A. SLIDESBENCH Details

A.1. Slide Deck Domains

The 10 domains we cover in SLIDESBENCH include:

1. Art Photos
2. Business
3. Career
4. Design
5. Entrepreneur
6. Environment
7. Food
8. Marketing
9. Social Media
10. Technology

A.2. Slide Deck Source

There existis large amount of slide decks on the internet including Google Search, Bing Search etc. For convenience, we collect a list of slides from the `slideshare.com` website.

A.3. Slide Deck Statistics Per Domain

The *average images per domain* and *average text blocks per domain* are shown in Figure 6.

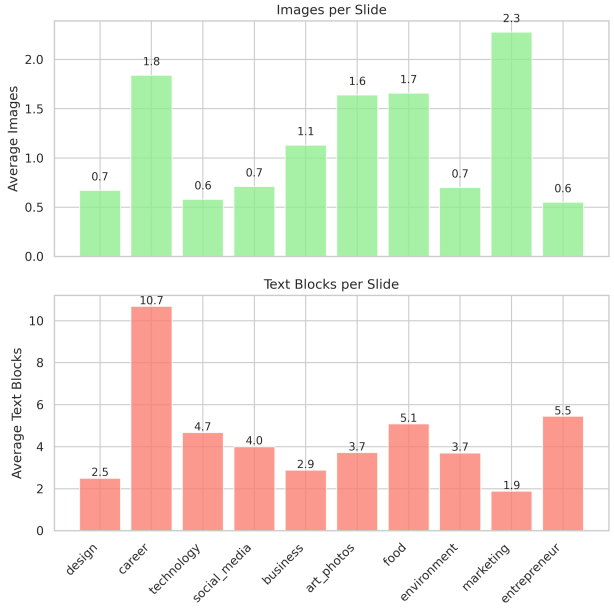

Figure 6. SlidesBench statistics on different domains.

B. SLIDESLIB Details

In this section, we provide the detailed documentation and examples for all functions in our SLIDESLIB.

B.1. SLIDESLIB Implementation

Figure 7 shows the basic functions and Figure 8 shows the image-oriented functions.

B.2. SLIDESLIB Usage Example

Figure 9 shows two example programs using multiple SLIDESLIB functions to produce slides.

B.3. SLIDESLIB Usage Percentage

The *percentage of each action* taken by GPT-4o, AUTOPRESENT, and Llama-3.1 in all 3 scenarios are reported in Figure 10. On average, the most common actions are `add_text` (36.3%), `add_image` (20.3%), and `add_title` (13.5%).

An example of the question is shown in Figure 11.

C. Training Details for AUTOPRESENT

The training parameters for AUTOPRESENT are summarized in Table 6.

Parameter	Value
LoRA Parameters	
LoRA rank	128
LoRA alpha	32
LoRA dropout	0
Random state	3407
RS-LoRA	Disabled
LoFT-Q config	None
Trainer Parameters	
Batch size (per device)	1
Gradient accumulation steps	2
Warmup steps	20
Epochs	1
Learning rate	3e-4
Mixed precision	FP16
Weight decay	0.01
Scheduler	Linear
Seed	3407

Table 6. Training details for AUTOPRESENT. LoRA and Trainer parameters are described in detail.

D. Refinement Details

We provide the prompts that we used for auto-refinement in Figure 12. We input the instruction and the in-context examples, the previous code generated by the model, and the snapshot of the slide generated by executing this code to the model and let it correct the code.

E. Detailed Results

We report two sets of evaluation metrics (reference-based and reference-free) in both their average value on all slides (i.e., un-weighted by execution success) and on successfully rendered slides (i.e., weighted by execution success).

