CSCI 5541: Natural Language Processing

Lecture 17: Multimodal NLP

computer science & engineering

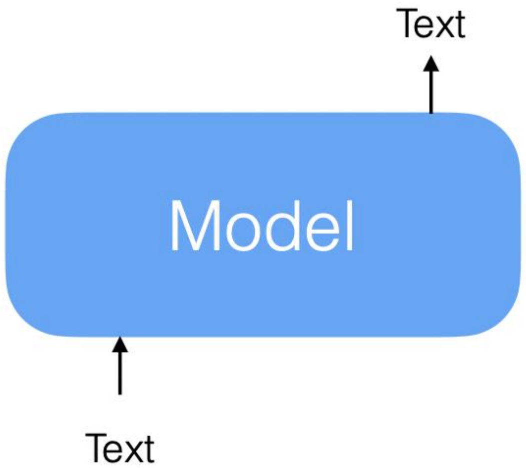
MINNESOTA NLP EST. 2021

Slides borrowed from [Sean Welleck (CMU)]

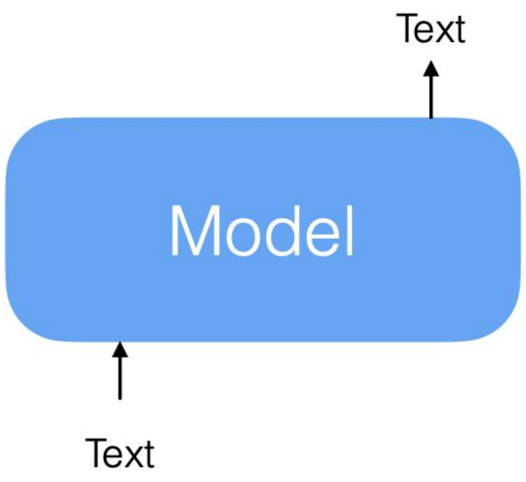
UNIVERSITY OF MINNESOTA
Driven to Discover®

Announcement (0417)

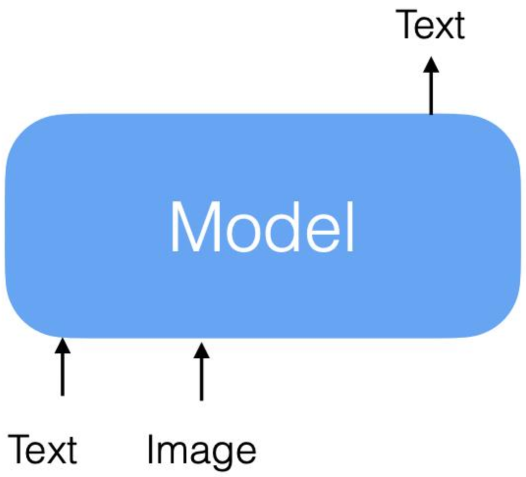
- HW5 due 4/22
- HW6 released this evening



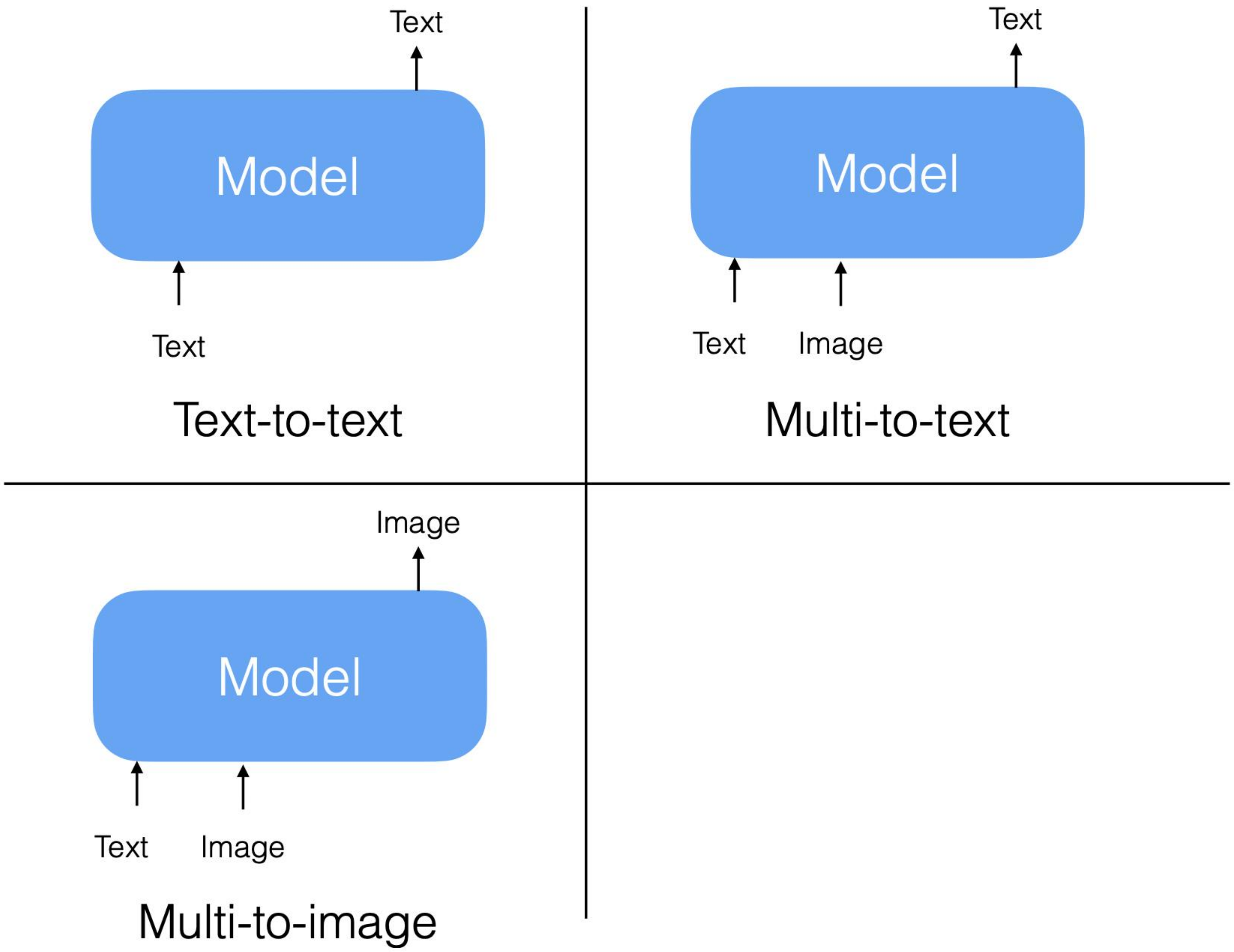
Text-to-text

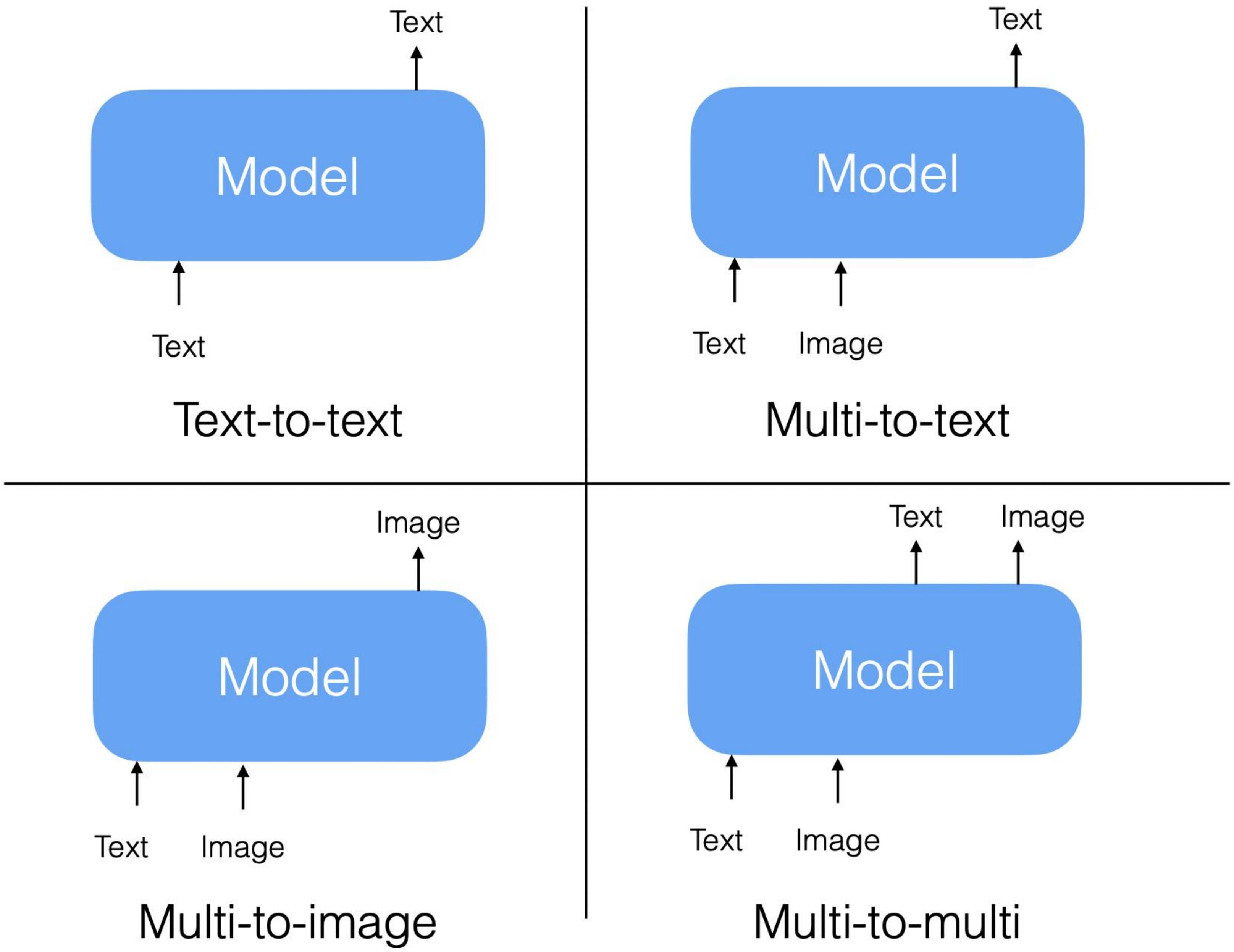


Text-to-text

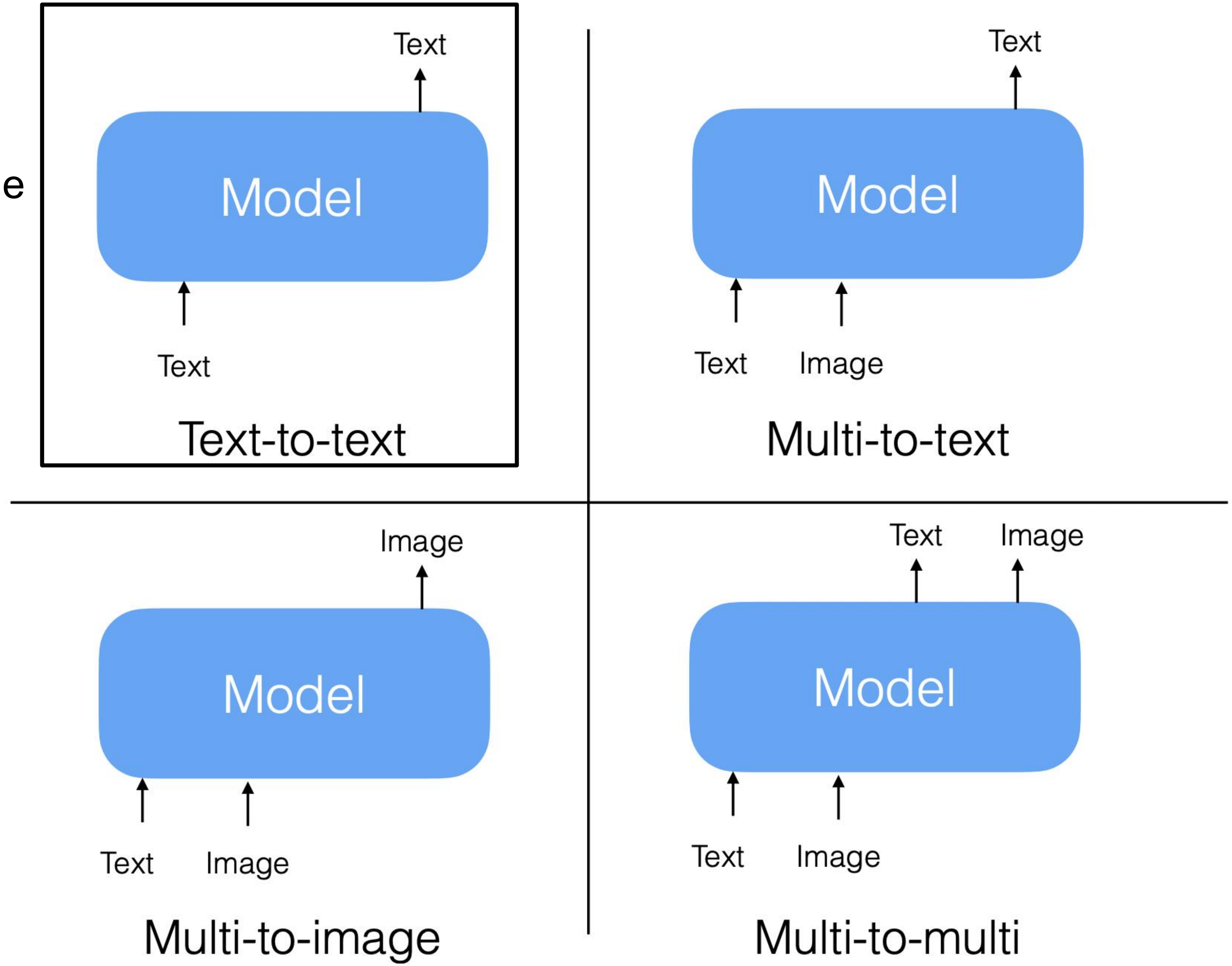


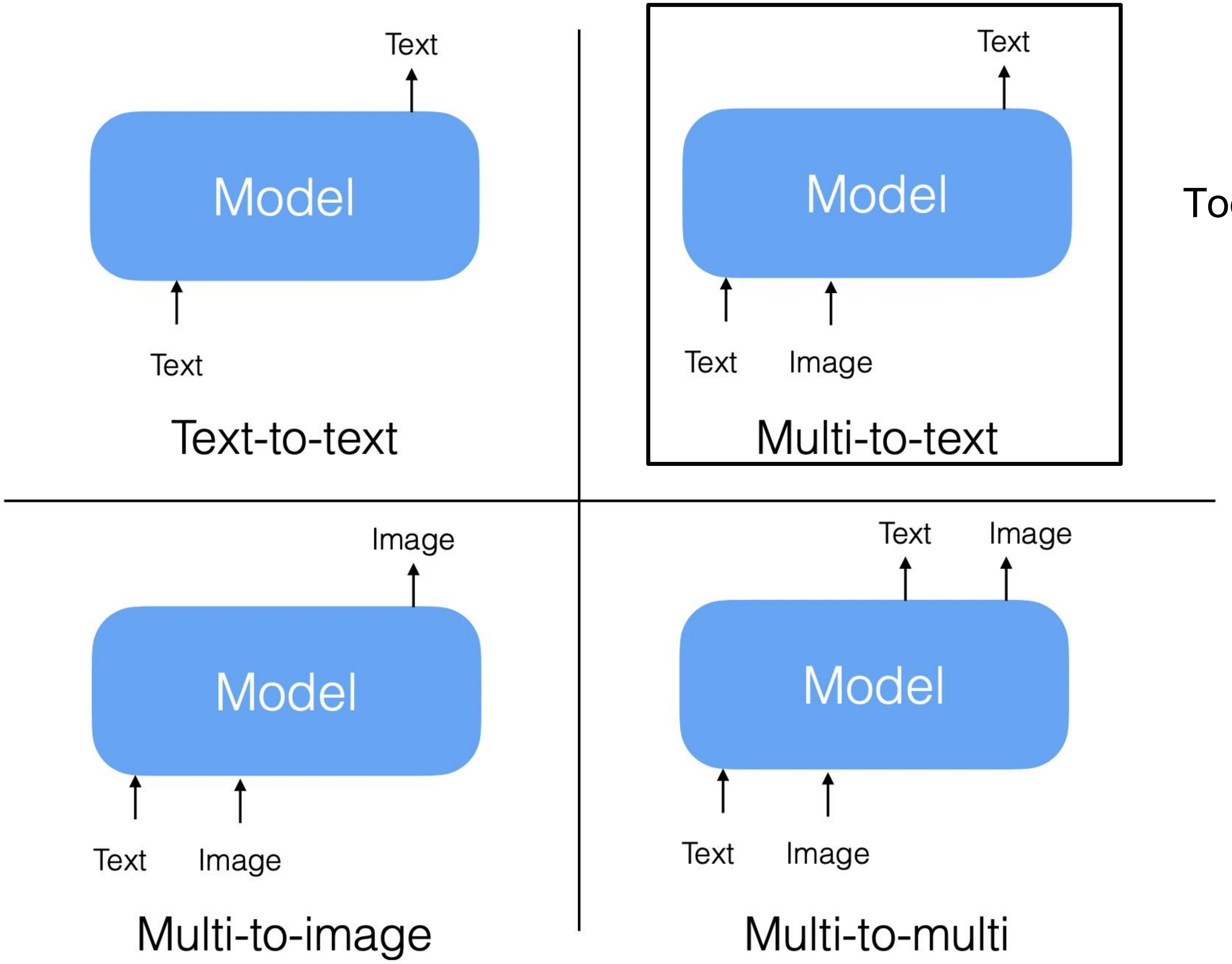
Multi-to-text


Text
Model
Text
Text-to-text
Text
Model
Text
Image
Multi-to-text
Image
Model
Text
Image
Multi-to-image


