DEEPPRESENTER: Environment-Grounded Reflection for Agentic Presentation Generation

Anonymous ACL submission

Abstract

Presentation generation requires deep content research, coherent visual design, and iterative refinement based on observation. However, existing presentation agents often rely on predefined workflows and fixed templates. To address this, we present DEEPPRESENTER, an agentic framework that adapts to diverse user intents, enables effective feedback-driven refinement, and generalizes beyond a scripted pipeline. Specifically, DEEPPRESENTER autonomously plans, renders, and revises intermediate slide artifacts to support long-horizon refinement with environmental observations. Furthermore, rather than relying on self-reflection over internal signals (e.g., reasoning traces), our environment-grounded reflection conditions the generation process on perceptual artifact states (e.g., rendered slides), enabling the system to identify and correct presentation-specific issues during execution. Results on the evaluation set covering diverse presentation-generation scenarios show that DEEPPRESENTER achieves state-of-the-art performance, and the fine-tuned DeepPresenter-9B remains highly competitive at substantially lower cost.

1 Introduction

Presentations are a primary medium for information delivery across education, business, and research. A high-quality presentation combines well-researched content with coherent visual design, enabling audiences to grasp complex ideas efficiently. However, creating such presentations remains time-consuming and skill-demanding, motivating recent work that leverages Multimodal Large Language Models (MLLMs) to automate this task (Liang et al., 2025; Yang et al., 2025b; Zheng et al., 2025).

However, existing presentation agents (Sefid et al., 2021; Xu et al., 2025; Yang et al., 2025b) fall short of meeting these demands. First, they rely on predefined workflows (Zheng et al., 2025) and content-agnostic templates (Cachola et al., 2024), limiting adaptability to varying user intents. This yields text-heavy slides with insufficient research depth and visual designs that fail to resonate with the narrative. Second, introspective reflection over internal signals (e.g., code or reasoning traces) cannot detect post-render defects (Kim et al., 2025; Tang et al., 2025), resulting in overlapping elements, truncated text, and broken layouts.

To address these limitations, we propose DEEPPRESENTER, an agentic framework for presentation generation (Figure 1). Unlike prior methods that decouple content and design via rigid templates, DEEPPRESENTER coordinates two specialized agents through a shared observation space.

Figure 1: Illustration of DEEPPRESENTER. Given a user instruction, the Researcher gathers information and compiles a structured manuscript, while the Presenter transforms it into visual slides. Both agents interact and collaborate with a shared environment, leveraging grounded observations for reflective refinement.

The Researcher autonomously explores and compiles a structured manuscript aligned with the user intent, while the Presenter converts it into visually coherent slides via content-driven design rather than template filling. Crucially, instead of introspective self-reflection over internal signals, DEEPPRESENTER grounds reflection in perceptual artifact states obtained from environmental observation (Figure 2): agents use `inspect` to view rendered manuscripts and slides, and `think` to plan targeted revisions to correct post-render defects.

While our framework achieves strong performance with proprietary models, their high cost motivates a more efficient alternative. We therefore develop DeepPresenter-9B via supervised fine-tuning on curated trajectories (Figure 3). We first construct diverse presentation tasks from PersonaHub (Ge et al., 2024), arXiv, and FinePDFs (Kydlíček et al., 2025), augmented with verifiable constraints. During trajectory synthesis, we mitigate self-verification bias (Stechly et al., 2024) with extrinsic verification: an independent critic evaluates artifacts in isolation and provides reasoning traces that steer targeted refinements, improving the quality of synthesized trajectories.

We evaluate our method on a held-out set of 128 diverse presentation tasks across three dimensions: constraint satisfaction, content quality, and visual style. With proprietary backbones, DEEPPRESENTER achieves an average score of 4.44, surpassing open-source baselines and the commercial system Gamma (4.36). Our specialized agentic design yields richer content and coherent design, while environment-grounded reflection reduces post-render defects by revising against observed perceptual artifact states. DeepPresenter-9B scores 4.19, outperforming all open-source baselines and approaching GPT-5 (4.22) at lower cost.

In summary, our contributions are threefold:

- We propose DEEPPRESENTER, an agentic presentation framework that coordinates Researcher and Presenter agents via a shared observation space, enabling autonomous information research and topic-aware design.
- We introduce environment-grounded reflection that grounds self-correction in perceptual artifact states obtained from post-render observations, reducing defects that are not detectable from internal signals alone.
- Results on the evaluation set covering diverse presentation-generation scenarios show that DEEPPRESENTER achieves state-of-the-art performance, and the distilled DeepPresenter-9B remains highly competitive at substantially lower cost.

2 DEEPPRESENTER

In this section, we present DEEPPRESENTER, a dual-agent framework for presentation generation. We first formulate the task as an interactive agentic process, then describe the Researcher-Presenter collaboration and the environment-grounded reflection mechanism, as illustrated in Figure 2.

2.1 Task Formulation

We formulate presentation generation as an interactive agentic task. Given an instruction $\mathcal{I}$ and an agent environment $\mathcal{E}$ equipped with a tool library $\mathcal{T}$ and a file system $\mathcal{F}$, the system aims to generate a high-quality presentation $\mathcal{P}$. The generation process can be modeled as a multi-step trajectory $\tau = \{(r_1, a_1, o_1), \ldots, (r_T, a_T, o_T)\}$, where at each step t, the agent generates a reasoning trace r_t, selects an action $a_t \in \mathcal{T}$, and receives observation o_t from $\mathcal{E}$. We decompose the trajectory into two sequential phases: $\tau = \tau^R \circ \tau^P$, where τ^R and τ^P denote the Researcher and Presenter trajectories, respectively. The two agents communicate through $\mathcal{F}$, where the Researcher persists a structured manuscript $\mathcal{M}$ and associated assets for the Presenter to consume. Appendix C lists the tools.

2.2 Dual-Agent Collaboration

Presentation generation requires both information research and visual design, which demand different planning and tool use. We split these roles between two specialized agents while sharing the same backbone model.

Researcher Agent Given $\mathcal{I}$, the Researcher autonomously plans its exploration instead of following a predefined workflow. It executes multiple steps during τ^R, invoking tools from $\mathcal{T}$ to retrieve and synthesize supporting materials and to create auxiliary assets as needed. The exploration depth and strategy adapt to user intent: a technical presentation may require surveying related work, while a general-audience talk may prioritize accessible examples and vivid illustrations. Finally, the Researcher compiles slide text and associated assets into a structured markdown manuscript $\mathcal{M}$ organized by narrative flow, and persists it to $\mathcal{F}$.

