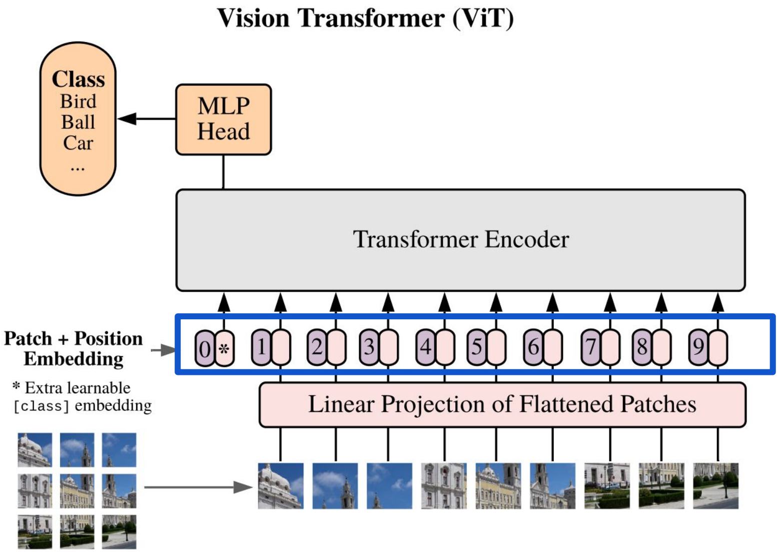
“Lecture” 12:
Brief takeaways from what we have seen in class so far

Announcements

- A1 grades have been released on gradescope
- A2 will be released today
- Discussion sessions begin next class, Mon Nov 4
    - Discussion instructions are on the class website and posted on Ed
    - Discussion assignments are posted on Ed, each person is assigned to present 1 paper and ask questions for 2 papers
    - Gradescope has assignments for “discussion slides” and “discussion questions”. You must submit your slides and questions (2 questions per assigned paper) the evening before the corresponding paper slot

First topic: Vision Transformers (ViT)

Key idea: Convert image into sequence of patches. Can then benefit from Transformer architecture and self-attention, which jointly attends over all patches



Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Converted image sequence ready for input to Transformer encoder (analogous to text sequence)

Dosovitsky et al. 2021

Key takeaways

- Architectural core of vision encoder in many of the modern models in this class
  - Although we saw some modeling improvements like the Swin Transformer (hierarchical ViT using shifted windows), vanilla ViT is still widely used
- Transformers have less inductive bias than CNNs (assumes less about spatial structure)
  - Consequence: Transformers work well when trained on very large amounts of data, less so when there are smaller / medium amounts of data (in this case, leveraging CNN's assumptions about data structure can be helpful)
- Weakness: ViT has quadratic computational complexity of self-attention, which can cause challenges e.g. scaling to high-resolution images, and can be more memory intensive

Key takeaways

- Some areas of ongoing research:
    - Improving computational efficiency (e.g., token pruning strategies from the last lecture)
    - Handling high-resolution and large-scale images, and multi-scale Transformers (Swin Transformer is an example of multi-scale)
    - Architectures that model more complex vision settings such as video, detection, segmentation, etc. (we saw DETR used in object detection, TimeSformer for video)
    - Improving positional encoding mechanisms (will see more in discussion sessions)

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

Generative Models

Train on huge amounts of data

Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

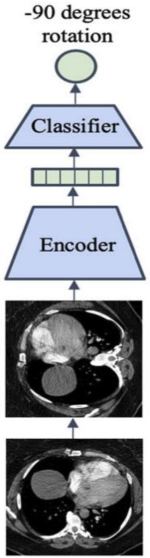
Generative Models

Train on huge amounts of data

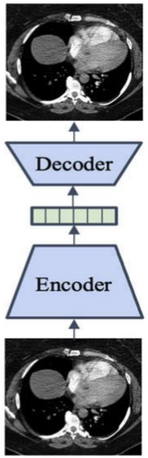
Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

Different representation learning paradigms



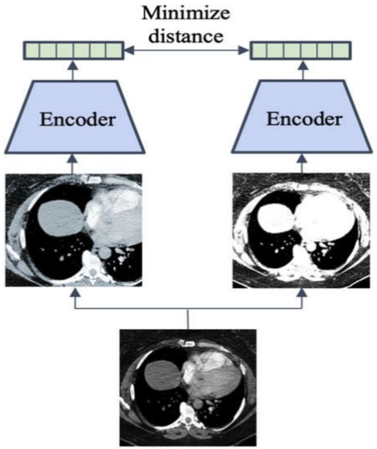
Innate relationship objective
E.g., predict rotation angle (or some other innate property) of an image



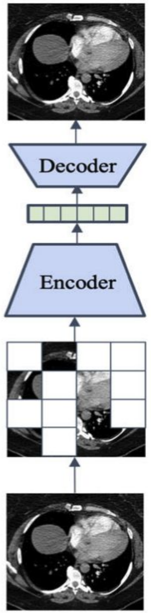
Generative objective
Compress and then reconstruct input image (e.g. autoencoders)

Figure credit: Huang et al. 2023.

Popular state-of-the-art approaches

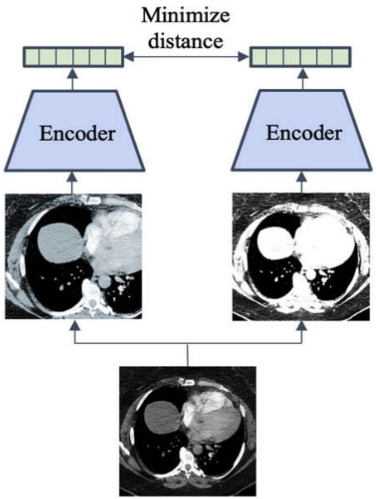


Contrastive objective
Different views of the same input should have more similar representation to each other than with a different input



Self-prediction objective
Mask parts of input data and predict these parts

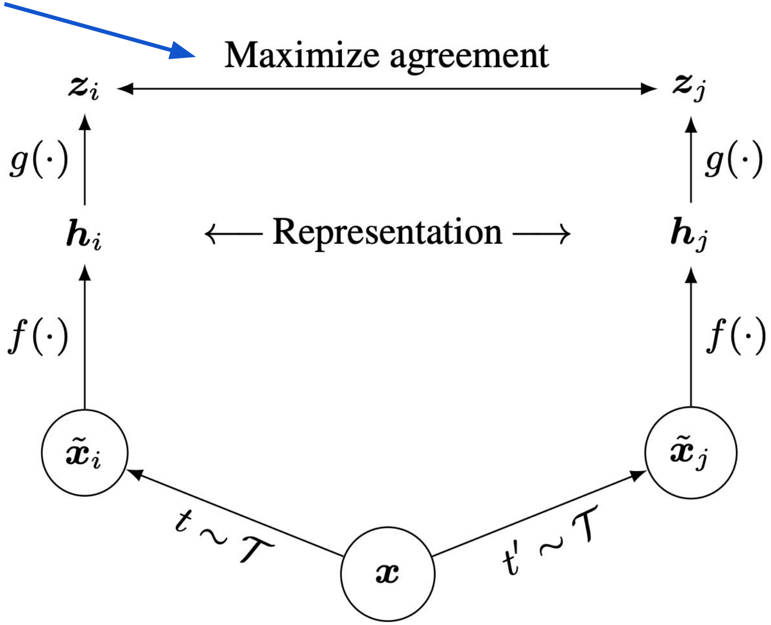
SimCLR: a foundational method for contrastive self-supervised learning



Contrastive objective
Different views of the same input should have more similar representation to each other than with a different input

Contrastive loss

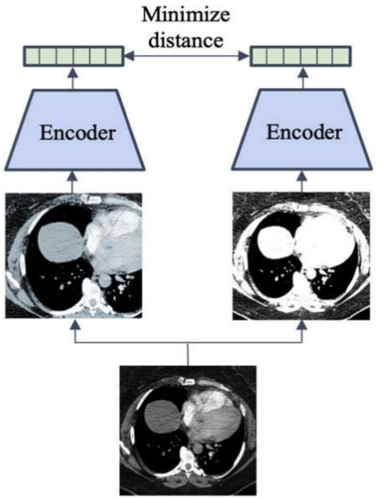
SimCLR formulation



Input

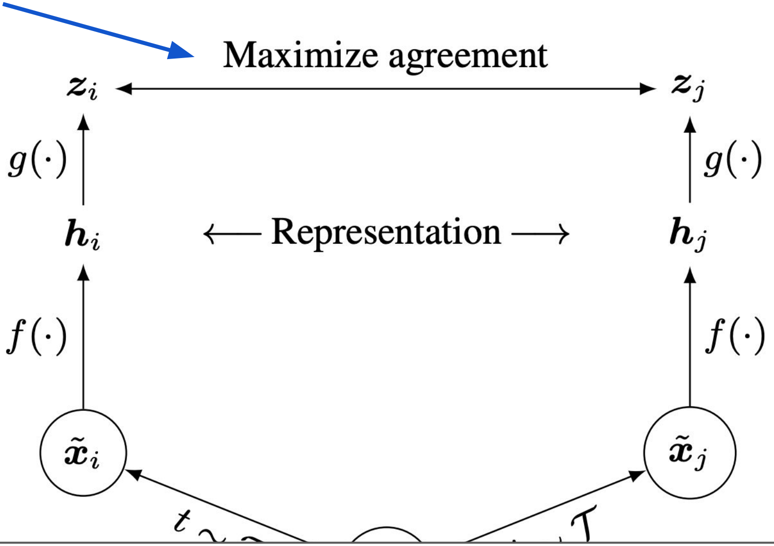
Chen et al. 2020

SimCLR: a foundational method for contrastive self-supervised learning



Contrastive loss

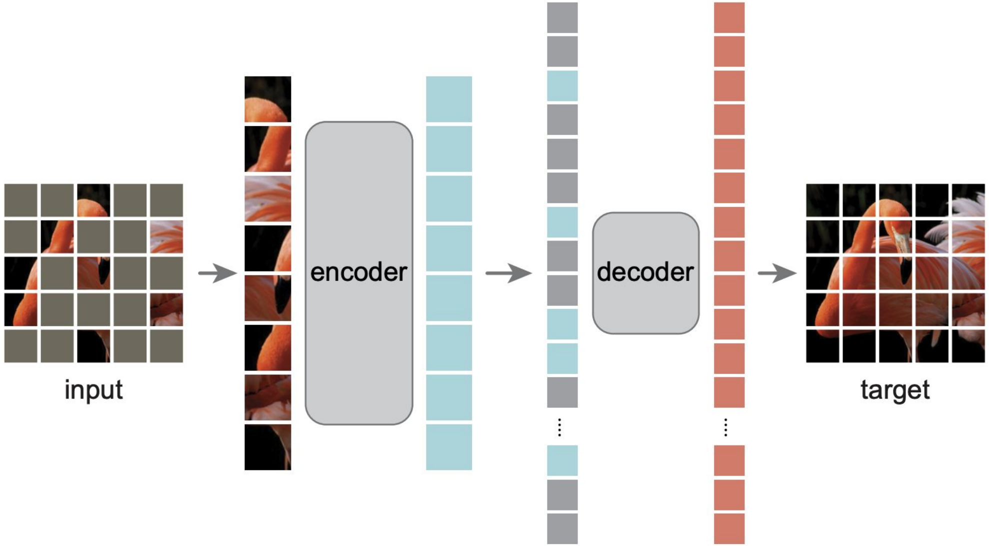
SimCLR formulation



After self-supervised training, can fine-tune the encoder *f* on smaller labeled datasets. Can also directly extract learned representations h for downstream tasks.

