

# LLMs Meet Image and Video Generation

---

Yingqing He

<https://github.com/YingqingHe>

# Yingqing He

- Final-year PhD student at HKUST
- Supervisor: Prof. Qifeng Chen
- Research experiences:
  - Text-to-Video Generation
    - LVDM, VideoCrafter, ScaleCrafter, Make-A-Cheap-Scaling
  - Controllable Video Generation
    - Animate-A-Story, Follow-Your-Pose, Follow-Your-Click, Follow-Your-Emoji.
  - Multimodal Generation
    - Seeing-and-Hearing

2022.11

LVDM

Video generation of  
1000+ frames



2023.04

VideoCrafter 0.9

Open-sourcing Text-to-Video Generation toolbox



2023.11

VideoCrafter 1

ECCV2024  
AI4VA

Animate-A-Story



ICLR 2024  
(Spotlight)

ScaleCrafter



CVPR 2024

Seeing and Hearing

ECCV 2024

Make-A-Cheap-Scaling



# LLMs Meet Image and Video Generation

## Background:

- The achievements of LLMs (GPT) , image and video generation (Stable Diffusion, Flux, Sora, ...) can be attributed to **the large-scale generative pretraining.**

# Technology Wave of Large-scale Generative Pretraining

## NLP: Transformer

---

### Attention Is All You Need

---

**Ashish Vaswani\***  
Google Brain  
avaswani@google.com

**Noam Shazeer\***  
Google Brain  
noam@google.com

**Niki Parmar\***  
Google Research  
nikip@google.com

**Jakob Uszkoreit\***  
Google Research  
usz@google.com

**Llion Jones\***  
Google Research  
llion@google.com

**Aidan N. Gomez\* †**  
University of Toronto  
aidan@cs.toronto.edu

**Lukasz Kaiser\***  
Google Brain  
lukasz.kaiser@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com

Ashish Vaswani et al. NIPS 2017

### Language Tasks

- ✓ Machine translation
- ✓ Document generation

### Scalability

- 100M GPT1
- 300M Bert
- 1.5B GPT2
- 8B Megatron-Lm
- 11B T5
- 175B GPT3
- 1.8 trillion ChatGPT
- ...

# Technology Wave of Large-scale Generative Pretraining

## Image Generation: From GANs to Diffusion Models



StyleGAN 2019



DDPM 2020

(Denoising Diffusion Probabilistic Models )



Guided Diffusion 2021

(Diffusion Beat GANs)

# Technology Wave of Large-scale Generative Pretraining

Image Generation: From single domain -> open domain

## Single domain generation

A single generative model possesses the capability to generate images within a singular domain only.

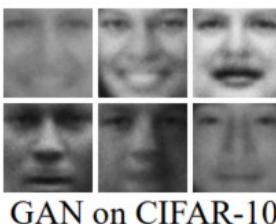
*VAE (Dec 2013), GAN (Jun 2014), Normalizing flow (May 2015), DDPM (Dec 2020) ...*

## Open domain generation

Generative models exhibit the capability to generate images across arbitrary domains guided by textual descriptions.

*DALL·E (Feb 2021), LDM (Dec 2021), DALL·E 2 (Apr 2022), Imagen (May 2022) ...*

2013 - 2020



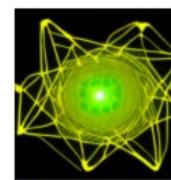
2021 - 2022



A painting of a squirrel eating a burger (LDM)



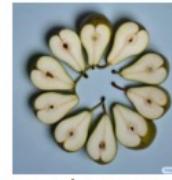
An animal half mouse half octopus (LDM)



A slightly conscious neural network (LDM)



A wine glass on top of a dog (Imagen)



A pear cut into seven pieces arranged in a ring (Imagen)



A wine glass on top of a dog (Imagen)

# Technology Wave of Large-scale Generative Pretraining

## Image Generation:

From single domain -> open domain -> interactive / free-form generation leveraging LLMs

### Single domain generation

A single generative model possesses the capability to generate images within a singular domain only.

*VAE (Dec 2013), GAN (Jun 2014), Normalizing flow (May 2015), DDPM (Dec 2020) ...*

### Open domain generation

Generative models exhibit the capability to generate images across arbitrary domains guided by textual descriptions.

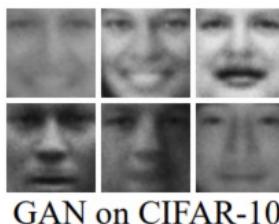
*DALL-E (Feb 2021), LDM (Dec 2021), DALL-E 2 (Apr 2022), Imagen (May 2022) ...*

### Interactive / Interleaved generation

Generative models create images through user interaction powered by  **Large Language Models**.

*DALL-E 3 (Sept 2023), DreamLLM (Sept 2023), mini DALL-E 3 (Oct 2023), CoDi-2 (Nov 2023) ...*

2013 - 2020



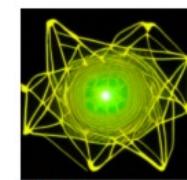
2021 - 2022



A painting of a squirrel eating a burger (LDM)



An animal half mouse half octopus (LDM)



A slightly conscious neural network (LDM)



A wine glass on top of a dog (Imagen)



A pear cut into seven pieces arranged in a ring (Imagen)

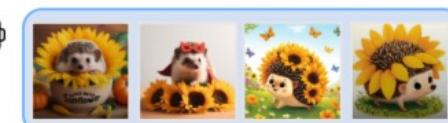


A wine glass on top of a dog (Imagen)

2023 -



My 5 year-old keeps talking about a "super-dupersunflower hedgehog" -- what does it look like?



Could you design some stickers for it?



Sure, here you are.



Interactive and interleaved generation of DALL-E 3

# Technology Wave of Large-scale Generative Pretraining

## Image Generation:

From single domain -> open domain -> interactive / free-form generation leveraging LLMs



Can you generate an image of Lucky sitting in front of the Golden Gate Bridge?



### Single domain generation

A single generative model possesses the capability to generate images within a singular domain only.

*VAE (Dec 2013), GAN (Jun 2014), Normalizing flow (May 2015), DDPM (Dec 2020) ...*

### Open domain generation

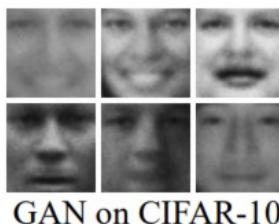
Generative models exhibit the capability to generate images across arbitrary domains guided by textual descriptions.

*DALL-E (Feb 2021), LDM (Dec 2021), DALL-E 2 (Apr 2022), Imagen (May 2022) ...*

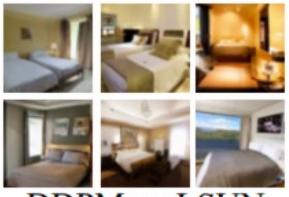
2013 - 2020



GAN on MNIST



GAN on CIFAR-10



DDPM on LSUN



DDPM on CelebA-HQ

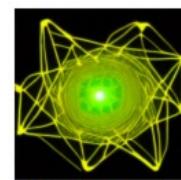
2021 - 2022



A painting of a squirrel eating a burger (LDM)



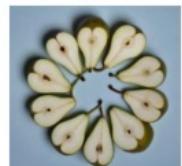
An animal half mouse half octopus (LDM)



A slightly conscious neural network (LDM)



A wine glass on top of a dog (Imagen)



A pear cut into seven pieces arranged in a ring (Imagen)



A wine glass on top of a dog (Imagen)



It looks great! Can you make Lucky wear sunglasses?



# Technology Wave of Large-scale Generative Pretraining

Increasing model parameters of text-to-image models

Stable Diffusion

~1B parameters

U-Net with transformer  
modules



Stable Diffusion 3

8B parameters

transformers



Flux

12B parameters

transformers



# Technology Wave of Large-scale Generative Pretraining

Video Generation: Video GANs -> Video Diffusion Models

StyleGAN-V

CVPR 2022

Snap



VDM

NIPS 2022

Google

Construction Site Activity



Clouds moving



sunset at sea



"A teddy bear washing dishes"



Imagen Video  
Arxiv 2022  
Google

"A panda bear driving a car"



Sora  
2024/02  
OpenAI



"A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about."

# Technology Wave of Large-scale Generative Pretraining

**Increasing model parameters of text-to-video models**

2024-01	2024-03	2024-09
VideoCrafter2	Open-Sora	CogVideo
1.8B parameters	5.8B parameters	5B parameters
3D U-Net with transformer modules	transformers	transformers

Large-scale pretraining demonstrates significant performance in language/image/video generation

