Beyond Text-to-Slides: An Edit-based Agent for Well-rounded Presentation Generation

Anonymous ACL submission

Abstract

Generating slides from documents is a complex and challenging task that requires balancing content quality, structural coherence, and visual design. Existing methods primarily focus on directly converting raw text into slides, prioritizing content quality while often neglecting critical aspects such as structural coherence and visual design. Inspired by human presentation creation processes, we propose *PPTAgent* , a novel framework that redefines presentation generation as an edit-based process using reference presentations, enabling LLMs to create well-rounded presentations. *PPTAgent* comprises two stages: (1) Presentation Analysis, which enhances LLMs' comprehension of the structure and content schemas by analyzing reference presentations. (2) Presentation Generation, which generates a detailed outline for the document and assigns specific document sections and reference slides to each slide. To comprehensively evaluate the quality of generated presentations, we introduce *PPTEval* , a comprehensive evaluation framework that assesses presentations across three key dimensions: content, design, and coherence. Experimental results demonstrate that *PPTAgent* significantly outperforms conventional presentation generation methods across all three key dimensions.

1 Introduction

PowerPoint is a widely used medium for information delivery, valued for its visual effectiveness in engaging and communicating with audiences. However, producing high-quality presentations requires a captivating storyline, visually appealing layouts, and rich, impactful content (Fu et al., 2022). Consequently, creating well-rounded presentations necessitates advanced presentation skills and considerable professional effort. Given the inherent complexity of presentation creation, there is growing interest in automating the presentation generation process by leveraging the generalization capabilities of large language models (LLM) (Mondal et al., 2024; Maheshwari et al., 2024).

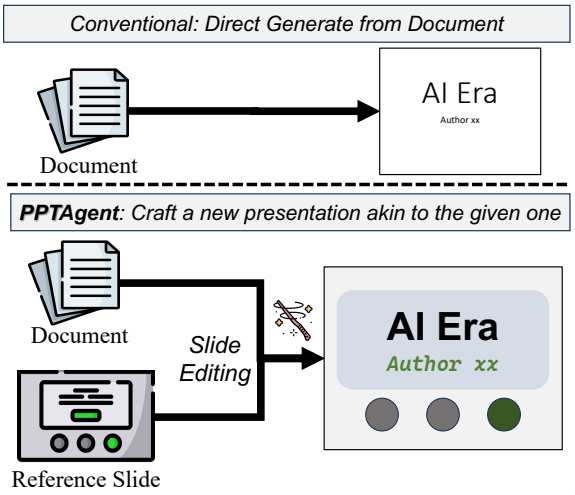

Figure 1: In the comparison between the conventional LLM-based (above) and our approach using *PPTAgent* (below), our method, which begins by editing a reference slide, aligns more closely with the human presentation creation process.

However, automated presentation generation remains significantly limited in practical applications. This is mainly because existing approaches often adopt an end-to-end paradigm, where the input is text and the output is a presentation, without incorporating diverse style references. Consequently, these methods inevitably produce simplistic and visually uninspiring presentations. Specifically, as illustrated in Figure 1, prior studies such as Mondal et al. (2024) and Li et al. (2021) predominantly focus on enhancing textual content while overlooking the visual-centric nature of presentations, leading to outputs that fail to effectively engage audiences. Furthermore, the complexity of the raw PowerPoint format, encoded in XML (Gryk, 2022), poses significant challenges for models to understand the structure and spatial layout of presentations, let alone generate captivating and well-rounded

presentations. More critically, existing research lacks comprehensive presentation evaluation frameworks, relying primarily on simplistic metrics like textual fluency and success rates. These inadequate metrics fail to account for essential aspects of a high-quality presentation, including a compelling narrative, visually appealing layouts, and impactful content. In some cases, they even encourage excessive alignment with the input document, sacrificing the brevity and clarity that are key to effective presentations. These limitations underscore the significant potential for growth in automated presentation generation, particularly in producing visually appealing presentations and establishing proper evaluation frameworks.

From a cognitive perspective, humans commonly create presentations by first selecting a visually appealing and well-structured slide as a reference, then summarizing and transferring key content from the text onto the selected slide (Duarte, 2010). Inspired by this process, we aim to enhance the visual quality of presentations by providing LLMs with reference styles to guide their generation. In this paper, we introduce *PPTAgent* , a novel framework that redefines presentation generation as an edit-based process utilizing reference presentations. At its core, *PPTAgent* selects different reference slides for various sections of a document and then edits the content of these reference slides based on the document to produce the target presentation. This process is divided into two stages: Presentation Analysis and Presentation Generation. As illustrated in Figure 2, given a document and a reference presentation, ***Stage I: Presentation Analysis*** enhances the textual description of each slide. This step provides detailed information about the purpose and type of each slide, such as bullet slides, opening slides, or display slides. Based on this analysis, ***Stage II: Presentation Generation*** generates a detailed outline for the document and assigns specific document sections and reference slides to each slide. For instance, for the meta-information in the document, the framework selects the opening slide as the reference slide to display details such as the title and authors. *PPTAgent* provides a set of editing action APIs, enabling LLMs to iteratively refine the reference slide through executable code actions, thereby completing the generation process.

To comprehensively evaluate the quality of generated presentations, we propose *PPTEval* , a robust evaluation framework that assesses presentation quality across multiple dimensions. Inspired by Kwan et al. (2024), we categorized presentation quality into three dimensions: Content, Design, and Coherence. *PPTEval* offers both quantitative scores and qualitative feedback, enabling further analysis and improvement. Experimental results demonstrate the ability of our method to generate high-quality presentations, achieving an average score of 3.67 across the three dimensions in *PPTEval* . These results span a variety of domains and highlight a high success rate of 97.8%, showcasing the versatility and robustness of our approach. Furthermore, human evaluations validate the reliability and effectiveness of *PPTEval* .

Our main contributions can be summarized as follows: [1]

- We propose *PPTAgent* , a novel framework that redefines presentation generation as an edit-based process using reference presentations, enabling LLMs to create well-rounded presentations without the need for human supervision or additional training.
- We developed *PPTEval* , a comprehensive evaluation framework that assesses presentations across three key dimensions: Content, Design, and Coherence. This framework provides fine-grained and reliable feedback, with evaluation results confirming its effectiveness.
- Experimental results show that our approach significantly outperforms the baseline method across all evaluated dimensions. Additionally, the high success rate of over 95% underscores the robustness of our method.

2 PPTAgent

In this section, we first establish the formulation of the presentation generation task. Subsequently, we describe the framework of our proposed *PPTAgent* , which operates in two distinct stages. In stage I, we extract semantic information and content schemas from the reference presentation. This process aims to enhance the expressiveness of the reference presentation for the convenience of further presentation generation, thereby facilitating subsequent presentation generation. In stage II, given an input document and the analyzed reference presentation, we aim to select the most suitable slides and generate the target presentation through an interactive editing process based on the selected slides. An

[1] We uploaded the code and datasets as supplemental materials, which will be openly released after acceptance.