Masked Autoencoders (MAE)

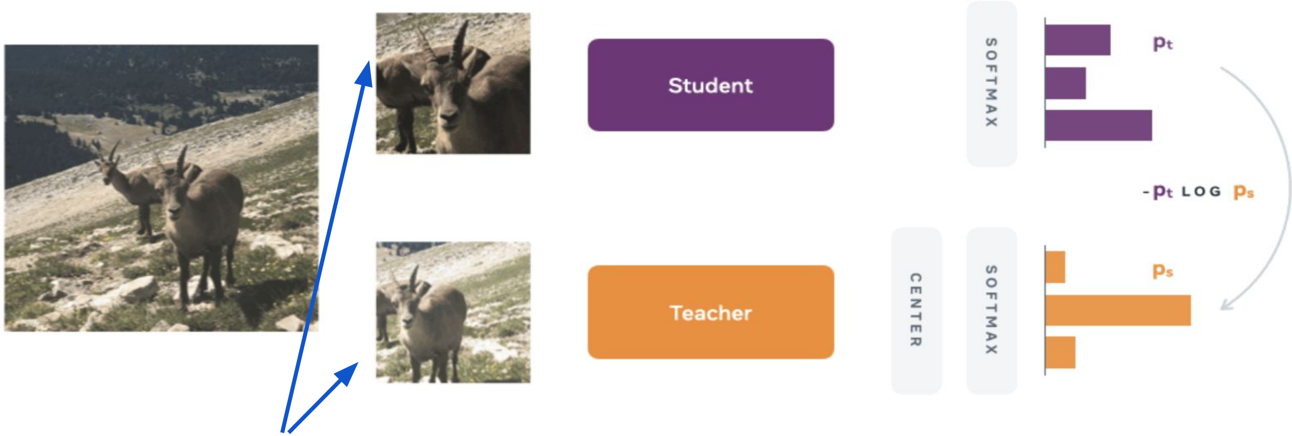
- Key idea: mask substantial parts of the input, train the model to reconstruct (predict) these parts
- Inspired by major self-supervised representation learning paradigm in NLP (e.g. BERT), that masks tokens in sentences and trains models to reconstruct them



He et al. 2021

Another SSL approach: DINO (Self-**Di**stillation with **No** Labels)

Not an example of a contrastive learning objective for self-supervised learning, but related! Builds on the notion of matching representations of augmented views of the same image, but no longer uses negative samples. Instead, proposes a **teacher-student framework**.

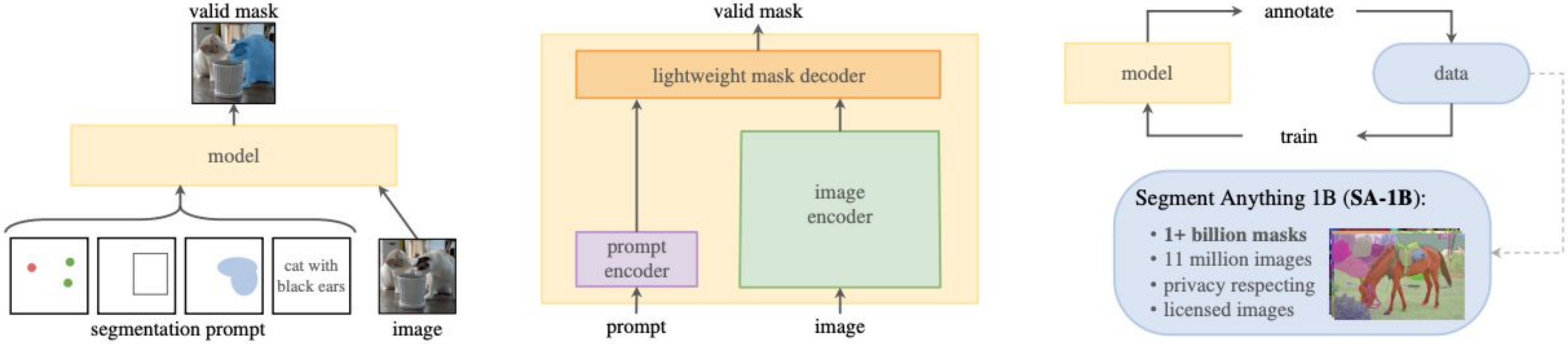


Augmented "views" of the input are passed to both the student and teacher networks.
Student views: more diverse and aggressive augmentations, to learn representations that generalize across augmentations
Teacher views: less aggressive augmentations (including larger crops), to provide a more stable target for the student to follow.

Caron et al. 2021
Oquab et al. 2024

Segment Anything Model (SAM): A foundation model targeted for segmentation

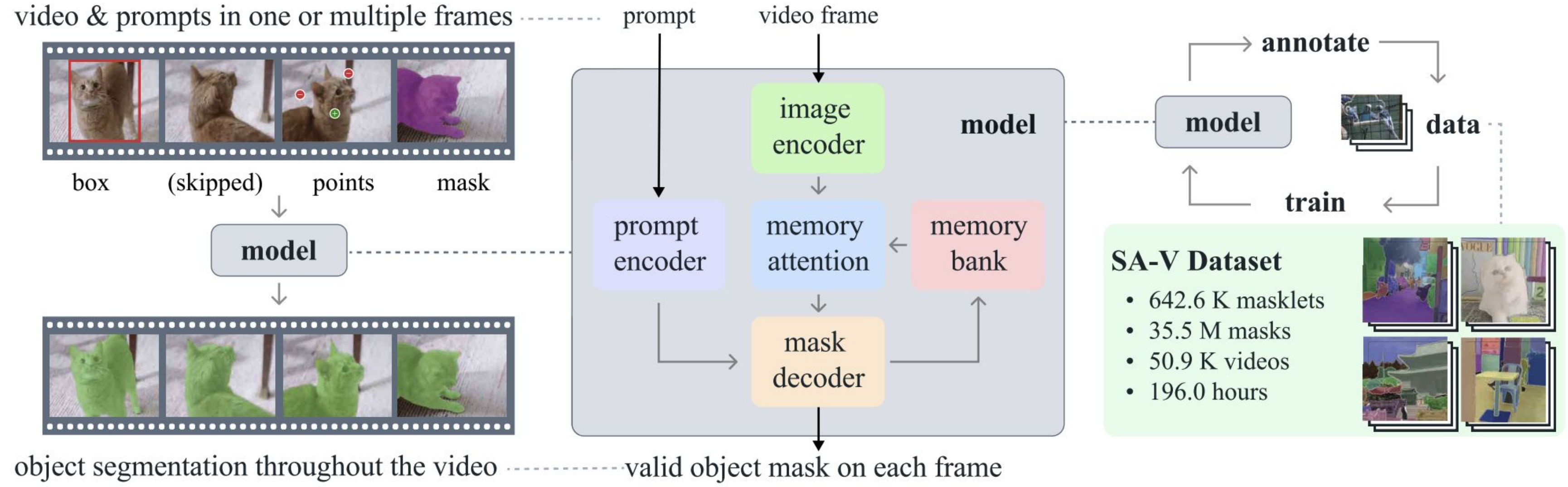
- Foundation model for promptable segmentation, based on a Transformer encoder-decoder architecture. Generalizes to many segmentation tasks.
- Not all representation learning needs to be through self-supervised learning. Here, a supervised paradigm that also achieves powerful representation learning.
- Trained on 1 billion masks from 11 million images, using the model in a data collection loop.



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

Kirrilov et al. 2023.

Segment Anything Model 2 (SAM 2): Extending to videos



(a) Task: promptable visual segmentation **(b) Model:** Segment Anything Model 2 **(c) Data:** data engine and dataset

Ravi et al. 2024

Key takeaways

- SimCLR, MoCo (also contrastive loss but alleviates large batch size requirement of SimCLR), MAE and their variants have been widely used for vision-only representation learning, are good to know
- Can be difficult to compare models – there can be subtle differences, some may work better in certain scenarios (e.g., linear probing vs fine-tuning a few last layers)
- DINOv2 pre-trained on 142 million images has emerged as a very strong representation learner. Recommended to try when looking for a strong neural network backbone for image tasks.
- For segmentation specifically, SAM and SAM2 are targeted to this task and very powerful, recommended to try. However, may need to combine with other types of models to get desired type of output (e.g. use object detector, then refine mask in bounding box with SAM), since it is a designed for promptable segmentation instead of classical setup of data with training labels.
- These takeaways reflected in biomedical AI literature: wide usage but unclear comparisons among SimCLR, MoCo, MAE etc., promising recent uses of DINOv2 and SAM (+ MedSAM) based methods.