E.1. Detailed Instructions with Images

Table 7 shows all metrics down-weighted by the execution success rate; Table 8 shows reference-based and reference-free metrics without down-weighting by execution success.

```
add_title(slide, text, font_size, font_color, background_color)
"""Add a title text to the slide with custom font size and font color (RGB tuple).
Args:
    slide: Slide object as in pptx library
    text: str, Title text to be added
    font_size: int, Font size in int (point size), e.g., 44
    font_color: tuple(int,int,int), RGB color, e.g., (0, 0, 0)
    background_color: Optional, tuple(int,int,int), RGB color, e.g., (255, 255, 255)
Rets:
    slide: Slide object with the title added
"""

add_text(slide, text, coords, font_size, bold, color, background_color, auto_size)
"""Add a text box at a specified location with custom text and color settings.
Args:
    slide: Slide object as in pptx library
    text: str, Text to be added
    coords: list(float), [left, top, width, height] in inches
    font_size: int, Font size in int (point size), e.g., 20
    bold: bool, True if bold-type the text, False otherwise
    color: tuple(int,int,int), RGB color, e.g., (0, 0, 0)
    background_color: Optional, tuple(int,int,int), RGB color, e.g., (255, 255, 255)
    auto_size: bool, True if auto-size the text box, False otherwise
Rets:
    slide: Slide object with the text box added
"""

add_bullet_points(slide, bullet_points, coords, font_size, color, background_color)
"""Add a text box with bullet points.
Args:
    slide: Slide object as in pptx library
    bullet_points: list(str), List of texts to be added as bullet points
    coords: list(float), [left, top, width, height] in inches
    font_size: int, Font size in int (point size), e.g., 18
    color: tuple(int,int,int), RGB color, e.g., (0, 0, 0)
    background_color: Optional, tuple(int,int,int), RGB color, e.g., (255, 255, 255)
Rets:
    slide: Slide object with the bullet points added
"""

add_image(slide, image_path, coords)
"""Add an image in the provided path to the specified coords and sizes.
Args:
    slide: Slide object as in pptx library
    image_path: str, Path to the image file
    coords: list(float), [left, top, width, height] in inches
Rets:
    slide: Slide object with the image added
"""

set_background_color(slide, color)
"""Set background color for the current slide.
Args:
    slide: Slide object as in pptx library
    color: tuple(int,int,int), RGB color, e.g., (255, 255, 255)
Rets:
    modified slide object
"""
```

Figure 7. Documentation for the basic functions in our SLIDESLIB.

E.2. Detailed Instructions Only

Table 9 shows all metrics down-weighted by the execution success rate; Table 10 shows reference-based and reference-free metrics without down-weighting by execution success.

E.3. High-Level Instructions Challenge

Table 11 shows all metrics down-weighted by the execution success rate; Table 12 shows reference-based and reference-free metrics without down-weighting by execution success.

```
google_search_screenshot(question, save_path)
"""Search a question on Google, and take a screenshot of the search result.
Save the screenshot to save_path, and return the path.
Args:
    question: str, The question to search on Google.
    save_path: str, The path to save the screenshot.
Returns:
    The path of the saved screenshot.
"""

search_image(query, save_path)
"""Search for an image on Google and download the result to save_path.
Args:
    query: str, The query to search for.
    save_path: str, The path to save the downloaded image.
Rets:
    the save_path.
"""

generate_image(query, save_path)
"""Generate an image using diffusion model based on a text query, and save the image to the path.
Args:
    query: str, The text query to generate the image.
    save_path: str, The path to save the generated image.
Rets:
    The path of the saved image
"""
```

Figure 8. Documentation for the image-oriented functions in our SLIDESLIB.

Method	Execution%	Reference-Based				Reference-Free				Average
		block	text	color	position	text	image	layout	color	
Human	100.0	-				59.7	81.5	73.5	65.7	-
Code Generation w/o Library										
LLaVA (7B)	11.3	7.0	11.0	0.7	8.0	4.7	11.3	3.3	2.9	6.1
LLaMA (8B)	2.1	1.5	1.9	0.3	1.7	1.0	0.2	1.0	1.0	1.3
GPT-4o	**89.2**	74.3	**80.7**	9.4	**68.7**	46.3	64.9	47.9	48.8	55.1
AUTOPRESENT (ours)	79.0	53.5	63.0	8.6	60.0	35.8	49.5	42.8	48.1	46.3
Code Generation w/ Expert-Designed Library										
LLaVA (7B)	20.0	16.1	16.1	0.7	12.8	7.5	9.6	5.9	8.7	9.7
LLaMA (8B)	54.4	42.6	49.6	4.1	37.8	25.0	37.1	25.9	28.9	33.5
GPT-4o	86.7	74.7	80.2	11.0	66.1	**47.3**	**72.5**	**61.1**	51.4	**58.0**
AUTOPRESENT (ours)	84.1	70.8	77.5	**15.2**	56.5	40.2	61.6	49.3	**54.4**	55.0

Table 7. Slide generation results (weighted by execution success) under the *detailed instructions with images* scenario.