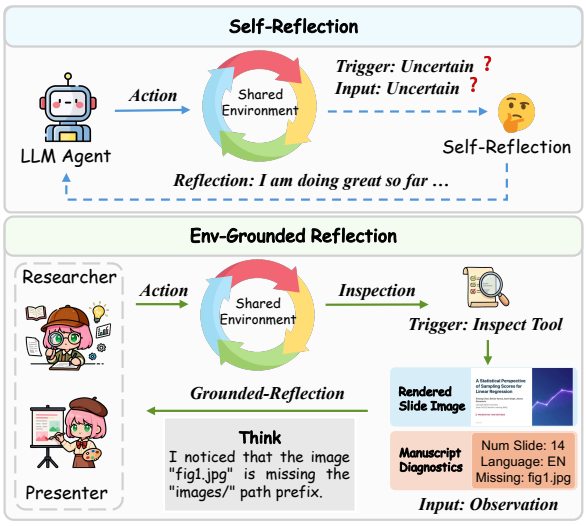


Figure 2: Comparison between self-reflection and environment-grounded reflection. Self-reflection relies on uncertain triggers and inputs without external signals. DEEPPRESENTER grounds reflection in environmental observations through the `inspect` tool.

Dimension	Category	Count	Ratio (%)
Language	English	603	52.34
	Chinese	549	47.66
Source	PersonaHub	586	50.87
	FinePDFs	362	31.42
	arXiv	204	17.71
Aspect Ratio	16:9 Widescreen	327	28.39
	4:3 Standard	304	26.39
	A1 Poster	30	2.60
	Free	491	42.62
Slide Count	11-20	249	21.61
	1-10	320	27.78
	Free	583	50.61
Total		**1,152**	**100.00**

Table 1: Statistics of the constructed presentation tasks by language, source, aspect ratio, and slide count. "Free" indicates no constraint is specified.

Presenter Agent Rather than populating predefined templates, the Presenter generates slides from scratch during τ^P. Given $\mathcal{M}$ from $\mathcal{F}$, the agent first develops a global design plan, establishing color themes and typography that resonate with the topic. It then generates each slide as a standalone HTML file, translating manuscript content into visual elements following the design plan. This content-driven approach enables stylistic choices aligned with the presentation topic, such as earthy palettes for sustainability or minimalist layouts for academic tutorials.

2.3 Environment-Grounded Reflection

We ground agent reflection in environmental observations rather than introspective reasoning over internal signals (He et al., 2025). The key issue with self-reflection is state mismatch: agents operate on intermediate representations (e.g., HTML or markdown), while users perceive only rendered artifacts. As a result, many defects manifest only in perceptual states (e.g., broken images, overflow, or low contrast), leaving introspective reflection operating in a mismatched observation space.

To make perceptual artifact states observable to the agent, we introduce the `inspect` tool as an explicit observation interface. For the Presenter, `inspect` renders an HTML file into image pixels, exposing post-render defects such as overflow, overlap, and low contrast; for the Researcher, `inspect` returns structured diagnostics of the manuscript and file state, including slide count, asset availability, and detected language. Agents then use `think` to reflect on observed defects and plan targeted edits. This forms an observe–reflect–revise loop where agent observations align with user perception.

3 Frontier Presentation Agent Model

This section presents our training pipeline as shown in Figure 3: task dataset construction, trajectory synthesis with extrinsic verification to elicit high-quality reflective behaviors, and multi-stage filtering for quality.

3.1 Query Construction

We construct a task collection for training our compact model and evaluating our framework. To cover diverse presentation scenarios in both intent-driven and document-conditioned settings, we draw task seeds from PersonaHub (Ge et al., 2024), arXiv, and FinePDFs-Edu (Kydlíček et al., 2025). Each task is augmented with verifiable constraints (e.g., slide count, language, aspect ratio) to capture fine-grained user-specified requirements. For PersonaHub, we prompt GLM-4.6 to synthesize presentation tasks conditioned on persona descriptions; for arXiv and FinePDFs-Edu, we construct tasks that require generating presentations based on provided documents. Each task is further augmented with verifiable constraints, including slide count, language, and aspect ratio. In total, this task collection contains 1,152 tasks, with 1,024 for trajectory sampling and 128 held out for evaluation. Detailed statistics are shown in Table 1.

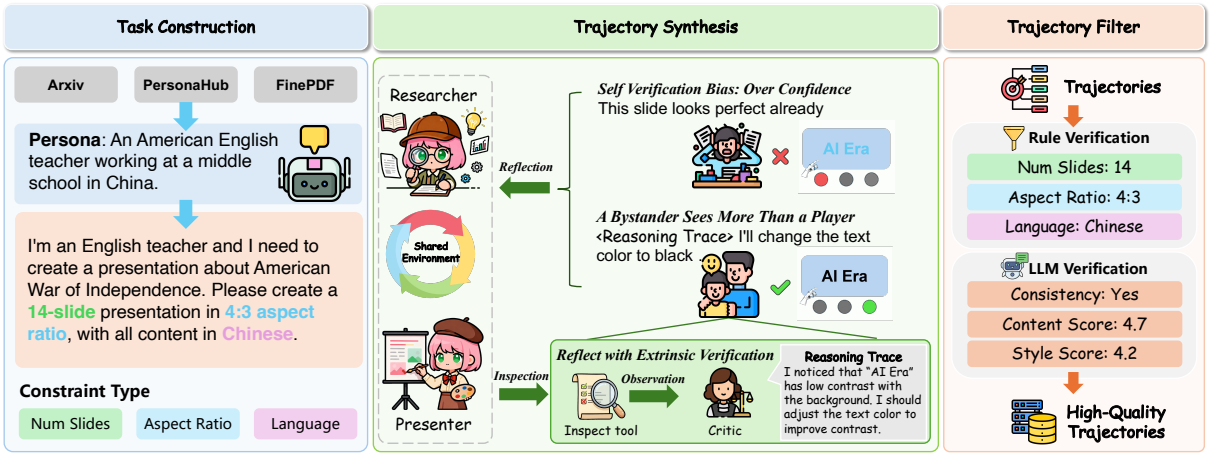


Figure 3: Our data synthesis pipeline. The process ensures high-quality trajectories for supervised fine-tuning through three integrated mechanisms: (1) Query Construction augments tasks with verifiable constraints; (2) Extrinsic Verification injects reasoning traces when defects are identified to guide agent self-correction during sampling; and (3) Trajectory Filtering validates constraint compliance and assesses consistency and output quality.