## LLMs/MLLMs

GPT4

LLaMA

LLaVA



## Text-to-image

Stable Diffusion  
Flux

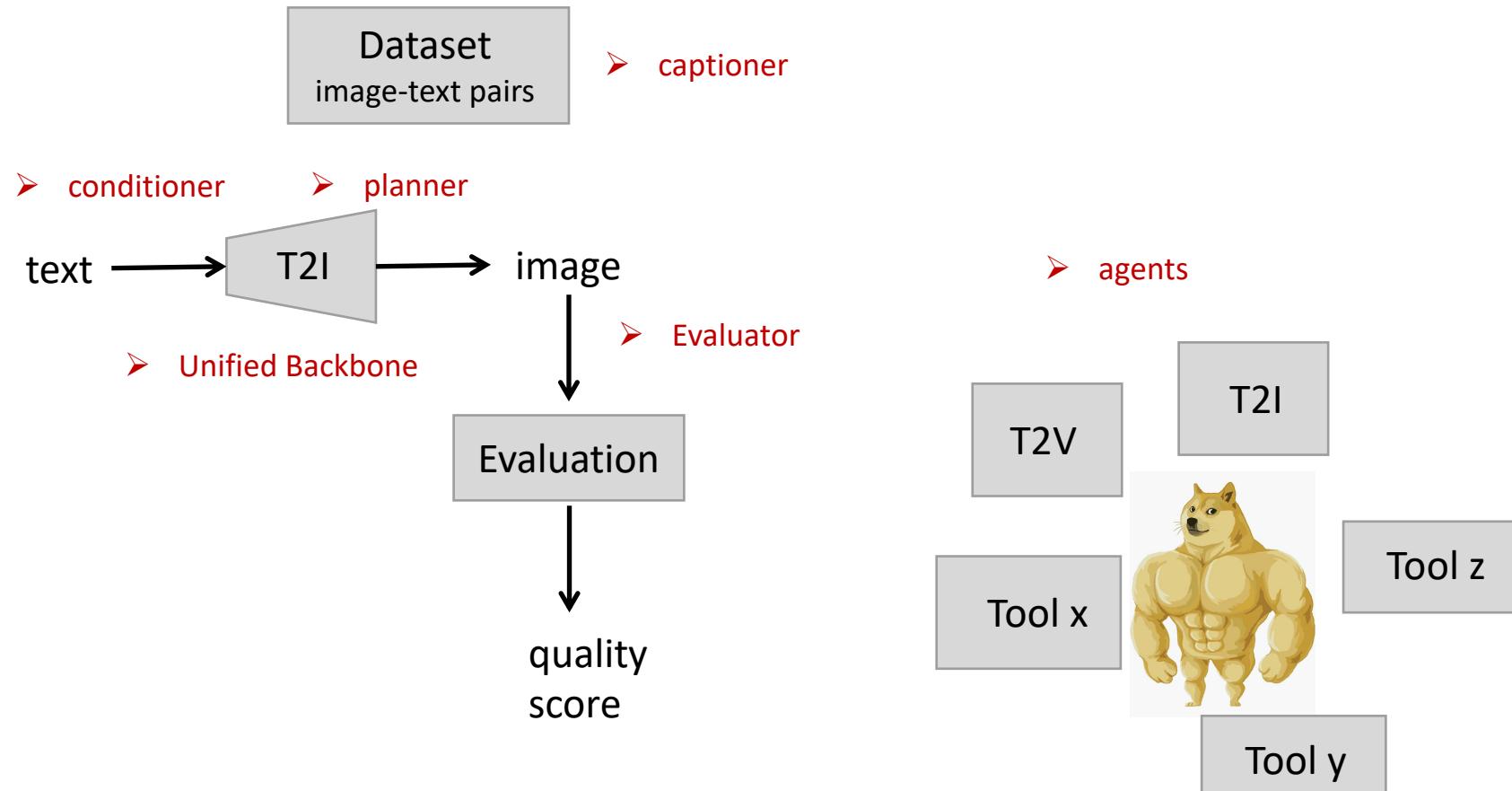
## Text-to-video

VideoCrafter  
ModelScope  
AnimateDiff  
Open-Sora  
Cogvideo

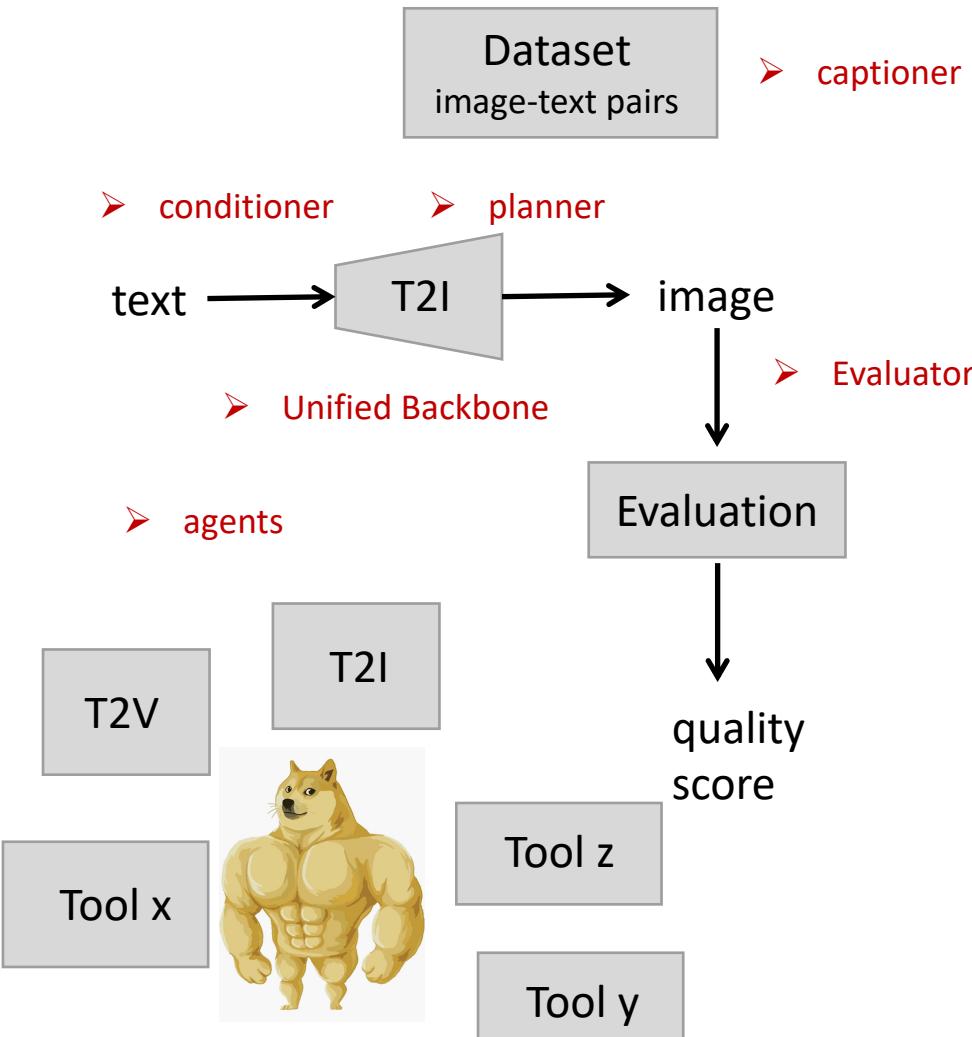
Can LLMs help the generation of images and videos?

**LLMs help the generation of image/video  
in various aspects**

# LLMs help the generation of image/video in various aspects



# LLMs help the generation of image/video in various aspects



1. LLMs as a Unified Backbone
2. LLMs as Planners
3. LLMs as Captioners
4. LLMs as Conditioners
5. LLMs as Evaluators
6. LLMs as Multimodal Agents

# LLM as a Unified Backbone

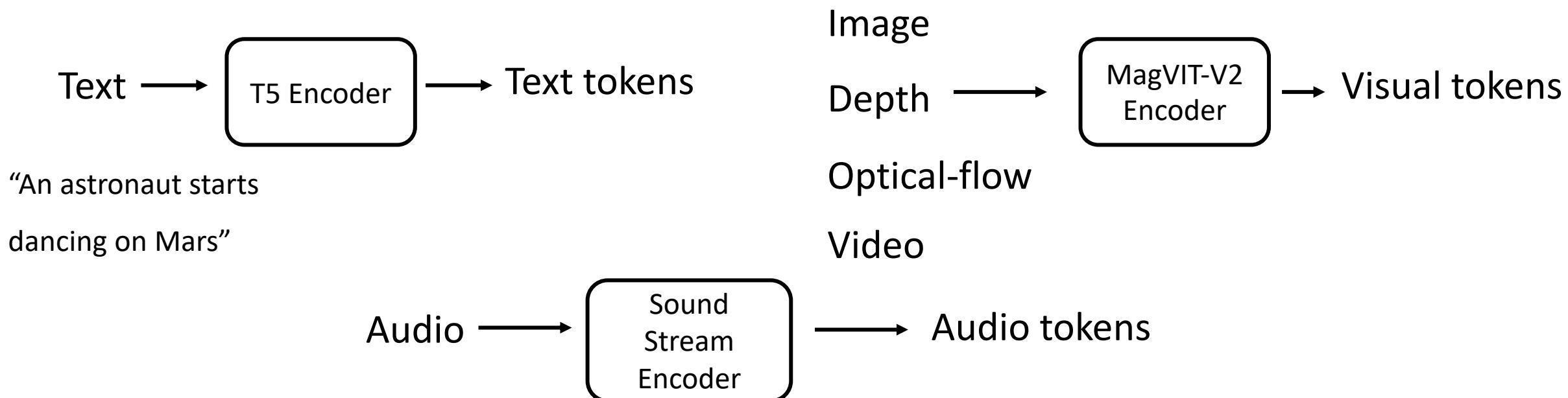
**All in discrete token space:**

Unifying data of different modalities into discrete tokens,  
and leverage the LLM architecture to perform the next-token prediction task

# LLM as a Unified Backbone

**All in discrete token space:**

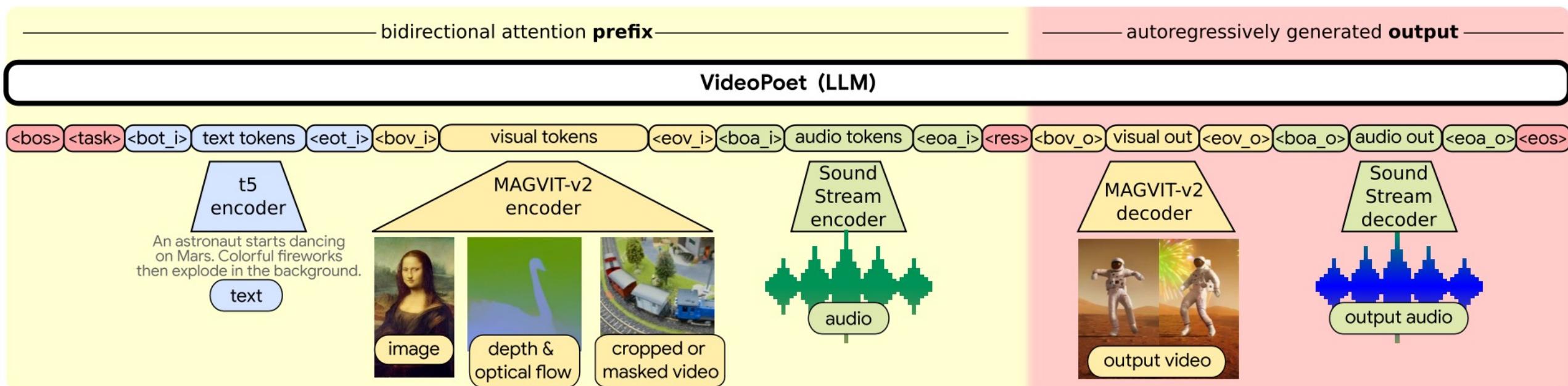
Unifying data of different modalities into discrete tokens,  
and leverage the LLM architecture to perform the next-token predict task



# LLM as a Unified Backbone

Autoregressively generate multimodal tokens:

And then decode using modality-specific decoders



Text-to-video generation, Video generation with audio, video inpainting.

# LLM as a Unified Backbone

**MLLM:** Make LLMs have the ability to understand and generate information of other modalities!

- LLM + Image captioning; LLM + Image generation

🤖: Generate a figure of a yellow dog sitting on the grass and a cute cat running in the garden

**ControlLLM**

🤖: I have generated an image for you. The image depicts a yellow dog sitting on the grass and a cute cat running in the garden. The image file name is 1dd44a\_image.png.

Here is a summary of the workflow:

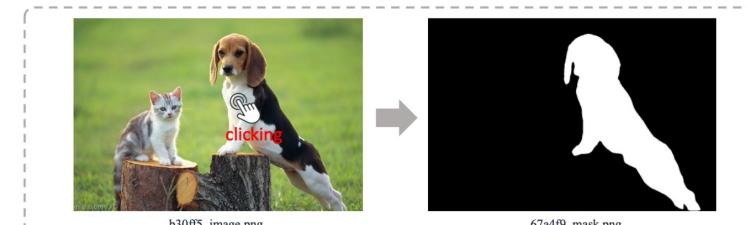
- 1.The text-to-image tool was used to generate the requested image.
- 2.The tool took the text description "The new image depicts a yellow dog sitting on the grass and a cute cat running in the garden."
- 3.The tool outputted the image file named 1dd44a\_image.png.

