AI504: Programming for Artificial Intelligence

Week 15: Image-Text Multimodal Learning

Edward Choi
Grad School of AI
edwardchoi@kaist.ac.kr

Last Year’s Topic

- Image-to-text (a.k.a Image captioning)
  - Show and Tell
  - Show, Attend and Tell
- Text-to-Image
  - Text-conditioned GAN
  - DALL-E
- Image-text pretraining
  - BERT-based models

Today's Topic

- Image-to-text (a.k.a Image captioning)
  - ~~Show and Tell~~
  - Show, Attend and Tell
- Text-to-Image
  - ~~Text-conditioned GAN~~
  - ~~DALL-E~~
  - CLIP
  - DALL-E 2
- Image-text pretraining
  - BERT-based model~~s~~
- Vision LLM
  - LLaMA
  - Alpaca
  - LLaVA

Image Captioning

Image-to-Text

- Sequence to sequence
  - Text in, text out
  - e.g. Translate French to English
- Image to sequence
  - Image in, text out
  - e.g. Describe a given image in text (i.e. Image Captioning)

Image Captioning

A person riding a motorcycle on a dirt road.

A group of young people playing a game of frisbee.

A herd of elephants walking across a dry grass field.

Encoder-Decoder Architecture

- Sequence to sequence
  - Encoder: RNN
  - Decoder: RNN
- Image to sequence
  - Encoder: CNN
  - Decoder: RNN

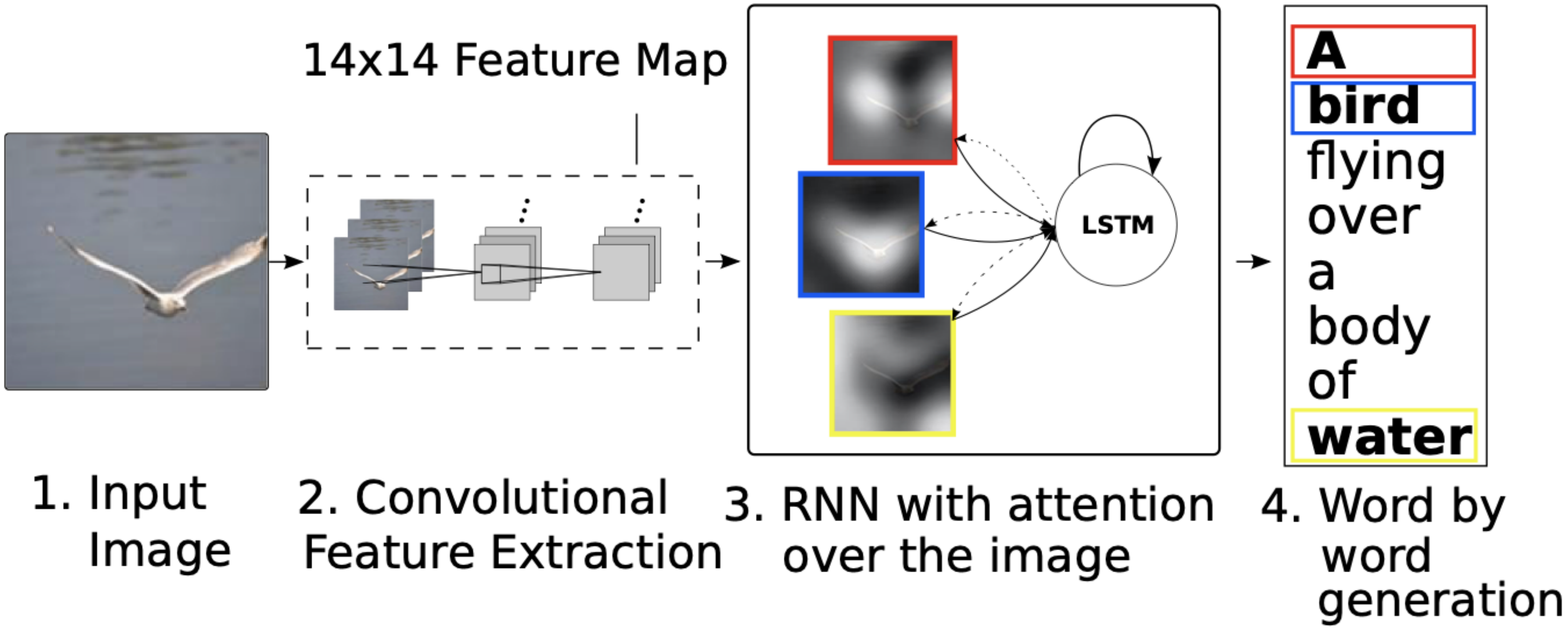
Show, Attend and Tell

Show, Attend and Tell

- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
  - Xu et al. ICML 2015
- Mixing attention mechanism with image captioning

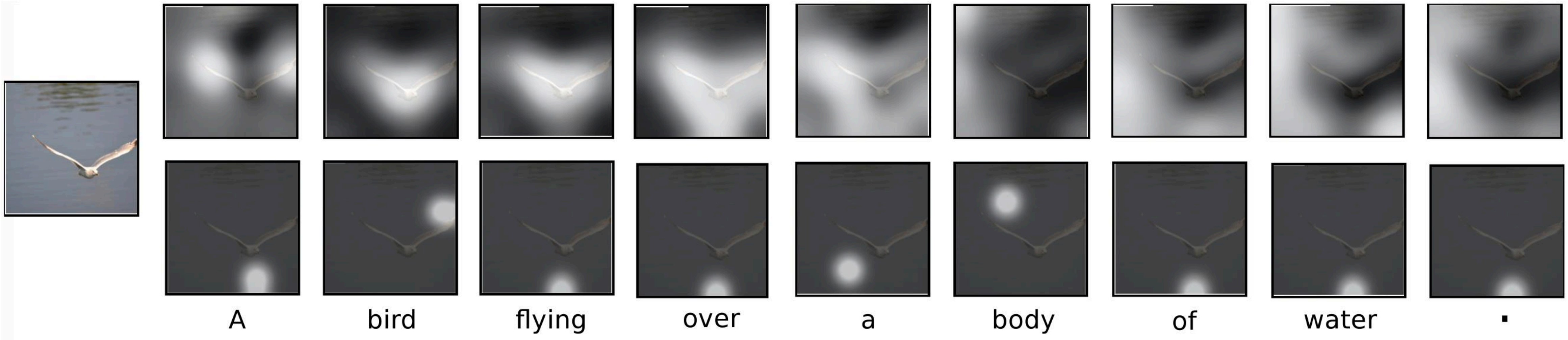
Show, Attend and Tell

- High-level architecture



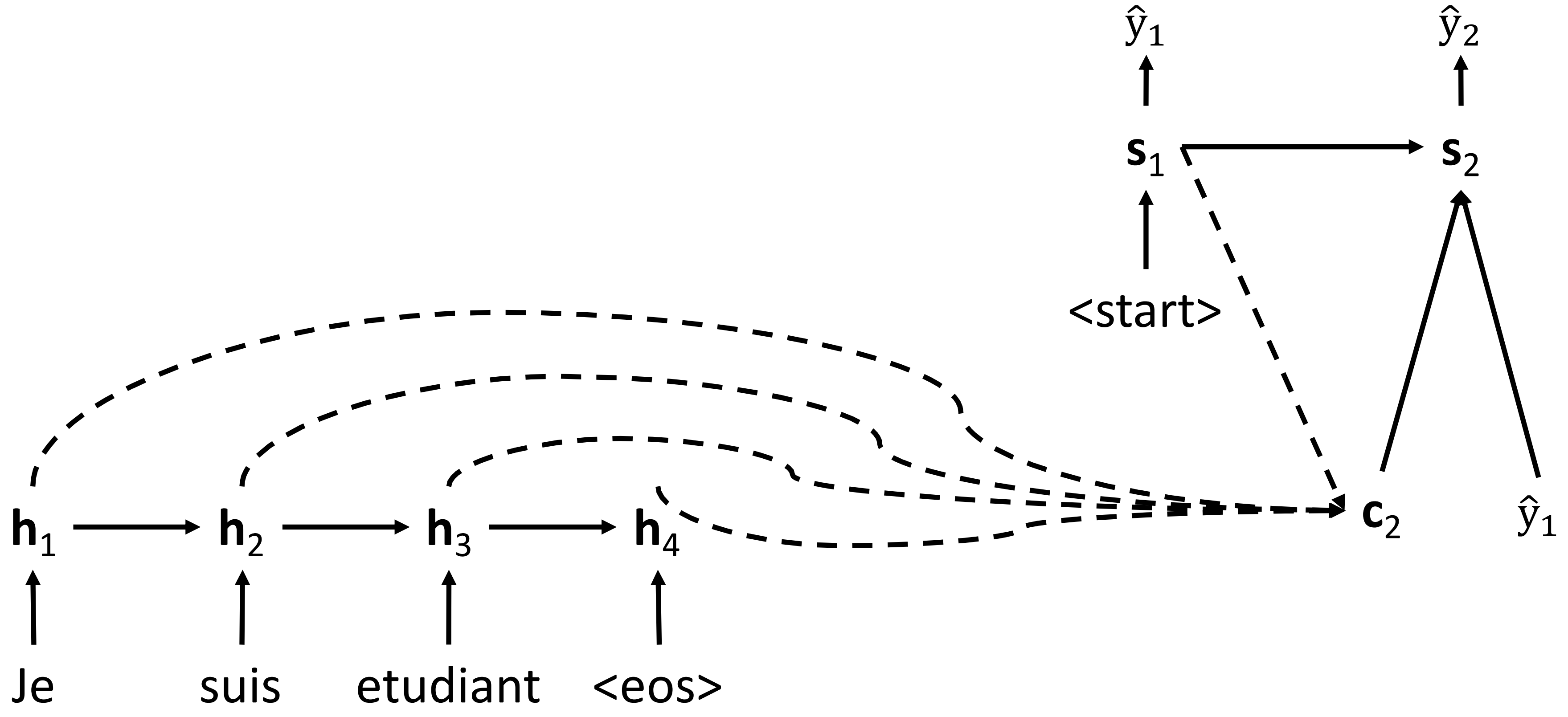
Show, Attend and Tell

- Example: “A bird flying over a body of water .”
  - Top row is “soft” attention, bottom row is “hard” attention.
- Model is “attending” to relevant part of image when generating word



Encoder-Decoder Architecture

- Seq2seq with attention



Encoder-Decoder Architecture

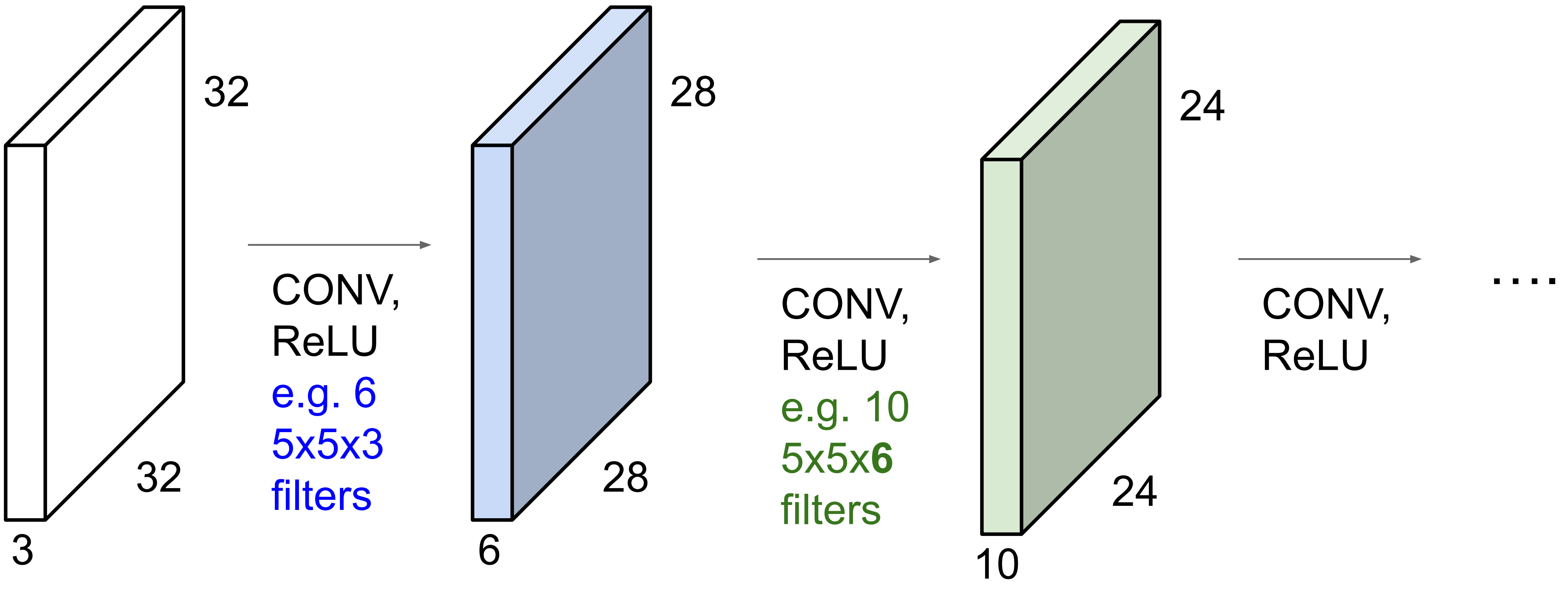
- **What we need:**
- Encoder to obtain image representation
- Decoder to generate caption
- Attention module to calculate attention weights