Text
Model
Text
Text-to-text
Text
Model
Text
Image
Multi-to-text
Image
Model
Text
Image
Multi-to-image
Text
Image
Model
Text
Image
Multi-to-multi

Most of this course





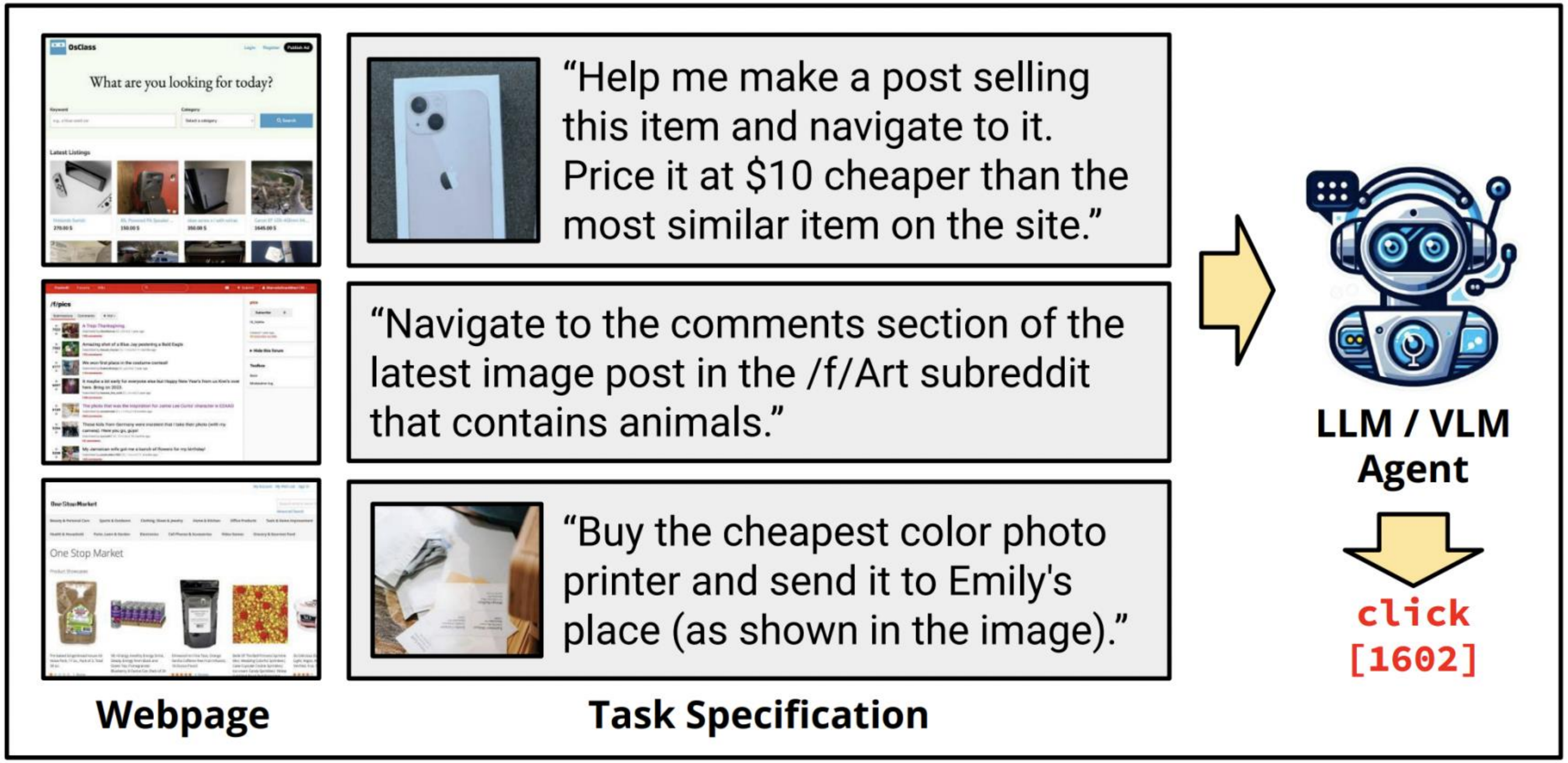
Today's Lecture

tell me about this image

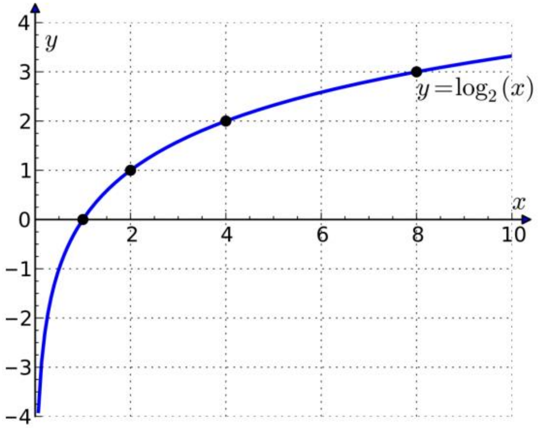
This image showcases a beautiful painting of Pittsburgh, presented in an ornate gold frame. The artwork captures the city's iconic skyline, featuring its distinctive yellow bridges spanning the blue waters below. The cityscape is rendered in soft, impressionistic brushstrokes that create a dreamy, almost ethereal atmosphere.

The painting's background is filled with fluffy white clouds against a light blue sky, adding to the sense of depth and tranquility. The overall color palette is subdued, with yellows from the bridges, blues from the sky and water, and greens from scattered trees, creating a harmonious and serene composition.

Example: AI2 Molmo interface



Example: Web Agents (Covered briefly in Agents/Modern Evaluation Lectures)



Question: The derivative of y at $x = 6$ is ____ that at $x = 8$.
Choices: (A) larger than (B) equal to (C) smaller than
Answer: (A) larger than

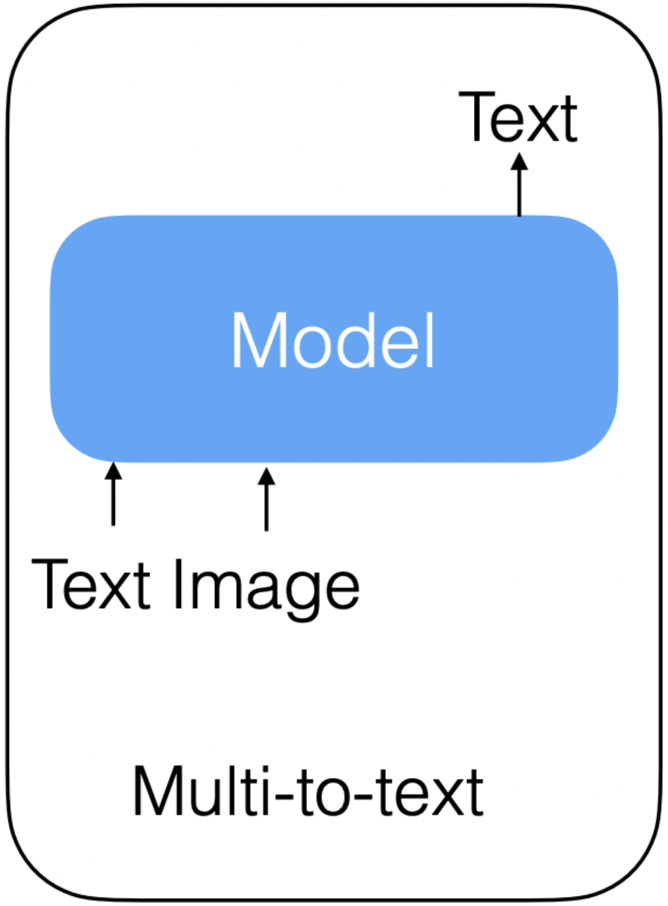
Question: How many zeros does this function have?
Answer: 1

Question: What is the value of y at $x = 1$?
Answer: 0

Example: mathematical reasoning [MathVista, Lu et al 2024]

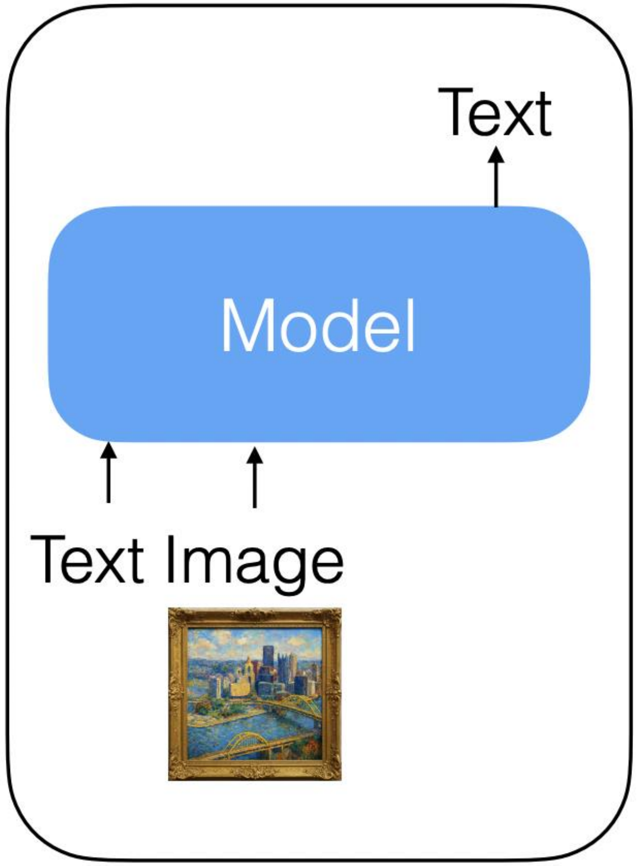
Today's lecture

- Vision architecture basics
  - ViT [Dosovitskiy et al 2020]
- Learning image representations
  - CLIP [Radford et al 2021]
- Combining with a language model
  - Llava [Liu et al 2023]



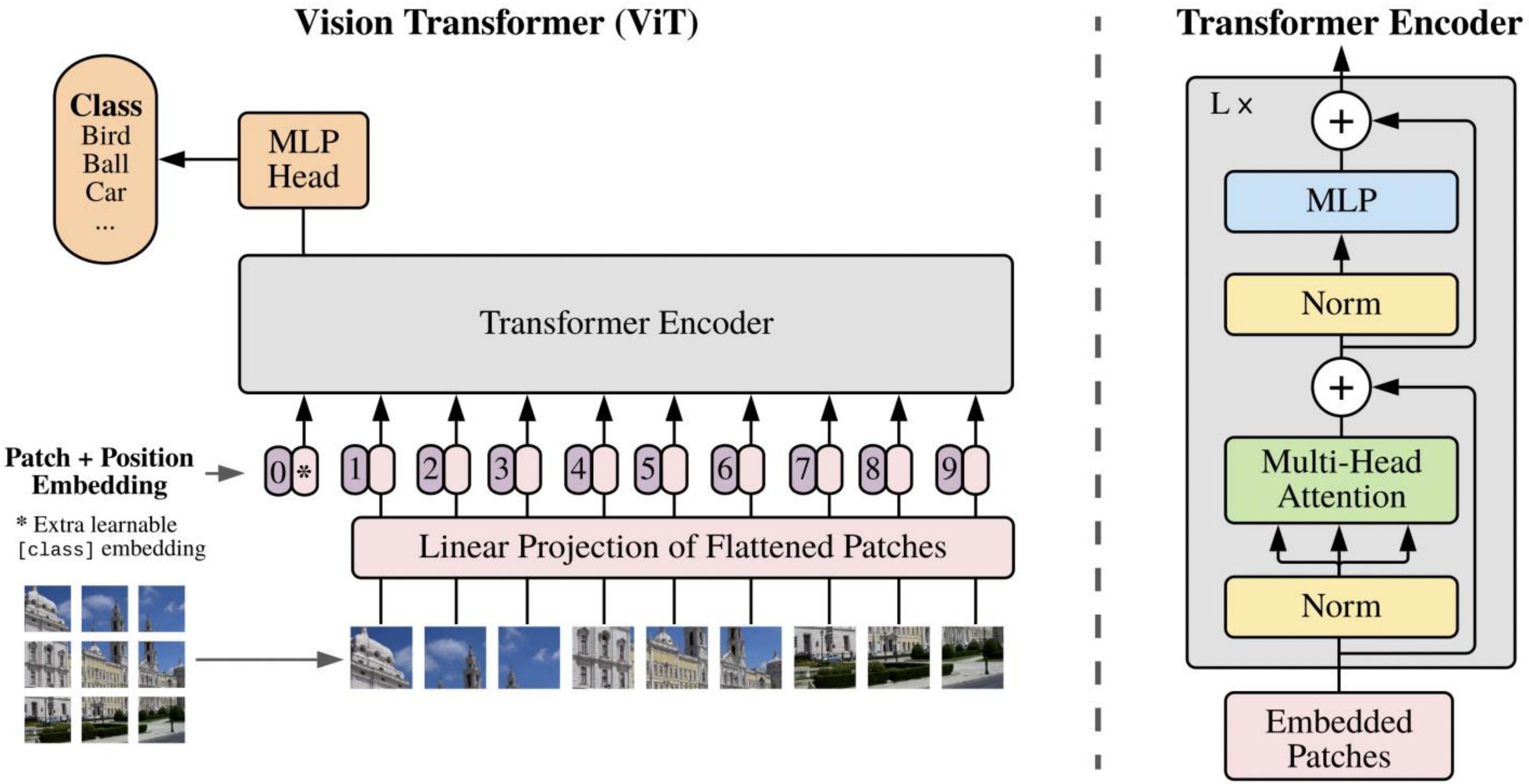
Key problem: representing images

- We represent text as a sequence of vectors (token embeddings)
- We want to also represent an image as a sequence of vectors
  - $f_{\text{enc}}(x_{\text{image}}) \rightarrow z_1, \ldots, z_L$
- Need:
  - Neural network architecture
  - Algorithm for learning good vectors



Vision Transformer (ViT)

- Idea: divide an image into patches, flatten the patches into vectors, use a standard transformer



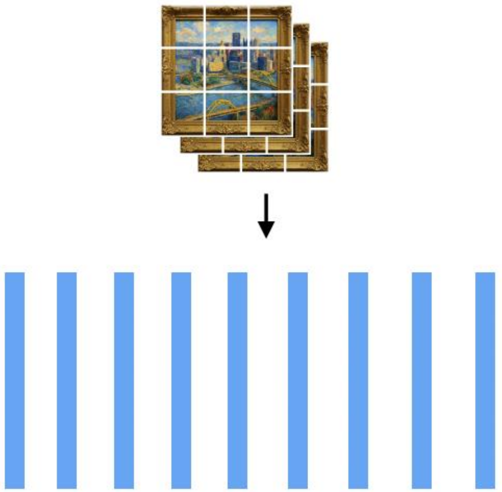
Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

120 x 120 x 3
Patch: 40 x 40

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

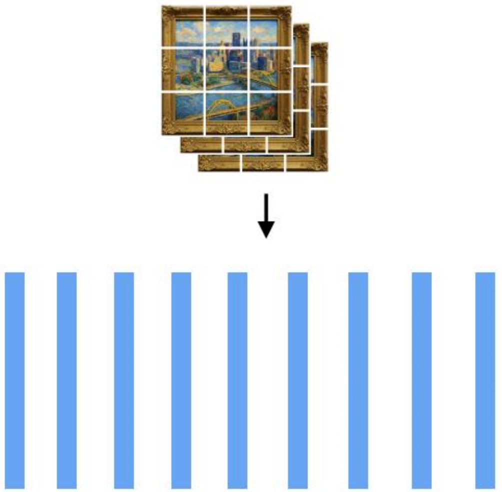


120 x 120 x 3
Patch: 40 x 40

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$
- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$



