Basics of Multi-modal Models (MMMs)

CSE 585 Advanced Scalable Systems for GenAI

Presentation

Sep 3, 2024

Shiqi He, Insu Jang, Mosharaf Chowdhury

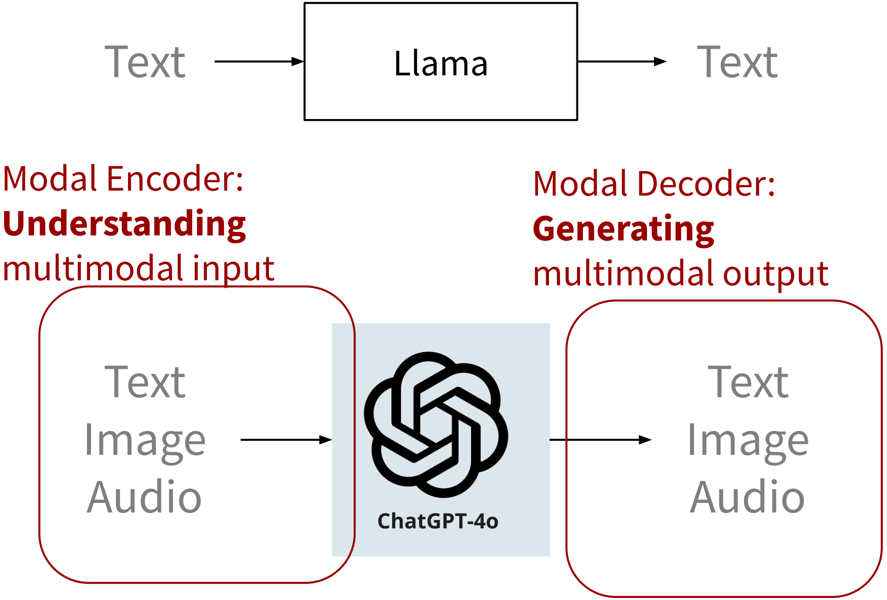
Agenda

- Background
  - Multi-modalities
  - CLIP: Contrastive Language-Image Pre-training
- Multi-modal Language Models
  - Vision-Language Models (VLM)
  - Any-to-Any Multi-modal Language Models
- Diffusion Models
  - Diffusion Process and Diffusion Models
  - Latent Diffusion Models (LDM)

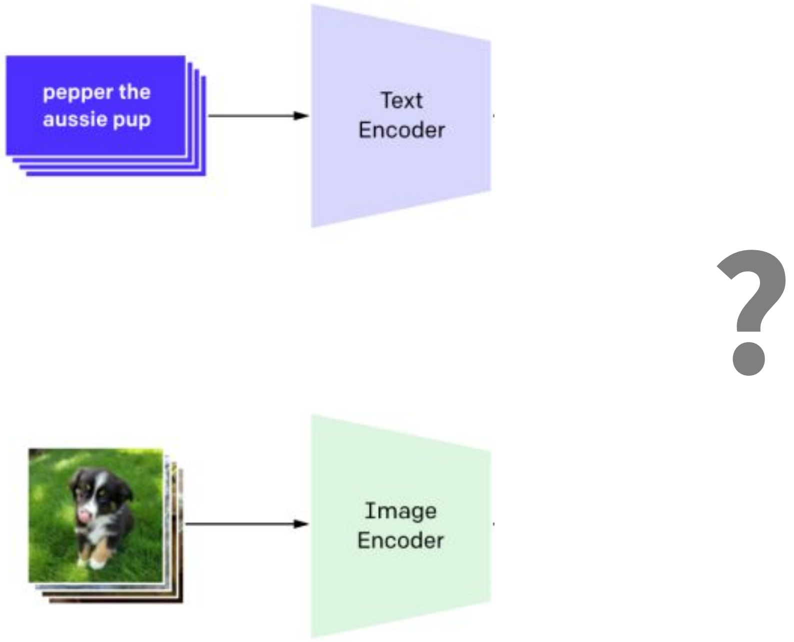
Background

Multi-Modalities: Towards Human-like AI

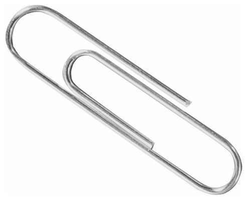
- Human intelligence: uses more types of data than just text
- Modality: ways in which we perceive and express information



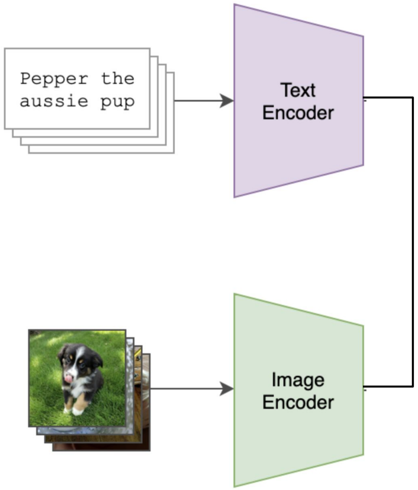
How to Connect Different Modalities?



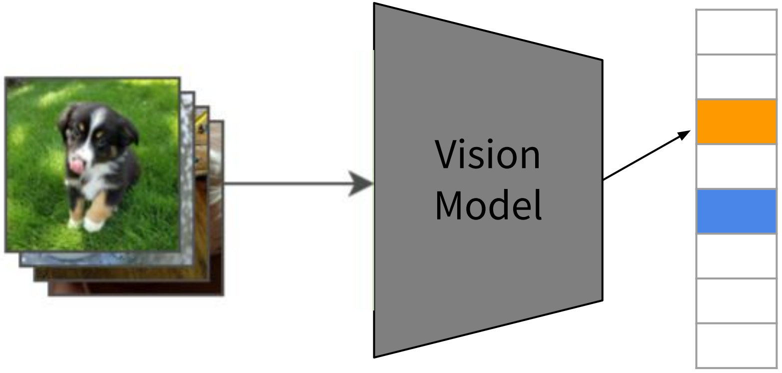
All Began with... CLIP



- OpenAI CLIP (ICML21): Contrastive Language-Image Pre-training
  - Connect textual descriptions with images

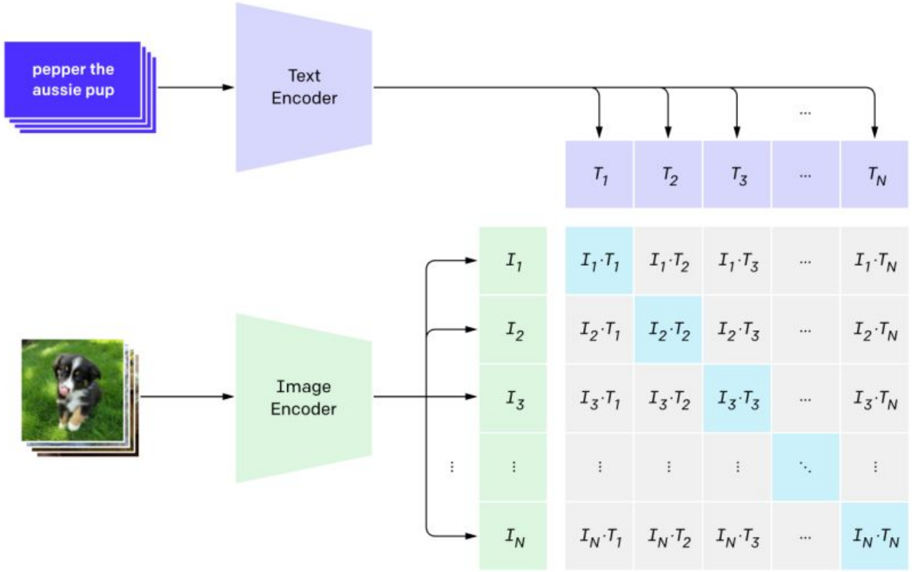


vs. Simple Vision Model Training (classification)