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

Generative Models

Train on huge amounts of data

Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

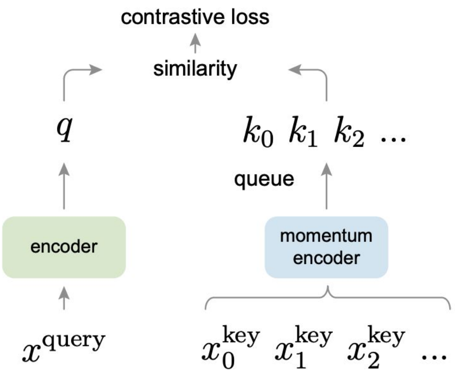
Generative Models

Train on huge amounts of data

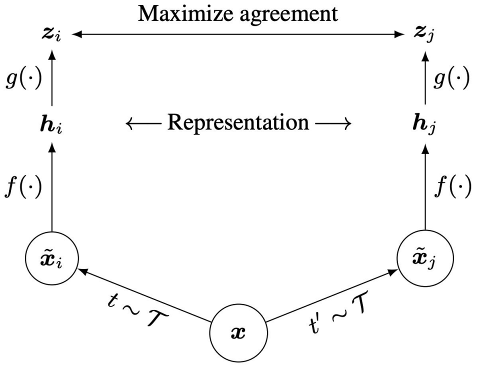
Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

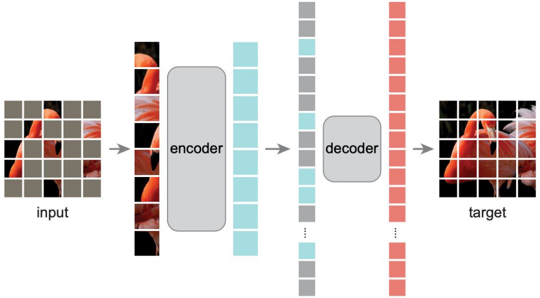
Self-Supervised Learning



MoCo (He et al. 2019)



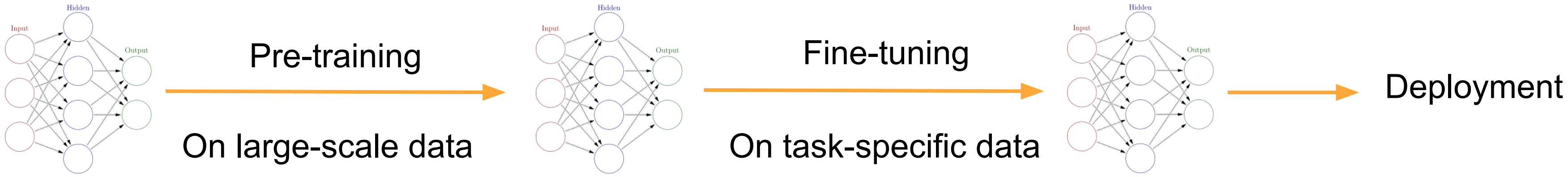
SimCLR (Chen et al. 2020)



MAE (He et al. 2021)

learning to extract powerful representations without human annotated labels

Adapt to Downstream Task



Pre-training

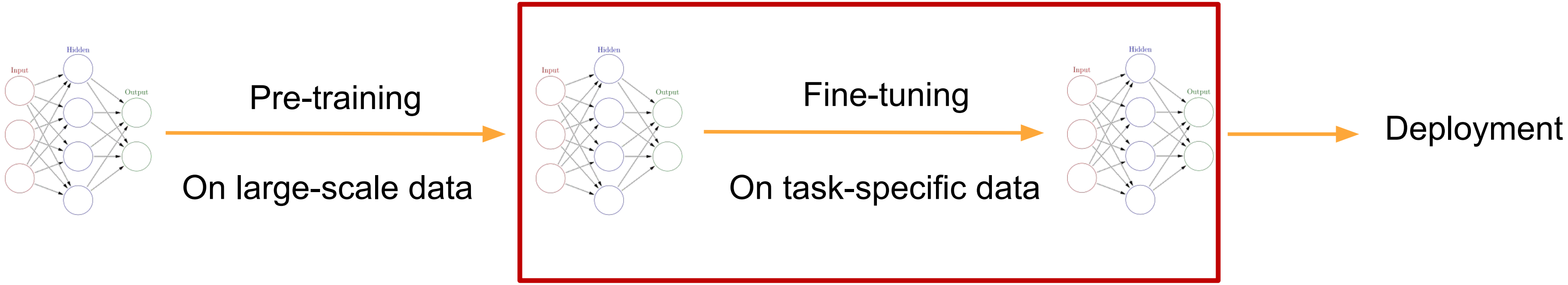
On large-scale data

Fine-tuning

On task-specific data

Deployment

Adapt to Downstream Task



Pre-training

On large-scale data

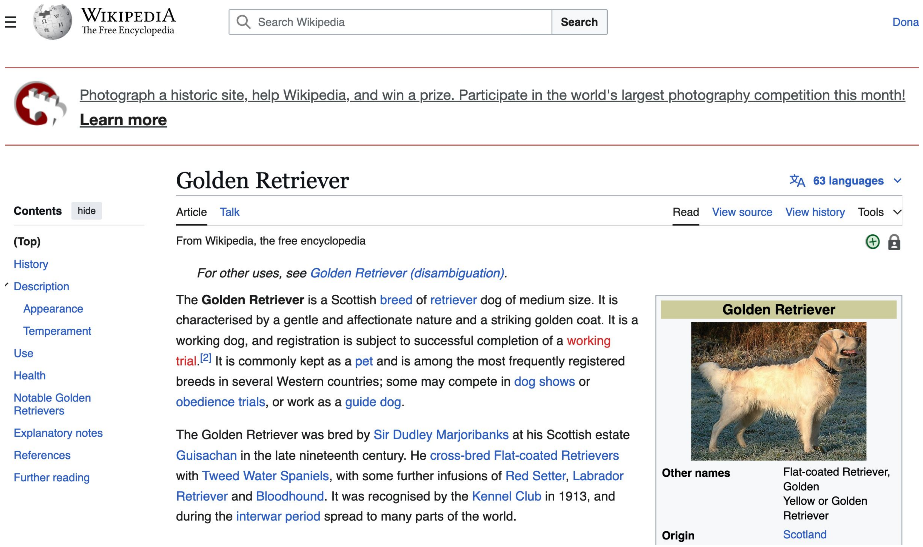
Fine-tuning

On task-specific data

Deployment

How to skip this step to enable zero-shot generalization?

Natural Language Supervision



Photograph a historic site, help Wikipedia, and win a prize. Participate in the world's largest photography competition this month! **Learn more**

Golden Retriever

From Wikipedia, the free encyclopedia

For other uses, see Golden Retriever (disambiguation).

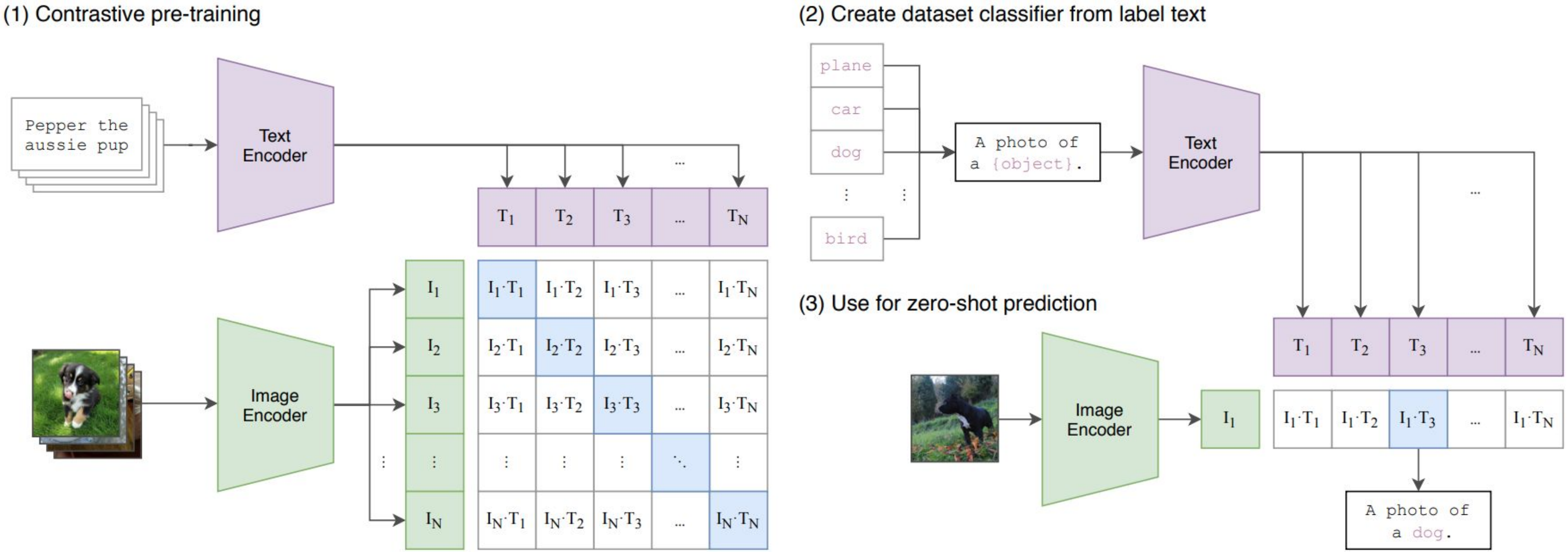
The **Golden Retriever** is a Scottish breed of retriever dog of medium size. It is characterised by a gentle and affectionate nature and a striking golden coat. It is a working dog, and registration is subject to successful completion of a working trial.[2] It is commonly kept as a pet and is among the most frequently registered breeds in several Western countries; some may compete in dog shows or obedience trials, or work as a guide dog.

The Golden Retriever was bred by Sir Dudley Marjoribanks at his Scottish estate Guisachan in the late nineteenth century. He cross-bred Flat-coated Retrievers with Tweed Water Spaniels, with some further infusions of Red Setter, Labrador Retriever and Bloodhound. It was recognised by the Kennel Club in 1913, and during the interwar period spread to many parts of the world.

