Lightricks

NVIDIA & NLP IL

Visual Editing with LLM-based Tool Chaining

Oren Sultan – AI / NLP Researcher

Computer Science PhD Researcher @Hebrew University

AI Researcher @Lightricks

Lightricks

About Lightricks

We are the **go-to innovative photo and video creation platform,** enabling creators and brands to produce engaging, top-performing content.

Lightricks removes barriers to creation by building cutting edge tools based on the most advanced technology.

30M

Monthly active creators

100M

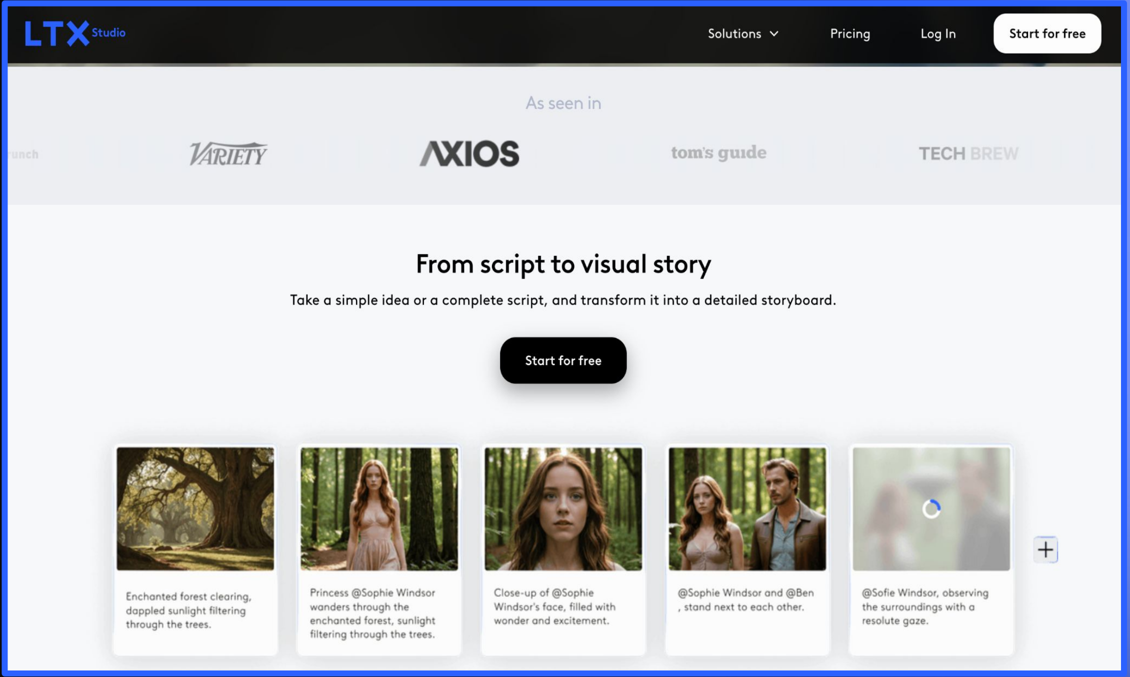
Monthly exports

Lightricks

AI-Powered Storytelling

LTX Studio Features

- AI Storytelling
- Scribble Creation Mode
- Character Consistency
- Character Dialogue
- Frame Control
- Face Motion Capture
- SFX
- Key Frame Control
- Style Reference
- Team Collaboration



From script to visual story

Take a simple idea or a complete script, and transform it into a detailed storyboard.

Enchanted forest clearing, dappled sunlight filtering through the trees.

Princess @Sophie Windsor wanders through the enchanted forest, sunlight filtering through the trees.

Close-up of @Sophie Windsor's face, filled with wonder and excitement.

@Sophie Windsor and @Ben , stand next to each other.

@Sofie Windsor, observing the surroundings with a resolute gaze.

LTX Studio by Lightricks

Lightricks

EMNLP 2024

hyadata lab

THE HEBREW UNIVERSITY OF JERUSALEM

“Visual Editing with LLM-based Tool Chaining: An Efficient Distillation Approach for Real-Time Applications”

Oren Sultan **Alex Khasin** **Guy Shiran** **Dr. Asi Messica** **Prof. Dafna Shahaf**

Lightricks

Background and Motivation

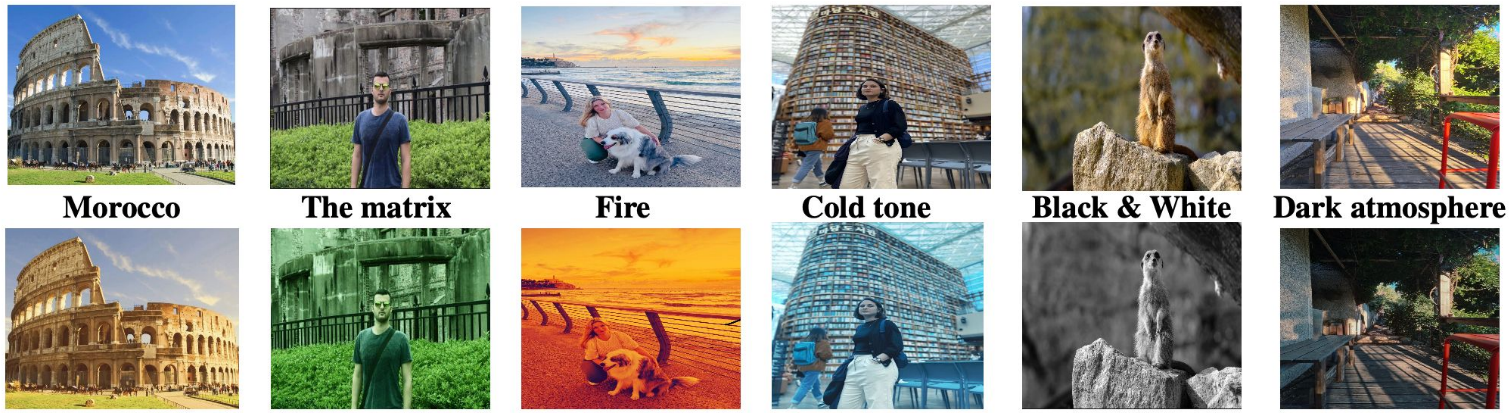
Lightricks

Visual Editing with LLM-based Tool Chaining

- **Idea.** to teach LLMs to use existing, **specialized tools** in **VideoLeap**

- **Goal.** to implement an AI assistant in VideoLeap, democratizing advanced capabilities.

- As a proof-of-concept, we focused on **tonal color adjustments**, allowing users to change a video's appearance via textual instructions.



Lightricks

Our Task: Visual Editing – Color Grading

Task.

Altering the appearance of an image or a video by adjusting its tonal colors, satisfying the user's request.

- **Input:** an asset (image/video) and a free text description of the requested appearance.
- **Output:** the altered image or video, based on the request.

Challenges.

- Interpreting user stylistic requests, specified in natural language (e.g., "The matrix", "Morocco").
- Choosing the appropriate tools and their parameters to achieve the desired visual effect.

$$f : I \rightarrow O$$

$$O = \{(T_i, P_i) \mid T_i \in \mathcal{T}, P_i \in \mathcal{P}(T_i)\}$$

$$P_i = \{(p_{i_1}, v_{i_1}), (p_{i_2}, v_{i_2}), \ldots, (p_{i_n}, v_{i_n})\}$$

Example: **“Golden hour”**

Adjust

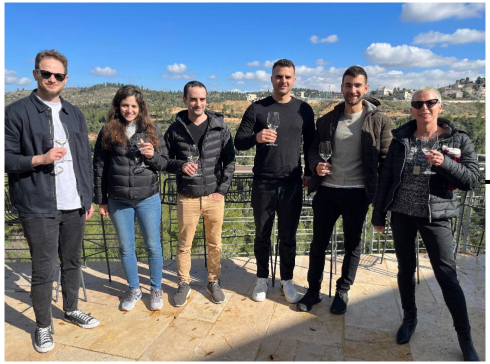
```
{
 "exposure": 0,
 "contrast": 10,
 "brightness": 10,
 "highlights": 20,
 "shadows": -10,
 "saturation": 15,
 "vibrance": 15,
 "temperature": 30,
 "tint": 10,
 "hue": 0,
 "bloom": 0,
 "sharpen": 0,
 "structure": 0,
 "linearOffset": 0
}
```

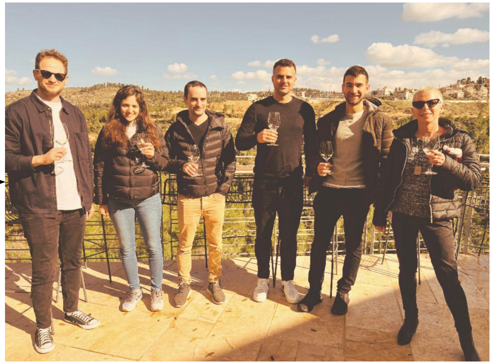
Selective adjust

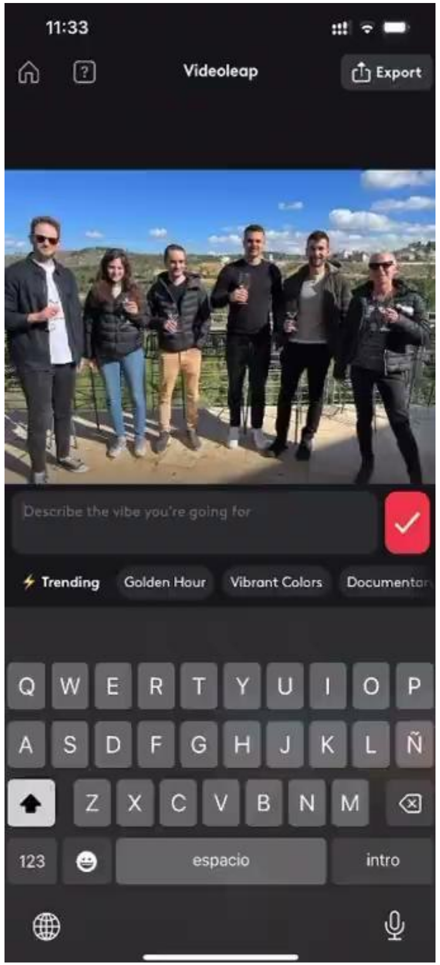
```
{
 "red": {"saturation": 20, "luminance": 10},
 "orange": {"saturation": 30, "luminance": 20},
 "yellow": {"saturation": 40, "luminance": 30},
 "green": {"saturation": -20, "luminance": 0},
 "cyan": {"saturation": -20, "luminance": 0},
 "blue": {"saturation": 0, "luminance": 0}
}
```