Encoder-Decoder Architecture

- **What we need:**
- Encoder to obtain image representation
  - Oxford VGGnet
- Decoder to generate caption
  - LSTM
- Attention module to calculate attention weights
  - MLP

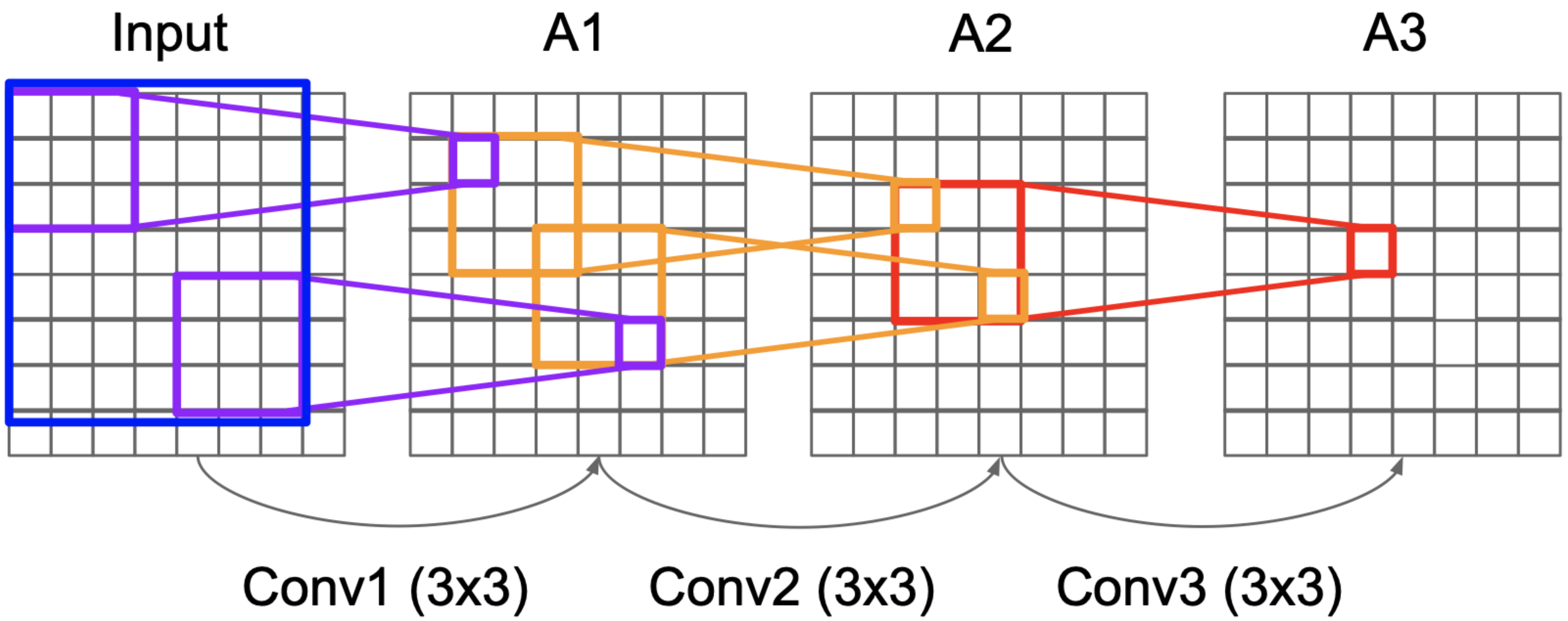
How to Attend to Part of Image

- Remember Convolution?



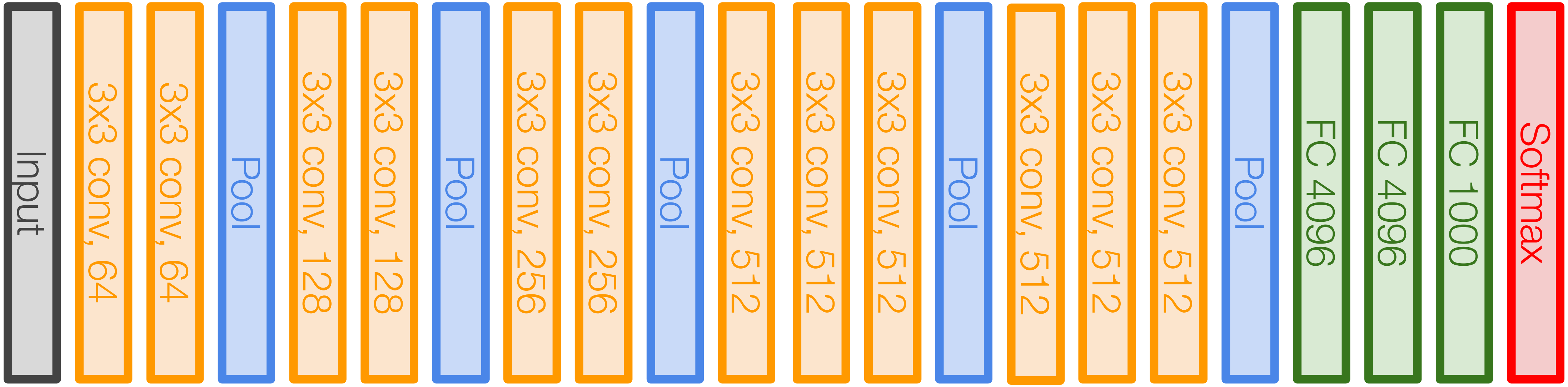
How to Attend to Part of Image

- Remember receptive field?



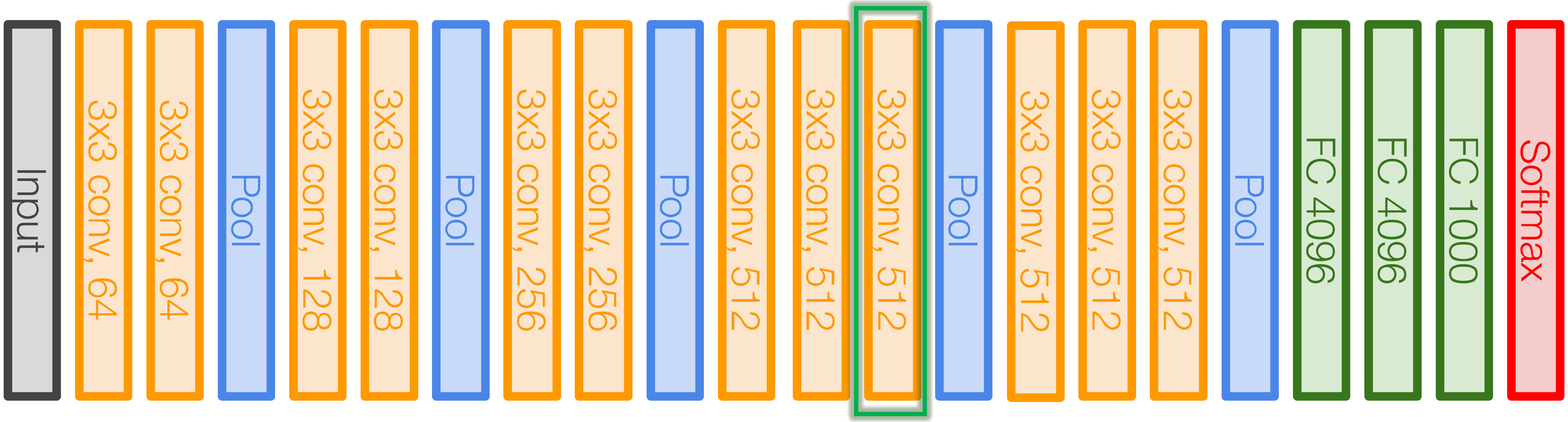
How to Attend to Part of Image

- Remember VGG 16?



How to Attend to Part of Image

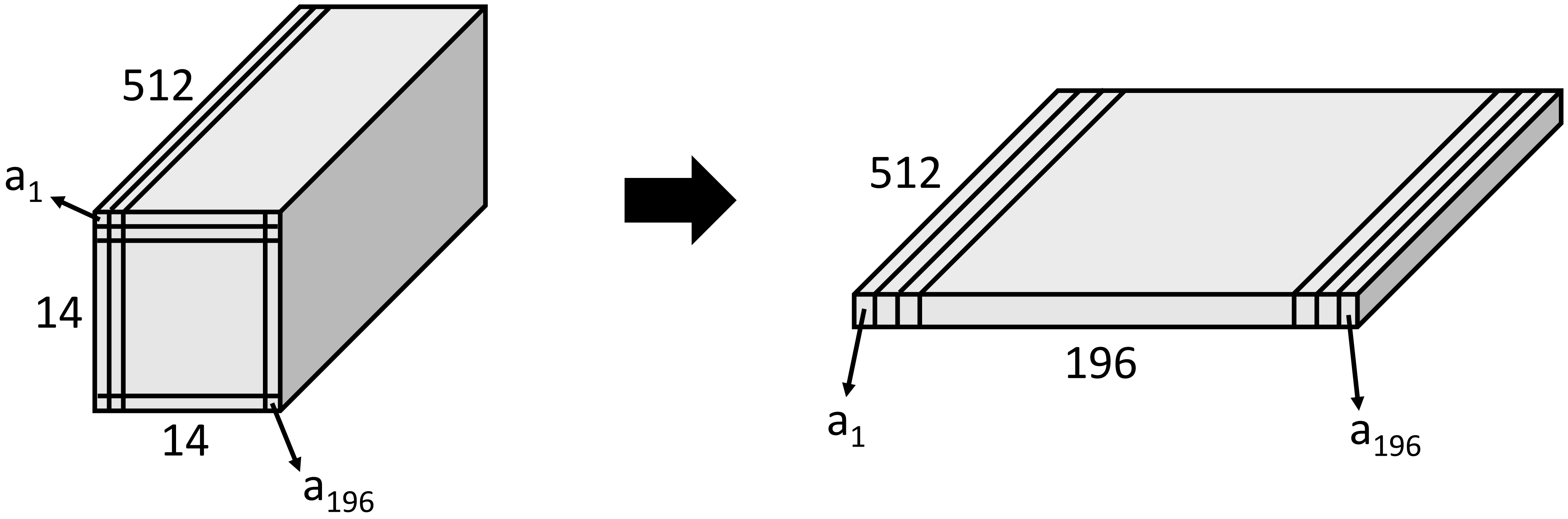
- Remember VGG 16?



Output of this convolution layer:
14 x 14 x 512 feature map
➔ 196 x 512 image representation vector