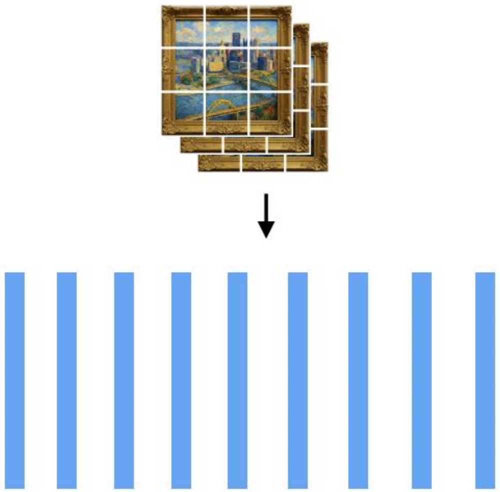
120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$



120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

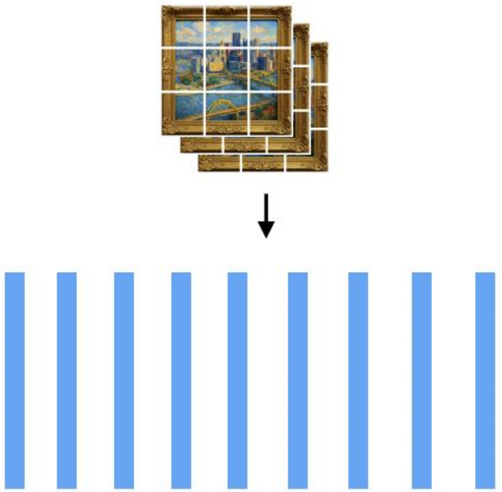
$$P^2 \cdot C = (40)(40)(3)$$

= 4800

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$



120 x 120 x 3

Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

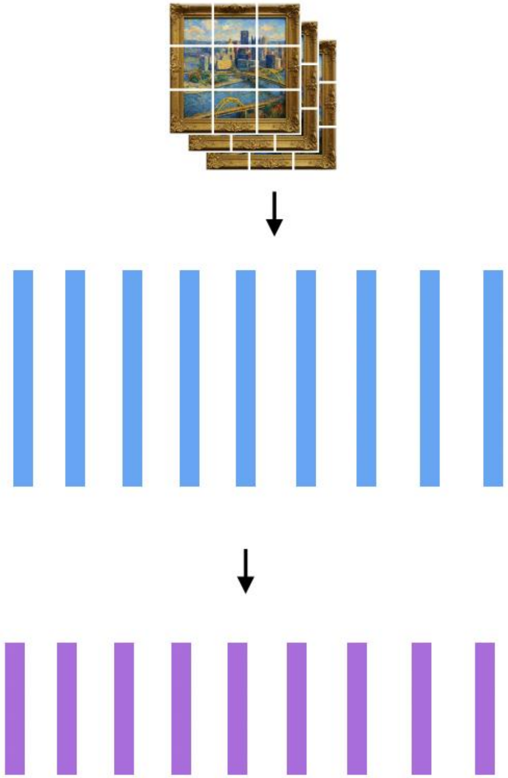
patches

$$P^2 \cdot C = (40)(40)(3)$$

= 4800

$$\implies x_p \in \mathbb{R}^{9 \times 4800}$$

Vision Transformer (ViT)



- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

$$P^2 \cdot C = (40)(40)(3)$$

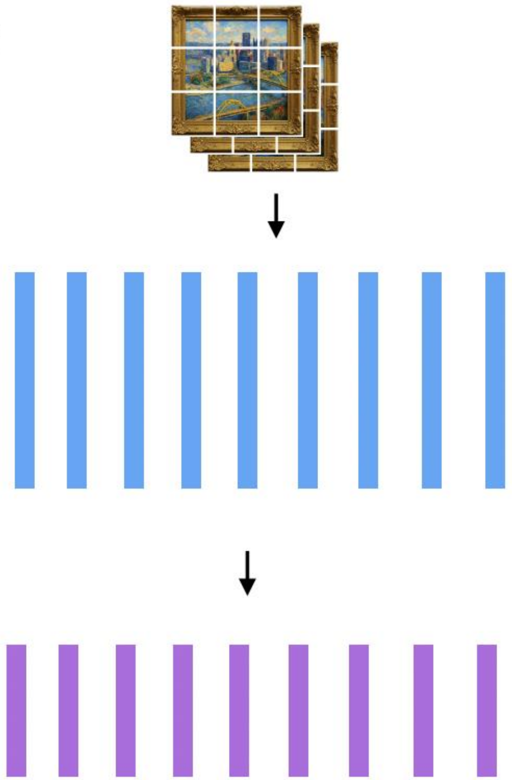
= 4800

$$\Longrightarrow \quad x_p \in \mathbb{R}^{9 \times 4800}$$

- $x \in \mathbb{R}^{N \times D}$

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$



120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

$$P^2 \cdot C = (40)(40)(3)$$

= 4800

$$\implies x_p \in \mathbb{R}^{9 \times 4800}$$

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

- $x \in \mathbb{R}^{N \times D}$
  - $x = W x_p$
    $W_e \in \mathbb{R}^{D \times (P^2 \cdot C)}$

Vision Transformer (ViT)

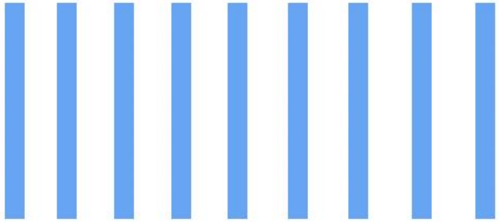
- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

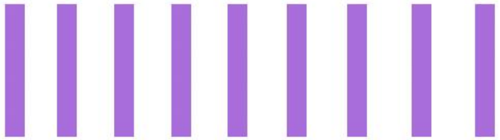
- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$



$P^2 \cdot C = (40)(40)(3)$
= 4800

$\Longrightarrow \ x_p \in \mathbb{R}^{9 \times 4800}$

- $x \in \mathbb{R}^{N \times D}$

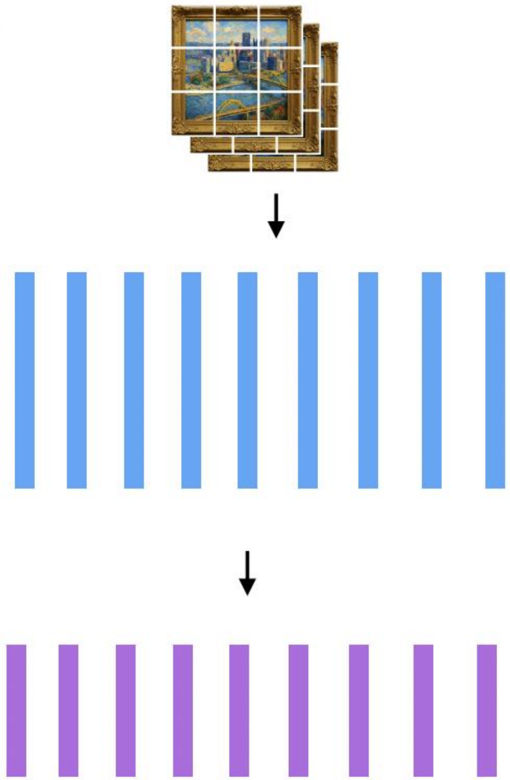


$W \in \mathbb{R}^{1024 \times 4800}$

  - $x = W x_p$
    $W_e \in \mathbb{R}^{D \times (P^2 \cdot C)}$

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$
- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$
- $x \in \mathbb{R}^{N \times D}$
    - $x = W x_p$
      $W_e \in \mathbb{R}^{D \times (P^2 \cdot C)}$



120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

$$P^2 \cdot C = (40)(40)(3)$$

= 4800

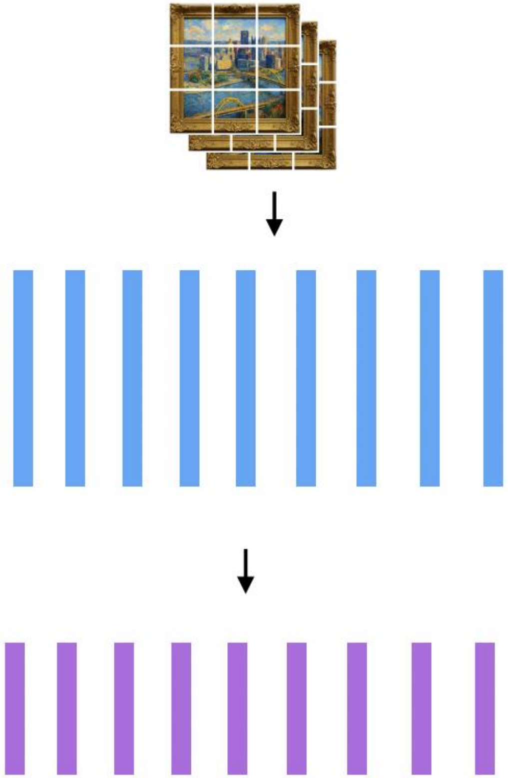
$$\implies x_p \in \mathbb{R}^{9 \times 4800}$$

$$W \in \mathbb{R}^{1024 \times 4800}$$

$$\implies x \in \mathbb{R}^{9 \times 1024}$$

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$
- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$
- $x \in \mathbb{R}^{N \times D}$
  - $x = Wx_p$
    $W_e \in \mathbb{R}^{D \times (P^2 \cdot C)}$



"Patch embeddings"

120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

$$P^2 \cdot C = (40)(40)(3)$$

= 4800

$$\implies x_p \in \mathbb{R}^{9 \times 4800}$$

$$W \in \mathbb{R}^{1024 \times 4800}$$

$$\implies x \in \mathbb{R}^{9 \times 1024}$$

Vision Transformer (ViT)

- $x_{\text{image}} \in \mathbb{R}^{H \times W \times C}$

120 x 120 x 3
Patch: 40 x 40

$$N = \frac{(120)(120)}{(40)(40)} = 9$$

patches

- $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

$$P^2 \cdot C = (40)(40)(3)$$

= 4800

$$\Longrightarrow \quad x_p \in \mathbb{R}^{9 \times 4800}$$

- $x \in \mathbb{R}^{N \times D}$
  - $x = Wx_p$
    $W_e \in \mathbb{R}^{D \times (P^2 \cdot C)}$

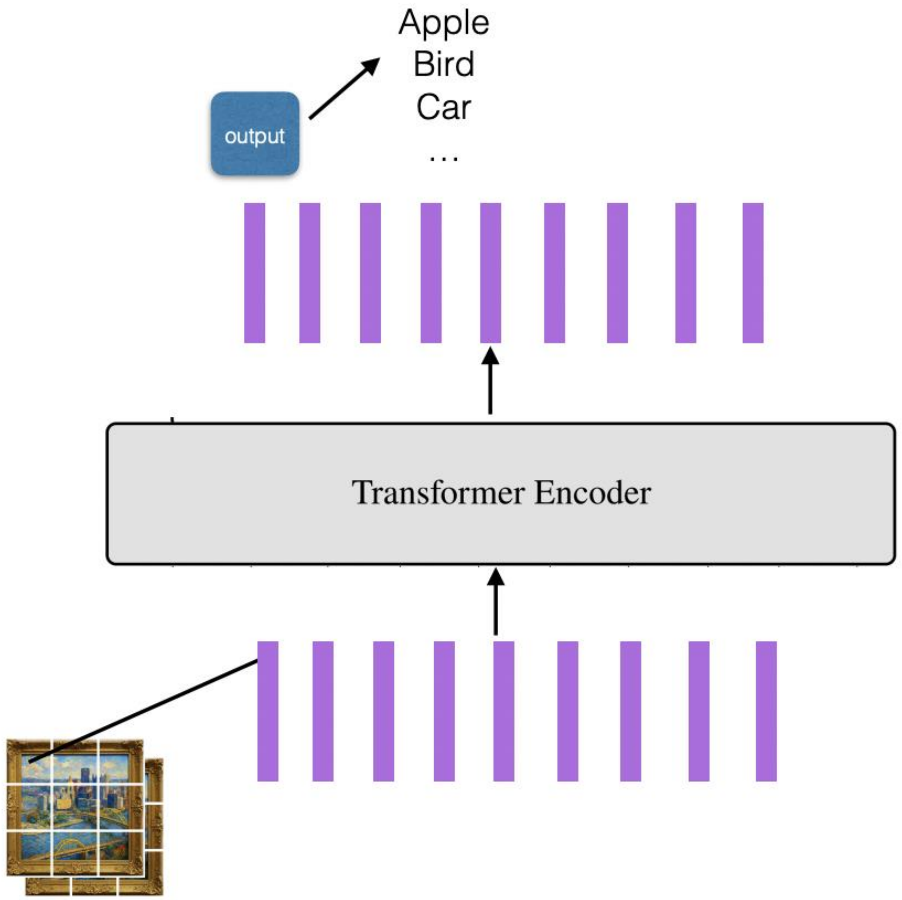
$$W \in \mathbb{R}^{1024 \times 4800}$$

$$\Longrightarrow \quad x \in \mathbb{R}^{9 \times 1024}$$

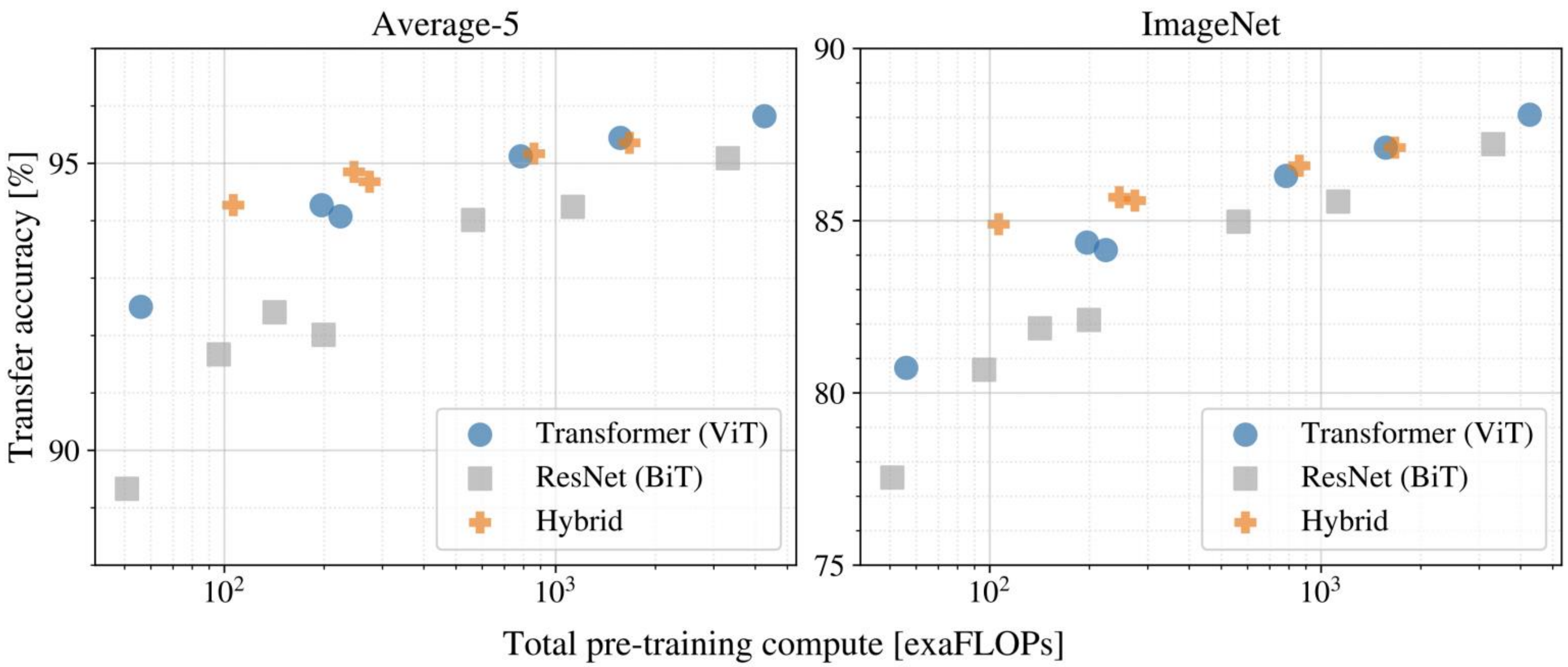
"Patch embeddings"
+ position embeddings

Vision Transformer (ViT)

- The transformer transforms the patch embeddings into vector representations $z_1, \ldots, z_N$
- We can train the model to perform a task such as classification