Filter

```
{
 "name": "faded_HighNoon",
 "intensity": 40
}
```

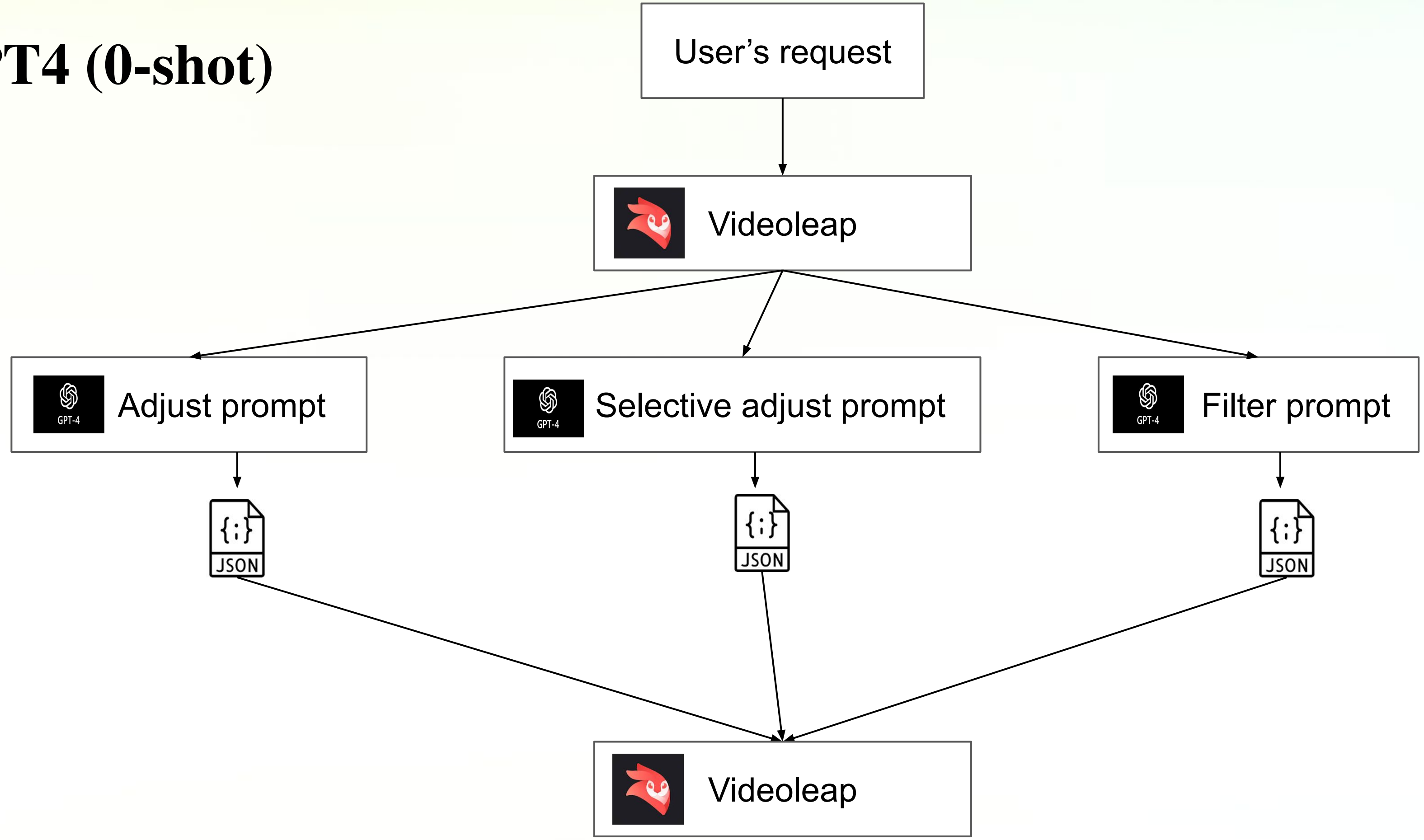

Source


“Golden hour” filter look



Lightricks

GPT4 (0-shot)



Lightricks

Prompt Engineering

GPT4 (1-shot).

- Shorter prompts and shorter responses → #tokens reduction → cost and latency reduction.
- ~50% reduction in #tokens and response time for every prompt.
- 100% validity of output json format.

GPT-3.5-turbo (0-shot).

- Does not produce response with a correct json format.
- After changes in the prompt, it produces a correct json format, but with wrong parameters.

GPT-3.5-turbo (1-shot).

- Produce a correct json format, with correct parameters.

GPT-3.5-turbo (1-shot) vs. GPT4 (0-shot).

- ~20x times cheaper.
- ~2x times faster than GPT4.
- Shows reasonable performance.

Lightricks

A 1-shot, Chain-of-Thought (CoT) prompt for the Adjust tool

You are a professional image and video editor. Your goal is to make the color adjust filters based on the user's request. The standard tools you can use are: global color grading tool which works globally on all colors, selective color grading (separate adjust for different color ranges) and LUT filter presets. **Suggest how to use these tools to achieve the requested look. The tools should be combined and work well when applied together. Write in TOOL whether you should use the color grading tool. If yes, suggest how.**
PARAMETERS:
"exposure", "contrast", "brightness", "highlights", "shadows", "saturation", "vibrance", "temperature", "tint", "linearOffset" (values between -100 and 100), "hue" (values between 0 and 360), and "bloom", "sharpen", "structure" (values between 0 and 100). Is the request asked to apply the look on a specific color range? If so then don't change the "hue" parameter of the global color grading. If all parameters are zero, then don't use this tool. **If you decide to use the global color grading tool, write in JSON only parameters from PARAMETERS with their correct values. Otherwise, write an empty json.**
Inputs: USER_REQUEST
Outputs: TOOL, JSON

Inputs:
USER_REQUEST: a golden hour filter look
Outputs:
TOOL: Yes, the global color grading tool should be used. **The temperature should be increased to add a warm, golden tone to the image**. The highlights and shadows should be adjusted to soften the image and mimic the diffused light of the golden hour. The saturation could also be increased to enhance the colors. The hue parameter should not be changed as the request does not specify a specific color range.
JSON: { "exposure": 0, "contrast": 0, "brightness": 0, "highlights": 20, "shadows": -20, "saturation": 10, "vibrance": 0, **"temperature": 30**, "tint": 0, "hue": 0, "bloom": 0, "sharpen": 0, "structure": 0, "linearOffset": 0}

Lightricks

A 1-shot, Chain-of-Thought (CoT) prompt for Selective Adjust tool

You are a professional image and video editor. Your goal is to make the color adjust filters based on the user's request. The standard tools you can use are: global color grading tool which works globally on all colors, selective color grading (separate adjust for different color ranges) and LUT filter presets. **Suggest how to use these tools to achieve the requested look. The tools should be combined and work well when applied together. The parameters are:** "red", "orange", "yellow", "green", "cyan", "blue". For each we have "saturation" and "luminance" (both between -100 and 100). **Write in TOOL whether you should use the selective color grading tool. If yes, suggest how.**
Write in JSON the parameters for selective color grading if you decide to use the selective color grading tool. You should write the colors ("red", "orange", "yellow", "green", "cyan", "blue"), and a value for each one of the "saturation", "luminance" parameters.
If you decide to use the global color grading tool or LUT filter presets instead, write an empty json.
Inputs: USER_REQUEST
Outputs: TOOL, JSON

Inputs:
USER_REQUEST: a golden hour filter look
Outputs:
TOOL: Yes, the selective color grading tool should be used. We would enhance the reds, oranges, and yellows to create the warm golden hour look. **We would also slightly reduce the saturation and luminance of the blues and greens to make the warm colors stand out more.**
JSON: {"red": {"saturation": 20, "luminance": 10},
"orange": {"saturation": 30, "luminance": 20},
"yellow": {"saturation": 40, "luminance": 30},
"green": {"saturation": -20, "luminance": -10},
"cyan": {"saturation": -20, "luminance": -10},
"blue": {"saturation": -30, "luminance": -20}}

Lightricks

A 1-shot, Chain-of-Thought (CoT) prompt for the Filter tool

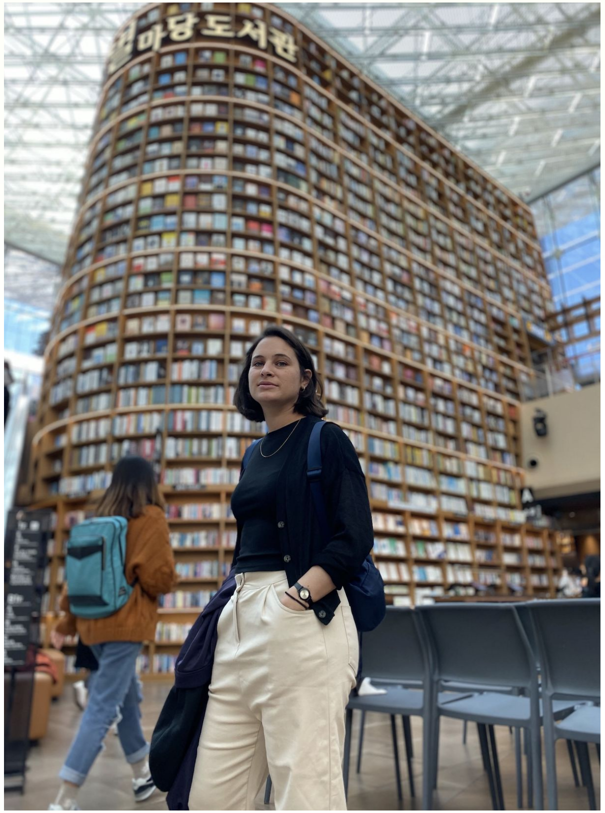
You are a professional image and video editor. Your goal is to make the color adjust filters based on the user's request. The standard tools you can use are: global color grading tool which works glob- ally on all colors, selective color grading (separate adjust for different color ranges) and LUT filter presets. **Available LUT presets are:** "none", "lovely_day", "action", "vivid", "north", "purple_rain", "winter", "faded_Retro", "faded_HighNoon", "faded_Mist", "faded_Terra", "faded_Vista", "faded_C1", "faded_AL2", "teal_and_orange_1", "teal_and_orange_1" "teal_and_orange_2", "teal_and_orange_3", "teal_and_orange_4", "teal_and_orange_5", "teal_and_orange_6", "fortune", "duotone_red", "spring", "duotone_pink", "enchanted", "duotone_green", "ultra", "duotone_yellow", "firecracker", "duo- tone_orange", "cyberpunk", "darkness", "night_vision", "negative". **Suggest how to use these tools to achieve the requested look. The tools should be combined and work well when applied together. Write in TOOL whether the LUT preset is suitable for the requested look. The default preset is "none". Only use other than "none" LUTs if you find a very close or exact preset. Otherwise use "none" preset.** If duotone was not explicitly mentioned in the request, but you still choose to use a duotone filter, then use very low intensity of 15. Suggest a specific preset from the list. Try to find a preset name that is semantically close to the requested look. The closer the preset, the more intense it should be. If the semantic match is quite close, but not exact use intensity of around 40 to 60. For example for "cool morning" you can choose the presets named "north" or "winter" because both associated with cold, but the intensity should be low, about 40 to 60 because cool is not exactly north or winter. But if the request contains the word "winter" then you should choose the winter LUT with high intensity of around 80. Write in JSON the correct LUT filter preset from the available LUT presets, and the correct intensity between 0 and 100.
Inputs: USER_REQUEST
Outputs: TOOL, JSON

Inputs:
USER_REQUEST: a welding mask filter look
Outputs:
TOOL: The "night_vision" LUT preset seems the most appropriate since night vision also tends to produce greenish images.
JSON: {**"name": "night_vision"**, "intensity": 60}

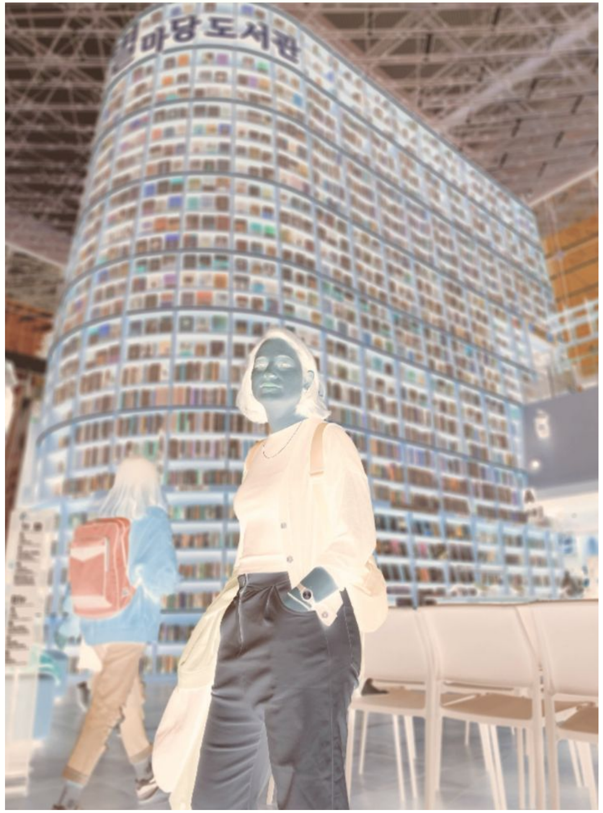
Lightricks

Blind Experiment – which model produces a better result?