Please let me know if there is anything else I can help you with.



1dd44a\_image.png

- Interactive image editing following human instructions



b30ff5\_image.png → 67a4f9\_mask.png

🤖: remove the masked object.

**ControlLLM**

🤖: I have removed the masked object from the image. The result can be found in the file 34b15a\_image.png.

To remove the masked object, the workflow consists of using the image\_inpainting tool. The input image is b30ff5\_image.png and the mask is 67a4f9\_mask.png. The output image after removing the masked object is 34b15a\_image.png.



34b15a\_image.png

- Generate & edit  
Image/video/3D/audio

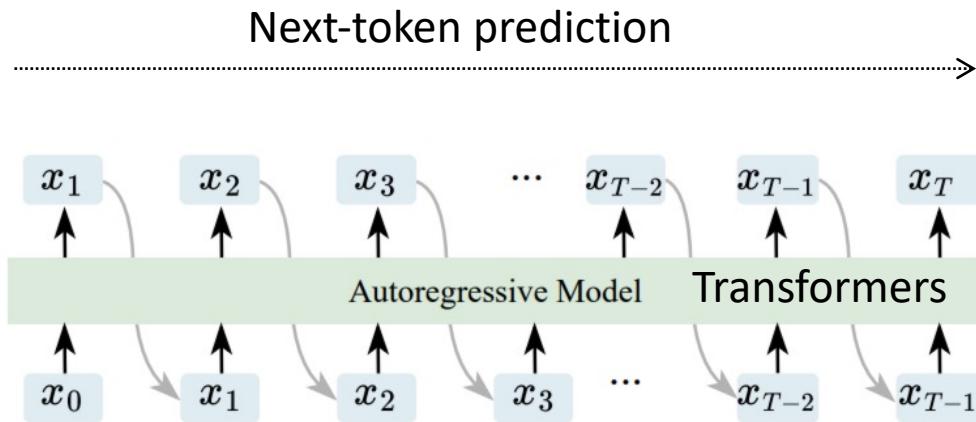


a784c4\_image.png → 4115c8\_video.mp4

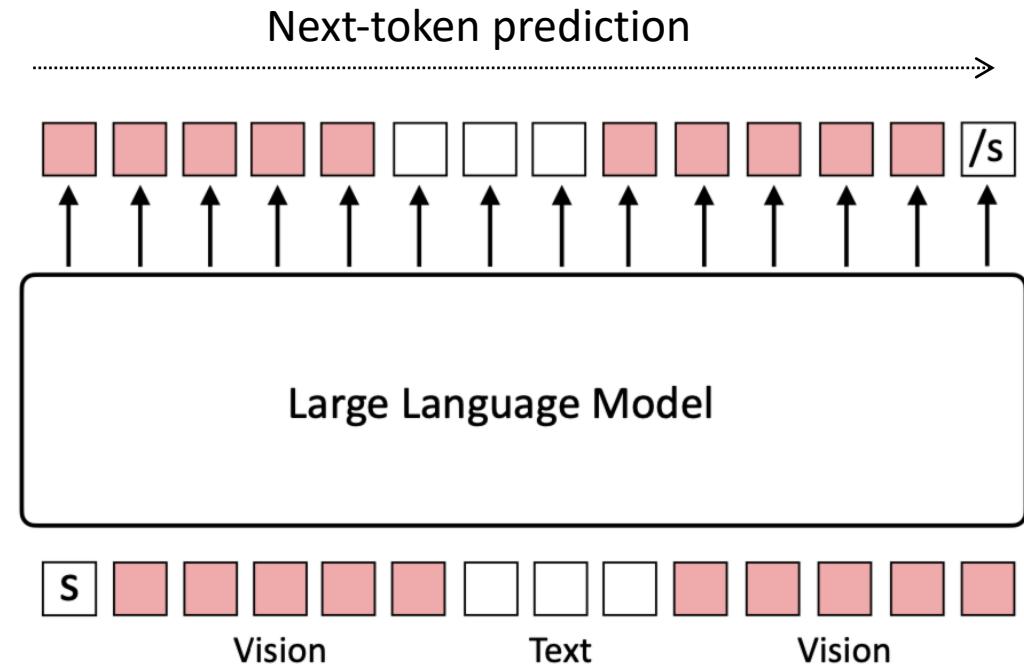
# LLM as a Unified Backbone

---

LLM: Next-token prediction

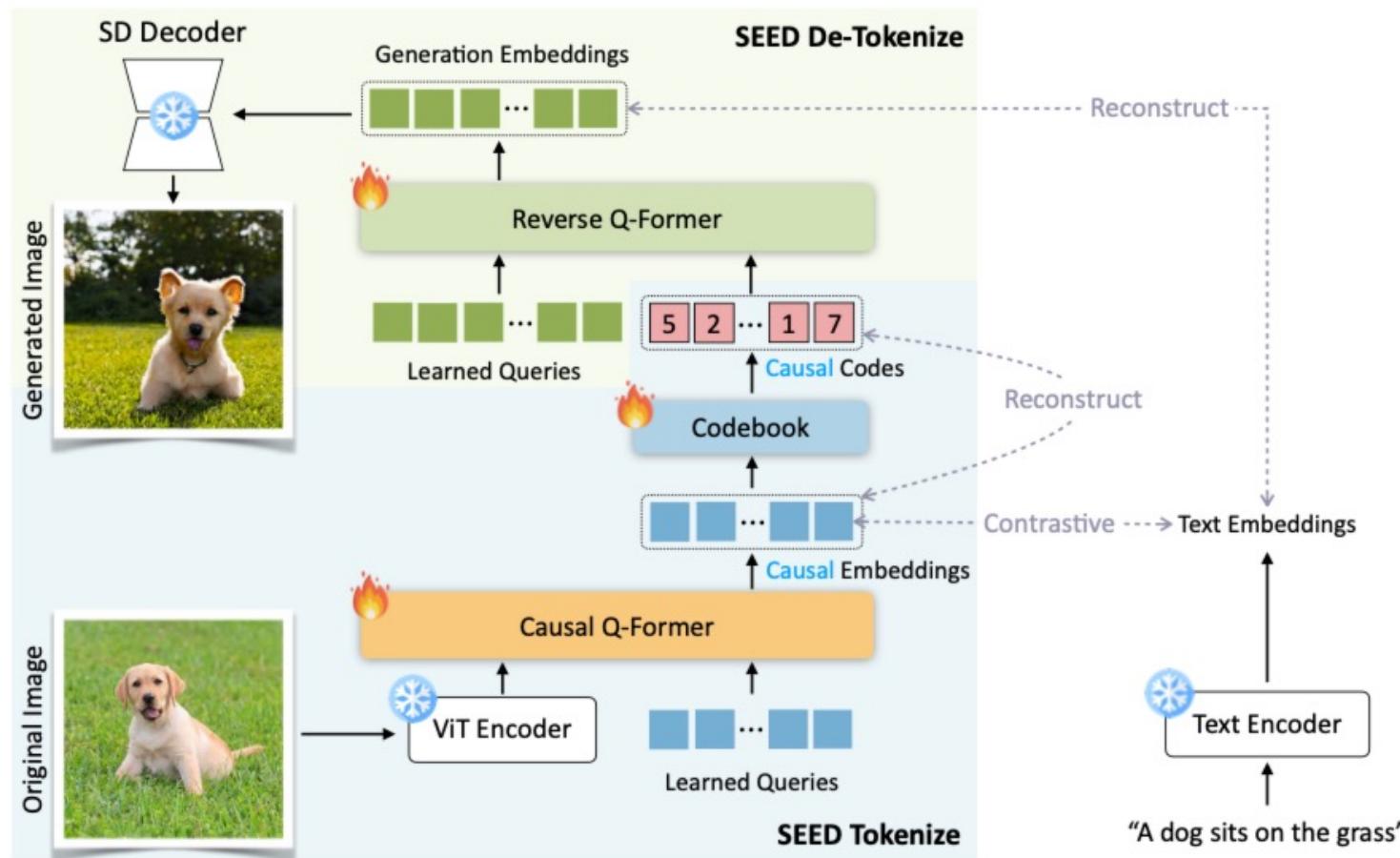


Multimodal autoregression with Visual tokenizers



# Visual Tokenizers & De-tokenizers

Goal: semantically consistent after tokenization and detokenization

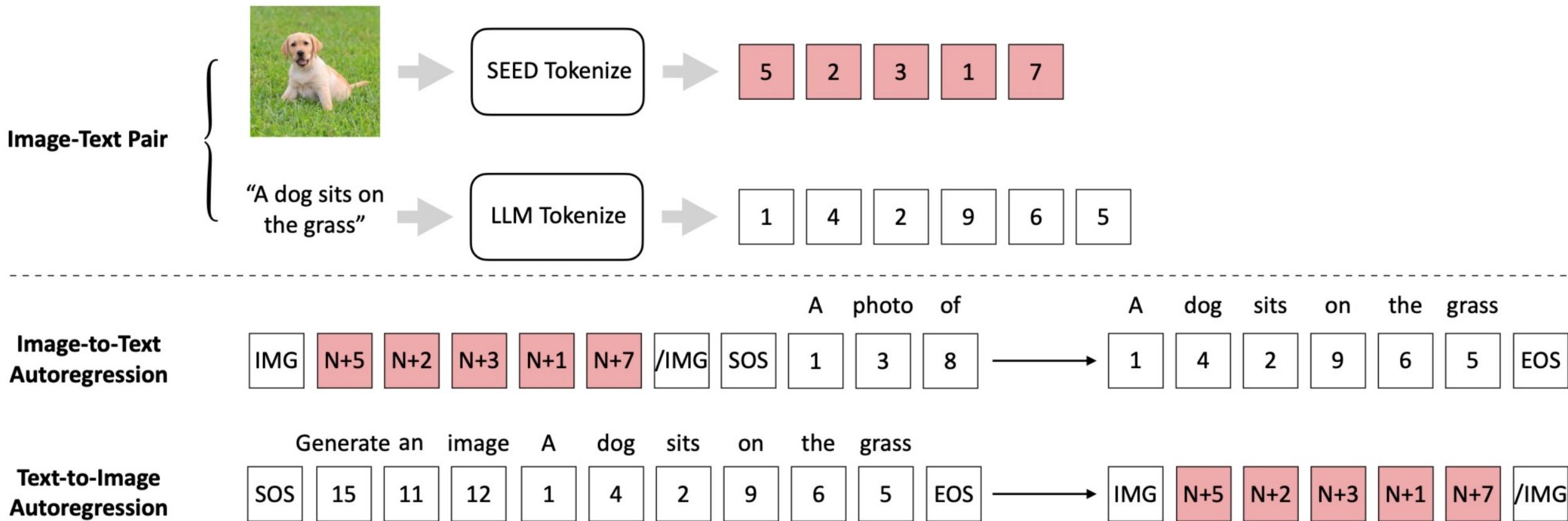


Training stages

1. Causal Q-Former, convert visual features to causal embeddings, contrastive loss
2. Train a visual codebook to discretize the causal embeddings to discrete codes, maximize causal embedding cosine similarity