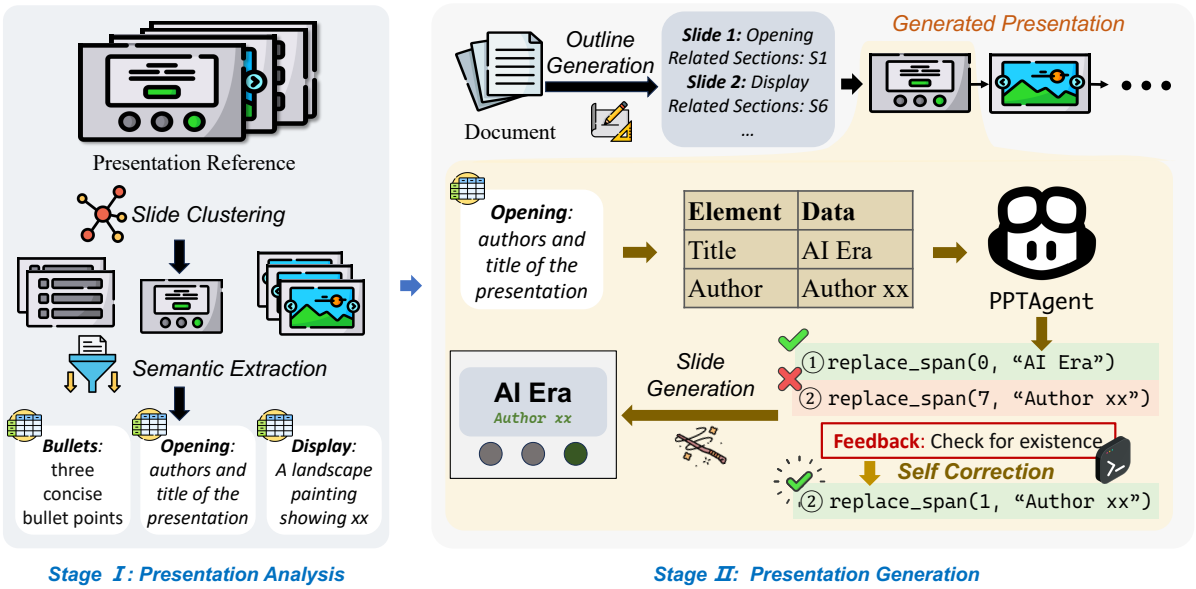

Figure 2: Overview of the *PPTAgent* workflow. ***Stage I: Presentation Analysis*** involves analyzing the input presentation to cluster slides into groups and extract their semantic information. ***Stage II: Presentation Generation*** generates new presentations guided by the outline, incorporating feedback mechanisms to ensure robustness.

overview of our proposed workflow is illustrated in Figure 2.

2.1 Problem Formulation

PPTAgent is designed to generate an engaging presentation via an edit-based process. We will provide formal definitions for both *PPTAgent* and the conventional method, illustrating their divergence.

The conventional method for creating each slide $\boldsymbol{S}$ can be described in Equation 1, where n represents the number of elements on the slide, and C denotes the source content composed of sections and figures. Each element on the slide, e_i, is defined by its type, content, and styling attributes, such as $(\text{Textbox}, \text{"Hello"}, \{\text{border}, \text{size}, \text{position}, \dots\})$.

$$\boldsymbol{S} = \sum_{i=1}^{n} e_i = f(C) \tag{1}$$

Compared to the conventional method, *PPTAgent* adopts an edit-based generation paradigm for creating new slides, addressing challenges in processing spatial relationships and designing styles. This approach generates a sequence of actions to modify existing slides. Within this paradigm, both the input document and the reference presentation serve as inputs. This process can be described as Equation 2, where m represents the number of generated actions. Each action a_i represents a line of executable code, and R_j is the target slide being edited.

$$\boldsymbol{A} = \sum_{i=1}^{m} a_i = f\left(C \mid R_j\right) \tag{2}$$

2.2 Stage I: Presentation Analysis

To facilitate the presentation generation process, we first cluster the slides in the reference presentation and extract their semantic information. This detailed semantic description aids LLMs in determining which slides to edit.

Slide Clustering Slides can be categorized into two main types based on their functionalities: slides that support the structure of the presentation (e.g., opening slides) and slides that convey specific content (e.g., title slides with bullet points). To effectively cluster slides in the presentation, we employ different clustering algorithms based on their textual or visual characteristics. For structural slides, we leverage LLMs to infer the functional role of each slide and group them accordingly, as these slides often exhibit distinctive textual features. For the remaining slides, which primarily convey specific content, we use hierarchical clustering based on image similarity, which is detailed at Appendix C. For each cluster, we infer the layout patterns of each cluster using MLLMs.

Semantic Extraction After clustering slides to facilitate the selection of slide references, we fur-

ther analyzed their content schemas to ensure purposeful alignment of the editing. Given the complexity and fragmentation of real-world slides, we utilized the context perception capabilities of LLMs (Chen et al., 2024a) to extract diverse content schemas. Specifically, we defined an extraction framework where each element is represented by its category, modality, and content. Based on this framework, the schema of each slide was extracted through LLMs' instruction-following and structured output capabilities. Detailed instructions are provided in Appendix E.

2.3 Stage II: Presentation Generation

In this stage, we begin by generating an outline that specifies the reference slide and relevant content for each slide in the new presentation. For each slide, LLMs iteratively edit the reference slide using interactive executable code actions to complete the generation process.

Outline Generation Following human preferences, we instruct LLMs to create a structured outline composed of multiple entries. Each entry specifies the reference slide, relevant document section indices, as well as the title and description of the new slide. By utilizing the planning and summarizing capabilities of LLMs, we provide both the document and semantic information to generate a coherent and engaging outline for the new presentation, which subsequently orchestrates the generation process.

Slide Generation Guided by the outline, the slide generation process iteratively edits a reference slide to produce the new slide. To enable effective interaction between LLMs and slides, we introduce five specialized APIs that allow LLMs to perform actions such as editing, removing, and duplicating slide elements. To further enhance the comprehension of slide structure, inspired by Feng et al. (2024) and Tang et al. (2023), we convert slides from their raw XML format into an HTML representation, which is more interpretable for LLMs. For each slide, text retrieved from section indices and captions of available images is provided to the LLMs, with the content of the new slide generated under the guidance of the content schema.

Subsequently, the LLMs are provided with the generated slide content, the HTML representation of the reference slide, and detailed API documentation to produce executable editing actions.