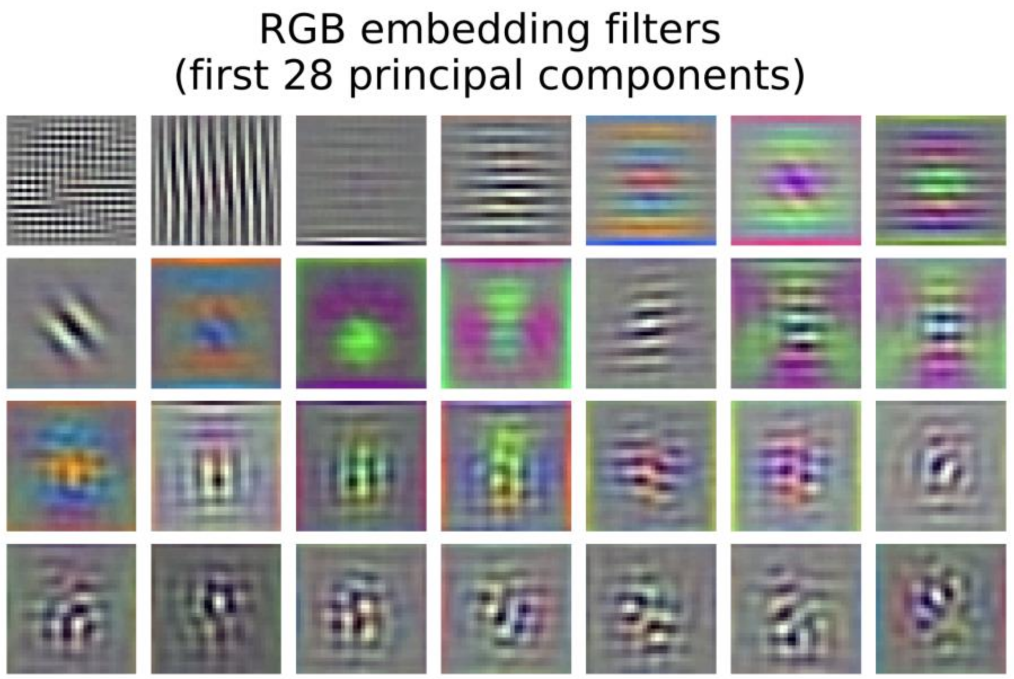


Vision Transformer (ViT)



Performance versus pre-training compute for different architectures

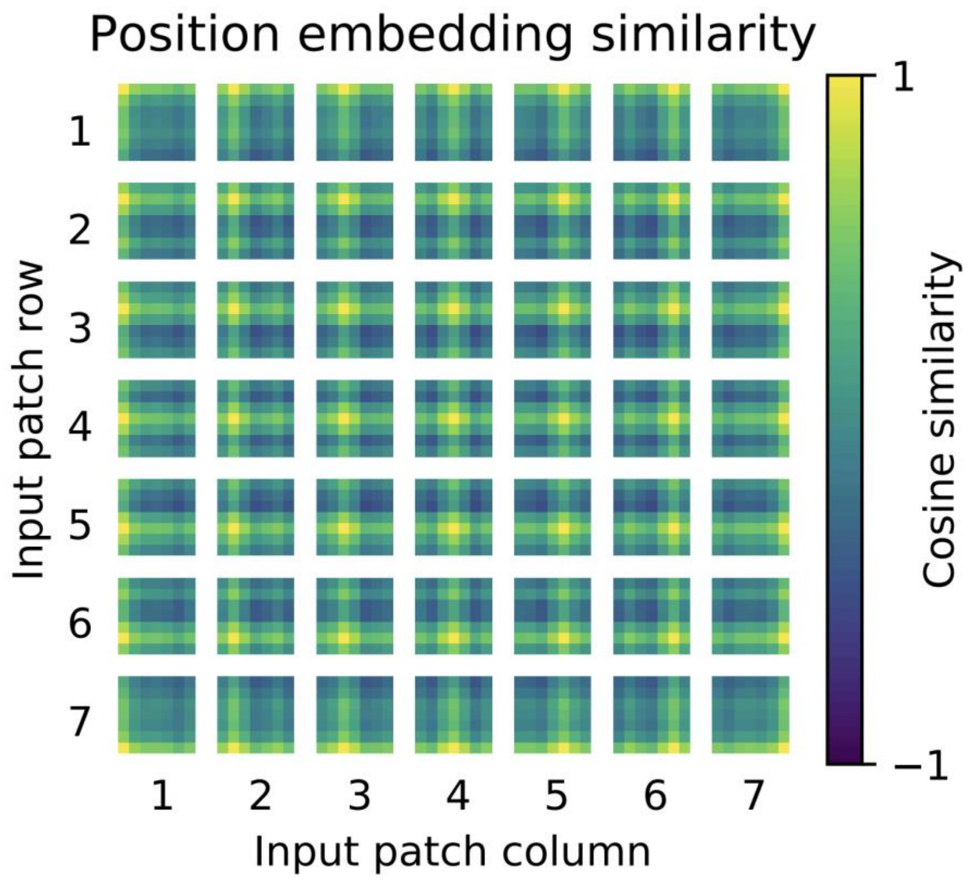
Vision Transformer (ViT)



$$x = Wx_p,\ W_e \in \mathbb{R}^{D\times(P^2\cdot C)}$$

Reshape rows into P x P, visualize principal components

Vision Transformer (ViT)

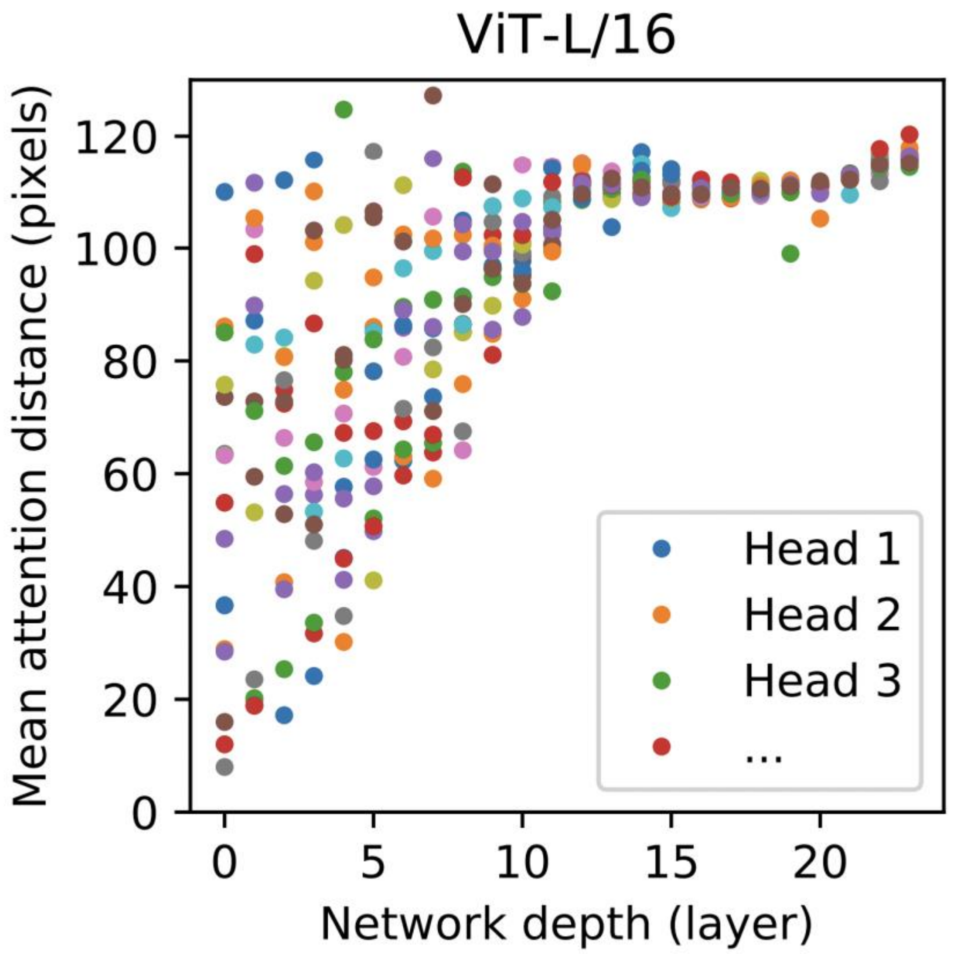


Cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches

Vision Transformer (ViT)

Can attend to regions that are salient for the task (here classification)

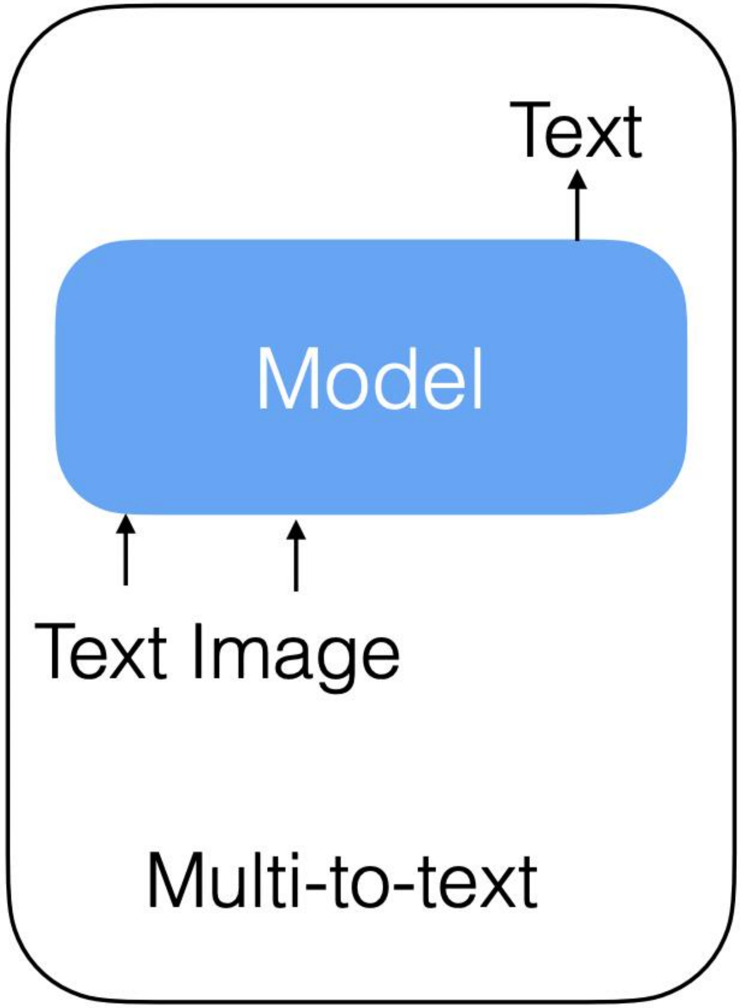
Vision Transformer (ViT)



Early layers either attend to large regions or narrow regions; later layers generally attend to larger regions

Today's lecture

- Vision architecture basics
  - ViT
- **Learning image representations**
  - **CLIP**
- Combining with a language model
  - Llava



Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text
  - At the time, models were pre-trained on classification: the only textual supervision was from the class label.

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text
  - At the time, models were pre-trained on classification: the only textual supervision was from the class label.
  - A textual description of an image provides much more information than one class label.

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text
    - At the time, models were pre-trained on classification: the only textual supervision was from the class label.
    - A textual description of an image provides much more information than one class label.
- Scalable

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text
  - At the time, models were pre-trained on classification: the only textual supervision was from the class label.
  - A textual description of an image provides much more information than one class label.
- Scalable
  - At the time, image pre-training was largely limited to hand labeled data.

Contrastive Language-Image Pre-training (CLIP)

Goal: pre-training objective for learning image representations

- Learn from text
  - At the time, models were pre-trained on classification: the only textual supervision was from the class label.
  - A textual description of an image provides much more information than one class label.
- Scalable
  - At the time, image pre-training was largely limited to hand labeled data.
  - Want to have the property of improving by adding more compute.

CLIP

- Idea: learn image and text representations jointly in a shared embedding space

CLIP

- Idea: learn image and text representations jointly in a shared embedding space
  - Learn an image encoder $f_I(x) \rightarrow z_I$

CLIP

- Idea: learn image and text representations jointly in a shared embedding space
  - Learn an image encoder $f_I(x) \rightarrow z_I$
  - Learn a text encoder $f_T(y) \rightarrow z_T$

CLIP

- Idea: learn image and text representations jointly in a shared embedding space
  - Learn an image encoder $f_I(x) \rightarrow z_I$
  - Learn a text encoder $f_T(y) \rightarrow z_T$
  - The representations for a paired image and its text should be close together.

CLIP

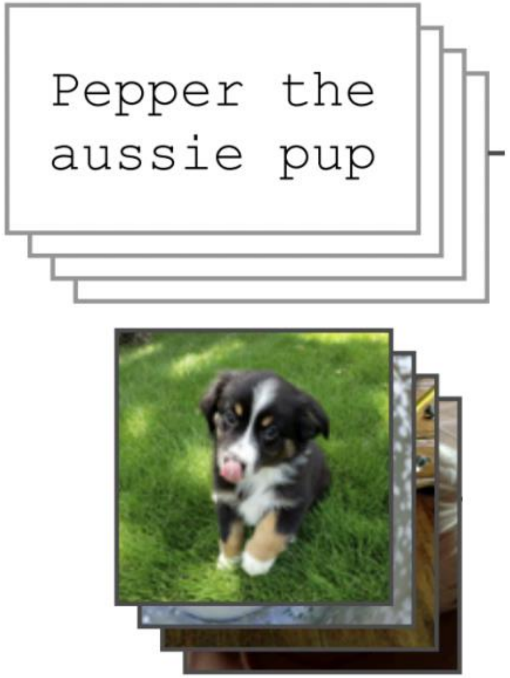
- Idea: learn image and text representations jointly in a shared embedding space
  - Learn an image encoder $f_I(x) \rightarrow z_I$
  - Learn a text encoder $f_T(y) \rightarrow z_T$
  - The representations for a paired image and its text should be close together.
  - The representations for an unpaired image and text should be far apart.

CLIP

- Idea: learn image and text representations jointly in a shared embedding space
  - Learn an image encoder $f_I(x) \rightarrow z_I$
  - Learn a text encoder $f_T(y) \rightarrow z_T$
  - The representations for a paired image and its text should be close together.
  - The representations for an unpaired image and text should be far apart.
- Apply the method over a large dataset of (image, text) pairs

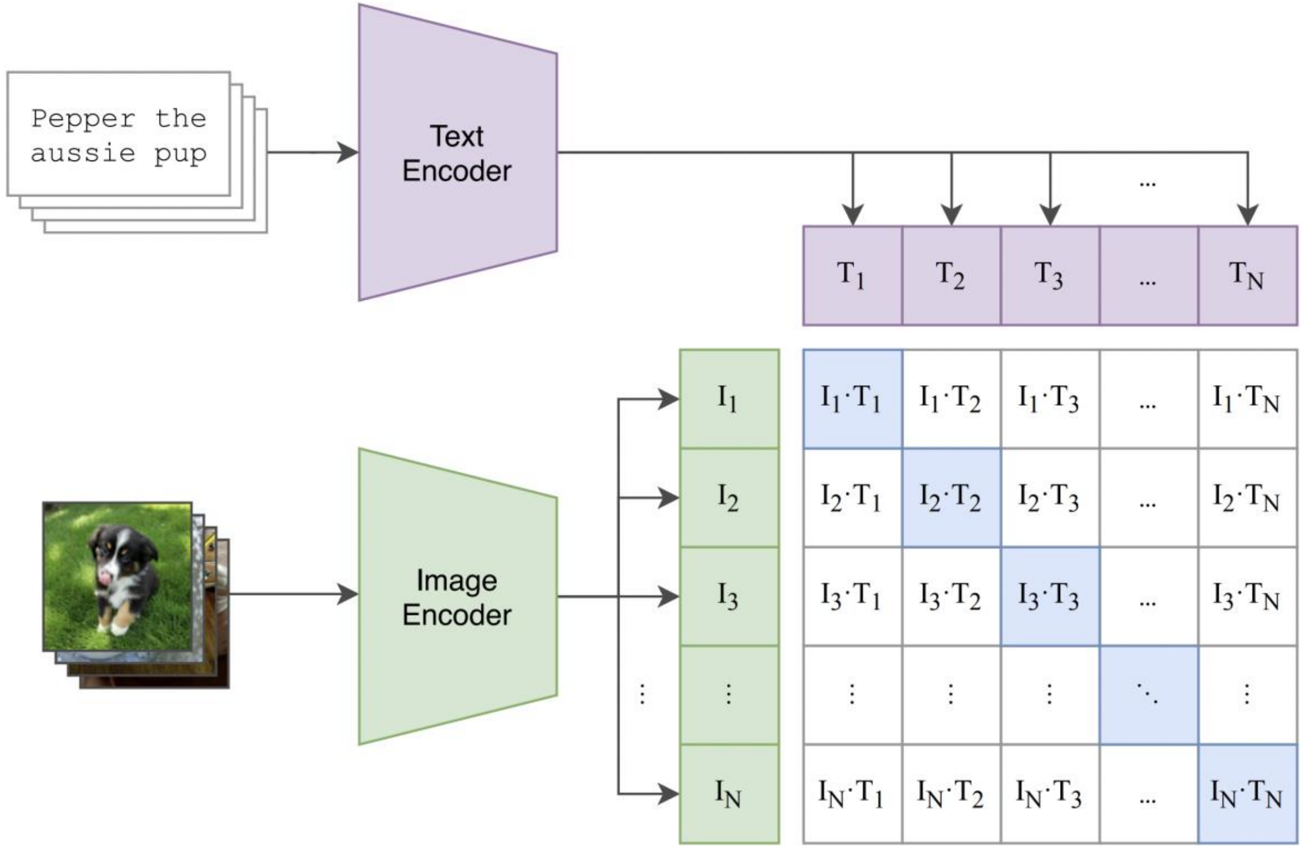
CLIP

- Data: pairs of (image, text)
  - E.g. 400 million web images with their text descriptions
- Image encoder $f_I(x) \rightarrow z_I$
  - E.g. vision transformer
- Text encoder $f_T(y) \rightarrow z_T$
  - E.g. transformer