# Multimodal Autoregression

---





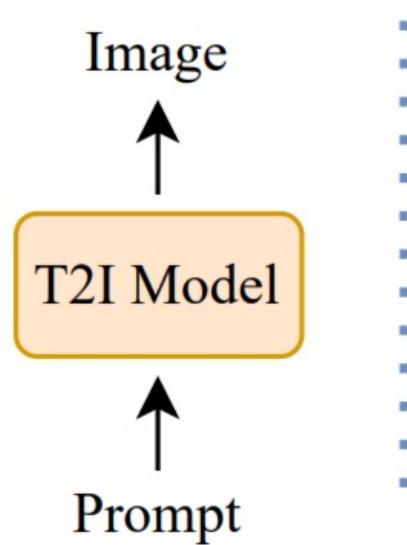
# LLM as Planner: Always instruct the visual generation model in detail

---

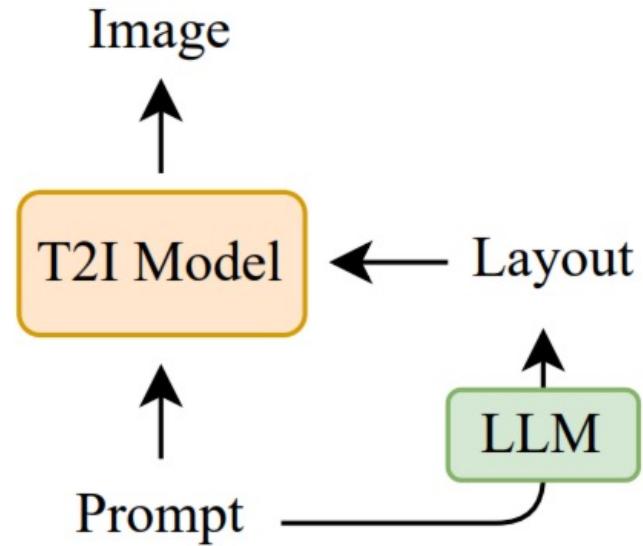
- Layouts/Bounding boxes for spatial location planning
- Semantic descriptions
  - Motion descriptions for temporal planning
  - Patch descriptions for Image scale planning
  - Entity descriptions for character/object design across different video clips

# LLM as Planner

---



(a) General T2I  
models



(b) Layouts as intermediate  
results for image generation

# LLM as Planner: Spatial Layout Planning

---

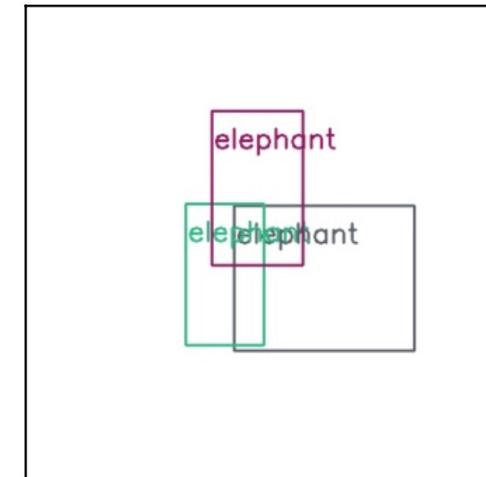
## Quantity Correctness

“There are **three elephants** standing beside a pool of water.”

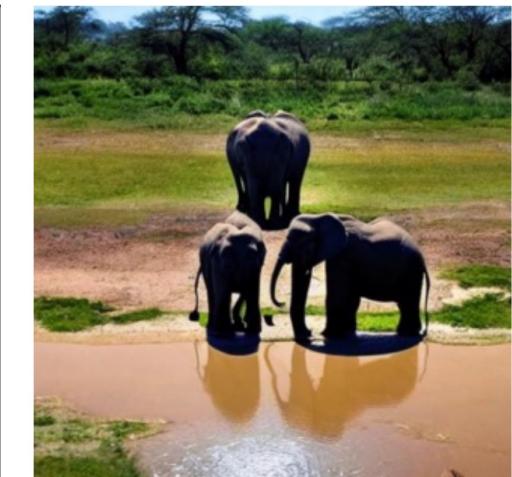
Text-to-Image



Text-to-Layout



Layout-to-Image



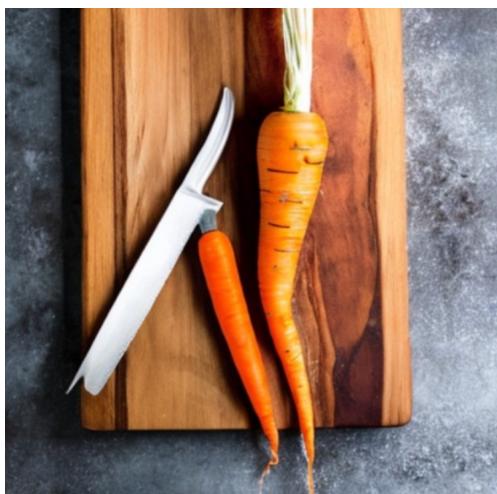
# LLM as Planner: Spatial Layout Planning

---

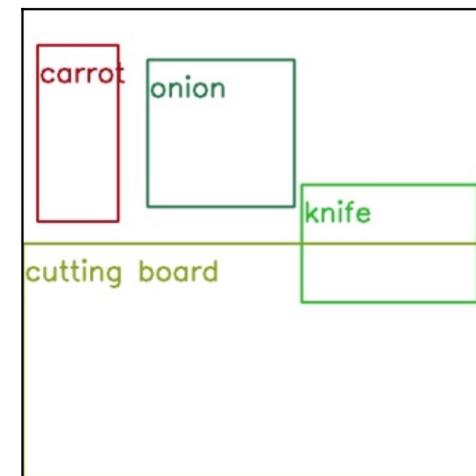
## Spatial Position Relation

“A **carrot** and some **onion** next to a **knife** on a **cutting board**.”

Text-to-Image



Text-to-Layout



Layout-to-Image



# LLM as Planner: Spatial Layout Planning

## Task instruction

Instruction:

Given a sentence prompt that will be used to generate an image, plan the layout of the image. The generated layout should follow the CSS style, where each line starts with the object description and is followed by its absolute position.

Formally, each line should be like "object {width: ?px; height: ?px; left: ?px; top: ?px; }". The image is 64px wide and 64px high. Therefore, all properties of the positions should not exceed 64px, including the addition of left and width and the addition of top and height.

## In-context learning Examples

/\* first in-context exemplar\*/

Prompt: A building with a tower on top that contains a clock.

Layout:

```
clock {  
height: 37px;  
width: 38px;  
top: 104px;  
left: 60px;  
}
```



/\* second in-context exemplar\*/

...

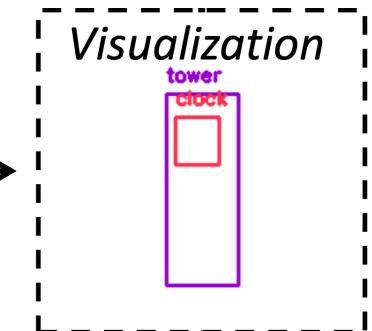
Inference prompt: Prompt: a tower of a building that has a clock on it.



/\* LLM Output\*/

```
tower {  
height: 149px;  
width: 56px;  
top: 65px;  
left: 100px;  
}...
```