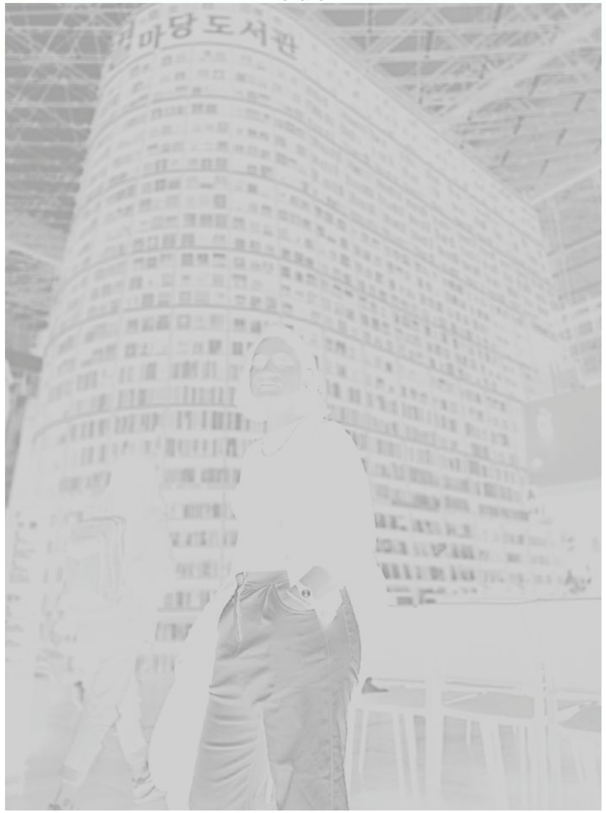
(1) User’s prompt: **X ray**



Model A

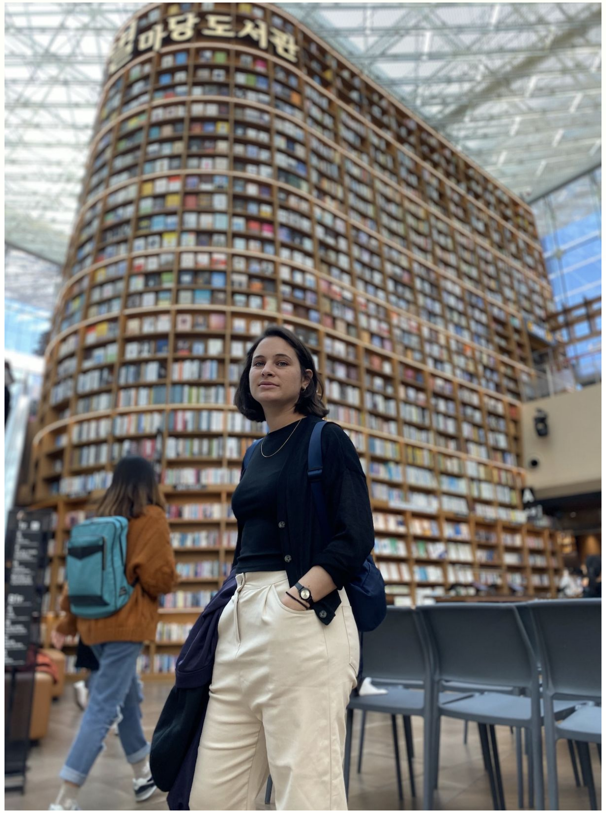


Model B

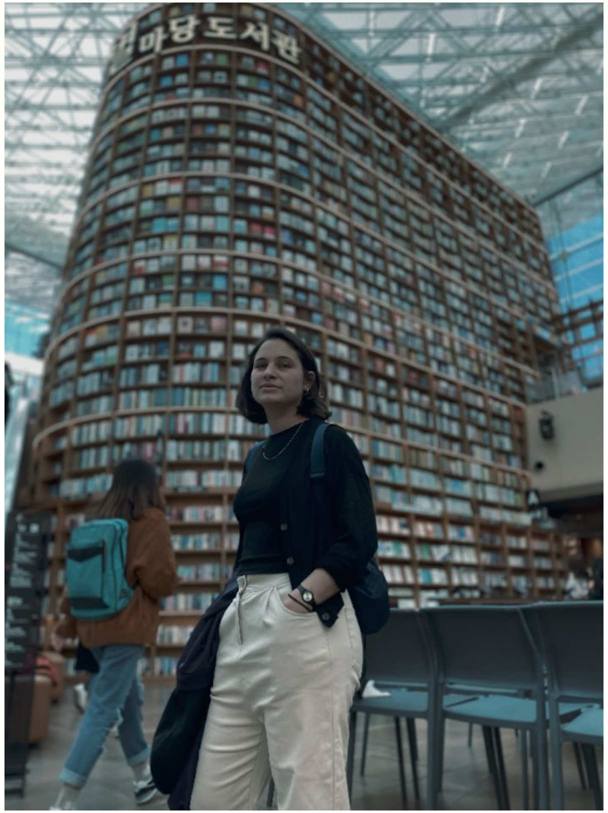


Lightricks

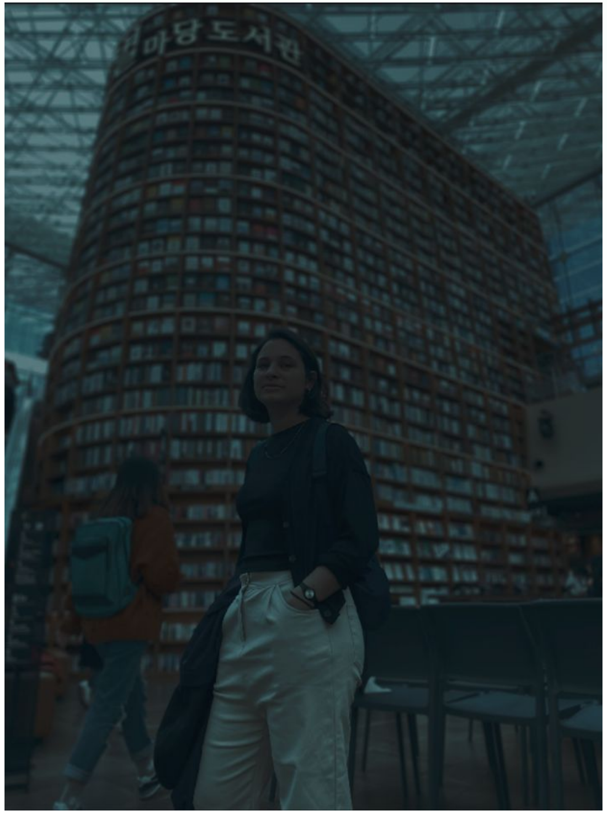
(2) User's prompt: **Dark Atmosphere**



Model A

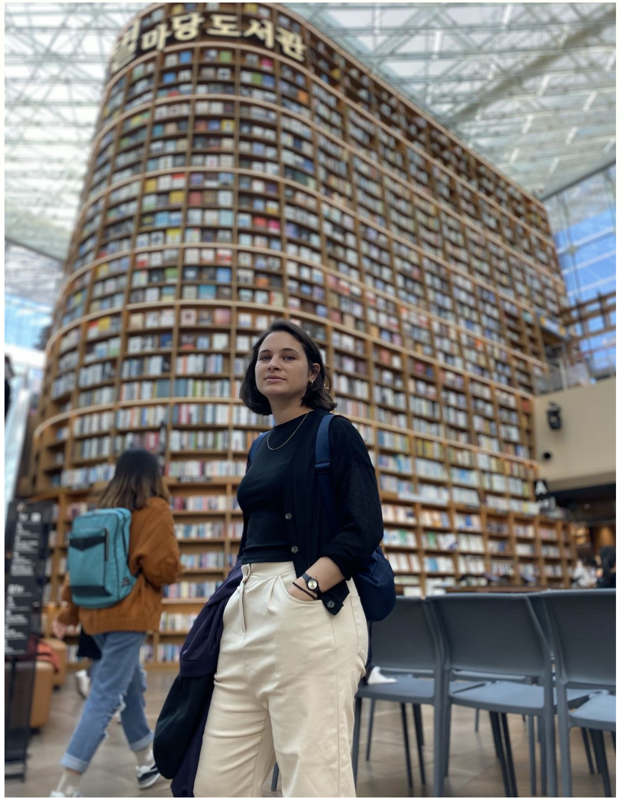


Model B

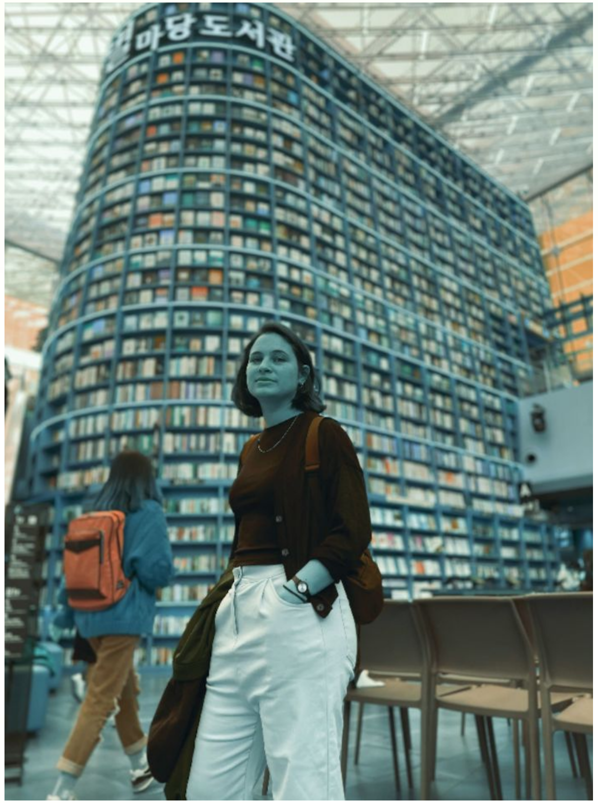


Lightricks

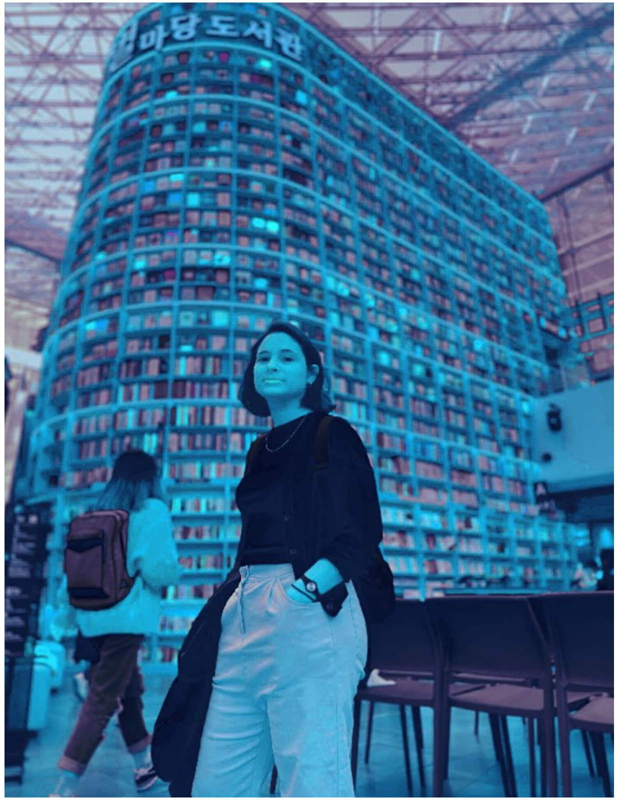
(3) User’s prompt: 😈



Model A

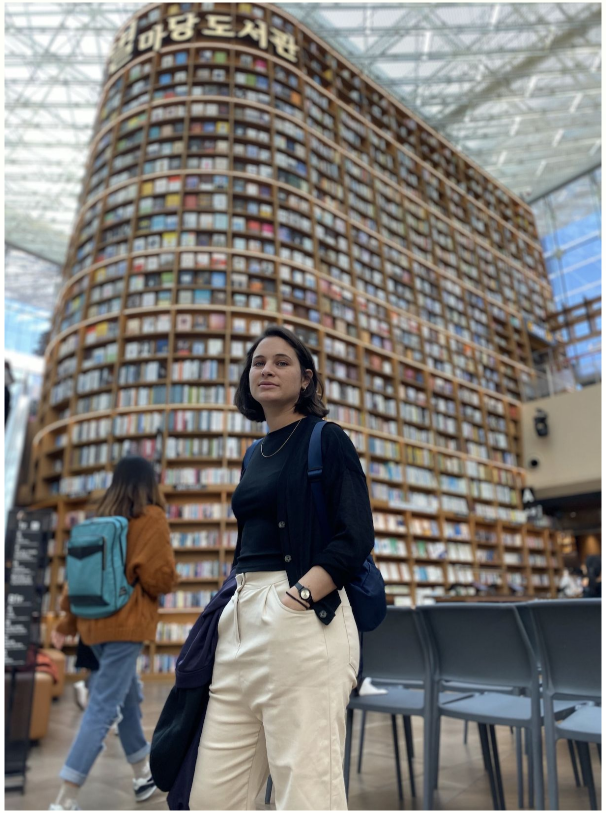


Model B

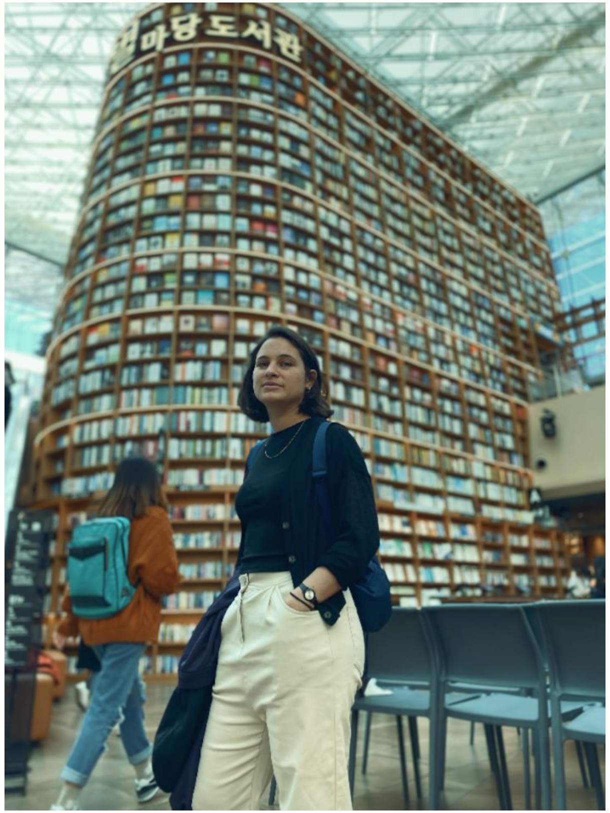


Lightricks

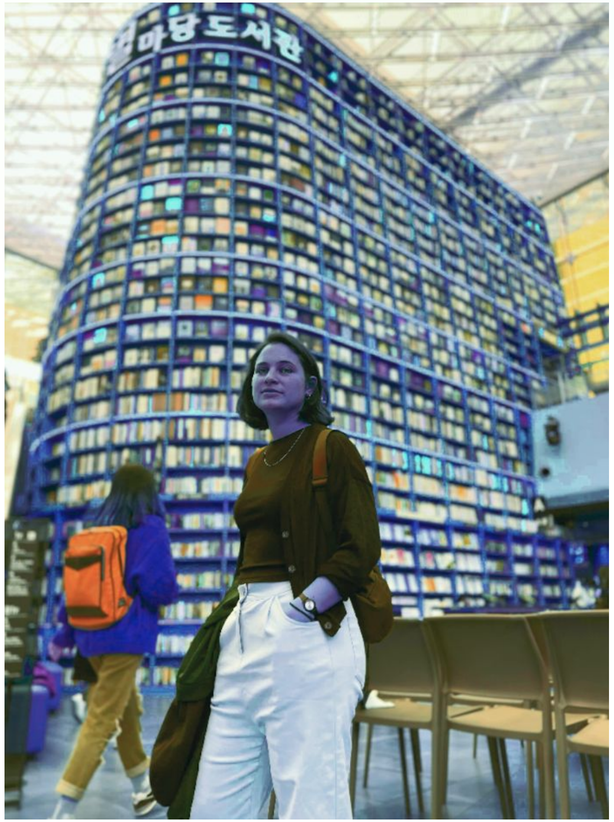
(4) User's prompt: **Neon**



Model A

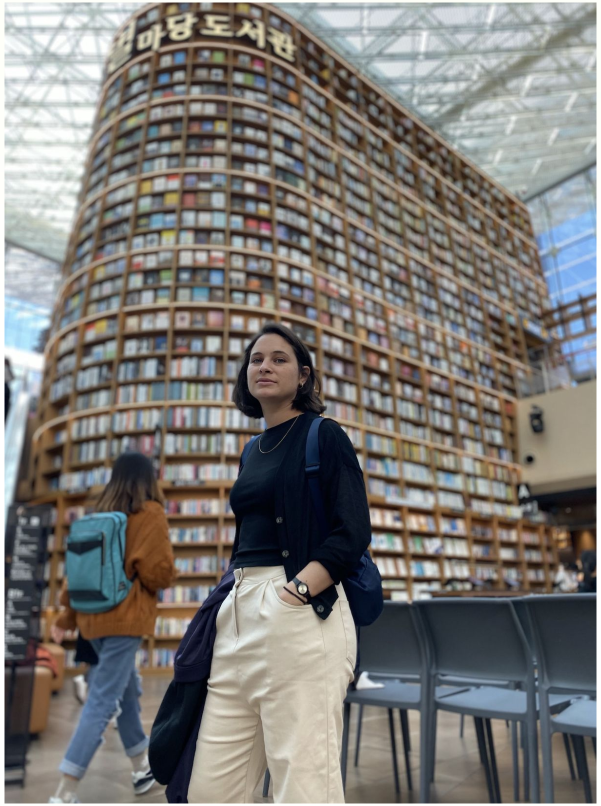


Model B

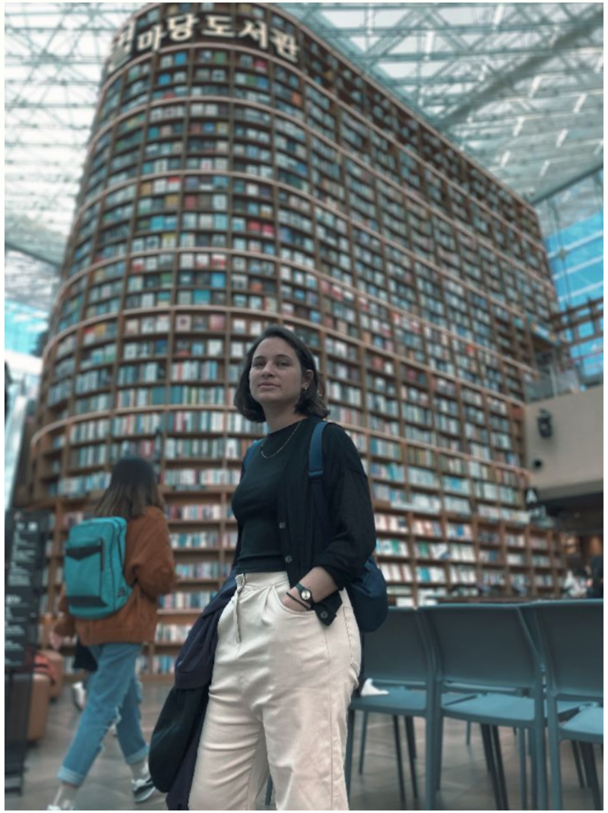


Lightricks

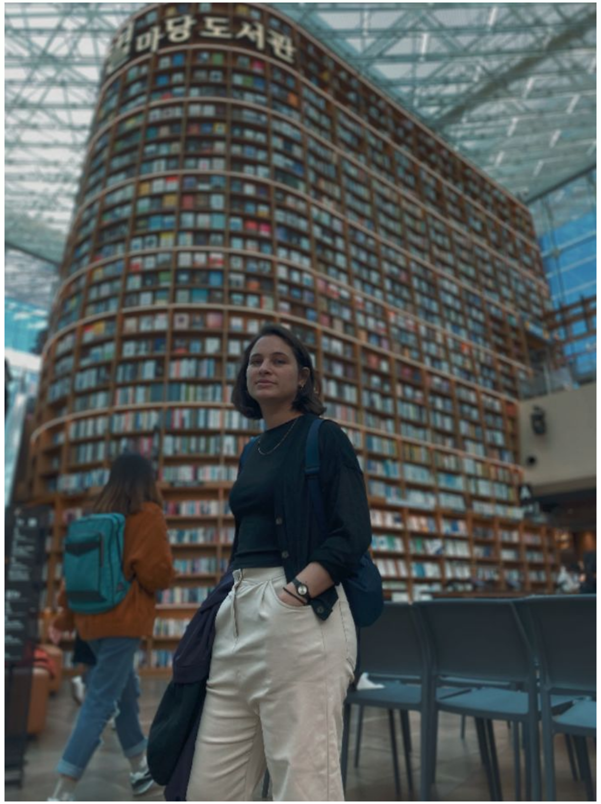
(5) User’s prompt: **Shadow Depth**



Model A

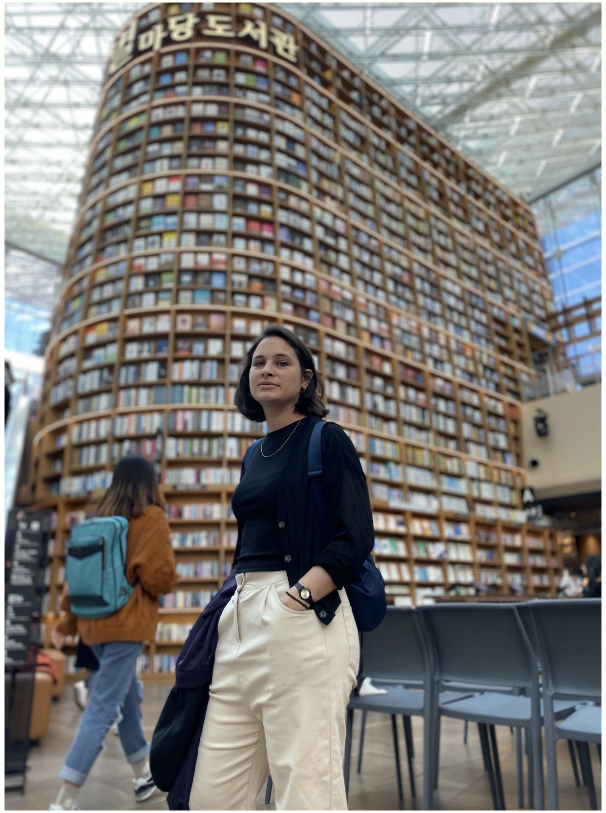


Model B

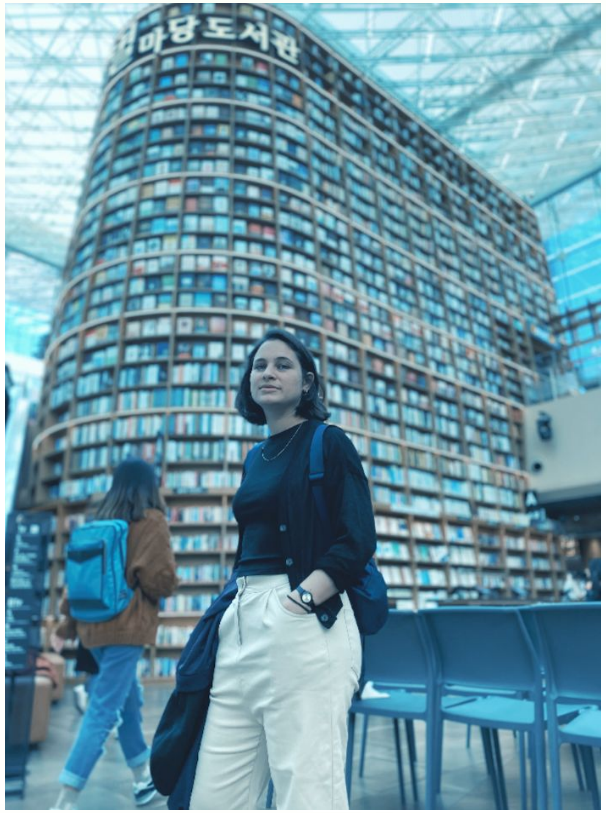


Lightricks