CLIP

- Basic idea: Given *N* (image, text) pairs, classify which image is paired with which text



CLIP

$$L((x_1, y_1), \ldots, (x_N, y_N)) =$$

$$-\frac{1}{2}\sum_{n=1}^{N}\left[\log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_j)^\top f_T(y_n)\right)} + \log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_n)^\top f_T(y_j)\right)}\right]$$

Softmax over images

Softmax over text

CLIP

$$L((x_1, y_1), \ldots, (x_N, y_N)) =$$

Push up dot product
of the correct pair

$$-\frac{1}{2}\sum_{n=1}^{N}\left[\log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_j)^\top f_T(y_n)\right)} + \log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_n)^\top f_T(y_j)\right)}\right]$$

Softmax over images

Softmax over text

CLIP

$$L((x_1, y_1), \ldots, (x_N, y_N)) =$$

Push up dot product
of the correct pair

$$-\frac{1}{2}\sum_{n=1}^{N}\left[\log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_j)^\top f_T(y_n)\right)} + \log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)}{\sum_j \exp\left(f_I(x_n)^\top f_T(y_j)\right)}\right]$$

Push down dot product
of other pairs

Softmax over images

Softmax over text

CLIP

$$L((x_1, y_1), \ldots, (x_N, y_N)) =$$

Push up dot product of the correct pair

$$-\frac{1}{2}\sum_{n=1}^{N}\left[\log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)\uparrow}{\sum_j \exp\left(f_I(x_j)^\top f_T(y_n)\right)\downarrow} + \log \frac{\exp\left(f_I(x_n)^\top f_T(y_n)\right)\uparrow}{\sum_j \exp\left(f_I(x_n)^\top f_T(y_j)\right)}\right]$$

Push down dot product of other pairs

Softmax over images

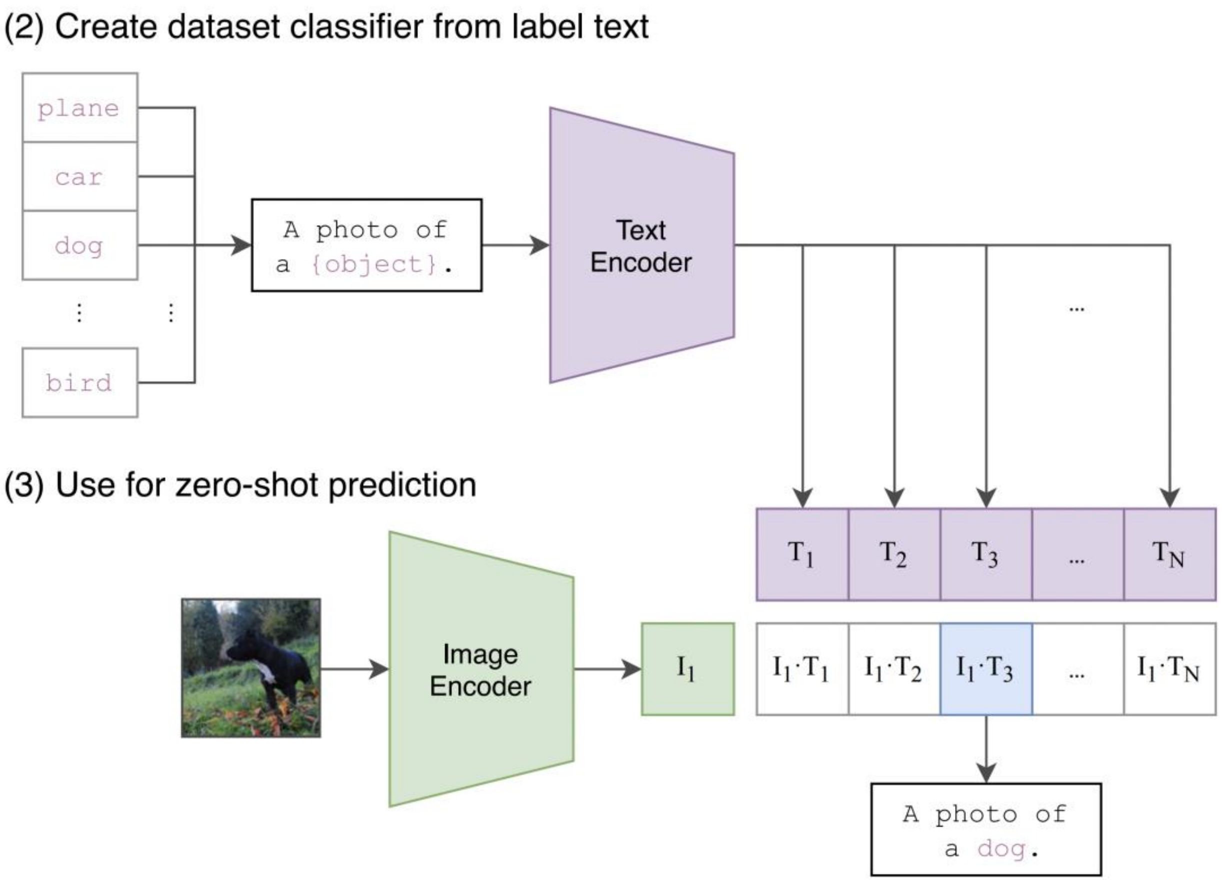
Softmax over text

Each term is a cross-entropy loss, where $p_\theta \propto f_I(x)^\top f_T(y)$ and p_* puts all of its mass on the correct pair