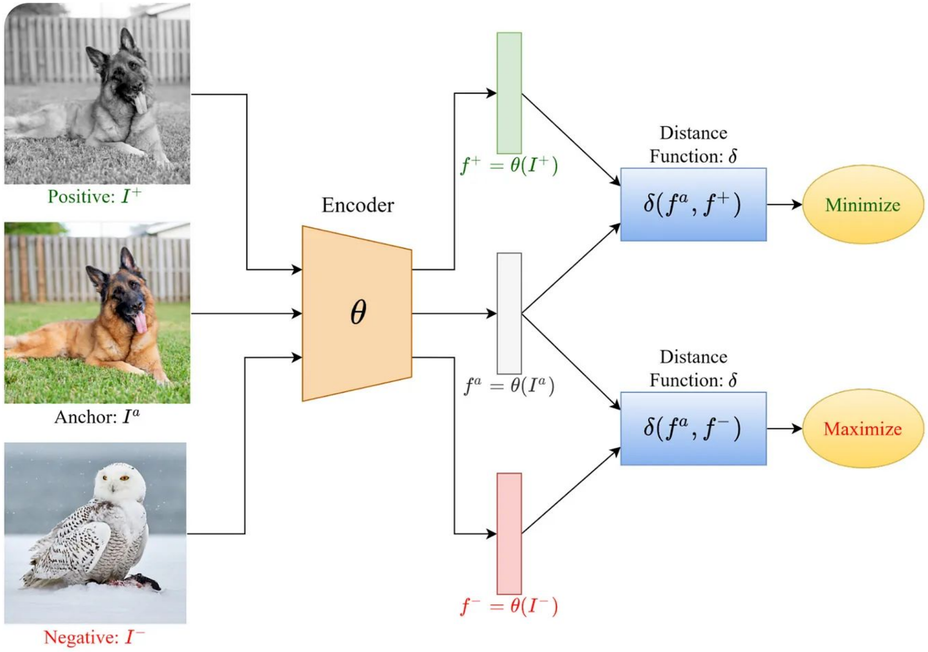
Compare classification ground-truth labels

CLIP Overview - Training



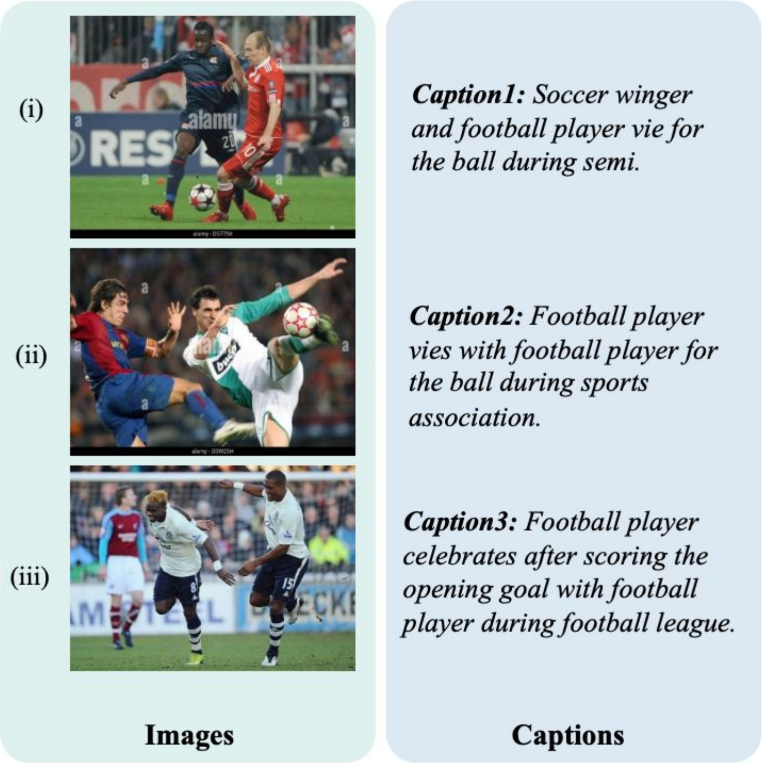
- Learn the relationship between textual description and image
  - Positive score for its own label
  - Negative score for all the others

CLIP Overview - Training

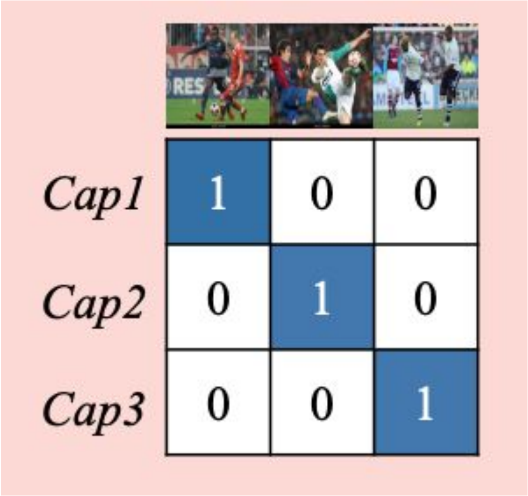


- Use (Anchor, positive, negative) set to train the model
  - Anchor: an image from one class
  - Positive: another image in the same class with the anchor image
  - Negative: an image from different class
- Minimize distance (Anchor, Pos)
  Maximize distance (Anchor, Neg)

CLIP Limitations

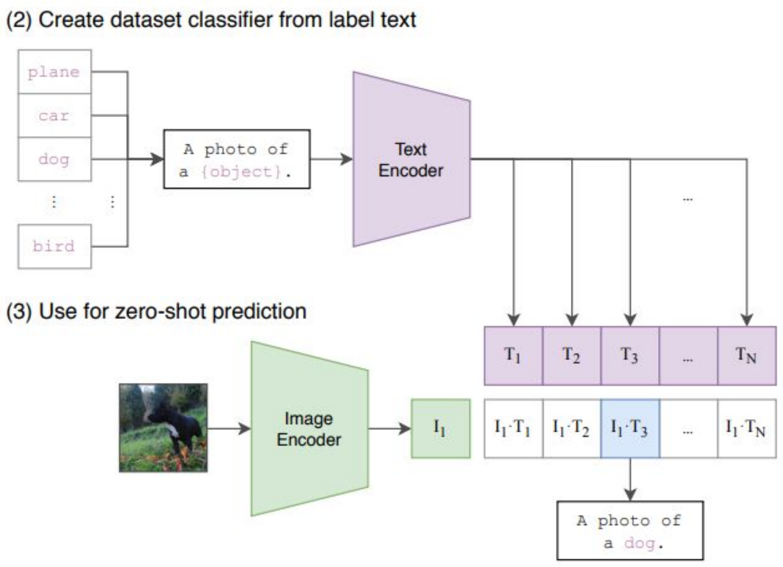


- Is it really correct to mark all the other images simply negative?



SoftCLIP: Softer Cross-Modal Alignment Makes CLIP Stronger [AAAI'24]

CLIP Overview - Inference



- Select a class with the highest similarity with the image, and fill the given sentence (zero-shot)

A photo of a {object}. + (with a dog image)

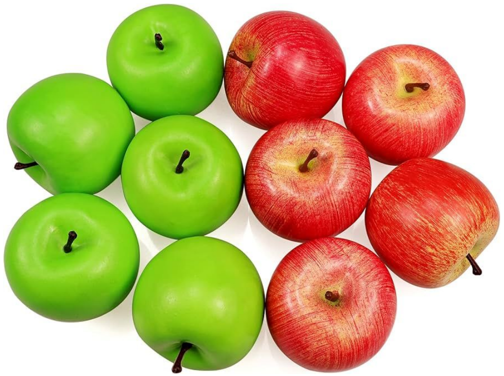
A model is asked to fill a proper object name into the text.

→ A photo of a dog.

Multi-modal Language Models

What Is a Vision-Language Model (VLM)?

- **Exploit the reasoning and generation capability of LLM** to multi-modal models



CLIP

There are {object}s in the image.

There are apples in the image.

vs

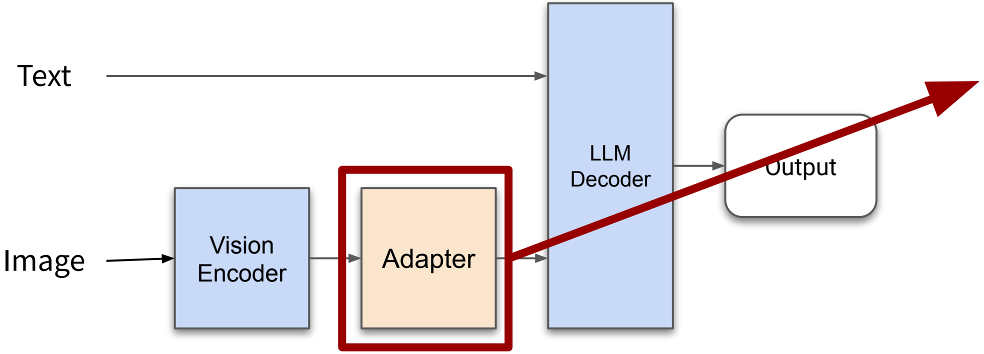
VLM

What is in the image?

There are apples in the image,
some are green and others are red.