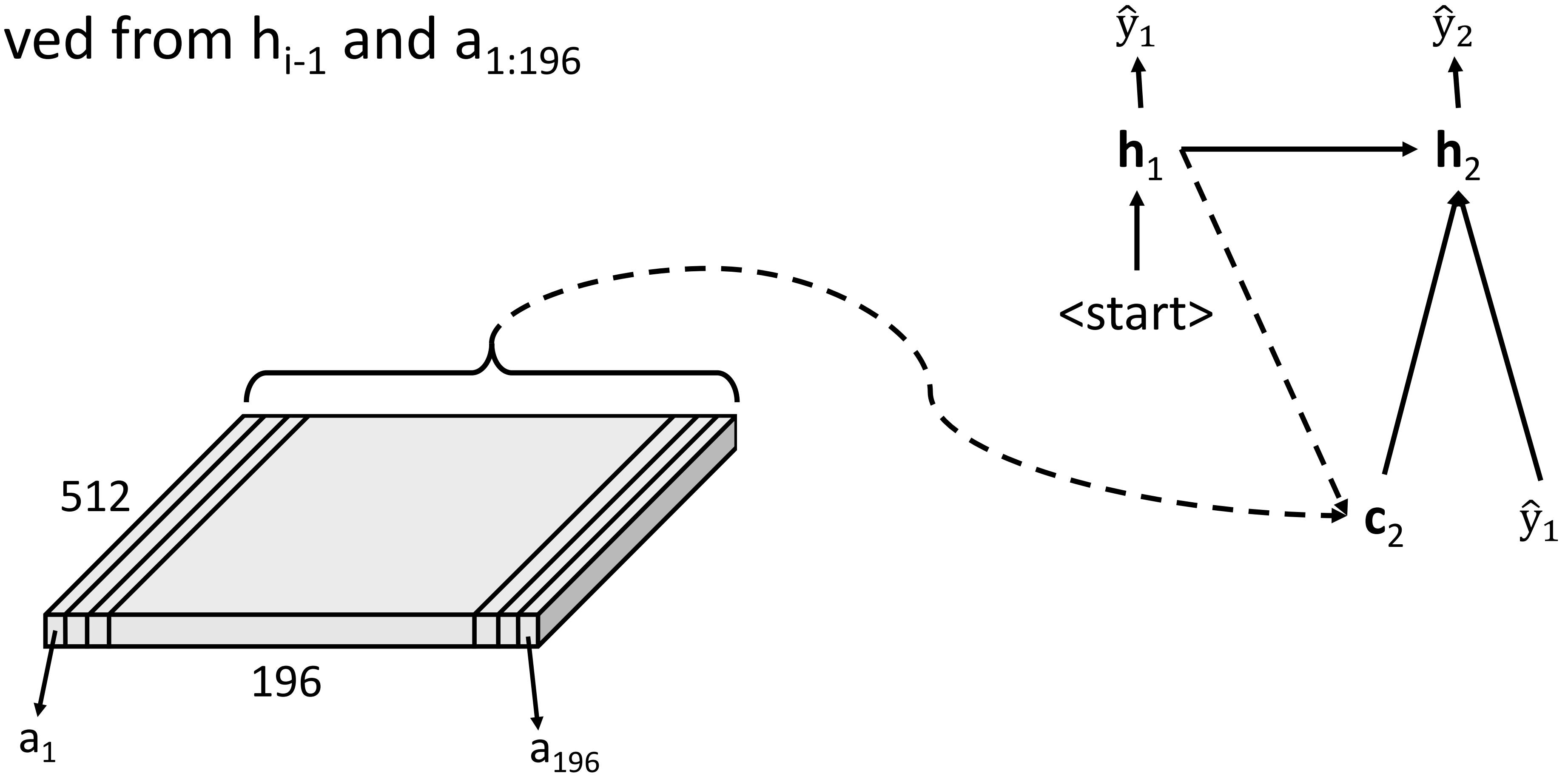
Model Architecture

- Flattening the image feature maps



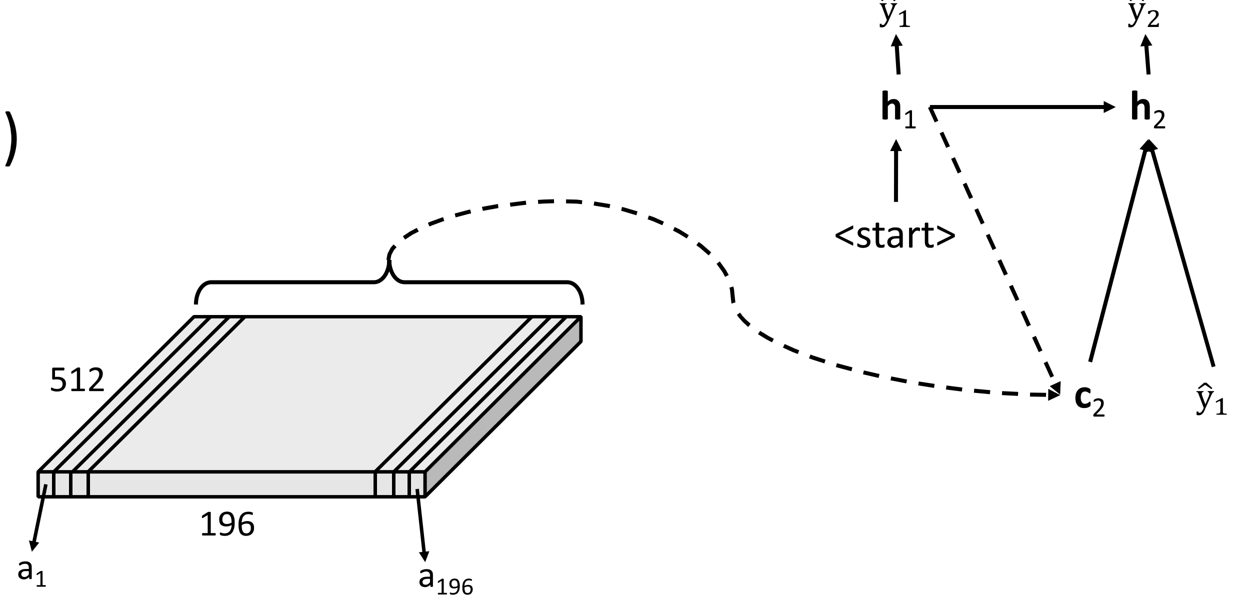
Show, Attend and Tell

- Each y_i is predicted based on h_i
- Each h_i is derived based on h_{i-1}, y_{i-1}, c_i
- c_i is derived from h_{i-1} and $a_{1:196}$



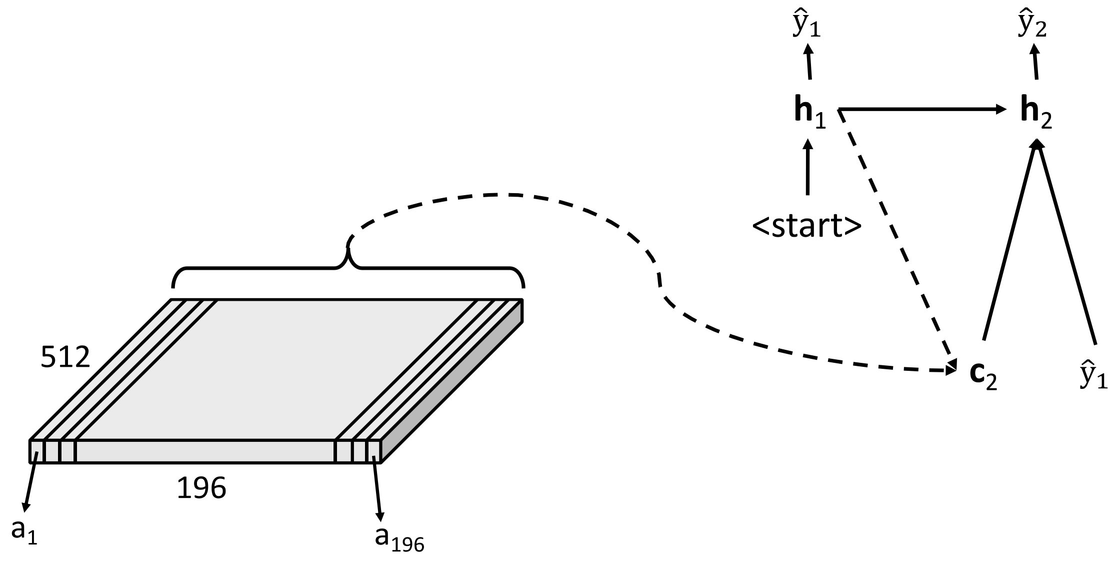
Show, Attend and Tell

- Each y_i is predicted based on h_i
  - $\hat{y}_1$ = Softmax($W_w h_i$ + b)
- Each h_i is derived based on h_{i-1}, y_{i-1}, c_i
  - h_i = RNN(h_{i-1}, $[y_{i-1};c_i]_{concat}$)
- c_i is derived from h_{i-1} and $a_{1:196}$
  - c_i = sum(α_i*a_i)
  - α_i = Softmax(f(h_{i-1}, a_1), … , f(h_{i-1}, a_{196}))
  - f(h_{i-1}, a_j) = $h_{i-1}^T W_f a_j$



Show, Attend and Tell

- **Some technical details**
- RNN's initial hidden state is learned
  - h_0 = MLP$\left(\frac{1}{L}\sum_{i=1}^{L} a_{1:L}\right)$
- Authors also tried "hard" attention.
  - Stochastically select only one a_i at each step.
  - Use reinforcement learning to train.
- Encourage $\sum_t \alpha_{ti} \approx 1$
  - Make the model pay equal attention to every part of image during text generation.

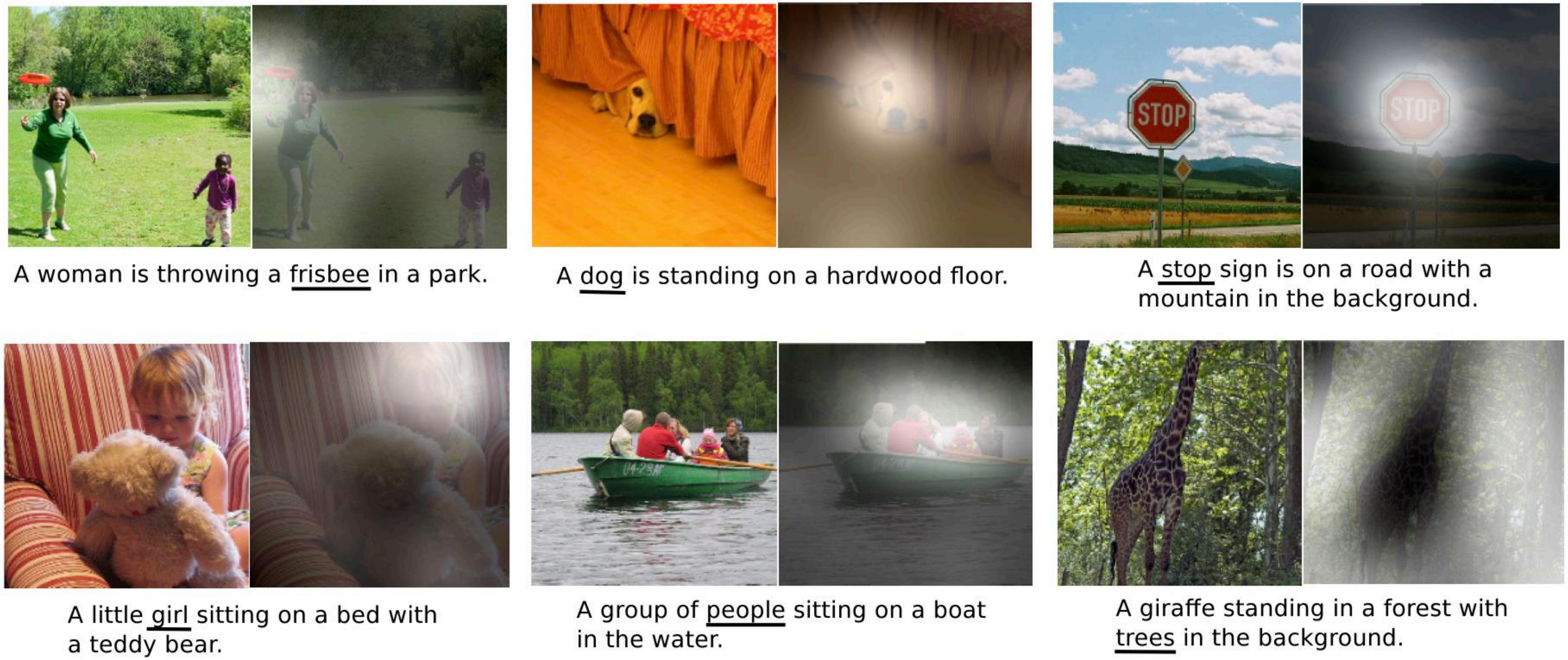


Model Performance

Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

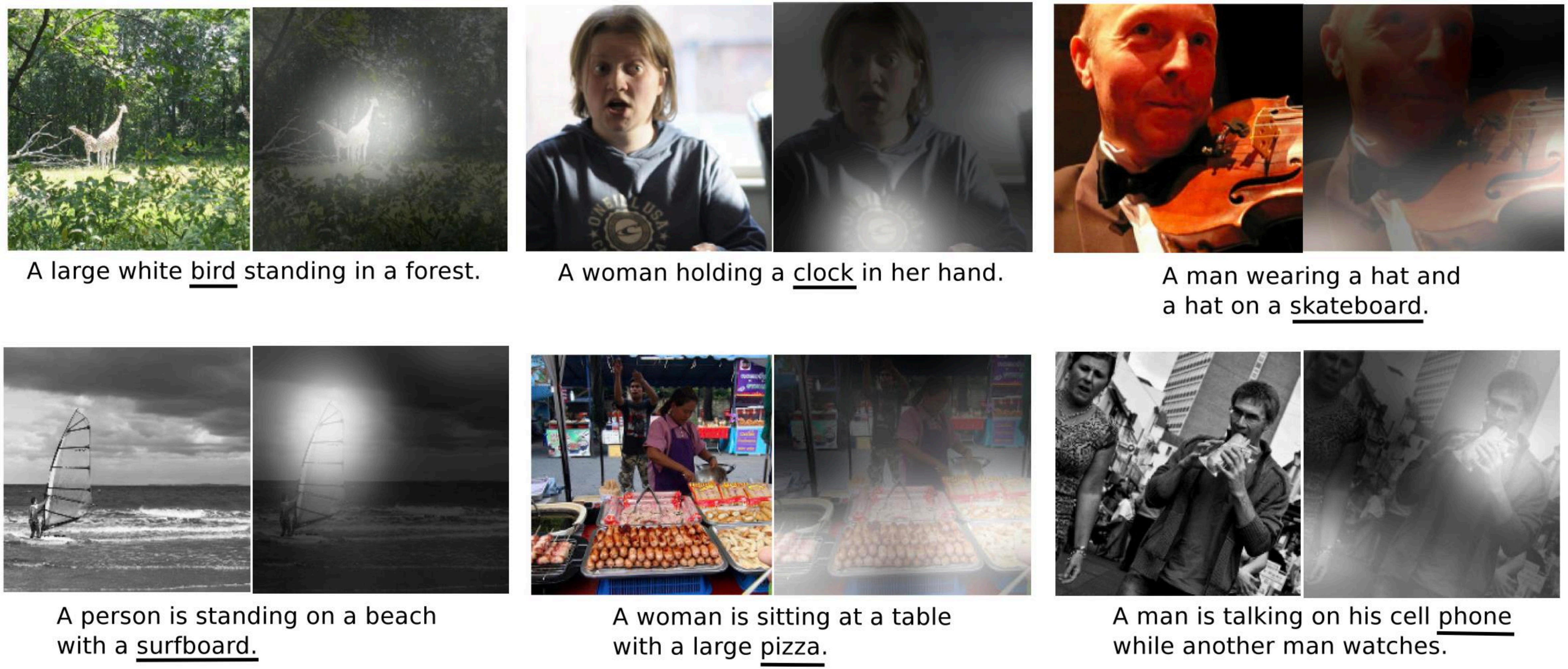
Correction Attention Examples

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



Incorrect Attention Examples

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

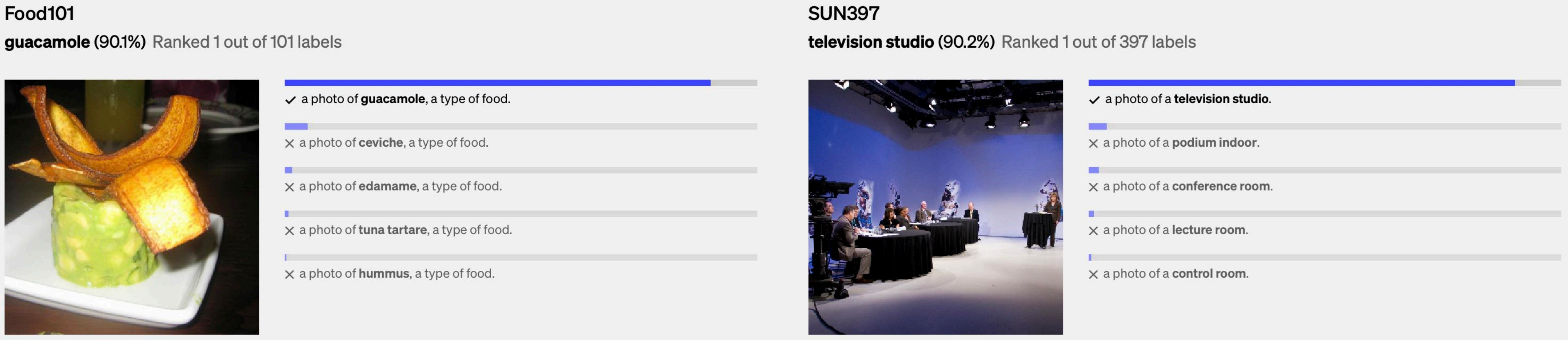


A woman is throwing a frisbee in a park.