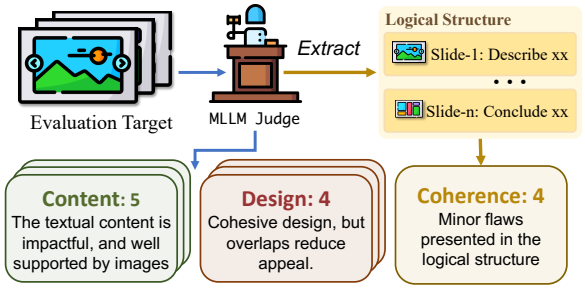

Figure 3: This figure illustrates the evaluation process in *PPTEval* , which assesses three key dimensions: content, design, and coherence. Content evaluates the quality of text and images within the slides. Design examines the visual consistency and appeal. Coherence focuses on the logical flow of the presentation. Each dimension is rated on a scale from 1 to 5, with detailed feedback provided for improvement.

These actions are executed in a REPL[2] environment, where the system detects errors during execution and provides real-time feedback for self-correction. The self-correction mechanism leverages intermediate results to iteratively refine the editing actions, enhancing the robustness of the generation process.

3 PPTEval

To address the limitations of existing automated metrics for presentation evaluation, we introduce *PPTEval* , a comprehensive framework for assessing presentation quality from multiple perspectives. The framework provides scores on a 1-to-5 scale and offers detailed feedback to guide the improvement of future presentation generation methods. The overall evaluation process is depicted in Figure 3, and the scoring criteria are outlined in Appendix E.

Drawing from Duarte (2008, 2010), we have identified three key dimensions for evaluating presentation quality:

Content: The content dimension evaluates the information presented on the slides, focusing on both text and images. We assess content quality from three perspectives: the amount of information, the clarity and quality of textual content, and the support provided by visual content. High-quality textual content is characterized by clear, impactful text that conveys the proper amount of information. Additionally, images should complement and reinforce the textual content, making the information

[2] https://en.wikipedia.org/wiki/Read-eval-print_loop

more accessible and engaging. To evaluate content quality, we employ MLLMs on slide images, as slides cannot be easily comprehended in a plain text format.

Design: Good design not only captures attention but also enhances content delivery. We evaluate the design dimension based on three aspects: color schemes, visual elements, and overall design. Specifically, the color scheme of the slides should have clear contrast to highlight the content while maintaining harmony. The use of visual elements, such as geometric shapes, can make the slide design more expressive. Finally, good design should adhere to basic design principles, such as avoiding overlapping elements and ensuring that design does not interfere with content delivery.

Coherence: Coherence is essential for maintaining audience engagement in a presentation. We evaluate coherence based on the logical structure and the contextual information provided. Effective coherence is achieved when the model constructs a captivating storyline, enriched with contextual information that enables the audience to follow the content seamlessly. A coherent presentation should exhibit a consistent flow of information, consistent use of terminology, and clear, logically connected transitions between slides. We assess coherence by analyzing the logical structure and contextual information extracted from the presentation.

4 Experiment

4.1 Dataset

Data Collection Existing presentation datasets, such as Mondal et al. (2024); Sefid et al. (2021); Sun et al. (2021); Fu et al. (2022), have two main issues. First, they are mostly stored in PDF or JSON formats, which leads to a loss of semantic information, such as structural relationships and styling attributes of elements. Additionally, these datasets are primarily derived from academic reports, limiting their diversity. To address these limitations, we introduce *Zenodo10K*, a new dataset sourced from Zenodo (European Organization For Nuclear Research and OpenAIRE, 2013), an open digital repository hosting diverse artifacts from different domains. We have curated 10,448 presentations from this source and made them publicly available to support further research. Following Mondal et al. (2024), we sampled 50 presentations across five domains to serve as reference presentations. Additionally, we collected 50 documents from the same domains to be used as input documents. Details of the sampling criteria are provided in Appendix A.

Domain	*Document*		*Presentation*		
	#Chars	#Figs	#Chars	#Figs	#Pages
Culture	12,708	2.9	6,585	12.8	14.3
Education	12,305	5.5	3,993	12.9	13.9
Science	16,661	4.8	5,334	24.0	18.4
Society	13,019	7.3	3,723	9.8	12.9
Tech	18,315	11.4	5,325	12.9	16.8

Table 1: Statistics of the dataset used in our experiments, detailing the number of characters ('#Chars') and figures ('#Figs'), as well as the number of pages ('#Pages').

Data Preprocessing We prepare the documents by extracting textual and visual content using VikParuchuri (2023). Furthermore, textual content was organized into a series of sections using Qwen2.5-72B-Instruct (Yang et al., 2024). Additionally, images from both presentations and documents were captioned using Qwen2-VL-72B-Instruct (Wang et al., 2024a). We further reduced the redundancy of our dataset, images within presentations and documents were considered duplicates and removed if their ViT (Wu et al., 2020) embeddings had a cosine similarity score exceeding 0.85. Following Fu et al. (2022), slides were removed if their text embeddings, computed using Chen et al. (2024b), had a cosine similarity score exceeding 0.8 relative to the preceding slide. Detailed statistics of the dataset are provided in the Table 1.

4.2 Experimental Settings and Baseline

Models We evaluate our method using three state-of-the-art models: GPT-4o-2024-08-06 (GPT-4o), Qwen2.5-72B-Instruct (Qwen2.5, (Yang et al., 2024)), and Qwen2-VL-72B-Instruct (Qwen2-VL, Wang et al., 2024a). These models are categorized based on their capabilities in processing textual and visual information, as indicated by their subscripts. Specifically, we define configurations as combinations of a language model (LM) and a vision model (VM), such as Qwen2.5$_{\texttt{LM}}$+Qwen2-VL$_{\texttt{VM}}$.

During experiments, we allow up to 2 iterations of self-correction for each slide generation task. Each configuration generates $5 \times 10 \times 10 = 500$ slides. All open-source LLMs are deployed using the VLLM framework (Kwon et al., 2023) on a

Setting		Existing Metrics			PPTEval			
		SR(%)↑	PPL↓	FID↓	Content↑	Design↑	Coherence↑	Avg.↑
Baseline								
GPT-4o	–	–	**110.6**	–	2.98	2.33	3.24	2.85
Qwen2.5	–	–	122.4	–	2.96	2.37	3.28	2.87
PPTAgent								
GPT-4o$_{\text{LM}}$	GPT-4o$_{\text{VM}}$	**97.8**	459.7	7.48	3.25	3.24	4.39	3.62
Qwen2-VL$_{\text{LM}}$	Qwen2-VL$_{\text{VM}}$	43.0	322.3	7.32	3.13	**3.34**	4.07	3.51
Qwen2.5$_{\text{LM}}$	Qwen2-VL$_{\text{VM}}$	95.0	313.9	**6.20**	**3.28**	3.27	**4.48**	**3.67**
Ablation								
	w/o Outline	91.0	2304.3	6.94	3.24	3.30	3.36	3.30
	w/o Schema	78.8	164.8	7.12	3.08	3.23	4.04	3.45
	w/o Structure	92.2	189.9	7.66	3.28	3.25	3.45	3.32
	w/o CodeRender	74.6	231.0	7.03	3.27	3.34	4.38	3.66

Table 2: Table showing the evaluation of different settings under automated metrics and *PPTEval* .

