AUToPRESENT: Automated Slide Generation from Natural Language

- AUToPRESENT: A tool-augmented code generation method
- Digital agents excel in software engineering
- Human experts spend significant time refining slides
- This study explores AI agents' slide creation capability

Press Space for next page →

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

SLIDESBENCH: The First Slide Generation Benchmark

- First benchmark for slide generation with 7k training and 585 testing examples
- Data collected from 310 slide decks across 10 domains (art, marketing, environment, technology, etc.)
- Supports both reference-based (similarity to target) and reference-free (design quality) evaluations
- Manual review ensures structured layouts and extractable media
- Slide URLs provided for download; creators can opt out of dataset inclusion

Task Formulation and Instruction Types

- NL-to-slide generation task: converting natural language instructions into editable presentations
- Three types of user instructions:
  - Detailed with images: provides all necessary content and assets
  - Detailed only: natural language instructions replace images with descriptions
  - High-level: general topical ideas requiring interpretation and execution
- Instruction word counts: 115.6 (detailed with images), 118.3 (detailed only), 26.6 (high-level)

Evaluation Methodology

- Two evaluation metric sets:
  - Reference-based: element matching, content similarity, color similarity, position similarity
  - Reference-free: design quality based on principles (0-5 scale, scaled to 0-100)
- Executability metric to evaluate model success rates
- GPT-4o used for reference-free scoring, validated by high human-model ICC agreement (73.8%-85.3%)
- Scores reported only for executable slides to ensure fair comparison

Technical Approach: Code Generation Method

- Slide generation via program generation: models generate Python programs from instructions
- Programs executed to create PPTX slides using libraries like Python-pptx
- Without helper functions: average 170 lines of complex code
- SLIDEsLIB library provides high-level functions for common actions, reducing programs to ~13 lines
- Includes 4 basic operation functions and 3 image-specific functions

AUToPRESENT Model Architecture

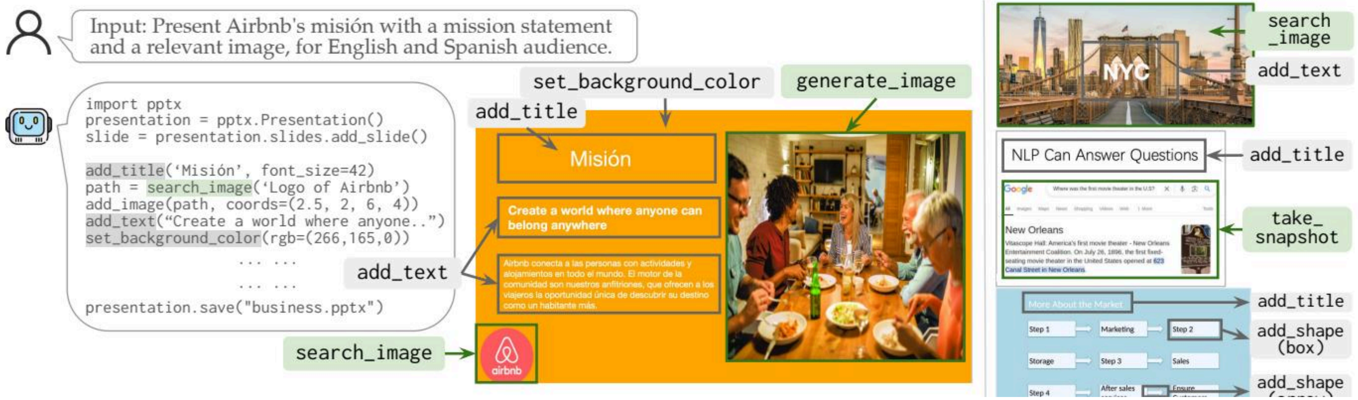
- 8B LLAMA-based model fine-tuned using LoRA on LLAMA-3.1-8BInstruct

- Trained on 7,000 instruction-program pairs from SLIDESBENCH

- Two types of canonical programs: basic Python and SLIDEsLIB-based

- Four specialized models trained to address different challenges

- Achieves performance comparable to closed-source GPT-4o



Experimental Results: Model Comparison

• End-to-end image generation methods (Stable Diffusion, DALL-E) produce creative but non-compliant visuals

• Code generation methods significantly outperform image generation approaches

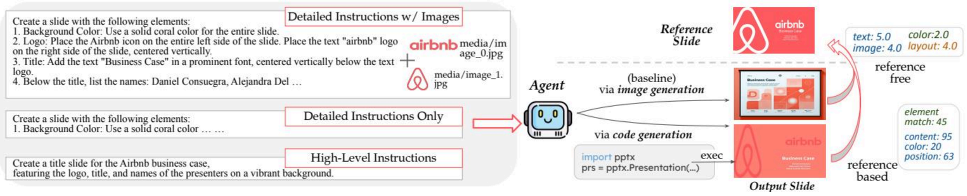
• Small models (LLaMA, LLaVA) perform poorly with minimal output

• GPT-4o achieves highest execution percentage (89.2%) and strong overall performance