Aligning Modality Representation

- Project modality embedding space (output representation) to LLM embedding space to make LLM understand various modality representation
- Various implementations of adaptors are proposed

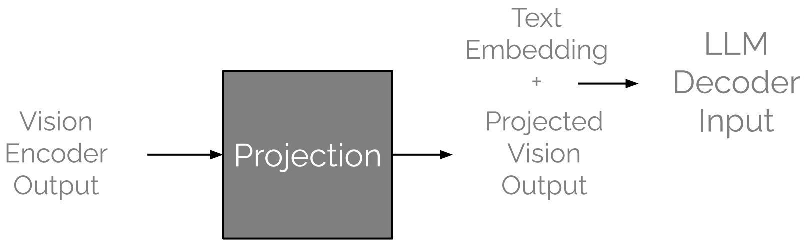


Linear
MLP
Cross-attention
Q-former
etc

Different Types of Adaptors

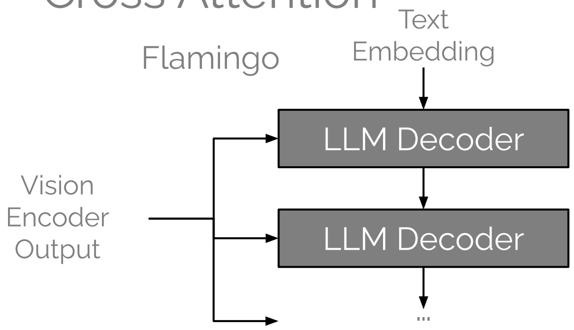
Linear

LLaVA



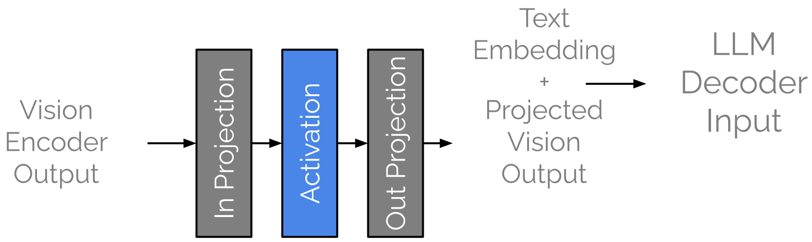
Cross Attention

Flamingo



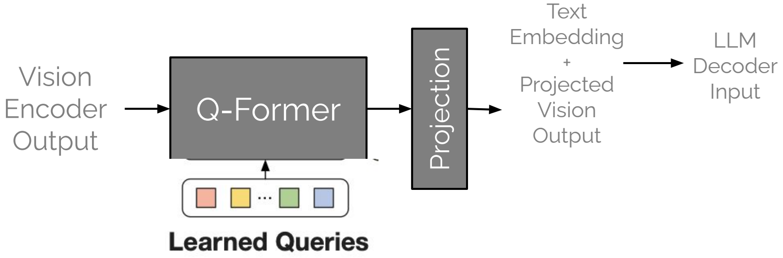
MLP

LLaVA-Next



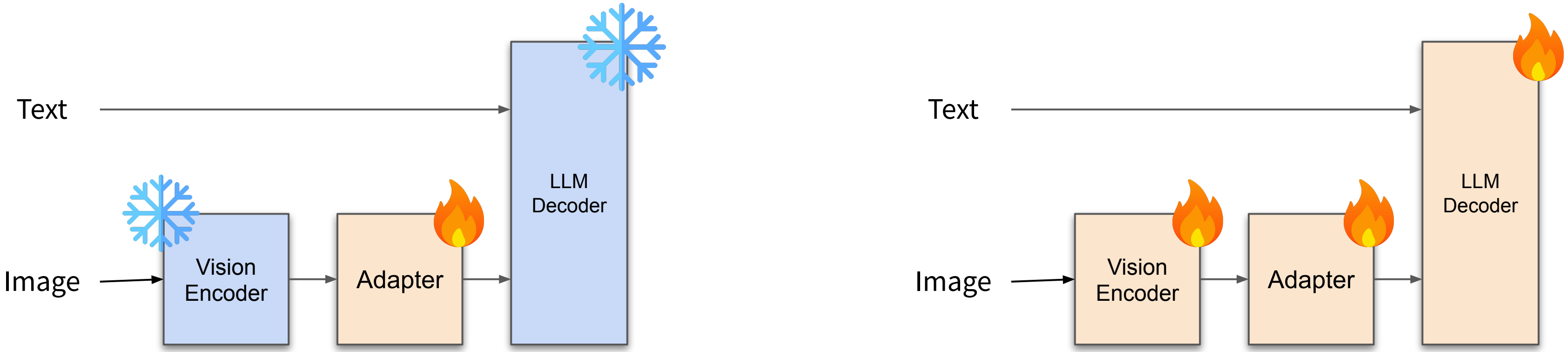
Querying Transformer (Q-former)

BLIP-2



Training VLM

- **Common to use pre-trained vision encoder and LLM** instead of training entire model from scratch to reduce training cost
- Two-stage training
  a. Pre-train a projector between vision encoder and LLM
  b. Instruction-tuning the entire model solve unseen problems