Method	Execution%	Reference-Based				Reference-Free				Avg
		block	text	color	pos	text	img	layout	color	
Human	100.0	-				59.7	81.5	73.5	65.7	-
Code Generation w/o Library										
LLaVA (7B)	11.3	61.9	**97.3**	6.2	70.8	41.6	**100.0**	29.2	25.7	6.1
LLaMA (8B)	2.1	74.0	94.6	12.5	**81.2**	50.0	8.3	50.0	50.0	1.3
GPT-4o	**89.2**	83.3	91.6	10.5	77.0	51.9	72.8	53.7	54.7	55.1
AUTOPRESENT	79.0	67.7	79.7	10.9	75.9	45.3	62.7	54.2	60.9	46.3
Code Generation w/ Expert-Designed Library										
LLaVA (7B)	20.0	80.5	80.5	3.5	64.0	37.5	48.0	29.5	43.5	9.7
LLaMA (8B)	54.4	78.3	91.2	7.5	69.5	46.0	68.2	47.6	53.1	33.5
GPT-4o	86.7	**86.2**	92.5	12.7	76.3	**54.6**	83.7	**70.5**	59.4	**58.0**
AUTOPRESENT (ours)	84.1	84.2	92.2	**18.1**	67.2	47.8	73.2	58.6	**64.7**	55.0

Table 8. Slide generation results (un-weighted by execution success) under the *detailed instructions with images* scenario.

```
# Create slide with the title 'NLP Can Answer Questions' in large, bolded font in the top center of the
    page. Below it, put a screenshot of the google search result of the question 'Where was the first
    movie theater in the U.S?' in the middle of the page.

from pptx import Presentation
from pptx.util import Inches, Pt
from library import add_text, google_search_screenshot, add_image

presentation = Presentation()
presentation.slide_width = Inches(16)
presentation.slide_height = Inches(9)

slide_layout = presentation.slide_layouts[0] # choose a layout template
slide = presentation.slides.add_slide(slide_layout)
add_text(slide, "NLP Can Answer Questions", coords=(1, 0.5, 8, 1), font_size=36)
img_path = google_search_screenshot("Where was the first movie theater in the U.S?", save_path="
    screenshot.png")
add_image(slide, "screenshot.png", coords=(2.5, 2, 6, 4))
presentation.save("target_path.pptx")


# Create a slide titled 'Interior Design' in bold, dark-green color in the center of the page. For the
    background, consider using a picture with a color, artistic vibe, ensure enough contrast between
    the colors of text and background.

from pptx import Presentation
from pptx.util import Inches, Pt
from library import generate_image, add_image, add_text

presentation = Presentation()
presentation.slide_width = Inches(16)
presentation.slide_height = Inches(9)
slide_layout = presentation.slide_layouts[5] # choose a layout template
slide = presentation.slides.add_slide(slide_layout)

background_img = generate_image("An colorful, artistic background", "colorful.png")
add_image(slide, "colorful.png", coords=(0.0, 0.0, 16, 9))
add_text(slide, 'Interior Design', coords=(0.0, 2.4, 13.3, 1.3), font_size=80, bold=True, color=(0, 0,
    0), background_color=(255, 255, 255), auto_size=True)
presentation.save("path.pptx")
```

Figure 9. Example programs to produce slides using SLIDESLIB.

Method	Execution%	Reference-Based				Reference-Free				Average
		block	text	color	position	text	image	layout	color	
End-to-End Image Generation										
Stable-Diffusion	**100.0**	74.5	33.4	9.0	75.0	19.6	45.1	36.9	40.5	48.0
DALLE 3	**100.0**	**75.5**	39.9	9.2	76.1	32.7	**87.3**	56.7	53.4	50.2
Code Generation w/o Library										
LLaVA (7B)	17.9	12.2	16.3	1.4	12.4	7.9	15.3	5.7	5.0	9.5
LLaMA (8B)	4.6	63.0	87.0	**17.4**	**80.4**	30.4	19.6	41.3	47.8	2.8
GPT-4o	50.3	42.2	50.0	6.0	39.8	27.1	15.3	29.0	29.2	32.2
Code Generation w/ Expert-Designed Library										
LLaVA (7B)	17.4	15.6	15.5	0.9	10.5	5.7	6.2	4.1	7.5	8.3
LLaMA (8B)	60.5	45.1	55.5	5.2	43.6	29.5	44.3	29.6	33.4	37.4
GPT-4o	87.7	72.3	80.8	6.0	65.9	**46.6**	73.0	**58.5**	52.9	**56.3**
AUTOPRESENT (ours)	89.2	70.2	**82.7**	9.3	58.5	43.0	47.7	55.3	**63.2**	55.2

Table 9. Results (weighted by execution success) under *detailed instructions only* scenario.