In CSS format

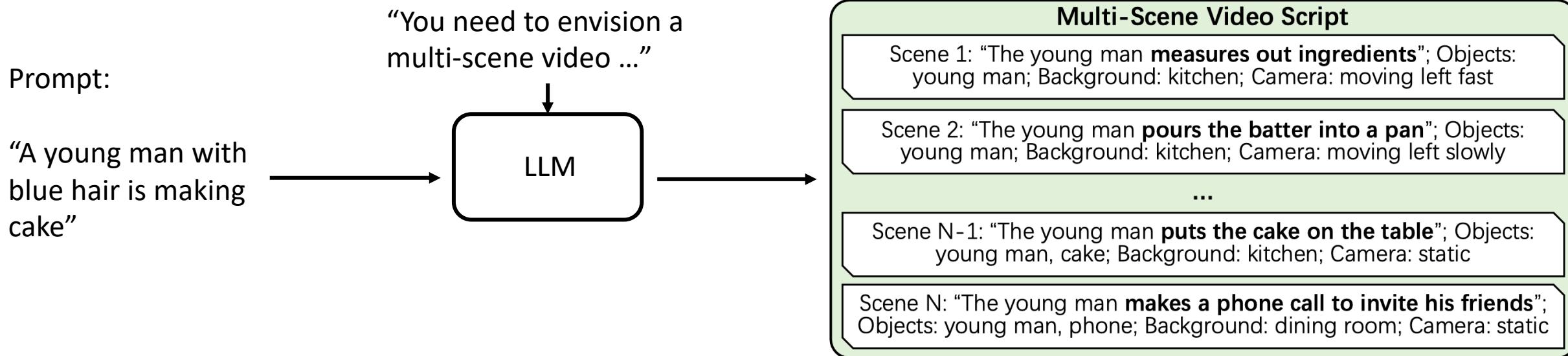


Layout-to-  
Image

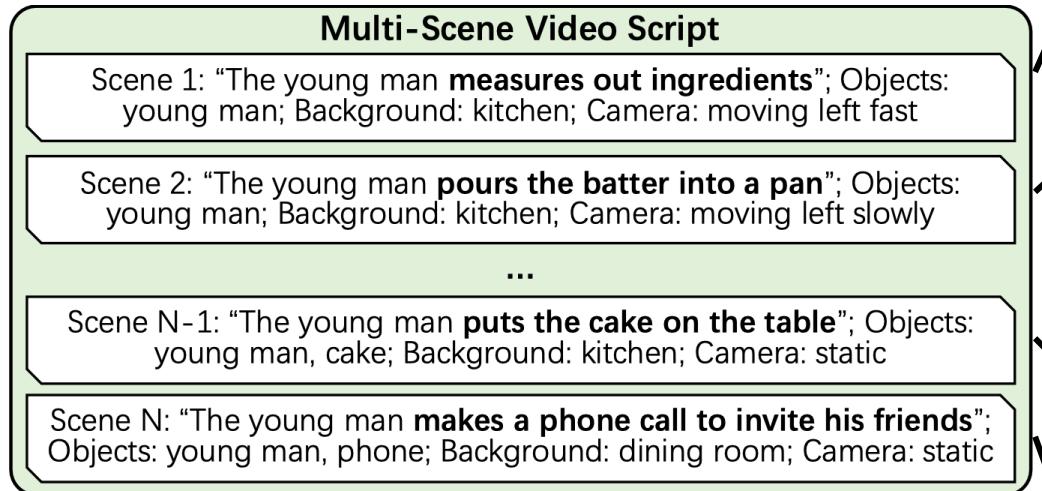


# LLM as Planner: From single-scene to multi-scene long video generation

## Prompt Decomposition

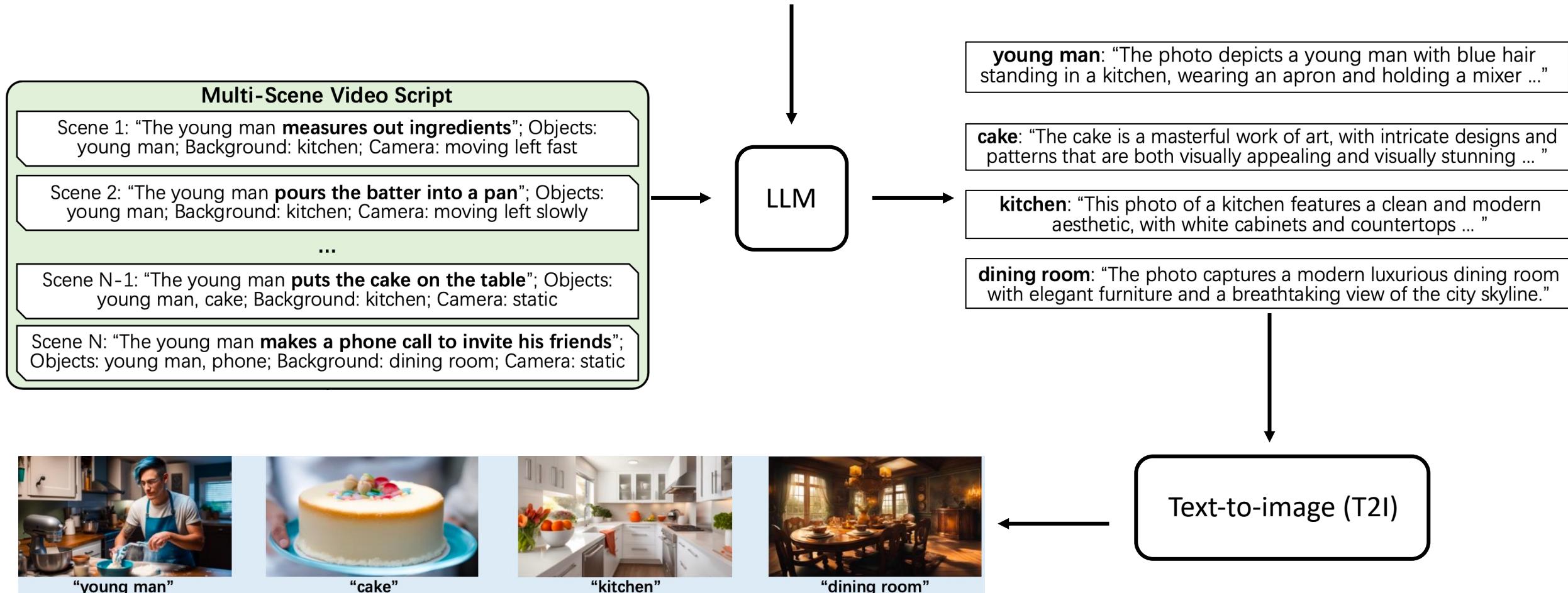


# Multi-scene video generation with the decomposed prompt



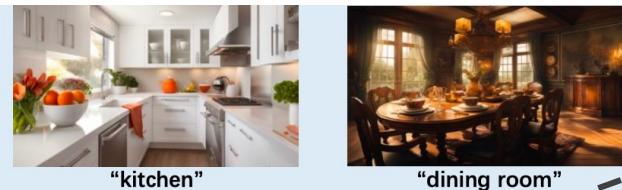
# Entity-specific descriptions generated by LLM

“You need to describe  
the entities in detail”



Pipeline: Text -> entity descriptions -> entity images -> scene images -> scene clips  
-> scene descriptions

"The young man puts the cake on the table"



Foreground and background controlled T2I

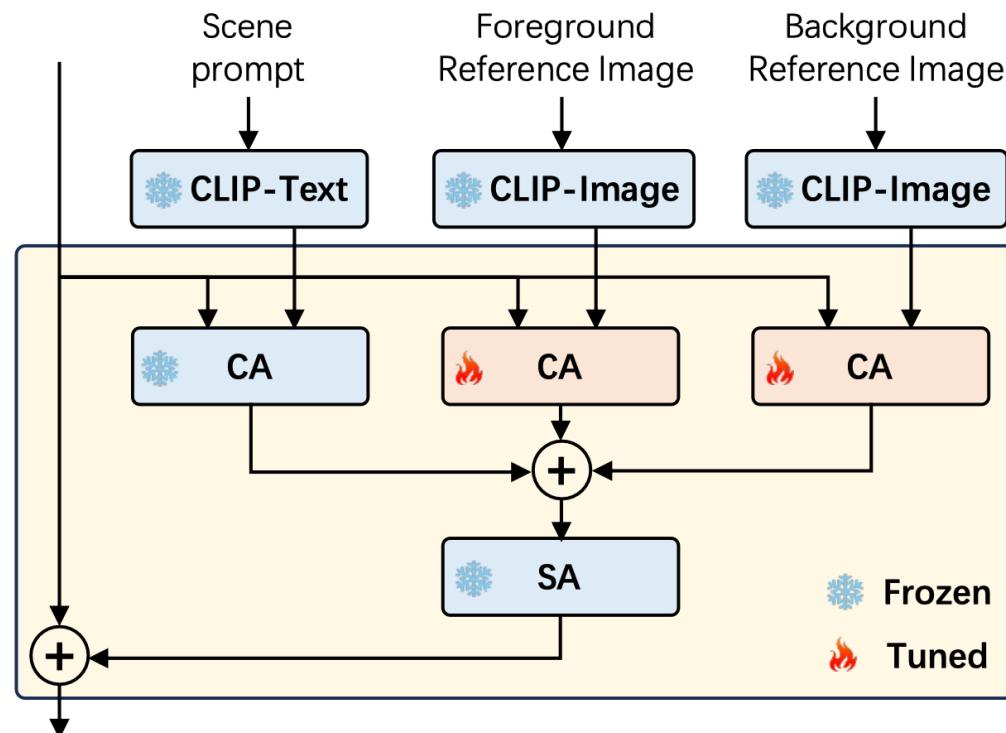


Image-to-video (I2V)



# LLM as Planner: Hierarchical prompts for tuning-free higher-resolution generation

Tuning-free higher-resolution image generation

"A corgi sits on a beach chair on a beautiful beach, with palm trees behind, high details"



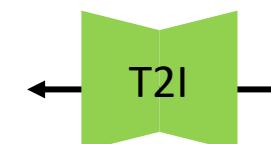
[H,W]

Upsample -> [2\*H, 2\*W]  
Split into patches  
With patch size = [H,W]



[2H,2W]

Textual Details Generation



SDE Edit  
And input to an intermediate timestep of T2I



Blurry Patches with [H,W]

# LLM as Planner: Hierarchical prompts for tuning-free higher-resolution generation

Hierarchical prompt: Generate image content of multiple scales via prompts with different granularity.

Local patch description

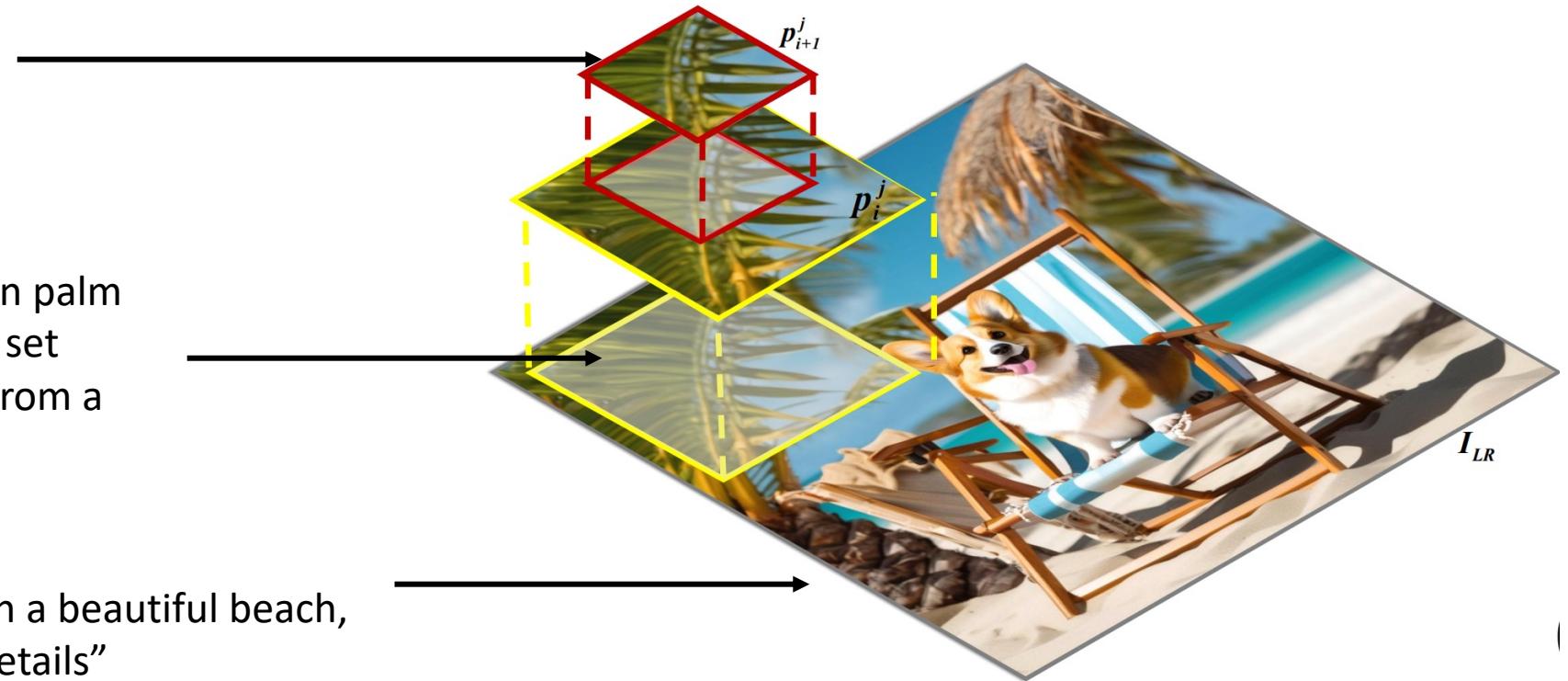
...

Local patch description:

A detailed image of a lush green palm leaf, with a vibrant yellow hue, set against a clear blue sky, taken from a close-up perspective.