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modali
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis
loss_t = cross_entropy_loss(logits, labels, axis
loss   = (loss_i + loss_t)/2
```

CLIP

- Example "zero-shot" usage



CLIP

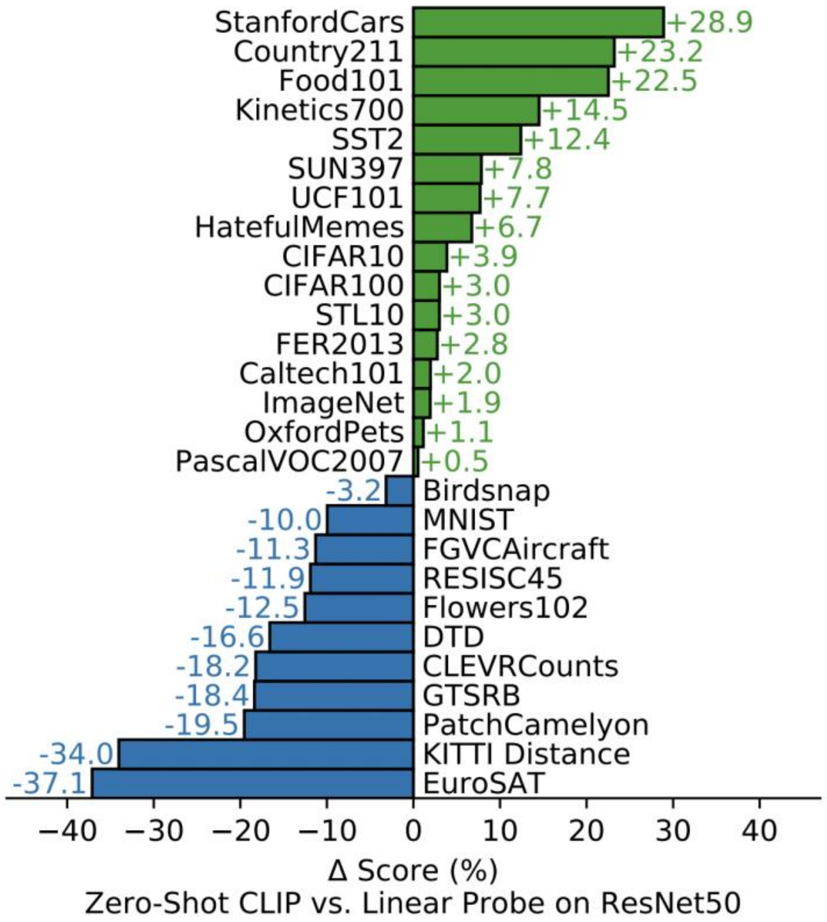
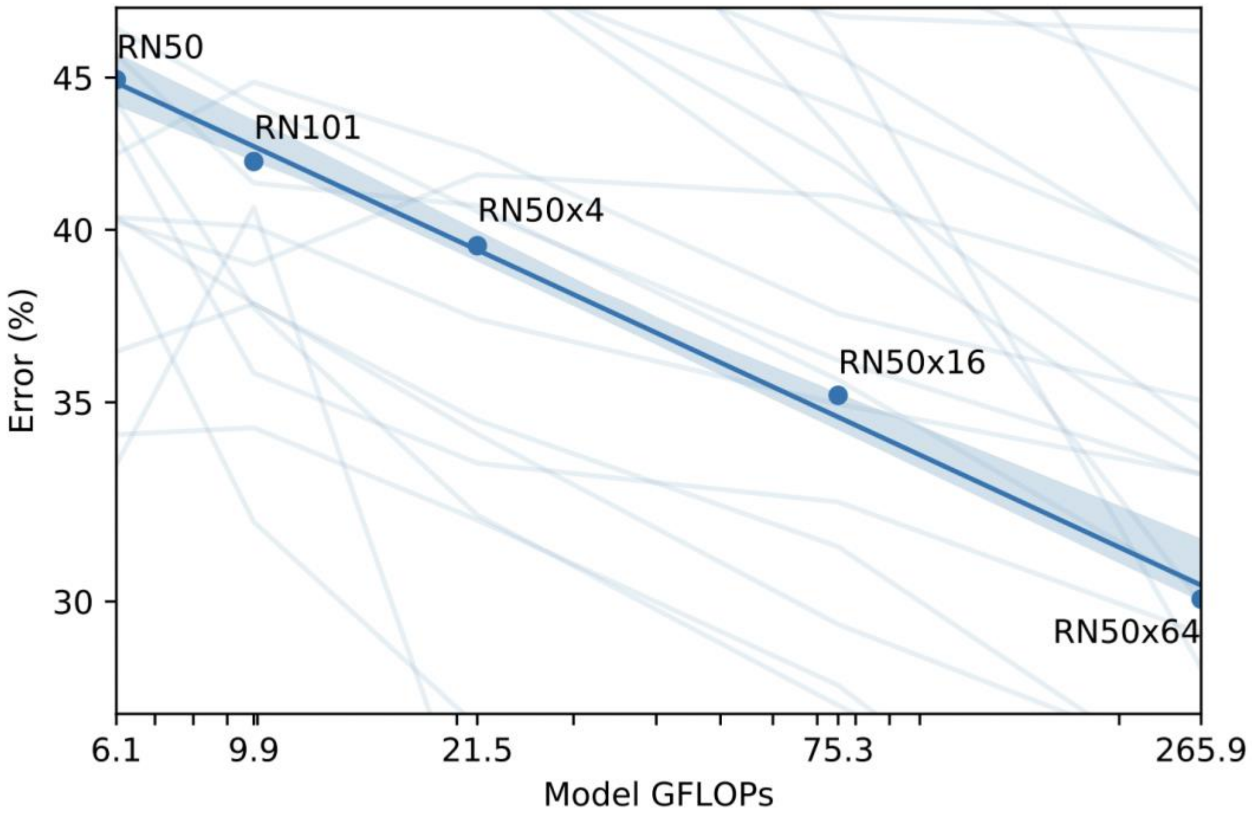


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

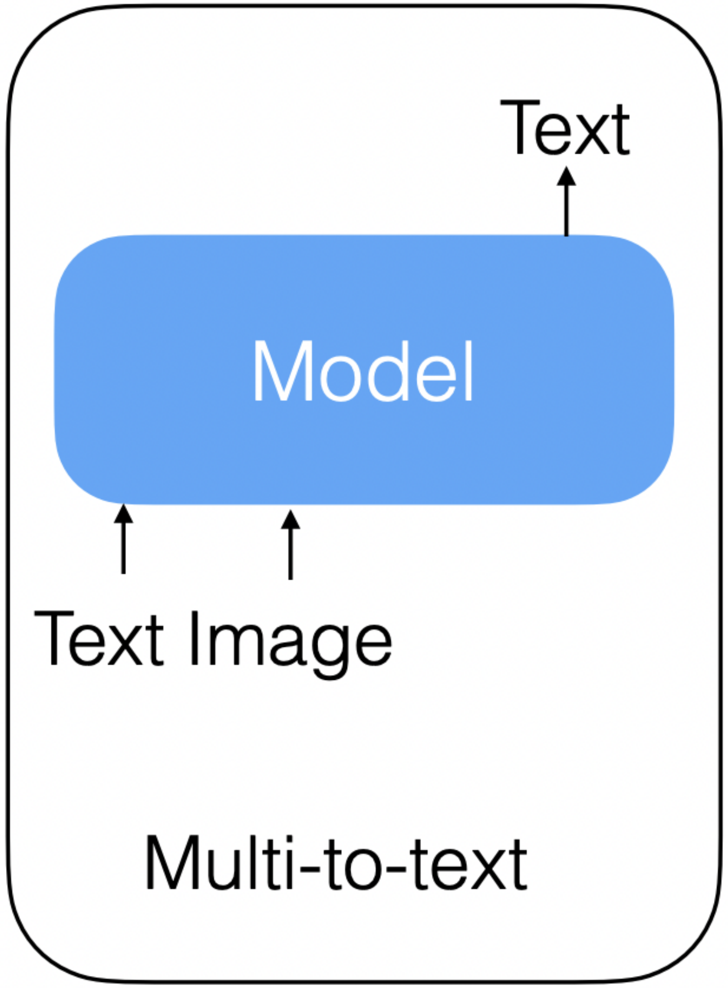
CLIP



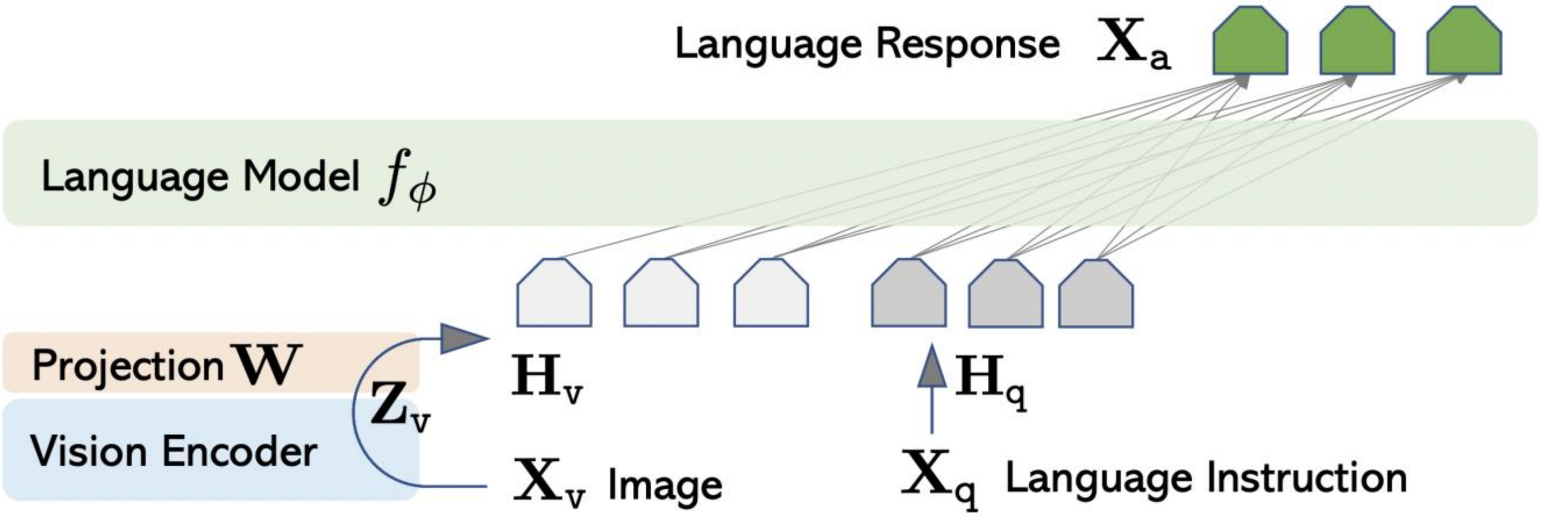
Zero-shot performance scaling as a function of pre-training compute

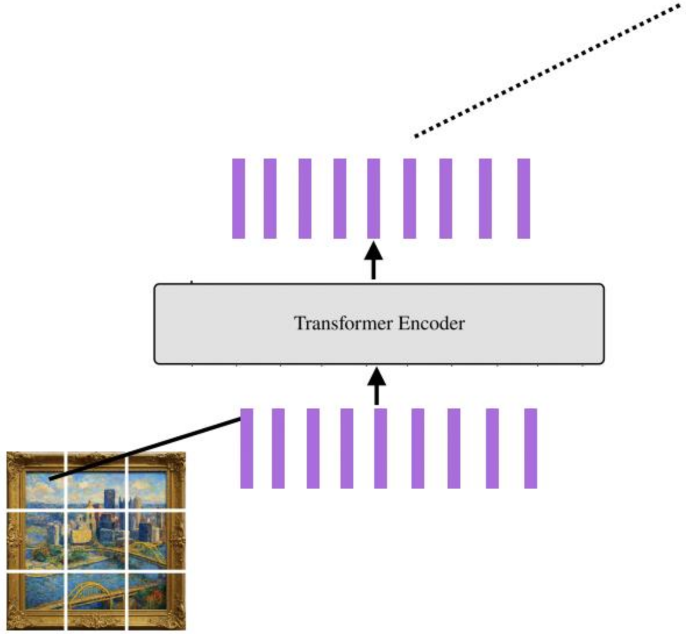
Today's lecture

- Vision architecture basics
  - ViT
- Learning image representations
  - CLIP
- **Combining with a language model**
  - **Llava**



Llava





- 'Image tokens':
  - Get sequence of vectors from the pretrained CLIP ViT
- Text tokens: same as usual

Llava

Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

User — What is unusual about this image?

LLaVA — The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, using a regular ironing board. The scene depicted in the image is peculiar as it involves a makeshift ironing setup on a vehicle, which can be both unsafe and unconventional. Additionally, it is not clear how the man is able to maintain balance and stability while ironing clothes in such an unstable environment.

General pipeline

- Image preprocessing
  - E.g. split into patches and vectorize

General pipeline

- Image preprocessing
    - E.g. split into patches and vectorize
- Image encoding
    - E.g. use a pre-existing CLIP model, get the ViT vectors from the last layer

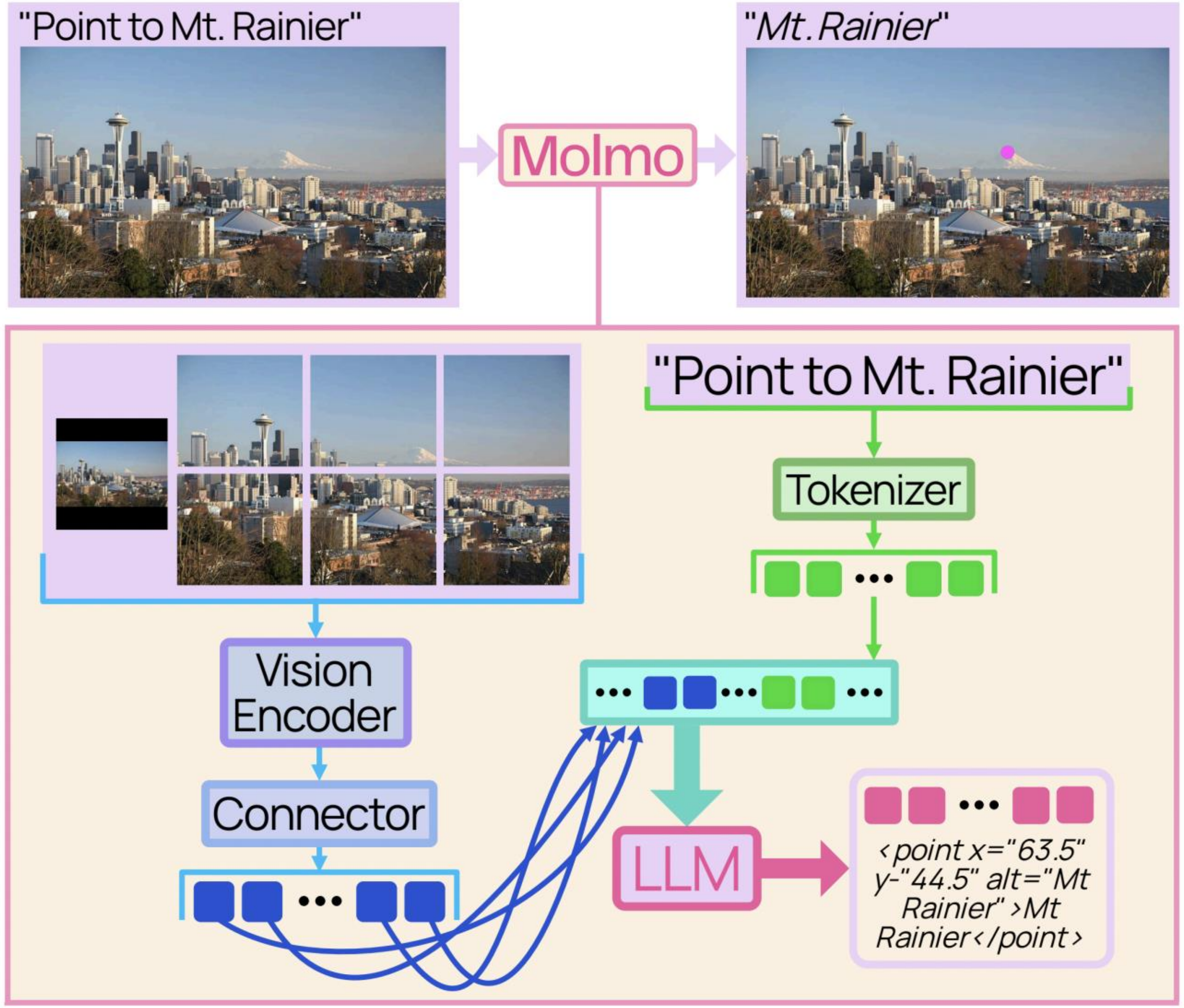
General pipeline

- Image preprocessing
  - E.g. split into patches and vectorize
- Image encoding
  - E.g. use a pre-existing CLIP model, get the ViT vectors from the last layer
- Provide the encodings to a LLM
  - E.g. linearly transform the vectors to be the model's embedding dimension

General pipeline

- Image preprocessing
  - E.g. split into patches and vectorize
- Image encoding
  - E.g. use a pre-existing CLIP model, get the ViT vectors from the last layer
- Provide the encodings to a LLM
  - E.g. linearly transform the vectors to be the model's embedding dimension
- Train/fine-tune on data that has text and images
  - For image positions, skip the loss

Example: MOLMO (AI2)