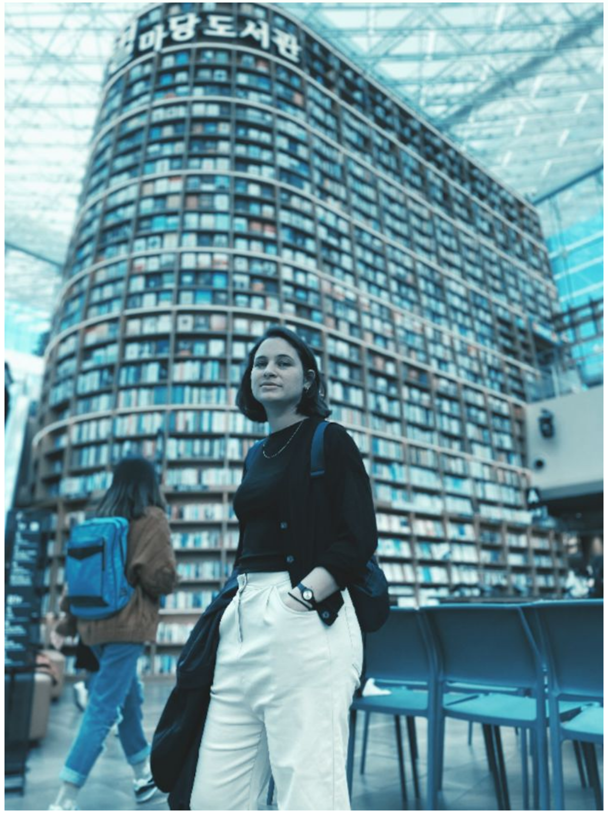
(6) User's prompt: **Cold Tone**



Model A

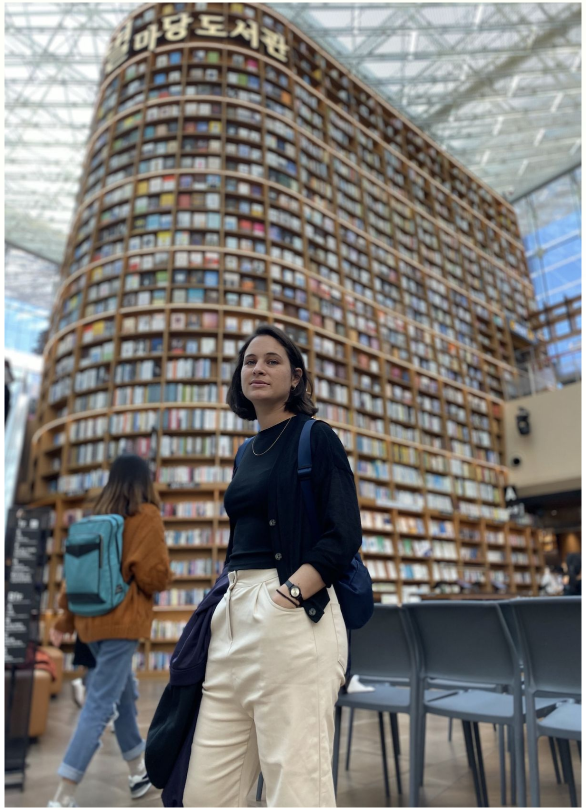


Model B

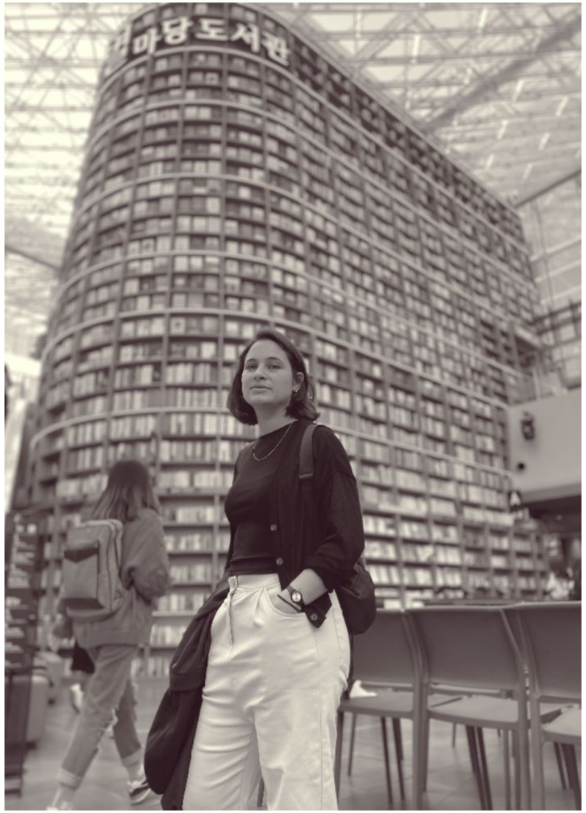


Lightricks

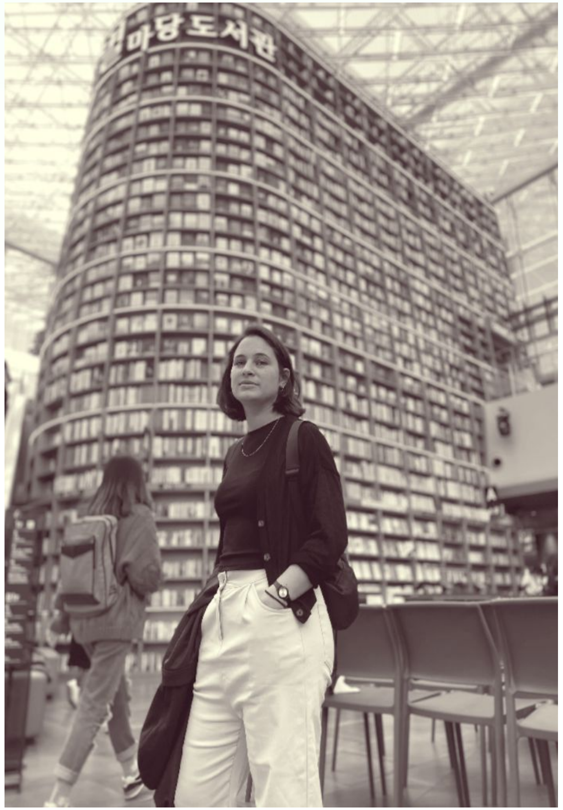
(7) User's prompt: **Charlie Chaplin**



Model A

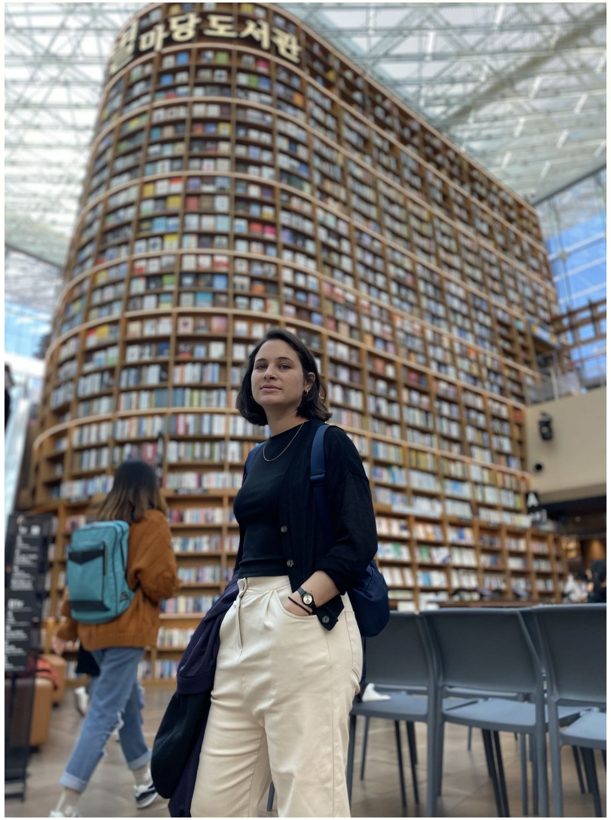


Model B

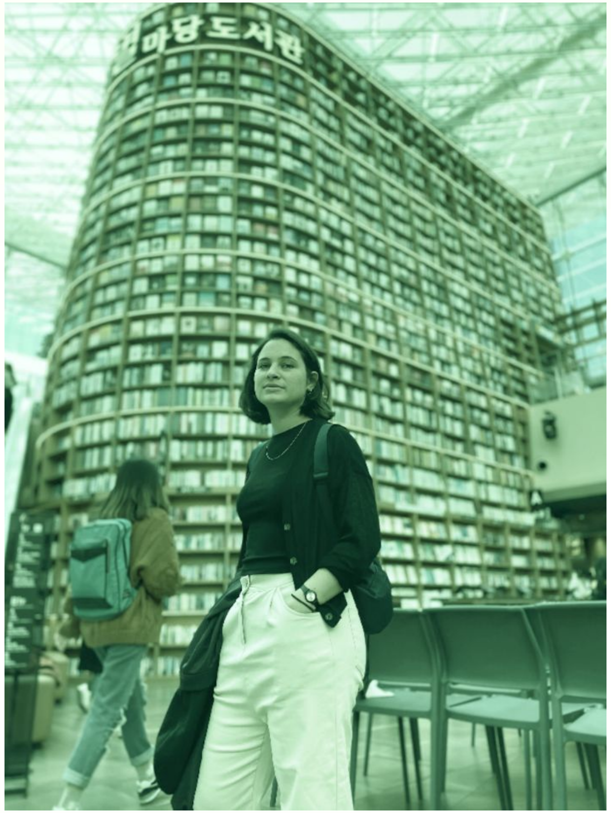


Lightricks

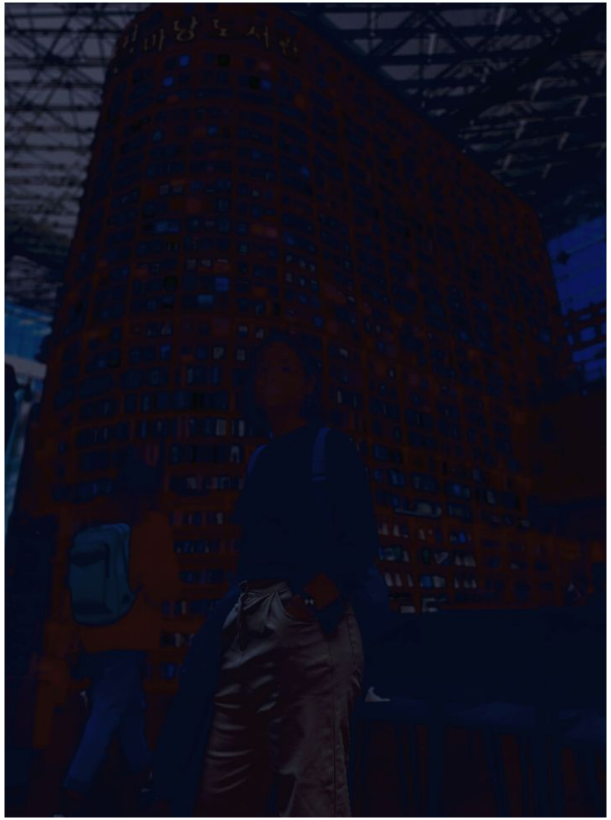
(8) User's prompt: **welding mask**



Model A

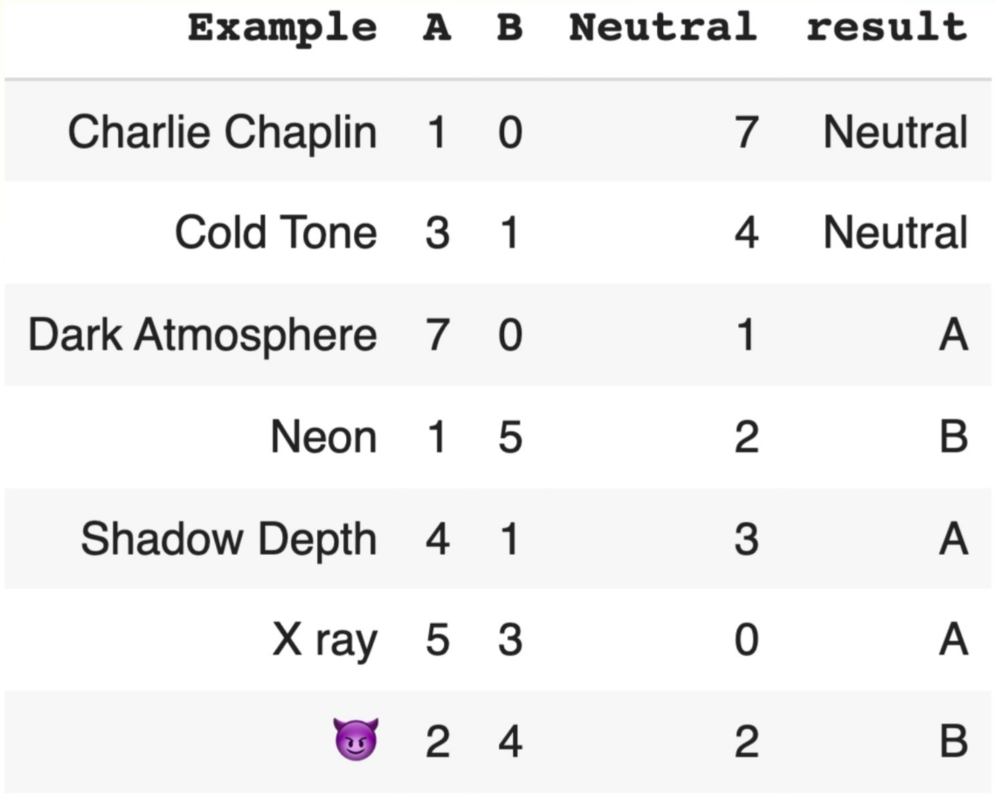


Model B



Lightricks

Results



Example	A	B	Neutral	result
Charlie Chaplin	1	0	7	Neutral
Cold Tone	3	1	4	Neutral
Dark Atmosphere	7	0	1	A
Neon	1	5	2	B
Shadow Depth	4	1	3	A
X ray	5	3	0	A
😈	2	4	2	B

Lightricks

A/B Test Results

Project completion rate – VL iOS - AI Recolor V2 experiment - Features project completion rate

What % of projects/users that used each feature were exported with the added feature

segmentation

01 All Users

[redacted]

