# PPT Co-Pilot

# Past Papers

## Enhancing Presentation Slide Generation by LLMs with a Multi-Staged End-to-End Approach

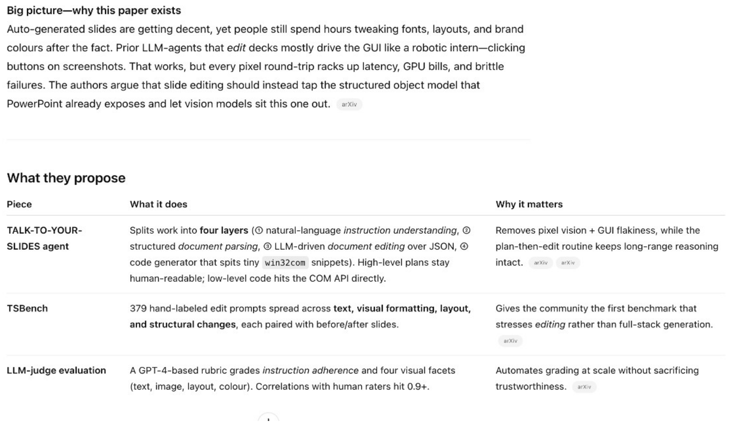
- Generating presentation from document containing multi-modal information (DocPres)
- Problem: context too long(hallucinations, failure at picking up context at middle) → divide doc into sections and subsections
- Uses LLM to create outline of presentation, generate bird’s eye view + K important topics w titles through chain-of-thought prompt
- Maps slides to outline(titles) ask LLM to associate section of document with title (uses edit distance, selects match if over 90% provides robustness against hallucinations)
- Concatenate content(text) from sections and feed to LLM in form of bullet points
- Uses heuristics + ranking algorithm to determine image and text similarity, VLM for image extraction
- Suitability score computed using cosine distance of CLIP embedding
- Better than gpt-3.5, limited by CLIP’s capabilities and multiple documents
- Human evaluation score based on desired qualities

# Past Papers

## PPTAgent: Generating & Evaluating Presentations Beyond Text-to-Slides

- https://github.com/icip-cas/PPTAgent
- “The key challenge lies in enhancing LLMs’ understanding of reference presentations’ structure and content patterns. Second, most presentations are saved in PowerPoint’s XML format, as demonstrated in Figure 11, which is inherently verbose and redundant (Gryk, 2022), making it challenging for LLMs to robustly perform editing operations.” Change XML format to HTML
- PPT has two stages, comprehensive analysis and edit APIs with HTML-rendered representation that simplifies slide modifications through code interaction
- PPTEval (MLLM-as-a-judge) evaluates presentations though content, design, and coherence
- Zenodo10k (presentation dataset)
- Stage 1: (slide clustering(2 types ~ content/structure), content extraction, slide element description)
- Stage 2: (Outline generation – (contains reference slide and relevant document content), iterative slide generation through edit-based API’s)
- Increase robustness, implement a self-correction mechanism using a REPL environment
- Framework asses content, design, and coherence using numeric scores 1-5.
- PPT Agent implementation consists of GPT-4o, Qwen2.5-72B Instruct + Qwen2
- Evaluation metrics: Success Rate, Perplexity, F1 score(Rouge-L), FID (similarity), PPTEval employs GPT-4o as judge
- Limitation → Incorporating visual information into generation process

# Past Papers

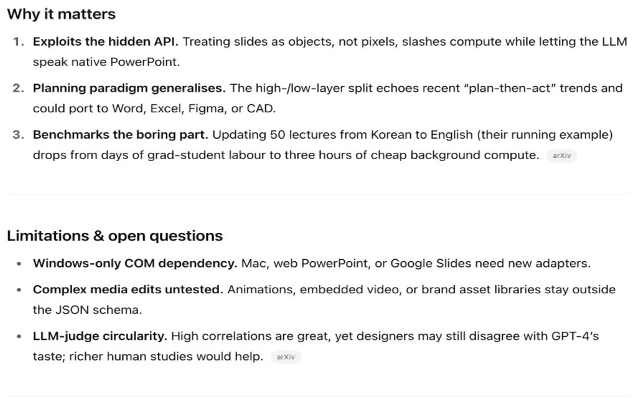


**Big picture—why this paper exists**

Auto-generated slides are getting decent, yet people still spend hours tweaking fonts, layouts, and brand colours after the fact. Prior LLM-agents that *edit* decks mostly drive the GUI like a robotic intern—clicking buttons on screenshots. That works, but every pixel round-trip racks up latency, GPU bills, and brittle failures. The authors argue that slide editing should instead tap the structured object model that PowerPoint already exposes and let vision models sit this one out. arXiv