Method	Execution%	Reference-Based				Reference-Free				Overall
		block	text	color	position	text	image	layout	color	
End-to-End Image Generation										
Stable-Diffusion	**100.0**	74.5	33.4	9.0	75.0	19.6	45.1	36.9	40.5	48.0
DALLE 3	**100.0**	75.5	39.9	9.2	76.1	32.7	**87.3**	56.7	53.4	50.2
Code Generation w/o Library										
LLaVA (7B)	17.9	68.2	91.1	7.8	69.3	44.1	85.8	31.8	27.9	9.5
LLaMA (8B)	4.6	2.9	4.0	0.8	3.7	1.4	0.9	1.9	2.2	2.8
GPT-4o	50.3	83.9	92.4	**11.9**	**79.1**	**53.9**	30.4	57.7	58.1	32.2
Code Generation w/ Expert-Designed Library										
LLaVA (7B)	17.4	**89.7**	89.1	5.2	60.3	32.8	35.6	23.6	43.1	8.3
LLaMA (8B)	60.5	74.5	91.7	8.6	72.1	48.8	73.2	29.6	48.9	37.4
GPT-4o	87.7	82.4	92.2	6.9	75.2	53.1	83.3	**66.7**	60.3	**56.3**
AUTOPRESENT (ours)	89.2	78.7	**92.7**	10.4	65.6	48.2	53.5	62.0	70.9	55.2

Table 10. Results (un-weighted by execution success) under *detailed instructions only* scenario.

Method	Execution%	Reference-Based				Reference-Free				Average
		block	text	color	position	text	image	layout	color	
End-to-End Image Generation										
Stable-Diffusion	**100.0**	72.0	33.2	8.3	77.2	3.3	49.3	35.6	37.8	47.7
DALLE 3	**100.0**	73.5	48.2	7.6	**77.3**	14.9	**89.7**	57.2	52.4	51.7
CodeGen-based Methods w/o Library										
LLaVA (7B)	19.5	14.9	13.2	1.7	13.6	8.0	16.8	5.9	6.2	10.0
LLaMA (8B)	8.7	7.6	6.3	0.7	4.7	4.6	2.4	5.0	5.4	4.8
GPT-4o	70.8	54.6	54.2	7.5	54.4	42.4	19.2	51.9	48.0	39.0
CodeGen-based Methods w/ Library										
LLaVA (7B)	25.1	20.4	17.8	1.6	15.4	9.2	9.7	6.9	11.0	11.5
LLaMA (8B)	76.9	55.4	58.3	5.6	55.7	39.5	56.5	40.3	43.0	43.7
GPT-4o	97.4	**77.0**	**75.8**	7.7	73.7	**59.7**	73.8	**78.7**	65.4	**58.5**
AUTOPRESENT (ours)	86.6	63.5	66.4	10.2	51.1	41.4	34.2	64.0	**73.3**	47.8

Table 11. Results (weighted by execution success) under *high-level instructions* scenario.

Method	Execution%	Reference-Based				Reference-Free				Average
		block	text	color	position	text	image	layout	color	
End-to-End Image Generation										
Stable-Diffusion	**100.0**	72.0	33.2	8.3	77.2	3.3	49.3	35.6	37.8	47.7
DALLE 3	**100.0**	73.5	48.2	7.6	**77.3**	14.9	**89.7**	57.2	52.4	51.7
CodeGen-based Methods w/o Library										
LLaVA (7B)	19.5	76.4	67.7	8.7	69.7	41.0	86.2	30.3	31.8	10.0
LLaMA (8B)	8.7	**87.4**	72.4	8.0	54.0	52.9	27.6	57.5	62.1	4.8
GPT-4o	70.8	77.1	76.8	10.6	76.8	59.9	27.1	73.3	67.8	39.0
CodeGen-based Methods w/ Library										
LLaVA (7B)	25.1	81.3	70.9	6.4	61.4	36.7	38.6	27.5	43.8	11.5
LLaMA (8B)	76.9	72.0	75.7	7.3	72.4	51.3	73.4	52.4	55.9	43.7
GPT-4o	97.4	79.0	**77.8**	7.9	75.6	**61.3**	75.8	**80.7**	67.1	**58.5**
AUTOPRESENT (ours)	86.6	73.3	76.7	**11.8**	59.0	47.8	39.5	73.9	**84.6**	47.8