Text-to-Image

CLIP

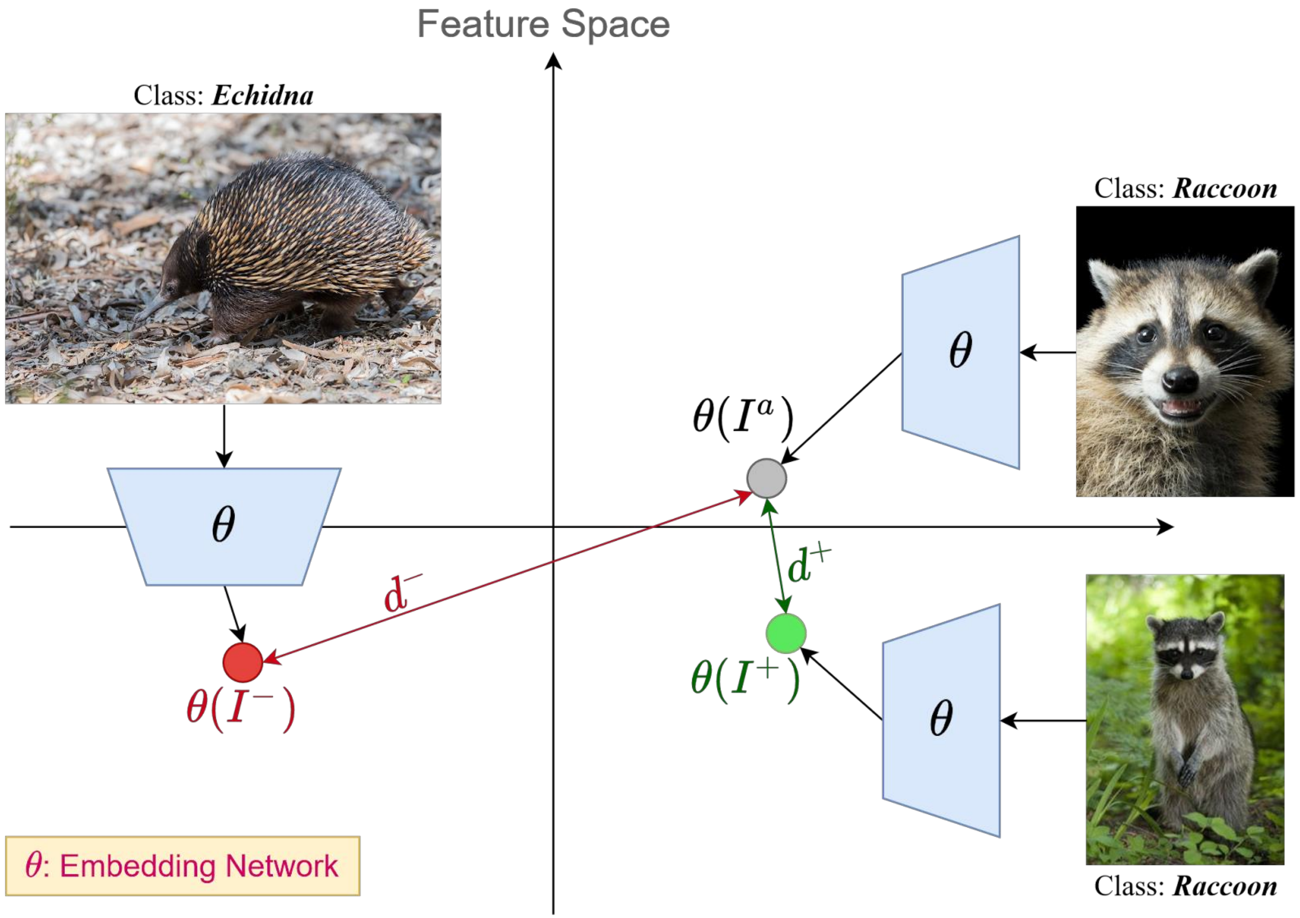
- Learning Transferable Visual Models From Natural Language Supervision
  - Radford, Kim et al. 2021 (OpenAI)
  - Contrastive learning between text and image
  - Great zero-shot performance
  - Understands the relationship between text and image very well



CLIP's zero-shot prediction examples

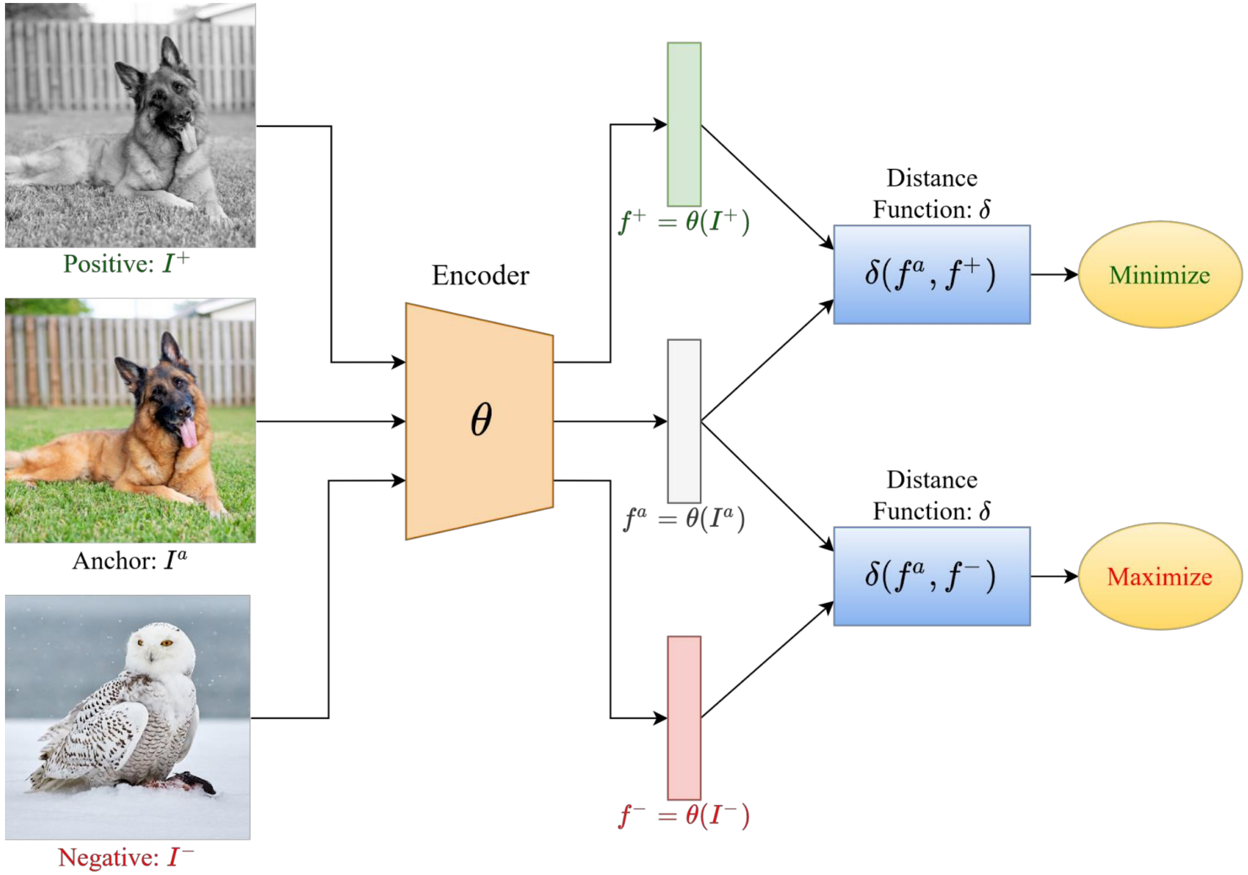
Contrastive Learning

- Focuses on “postive pairs” and “negative pairs”
  - Positive pairs
    - Should be closer to each other
  - Negative pairs
    - Should be far apart
- Learn an embedding space that respects such property



Contrastive Learning

- Training



Anchor

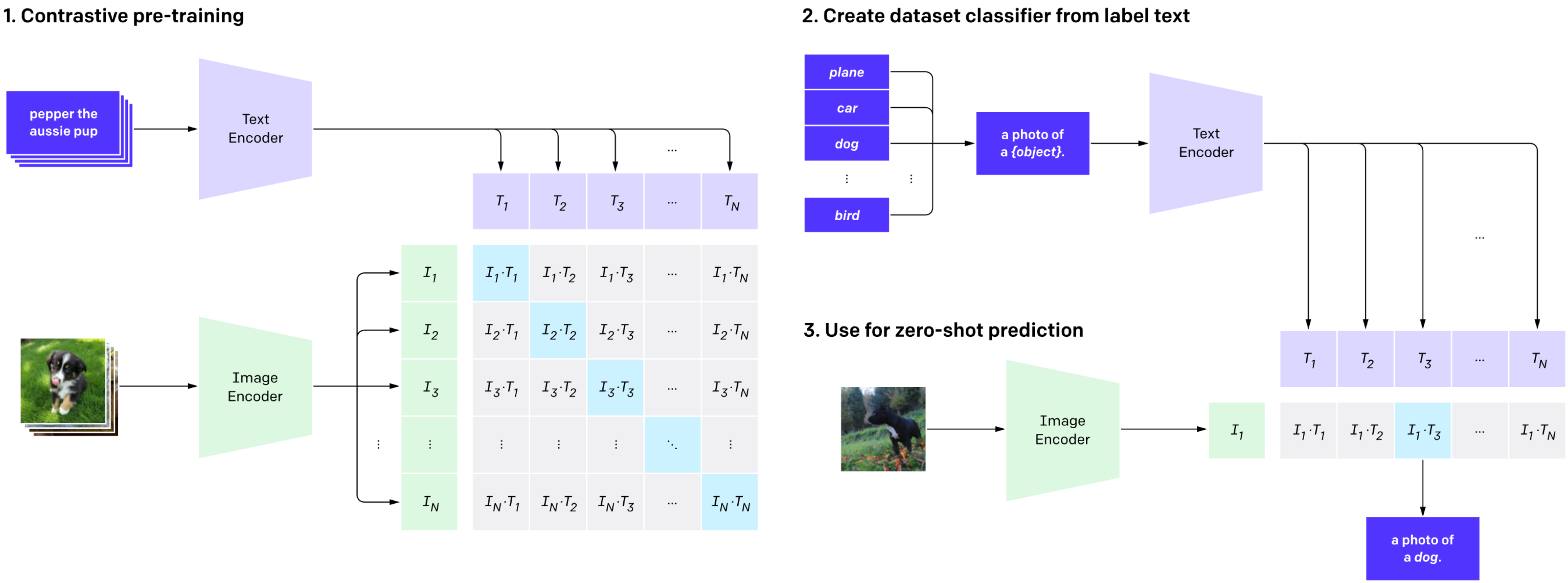
Positive

$$\textit{InfoNCE Loss} = -\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{k=0}^{N} \exp(\mathbf{z}_i \cdot \mathbf{z}_k / \tau)}$$

Other samples

CLIP Training

- Use cosine similarity
  - Not inner product, L2 distance
- Text encoder is transformer (e.g. BERT)
- Image encoder is Vision Transformer