Global description:

“A corgi sits on a beach chair on a beautiful beach, with palm trees behind, high details”

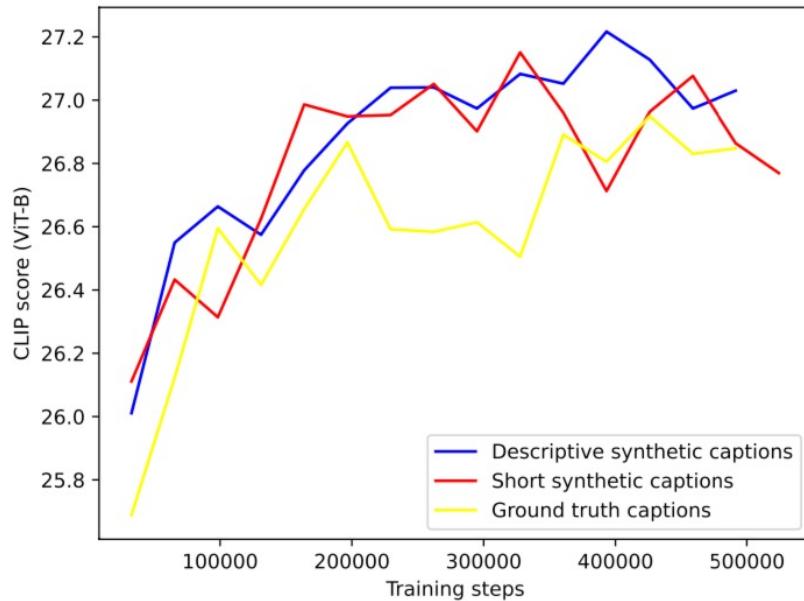


# LLM as Captioner

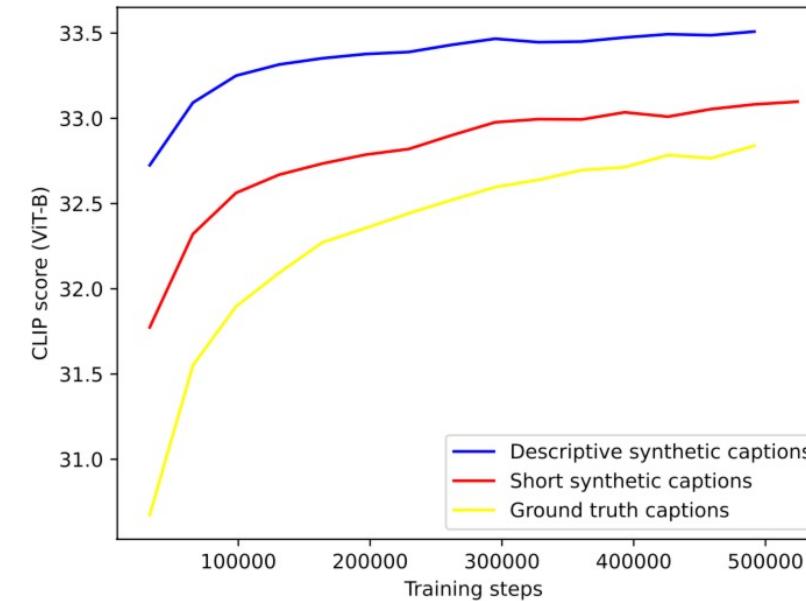
---

# LLM as Captioner

---



Evaluate with GT captions



Evaluate with descriptive synthetic captions

Training on long synthetic captions achieves the best performance with CLIP score.

# LLM as Captioner



COCO:

Young children standing on a platform waiting for a train to arrive.  
Adults and children watching a train slowly leave.  
A family near a railroad track watching the train pass.  
People waiting on a platform as a train pulls up.  
A train station with a green chain on the tracks and children waiting for it to go by.

LLaVA: At a train station, a group of people, including both young children and adults, are standing on a platform waiting for a train to arrive. The train is already present on the tracks, partially visible on the right side of the image. Some of the people watch the train closely, while others seem to be patiently anticipating its departure.

There is a total of eight individuals waiting for the train, with one child in the middle of the platform and the others scattered around. A backpack can be found on the far left side of the platform, suggesting that someone may have set it down while waiting.

ShareGPT4V: The image captures a moment at a train station. A green train is on the tracks, moving away from the platform labeled "Platform 2". The train's motion is observed by people standing on the platform, their attention drawn to the departing vehicle.

A red signal stands on the tracks, its vibrant color contrasting with the green of the train. Signs are prominently displayed around the platform. One warns "Beware of Trains", a cautionary reminder of the powerful machines that frequent these tracks. Another sign advises "Please Use The Footbridge To Cross The Line. If You Require Assistance Please Ask The Station Staff", guiding passengers to safely navigate the station.

The sky above is a clear blue, providing a serene backdrop to the bustling activity of the station. In the distance, trees can be seen, adding a touch of nature to this man-made setting. The image is a snapshot of everyday life at a train station, capturing both its routine operations and its inherent charm.

# LLM as Captioner

- Speech style descriptions



- Age: We first bin the age into 4 different categories namely less than twenty (<20), young adults (20-35), middle age (40-60), and elders (>60). We then fine-tune our dataset from pre-trained WavLM-base checkpoint with 3200 hours speech and age metadata from our training set (consisted of conversational and reading speech with various quality).
- Gender: We fine-tune on top of WavLM-base checkpoint with 4300 hours speech and gender metadata from our training set (consisted of conversational and reading speech with various quality).
- Audio Quality: We use TorchAudio-Squim [Kumar et al. \(2023\)](#) library and extract Perceptual Evaluation of Speech Quality (PESQ) [Rix et al. \(2001\)](#) score. We then bin the score into three categories: Low quality ( 0-2.39 ), Normal quality ( 2.39-3.8 ) and Studio Quality ( >3.8 ).
- Pitch: We use PyWorld vocoder <sup>6</sup> to extract fundamental frequency (f0) and then calculate the geometric mean across all voiced region. We use gender dependent threshold for binning the pitch into three different categories: low, normal, high. For gender masculine, we set low pitch (0-40 percentile), normal pitch (40-90 percentile) and high pitch (>90 percentile). For gender feminine, we set low pitch (percentile 0-10), normal pitch (10-60 percentile) and high pitch (>60 percentile). The logic behind asymmetric threshold is because in general people will perceive most of masculine voice have lower pitch and most of feminine voice have higher pitch.
- Speaking rate: Given the transcript and audio, we first apply VAD to remove the silence segments. We then calculate character per seconds (CPS) and bin them into 3 categories: slow (<9.2 CPS), high (>20.8 CPS) and normal (9.2 <= x <= 20.8 CPS).
- Accent: We use the accent from the metadata whenever available in the metadata, otherwise leave it blank.
- Emotion: We use the emotion labels whenever available in the metadata, otherwise we leave it as blank.
- Environment: We use the environment tags such as inside a room, outside whenever available from the datasets.

- A young male adult voice, conveys anger and frustration. The audio, of normal quality, is recorded inside a small space. The person speaks with South Asia accent and a normal speaking pace.
- This young bloke's ticked off, audio's all good. He's in some small space and has a South Asian accent. Talks normal speed.
- Got this young dude who's mad, audio's decent. He's in a tight spot, has that South Asian accent, and talks at a chill pace.
- Young man is angry. Audio is okay, small place. Accent from South Asia. Speaks normal.

# LLM as Conditioner

---

# LLM as Conditioner

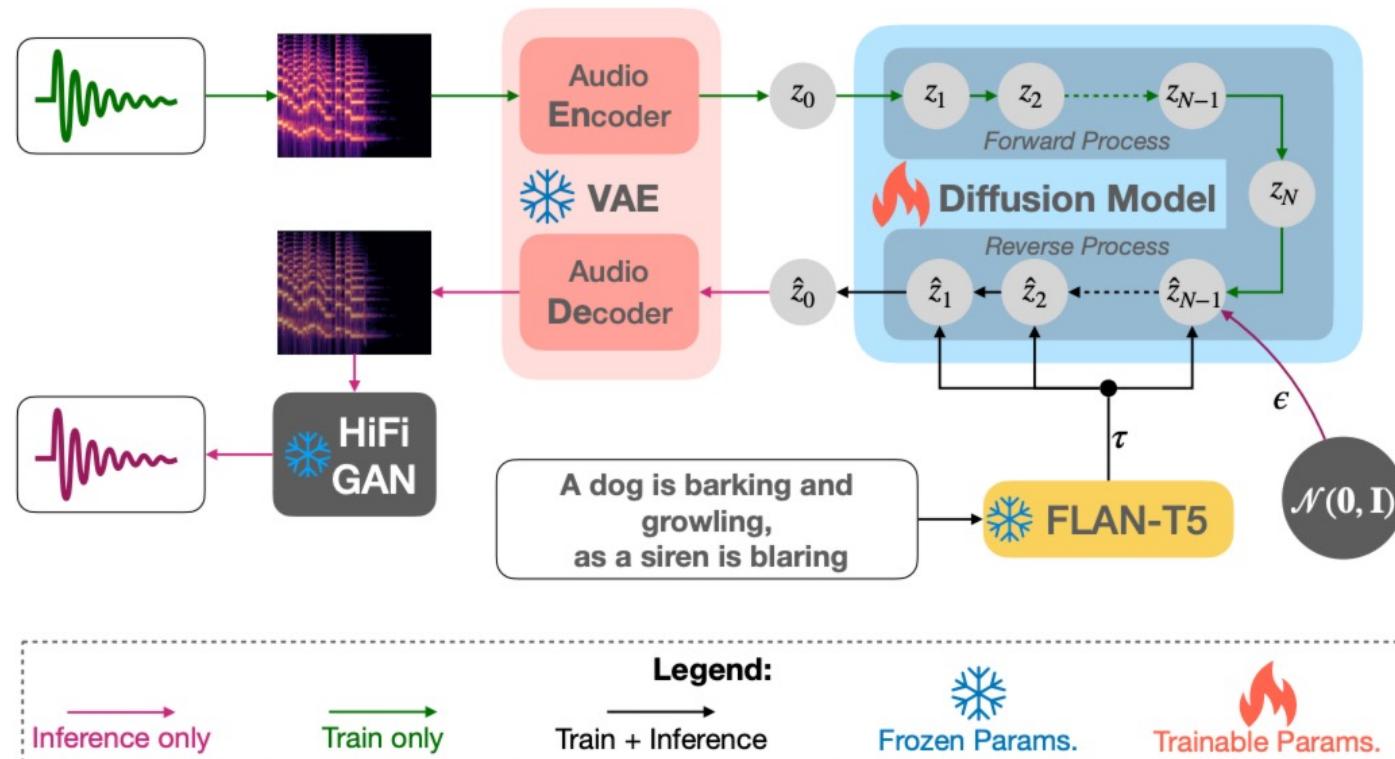


Figure 1: Overall architecture of TANGO.