• AUToPRESENT shows competitive results with 79.0% execution and balanced scores

SLIDEsLIB Impact and Refinement

- SLIDEsLIB significantly enhances performance in LLaMA and LLaVA models by up to 34.0 points
- Improves GPT-4o, particularly in image-free scenarios
- Iterative refinement process using GPT-4o to self-refine slides
- First iteration provides largest performance boost
- Refinement enhances content layout, coloring, and sizing controls



Performance Analysis

- Human-created reference slides achieve highest quality scores
- Performance gaps between GPT-4o and open-source models significant (49.9–55.0 points) with images
- Gaps reduce (22.2–34.6 points) when no visual information provided
- Obtaining appropriate images presents major challenge for models
- AUToPRESENT performance comparable to GPT-4o, unlike other open-weight models

Human Evaluation and User Preference

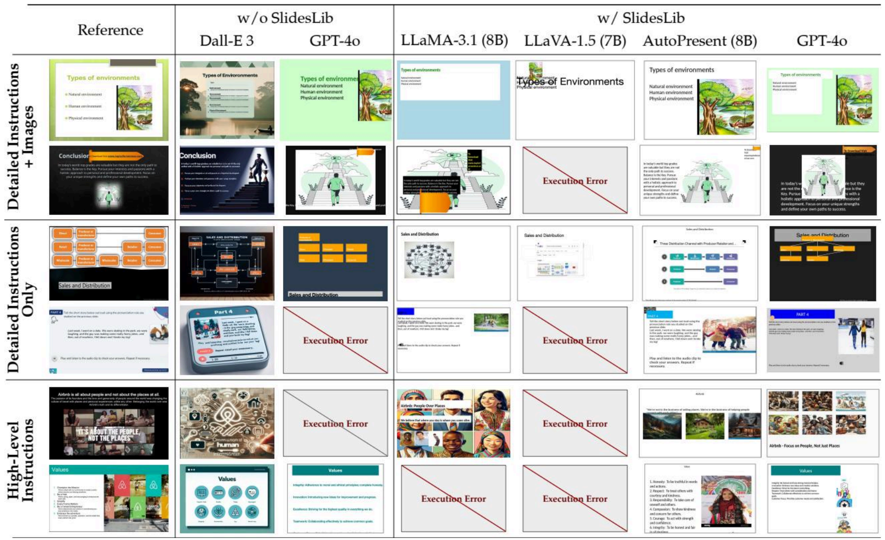
- Perceptual evaluation conducted on slides from GPT-4o, LLaMA-8B, and AUToPRESENT
- AUToPRESENT and GPT-4o perform significantly better than LLaMA-8B
- No significant differences between GPT-4o and AUToPRESENT in detailed+images setting
- AUToPRESENT slightly worse than GPT-4o in detailed-only setting
- All models have performance gap compared to human-designed slides

Limitations and Future Work

- Focuses on single-slide code generation in one pass
- Does not use iterative design workflows
- Future research should expand to full slide decks
- Suggests gradual and interactive slide generation
- Recommends incorporating slide-specific features like bullet points, animations, and transitions
- Emphasizes integrating design principles for attention and clarity

Related Work

- Previous slide generation works focus on extraction or content generation, not visual organization
- Recent methods address slide editing but not generation from scratch
- Program generation useful for structured visuals but requires detailed inputs
- End-to-end image generation excels in scenic images but struggles with structured visuals

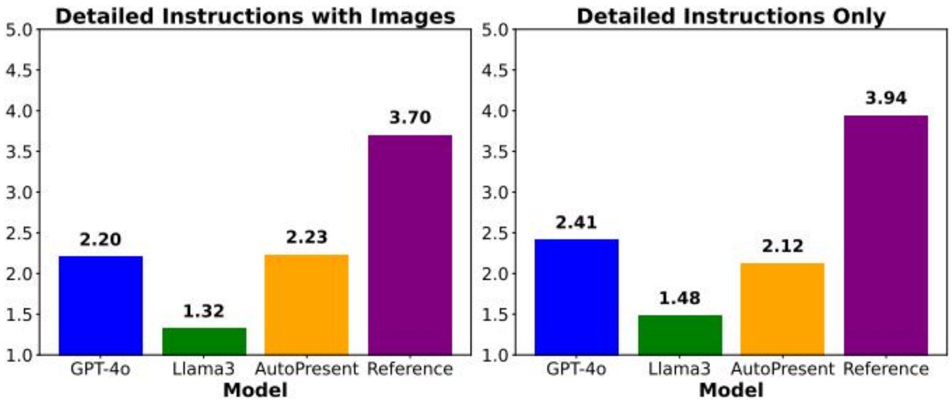


Conclusion

- AUToPRESENT introduces NL-to-slide generation and benchmarking
- SLIDESBENCH serves as foundation for future structured visual generation research
- Code generation methods produce higher-quality, user-interactable slides
- AUToPRESENT with SLIDESLIB achieves performance comparable to GPT-4o
- Iterative refinement shows effectiveness in self-improvement
- Initial step toward automated structured visual generation

Resources and Availability

- Code, data, demo, and videos available at provided GitHub link
- SLIDESBENCH benchmark with comprehensive evaluation metrics
- AUToPRESENT model open-sourced for research use
- SLIDEsLIB library available for slide program generation



Thank You

Questions?

GitHub Demo