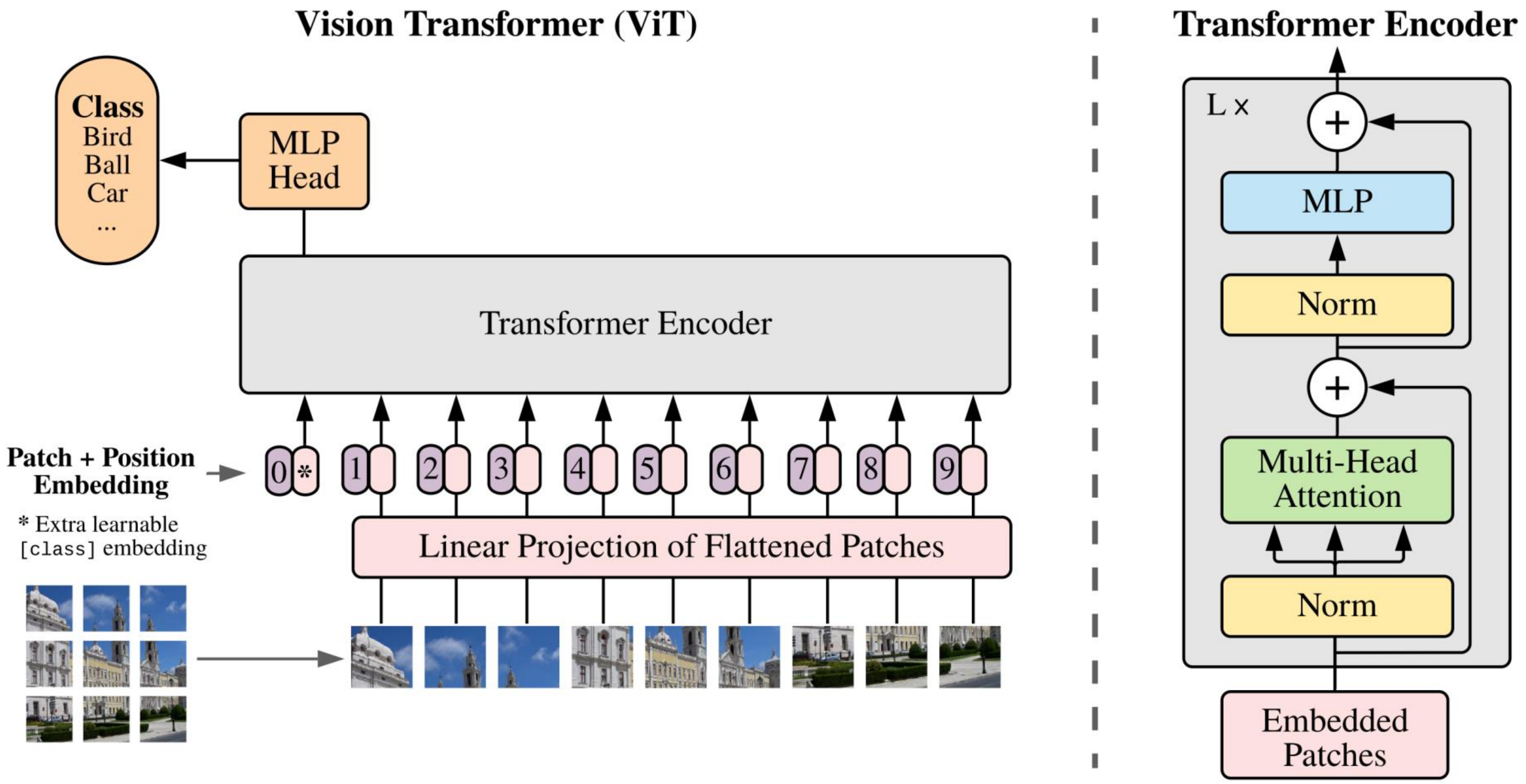
Vision Transformer

- “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”
  - Google Research, 2021

- Image classification via pure Transformer
  - No CNN
  - Comparable performance to the most powerful CNN

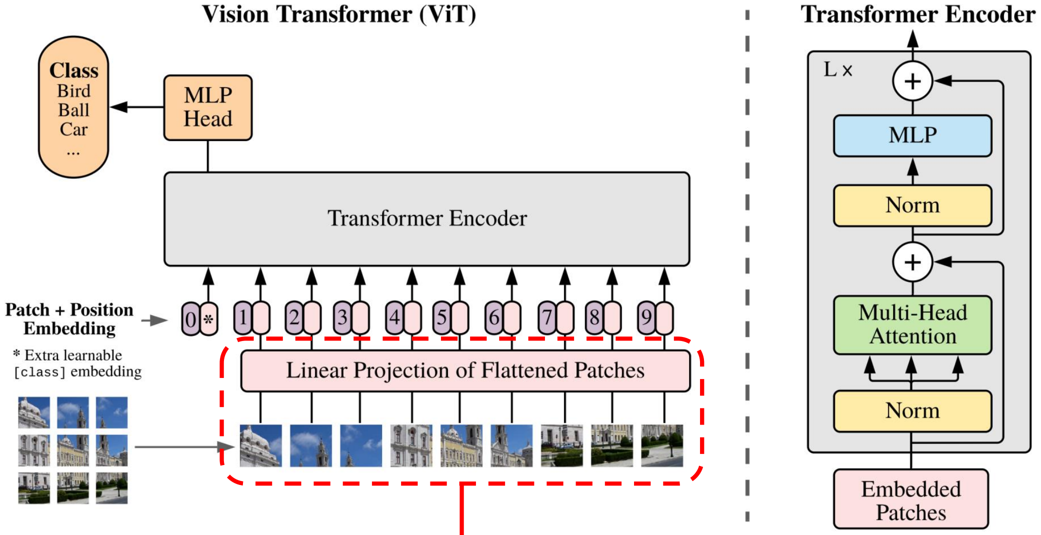
Vision Transformer

- "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"
  - Google Research, 2021



Vision Transformer

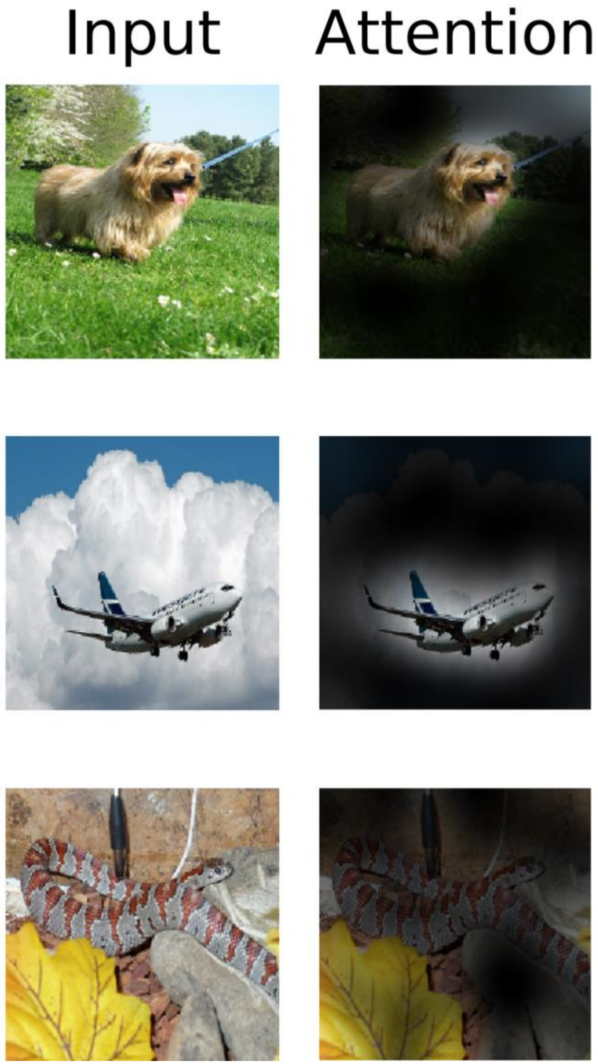
- "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"
  - Google Research, 2021



- Total 632M parameters
- 32 layers, 1280 hidden size, 16 attention heads

1. Cut an image in grids
2. Linearly transform each patch
3. Add position embeddings and feed into the Transformer encoder

ViT Attention Visualization



DALL-E 2

- Hierarchical Text-Conditional Image Generation with CLIP Latents
  - Ramesh et al. 2022 (OpenAI)
  - Text-to-image generation using CLIP priors and classifier-free guided diffusion
  - Two-step upsampling (also diffusion)

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

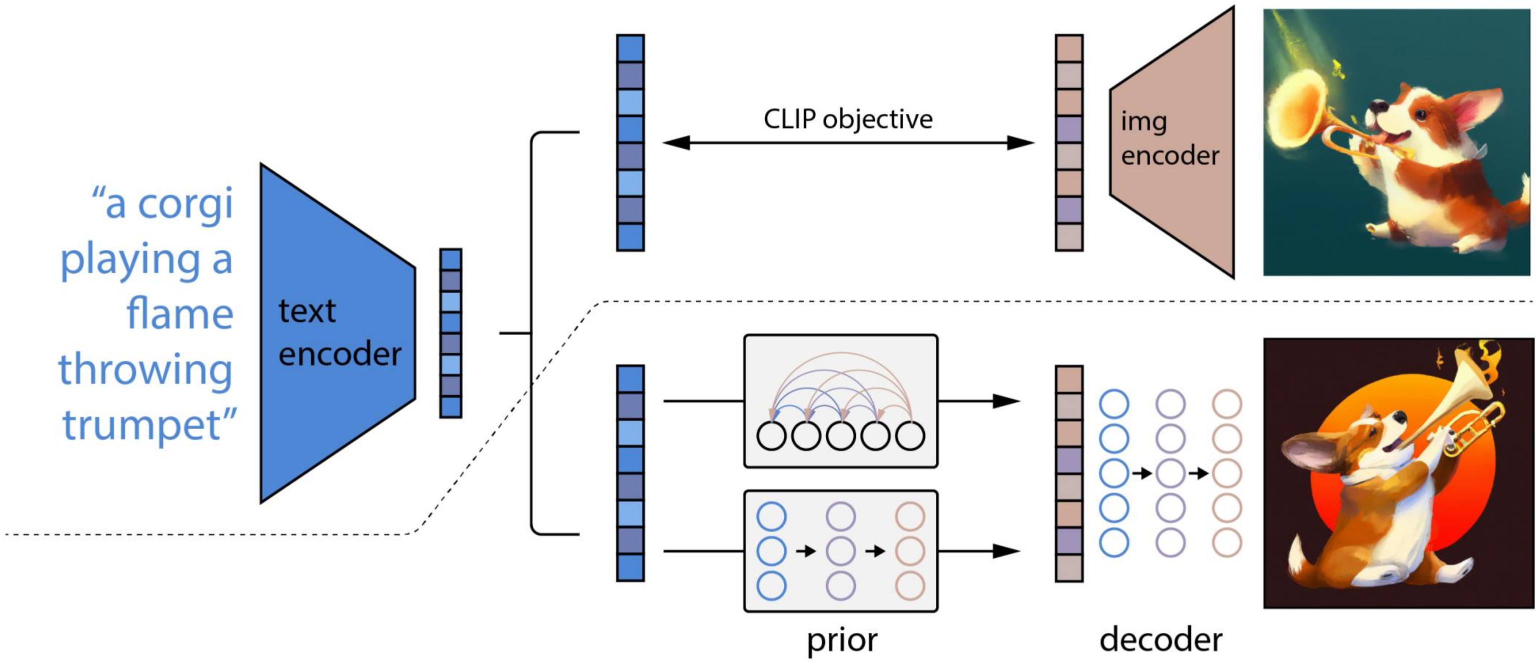
a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddy bear on a skateboard in times square

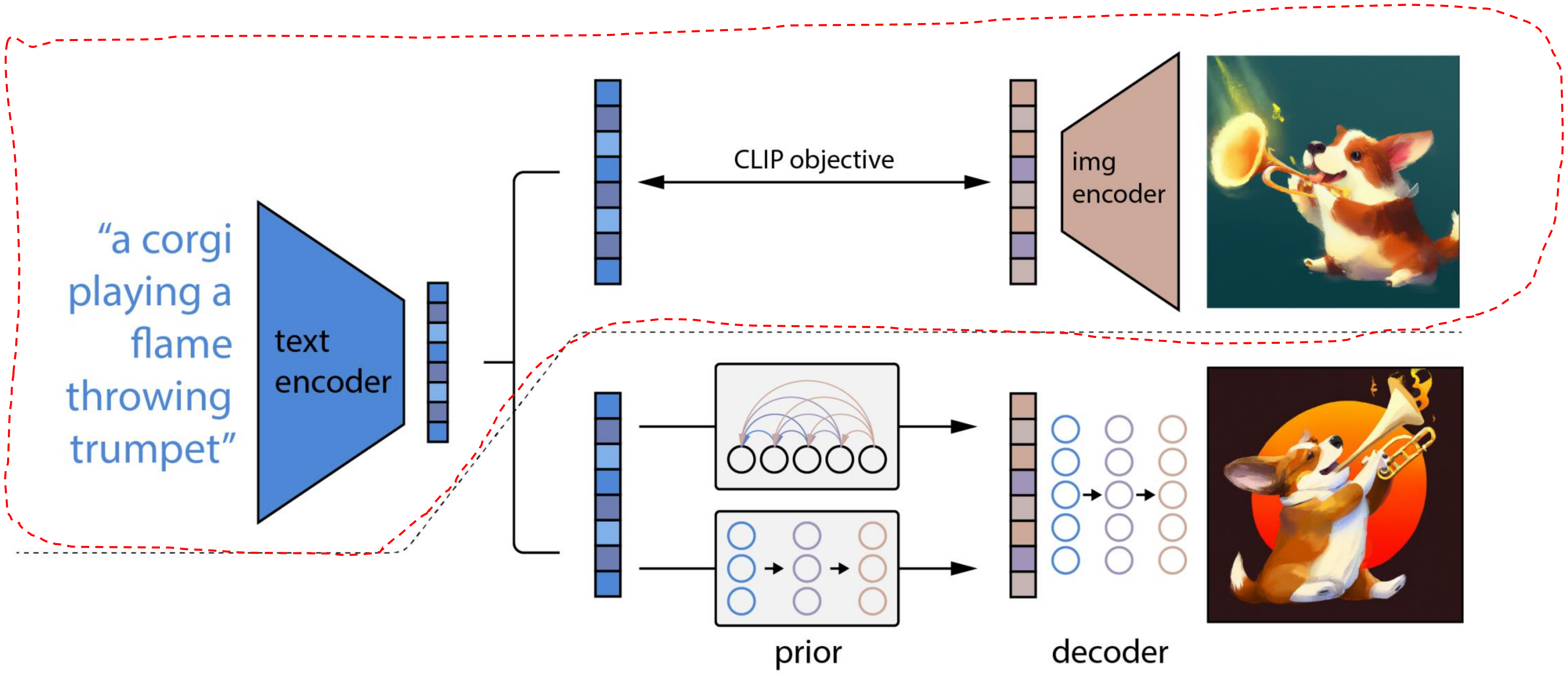
DALL-E 2

- Step-by-step
  1. Input: Text input — Output: CLIP text embedding
  2. Input: CLIP text embedding — Output: CLIP image embedding
  3. Input: CLIP image embedding — Output: Raw image (64x64)
  4. Input: Raw image (64x64) — Output: Raw image (256x256)
  5. Input: Raw image (256x256) — Output: Raw image (1024x1024)