Example: MOLMO (AI2)

- Image preprocessing

Example: MOLMO (AI2)

- Image encoding: CLIP ViT-L/14 336px

Example: MOLMO (AI2)

- Image encoding: CLIP ViT-L/14 336px
  - 336 x 336 image
  - 14 x 14 patches
  - => 24 x 24 grid

Example: MOLMO (AI2)

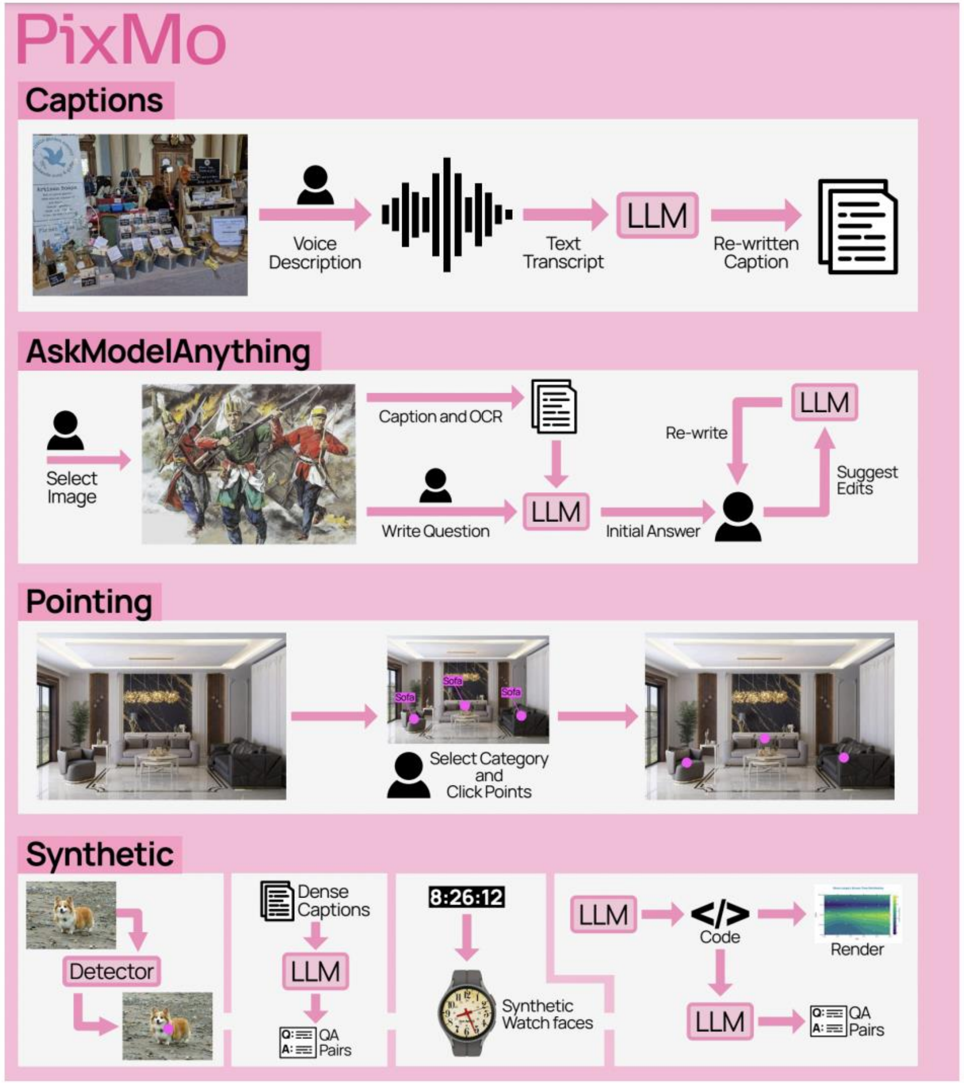
- Image encoding: CLIP ViT-L/14 336px
  - 336 x 336 image
  - 14 x 14 patches
  - => 24 x 24 grid
- Pool together each 2x2 patch subset then transform to the LLM's embedding dimension
  - => 12 x 12 vectors

Example: MOLMO (AI2)

- Image encoding: CLIP ViT-L/14 336px
  - 336 x 336 image
  - 14 x 14 patches
  - => 24 x 24 grid
- Pool together each 2x2
  patch subset then transform
  to the LLM's embedding dimension
  - => 12 x 12 vectors
- Do the above for 1 full image and 12 crops

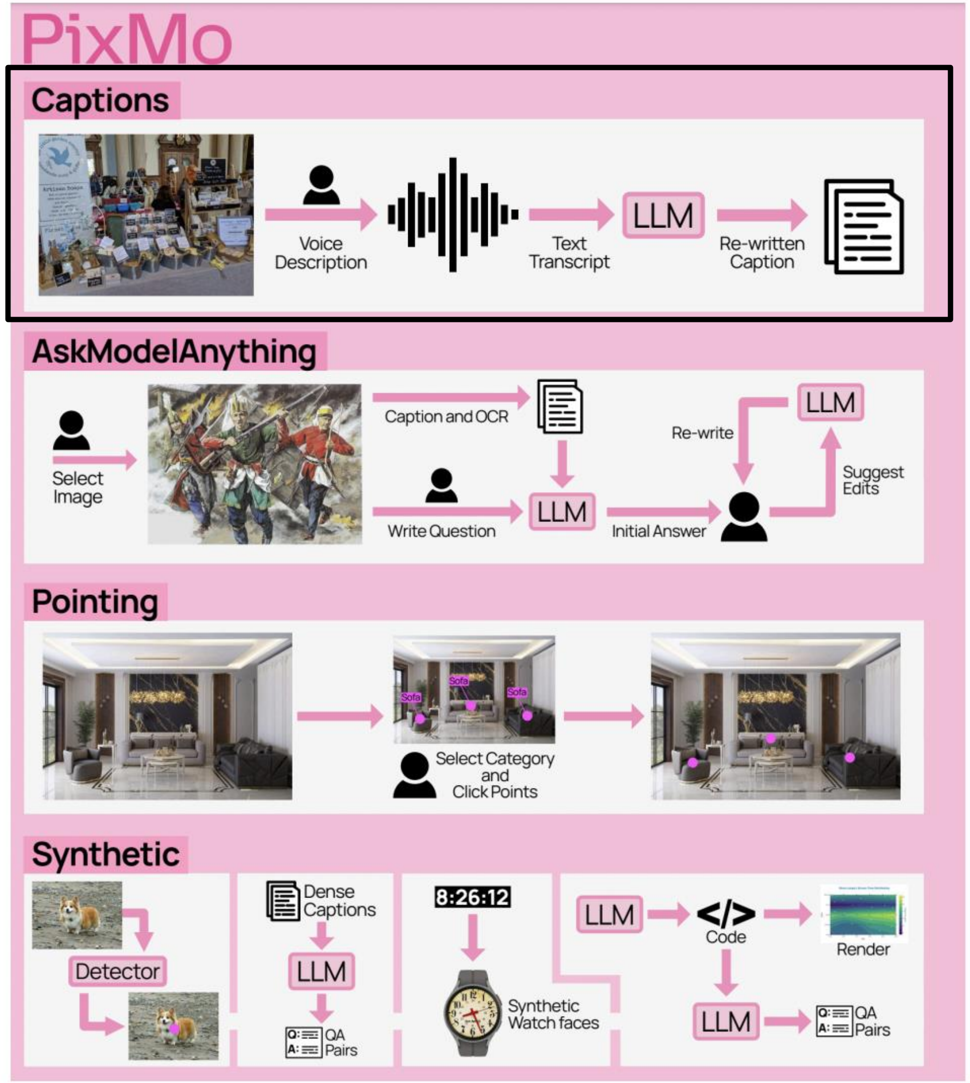
Example: MOLMO (AI2)

- Data



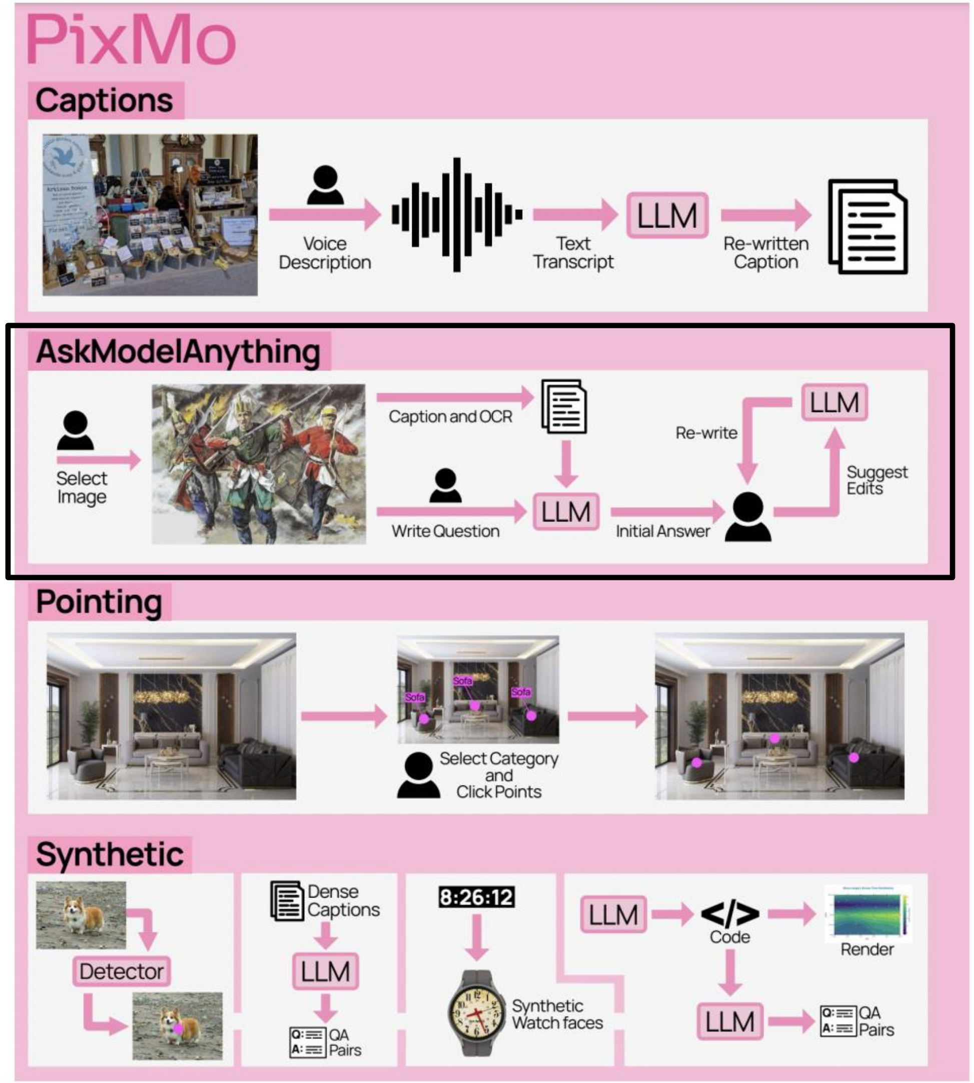
Example: MOLMO (AI2)

- Data



Example: MOLMO (AI2)

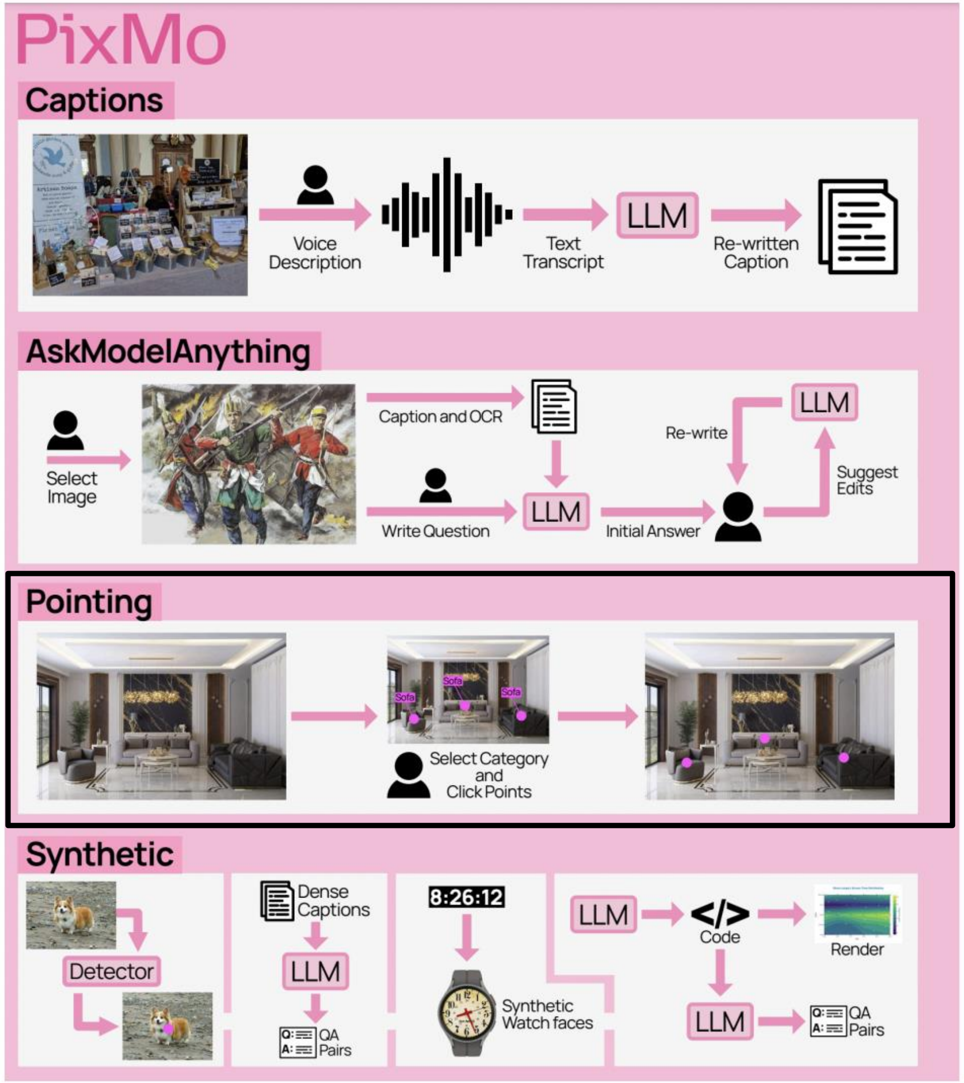
- Data



 https://files.syserss.com/cp/upload/bouncinbswebsite/items/Popcorn_Machine_Rental_for_your_Tacoma_Event.png ↗	e1f9826dddb0817a42191f1645cc1af730d0aa4607c1988ac38b13a6871a76ea	What type of machine is this?	This is a vintage-style popcorn cart. It features a red body, glass front, and gold lettering. The cart is on wheels, making it mobile and perfect for events like carnivals, fairs, or movie theaters. It combines functionality with a charming 1950s aesthetic, ideal for making and displaying fresh popcorn.

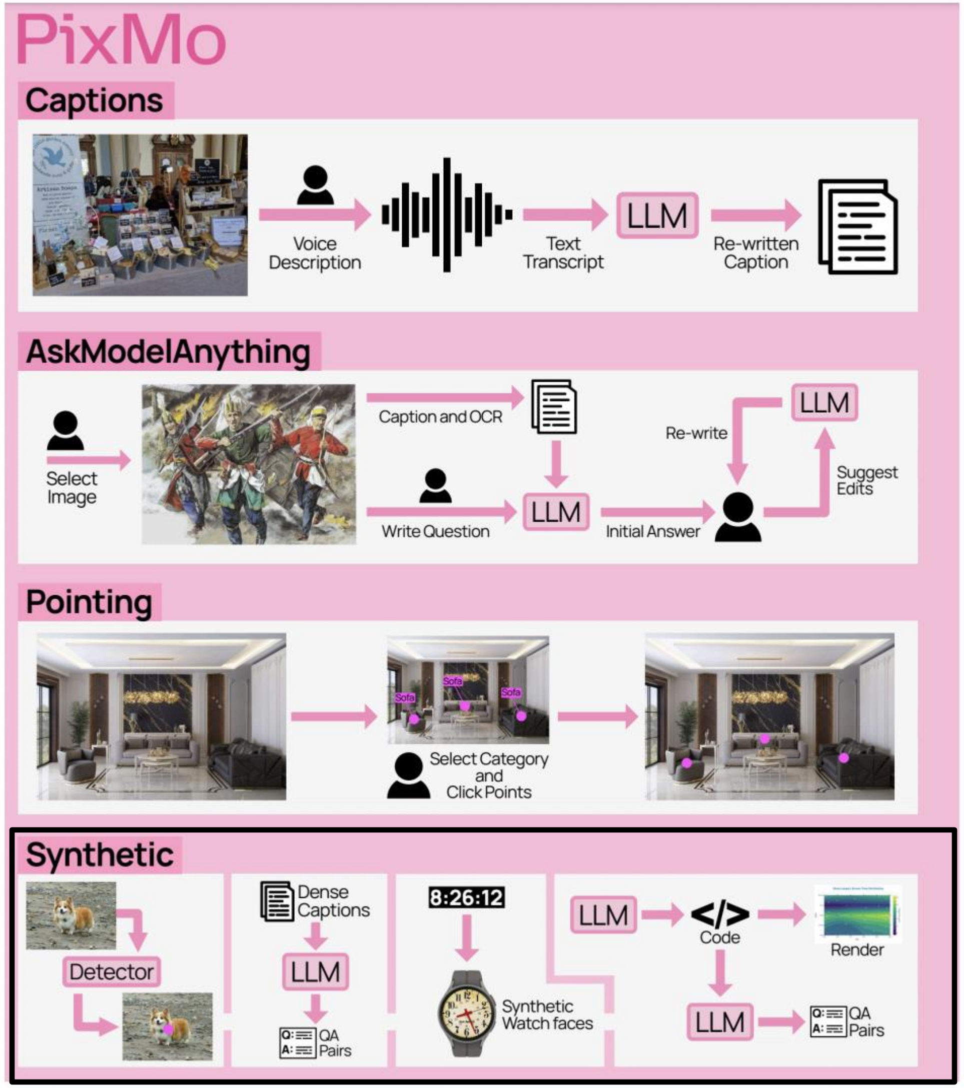
Example: MOLMO (AI2)

- Data

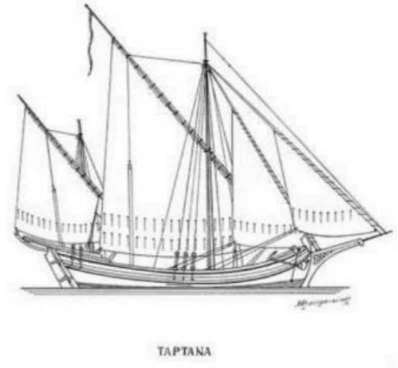


Example: MOLMO (AI2)

- Data



Example: MOLMO (AI2)



Generate a long caption about this image.

This black and white sketch depicts a boat, elegantly stationed at a pier. The boat, seemingly wooden, showcases a classic design with four prominent sails catching an imagined breeze. Below the boat, text inscribed reads "TAPTANA" in clear, uppercase letters. In the lower right-hand corner, another text, written in cursive and appearing less legible, adds an air of authenticity to the artwork. The meticulous detail in the sails and the overall structure of the boat highlights the craftsmanship and artistry involved in this sketch.

Example: MOLMO (AI2)

how many musicians.