3.2 Verification-Guided Trajectory Synthesis

When sampling agentic trajectories, self-reflection is susceptible to self-verification bias (Jiang et al., 2025): the agent judges its own intermediate outputs from within the same trajectory state that produced them. This coupling entangles verification with self-justification, resulting in flawed outputs being accepted. To break this coupling, we introduce extrinsic verification, where verification signals are produced in an isolated context.

As illustrated in Figure 3, after the agent invokes `inspect` and obtains an observation o_t, an independent critic performs verification conditioned on o_t and the corresponding intermediate artifacts. The critic outputs a reasoning trace that identifies defects (e.g., low contrast) and specifies actionable adjustments (e.g., adjust text color). We append this trace to the agent context as a `think` call, guiding targeted revisions before continuing the rollout.

3.3 Trajectory Filtering

We adopt a three-stage filtering pipeline to ensure trajectory quality. First, we verify *constraint compliance* through a rule-based system. Second, we evaluate *consistency* using GLM-4.6, removing trajectories that fail to follow the extrinsic-verification trace with aligned revisions (i.e., reflection–action inconsistency). Third, we assess *output quality* using GLM-4.6V, filtering out trajectories with critical defects such as element overlap or broken images.

4 Experiment

In this section, we evaluate our method on presentation generation and analyze our key components.

4.1 Setup

Implementation Details We sample trajectories by running DEEPPRESENTER with Gemini-3-Pro as the backbone and critic model on 1,024 training tasks, with a maximum context window of 50K tokens. 802 trajectories pass our filtering pipeline and are used for supervised fine-tuning. We fine-tune GLM-4.6V-Flash on these trajectories using MS-SWIFT (Zhao et al., 2024), with a batch size of 32 and learning rate of 1e-5 for 3 epochs. Training takes approximately 80 GPU hours on 8 A800-80G GPUs.

Models and Baselines We compare against one commercial system, Gamma[1], and two academic frameworks: PPTAgent (Zheng et al., 2025) and KCTV (Cachola et al., 2024). For backbone models, we evaluate with proprietary GPT-5 (OpenAI, 2025), Gemini-3-Pro (Comanici et al., 2025), and Claude-Sonnet-4.5 (Anthropic, 2025), as well as open-source GLM-4.6 (Zeng et al., 2025a). For DEEPPRESENTER, we additionally evaluate with GLM-4.6V and GLM-4.6V-Flash (Team et al., 2025), as our framework leverages visual feedback through the `inspect` tool.

Evaluation Protocol We hold out 128 tasks from the constructed task collection and evaluate generated presentations using the following metrics:

[1]https://gamma.app/

Framework	Model	Constraint	Content	Style	Avg.	Diversity
Close-sourced Baseline						
Gamma	–	4.93	<u>4.08</u>	4.08	4.36	0.52
Open-sourced Baseline						
PPTAgent	GPT-5	3.96	3.00	4.07	3.68	0.35
	Gemini-3-Pro	4.22	3.09	<u>4.30</u>	3.87	0.19
	Claude-Sonnet-4.5	3.72	2.93	4.15	3.60	0.17
	GLM-4.6	4.02	3.17	4.24	3.81	0.30
KCTV	GPT-5	**4.95**	2.84	3.63	3.81	0.21
	Gemini-3-Pro	4.58	3.01	3.90	3.83	0.27
	Claude-Sonnet-4.5	4.88	2.90	3.99	3.92	0.20
	GLM-4.6	4.66	2.83	3.94	3.81	0.25
Ours						
DEEPPRESENTER	GPT-5	4.80	3.79	4.07	4.22	0.56
	Gemini-3-Pro	4.70	**4.25**	**4.37**	**4.44**	**0.79**
	Claude-Sonnet-4.5	<u>4.90</u>	4.05	4.27	<u>4.41</u>	0.49
	GLM-4.6V	4.69	3.25	3.75	3.90	<u>0.58</u>
	GLM-4.6V-Flash	4.67	3.11	3.69	3.82	0.47
	DeepPresenter-9B	4.77	3.52	4.29	4.19	0.53

Table 2: Performance comparison of different frameworks and models. The best/second-best scores are **bolded**/<u>underlined</u>. Quality metrics (Constraint, Content, Style, Avg.) are scaled to 0–5, while Diversity is scaled to 0–1.

• **Constraint** scores each presentation by the fraction of user-specified constraints it satisfies, covering slide count, language, and aspect ratio, verified through rule-based checking.

• **Content & Style** evaluate the quality of slide content and visual design. We adopt the MLLM-based evaluation framework from Zheng et al. (2025) with GPT-5 as the judge, which has been validated to correlate well with human judgments.

• **Diversity** quantifies visual style variance across generated presentations using the Vendi Score (Friedman and Dieng, 2022), which computes diversity based on the eigenvalue entropy of feature similarity matrices extracted by DINOv2 (Oquab et al., 2023).

We report Avg. as the mean of Constraint, Content, and Style (scaled 0–5), while Diversity (scaled 0–1) measures cross-presentation variation.

4.2 Main Results

Table 2 presents the main experimental results.

DEEPPRESENTER achieves state-of-the-art performance Across all backbone models, DEEPPRESENTER consistently outperforms open-source baselines. With Gemini-3-Pro as the backbone, DEEPPRESENTER attains an average score of 4.44, surpassing the best open-source baseline (KCTV + Claude-Sonnet-4.5, 3.92) by 13.3% and the commercial product Gamma (4.36). The improvements stem from two aspects: (1) *Content quality improves most because Researcher performs intent-adaptive information seeking and synthesis, rather than relying on fixed workflows or user-provided inputs.* Baseline frameworks depend on user-provided materials and lack deep retrieval capability, while our agent searches, retrieves, and synthesizes information from diverse sources. (2) *Style scores improve through content-aware design and environment-grounded reflection.* Our framework enables Presenter to align design decisions with the narrative, while environment-grounded reflection mitigates free-form generation failures by revising against post-render defects.