Golden Retriever	
Other names	Flat-coated Retriever, Golden Yellow or Golden Retriever
Origin	Scotland

- ❖ Unlimited Data: vast amount of text-image pairs on the internet
- ❖ Flexible zero-shot transfer: learn a representation that connects visual content to language

CLIP learns a joint representation space for images and text


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
T_1 T_2 T_3 ... T_N
Image Encoder
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
⋮ ⋮ ⋮ ⋱ ⋮
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
T_1 T_2 T_3 ... T_N
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

Radford et al. 2021.

Key takeaways

- Why CLIP achieves such success
  - Transferability: Bridging Vision and Language
  - Scalability: Large-Scale Pretraining on 400M+ Data
  - Simplicity: Contrastive Learning Enables the Large-Scale Pre-Training

- Despite efforts to improve CLIP, the original formulation is still dominant. It’s hard to improve!
  - Consider using CLIP, OpenCLIP, or SigLIP (current best model)

- Ongoing areas of research:
      - Learning to prompt
      - Used in frameworks for different tasks like ZegCLIP (zero-shot semantic segmentation), DALL-E (image generation)
      - Different settings: video CLIP4Clip, region-based CLIP

- A plethora of works directly applying CLIP to zero-shot prediction in biomedical applications: ConVIRT (CLIP precursor), CheXzero, PLIP, QuiltNet, MONET, BiomedCLIP, etc.

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

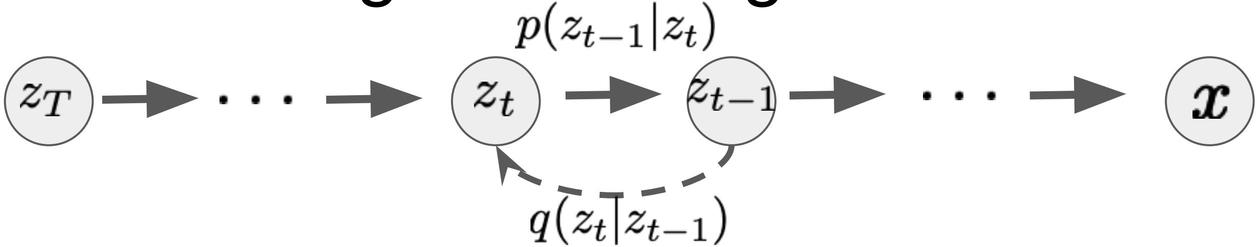
Generative Models

Train on huge amounts of data

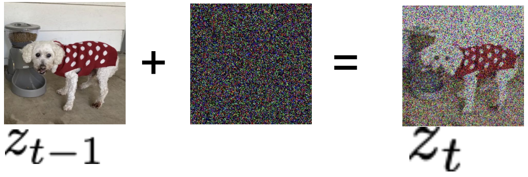
Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

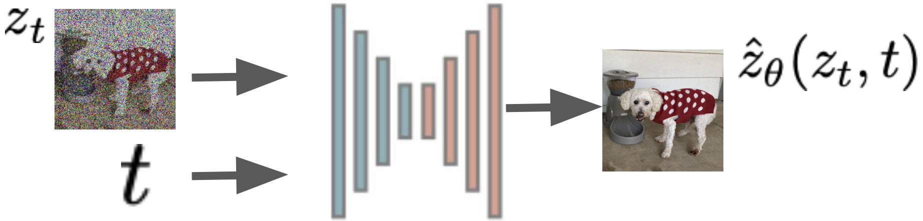
Diffusion training as learning to reverse noising



Sample t,
Apply noise: $q(z_t|z_{t-1})$



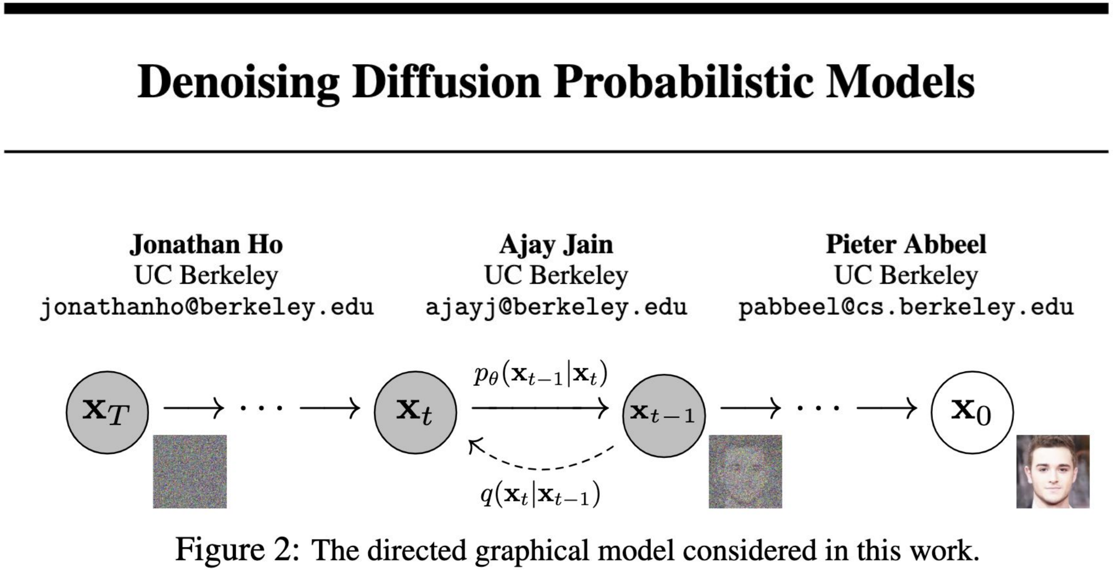
Use Unet to reverse the noise



Reconstruction loss

$$\mathbb{E}_{t\sim U(2,T)}\left[c(\alpha_t)\|\hat{z}_\theta(z_t,t)-z_{t-1}\|\right]$$

DDPM: the original pixel-level diffusion model



Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

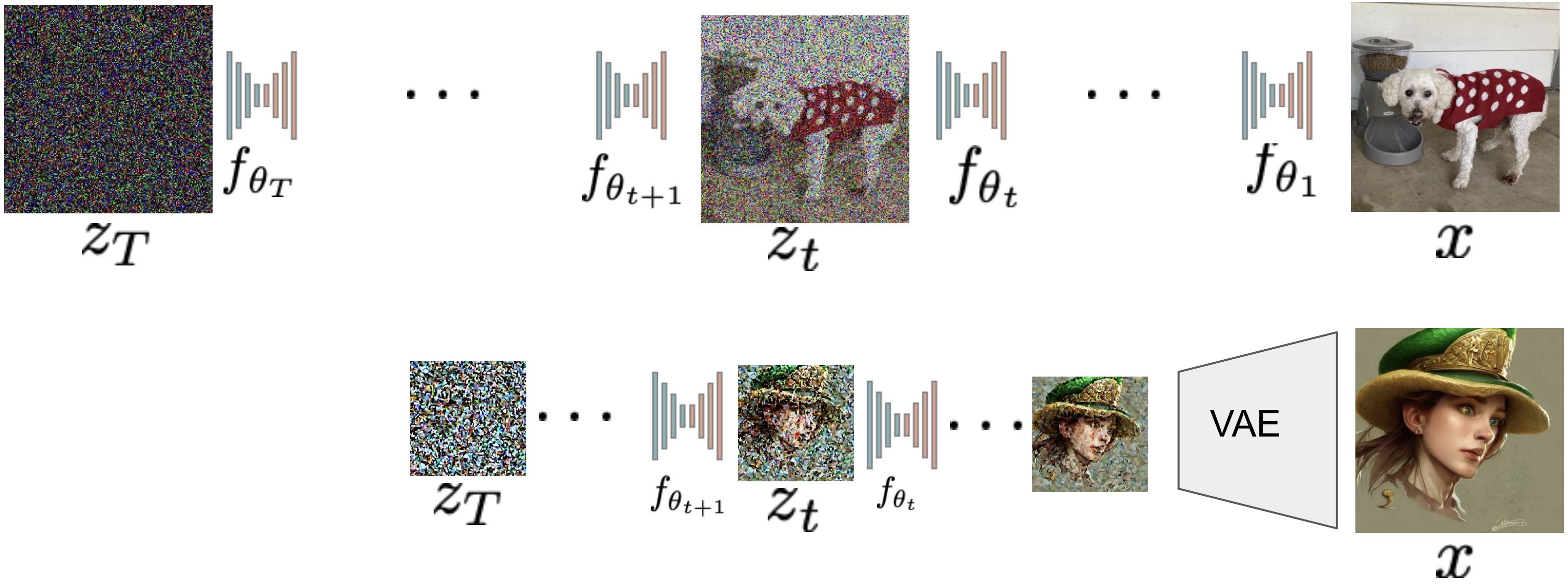
Figure 2: The directed graphical model considered in this work.

Stable Diffusion or Latent Variable Models

Idea: instead of diffusion models on *pixels*, do diffusion in a *compressed latent space*

Diffusion (DDPM)

Latent diffusion (Stable Diffusion)

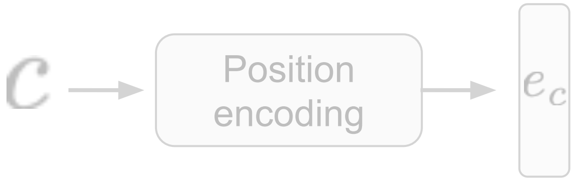


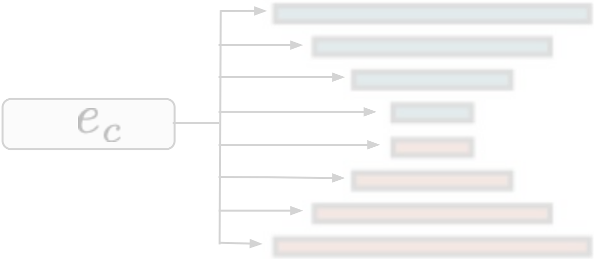
"High-Resolution Image Synthesis with Latent Diffusion Models"
"Score-based Generative Modeling in Latent Space"