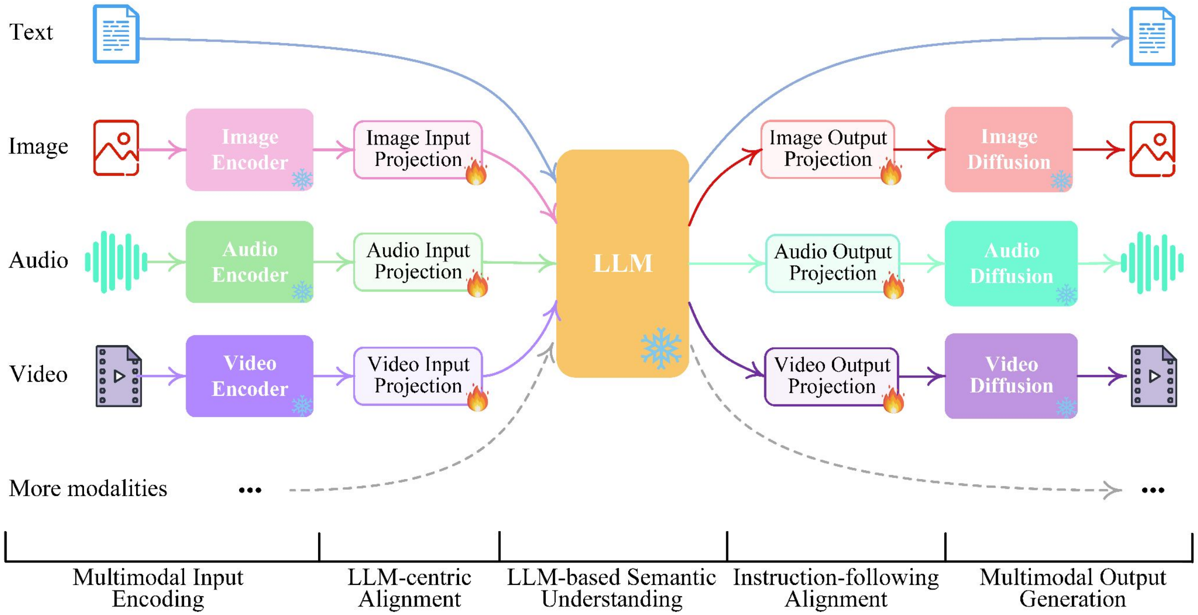


Stage 1: Pre-training a projector

Stage 2: instruction tuning

Towards General Any-to-Any Multimodal LLM

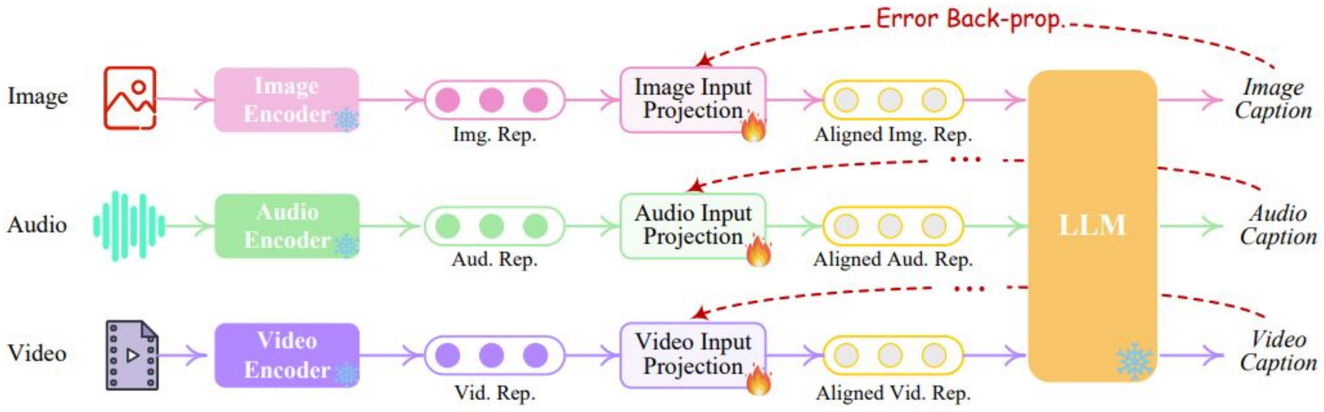
- Example: NextGPT



	Input	Output
LLM	Text	Text
VLM	Image, Text	Text
Any-to-Any MLLM	Any	Any

Any-to-Any MLLM Training: Encoder

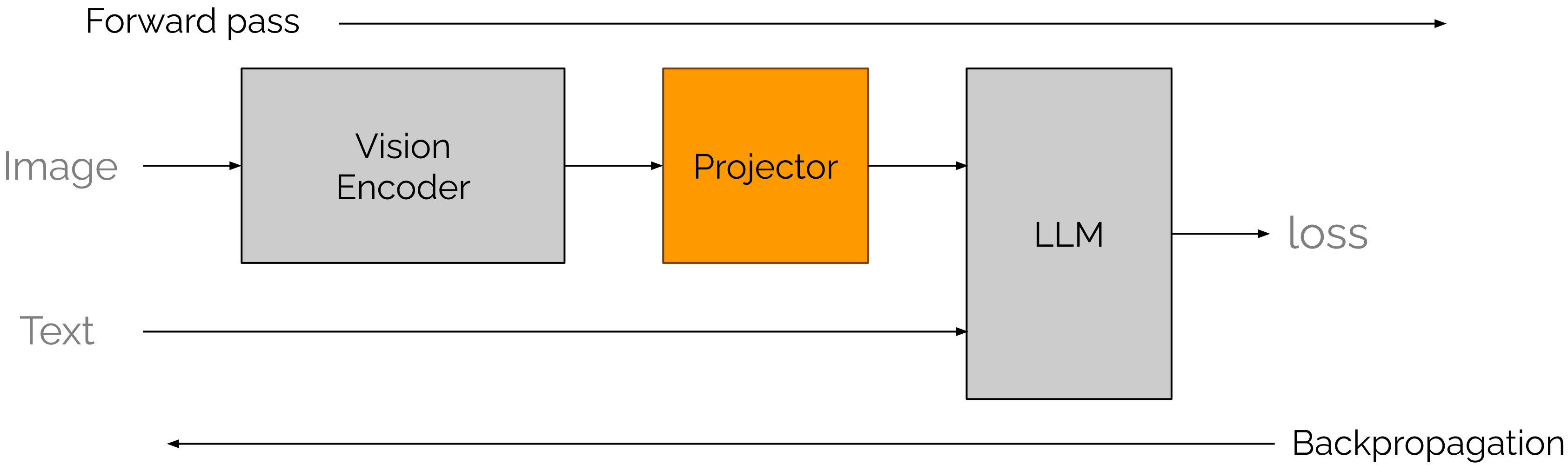
- Encoding-side (understanding)
  - Similar to training VLM: align encoder and LLM embedding space



(a) Encoding-side LLM-centric Alignment

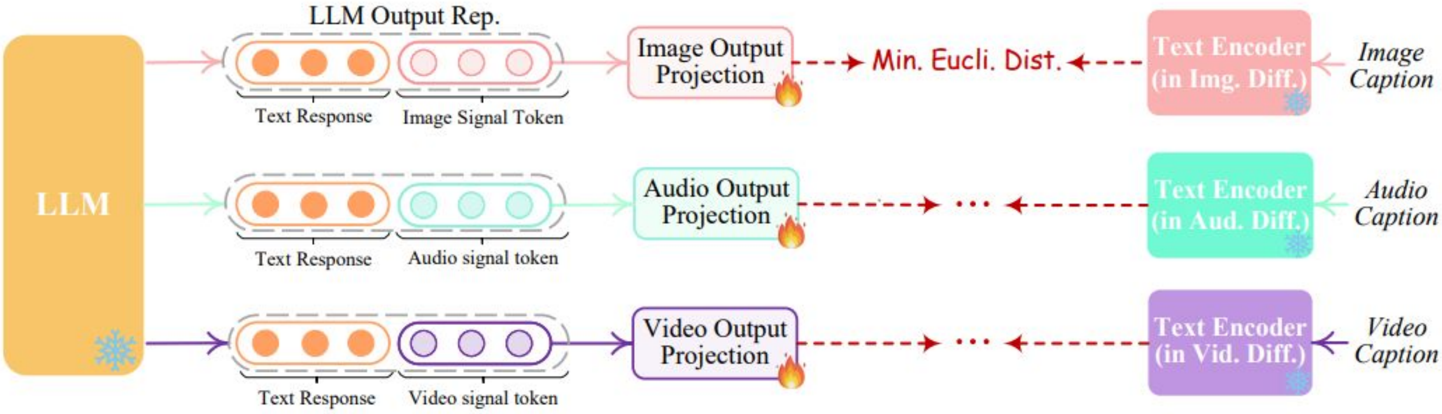
Any-to-Any MLLM Training: Encoder

- Encoding-side (understanding)
  - Similar to training VLM: align encoder and LLM embedding space



Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
  - Minimize the distance between the LLM's output representations and the conditional text representations of the diffusion models



(b) Decoding-side Instruction-following Alignment

Figure 3: Illustration of the lightweight multimodal alignment learning of encoding and decoding.

Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
    - First need to see generation model architecture briefly

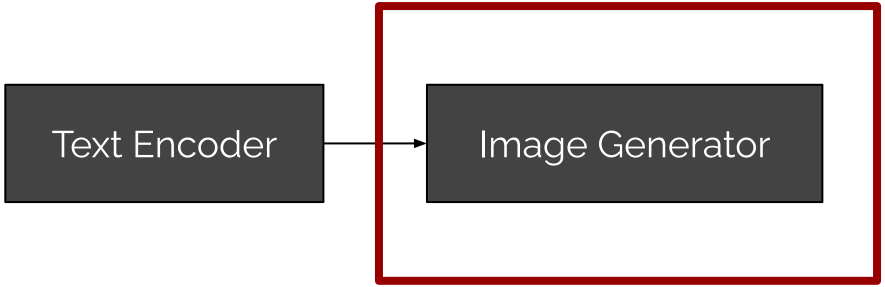
Prompt:
Tom Cruise the actor in Iron Man Suit from Marvel → Text Encoder → Image Generator →

Text encoder for understanding the given prompt

https://stablediffusionweb.com/image/13298136-tom-cruise-in-marvel-s-iron-man-suit

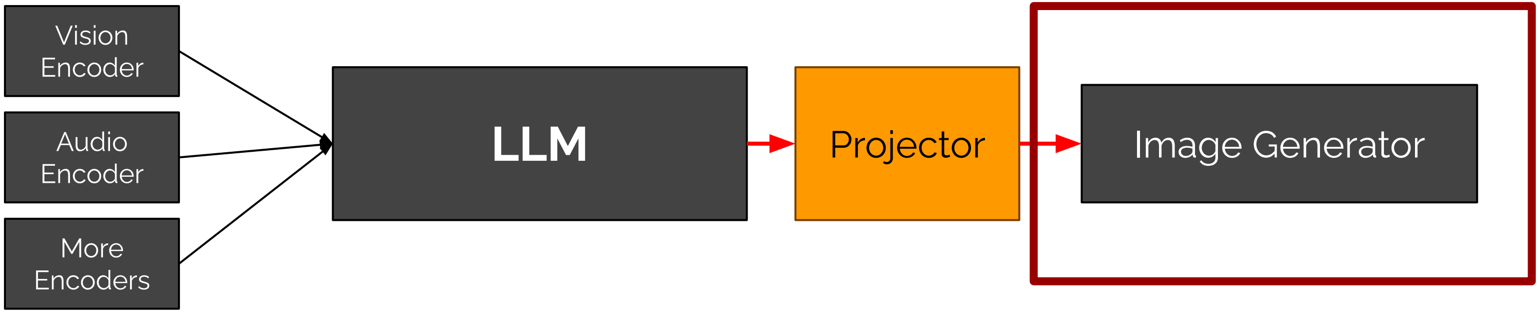
Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
  - **Use the pre-trained image generator part** in Any-to-Any
  - Replace model's own text encoder with LLM in Any-to-Any architecture