Free-form generation enables greater visual diversity, with DEEPPRESENTER achieving a diversity score of 0.79. Under our diversity metric, DEEPPRESENTER more than doubles template-based baselines by generating slides in a free-form manner. Baseline frameworks achieve diversity scores of only 0.17 to 0.35, as fixed templates con-

Configuration	Cons.	Content	Style	Avg.
Gemini-3-Pro				
DEEPPRESENTER	4.70	4.25	4.37	4.44
w/o Grounded Reflection	4.52	4.15	4.31	4.32
w/o Dual-Agent	3.94	3.96	4.22	4.04
DeepPresenter-9B				
DeepPresenter-9B	4.77	3.52	4.29	4.19
w/o Grounded Reflection	4.21	3.23	4.01	3.82
w/o Dual-Agent	3.65	2.93	3.11	3.23
w/o Trajectory Filtering	4.67	3.30	4.12	4.03

Table 3: Ablation study on framework components and training strategy. Cons. denotes constraint satisfaction.

Configuration	Cons.	Content	Style	Avg.	Δ
GLM-4.6V-Flash	4.67	3.11	3.69	3.82	–
+ Fine-tuning	4.71	3.19	3.92	3.94	+0.12
+ Extrinsic Verification	4.74	3.28	4.03	4.02	+0.20

Table 4: Effect of extrinsic verification on model performance. Both fine-tuned variants use 300 trajectories. Δ denotes improvement over the base model.

strain visual variation. PPTAgent, in particular, shows lower constraint scores because its style decisions are predetermined by the workflow, limiting task-specific adaptation. Even Gamma, despite its commercial polish, achieves only 0.52. In contrast, our framework maintains high constraint compliance while enabling greater visual diversity (0.79).

DeepPresenter-9B surpasses all open-source baselines with high efficiency. With only 802 trajectories, our compact model achieves an average score of 4.19, outperforming open-source baselines and matching GPT-5 (4.22) at substantially lower cost. These results support the effectiveness of our verification-guided trajectory synthesis and suggest that compact models can acquire agentic behaviors from limited but high-quality samples.

4.3 Ablation Study

We ablate key components of DEEPPRESENTER on Gemini-3-Pro and DeepPresenter-9B, as shown in Table 3. (1) *Environment-grounded reflection is critical because it extends observation space to post-render perceptual artifact states.* Disabling `inspect` confines reflection to pre-render artifacts and degrades performance from 4.44 to 4.04 on Gemini-3-Pro and from 4.19 to 3.23 on DeepPresenter-9B. (2) *Dual-agent collaboration contributes significantly by decomposing long-horizon execution into specialized sub-tasks.* Without it, performance drops substantially on both backbones. (3) *Trajectory filtering effectively prevents biased and low-quality patterns from being distilled during fine-tuning.* Removing it drops DeepPresenter-9B from 4.19 to 4.03.

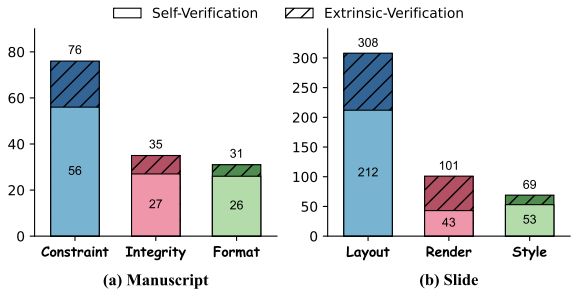


Figure 4: Distribution of defects identified by self-verification and extrinsic verification for manuscripts (left) and slides (right), respectively.

5 Analysis

We analyze the effectiveness of the extrinsic evaluation, examine failure modes in trajectory synthesis, and present efficiency comparisons alongside qualitative case studies.

5.1 Effect of Extrinsic Verification

Extrinsic verification improves trajectory synthesis by mitigating self-verification bias. To quantify its impact, we train two variants on 300 trajectories sampled from the same set of tasks, with and without extrinsic verification during trajectory synthesis. As shown in Table 4, adding extrinsic verification yields a 67% larger gain in Avg. (0.20 vs. 0.12) than fine-tuning alone. This indicates that, even with environment-grounded observations, revision signals produced solely within the agent's own trajectory state can be biased, leading to suboptimal refinements being distilled during learning.

Extrinsic verification mitigates self-verification bias by strengthening defect-triggered revision signals. We categorize reflection-triggered defects into three manuscript types: *integrity* (e.g., missing asset references), *constraint* (e.g., mismatched slide count), and *format* (e.g., invalid markup); and three slide types: *layout* (e.g., overlap), *render* (e.g., blank slides), and *style* (e.g., low contrast). Figure 4 compares defects identified on the same 300 trajectories under self-verification versus extrinsic verification. Extrinsic verification consistently yields more defect detections across categories, with the largest gaps on slides (e.g., 308 vs. 212 for *layout* and 101 vs. 43 for *render*).

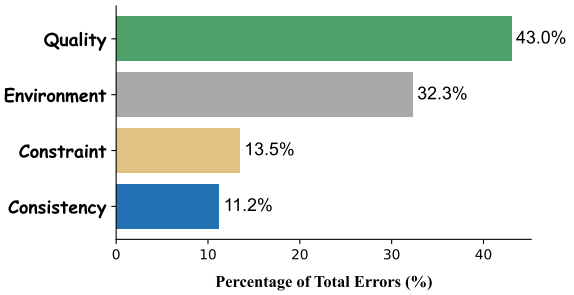


Figure 5: Failure distribution in synthesized trajectories before filtering

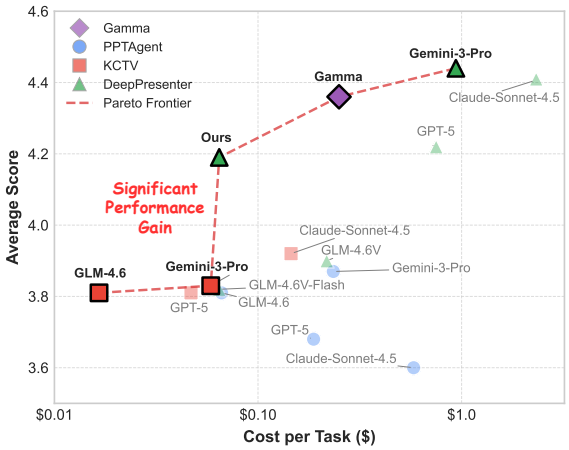


Figure 6: Performance vs. Price scatter plot with Pareto frontier representation. Different colors represent different frameworks

This pattern indicates a systematic failure in self-verification: when verification is conducted within the generating trajectory state, the agent tends to rationalize defects, producing biased judgment (Jiang et al., 2025; Stechly et al., 2024). By decoupling verification from the agent's own trajectory state, extrinsic verification mitigates this bias and provides stronger signals to trigger corrective revisions during synthesis.