### What they propose

| Piece | What it does | Why it matters |
|---|---|---|
| **TALK-TO-YOUR-SLIDES agent** | Splits work into **four layers** (① natural-language *instruction understanding*, ② structured *document parsing*, ③ LLM-driven *document editing* over JSON, ④ code generator that spits tiny `win32com` snippets). High-level plans stay human-readable; low-level code hits the COM API directly. | Removes pixel vision + GUI flakiness, while the plan-then-edit routine keeps long-range reasoning intact. arXiv arXiv |
| **TSBench** | 379 hand-labeled edit prompts spread across **text, visual formatting, layout, and structural changes**, each paired with before/after slides. | Gives the community the first benchmark that stresses *editing* rather than full-stack generation. arXiv |
| **LLM-judge evaluation** | A GPT-4-based rubric grades *instruction adherence* and four visual facets (text, image, layout, colour). Correlations with human raters hit 0.9+. | Automates grading at scale without sacrificing trustworthiness. arXiv |

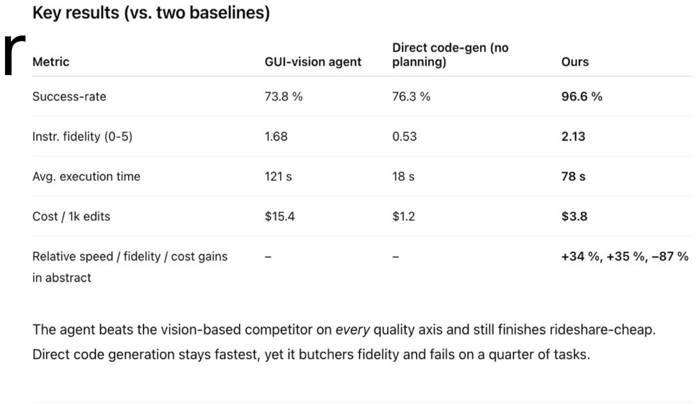


### Why it matters

1. **Exploits the hidden API.** Treating slides as objects, not pixels, slashes compute while letting the LLM speak native PowerPoint.
2. **Planning paradigm generalises.** The high-/low-layer split echoes recent "plan-then-act" trends and could port to Word, Excel, Figma, or CAD.
3. **Benchmarks the boring part.** Updating 50 lectures from Korean to English (their running example) drops from days of grad-student labour to three hours of cheap background compute. arXiv

### Limitations & open questions

- **Windows-only COM dependency.** Mac, web PowerPoint, or Google Slides need new adapters.
- **Complex media edits untested.** Animations, embedded video, or brand asset libraries stay outside the JSON schema.
- **LLM-judge circularity.** High correlations are great, yet designers may still disagree with GPT-4's taste; richer human studies would help. arXiv



### Key results (vs. two baselines)

| Metric | GUI-vision agent | Direct code-gen (no planning) | Ours |
|---|---|---|---|
| Success-rate | 73.8 % | 76.3 % | **96.6 %** |
| Instr. fidelity (0-5) | 1.68 | 0.53 | **2.13** |
| Avg. execution time | 121 s | 18 s | **78 s** |
| Cost / 1k edits | $15.4 | $1.2 | **$3.8** |
| Relative speed / fidelity / cost gains in abstract | – | – | **+34 %, +35 %, −87 %** |

The agent beats the vision-based competitor on *every* quality axis and still finishes rideshare-cheap. Direct code generation stays fastest, yet it butchers fidelity and fails on a quarter of tasks.

https://anonymous.4open.science/r/Talk-to-Your-Slides-0F4C/README.md

- If LLM Is the Wizard, Then Code Is the Wand: A Survey on How Code Empowers Large Language Models to Serve as Intelligent Agents
- AUTOPRESENT: Designing Structured Visuals from Scratch

# Other resources

- 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.

- Executable code actions elicit better llm agents.
- Evaluating large language models for powerpoint task completion

- https://plusai.com/use-cases/ai-for-teachers