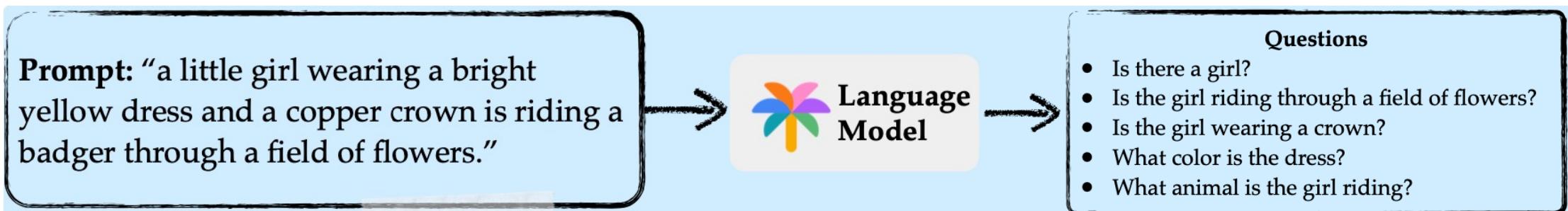
# **LLM as Evaluator**

---

# LLM as Evaluator

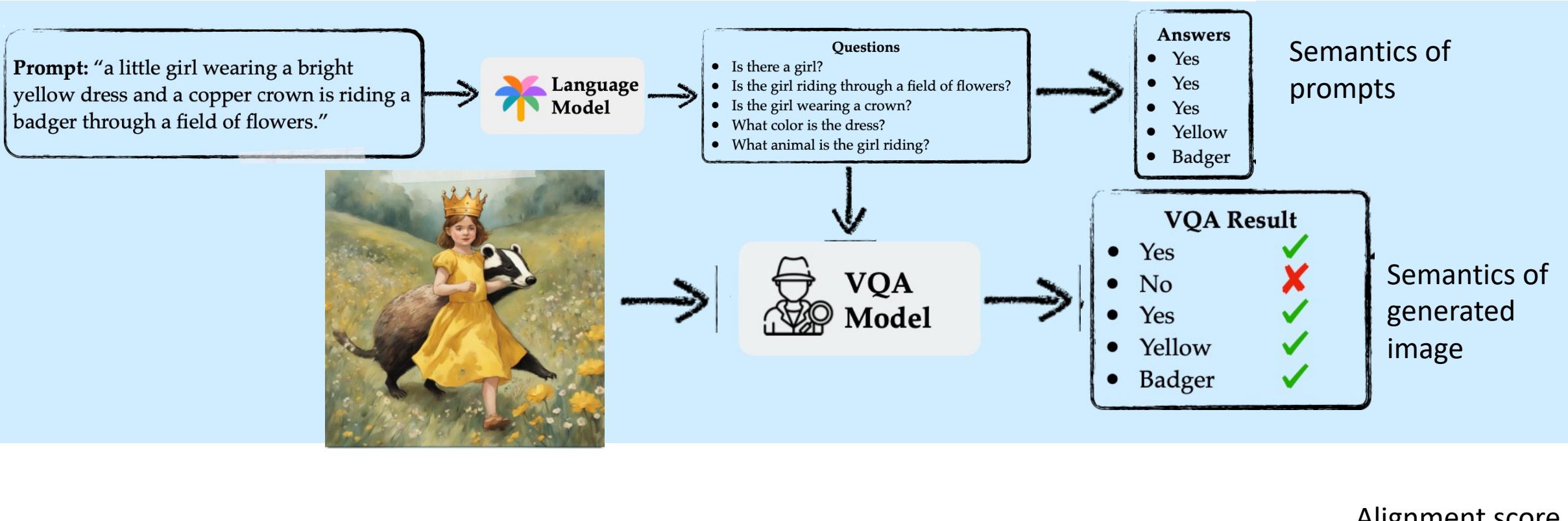
---

Use LLMs to generate prompt-related questions



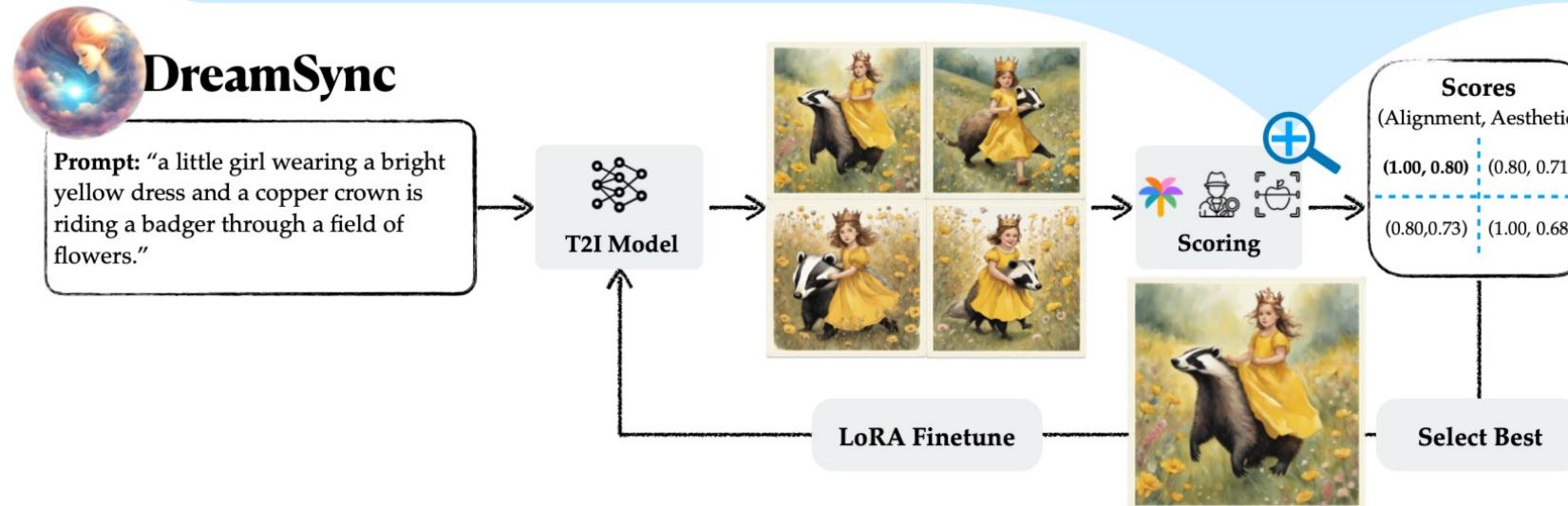
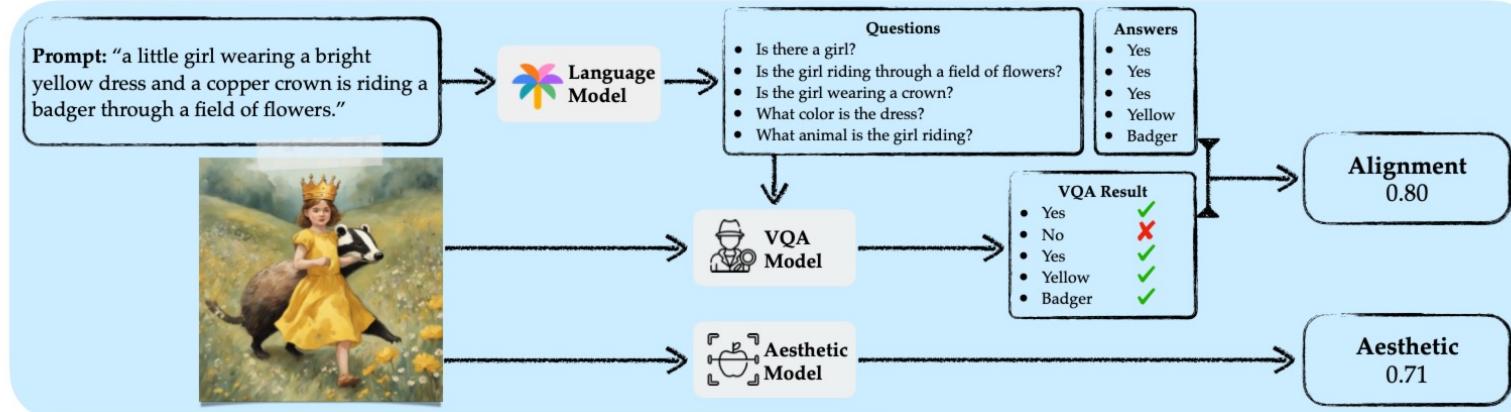
# LLM as Evaluator

Questions and generated image are input to a VAQ model



# LLM as Evaluator

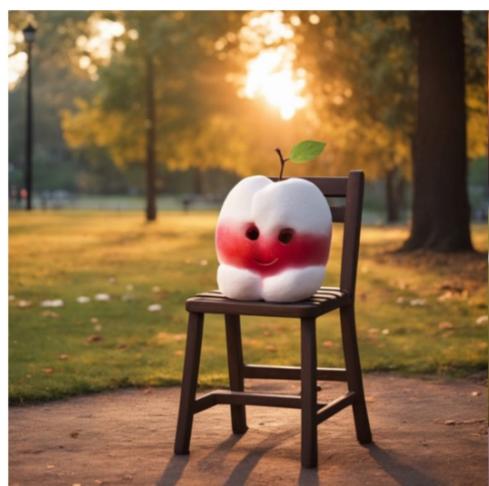
Then conduct lora finetune to use a new subset of generated images to finetune the T2I



DreamSync: Aligning Text-to-Image Generation with Image Understanding Feedback

# LLM as Evaluator

## Improved semantics



Before



After



Before



After

# LLM as Multimodal Agent

---

# LLM as Agents: Dealing with complex tasks/ user instructions

Complex tasks:

- Complex content:
  - “An oil painting, where a green vintage car, a blue scooter on the left of it and a black bicycle on the right of it, are parked on the road, with two birds in the sky.”
  - Multiple objects; Attribute binding; Position relationship; Object quantity;
- Multiple-task / Multi-modal system:
  - “Can you generate a video of a car running on the road” [Video Generation]
  - -> “Can you generate the corresponding sound” [Video-to-audio Generation]
  - -> “Can you modify the car into a big blue truck” [Video Editing]

# LLM as Agents: Dealing with complex tasks/ user instructions

## A Basic Pipeline

Task/User instructions

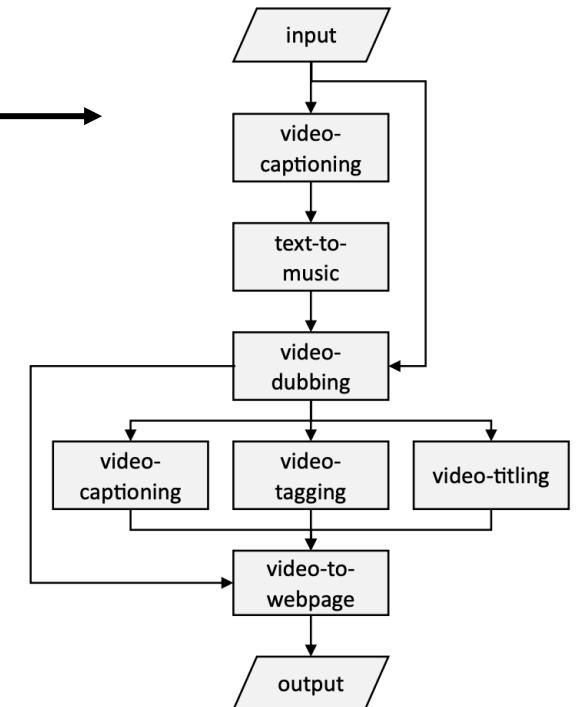
→ Task Decomposition



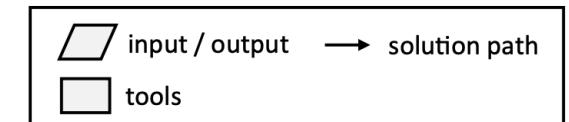
**SubTask 1**  
**Description:** Dub the video with a background music.  
**Domain:** video-editing  
**Args:** <video>  
**Return:** <video>

**SubTask 2**  
**Description:** Generate a HTML page to introduce the video.  
**Domain:** HTML-generation  
**Args:** <video>  
**Return:** <HTML>

## Task Planning



Searching optimal solution on the tool graph



# LLM as Agents: Dealing with complex tasks/ user instructions

## A Basic Pipeline

Task/User instructions



I have designed the HTML web page to introduce the video with light background music:



Task Decomposition

### SubTask 1

Description: Dub the video with a background music.