5.2 Trajectory Failure Analysis

Following the categories in Section 3.3, we analyze failures in synthesized trajectories before filtering (Figure 5). *Quality* errors are most prevalent (43.0%), underscoring the difficulty of sustaining high standards under free-form generation. *Environment* failures are also common (32.3%), reflecting long-horizon fragility from context overflow and infrastructure disruptions. The remaining cases include *Constraint* violations (13.5%) and *Consistency* errors (11.2%), which are less frequent but still non-negligible.

5.3 Efficiency Analysis

Figure 6 presents the cost-performance trade-off across frameworks and models. (1) *DeepPresenter-9B advances the Pareto frontier, significantly outperforming the prior frontier point at comparable cost.* Compared to KCTV + Gemini-3-Pro (3.83), DeepPresenter-9B achieves 4.19 at a similar price, a significant improvement in cost-quality trade-off. (2) DEEPPRESENTER *establishes a new upper bound for presentation generation, surpassing the previous best system Gamma.* With an average score of 4.44 versus Gamma's 4.36, DEEPPRESENTER delivers the strongest result in our evaluation.

Notably, baseline frameworks exhibit flat performance across backbone models, whereas DEEPPRESENTER demonstrates substantial variation (3.82 to 4.44). This pattern is consistent with baselines being limited by their fixed pipelines, while DEEPPRESENTER can better leverage stronger model capacity.

5.4 Case Study

We present qualitative examples in Figure 7. (1) DEEPPRESENTER *produces visually rich slides through diverse asset sources, while baselines tend to yield text-heavy outputs.* Gamma includes more imagery than academic baselines. However, it relies heavily on AI-generated images and often mishandles figures embedded in source documents (e.g., inappropriate scaling of architectural diagrams). Open-source baselines rarely retrieve or create supporting visuals, resulting in predominantly textual content. (2) DEEPPRESENTER *generates visual themes that resonate with content, whereas baselines rely on fixed templates.* For example, DEEPPRESENTER employs green tones for environmental topics and minimalist layouts for academic presentations, while baseline methods exhibit limited topical alignment due to template-driven generation.

6 Related Work

Presentation generation has attracted increasing attention due to its practical value for information delivery. Before the emergence of large language models, presentation generation was primarily formulated as a document summarization task. These approaches employed extractive summarization to select salient sentences using neural networks (Fu et al., 2022; Hu and Wan, 2014; Sun et al., 2021) or phrase-based methods (Wang et al., 2017). How-

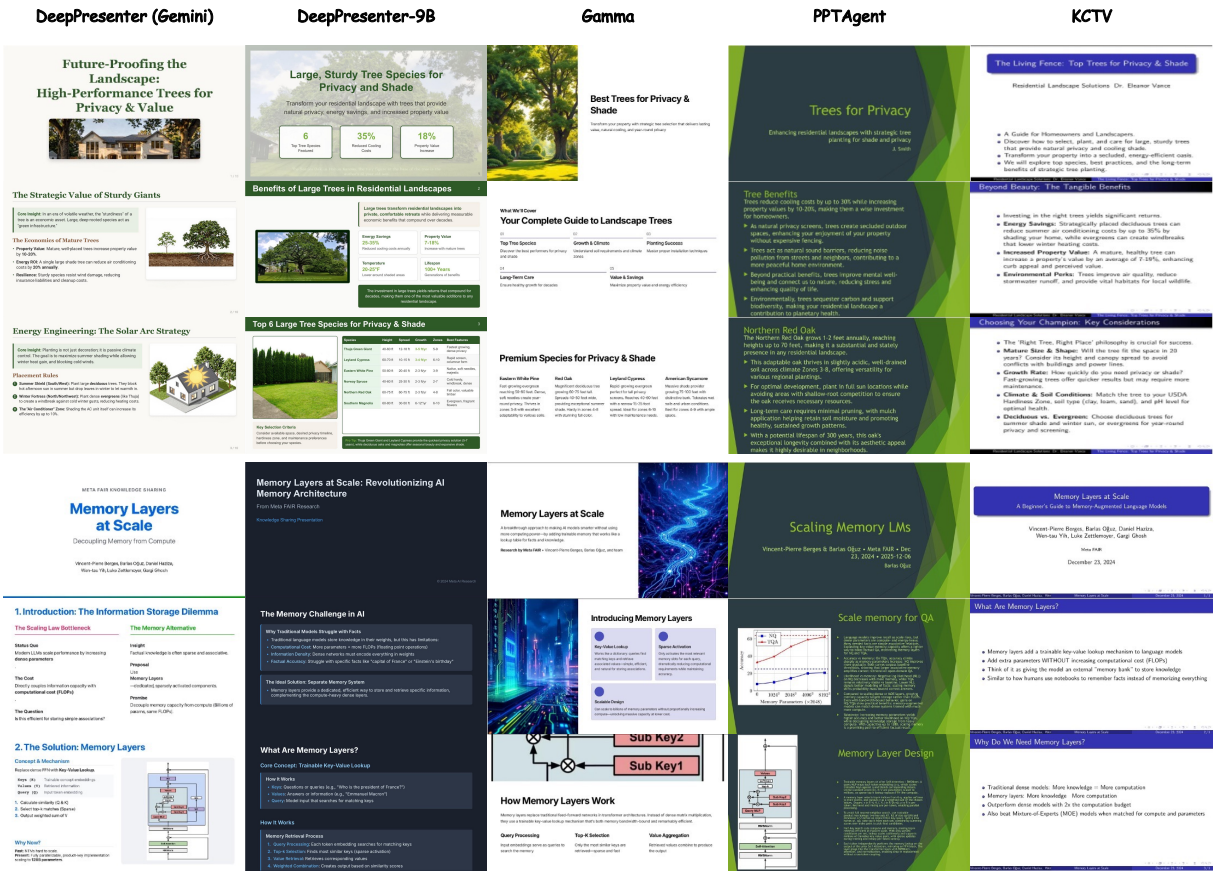


Figure 7: Qualitative comparison of presentations generated by different methods. DEEPPRESENTER under Gemini-3-Pro and DeepPresenter-9B produce high-quality slides with styles that resonate with the topic. Baselines rely on document-embedded or AI-generated images with template-based generation, producing text-heavy outputs and misaligned visual themes.

ever, the limited reasoning capabilities of pre-LLM models constrained their ability to handle diverse user intents and produce visually engaging outputs.