Lightricks

Can we do better?

Current drawbacks

- Dependency on GPT-3.5-Turbo, a **closed model** with usage **costs**
- Larger LMs like GPT-3.5-Turbo have **high latency**
- Lack of integration of user preferences

Our proposed solution

A Distillation framework – fine-tune a (**smaller**) **student LLM** with guidance from a (**larger**) **teacher LLM** and **users behavioral signals**

Our proposed solution advantages

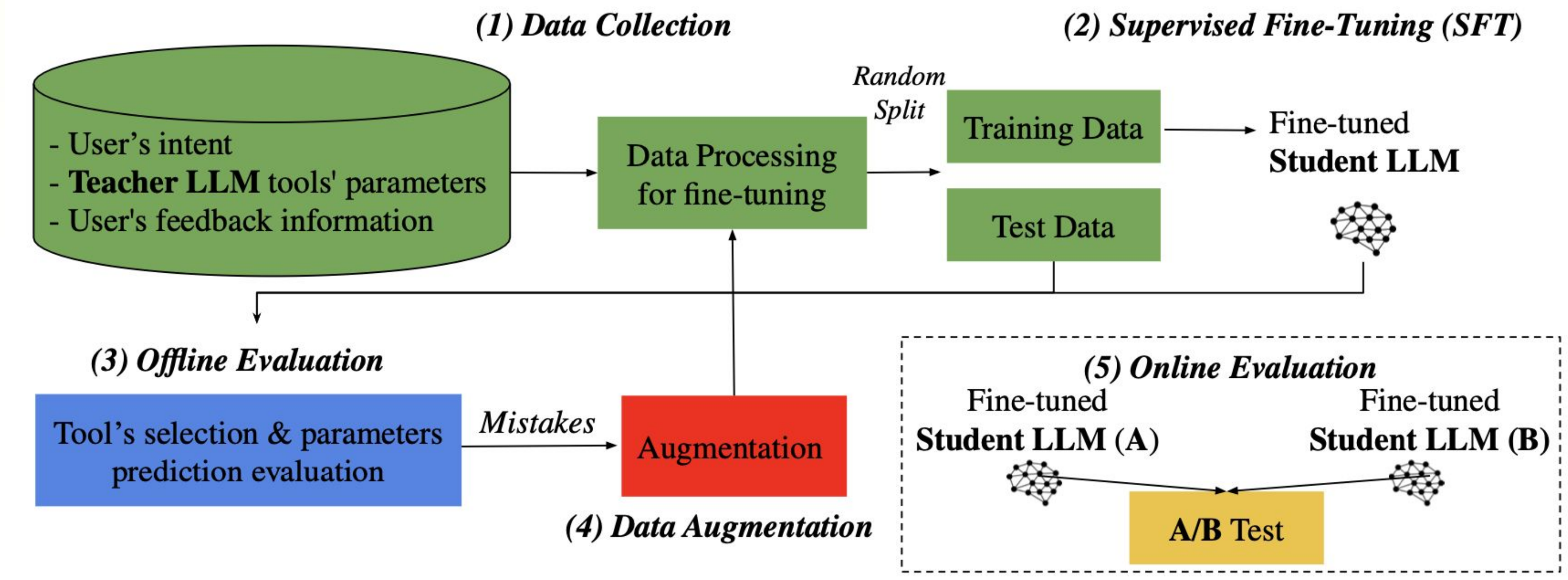
- Open-source models are **free**
- Smaller LMs have a **better latency**
- Fine-tuning on high-quality data to better **align our user preferences**

Lightricks

Our distillation framework approach

Lightricks

Our distillation framework approach


(1) Data Collection
- User's intent
- Teacher LLM tools' parameters
- User's feedback information
Data Processing for fine-tuning
Random Split
Training Data
Test Data
(2) Supervised Fine-Tuning (SFT)
Fine-tuned Student LLM
(3) Offline Evaluation
Tool's selection & parameters prediction evaluation
Mistakes
Augmentation
(4) Data Augmentation
(5) Online Evaluation
Fine-tuned Student LLM (A)
Fine-tuned Student LLM (B)
A/B Test

Lightricks

1) Data Collection

Gathering Teacher LLM Outputs.

- **Teacher LLM: GPT-3.5-Turbo** (serving users for **four months** – data collection period).
- **A data row includes:**
  - The **user's intent** with the requested vibe (e.g., "x-ray").
  - The **output of the teacher LLM** to this intent, including the tools to use and their parameters.
  - Whether the **user exports** the result per tool (highly satisfied users export results).

- **Data Filtering:**
  - Samples with zero exports.
  - Our teacher LLM can generate different outputs per intent (across different calls); We take as ground truth the result that **maximizes the export rate**.

- **Prompts: one-shot** example for user intent, with **rational (CoT)** and **output parameters** per tool.