Any-to-Any MLLM Training: Decoder

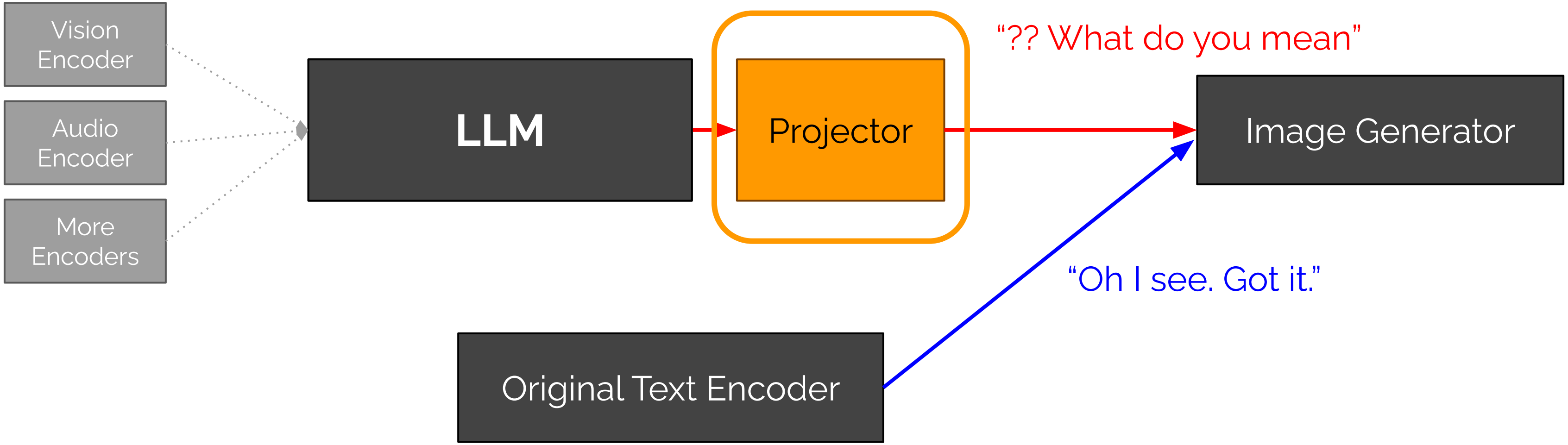
- Decoding-side (Generation)
  - **Use the pre-trained image generator part** in Any-to-Any
  - Replace model's own text encoder with LLM in Any-to-Any architecture



Any-to-Any MLLM Training: Decoder

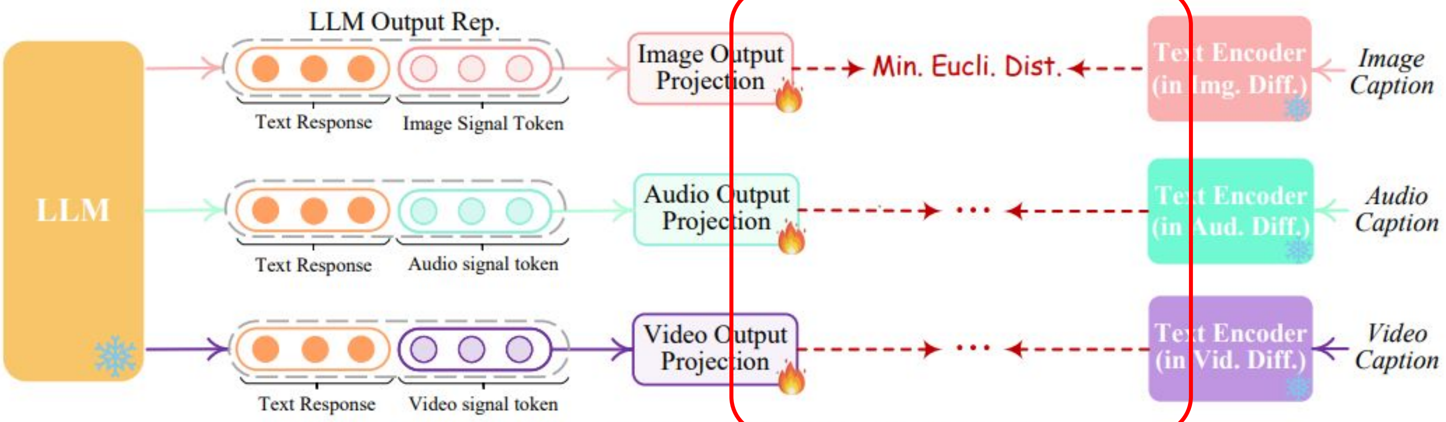
- Decoding-side (Generation)
  - Image generator is not aligned with LLM
  - Projector between LLM and generator aligns their embedding space

Trains the projector to align outputs of LLM with those of text encoder so that image generator can understand what LLM is saying



Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
  - **Minimum Euclidean distance** is used as loss in decoding-side training
    → Make output closer to that from text encoder

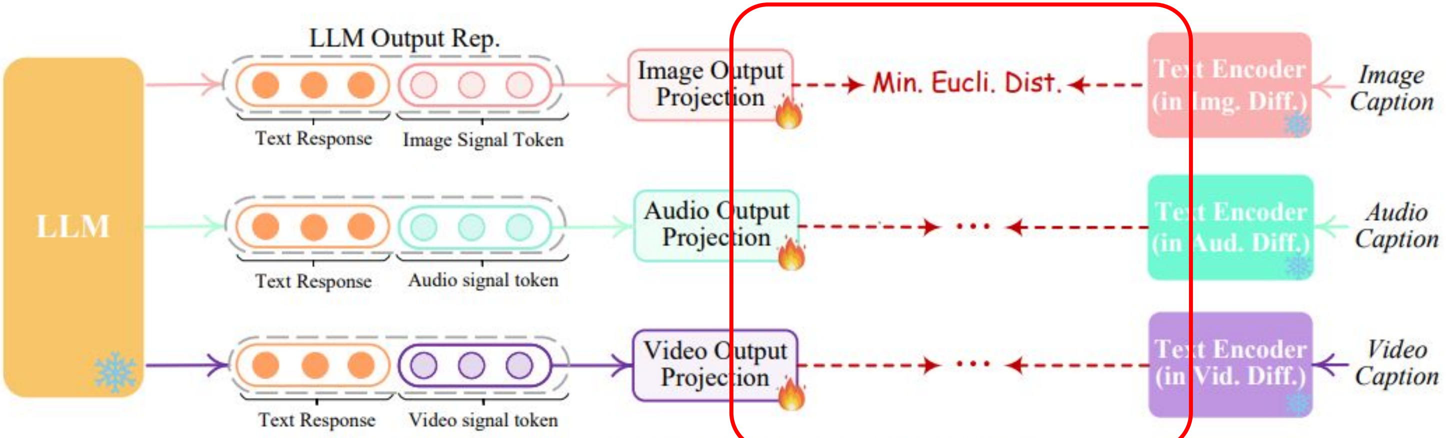


(b) Decoding-side Instruction-following Alignment

Figure 3: Illustration of the lightweight multimodal alignment learning of encoding and decoding.

Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
  - **Minimum Euclidean distance** is used as loss in decoding-side training → Make output closer to that from text encoder

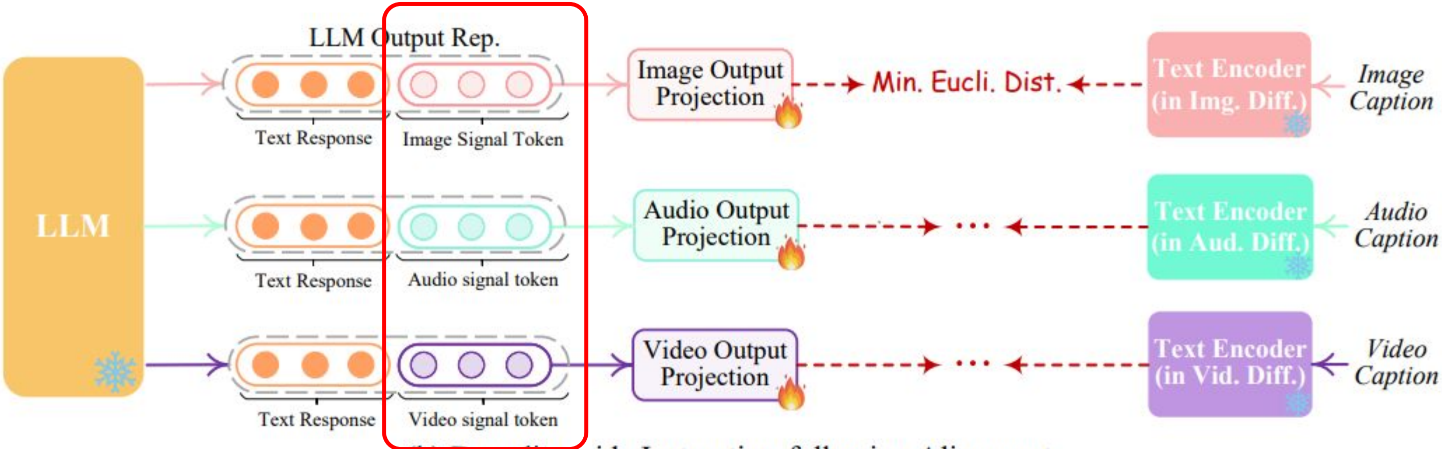


(b) Decoding-side Instruction-following Alignment

Figure 3: Illustration of the lightweight multimodal alignment learning of encoding and decoding.

Any-to-Any MLLM Training: Decoder

- Decoding-side (Generation)
  - The model needs to learn **when to make a specific modality output**
  - LLM is fine-tuned to **generate specific modality signal tokens**



(b) Decoding-side Instruction-following Alignment

Figure 3: Illustration of the lightweight multimodal alignment learning of encoding and decoding.