Implementing conditional sampling

Class-based conditioning

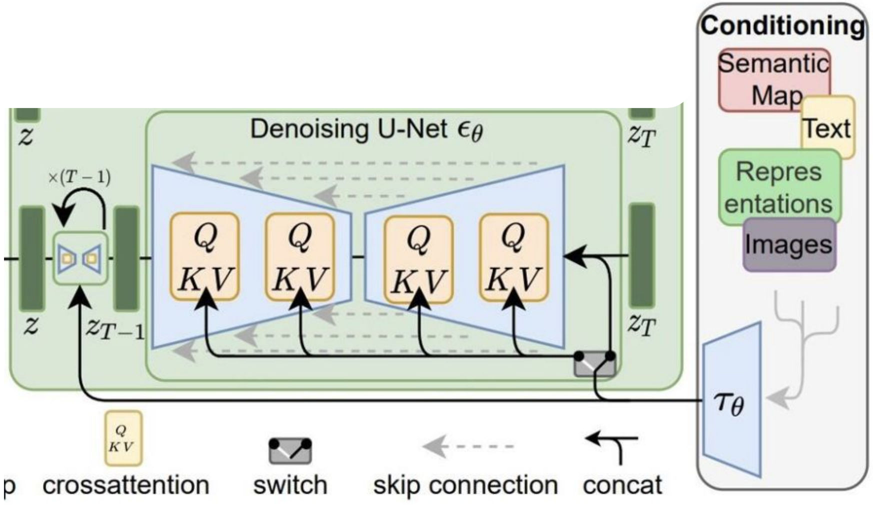
Encode the same way `t` is encoded





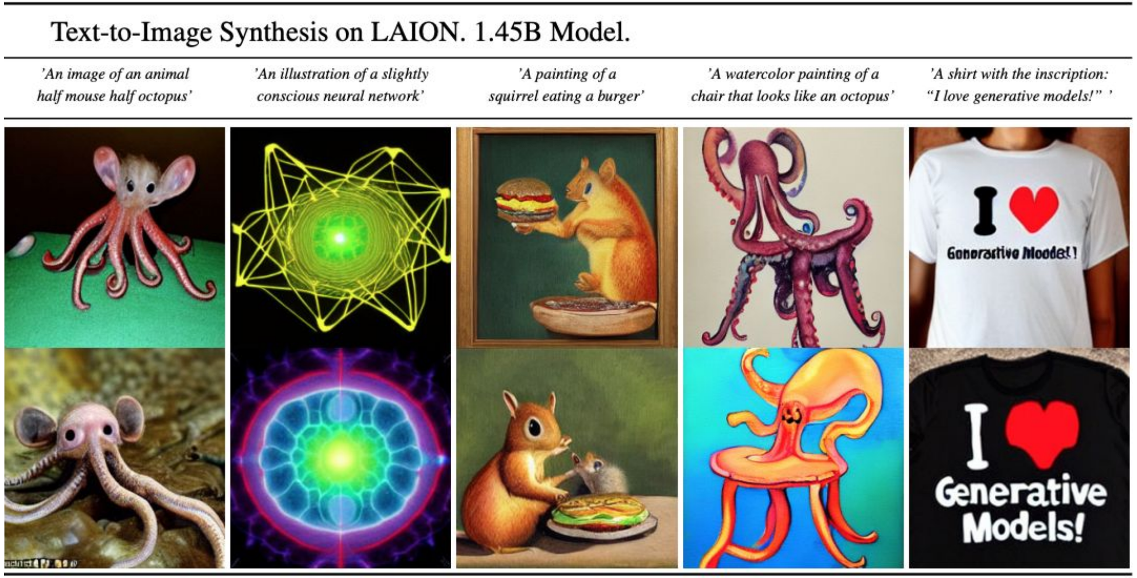
General conditioning via cross-attention

Much more common



Stable Diffusion: example of text -> image generative model (+ other types of conditioning!)

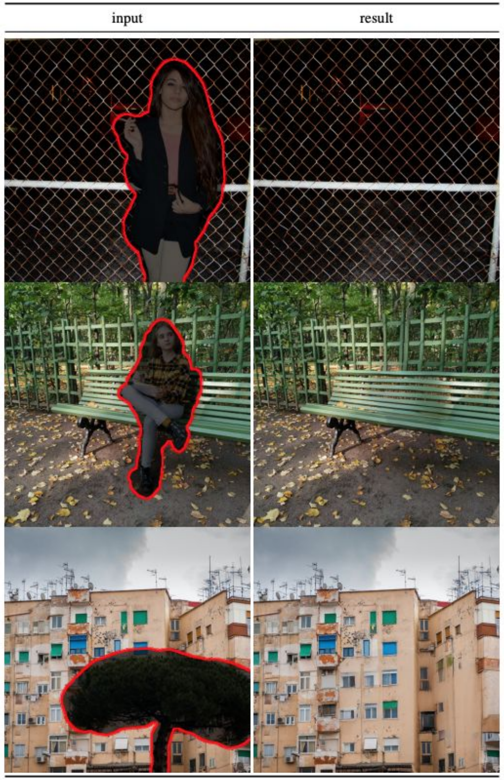
Text -> image model trained on LAION: open-source dataset of 400M image-text pairs



Text-to-Image Synthesis on LAION. 1.45B Model.

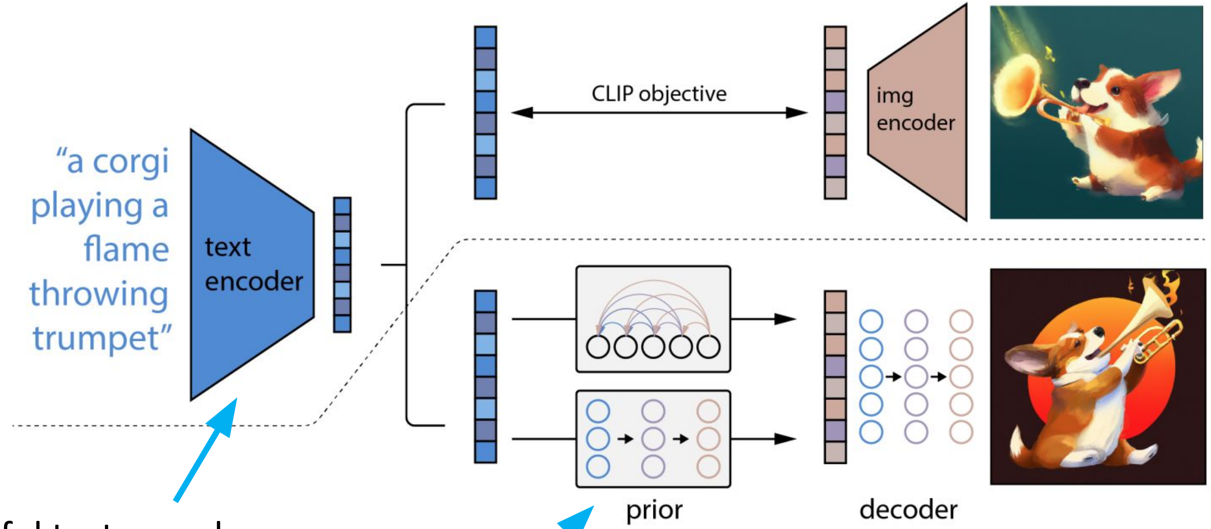
'An image of an animal half mouse half octopus'	'An illustration of a slightly conscious neural network'	'A painting of a squirrel eating a burger'	'A watercolor painting of a chair that looks like an octopus'	'A shirt with the inscription: "I love generative models!" '

Condition on provided input images to remove objects



Rombach et al. 2022.

DALL-E: related family of OpenAI models that focus on high quality text-conditioned image generation



Powerful text encoder trained on large quantities of image-caption pairs

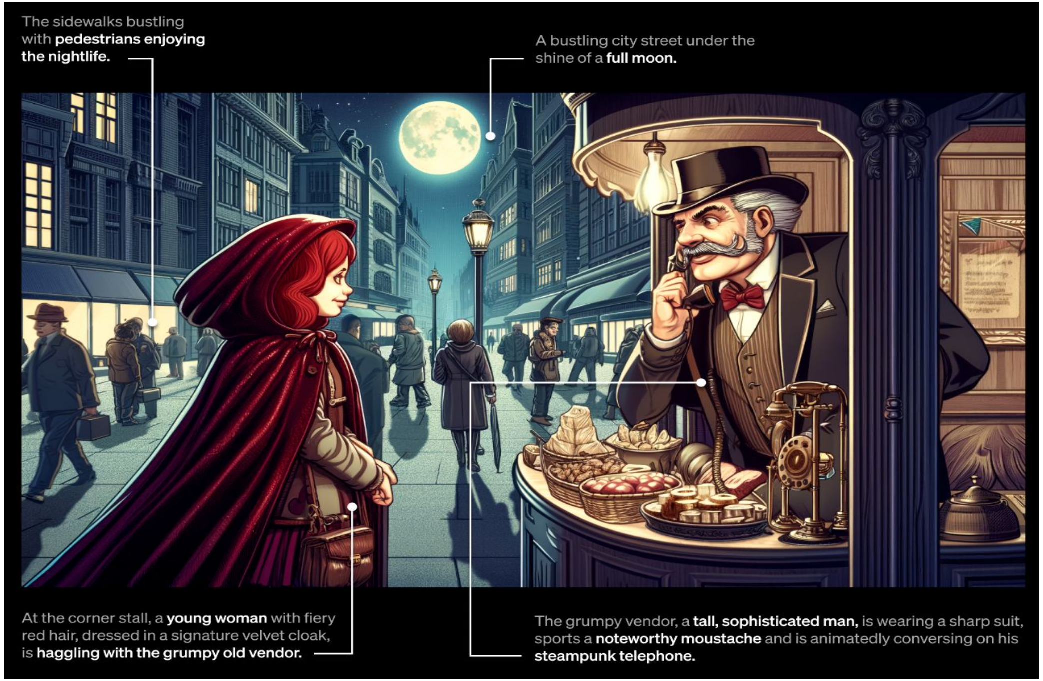
Learn to map text embedding to image embedding

Powerful diffusion-based image decoder

Ramesh et al. 2022.