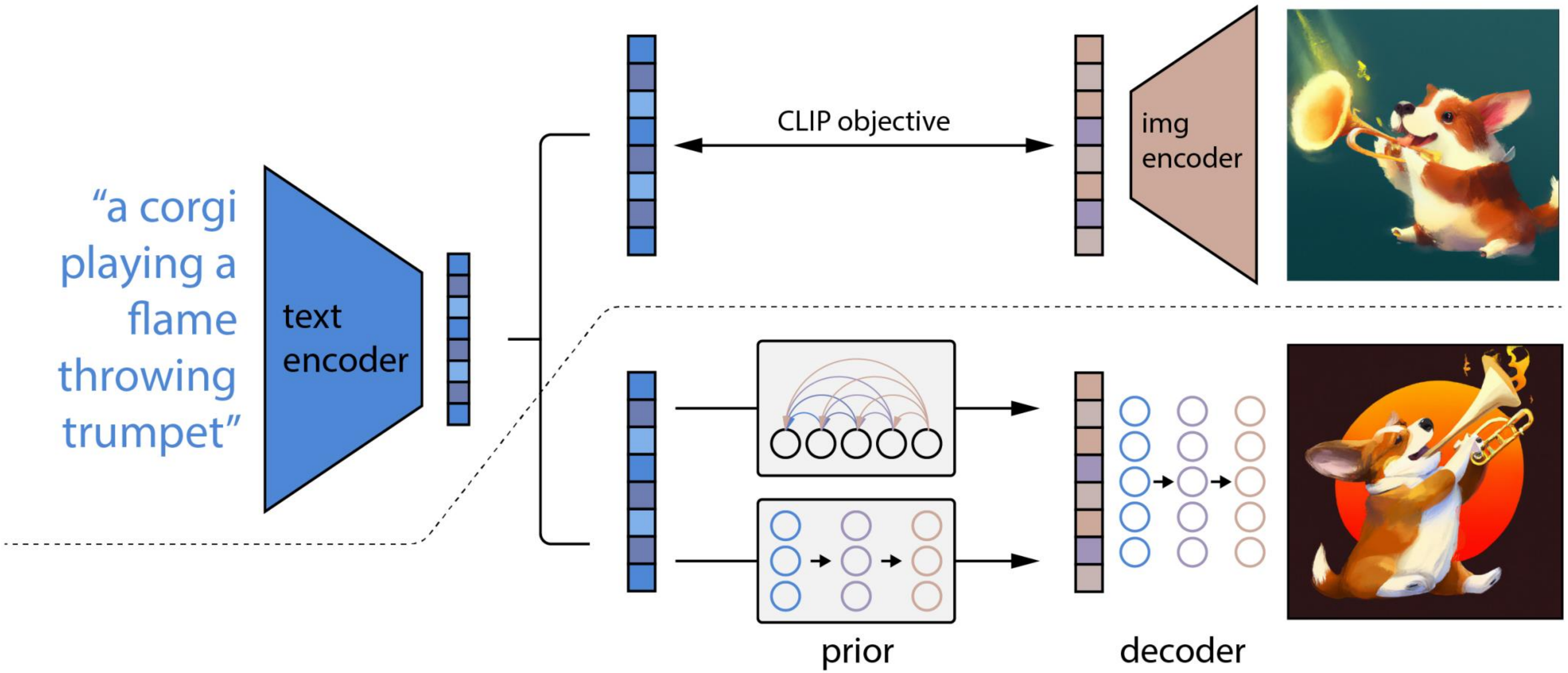
DALL-E 2

- Import a pre-trained CLIP
  - Comes with an image encoder (ViT) & a text encoder (BERT)



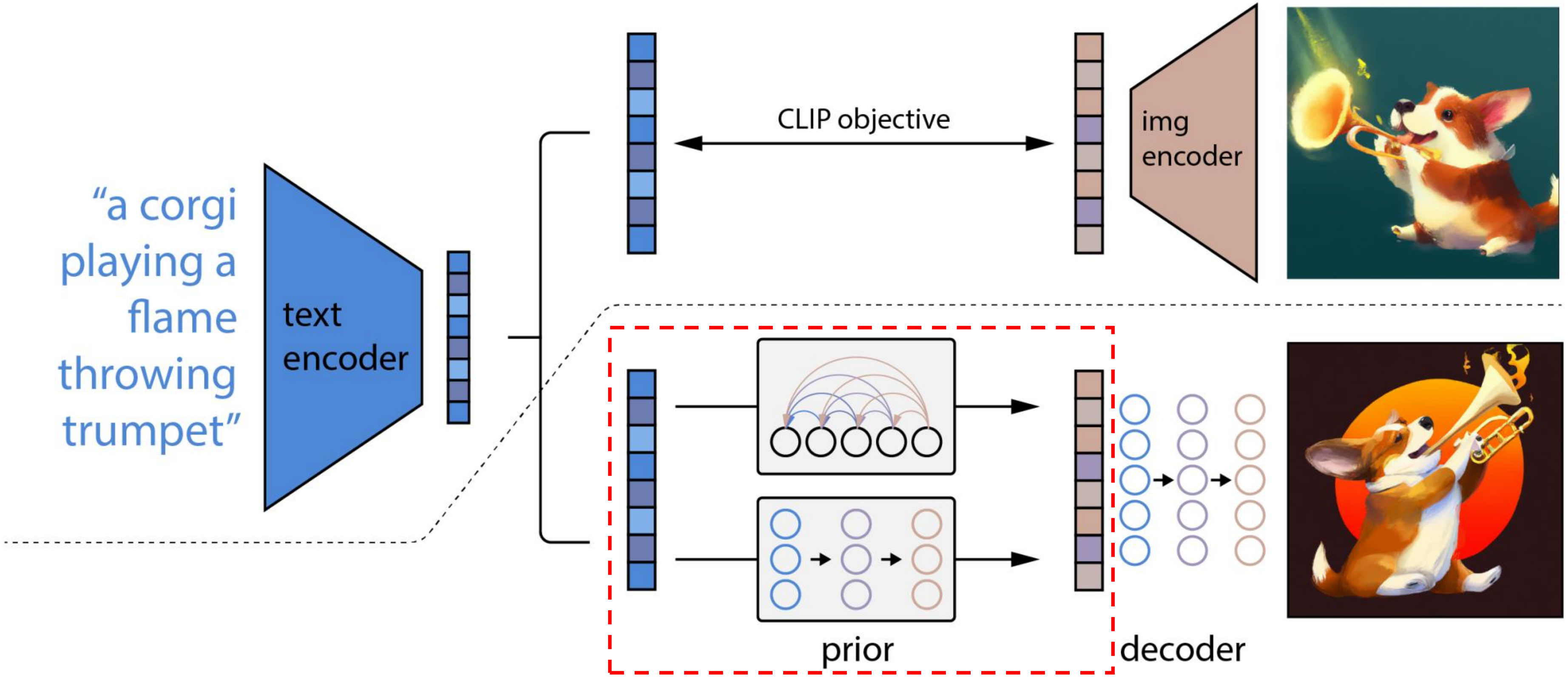
DALL-E 2

- Given a CLIP text embedding
  - Find the most compatible image embedding
    - This is the same as the zero-shot prediction
  - Generate the most compatible image embedding



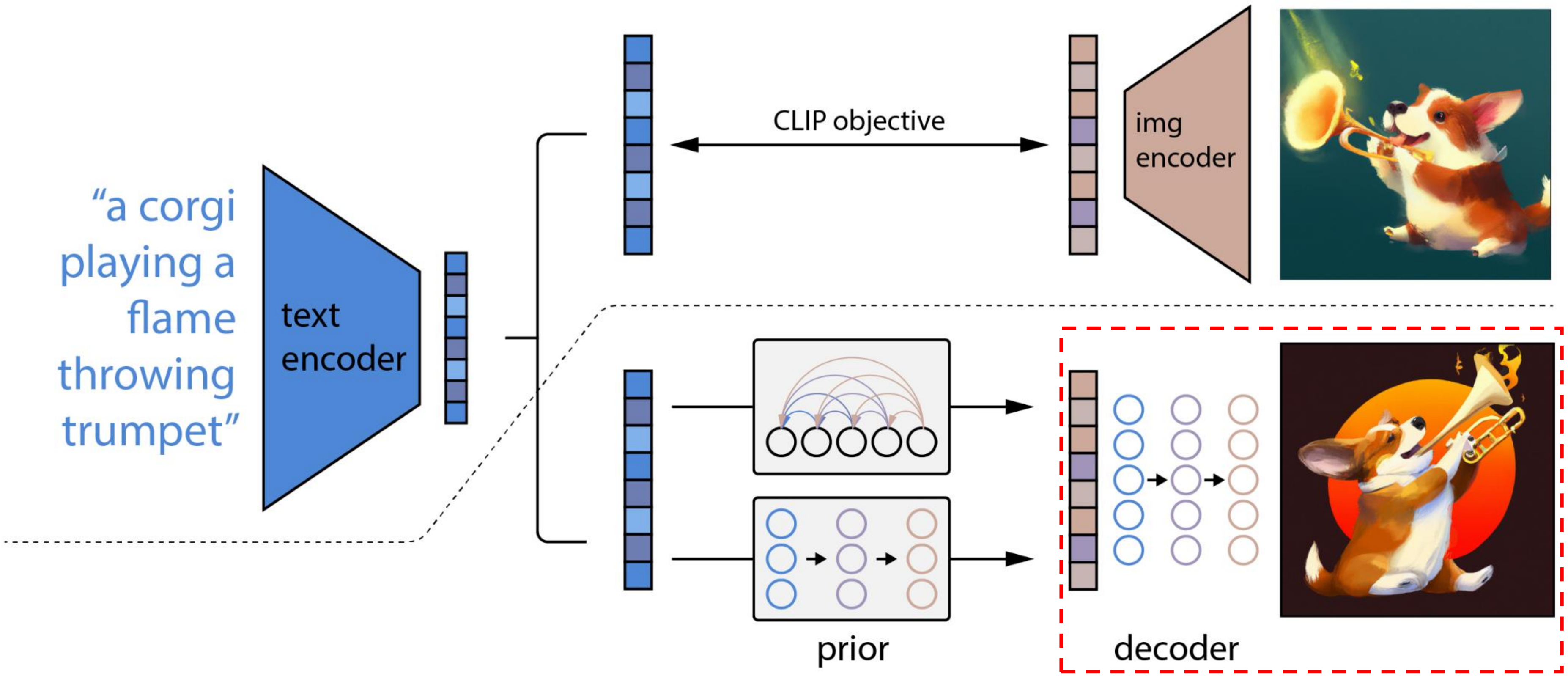
DALL-E 2

- Generating the most compatible image embedding
  - Generate via autoregression
  - Generate via CFG-Diffusion
  - Performances are comparable, but OpenAI went with the latter



DALL-E 2

- Given a CLIP image embedding
  - Generate a raw image using CFG-Diffusion



Latent DDPM

- DDPM + Autoencoder
  - Compress high-resolution images using an AE
  - Train DDPM in the latent space
  - Feed text embedding using classifier-free guidance