Domain: video-editing

Args: <video>

Return: <video>

### SubTask 2

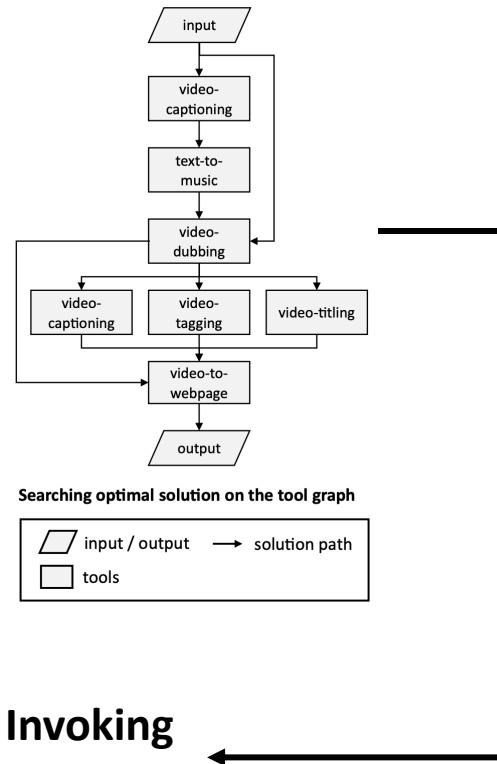
Description: Generate a HTML page to introduce the video.

Domain: HTML-generation

Args: <video>

Return: <HTML>

Task Planning



Summarizing the outputs

Tool Invoking & Execution

### Toolbox

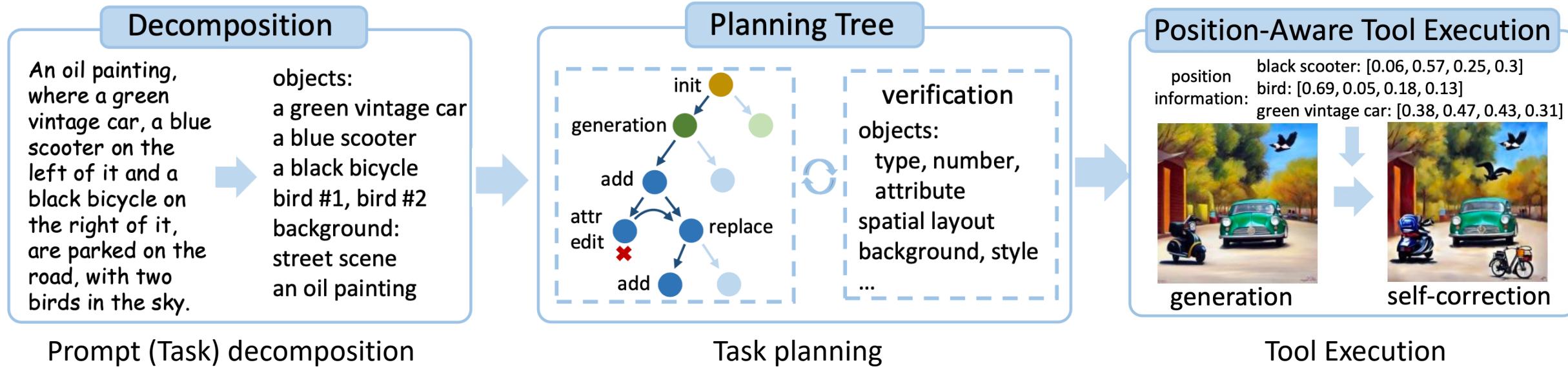
- Video captioning model
- Text-to-music model
- Video-to-music model

...

Finish the task

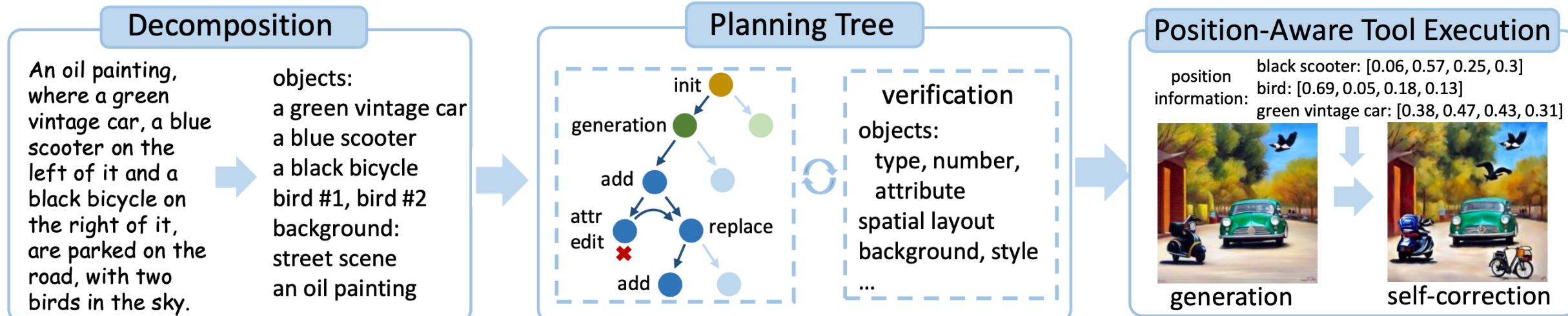
# LLM as Agents: Dealing with complex tasks/ user instructions

Another example for Text-to-image generation and editing



# LLM as Agents: Dealing with complex tasks/ user instructions

Another example for Text-to-image generation and editing

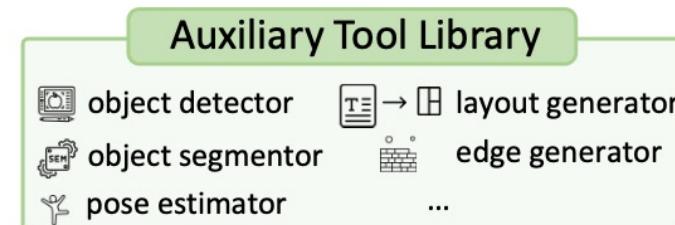
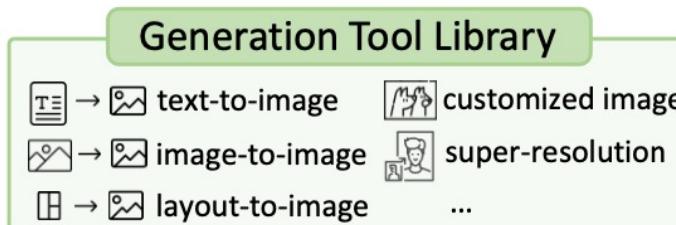


Prompt (Task) decomposition

Task planning

Tool Execution

Toolbox



# LLMs Meet Image and Video Generation

## Takeaways:

LLMs help visual generation in various aspects:

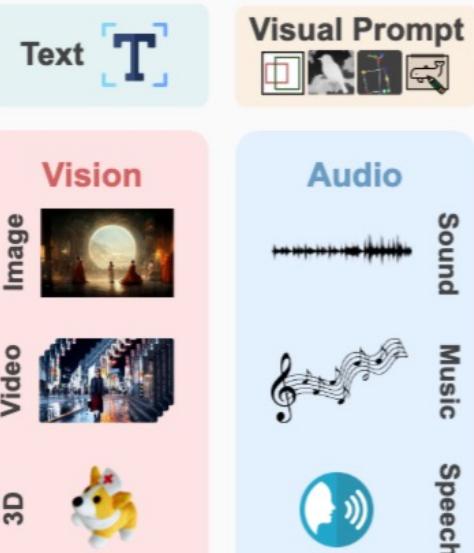
1. Unified backbone
2. Planner
3. Captioner
4. Conditioner
5. Evaluator

## What's next:

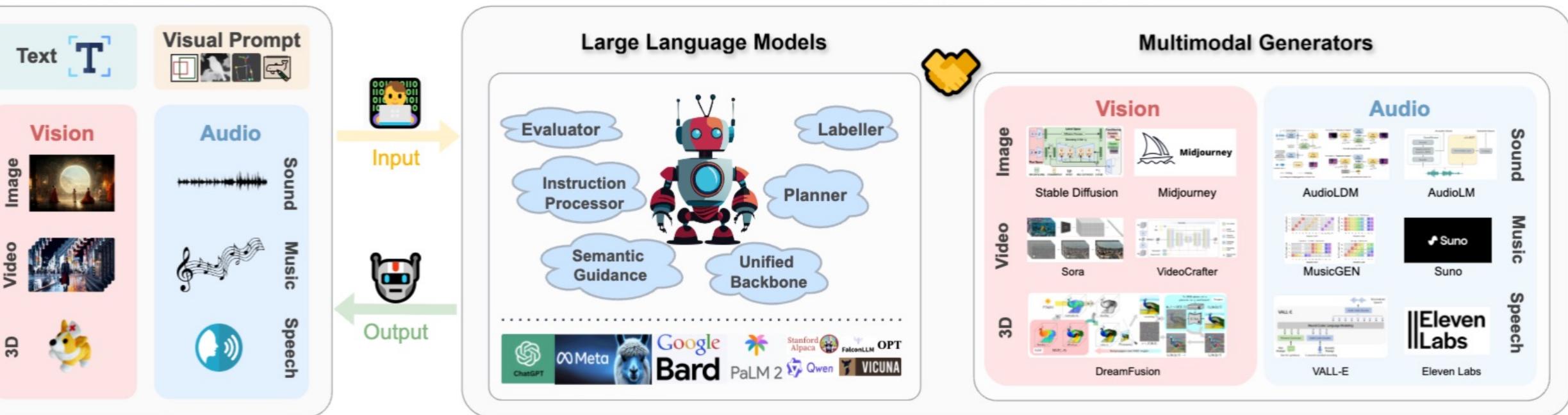
1. Leverage powerful LLMs to improve the visual generation performance, learning efficiency. More diverse methods and explorations from different aspects.
2. Integrate visual generation models into LLMs to build flexible multimodal LLM systems.

# LLMs Meet Multimodal Generation and Editing: A Survey

## Multimodal Content



## Generative Engine



<https://github.com/YingqingHe/Awesome-LLMs-meet-Multimodal-Generation>

# Thank You!