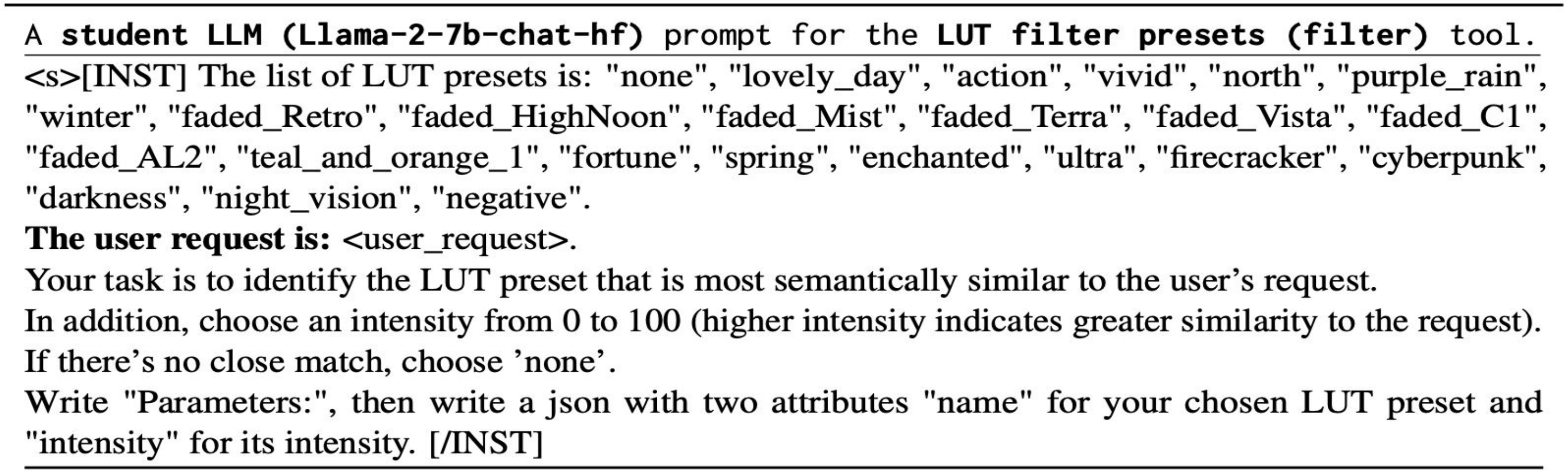
- In total, we collected **9,252 unique user intents**, each with corresponding teacher outputs for the 3 tonal adjustment tools, resulting in **27,756 rows**.

Lightricks

1) Data Collection

Data Processing for Fine-Tuning.

- We used the collected data to fine-tune a student LLM, using three **more concise** prompts.
- We decided not to request rational from the student, as we **prioritize low latency**.
- The student LLM is **trained on all three tools** (similar to multi-task instruction)



A **student LLM (Llama-2-7b-chat-hf)** prompt for the **LUT filter presets (filter)** tool.

<s>[INST] The list of LUT presets is: "none", "lovely_day", "action", "vivid", "north", "purple_rain", "winter", "faded_Retro", "faded_HighNoon", "faded_Mist", "faded_Terra", "faded_Vista", "faded_C1", "faded_AL2", "teal_and_orange_1", "fortune", "spring", "enchanted", "ultra", "firecracker", "cyberpunk", "darkness", "night_vision", "negative".
The user request is: <user_request>.
Your task is to identify the LUT preset that is most semantically similar to the user's request.
In addition, choose an intensity from 0 to 100 (higher intensity indicates greater similarity to the request).
If there's no close match, choose 'none'.
Write "Parameters:", then write a json with two attributes "name" for your chosen LUT preset and "intensity" for its intensity. [/INST]

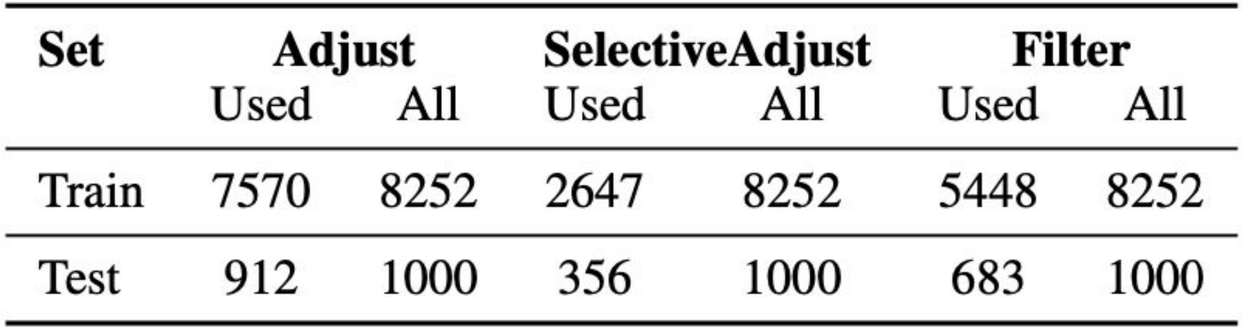
Lightricks

1) Data Collection

Data Splitting.

We randomly split the data for fine-tuning.

- **Test set: 1K unique user intents**, each with a teacher LLM output for each tool (**3K samples**).
- **Training set:** the remaining data (**8,252 rows**).
- Each row includes a user intent and three tool outputs.



Set	**Adjust**		**SelectiveAdjust**		**Filter**	
	Used	All	Used	All	Used	All
Train	7570	8252	2647	8252	5448	8252
Test	912	1000	356	1000	683	1000

Lightricks

2) Supervised Fine-tuning (SFT)

Filtered dataset.

$$D = \{(x, y)\}$$

Student LLMs.

- **Auto-regressive model (decoder only)**: Llama-2-7b-chat-hf (7B)
  - Maximize the log probability of the target word given prior words and models parameters.

Meta

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log P(y_t \mid y_1, y_2, \ldots, y_{t-1}; \theta)$$

- **Sequence-to-Sequence model (encoder decoder)**: FlanT5-base (250M)
  - Maximize the log probability of the target word given previous target words and the input sequence, using the models parameters.

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log P(y_t \mid y_1, y_2, \ldots, y_{t-1}, \mathbf{x}; \theta)$$

Lightricks

3) Offline Evaluation Metrics

- **Tool-selection:** the model's ability to decide correctly whether to use a tool.
  We measure ***precision*** and ***recall***, and report tool-selection score as the ***F1-score***.

- **Quality:** the model's ability to use a tool correctly.
  - For the **filter tool**: the ***accuracy*** on the filter name.
  - For the **adjust** and **selective adjust** tools: the ***mean cosine similarity*** across samples between predicted and ground-truth parameter values.

- **Final score:** the ***harmonic mean*** between ***tool-selection score*** and ***quality score***, emphasizing high performance in both.

- **Overall score:** the average of the final scores of all tools.

Lightricks

3) Offline Evaluation – Reality check

- We analyze the actual generated images/videos by applying the tools' predicted parameters in our app.

- In this study, we analyze a random sample, with three human annotators per sample (RQ1).

- Ideas for automatic evaluation of the generated images/videos:
  - **Aesthetic Score (Schuhmann et al., 2022, CVPR 2022):** predicts people's ratings of images.

  - **PickScore (Kirstain et al., 2023, NeurIPS 2023):** evaluates relevancy based on a preference model trained on text-to-image prompts and user preferences.

  - **VQA Score (Zhiqiu Lin, ECCV 2024):** Evaluates the alignment between images and text prompts using visual question answering (VQA) models.

  - **Proprietary models (e.g., GPT-4V):**

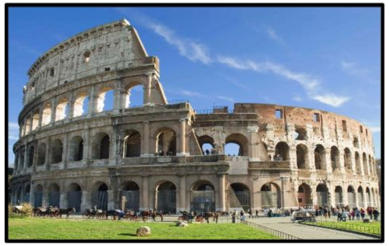
`GPT-4V output given the “Morocco” intent, and a source image with two edited images.`

Prompt:
“You are given three images: (A), (B), and (C). Describe the transformations of (B) and (C) from (A). Then choose which image (B) or (C) better represents the <user’s request>”

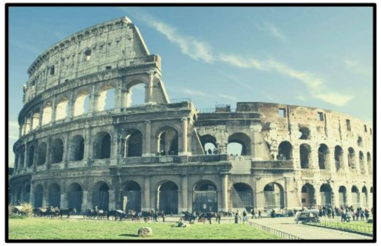
Output:
Image (A) is the original photo showing the Colosseum in Rome with a clear blue sky and natural daylight colors. **Image (B)** has undergone a transformation that includes a decrease in contrast, a slight increase in brightness, and a color shift towards a cooler, teal or cyan tint, particularly noticeable in the sky and the shadows. This gives the image a more muted and vintage look. **Image (C)** has been transformed with an increase in contrast and saturation, giving the colors a warmer, golden hue. This is especially evident in the sky, which has taken on a warm, orange tone, and the overall image has a more vibrant and sunny appearance.

The Morocco filter look is typically characterized by warm tones, with an emphasis on oranges, reds, and earthy colors that reflect the landscapes and architecture of Morocco.
Answer: (C) better represents the Morocco filter look due to its warm and vibrant color palette.

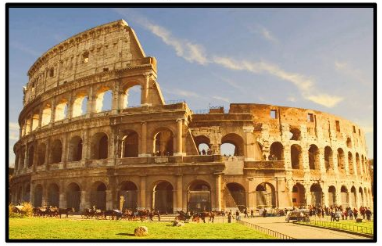
(A)



(B)



(C)



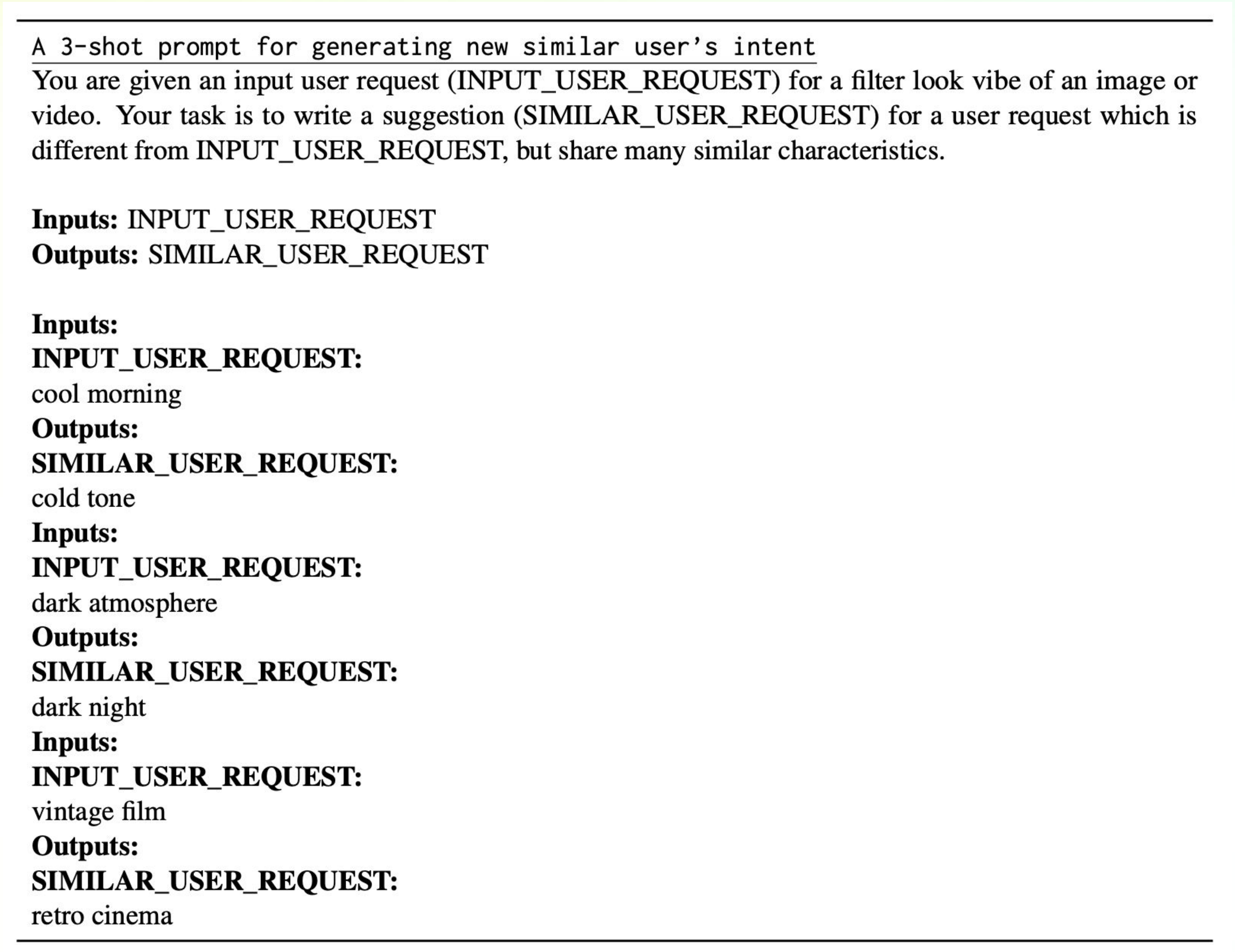
Lightricks

4) Data Augmentation

- We iteratively run the offline evaluation on the training set.