Domain	SR (%) ↑	PPL↓	FID↓	PPTEval ↑
Culture	93.0	185.3	5.00	3.70
Education	94.0	249.0	7.90	3.69
Science	96.0	500.6	6.07	3.56
Society	95.0	396.8	5.32	3.59
Tech	97.0	238.7	6.72	3.74

Table 3: Evaluation results of *PPTAgent* under the configuration of Qwen2.5 in different domains, using the success rate (SR) and the average PPTEval score across three evaluation dimensions.

cluster of 8 NVIDIA A100 GPUs. The total computational cost for these experiments is approximately 500 GPU hours.

Baseline We adopt the methodology described in Bandyopadhyay et al. (2024) atheur baseli. This approach employs a multi-staged end-to-end model to generate narrative-rich presentations, with an image similarity-based ranking algorithm to add images to the slides. The baseline method is evaluated using either GPT-4o or Qwen2.5, as it does not require the necessary processing of visual information. Each configuration generates $5 \times 10 = 50$ presentations, given that it does not require an input presentation. We do not report the success rate and FID of the baseline method for the same reason.

4.3 Evaluation Metrics

- **Success Rate (SR):** measures the robustness of the generation task by determining the percentage of presentations where all slides are successfully generated.
- **Perplexity (PPL):** measures the likelihood of the language model generating the given sequence. We calculate the average perplexity of slides within a presentation using GPT-2. A lower perplexity score indicates that the textual content is more fluent.
- **FID (Heusel et al., 2017):** measures the similarity between the generated presentation and the exemplar presentation in the feature space. Due to the limited sample size, we calculate the FID using a 64-dimensional output vector.
- **PPTEval:** measures the comprehensive quality of presentations across three dimensions: coherence, content, and design. We employ GPT-4o as the judge.

4.4 Result & Analysis

PPTAgent Enhances LLMs' Presentation Generation Capabilities As demonstrated in Table 2, our approach empowers LLMs to produce well-rounded presentations with a remarkable success rate, achieving $\geq$ 95% success rate for both Qwen2.5$_{\text{LM}}$+Qwen2-VL$_{\text{VM}}$ and GPT-4o$_{\text{LM}}$+GPT-4o$_{\text{VM}}$. This is a significant improvement compared to the highest accuracy of 10% for session-based template editing tasks as reported in Guo et al. (2023). This improvement can be attributed to three main factors: 1) *PPTAgent* concentrates on content modifying, thereby avoiding intricate stying operations. 2) Our streamlined API design allows LLMs to execute tasks with ease. 3) The code interaction module enhances LLMs' comprehension of slides and offers opportunities

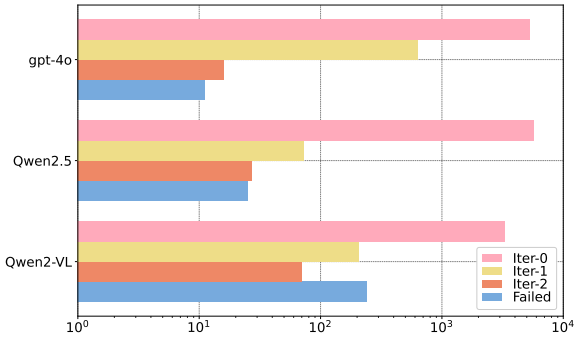

Figure 4: The number of iterative self-corrections required to generate a single slide under different models.

for self-correction, enabling them to generate accurate actions robustly. Moreover, detailed performance of Qwen2.5$_{\text{LM}}$+Qwen2-VL$_{\text{VM}}$ across various domains, as illustrated in Table 3, underscores the robustness of our approach.

PPTAgent Significantly Improves Overall Presentation Quality By adopting an Edit-based paradigm, *PPTAgent* allows elements within the presentation to inherit well-designed styling attributes from existing presentations. When using GPT-4o, experimental results demonstrate comprehensive improvements over the baseline. We significantly surpass the baseline method in the design dimension under *PPTEval* (3.24 vs 2.33), as the presentations generated by the baseline method lack basic design efforts. Furthermore, we achieved substantial enhancements in coherence (4.39 vs 3.28) and content (3.25 vs 2.98) dimensions, as the semantic information extracted during the Presentation Analysis stage effectively guided the LLMs.

Open-Source LLMs Rival GPT-4o in Performance GPT-4o consistently demonstrates outstanding performance across various evaluation metrics, highlighting its advanced capabilities. While Qwen2-VL exhibits limitations in linguistic proficiency due to the trade-offs from multimodal post-training, GPT-4o maintains a clear advantage in handling language tasks. However, the introduction of Qwen2.5 successfully mitigates these linguistic deficiencies, bringing its performance on par with GPT-4o. This underscores the significant potential of open-source LLMs as competitive and highly capable presentation agents.

4.5 Ablation Study

To better understand the impact of each component in our proposed method, we performed ablation studies using four different configurations. Specifically, we evaluated the method by: (1) randomly selecting a slide as the edit target (w/o Outline), (2) omitting structural information during outline generation (w/o Structure), (3) replacing the slide representation with the method described in Guo et al. (2023) (w/o CodeRender), and (4) removing guidance from the slide schema (w/o Schema). These configurations were tested using the Qwen2.5$_{\text{LM}}$+Qwen2-VL$_{\text{VM}}$.

Code Representation Enhances LLMs' Comprehension As shown in Table 2, the removal of the Code Render component leads to a significant drop in the model's success rate (SR) from 95.0 to 74.6. This underscores the critical role of code representation in leveraging LLMs' coding capabilities to improve their overall comprehension.

Presentation Analysis is Essential for Generating Targeted Presentations The removal of the outline and structural information significantly degrades coherence (from 4.48 to 3.36/3.45), underscoring their crucial role in maintaining logical flow. Furthermore, the absence of slide schema hinders LLMs from generating targeted content effectively, resulting in a drop in success rate from 95.0 to 78.8.

4.6 Error Analysis

Figure 4 illustrates the number of iterations required to generate a slide using different models. Although GPT-4o exhibits superior self-correction capabilities compared to Qwen2.5, Qwen2.5 encounters fewer errors in the first iteration (Iter-0). Additionally, we observed that Qwen2-VL experiences errors more frequently and has poorer self-correction capabilities, likely due to its multimodal post-training (Wang et al., 2024a). Ultimately, all three models successfully corrected more than half of the errors, demonstrating that our iterative self-correction mechanism effectively ensures the success of the generation process.