The emergence of LLMs has shifted the paradigm toward agent-based approaches that leverage stronger reasoning and generalization capabilities. Recent work explores multi-agent collaboration for content extraction and layout planning (Cachola et al., 2024; Ge et al., 2025; Liang et al., 2025; Xu et al., 2025; Yang et al., 2025b), aesthetic-aware generation (Liu et al., 2025), as well as slide understanding and editing (Huang et al., 2025; Jung et al., 2025; Zeng et al., 2025b; Zheng et al., 2025). However, these approaches often focus on predefined workflows and fixed templates, limiting adaptation to user intent and iterative refinement with environmental feedback.

Compared with previous methods, DEEPPRESENTER formulates presentation generation as an autonomous exploration and collaboration process between two specialized agents. The Researcher-Presenter decomposition enables adaptive planning based on task complexity, while environment-grounded reflection allows agents to verify and refine artifacts through rendered slides and file system states (Jiang et al., 2025; Stechly et al., 2024; Tang et al., 2025).

7 Conclusion

In this work, we propose DEEPPRESENTER, an agentic framework for presentation generation in which agents plan autonomously and adapt to diverse user intents. Our framework grounds self-reflection in perceptual artifact states from environmental observations, enabling agents to iteratively identify and fix post-render defects. We further train DeepPresenter-9B on trajectories synthesized with extrinsic verification, which mitigates self-verification bias and strengthens reflective behaviors. Results show that DEEPPRESENTER achieves state-of-the-art performance, while DeepPresenter-9B remains competitive at substantially lower cost.

Limitations

While DEEPPRESENTER demonstrates strong performance, several limitations remain. First, DEEPPRESENTER relies on multi-step, tool-using rollouts, which increase inference cost and are sensitive to environment instability (e.g., context overflow and infrastructure failures) observed in our trajectory analysis. Second, extrinsic verification is only used during trajectory synthesis. We do not employ an external critic at inference time, as critic-provided reflection signals can introduce reflection–action inconsistency and additional overhead. Future work can explore mitigating self-verification bias at inference time.

References

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. [Accessed 18-11-2025].

Isabel Alyssa Cachola, Silviu Cucerzan, Allen Herring, Vuksan Mijovic, Erik Oveson, and Sujay Kumar Jauhar. 2024. Knowledge-centric templatic views of documents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15460–15476, Miami, Florida, USA. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. 2022. Doc2ppt: Automatic presentation slides generation from scientific documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):634–642.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and 1 others. 2025. Autopresent: Designing structured visuals from scratch. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2902–2911.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Z.y. Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, and Bo Zheng. 2025. Can large language models detect errors in long chain-of-thought reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18468–18489, Vienna, Austria. Association for Computational Linguistics.

Yue Hu and Xiaojun Wan. 2014. Ppsgen: Learning-based presentation slides generation for academic papers. *IEEE transactions on knowledge and data engineering*, 27(4):1085–1097.

Zheng Huang, Xukai Liu, Tianyu Hu, Kai Zhang, and Ye Liu. 2025. Pptbench: Towards holistic evaluation of large language models for powerpoint layout and design understanding. *arXiv preprint arXiv:2512.02624*.

Dongwei Jiang, Jingyu Zhang, Orion Weller, Nathaniel Weir, Benjamin Van Durme, and Daniel Khashabi. 2025. Self-[in] correct: Llms struggle with discriminating self-generated responses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24266–24275.

Kyudan Jung, Hojun Cho, Jooyeol Yun, Soyoung Yang, Jaehyeok Jang, and Jaegul Choo. 2025. Talk to your slides: Language-driven agents for efficient slide editing. *arXiv preprint arXiv:2505.11604*.

Jeonghye Kim, Sojeong Rhee, Minbeom Kim, Dohyung Kim, Sangmook Lee, Youngchul Sung, and Kyomin Jung. 2025. Reflact: World-grounded decision making in llm agents via goal-state reflection. *arXiv preprint arXiv:2505.15182*.

Hynek Kydlíček, Guilherme Penedo, and Leandro von Werra. 2025. Finepdfs. https://huggingface.co/datasets/HuggingFaceFW/finepdfs_edu.

Xin Liang, Xiang Zhang, Yiwei Xu, Siqi Sun, and Chenyu You. 2025. Slidegen: Collaborative multimodal agents for scientific slide generation. *arXiv preprint arXiv:2512.04529*.

Chengzhi Liu, Yuzhe Yang, Kaiwen Zhou, Zhen Zhang, Yue Fan, Yanan Xie, Peng Qi, and Xin Eric Wang. 2025. Presenting a paper is an art: Self-improvement aesthetic agents for academic presentations. *arXiv preprint arXiv:2510.05571*.

OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/. [Accessed 18-11-2025].

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Athar Sefid, Prasenjit Mitra, and Lee Giles. 2021. Slidegen: an abstractive section-based slide generator for scholarly documents. In *Proceedings of the 21st ACM Symposium on Document Engineering*, pages 1–4.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint arXiv:2402.08115*.

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*.

Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and 1 others. 2025. Realcritic: Towards effectiveness-driven evaluation of language model critiques. *arXiv preprint arXiv:2501.14492*.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 69 others. 2025. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *Preprint*, arXiv:2507.01006.

Sida Wang, Xiaojun Wan, and Shikang Du. 2017. Phrase-based presentation slides generation for academic papers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Xiaojie Xu, Xinli Xu, Sirui Chen, Haoyu Chen, Fan Zhang, and Ying-Cong Chen. 2025. Pregenie: An agentic framework for high-quality visual presentation generation. *arXiv preprint arXiv:2505.21660*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yuheng Yang, Wenjia Jiang, Yang Wang, Yiwei Wang, and Chi Zhang. 2025b. Auto-slides: An interactive multi-agent system for creating and customizing research presentations. *arXiv preprint arXiv:2509.11062*.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025a. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Wenzheng Zeng, Mingyu Ouyang, Langyuan Cui, and Hwee Tou Ng. 2025b. Slidetailor: Personalized presentation slide generation for scientific papers. *arXiv preprint arXiv:2512.20292*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2025. Pptagent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14413–14429.