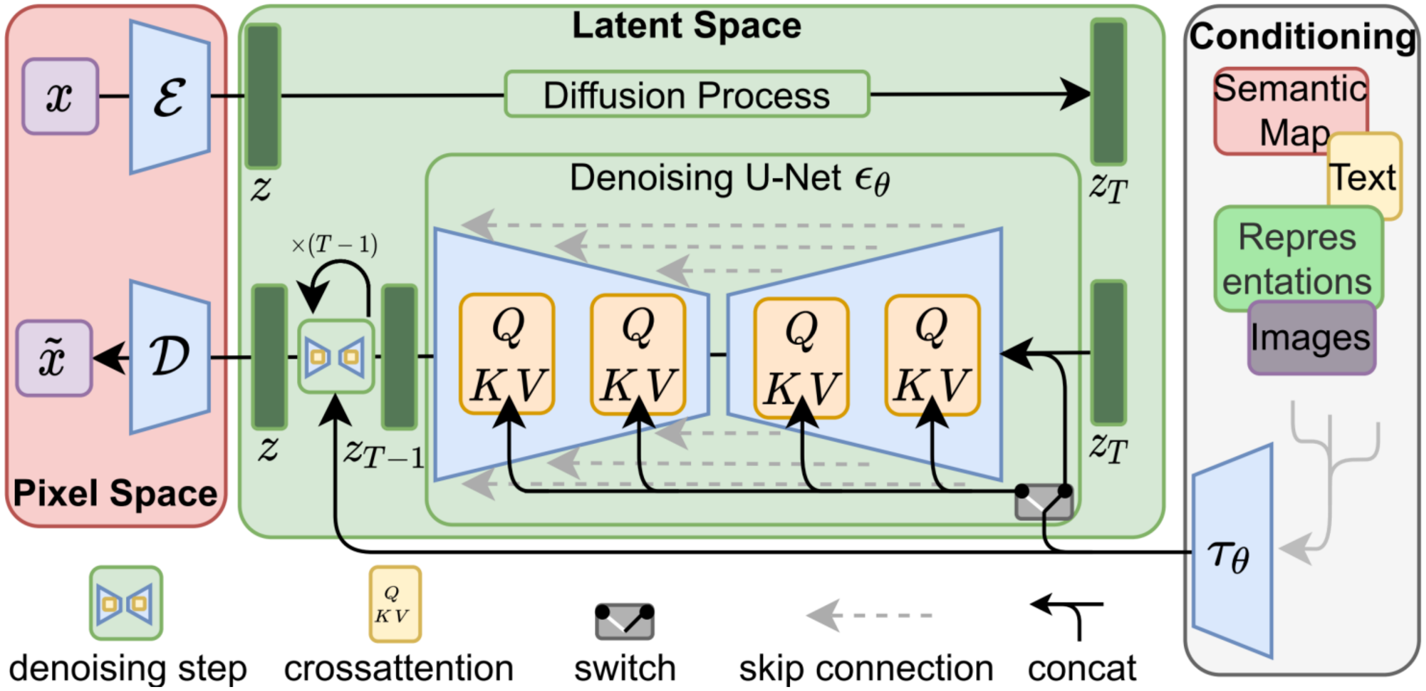


Image-Text
Multi-modal Pre-training

Image-Text Multi-modal Pretraining

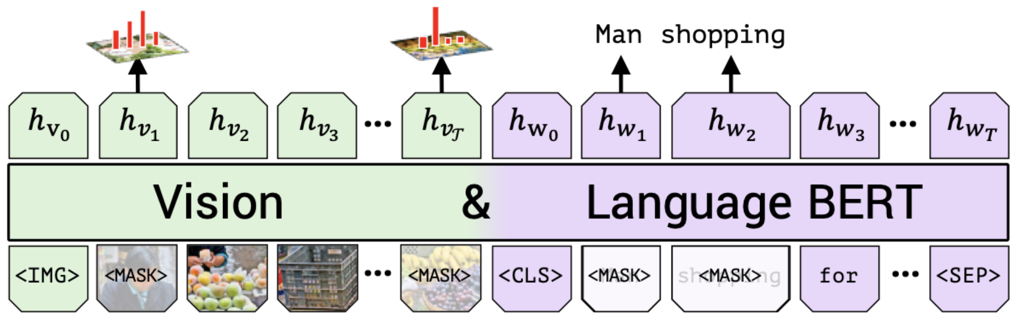
- Very active since 2019
  - VideoBERT, ViLBERT, InterBERT, LXMERT, UNITER, Unified VLP, PixelBERT, CoCa, Flamingo, BEiT v3
- Objective
  - Pre-train a model to “understand” the relationship between images and text
- Downstream tasks
  - Image retrieval
  - Visual question answering
  - Image captioning
  - Image generation
  - …

Common Strategy

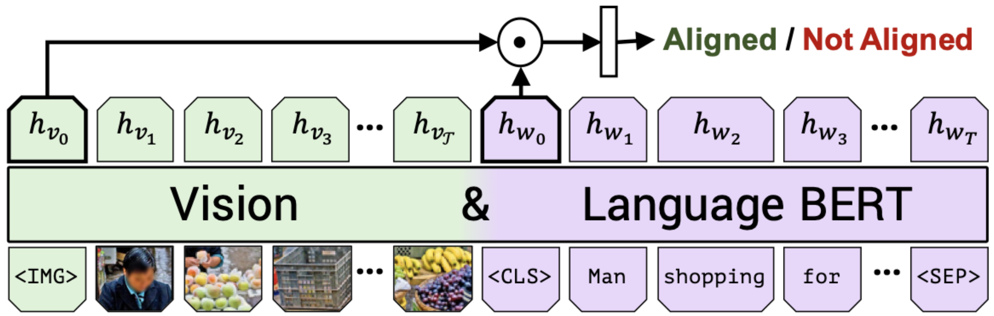
- Extract image features from the image
  - Pre-trained object detectors (e.g. Fast R-CNN, Mask R-CNN)
  - Directly feed pixel feature maps
  - Use VQVAE to quantize images into code
- Feed image features and text to BERT
- Optimize for some pre-training objective
  - Masked language modeling
  - Masked image prediction
  - Image-text alignment
  - …

ViLBERT

- ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks
    - Lu et al, NeurIPS 2019
- Masked image modeling
    - Predict the class distribution from Mask R-CNN
- Masked language modeling
    - Same as BERT
- Image-Text alignment prediction
    - Predict whether the given pair is a matching pair



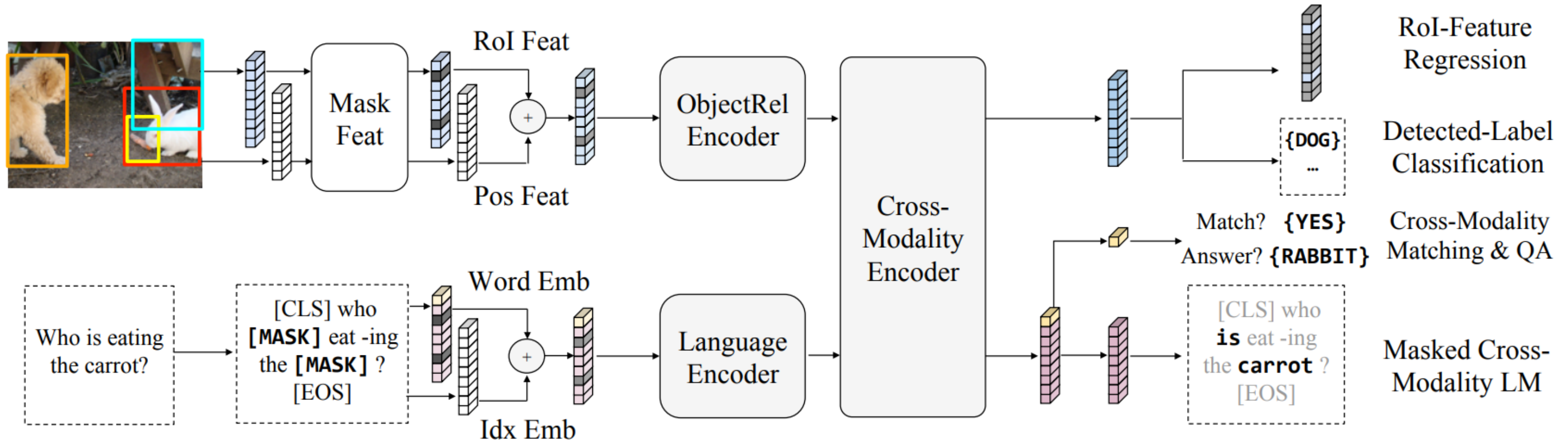
(a) Masked multi-modal learning



(b) Multi-modal alignment prediction

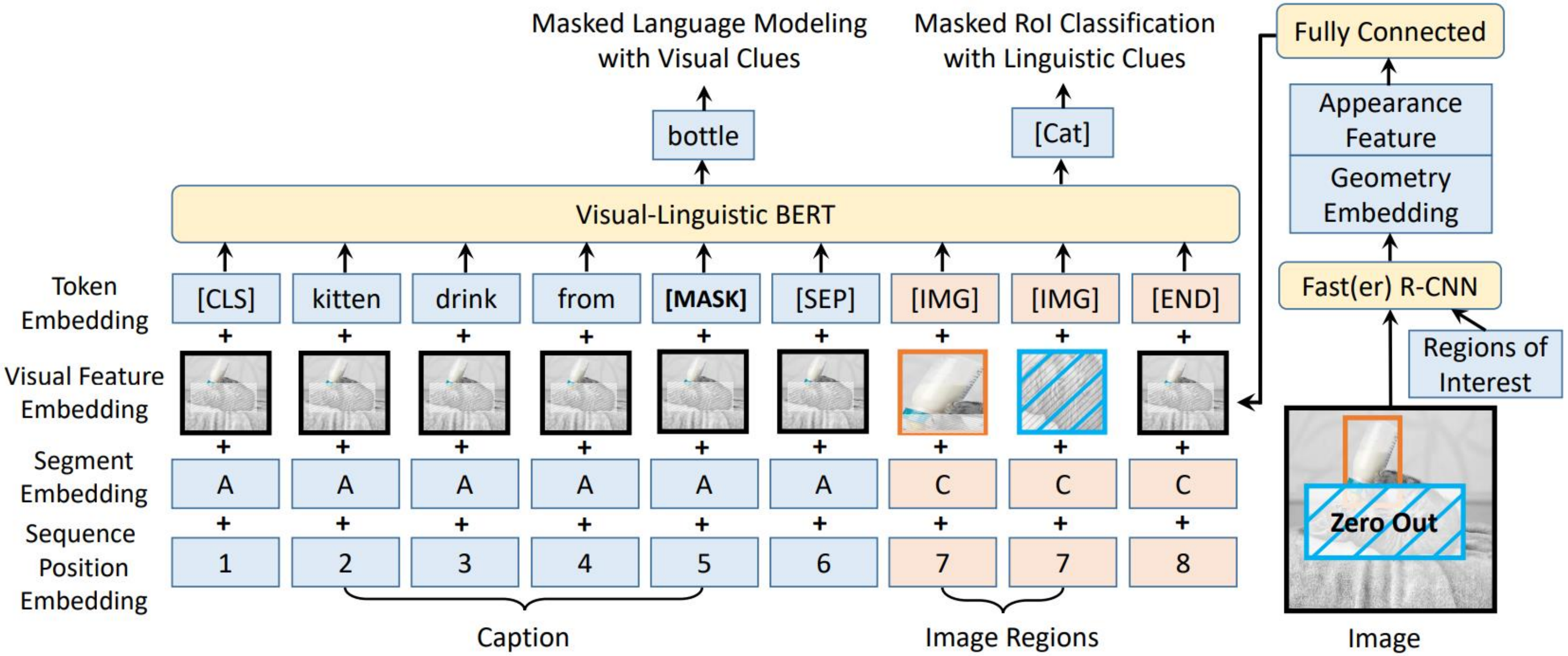
LXMERT

- LXMERT: Learning Cross-Modality Encoder Representations from Transformers
  - Tan and Bansal, EMNLP 2019
- Masked image modeling
  - Feature regression
  - Label classification
- Masked language modeling
- Image-Text alignment prediction



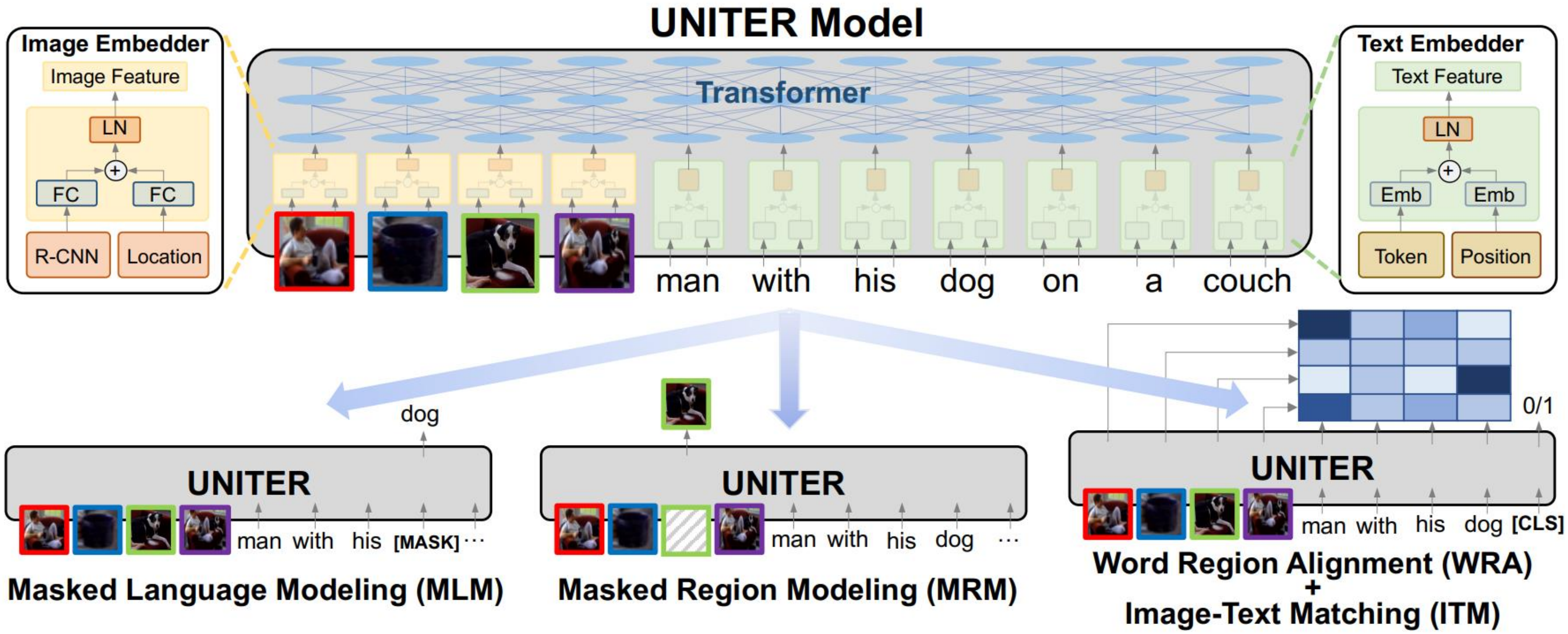
VL-BERT

- VL-BERT: Pre-training of Generic Visual-Linguistic Representations
  - Su et al., ICLR 2020