4.7 Effectiveness of PPTEval

Human Agreement Evaluation Despite Chen et al. (2024a) have highlighted the impressive human-like discernment of LLMs in various generation tasks. However, it remains crucial to assess the correlation between LLM evaluations and human evaluations in the context of presentations. This necessity arises from findings by Laskar et al.

Corelation	Content	Design	Coherence	Avg.
Pearson	0.70	0.90	0.55	0.71
Spearman	0.73	0.88	0.57	0.74

Table 4: The correlation scores between human ratings and LLM ratings under different dimensions (Coherence, Content, Design). All presented data of similarity exhibit a p-value below 0.05, indicating a statistically significant level of confidence.

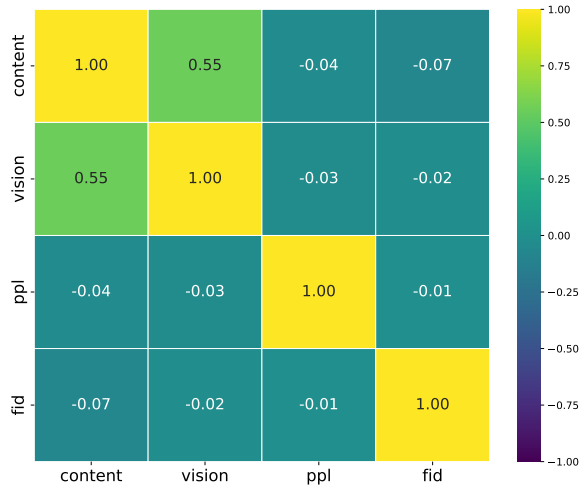

Figure 5: Correlation heatmap between existing automated evaluation metrics and *PPTEval* .

(2024), which indicate that LLMs may not be adequate evaluators for complex tasks. Table 4 shows the correlation of ratings between humans and LLMs. The average Pearson correlation of 0.71 exceeds the scores of other evaluation methods (Kwan et al., 2024), indicating that *PPTEval* aligns well with human preferences.

Moreover, the heatmap in Figure 5 compares the correlation between existing evaluation metrics and different dimensions of *PPTEval* . In human evaluations, both PPL and FID exhibit a lack of correlation with the Content and Design dimensions, demonstrating the necessity of *PPTEval* to achieve effective evaluation results.

5 Related Works

Automated Presentation Generation Recent proposed methods for slide generation can be categorized into rule-based and template-based based on how they handle element placement. Rule-based methods, such as those proposed by Mondal et al. (2024) and Li et al. (2021), often focus on enhancing textual content but neglect the visual-centric nature of presentations, leading to outputs that lack engagement. Template-based methods, including Cachola et al. (2024) and industrial solutions like Tongyi, rely on pre-designed templates to create visually appealing presentations. However, their dependence on extensive manual effort for template annotation significantly limits scalability and flexibility.

LLM Agent Numerous studies (Li et al., 2024; Deng et al., 2024; Wang et al., 2024c) have explored the potential of LLMs to act as agents assisting humans in a wide array of tasks. For example, Zheng et al. (2024); Wang et al. (2024b) demonstrate the capability of LLMs to accomplish tasks by generating executable actions and correcting errors based on feedback. Furthermore, Guo et al. (2023) introduces an evaluation system that assesses the ability of LLMs to perform multi-turn, multimodal slide editing tasks using APIs, which inspired the use of LLMs for complex tasks as proposed in this study.

LLM as a Judge LLMs have demonstrated strong capabilities in instruction following and context perception, leading to their widespread use as judges (Liu et al., 2023; Zheng et al., 2023). Further research by Zhuge et al. (2024) enhanced LLMs' abilities through external modules and functions, while Chen et al. (2024a) validated the feasibility of using multimodal large language models(MLLMs) as judges. Additionally, Kwan et al. (2024) introduced a multi-dimensional evaluation framework for multi-turn conversations, which inspired the development of our proposed *PPTEval* .

6 Conclusion

In this paper, we introduced *PPTAgent* , which conceptualizes presentation generation as a two-stage presentation editing task completed through the abilities of LLMs to understand and generate code. This approach leveraged the textual feature and layout patterns to organize slides into different functional groups. Our experiments across data from multiple domains have demonstrated the superiority of our method. Moreover, our proposed *PPTEval* ensured the assessability of presentations. This research provides a new paradigm for generating slides under unsupervised conditions and offers fresh insights for future work in presentation generation.

7 Limitations

While our method demonstrates its capability to produce high-quality presentations, there remain inherent challenges that impact its universal applicability. For instance, achieving a success rate of over 95% on our dataset is impressive, but not absolute, thus might limit its application. Moreover, parsing slides with intricate nested group shapes often proves to be a bottleneck, leading to less consistent results. Additionally, although *PPTAgent* shows noticeable improvements in layout optimization over prior approaches, it still falls short of exploiting the full potential of visual cues for refining stylistic consistency. This often manifests in design flaws, such as overlapping elements, undermining the visual harmony of the generated slides. Addressing these limitations calls for future enhancements that integrate visual information into the generation process.

8 Ethical Considerations

In the construction of *Zenodo10K*, we utilized the publicly available API to scrape data while strictly adhering to the licensing terms associated with each artifact. Specifically, artifacts that were not permitted for modification or commercial use under their respective licenses were filtered out to ensure compliance with intellectual property rights. Additionally, all annotation personnel involved in the project were compensated at rates exceeding the minimum wage in their respective cities, reflecting our commitment to fair labor practices and ethical standards throughout the dataset's development process.

References

Sambaran Bandyopadhyay, Himanshu Maheshwari, Anandhavelu Natarajan, and Apoorv Saxena. 2024. Enhancing presentation slide generation by llms with a multi-staged end-to-end approach. *arXiv preprint arXiv:2406.06556*.

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.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Pan Zhou, Yao Wan, and Lichao Sun. 2024a. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Nancy Duarte. 2008. *Slide: ology: The art and science of creating great presentations*, volume 1. O'Reilly Media Sebastapol.

Nancy Duarte. 2010. *Resonate: Present visual stories that transform audiences*. John Wiley & Sons.

European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2024. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36.

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.

Michael Robert Gryk. 2022. Human readability of data files. *Balisage series on markup technologies*, 27.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Duan Nan. 2023. Pptc benchmark: Evaluating large language models for powerpoint task completion. *arXiv preprint arXiv:2311.01767*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. Mt-eval: A multi-turn capabilities evaluation benchmark for large language models. *Preprint*, arXiv:2401.16745.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, Enamul Hoque, Shafiq Joty, and Jimmy Huang. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13785–13816, Miami, Florida, USA. Association for Computational Linguistics.