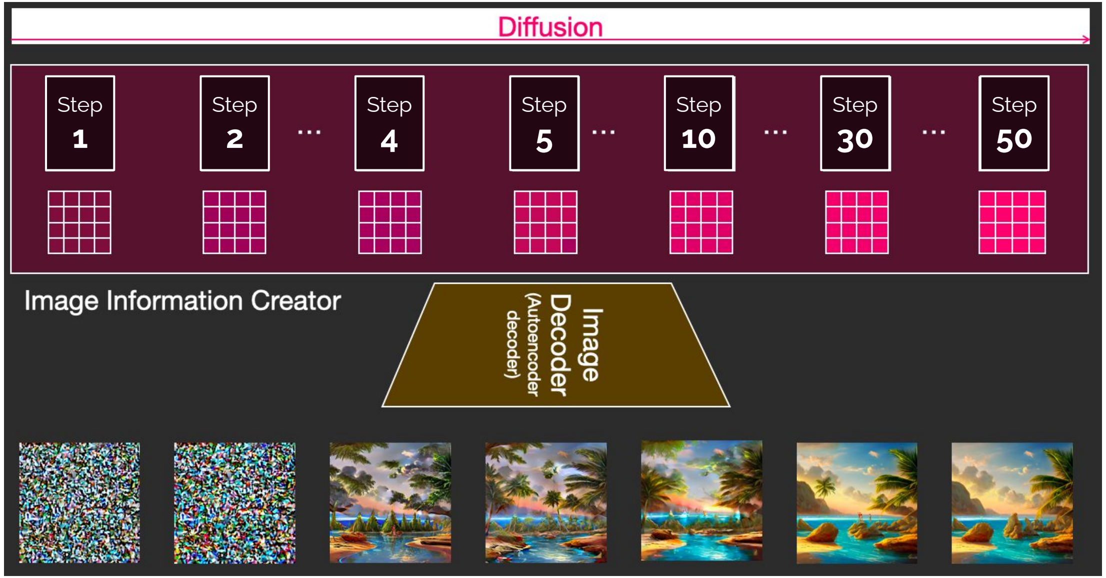
Diffusion Models

What is a Diffusion Model?

- Use diffusion process to generate continuous data (e.g. image, video, audio)
- Generation Tasks
  - Image - Stable diffusion
  - Video - Sora, AnimateDiff, Zeroscope
  - Audio - AudioLDM
- Not all generation tasks are done by diffusion. But diffusion is becoming the trend.

Diffusion Process

- Start from full random Gaussian noise, **Iteratively denoise** to create an image
- Introduced in Denoising diffusion probabilistic model (DDPM) [NeurIPS'20]



https://jalammar.github.io/illustrated-stable-diffusion/

Diffusion Model: Training

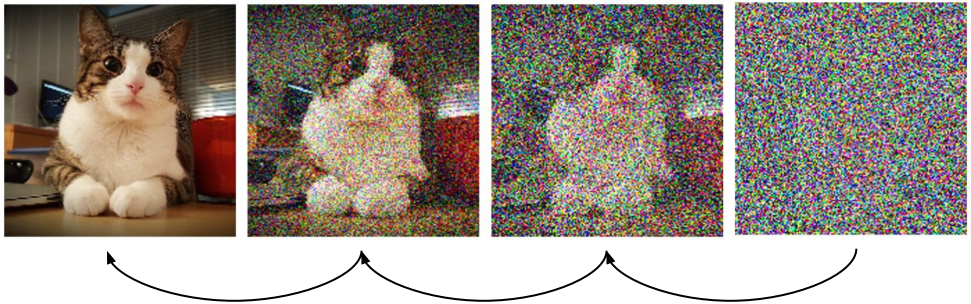
- Forward diffusion process: from original image, gradually add noise to create steps

Forward diffusion process: data to noise

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Diffusion Model: Training

- Forward diffusion process: from original image, gradually add noise to create steps
- Reverse diffusion process: Model is trained to **recover the previous step** from the current one by denoising



Backward diffusion process: denoise image with prompt embedding information

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Latent Diffusion Model (LDM)

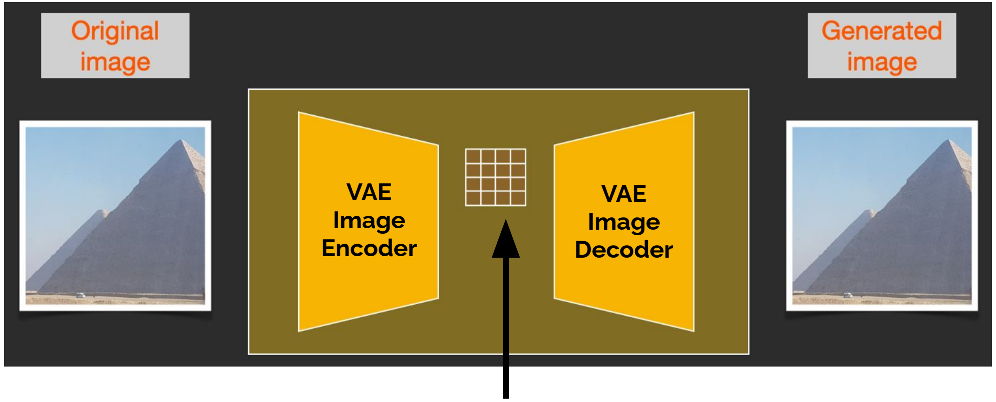
- Original diffusion model (DDPM) **operates in pixel space**
  Require more resources to train and inference
  - Pixel space dimension: 512 x 512 x 3 = 786432
  - Latent dimension for Stable Diffusion: 64 x 64 x 3 = 12288 (64 times smaller!)

Train OpenAI ADM		Train LDM
	vs	
150~1000* V100 GPU days		80~270 V100 GPU days

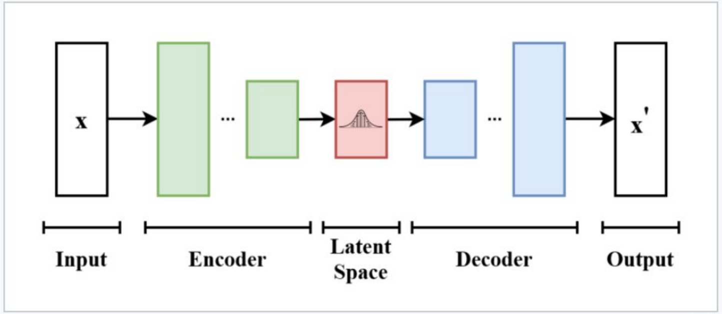
* Depending on its ImageNet image generation model size

Latent Space and Variational Autoencoder (VAE)

- **Latent Space: a lower-dimensional representational space**, but perceptually equivalent to data (e.g. pixel) space
- LDM utilizes a model architecture called Variational Autoencoder (VAE) [Arxiv'13]
  LDM performs diffusion process in latent space
  Convert data btw pixel space and latent space using VAE encoder and decoder



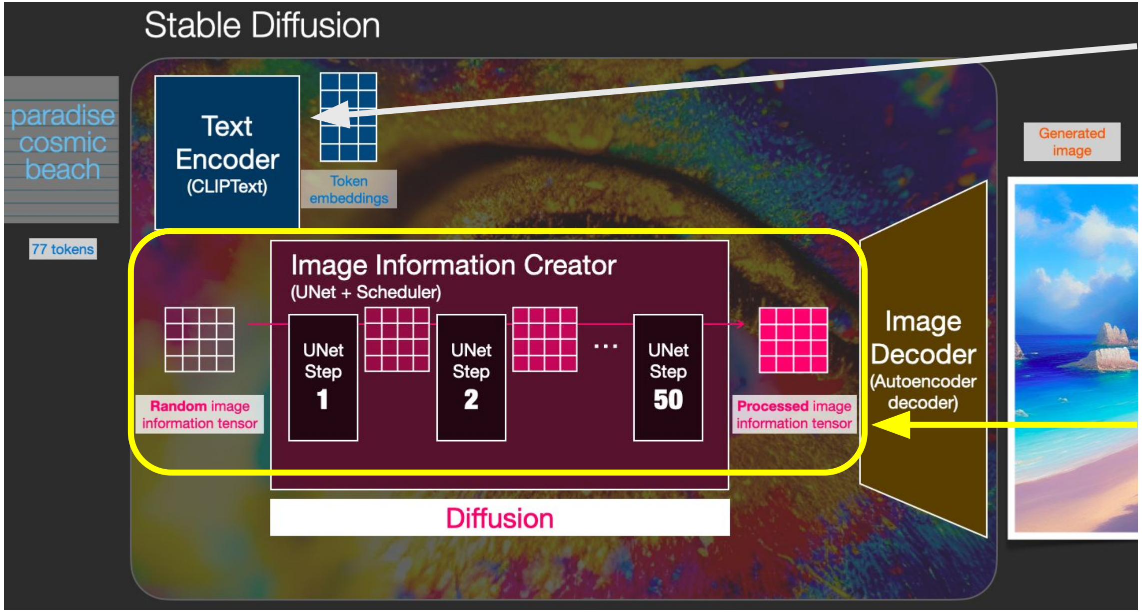
Latent data



The basic scheme of a variational autoencoder. The model receives x as input. The encoder compresses it into the latent space. The decoder receives as input the information sampled from the latent space and produces x' as similar as possible to x.