DALL-E: related family of OpenAI models that focus on high quality text-conditioned image generation

DALL-E 3 is trained on large amounts of detailed, synthetically generated captions to enable high-fidelity and high-detail generation



The sidewalks bustling with **pedestrians enjoying the nightlife.**

A bustling city street under the shine of a **full moon.**

At the corner stall, a **young woman** with fiery red hair, dressed in a signature velvet cloak, is **haggling with the grumpy old vendor.**

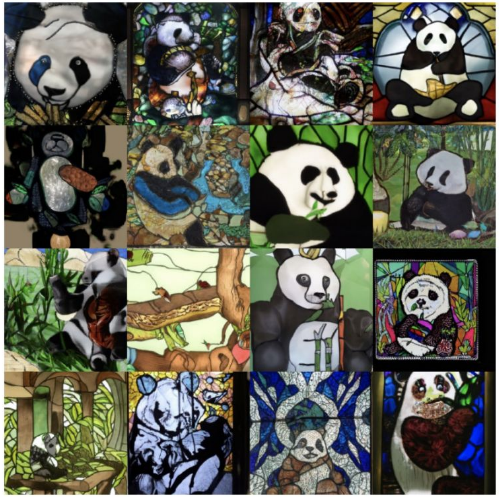
The grumpy vendor, a **tall, sophisticated man,** is wearing a sharp suit, sports a **noteworthy moustache** and is animatedly conversing on his **steampunk telephone.**

https://openai.com/dall-e-3

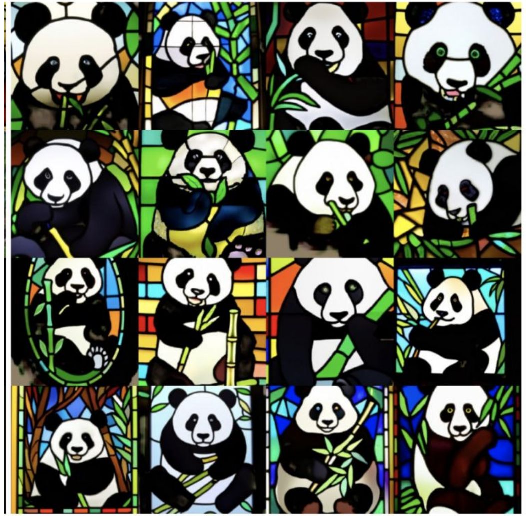
Conditional sampling in practice with “guidance”

The γ parameter lets us trade off diversity and fidelity

c=“*A stain glass window of a panda eating bamboo*”



Better diversity - smaller γ



Better fidelity - bigger γ

Good blog post: Guidance: a cheat code for diffusion models

Key takeaways

- Fundamental models to know are DDPM (original pixel-level diffusion model) and Stable Diffusion (diffusion in a compressed latent space, standard today)
- Common conditional diffusion-based generative models today include Stable Diffusion (text + other types of conditioning, open source) and DALL-E family (focus on high-quality text->image generation)
  - Connection to previous topic: CLIP is commonly used for the text encoder
- Evaluation can be challenging since there are various desiderata of generated images. Examples of evaluation metrics include Frechet Inception Distance (FID), precision-recall, CLIP score
- Ongoing areas of research: faster sampling, other forms of evaluation (e.g., compositional generalization), usage for tasks besides text->image generation (e.g., super-resolution, image inpainting, textual inversion, controlnet). Popular theme: better control of image generation.
- Major model class in AI broadly (e.g., art and commercial applications), but relatively less so far in biomedicine. Some work on synthetic data generation for training augmentation (Roentgen for CXR), MRI denoising, etc. Also discussed AlphaFold 3 as an example of non-image diffusion model

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

Generative Models

Train on huge amounts of data

Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

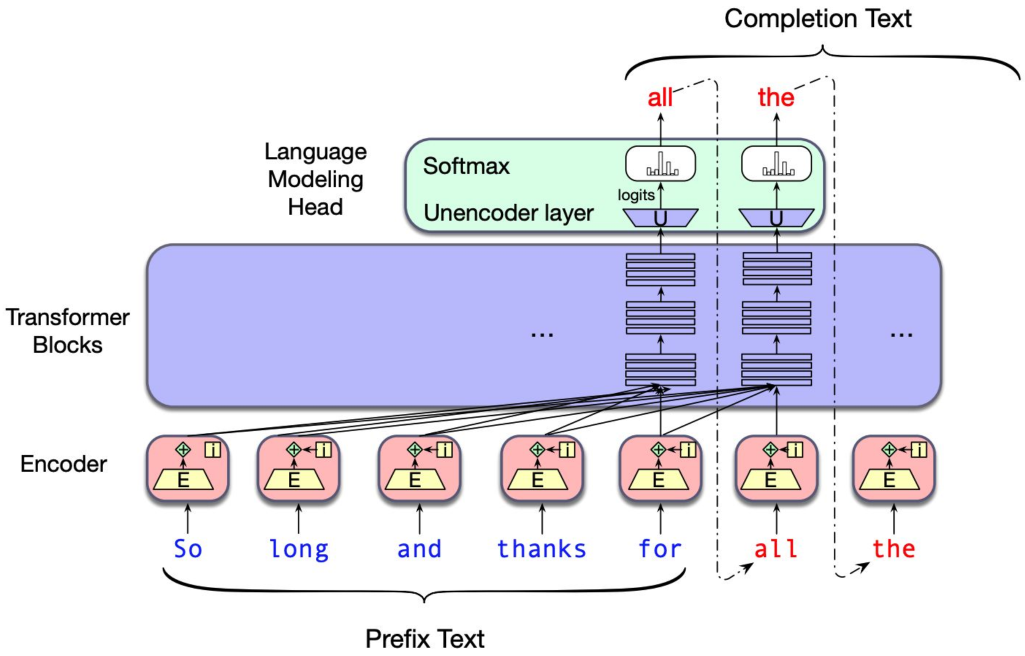
Generative Models

Train on huge amounts of data

Objective: learn to generate images and/or text

- Text -> Image (e.g. StableDiffusion, DALL-E)
- Text, Image -> Text (e.g. Flamingo, GPT4-V)

Autoregressive Language Models



LARGE LANGUAGE MODELS WITH TRANSFORMERS (Daniel Jurafsky & James H. Martin 2024)

How to Link Vision to LLMs?

Internal Linkage → Vision-Language Models

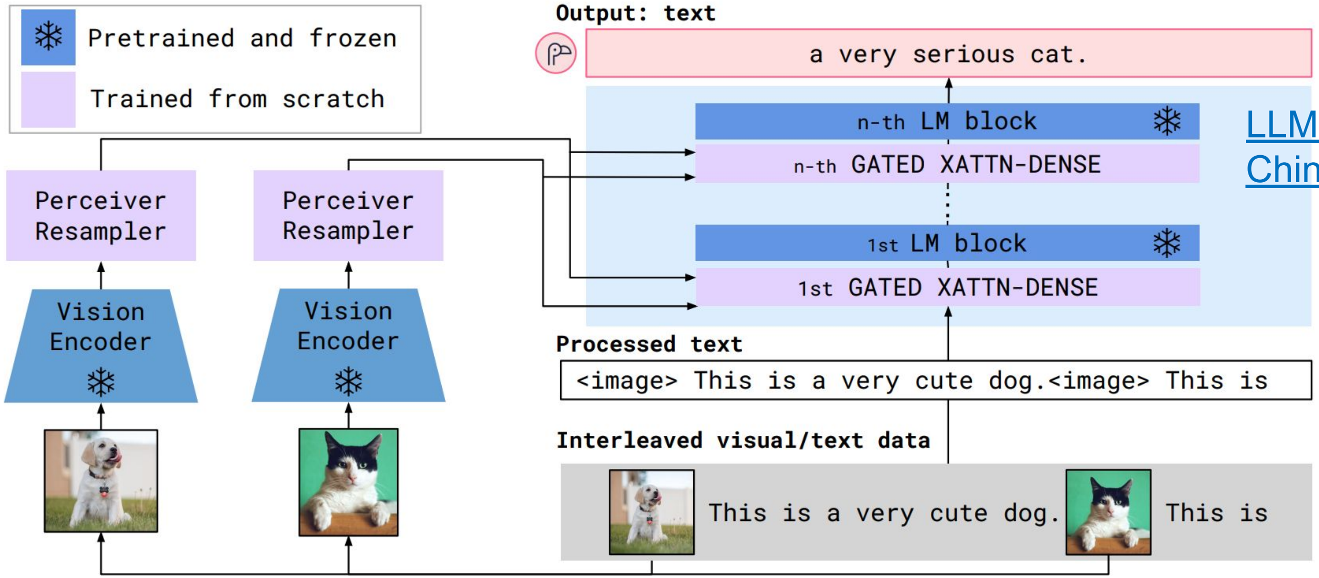
- ❖ Architecture
  - ○ Integrate Visual Features into Intermediate Layers
  - ○ Integrate Visual Features into Input Layer
  - ○ Integrate Visual Patches into Input Layer

External Linkage → Vision-Language Agents

Integrate Visual Features into Intermediate Layers

Perceiver Resampler: from varying-size large feature maps to few visual tokens

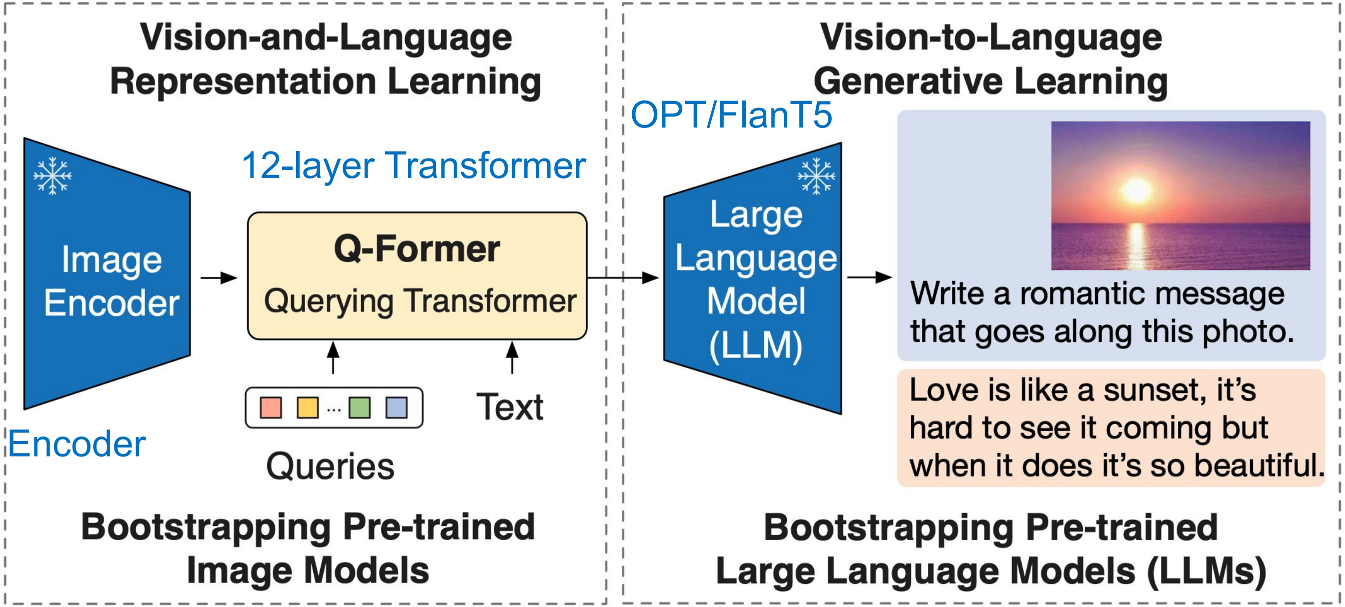
Vision Encoder: from pixels to features, CLIP-style Models