Da-Wei Li, Danqing Huang, Tingting Ma, and Chin-Yew Lin. 2021. Towards topic-aware slide generation for academic papers with unsupervised mutual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13243–13251.

Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. 2024. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Himanshu Maheshwari, Sambaran Bandyopadhyay, Aparna Garimella, and Anandhavelu Natarajan. 2024. Presentations are not always linear! gnn meets llm for document-to-presentation transformation with attribution. *arXiv preprint arXiv:2405.13095*.

Ishani Mondal, S Shwetha, Anandhavelu Natarajan, Aparna Garimella, Sambaran Bandyopadhyay, and Jordan Boyd-Graber. 2024. Presentations by the humans and for the humans: Harnessing llms for generating persona-aware slides from documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2664–2684.

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.

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*.

Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. 2023. Layoutnuwa: Revealing the hidden layout expertise of large language models. *arXiv preprint arXiv:2309.09506*.

VikParuchuri. 2023. marker.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024b. Executable code actions elicit better llm agents. *arXiv preprint arXiv:2402.01030*.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. 2024c. Opendevin: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision. *Preprint*, arXiv:2006.03677.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. 2024. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*.

A Data Sampling

Building on this criteria, we selected presentations and documents that met specific requirements: presentations ranging from 12 to 64 pages and documents between 3 to 30 pages, with text lengths from 2,048 to 20,480 characters.

B Details of PPTEval

We recruited four graduate students from Shanghai through a crowdsourcing platform to evaluate a random sample of 50 presentations from *Zenodo10K*, along with 100 presentations each generated by the baseline method and our approach. The evaluations were conducted across three dimensions, as proposed by *PPEval*. To ensure consistency with LLM Judges, we provide the same scoring criteria E along with converted slide images.

Moreover, we listed some scoring examples in Figure 6.

C Layout Analysis

We detail our hierarchical clustering algorithm used in layout analysis at 1, where slides are grouped into clusters using a similarity threshold θ of 0.65

Moreover, we listed some extracted slide clusters at Figure 7

D Code Interaction

We have listed the APIs and their functions in Table 5.

An example of rendering a slide to HTML is shown in Figure 8.

E Prompts

E.1 Prompts for Presentation Analysis

The prompts used for presentation analysis are illustrated in Figures 9, 10, and 11.

E.2 Prompts for Presentation Generation

The prompts used for generating presentations are shown in Figures 12, 13, and 14.

E.3 Prompts for PPTEval

The prompts used in PPTEval are depicted in Figures 15, 16, 17, 18, 19 and 20.

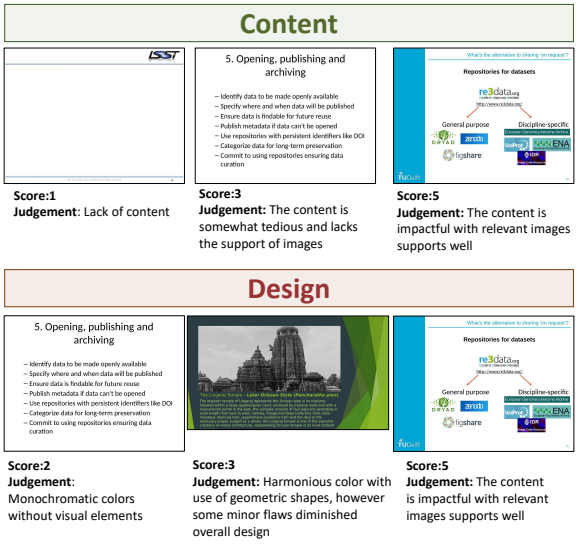

Figure 6: Scoring Examples of *PPTEval* .

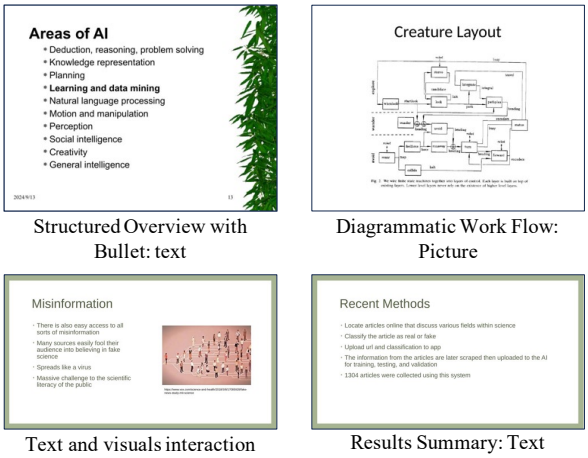

Figure 7: Example of slide clusters.

Algorithm 1 Slides Clustering Algorithm

Input: Similarity matrix of slides $S \in \mathbb{R}^{N \times N}$, similarity threshold θ
Initialize: $C \leftarrow \emptyset$
while $\max(S) \geq \theta$ **do**
  $(i, j) \leftarrow \arg\max(S)$
  if $\exists c_k \in C$ such that $(i \in c_k \vee j \in c_k)$ **then**
    $c_k \leftarrow c_k \cup \{i, j\}$
  else
    $c_{\text{new}} \leftarrow \{i, j\}$
    $C \leftarrow C \cup \{c_{\text{new}}\}$
  end if
  Update S:
    $S[:, i] \leftarrow 0$, $S[i, :] \leftarrow 0$
    $S[:, j] \leftarrow 0$, $S[j, :] \leftarrow 0$
end while
Return: C

```
<!DOCTYPE html>
<html>
<body style="width:780pt; height:540pt;">
<div id='0' data-relative-area='67.00%'>
<p id='0_3' >IP Issues in the Era of Artificial Intelligence </p>
<p id='0_8' >Prof. SpeakerName, Location</p>
<p id='0_14' >3rd EU China IP Academic Forum</p>
<p id='0_15' >Shanghai, November 22, 2018</p>
</div></body>
</html>
```

Figure 8: Example of rendering a slide into HTML format.

System Message:
You are an expert presentation analyst specializing in categorizing PowerPoint slides, particularly skilled at identifying structural slides (such as Opening, Transitions, and Ending slides) that guide the flow of the presentation. Please follow the specified output format strictly when categorizing the slides.

Prompt:
Objective: Analyze a set of slides provided in plain text format. Your task is to identify structural slides (such as Opening and Ending) based on their content and categorize all other slides under "Content."

Instructions:
1. Categorize structural slides in the presentation (such as Opening, Ending); assign all other slides to "Content."
2. Category names for structural slides should be simple, reflect their function, and contain no specific entity names.
3. Opening and Ending slides are typically located at the beginning or end of the presentation and may consist of only one slide.
4. Other transition categories must contain multiple slides with partially identical text.