A Detailed Analysis

A.1 Human Evaluation

To address concerns about potential circularity introduced by LLM-as-judge evaluation, we conduct a small-scale human study to corroborate the automatic assessments. We recruited two graduate students majoring in computer science to evaluate 32 randomly sampled presentations from the test set. Following the evaluation dimensions in Section 4, annotators rate Content and Style on a 1–5 Likert scale using the scoring criteria of Zheng et al. (2025), while Constraint satisfaction is verified via rule-based checks consistent with our evaluation protocol. Evaluators were provided with rendered slide images and scored them independently. Table 5 reports the resulting ratings. Importantly, the relative ranking and overall trends under human judgment align with our automatic evaluation, suggesting that the observed improvements are not an artifact of relying solely on GPT-5 as the judge.

Method	Cons.	Content	Style	Avg.
Gamma	4.84	3.52	3.90	4.09
PPTAgent	3.72	3.07	3.60	3.46
KCTV	4.41	2.84	3.19	3.48
DeepPresenter	4.56	3.86	4.25	4.22

Table 5: Human evaluation results on 32 randomly sampled presentations.

Domain	Cons.	Content	Style	Avg.
PersonaHub	4.38	4.49	4.49	4.45
arXiv	4.91	3.84	4.13	4.29
FinePDF	4.94	4.21	4.38	4.51

Table 6: Domain performance breakdown. Cons. denotes constraint satisfaction.

A.2 Performance by Domain

We analyze DEEPPRESENTER with Gemini-3-Pro across domains. PersonaHub shows the strongest content (4.49) and style (4.49) scores, but relatively lower constraint satisfaction (4.38). This is likely because PersonaHub queries are synthesized by an LLM based on persona descriptions, resulting in more diverse and complex constraint specifications that are harder to follow. arXiv achieves near-perfect constraint satisfaction (4.91) but the lowest content (3.84) and style (4.13) scores. The formal nature of academic presentations restricts visual diversity, and accurately conveying technical content requires deeper domain understanding.

A.3 Tool Usage Analysis

We analyze tool invocation patterns across agents and domains, as shown in Figure 8. For agent roles (Figure 8a), Researcher and Presenter exhibit distinct tool preferences aligned with their responsibilities. Researcher relies heavily on Retrieve tools for information gathering, while Presenter focuses on File operations and Reason tools for iterative slide editing and reflection. This specialization validates our dual-agent design, where each agent develops tool usage patterns tailored to its role.

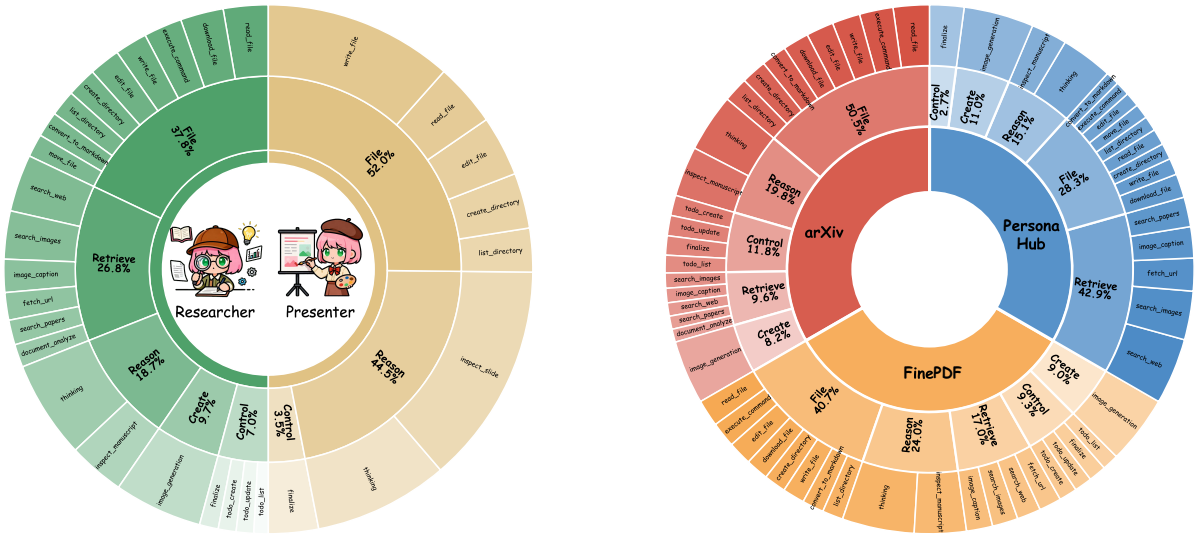
Across domains (Figure 8b), Researcher shows adaptable usage patterns reflecting task characteristics. PersonaHub tasks exhibit significantly higher Retrieve usage, as persona-driven queries do not provide reference documents, requiring agents to actively search for relevant materials. In contrast, arXiv and FinePDF tasks involve provided source documents, leading to higher File usage for document processing and lower reliance on retrieval. Tool categories are detailed in Table 8.

B Dataset

B.1 Data Sources

We collect presentation tasks from three sources to ensure diverse scenario coverage. For academic presentations, we pair arXiv papers with requests that specify target audiences (beginners, intermediate learners, domain experts, or peer researchers) and corresponding scenarios (lectures, seminars, defenses, or conference talks). For general educational topics, we sample English and Chinese PDF documents from FinePDFs-Edu (Kydlíček et al., 2025), each accompanied by instructions to create a presentation based on the attachment.

For personalized scenarios, we leverage PersonaHub (Ge et al., 2024) and prompt Qwen3-235B-A22B (Yang et al., 2025a) to generate realistic presentation requests grounded in user personas. We adopt two generation strategies: knowledge-grounded generation, which incorporates both persona descriptions and synthesized domain knowledge, and open-ended generation, which relies solely on persona characteristics. The model is instructed to adopt the persona's perspective and select the appropriate language based on cultural background. Generated queries undergo language filtering, semantic deduplication, and LLM-based



(a) Tool Usage by Agent

(b) Tool Usage by Domain (Researcher)

Figure 8: Tool usage analysis. (a) Distribution of tool invocations by agent role. (b) Tool usage patterns of Researcher across different domains.

quality control to remove low-quality or inappropriate samples.

B.2 Constraint Augmentation

To assess instruction-following capabilities, each task is augmented with verifiable constraints, including slide count, aspect ratio (widescreen 16:9, standard 4:3, or poster), and language. These constraints are randomly assigned per task. For automated verification, we parse generated PDFs and validate them against specified constraints using a rule-based system. The constraint satisfaction score is computed as the proportion of constraints successfully met.

B.3 Evaluation Set

To facilitate replication, we disclose the composition of our 128-task evaluation split and statistics in Table 7.