LLM: Chinchilla

Flamingo (Alayrac et al. 2022)

Integrate Visual Features into Input Layer



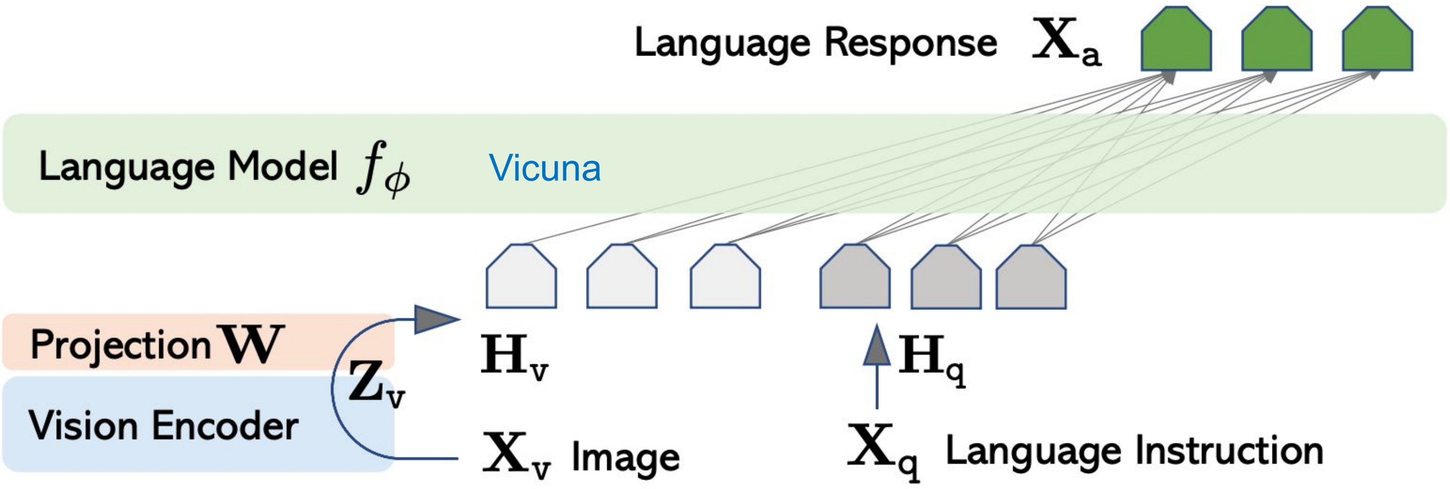
12-layer Transformer

OPT/FlanT5

CLIP Vision Encoder
ViT-L/14

BLIP-2 (Li et al. 2023)

Integrate Visual Features into Input Layer



Vicuna

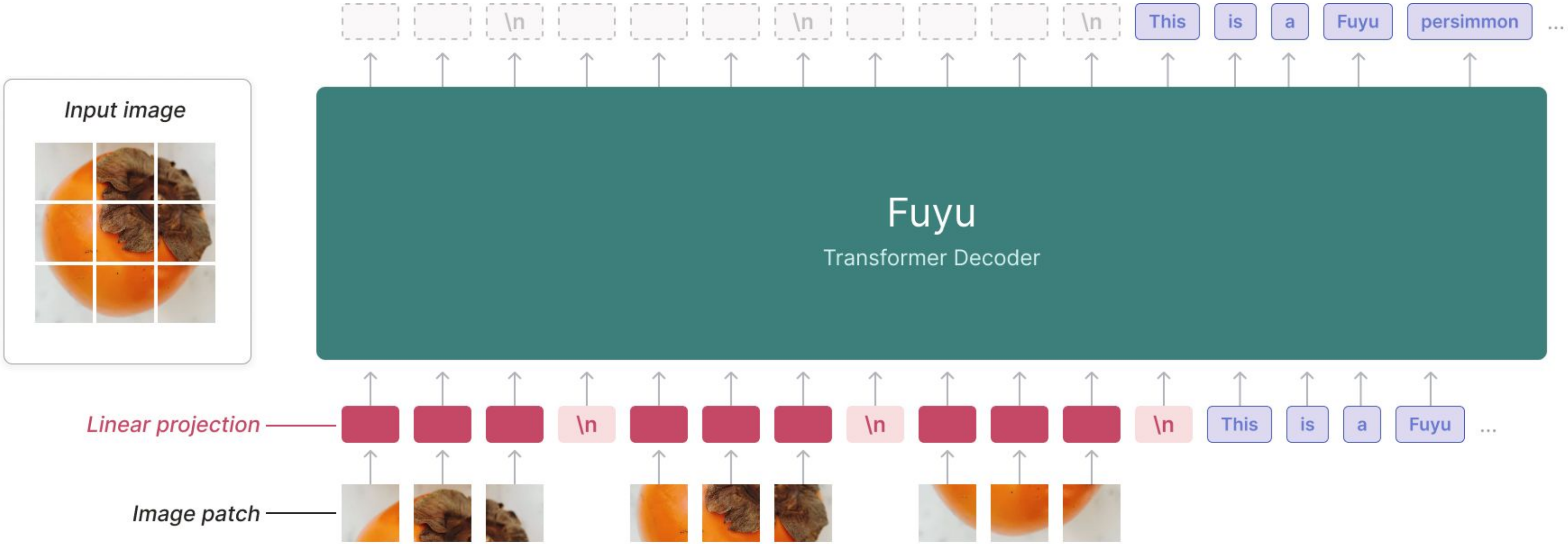
MLP

CLIP Vision Encoder
ViT-L/14

Stage 1: Pre-training for Feature Alignment
Stage 2: Fine-tuning End-to-End (W and Ø)

LLaVA (Liu et al. 2023)

Integrate Visual Patches into Input Layer



Fuyu (Rohan et al. 2023)

Which One is the Best?

Architecture

- ❖ Integrate Visual Features into Intermediate Layers
- ❖ Integrate Visual Features into Input Layer
- ❖ Integrate Visual Patches into Input Layer

Architecture of GPT-4, Gemini 1.5, and Claude 3?

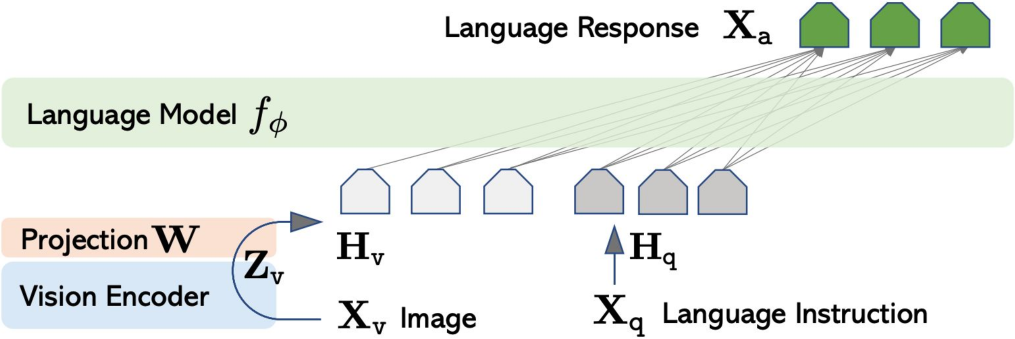
- Details Unknown

Based on the latest open-source information

- the LLaVA-style VLM (Architecture + Training Recipe) is the most effective. (Oct 2024)

LLaVA Training Recipe, Stage 1: Pre-training

Goal: Align visual features to LLM's word embedding space



Data: Converted from image captioning data

- 595K image-text pairs filtered from CC3M
- convert to instruction-following format

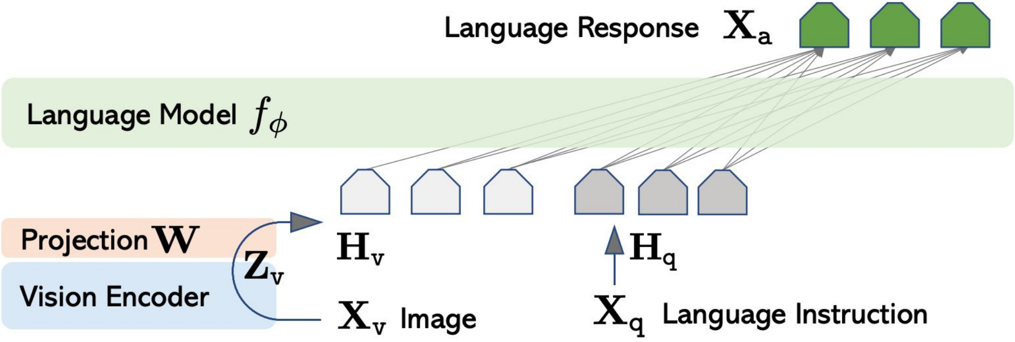
Trainable Parameters

- Only W

LLaVA (Liu et al. 2023)

LLaVA Training Recipe, Stage 2: Instruction Tuning (Supervised Finetuning)

Goal: Visual Captioner → Visual Assistant (Follow user instructions)