Output format requirements:
Use the Functional key to group all categorized structural slides, with category names that reflect only the slide's function (e.g., "Opening," "Ending") and do not describe any specific content.
Use the Content key to list all slides that do not fall into structural categories.

Example output:
```json
{
  "functional": {
    "opening": [1],
    "table of contents": [2, 5],
    "section header": [3, 6],
    "ending": [10]
  },
  "content": [4, 7, 8, 9]
}
```

Ensure that all slides are included in the categorization, with their corresponding slide numbers listed in the output.

Input: {{slides}}

Output:

Figure 9: Illustration of the prompt used for clustering structural slides.

System Message:
You are a helpful assistant

Prompt:
Analyze the content layout and media types in the provided slide images.
Your objective is to create a concise, descriptive title that captures purely the presentation pattern and structural arrangement of content elements.
Requirements:
Focus on HOW content is structured and presented, not WHAT the content is
Describe the visual arrangement and interaction between different content types (text, images, diagrams, etc.)

Avoid:
Any reference to specific topics or subjects
Business or industry-specific terms
Actual content descriptions

You cannot use the following layout names:
{{ existed_layoutnames }}

Example Outputs:
Hierarchical Bullet Points with Central Image
Presentation of Evolution Through a Timeline
Analysis Displayed Using a Structured Table
Growth Overview Illustrated with Multiple Charts
Picture and illustrative key points
Layout
Output: Provide a one-line layout pattern title.

Figure 10: Illustration of the prompt used to infer layout patterns.

System Message:
You are a helpful assistant

Prompt:
Please analyze the slide elements and create a structured template schema in JSON format. The schema should:

1. Identify key content elements (both text and images) that make up the slide
2. For each element, specify:
  - "description": A clear description of the element's purpose, do not mention any detail
  - "type": "text" or "image" determined that according the tag of element: "image" is assigned for <img> tags
  - "data":
    * For text elements: The actual text content as string or array in paragraph level(<p> or <li>), merge inline text segments()
    * For image elements: Use the `alt` attribute of the <img> tag as the data of the image

Example format:
```
{
 "element_name": {
   "description": "purpose of this element", # do not mention any detail, just purpose
   "type": "text" or "image",
   "data": "actual text" or "<type>:<50-word description>" # detail here, cannot be empty or null
          or ["text1", "text2"]  # Multiple text elements
          or ["logo:...", "logo:..."]  # Multiple image elements
 }
}
```
Input:
{{slide}}
Please provide a schema that could be used as a template for creating similar slides.

Figure 11: Illustration of the prompt used to extract the slide schema.

System Message:
You are a professional presentation designer tasked with creating structured PowerPoint outlines. Each slide outline should include a slide title, a suitable layout from provided options, and concise explanatory notes. Your objective is to ensure that the outline adheres to the specified slide count and uses only the provided layouts. The final deliverable should be formatted as a JSON object. Please ensure that no layouts other than those provided are utilized in the outline.

Prompt:
Steps:

1. Understand the JSON Content:
  Carefully analyze the provided JSON input.
  Identify key sections and subsections.

{{ json_content }}

2. Generate the Outline:
  Ensure that the number of slides matches the specified requirement.
  Keep the flow between slides logical and ensure that the sequence of slides enhances understanding.
  Make sure that the transitions between sections are smooth through functional layouts.
  Carefully analyze the content and media types specified in the provided layouts.

For each slide, provide:
A Slide Title that clearly represents the content.
A Layout selected from provided layouts tailored to the slide's function.
Slide Description, which should contain concise and clear descriptions of the key points.

Please provide your output in JSON format.

Example Output:
```
{
  "Opening of the XX": {
    "layout": "layout1(media_type)",
    "subsection_keys": [],
    "description": "..."
  },
  "Introduction to the XX": {
    "layout": "layout2(media_type)", # select from given layouts(functional or content)
    "subsection_keys": ["Title of Subsection 1.1", "Title of Subsection 1.2"],
    "description": "..."
  }
}
```

Input:
Number of Slides: {{ num_slides }}
Image Information:
{{ image_information }}

you can only use the following layouts
Content Layouts:
{{ layouts }}
Functional Layouts:
{{ functional_keys }}

Output:

Figure 12: Illustration of the prompt used for generating the outline.

Function Name	Description
`del_span`	Deletes a specific span.
`del_image`	Deletes an image.
`clone_paragraph`	Creates a duplicate of an existing paragraph.
`replace_span`	Replaces the content of a specific span.
`replace_image`	Replaces an image with a new image.

Table 5: Definition and function of our provided APIs.

System Message:

You are an Editor agent for presentation content. You transform reference text and available images into structured slide content following schemas. You excel at following schema rules like content length and ensuring all content is strictly derived from provided reference materials. You never generate new content or use images not explicitly provided.

Prompt:

Generate slide content based on the provided schema.
Each schema element specifies its purpose, and its default quantity.

Requirements:
1. Content Generation Rules:
- Follow default_quantity for elements, adjust when necessary
- All generated content must be based on reference text or image information
- Ensure text content meets character limits
- Generated text should use concise and impactful presentation style
- For image elements, data should be the image path # eg: "images/logo.png"
- Type of images should be a critical factor of image selection, if no relevant image(similar type or purpose) provided, leave it blank

2. Core Elements:
- Must extract essential content from reference text (e.g., slide_title, main_content) and maintain semantic consistency
- Must include images that support the main content (e.g., diagrams for explanations, visuals directly discussed in text)

3. Supporting Elements (e.g., presenters, logo images):
- Generate only when relevant content exists in reference text or image information

Generate content for each element and output in the following format:

```
{
  "element1": {
    "data": ["text1", "text2"] for text elements
        or ["/path/to/image", "..."] for image elements
  },
}
```

Input:
Schema:
{{schema}}

Outline of Presentation:
{{outline}}

Metadata of Presentation:
{{metadata}}

Reference Text:
{{text}}

Available Images:
{{images_info}}

Output: the keys in generated content should be the same as the keys in schema

Figure 13: Illustration of the prompt used for generating slide content.

System Message:

You are a Code Generator agent specializing in slide content manipulation. You precisely translate content edit commands into API calls by following HTML structure, distinguishing between tags, and maintaining proper parent-child relationships to ensure accurate element targeting.

Prompt:

Generate the sequence of API calls based on the provided commands, ensuring compliance with the specified rules and precise execution.
You must determine the parent-child relationships of elements based on indentation and ensure that all and <img> elements are processed, leaving no unhandled content.

Each command follows this format: (element_class, type, quantity_change: int, old_data, new_data).

