2404.08189*.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.

Chandler, P.; and Sweller, J. 1991. Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4): 293–332.

Cremin, C. J.; Dash, S.; and Huang, X. 2022. Big data: historic advances and emerging trends in biomedical research. *Current Research in Biotechnology*, 4: 138–151.

Dan, Y.; Lei, Z.; Gu, Y.; Li, Y.; Yin, J.; Lin, J.; Ye, L.; Tie, Z.; Zhou, Y.; Wang, Y.; et al. 2023. EduChat: A Large-Scale Language Model-Based Chatbot System for Intelligent Education. *arXiv preprint arXiv:2308.02773*.

Doughty, J.; Wan, Z.; Bompelli, A.; Qayum, J.; Wang, T.; Zhang, J.; Zheng, Y.; Doyle, A.; Sridhar, P.; Agarwal, A.; et al. 2024. A comparative study of AI-generated (GPT-4) and human-crafted MCQs in programming education. In *Proceedings of the 26th Australasian Computing Education Conference*, 114–123.

Elkins, S.; Kochmar, E.; Cheung, J. C.; and Serban, I. 2024. How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23084–23091.

Fagbohun, O.; Iduwe, N.; Abdullahi, M.; Ifaturoti, A.; and Nwanna, O. 2024. Beyond traditional assessment: Exploring the impact of large language models on grading practices. *Journal of Artifical Intelligence and Machine Learning & Data Science*, 2(1): 1–8.

Flier, J. S. 2023. Publishing Biomedical Research: a rapidly evolving ecosystem. *Perspectives in Biology and Medicine*, 66(3): 358–382.

Gupta, T. 2023. Automatic Presentation Slide Generation Using LLMs.

Hu, Y.; and Wan, X. 2015. PPSGen: Learning-Based Presentation Slides Generation for Academic Papers. *IEEE Transactions on Knowledge and Data Engineering*, 27(4): 1085–1097.

Jiang, J.; Wang, F.; Shen, J.; Kim, S.; and Kim, S. 2024. A Survey on Large Language Models for Code Generation. *arXiv preprint arXiv:2406.00515*.

Jiang, Z.; Xu, F. F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Jin, D.; Pan, E.; Oufattole, N.; Weng, W.-H.; Fang, H.; and Szolovits, P. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14): 6421.

Jin, Q.; Leaman, R.; and Lu, Z. 2023. Retrieve, Summarize, and Verify: How will ChatGPT impact information seeking from the medical literature? *Journal of the American Society of Nephrology*, 10–1681.

Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; and Kasneci, G. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103: 102274.

Kunz, J.; and Kuhlmann, M. 2024. Properties and Challenges of LLM-Generated Explanations. *arXiv preprint arXiv:2402.10532*.

Kuo, B.-C.; Chang, F. T.; and Bai, Z.-E. 2023. Leveraging LLMs for Adaptive Testing and Learning in Taiwan Adaptive Learning Platform (TALP). In *LLM@ AIED*, 101–110.

Lála, J.; O'Donoghue, O.; Shtedritski, A.; Cox, S.; Rodriques, S. G.; and White, A. D. 2023. PaperQA: Retrieval-Augmented Generative Agent for Scientific Research. *arXiv preprint arXiv:2312.07559*.

Leinonen, J.; Denny, P.; MacNeil, S.; Sarsa, S.; Bernstein, S.; Kim, J.; Tran, A.; and Hellas, A. 2023. Comparing code

explanations created by students and large language models. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 124–130.

Lensen, A.; Xue, B.; and Zhang, M. 2021. Genetic programming for manifold learning: Preserving local topology. *IEEE Transactions on Evolutionary Computation*, 26(4): 661–675.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, H.; Li, C.; Xing, W.; Baral, S.; and Heffernan, N. 2024. Automated Feedback for Student Math Responses Based on Multi-Modality and Fine-Tuning. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*, 763–770.

Lu, P.; Peng, B.; Cheng, H.; Galley, M.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; and Gao, J. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36: 43447–43478.

Mayer, R. E. 2002. Multimedia learning. In *Psychology of learning and motivation*, volume 41, 85–139. Elsevier.

Mayer, R. E. 2005a. Cognitive theory of multimedia learning. *The Cambridge handbook of multimedia learning*, 41(1): 31–48.

Mayer, R. E. 2005b. Introduction to multimedia learning. *The Cambridge handbook of multimedia learning*, 2(1): 24.

Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Rozière, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Microsoft. 2023. Introducing Microsoft 365 Copilot—Your copilot for work. https://www.microsoft.com. Accessed: 2023-03-16.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Paas, F.; and Sweller, J. 2014. 2 Implications of Cognitive Load Theory for Multimedia Learning. *The Cambridge Handbook of Multimedia Learning*, 27.

Prihar, E.; Lee, M.; Hopman, M.; Kalai, A. T.; Vempala, S.; Wang, A.; Wickline, G.; Murray, A.; and Heffernan, N. 2023. Comparing different approaches to generating mathematics explanations using large language models. In *International Conference on Artificial Intelligence in Education*, 290–295. Springer.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36: 68539–68551.

Sefid, A.; Wu, J.; Mitra, P.; and Giles, C. L. 2019. Automatic Slide Generation for Scientific Papers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sefid, A.; Wu, J.; Mitra, P.; and Giles, L. 2021. Extractive Research Slide Generation Using Windowed Labeling Ranking. In *Proceedings of the Second Workshop on Scholarly Document Processing*. NAACL.

Shridhar, K.; Macina, J.; El-Assady, M.; Sinha, T.; Kapur, M.; and Sachan, M. 2022. Automatic generation of socratic subquestions for teaching math word problems. *arXiv preprint arXiv:2211.12835*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, S.; Wan, X.; and Du, S. 2017. Phrase-Based Presentation Slides Generation for Academic Papers. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, S.; Xu, T.; Li, H.; Zhang, C.; Liang, J.; Tang, J.; Yu, P. S.; and Wen, Q. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Xiao, C.; Ma, W.; Xu, S. X.; Zhang, K.; Wang, Y.; and Fu, Q. 2024. From Automation to Augmentation: Large Language Models Elevating Essay Scoring Landscape. *arXiv preprint arXiv:2401.06431*.

Xiao, C.; Xu, S. X.; Zhang, K.; Wang, Y.; and Xia, L. 2023. Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 610–625.

Yadav, G.; Tseng, Y.-J.; and Ni, X. 2023. Contextualizing problems to student interests at scale in intelligent tutoring system using large language models. *arXiv preprint arXiv:2306.00190*.

Yang, J.; Jimenez, C. E.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K.; and Press, O. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37: 50528–50652.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33: 17283–17297.