Latent Diffusion Model (LDM): Inference

- LDM performs diffusion process in latent space



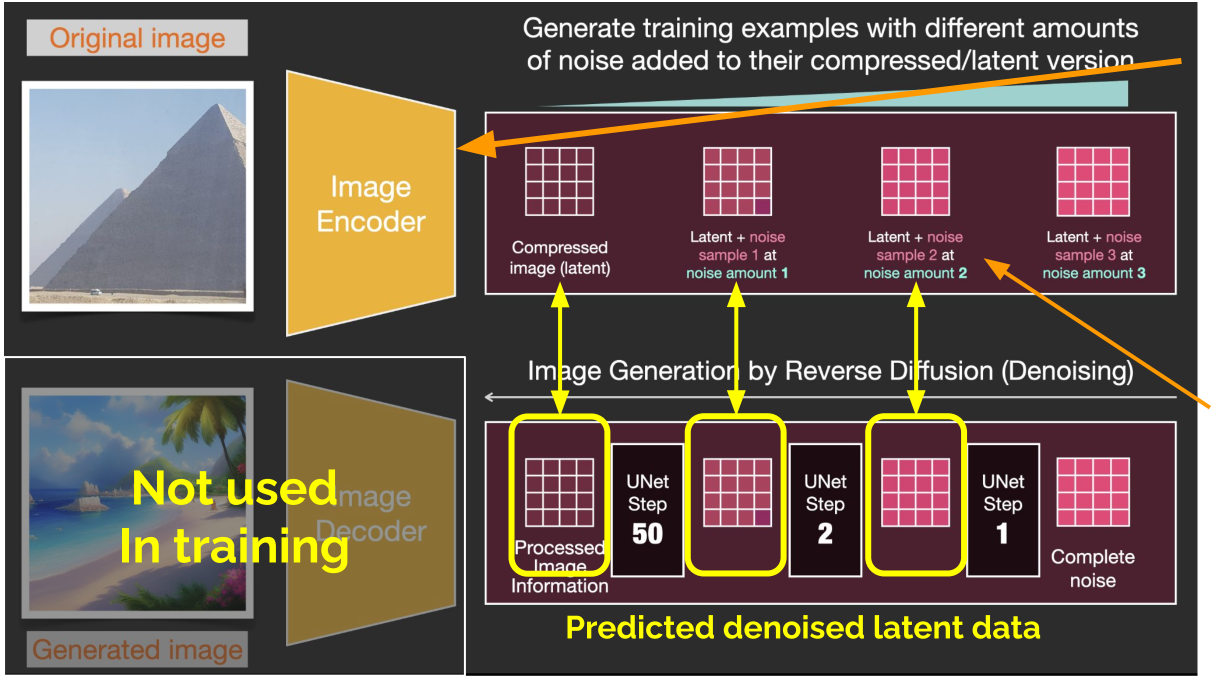
Text encoder is used to encode textual prompt.

In text-to-image generation, VAE image encoder is not used as generation starts from random latent noise

Diffusion process is done in latent space, not pixel space

Latent Diffusion Model (LDM): Training

- LDM performs diffusion process in latent space

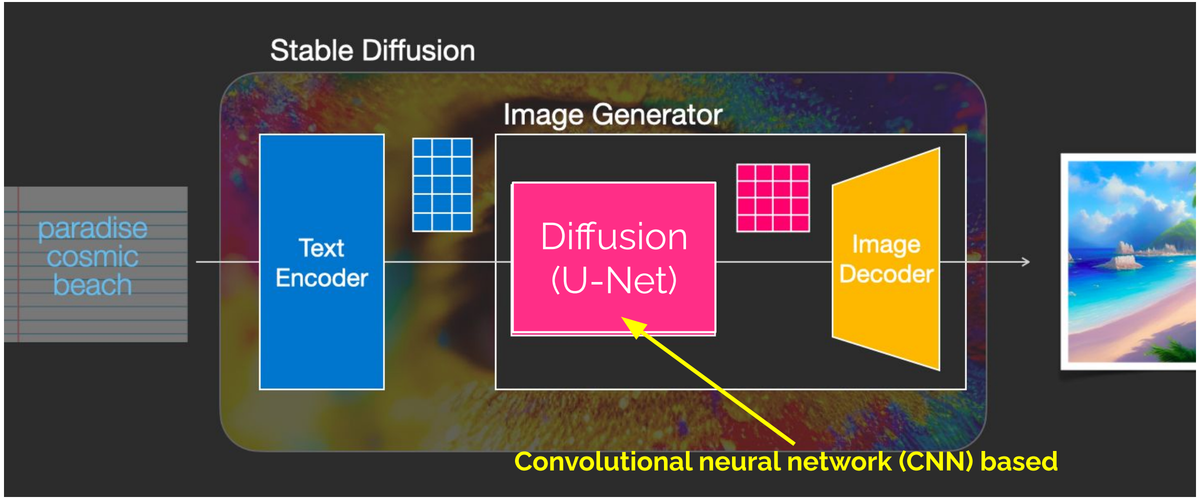


Use VAE image encoder to encode pixels into latent space
Then add noises to generate samples
All samples are in latent space

Model is trained to make predictions be close to **the data samples**

(Latent) Diffusion Transformers (DiT)

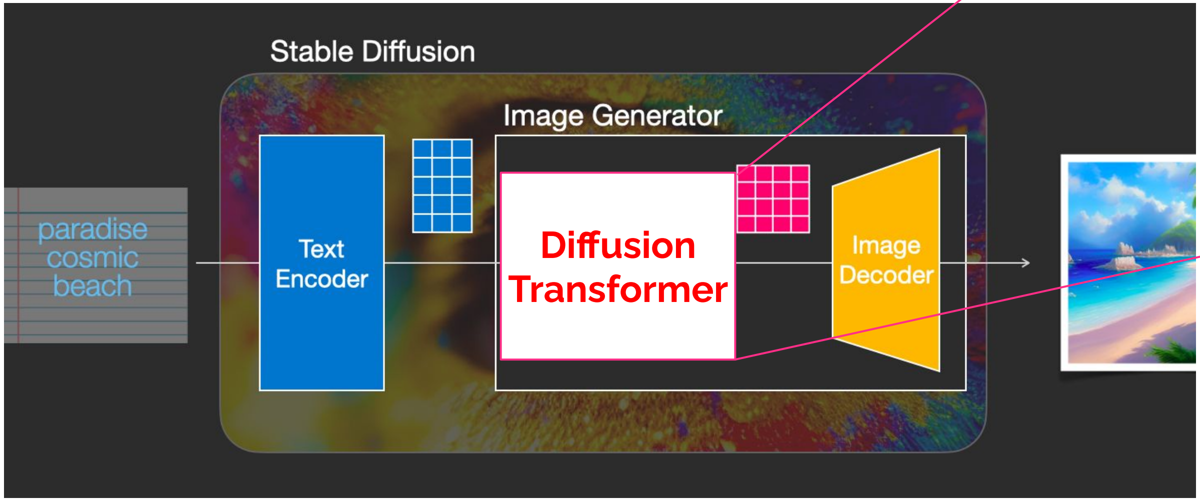
- Convolution-based U-Net lacks scalability

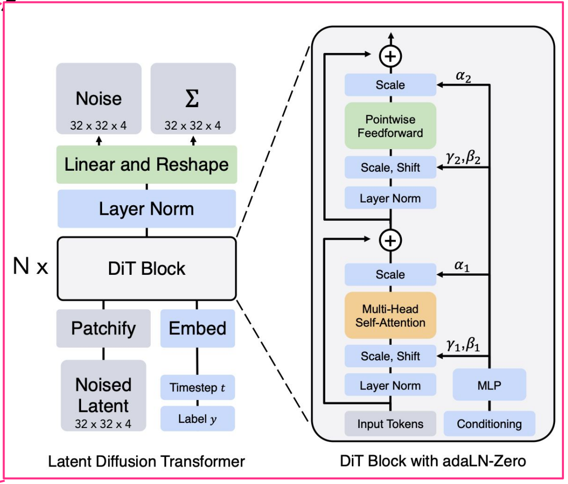


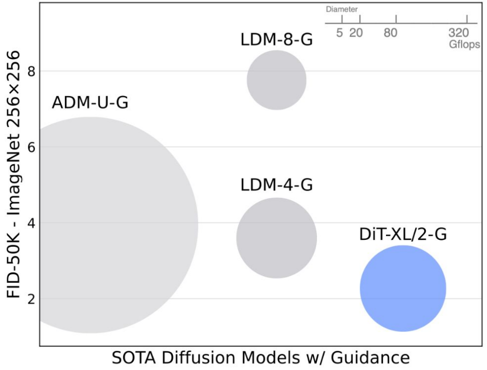
[U-Net: Convolutional Networks for Biomedical Image Segmentation [MICCAI'15]]
[Scalable Diffusion Models with Transformers. [CVPR23]]