Steps

1. Quantity Adjustment:
- quantity_change Rules:
- If quantity_change = 0, do not perform clone_paragraph or del_span operations. Only replace the content.
- If quantity_change > 0, use clone_paragraph to add the corresponding number of paragraphs:
- When cloning, prioritize paragraphs from the same element_class that already have special styles (e.g., bold, color) if available.
- The paragraph_id for newly cloned paragraphs should be the current maximum paragraph_id of the parent element plus 1, while retaining the span_id within the cloned paragraph unchanged.
- If quantity_change < 0, use del_span or del_image to reduce the corresponding number of elements. Always ensure to remove span elements from the end of the paragraph first.
Restriction:
- Each command's API call can only use either clone_paragraph or del_span/del_image according to the `quantity_change`, but not both.
2. Content Replacement:
- Text Content: Use replace_span to sequentially distribute new content into one or more elements within a paragraph. Select appropriate tags for emphasized content (e.g., bold, special color, larger font).
- Image Content: Use replace_image to replace image resources.
3. Output Format:
- Add comments to each API call group, explaining the intent of the original command and the associated element_class.
- For cloning operations, annotate the paragraph_id of the newly created paragraphs.

Available APIs

{{api_docs}}

Example Input:

Please output only the API call sequence, one call per line, wrapped in ```python and ```, with comments for corresponding commands.

Figure 14: Illustration of the prompt used for generating editing actions.

System Message:

You are a help assistant

Prompt:

Please describe the input slide based on the following three dimensions:
1. The amount of information conveyed
Whether the slide conveys too lengthy or too little information, resulting in a large white space without colors or images.
2. Content Clarity and Language Quality
Check if there are any grammatical errors or unclear expressions of textual content.
3. Images and Relevance
Assess the use of visual aids such as images or icons, their presence, and how well they relate to the theme and content of the slides.

Provide an objective and concise description without comments, focusing exclusively on the dimensions outlined above.

Figure 15: Illustration of the prompt used to describe content in PPTEval.

System Message:

You are a help assistant

Prompt:

Please describe the input slide based on the following three dimensions:
1. Visual Consistency
Describe whether any style diminished the readability, like border overflow or blur, low contrast, or visual noise.
2. Color Scheme
Analyze the use of colors in the slide, identifying the colors used and determining whether the design is monochromatic (black and white) or colorful (gray counts in).
3. Use of Visual Elements
Describe whether the slide include supporting visual elements, such as icons, backgrounds, images, or geometric shapes (rectangles, circles, etc.).

Provide an objective and concise description without comments, focusing exclusively on the dimensions outlined above.

Figure 16: Illustration of the prompt used to describe style in PPTEval.

System Message:
You are an expert presentation content extractor responsible for analyzing and summarizing key elements and metadata of presentations. Your task is to extract and provide the following information:

Prompt:
Scoring Criteria (Five-point scale):
1. Slide Descriptions: Provide a concise summary of the content and key points covered on each slide.
2. Presentation Metadata: Identify explicit background information(which means it should be a single paragraph, not including in other paragraphs), such as the author, speaker, date, and other directly stated details, from the opening and closing slides.

Example Output:
```
{
  "slide_1": "This slide introduces the xx, xx.",
  "slide_2": "...",
  "background": {
     "speaker": "speaker x",
     "date": "date x"
  }
}
```

Input:
{{presentation}}

Output:.

Figure 17: Illustration of the prompt used to extract content in PPTEval.

System Message:
You are an unbiased presentation analysis judge responsible for evaluating the quality of slide content. Please carefully review the provided slide image, assessing its content, and provide your judgement in a JSON object containing the reason and score. Each score level requires that all evaluation criteria meet the standards of that level.

Prompt:
Scoring Criteria (Five-Point Scale):

1 Point (Poor):
The text on the slides contains significant grammatical errors or is poorly structured, making it difficult to understand.

2 Points (Below Average):
The slides lack a clear focus, the text is awkwardly phrased, and the overall organization is weak, making it hard to engage the audience.

3 Points (Average):
The slide content is clear and complete but lacks visual aids, resulting in insufficient overall appeal.

4 Points (Good):
The slide content is clear and well-developed, but the images have weak relevance to the theme, limiting the effectiveness of the presentation.

5 Points (Excellent):
The slides are well-developed with a clear focus, and the images and text effectively complement each other to convey the information successfully.

Example Output:
```
{
 "reason": "xx",
 "score": int
}
```
Input: {{descr}}
Let's think step by step and provide your judgment.

Figure 18: Illustration of the prompt used to evaluate content in PPTEval.

System Message:
You are an unbiased presentation analysis judge responsible for evaluating the visual appeal of slides. Please carefully review the provided description of the slide, assessing their aesthetics only, and provide your judgment in a JSON object containing the reason and score. Each score level requires that all evaluation criteria meet the standards of that level.

Prompt:
Scoring Criteria (Five-point scale):

1 Point (Poor):
There is a conflict between slide styles, making the content difficult to read.

2 Points (Fair):
The slide uses monotonous colors(black and white), ensuring readability while lacking visual appeal.

3 Points (Average):
The slide employs a basic color scheme; however, it lacks supplementary visual elements such as icons, backgrounds, images, or geometric shapes(like rectangles), making it look plain.

4 Points (Good):
The slide uses a harmonious color scheme and contains some visual elements(like icons, backgrounds, images, or geometric shapes); however, minor flaws may exist in the overall design.

5 Points (Excellent):
The style of the slide is harmonious and engaging, the use of supplementary visual elements like images and geometric shapes enhances the slide's overall visual appeal.

Example Output:
```
{
 "reason": "xx",
 "score": int
}
```

Input: {{descr}}
Let's think step by step and provide your judgment.

Figure 19: Illustration of the prompt used to evaluate style in PPTEval.

System Message:
You are an unbiased presentation analysis judge responsible for evaluating the coherence of the presentation. Please carefully review the provided summary of the presentation, assessing its logical flow and contextual information, each score level requires that all evaluation criteria meet the standards of that level.

Prompt:
Scoring Criteria (Five-Point Scale)

1 Point (Poor):
Terminology are inconsistent, or the logical structure is unclear, making it difficult for the audience to understand.

2 Points (Fair):
Terminology are consistent and the logical structure is generally reasonable, with minor issues in transitions.

3 Points (Average):
The logical structure is sound with fluent transitions; however, it lacks basic background information.

4 Points (Good):
The logical flow is reasonable and include basic background information (e.g., speaker or acknowledgments/conclusion).

5 Points (Excellent):
The narrative structure is engaging and meticulously organized with detailed and comprehensive background information included.

Example Output:
```
{
 "reason": "xx",
 "score": int
}
```

Input:
{{presentation}}

Let's think step by step and provide your judgment, focusing exclusively on the dimensions outlined above and strictly follow the criteria.

Figure 20: Illustration of the prompt used to evaluate coherence in PPTEval.