- **(1) Identifying where the student LLM predictions differ from the teacher's (a mistake)**
  - For the **filter tool**, a mistake occurs when the predicted filter name is incorrect.
  - For the **adjust and selective adjust**, a mistake occurs when a sample's cosine similarity is lower than the tool's mean cosine similarity without data augmentation.

- **(2) Using another LLM to generate similar input user intents where the student LLM made mistakes (e.g., "cool tone" from "cool morning")**
  - These new intents, along with the teacher LLM original answers, are added to the **training set**.
  - We augmented an intent whenever a mistake was identified by at least one tool.



A 3-shot prompt for generating new similar user’s intent

You are given an input user request (INPUT_USER_REQUEST) for a filter look vibe of an image or video. Your task is to write a suggestion (SIMILAR_USER_REQUEST) for a user request which is different from INPUT_USER_REQUEST, but share many similar characteristics.

Inputs: INPUT_USER_REQUEST
Outputs: SIMILAR_USER_REQUEST

Inputs:
INPUT_USER_REQUEST:
cool morning
Outputs:
SIMILAR_USER_REQUEST:
cold tone
Inputs:
INPUT_USER_REQUEST:
dark atmosphere
Outputs:
SIMILAR_USER_REQUEST:
dark night
Inputs:
INPUT_USER_REQUEST:
vintage film
Outputs:
SIMILAR_USER_REQUEST:
retro cinema

Lightricks

5) Online Evaluation

- When our offline evaluation shows it is worthwhile to consider a new student LLM, we confirm it in an online A/B test experiment.
- **Metric of interest**: *project_completion_rate = #projects_exported / #projects_started*.
- This metric indicates the total user satisfaction with the results and the overall experience.

Lightricks

Experiments

Lightricks

Experiments

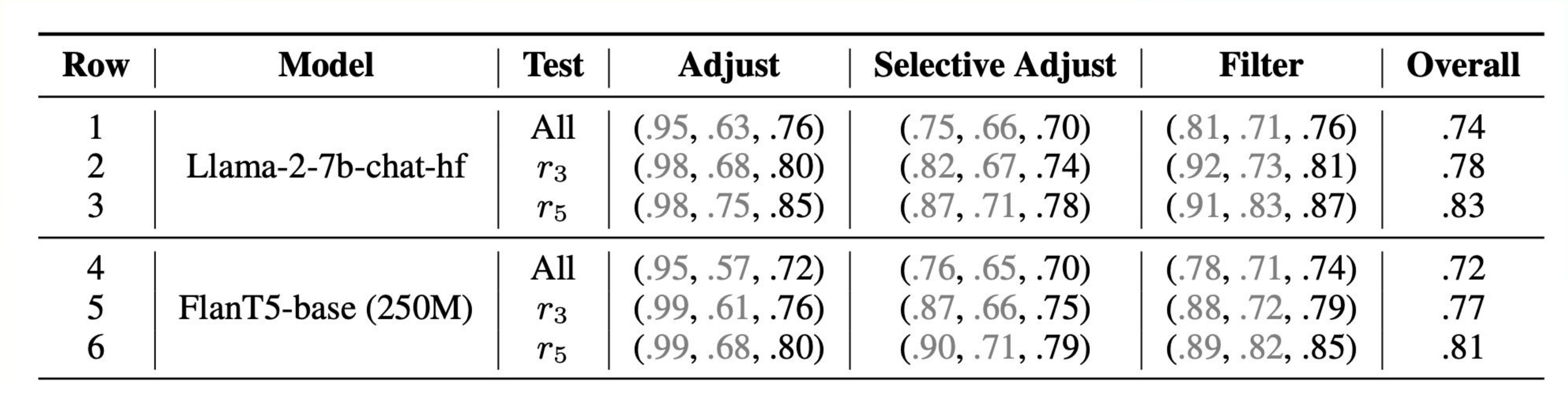
Research Questions.

- **RQ1:** How well do student LLMs perform, and do they effectively mimic the teacher LLM?
- **RQ2:** Is augmentation effective in low-data regimes?

Models.

- **Teacher LLM**: **GPT-3.5-Turbo**
- **Student LLMs**:
  - **Llama-2-7b-chat-hf** with Low Rank Adaptations (LoRA) + Quantization, A100 GPU.
  - **FlanT5-base (250M)** (faster), L4 GPU (5 times cheaper).
- Fine-tune for 10 epochs, selecting the best checkpoint from last 3 epochs, based on the highest final average tool score.

RQ1: Student LLMs Performance – Offline Evaluation



Row	Model	Test	Adjust	Selective Adjust	Filter	Overall
1	Llama-2-7b-chat-hf	All	(.95, .63, .76)	(.75, .66, .70)	(.81, .71, .76)	.74
2		r_3	(.98, .68, .80)	(.82, .67, .74)	(.92, .73, .81)	.78
3		r_5	(.98, .75, .85)	(.87, .71, .78)	(.91, .83, .87)	.83
4	FlanT5-base (250M)	All	(.95, .57, .72)	(.76, .65, .70)	(.78, .71, .74)	.72
5		r_3	(.99, .61, .76)	(.87, .66, .75)	(.88, .72, .79)	.77
6		r_5	(.99, .68, .80)	(.90, .71, .79)	(.89, .82, .85)	.81

- **Metrics**: (tool-selection score, quality score, final score). **Overall**: avg. of final scores across the tools.
- **FlanT5-base performs very similarly to Llama-2-7b-chat-hf (rows 1, 4**).

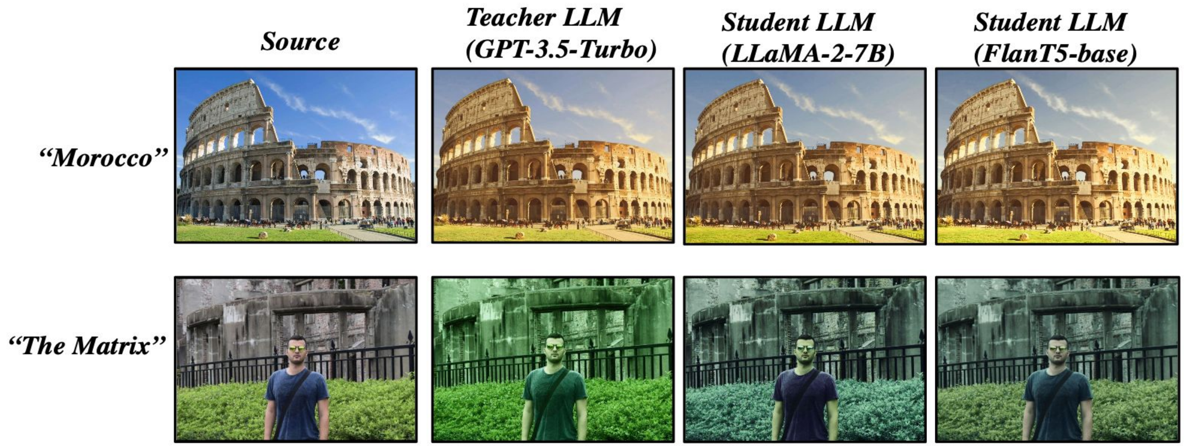
- (r5 > r3 > All), where r_i denotes user intents with at least i calls.
  Interestingly, both models **perform better on a test subset with more popular user intents**

- **The average latency for all tools**:
  GPT 4: 9-10s, **GPT 3.5:** 3-4s.
  LLaMA2-7b-chat: 1.63s (A100 GPU), **FlanT5-base:** 1.38s (L4 GPU).

Lightricks

RQ1: Student LLMs Performance – Offline Evaluation

- Three calibrated team annotators reviewed each sample according to two criteria:
  - Is the image relevant to the intent?
  - Does the student model correctly mimic the teacher?



- **Relevancy**: **87%-93%** for all models.
- **Student LLM correctly mimic the teacher**: **73%** for both (not the same).

Lightricks

RQ1: Student LLMs Performance – Online Evaluation (A/B Test)

Metric. project completion rate (as an indicator for user satisfaction)

Experiment 1.

- **Teacher LLM:** GPT-3.5-Turbo (94,317 projects) **vs. Student LLM:** Llama-2-7b-chat (93,495 projects)
- **Results:** the completion rate for the teacher was **96.1%** of that of Llama-2-7b-chat
- **Conclusion:** Comparable performance, we chose Llama-2-7b-chat for its lower latency and cost

Experiment 2.

- **Student LLM:** FlanT5-base (20,294 projects) **vs. Student LLM:** Llama-2-7b-chat (20,282 projects)
- **Results:** the completion rate of FlanT5-base was **99%** of that of Llama-2-7b-chat
- **Conclusion:** Comparable performance, we chose FlanT5-base for its lower latency and cost

- **Our offline metrics align with the results of the online A/B tests.**

RQ2: Augmentation in low-data regimes

- We evaluated the augmentation on different sizes of our training set, using random sampling.
- Mistakes were evaluated on a random sample of 1,000 instances.
- We augmented an intent whenever a mistake was identified by at least one tool.
- Using GPT-4, we generated similar user intents.

Train %	**Augmentations**	**Train Size**	**Overall Score**
100	0	8,252	0.72
12.5%	0	1,031	0.52
12.5%	**806 (43.8%)**	**1,837**	**0.65**

- **Our results show a 25% performance improvement (+0.13), in low data regimes (1/8 of the training) with just one iteration!**

Lightricks

Conclusions

Lightricks

Which LLM to Use?

GPT 4

- Zero Shot
- Long complex prompt and response
- Expensive
- High latency

GPT 3.5

- One Shot
- Shorter prompt and shorter response
- 50% latency
- 20x cost reduction

Custom LLM (Fine-tuned LLaMA 2 / FlanT5)

- Additional latency reduction
- Cost reduction
- Learning from users’ exports data

Lightricks

Key takeaways

★ Combine LLMs and classic algorithms / tools for best of both worlds.

★ Always start with prompt engineering.

★ Fine-tuned small language models can deliver more efficiently.

★ Leverage your users data.

★ Offline evaluation metrics let you iterate quickly and cheaply.

Lightricks

Future Work

- To test potential fine-tuning improvements by **adding rational as an additional label** for supplementary supervision in a **multi-task framework** (Hsieh et al., 2023).
- To quantify the **benefits of integrating user signals**, and to explore **other methods for combining user feedback** (e.g, **personalization**).
- To extend our **one-hop responses** to **conversational agents / dialogue systems**.

Lightricks

Questions?

Thank You!