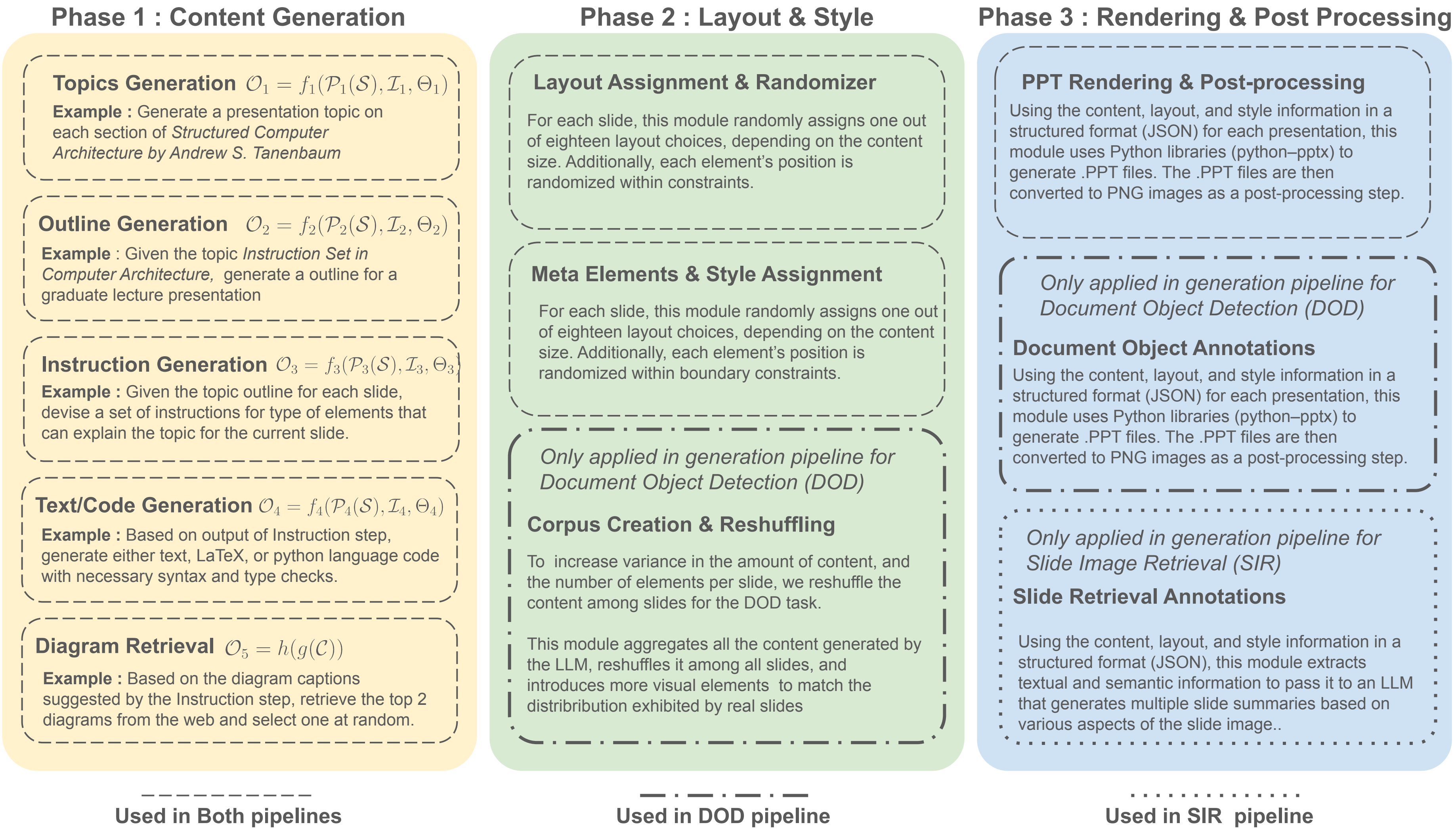

Phase 1 : Content Generation

Topics Generation $\mathcal{O}_1 = f_1(\mathcal{P}_1(\mathcal{S}), \mathcal{I}_1, \Theta_1)$

Example : Generate a presentation topic on each section of *Structured Computer Architecture by Andrew S. Tanenbaum*

Outline Generation $\mathcal{O}_2 = f_2(\mathcal{P}_2(\mathcal{S}), \mathcal{I}_2, \Theta_2)$

Example : Given the topic *Instruction Set in Computer Architecture,* generate a outline for a graduate lecture presentation

Instruction Generation $\mathcal{O}_3 = f_3(\mathcal{P}_3(\mathcal{S}), \mathcal{I}_3, \Theta_3)$

Example : Given the topic outline for each slide, devise a set of instructions for type of elements that can explain the topic for the current slide.

Text/Code Generation $\mathcal{O}_4 = f_4(\mathcal{P}_4(\mathcal{S}), \mathcal{I}_4, \Theta_4)$

Example : Based on output of Instruction step, generate either text, LaTeX, or python language code with necessary syntax and type checks.

Diagram Retrieval $\mathcal{O}_5 = h(g(\mathcal{C}))$

Example : Based on the diagram captions suggested by the Instruction step, retrieve the top 2 diagrams from the web and select one at random.

Phase 2 : Layout & Style

Layout Assignment & Randomizer

For each slide, this module randomly assigns one out of eighteen layout choices, depending on the content size. Additionally, each element's position is randomized within constraints.

Meta Elements & Style Assignment

For each slide, this module randomly assigns one out of eighteen layout choices, depending on the content size. Additionally, each element's position is randomized within boundary constraints.

Only applied in generation pipeline for Document Object Detection (DOD)

Corpus Creation & Reshuffling

To increase variance in the amount of content, and the number of elements per slide, we reshuffle the content among slides for the DOD task.

This module aggregates all the content generated by the LLM, reshuffles it among all slides, and introduces more visual elements to match the distribibution exhibited by real slides

Phase 3 : Rendering & Post Processing

PPT Rendering & Post-processing

Using the content, layout, and style information in a structured format (JSON) for each presentation, this module uses Python libraries (python–pptx) to generate .PPT files. The .PPT files are then converted to PNG images as a post-processing step.

Only applied in generation pipeline for Document Object Detection (DOD)

Document Object Annotations

Using the content, layout, and style information in a structured format (JSON) for each presentation, this module uses Python libraries (python–pptx) to generate .PPT files. The .PPT files are then converted to PNG images as a post-processing step.

Only applied in generation pipeline for Slide Image Retrieval (SIR)

Slide Retrieval Annotations

Using the content, layout, and style information in a structured format (JSON), this module extracts textual and semantic information to pass it to an LLM that generates multiple slide summaries based on various aspects of the slide image..

Used in Both pipelines

Used in DOD pipeline

Used in SIR pipeline