Table 12. Results (un-weighted by execution success) under *high-level instructions* scenario.

F. Perceptual Analysis

In this section, we provide perceptual analysis details. We build a google doc and ask the user to score each slide from 1-5 (1 is the worst and 5 is the best), as shown in Figure 13.

The result of the paired t-test is shown in Table 13.

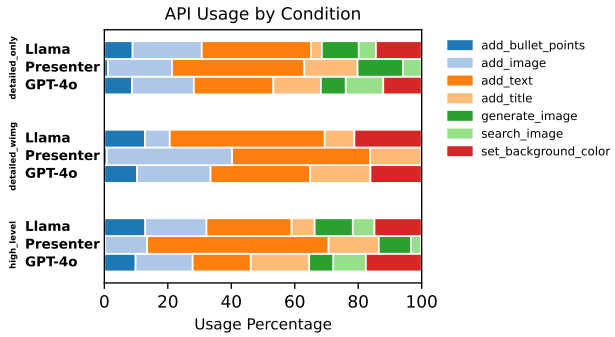

Figure 10. The percentage of each action taken by different models.

Figure 11. **An example of the perceptual analysis question.** We ask the human to score the quality of the slide from 1-5.

Model Pairs	Detailed+Images		Detailed Only	
	t-stat	p-val	t-stat	p-val
(GPT-4o, LLAMA)	13.206	0.000	8.630	0.000
(AUTOPRESENT, LLAMA)	13.180	0.000	2.955	0.004
(GPT-4o, AUTOPRESENT)	-0.445	0.657	8.203	0.000

Table 13. Paired t-test results comparing model performance across *detailed instruction only* setting and *detailed instruction with images* setting. AUTOPRESENT and GPT-4o outperforms LLAMA with a statistically significant difference in both settings.

```
"""
You are an expert presentation slides designer who creates modern, fashionable, and stylish slides
    using Python code. Your job is to generate the required PPTX slide by writing and executing a
    Python script. Make sure to follow the guidelines below and do not skip any of them:
1.  Ensure your code can successfully execute. If needed, you can also write tests to verify your code.
2. Maintain proper spacing and arrangements of elements in the slide: make sure to keep sufficient
    spacing between different elements; do not make elements overlap or overflow to the slide page.
3. Carefully select the colors of text, shapes, and backgrounds, to ensure all contents are readable.
4. The slides should not look empty or incomplete. When filling the content in the slides, maintain
    good design and layout.

Follow the instruction below to create the slide.
If the instruction is long and specific, follow the instruction carefully and add all elements as
    required;
if it is short and concise, you will need to create some content (text, image, layout) and implement it
     into the slide.
If you need to use the provided images, refer to the image file names in the instructions.

Finally, your code should save the pptx file to path "output.pptx"

API Libraries:
# INSERT_API_DESCRIPTIONS_HERE

## Examples
# INSERT_IN_CONTEXT_EXAMPLES_HERE

Modification Task:
Instruction: INSERT_INSTRUCTION_HERE

Previous Code:
INSERT_PREV_CODE_HERE

Slide Snapshot : See image.
Task: Based on the observed drawbacks in the provided slide image, modify the existing code accordingly
     to improve the slide's design and functionality.

Your modification:
def generate_presentation():
"""
```

Figure 12. **Prompt we used for Auto-Refinement.** The model receives the APIs and instruction, the previous generated slide and code, and is tasked to re-write the code to do slide refinement.

```
"""
Please score each slide from 1-5 based on your preference to use this slide in a real presentation. 5
    is the best, 1 is the worst.

Carefully reading each slide's content before ranking.
"""
```

Figure 13. **Instruction we used for the perceptual evaluation.**