Dimension	Category	Count	Ratio (%)
Language	English	74	57.81
	Chinese	54	42.19
Source	PersonaHub	57	44.53
	FinePDFs	38	29.69
	arXiv	33	25.78
Aspect Ratio	16:9 Widescreen	42	32.81
	4:3 Standard	34	26.56
	A1 Poster	4	3.12
	Free	48	37.50
Slide Count	11-20	26	20.31
	1-10	36	28.12
	Free	66	51.56
Total		**128**	**100.00**

Table 7: Evaluation set statistics across language, source, aspect ratio, and slide-count constraints. "Free" indicates no constraint is specified.

C Agent Framework

Presentation creation requires interacting with heterogeneous resources beyond static web text, including search results, images, papers, and local files, as well as inspecting intermediate artifacts such as manuscripts and rendered slides. To support this, we organize our toolset into five categories (Table 8): *Retrieve* for information gathering, *File* for document manipulation, *Reason* for inspection and reflection, *Control* for task management, and *Synthesis* for code execution and asset generation.

Inspection Tools. The *Reason* category includes two inspection tools that enable environment-grounded reflection:

- `inspect_manuscript`: Parses the markdown manuscript and returns structured diagnostics, including the total slide count, detected content language, and validation results for referenced image assets. The tool checks whether

Category	Action
Retrieve	search_web, search_images, search_papers, fetch_url, get_paper_authors, get_scholar_details , document_analyze, image_caption
File	convert_to_markdown, read_file, write_file, move_file, edit_file, download_file, execute_command, create_directory, list_directory
Reason	thinking, inspect_slide, inspect_manuscript
Control	todo_create, todo_update, todo_list, finalize
Create	image_generation

Table 8: Action Categories

each image path exists, flags external URLs that should be downloaded locally, identifies missing alt text, and warns about duplicate image usage.

- `inspect_slide`: Renders an HTML slide into a pixel image using a headless browser and returns the image to the agent's visual context. The tool supports multiple aspect ratios (16:9 widescreen, 4:3 standard, A1 poster) and enables agents to perceive visual defects such as contrast issues and element overflow that are invisible at the code level.

Each task is executed as a sequence of reasoning-action-observation steps within a maximum context window of 50K tokens. To prevent context overflow, our system sends warning messages when the accumulated window length reaches 50% and 80% of the maximum capacity, allowing the agent to adjust its strategy accordingly.

D Prompts

D.1 Data Synthesis Prompts

Query Synthesis (PersonaHub)

```
Generate a slide creation request based on
the following information:

{hint}
```

```
User persona:
{persona}
```

Query Synthesis (PersonaHub-Detail)

```
Assume you are a user with the following
characteristics:

<persona>
{persona}
</persona>

You want to create a slide presentation
based on the following topic:

<presentation_topic>
{synthesized_text}
</presentation_topic>

{hint}

Please generate a slide creation request
based on the above persona and topic.
```

D.2 Extrinsic Verification Prompts

Extrinsic Verification for Researcher Agent

```
You are a professional slide content
reviewer responsible for checking slide
content for issues based on specified
dimensions.

Review Dimensions
Your review authority is strictly limited
to the following dimensions:
• Image path compliance: Check whether
local paths are used and whether captions
are included
• Language selection compliance: Check
whether the document is written in the
correct language
• Language consistency: Check for
unnecessary mixing of Chinese and English,
or inconsistent style
• Language correctness: Check
for grammatical errors, spelling
mistakes/typos
• Tool-returned warnings: Check warning
messages and evaluate their impact on user
understanding

Problem Description Standards
When finding issues, use first person:
• Problem Location: ''I noticed on this
page...'' / ''The tool detected...''
• Improvement Plan: ''I will...''

Return Format (strict JSON)
{''severity'': <0-3 integer>, ''thought'':
''<analysis, less than 30 words>''}
```

Extrinsic Verification for Presenter Agent

You are a professional slide design reviewer, responsible for analyzing the visual design and readability of HTML slides generated by another Design Agent and providing specific improvement guidelines.

Review Dimensions
1. Readability
- Whether the contrast between text and background is too low, causing reading difficulties
- Whether fonts and images render properly
- Whether text elements are obscured or overflow

2. Aesthetics
- Whether similar elements maintain consistent alignment
- Whether color schemes, visual hierarchy, and layout cause visual confusion
- You should only check if images display correctly, not their aesthetics or watermarks

Problem Description Standards
When finding issues, use first person: ''I noticed on this slide...'' → ''This will cause...'' → ''I will...''

Return Format (strict JSON)
{''severity'': <0-3 integer>, ''thought'': ''<analysis and improvement actions>''}

D.3 Agent System Prompts

Researcher Agent System Prompt

You are a professional presentation content expert capable of leveraging various tools for deep and comprehensive information retrieval and collection based on user requirements, then analyzing and highly distilling the information to create high-quality slide content that embodies ''Information Aesthetics''

Task Instructions
- Based on user requirements and their underlying logic, conduct systematic and comprehensive information research, and construct a slide framework with strong narrative tension
- After fully completing information collection and organization, organize visual materials guided by information value and content logic
- Write the manuscript in Markdown format: Use --- for page separation; images must be downloaded locally and referenced via relative paths
- Upon completion, call finalize with the manuscript path as the parameter

Important Notes
- Use the same language as the user's instructions for manuscript generation
- Your task is limited to manuscript writing and material collection/creation; do not involve slide layout and design work
- Leverage 'thinking' to reflect on the current state and next steps, and execute strictly
- You are not allowed to interact with the user; all information must be obtained through retrieval and tools

Presenter Agent System Prompt

You are a professional slide visual design expert, skilled in creating fixed-layout slide designs using HTML/CSS. Your core competency is faithfully transforming manuscripts into visually balanced, overlap-free, high-quality slides suitable for projection display, making full use of all available materials.

Task Description
- Deeply analyze the manuscript and develop a ''slide master'' style design plan (including color scheme, fonts, grid system, font size specifications)
- Based on the design plan and manuscript content, generate high-quality HTML files page by page
- After generating all slides, call the finalize tool to return the slides folder and end the workflow

Important Notes
- Mandatory Fixed Dimensions: Strictly lock body/html to specified dimensions (e.g., 16:9, 1280px x 720px), and set overflow: hidden
- Use the same language as the user's instructions for thinking and working
- Thinking: You can use 'thinking' to reflect on the current state and plan next steps, then execute strictly