Counting the <points x1="8.2" y1="37.6" x2="10.9" y2="45.3" x3="13.9" y3="19.4" x4="16.3" y4="11.4" x5="17.4" y5="28.8" x6="23.3" y6="35.5" x7="39.1" y7="27.3" x8="42.4" y8="23.5" x9="56.0" y9="4.8" x10="63.1" y10="28.3" x11="69.3" y11="19.6" x12="71.5" y12="29.7" x13="86.2" y13="17.1" x14="94.6" y14="28.…

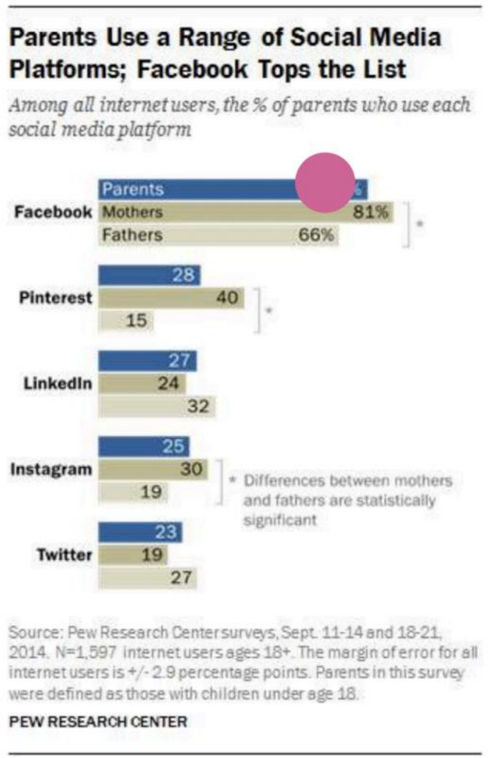
Example: MOLMO (AI2)

What kind of donut is this?

This appears to be a cinnamon sugar donut. The donut itself has a light tan color typical of a traditional fried dough. It's coated with a generous layer of cinnamon sugar, giving it that classic sweet and slightly spicy flavor profile.

The topping isn't cheese as…

Example: MOLMO (AI2)



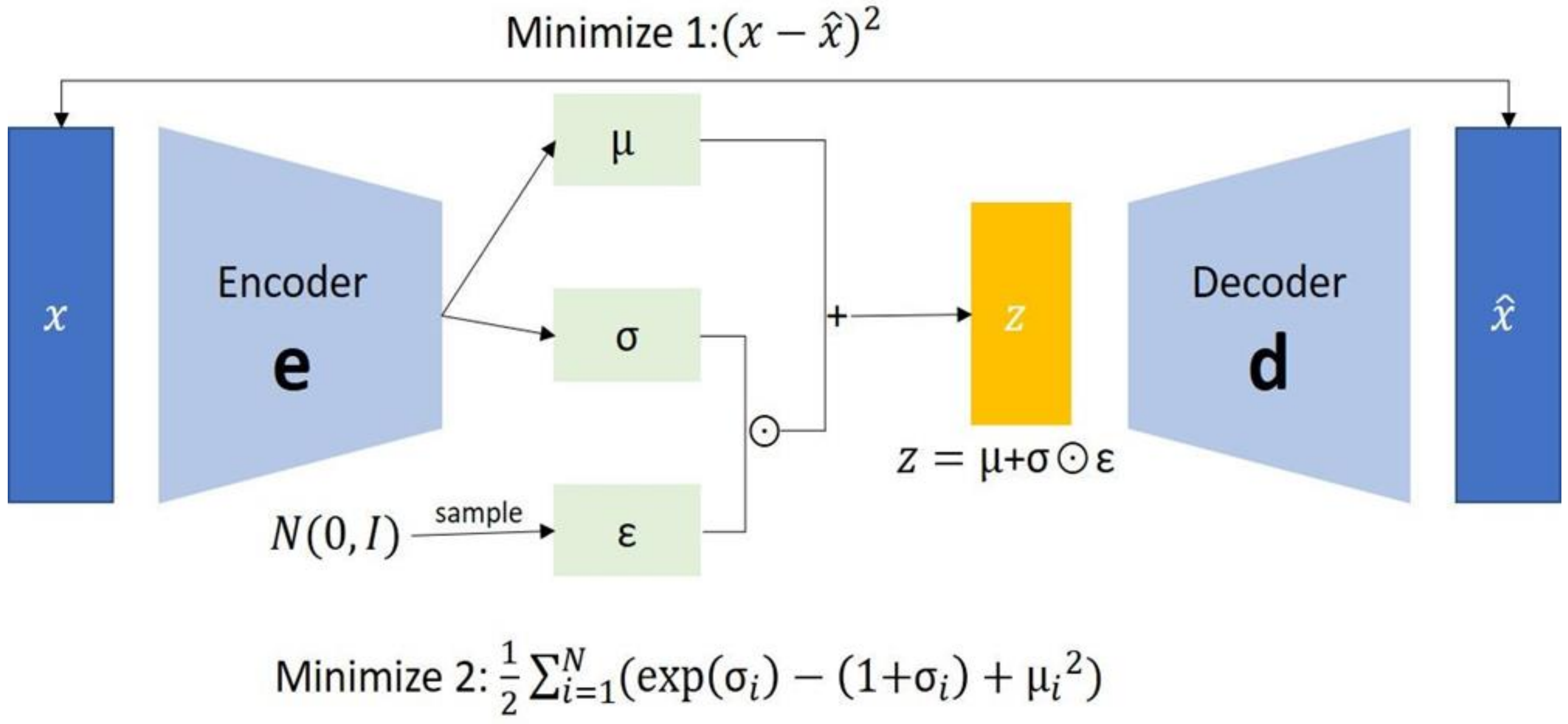
What percentage of parents use Facebook?

The image shows a breakdown of social media usage among parents. <point x="66.6" y="24.2" alt="Percentage of parents using Facebook">74 percent</point> of parents use Facebook, making it the most popular social media platform among this demographic. This statistic combines both mothers an…

A Peek at Image Generation

- VAE (Variational Autoencoder)
- VQ-VAE (Vector Quantized Variational Autoencoder)
- GANs (Generative Adversarial Networks)
- Diffusion
- Modern Image Generators

VAE



Minimize 1:$(x-\hat{x})^2$
x
Encoder
e
μ
σ
ε
$N(0,I)$
sample
+
⊙
z
$z = \mu + \sigma \odot \varepsilon$
Decoder
d
$\hat{x}$
Minimize 2: $\frac{1}{2}\sum_{i=1}^{N}(\exp(\sigma_i) - (1+\sigma_i) + \mu_i^2)$

VQ-VAE

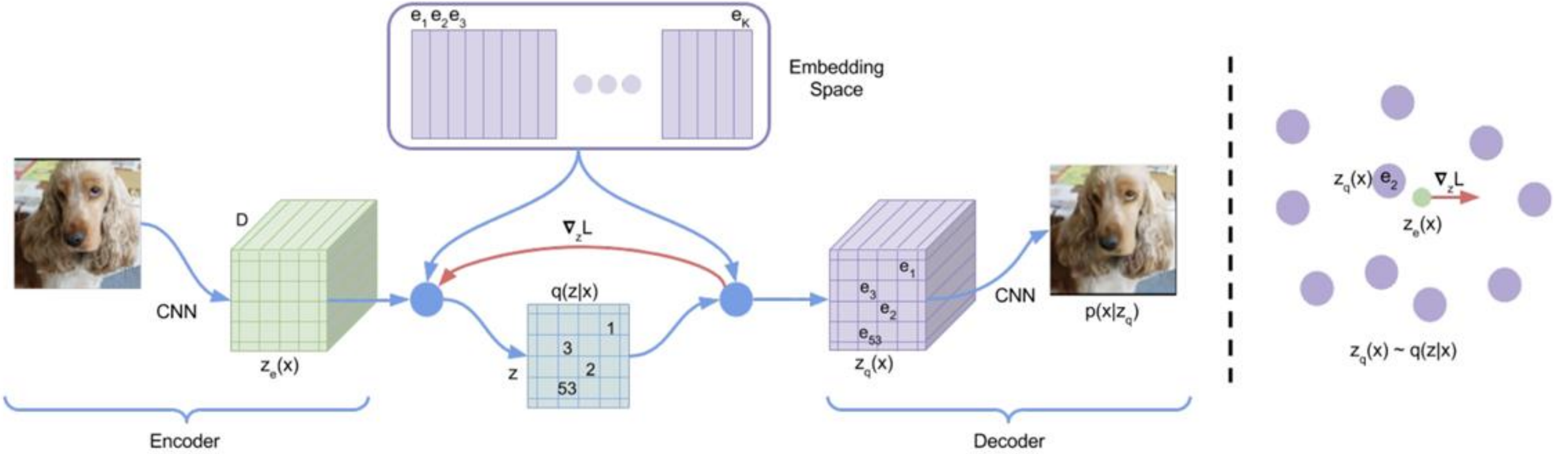
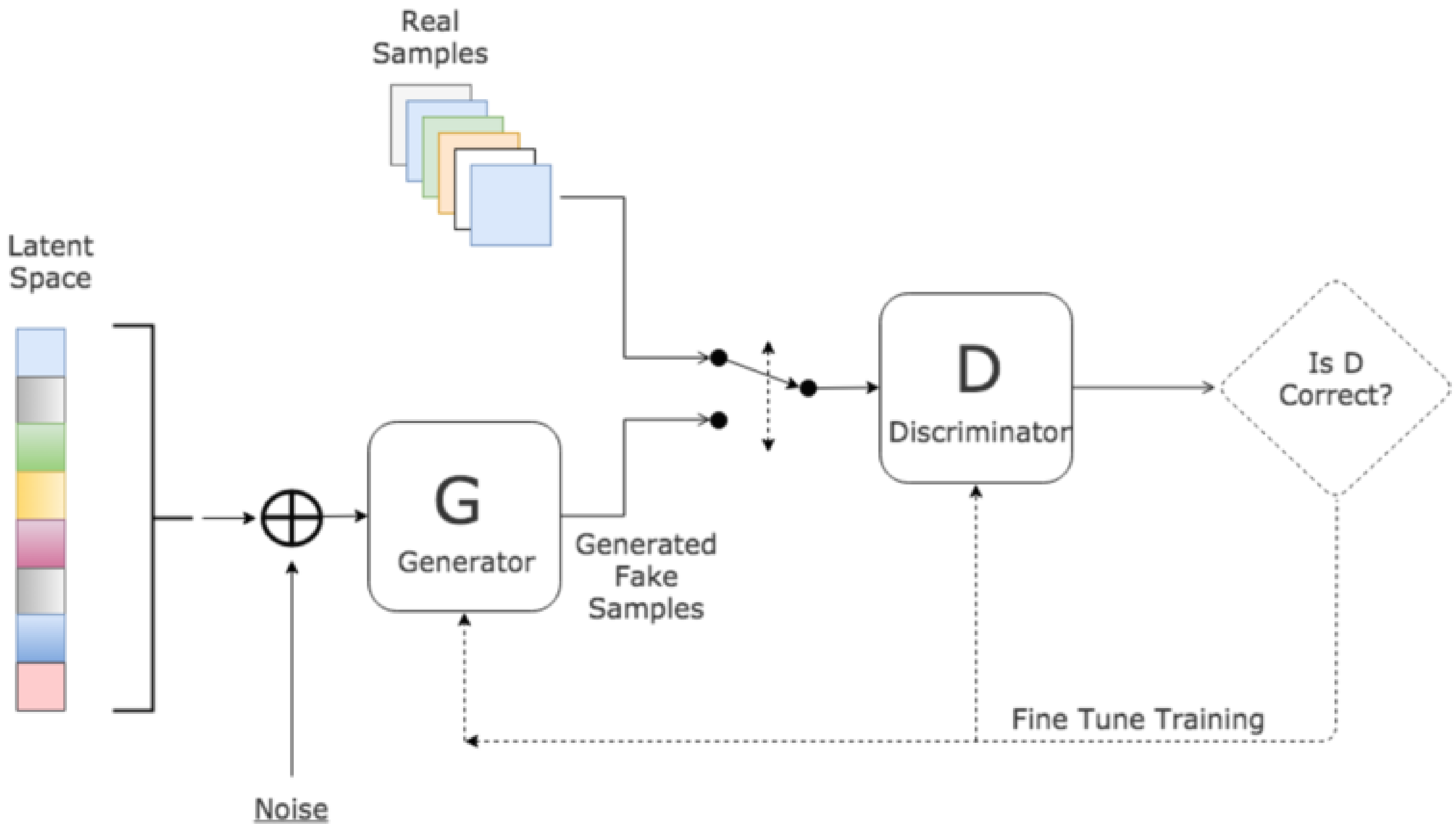
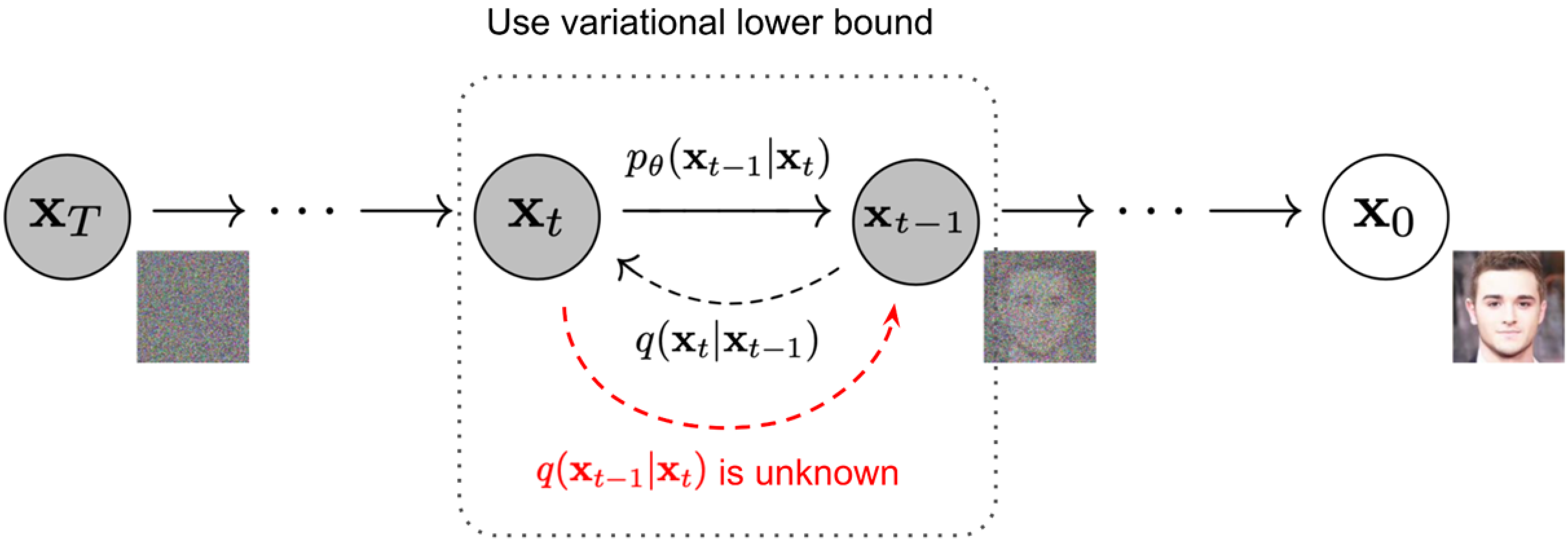


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

GAN

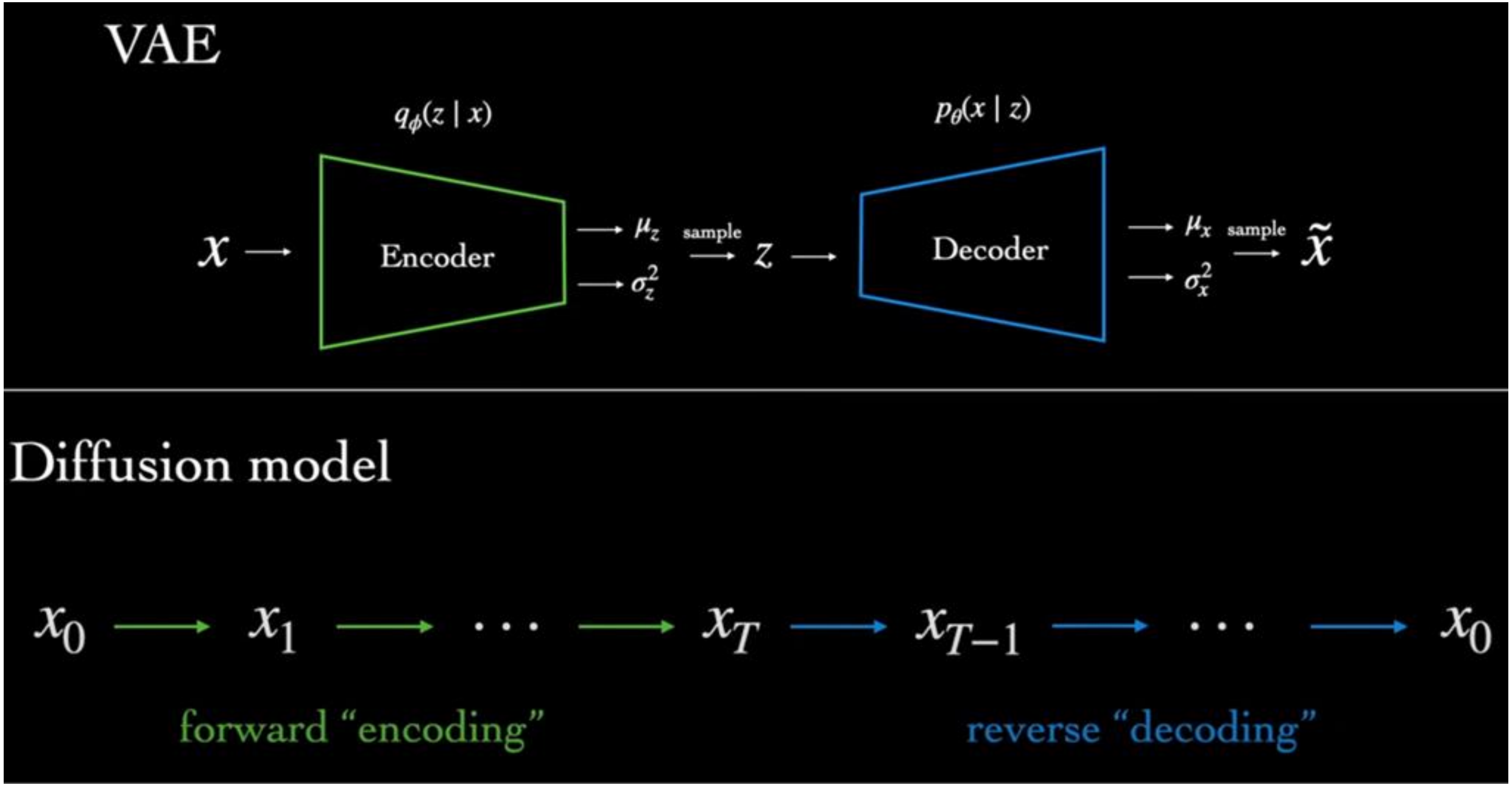

Real
Samples
Latent
Space
G
Generator
Generated
Fake
Samples
D
Discriminator
Is D
Correct?
Fine Tune Training
Noise

Diffusion

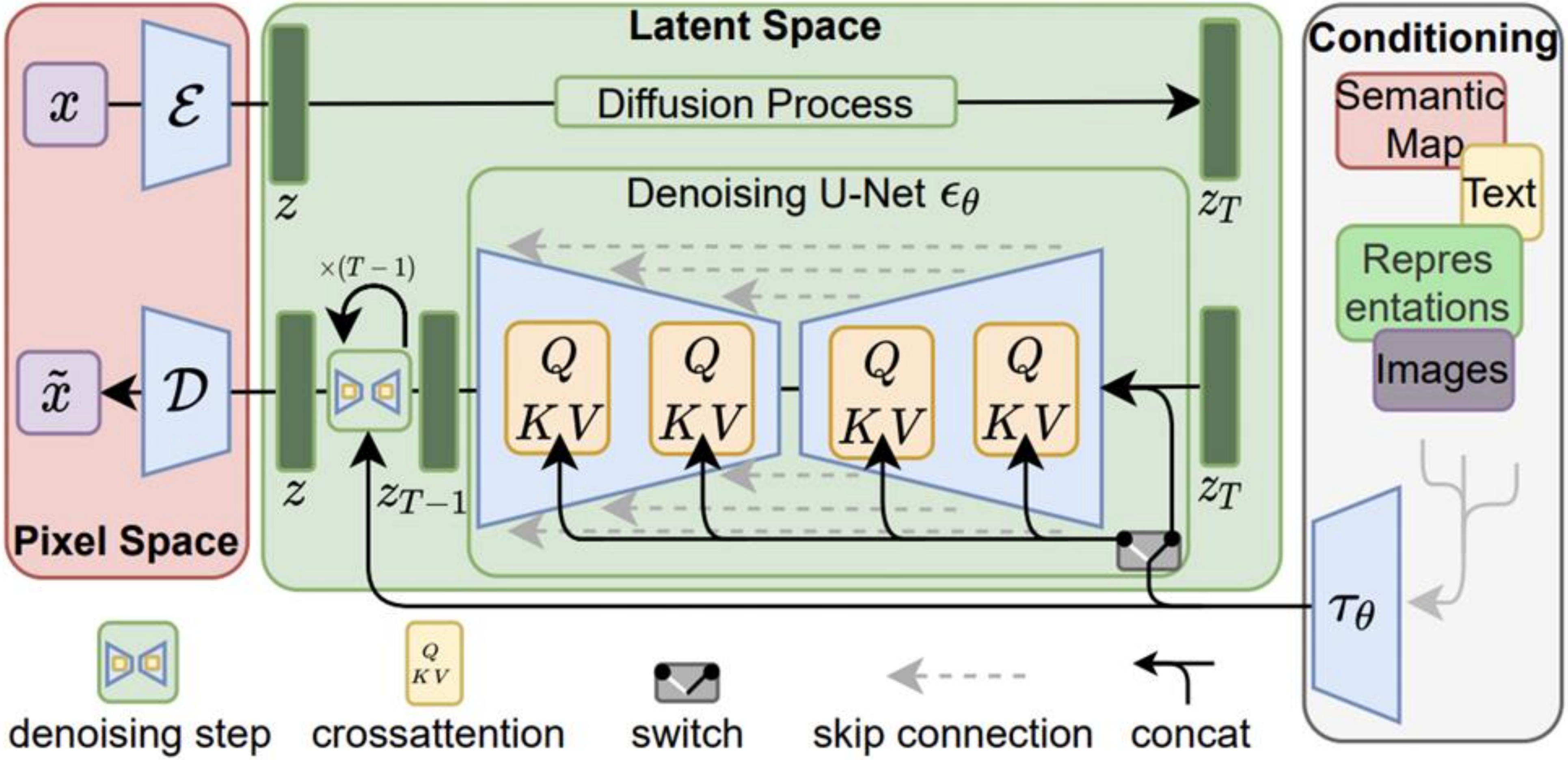


Use variational lower bound
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

Diffusion


VAE
$q_\phi(z \mid x)$
$p_\theta(x \mid z)$
x
Encoder
μ_z
σ_z^2
sample
z
Decoder
μ_x
σ_x^2
sample
$\tilde{x}$
Diffusion model
$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_T \longrightarrow x_{T-1} \longrightarrow \cdots \longrightarrow x_0$
forward "encoding"
reverse "decoding"

Modern Image Generators (Stable Diffusion)



Modern Image Generators (Llama3)