Data: leverage ChatGPT/GPT-4 for multimodal instruction-following data collection

- 158K language-image instruction-following data
- 3 Types: Conversation (Multi-Turn), Detailed description, Complex reasoning

Trainable Parameters

- $\{W, \emptyset\}$

LLaVA (Liu et al. 2023)

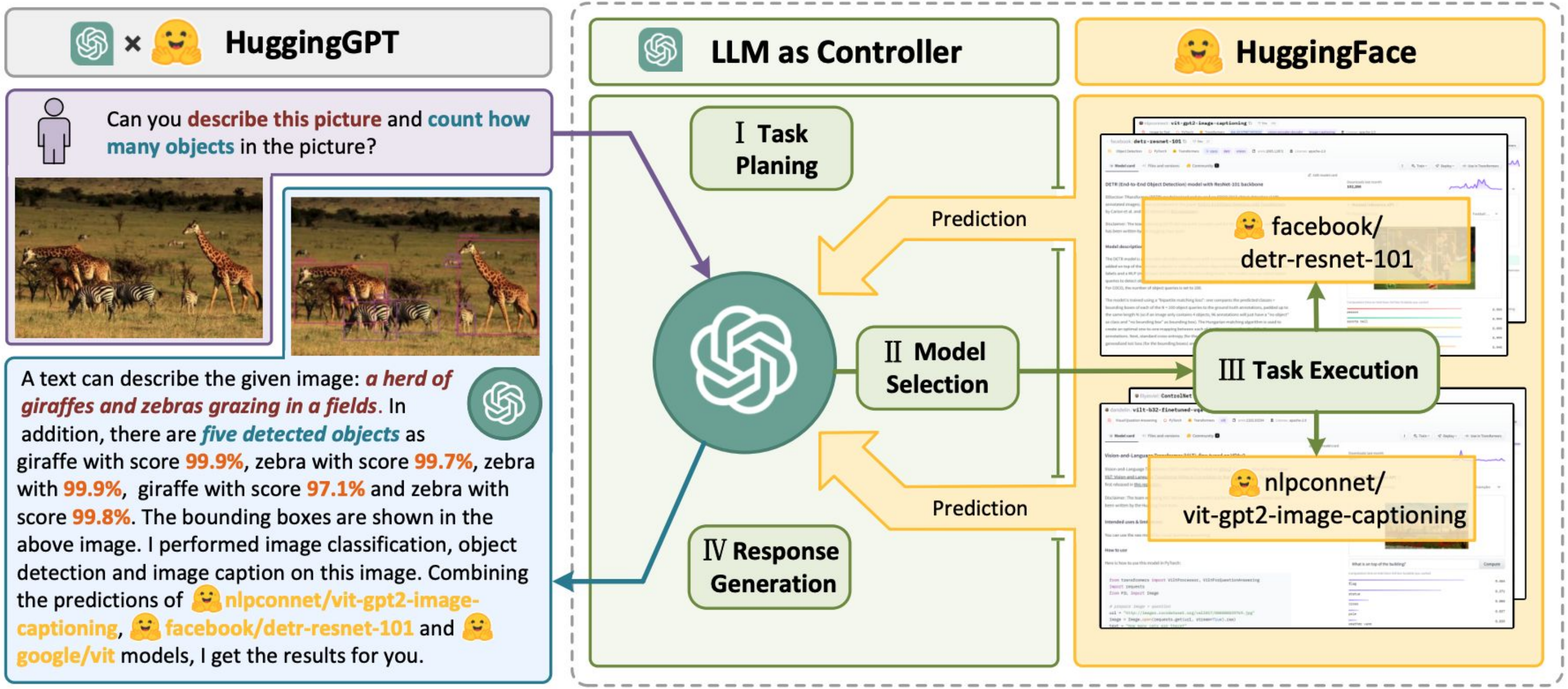
How to Link Vision to LLMs?

Internal Linkage → Vision-Language Models

- ❖ Architecture
  - ○ Integrate Visual Features into Intermediate Layers
  - ○ Integrate Visual Features into Input Layer
  - ○ Integrate Visual Patches into Input Layer

External Linkage → Vision-Language Agents

Language to Connect Vision Models

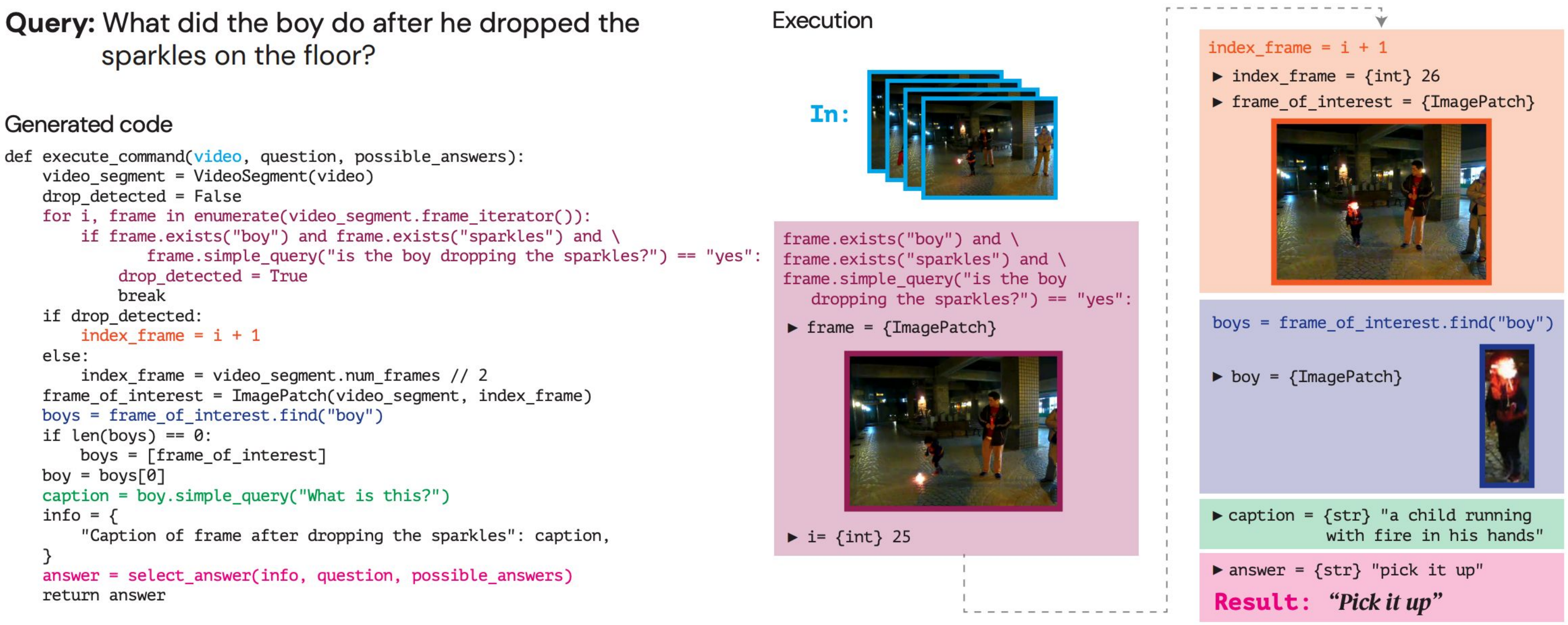


Can you **describe this picture** and **count how many objects** in the picture?

A text can describe the given image: ***a herd of giraffes and zebras grazing in a fields***. In addition, there are ***five detected objects*** as giraffe with score **99.9%**, zebra with score **99.7%**, zebra with **99.9%**, giraffe with score **97.1%** and zebra with score **99.8%**. The bounding boxes are shown in the above image. I performed image classification, object detection and image caption on this image. Combining the predictions of 🤗 **nlpconnet/vit-gpt2-image-captioning**, 🤗 **facebook/detr-resnet-101** and 🤗 **google/vit** models, I get the results for you.

HuggingGPT (Shen et al. 2023)

Code to Connect Vision Models



Query: What did the boy do after he dropped the sparkles on the floor?

Generated code

```
def execute_command(video, question, possible_answers):
    video_segment = VideoSegment(video)
    drop_detected = False
    for i, frame in enumerate(video_segment.frame_iterator()):
        if frame.exists("boy") and frame.exists("sparkles") and \
                frame.simple_query("is the boy dropping the sparkles?") == "yes":
            drop_detected = True
            break
    if drop_detected:
        index_frame = i + 1
    else:
        index_frame = video_segment.num_frames // 2
    frame_of_interest = ImagePatch(video_segment, index_frame)
    boys = frame_of_interest.find("boy")
    if len(boys) == 0:
        boys = [frame_of_interest]
    boy = boys[0]
    caption = boy.simple_query("What is this?")
    info = {
        "Caption of frame after dropping the sparkles": caption,
    }
    answer = select_answer(info, question, possible_answers)
    return answer
```

Execution

```
frame.exists("boy") and \
frame.exists("sparkles") and \
frame.simple_query("is the boy
   dropping the sparkles?") == "yes":
▶ frame = {ImagePatch}
▶ i= {int} 25
```

```
index_frame = i + 1
▶ index_frame = {int} 26
▶ frame_of_interest = {ImagePatch}
```

```
boys = frame_of_interest.find("boy")
▶ boy = {ImagePatch}
```

```
▶ caption = {str} "a child running
             with fire in his hands"
```

```
▶ answer = {str} "pick it up"
```

Result: *"Pick it up"*

ViperGPT (Surís et al. 2023)

Key takeaways

- Wide variety of commercial vision-language generative models (GPT, Gemini, Claude families, etc.) These are the most powerful models available, typically only high-level details available and not-open source. The best model changes frequently.
- LLaVA is a strong open-source VLM model, frequently used in research settings
  - Qwen2-VL is another recent model to check out (in the discussion papers list)
- Common practice is fine-tuning LLaVA, we have discussed training recipe
- Generating and improving instruction-tuning data is a major area of research, we also saw examples in biomedical applications (e.g., LLaVA-Med and Med-Gemini generalist models, CheXagent and LLaVA-Rad specialist models)
- Externally linked vision-language agents are also interesting directions of exploration

Now:

- Special guest: Troy Tazbaz