(Latent) Diffusion Transformers (DiT)

- Higher Gflops, lower FID (better), better scalability





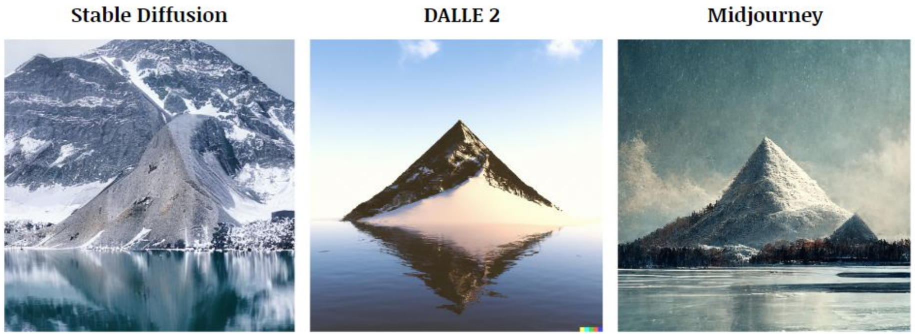


[U-Net: Convolutional Networks for Biomedical Image Segmentation [MICCAI'15]]
[Scalable Diffusion Models with Transformers. [CVPR23]]

Example of Diffusion Models

OpenAI DALL-E StabilityAI Stable Diffusion Midjourney

Dark alley at night 4k raining aesthetic



Pyramid shaped mountain above a still lake, covered with snow

https://www.marktechpost.com/2022/11/14/how-do-dall-e-2-stable-diffusion-and-midjourney-work/

Compute Characteristics of Diffusion Models

- Diffusion is really compute intensive although model size is small compared to LLM

Model	Params	GPU	Throughput
Stable Diffusion XL	2.6B	8xH100	**11.83 images/s**
Llama2 70b	70B		**15875.80 tokens/s**
Stable Diffusion XL	2.6B	1xGH200	**2.30 images/s**
Llama2 70b	70B		**4067.52 tokens/s**

* Used 20 inference steps per image generation.

Benchmark data from MLPerf Inference: https://mlcommons.org/benchmarks/inference-datacenter/

Stable Diffusion in HuggingFace

- HF provides a library called *diffusers* for diffusion models train/inference

Model from hub

https://huggingface.co/stabilityai/stable-diffusion-3-medium

Hugging Face | Search models, datasets, users...

stabilityai / **stable-diffusion-3-medium**

```
import torch
from diffusers import StableDiffusion3Pipeline

pipe = StableDiffusion3Pipeline.from_pretrained("stabilityai/stable-diffusion-3-medium")
prompt = "a photograph of an astronaut riding a horse"
image = pipe(prompt).images[0]
```

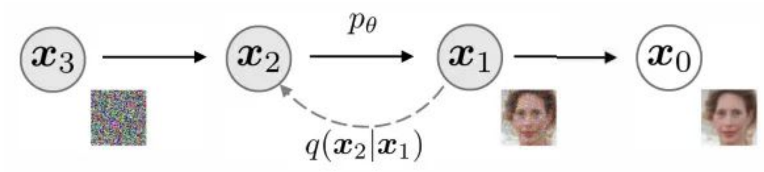
Library

Diffusers

Discussions

- Skipping iteration to accelerate generation with trade-off image quality
  Denoising Diffusion Implicit Models (DDIM) [ICLR'21]

DDPM



DDIM

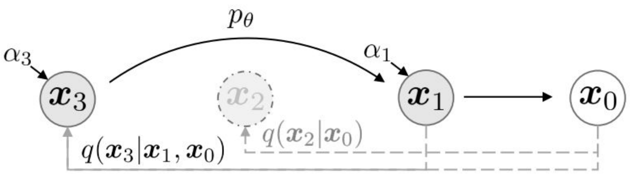


Figure 2: Graphical model for accelerated generation, where $\tau = [1, 3]$.

Discussions

- Parallelization of U-Net: DistriFusion

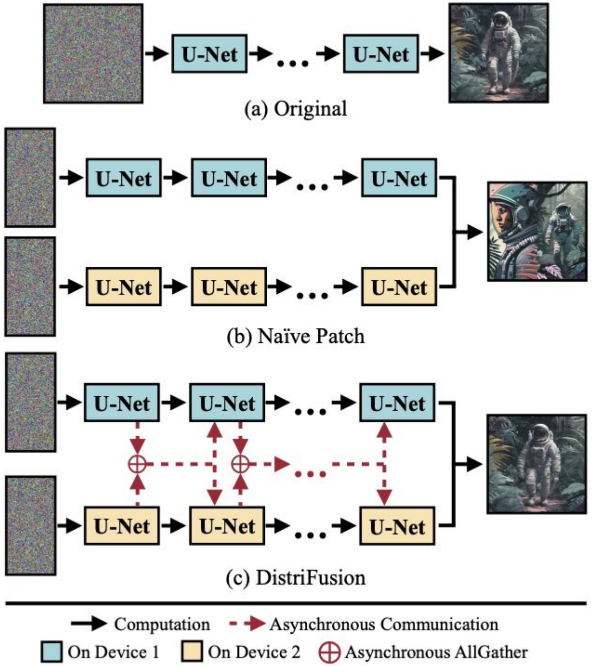
DistriFusion: Distributed Parallel Inference for High-Resolution Diffusion Models

Muyang Li[1*] Tianle Cai[2*] Jiaxin Cao[3] Qinsheng Zhang[4] Han Cai[1]
Junjie Bai[3] Yangqing Jia[3] Kai Li[2] Song Han[1,4]

[1]MIT [2]Princeton [3]Lepton AI [4]NVIDIA
https://github.com/mit-han-lab/distrifuser

Original	Naïve Patch (4 Devices)	Ours (4 Devices)
MACs: 907T	MACs Per Device: 190T (4.8× Less)	MACs Per Device: **227T** **(4.0× Less)**
Latency: 12.3s	Latency: 3.14s (3.9× Faster)	Latency: **4.16s** **(3.0× Faster)**

Prompt: *Ethereal fantasy concept art of an elf, magnificent, celestial, ethereal, painterly, epic, majestic, magical, fantasy art, cover art, dreamy.*



Discussions

- Parallelization of U-Net: PipeFusion

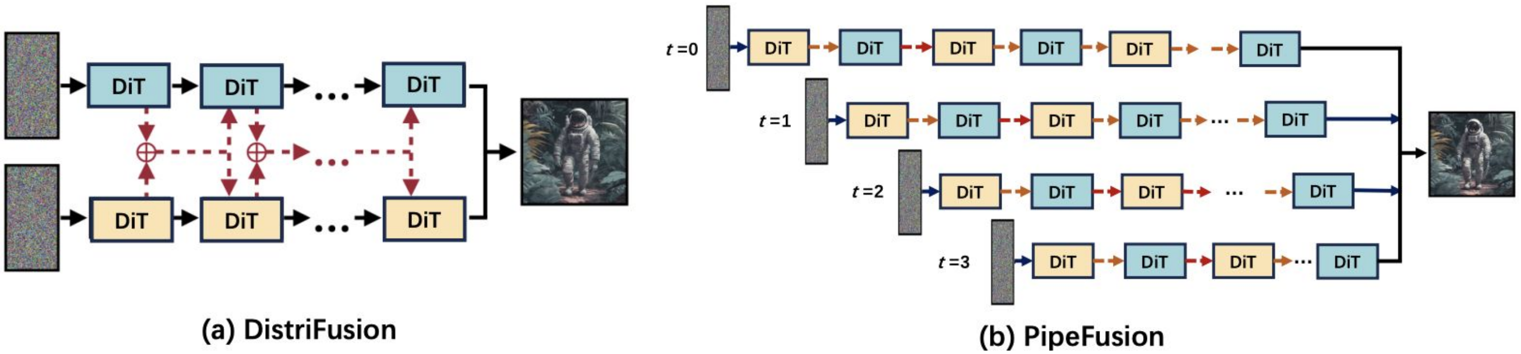


Figure 1: (a) DistriFusion replicates DiT parameters on two devices. It splits an image into 2 patches and employs asynchronous allgather for activations of every layer. (b) PipeFusion shards DiT parameters on two devices. It splits an image into 4 patches and employs asynchronous P2P for activations across two devices.