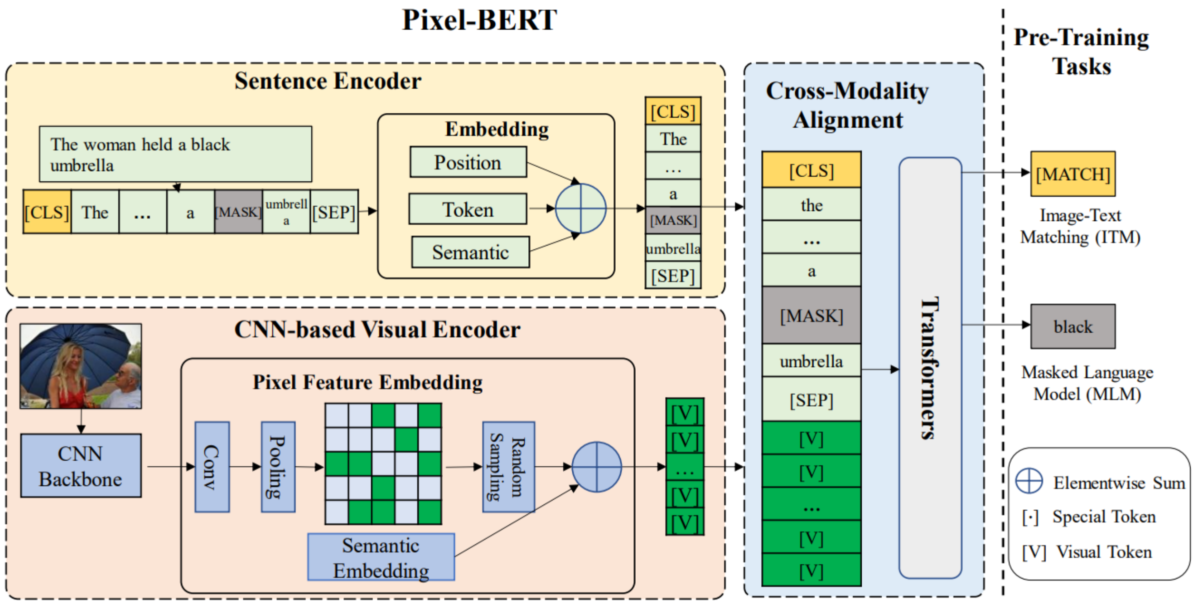
UNITER

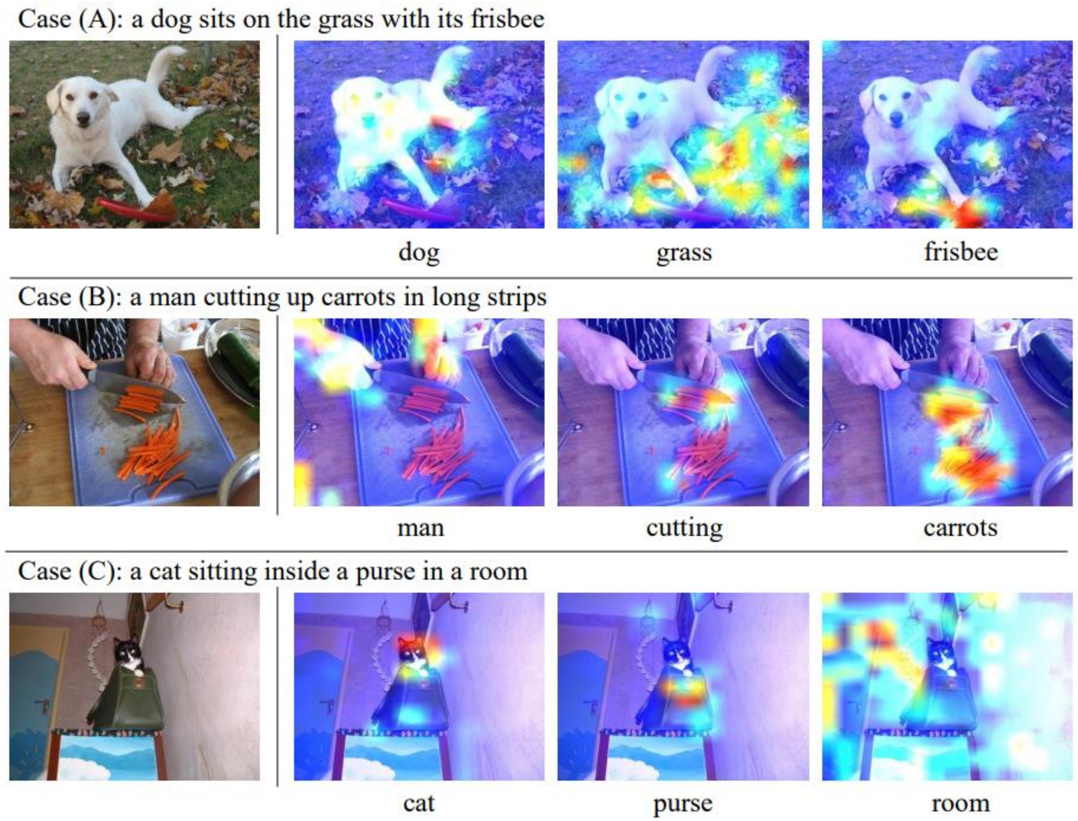
- UNITER: UNiversal Image-TExt Representation Learning
  - Chen et al., ECCV 2020



Pixel-BERT

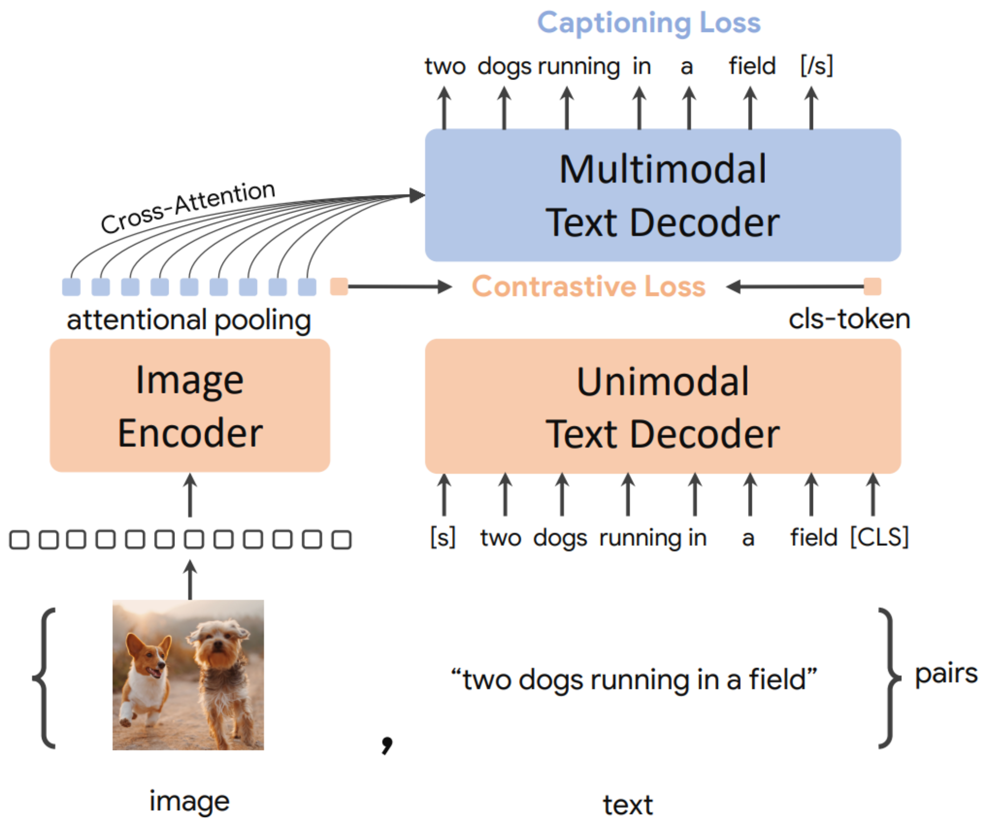
- Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers
  - Huang et al. 2020
  - Simple architecture (only CNN + Transformer, NO Object detector)





CoCa

- CoCa: Contrastive Captioners are Image-Text Foundation Models
  - Yu et al. 2022 (Google)
  - Contrastive loss + captioning loss



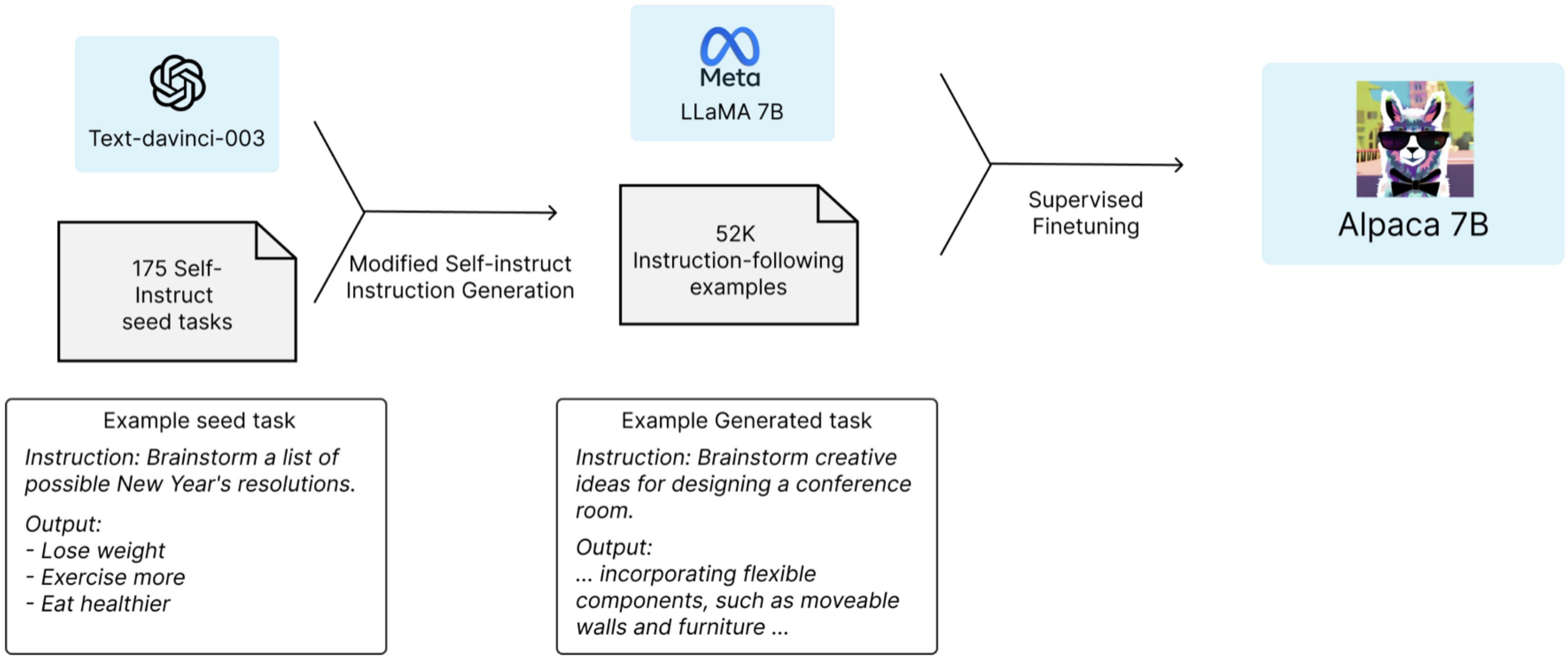
Vision LLM

LLaMA & LLaMA2

- Pre-trained language model released by Meta
  - LLaMA (Feb. 24th, 2023)
    - Model sizes: 7B, 13B, 33B, 65B
    - Context length: 2048
    - Trained on 1T-1.4T tokens
    - Training data consist of text from 20 most-spoken languages (Latin & Cyrillic alphabets)
  - LLaMA2 (Jul. 18th 2023)
    - Model sizes: 7B, 13B, 70B
    - Context length: 4096
    - Trained on 2T tokens
    - Commercial use allowed
    - Chat-finetuned model available
      - Finetuned on 100K chats & 1M human preferences

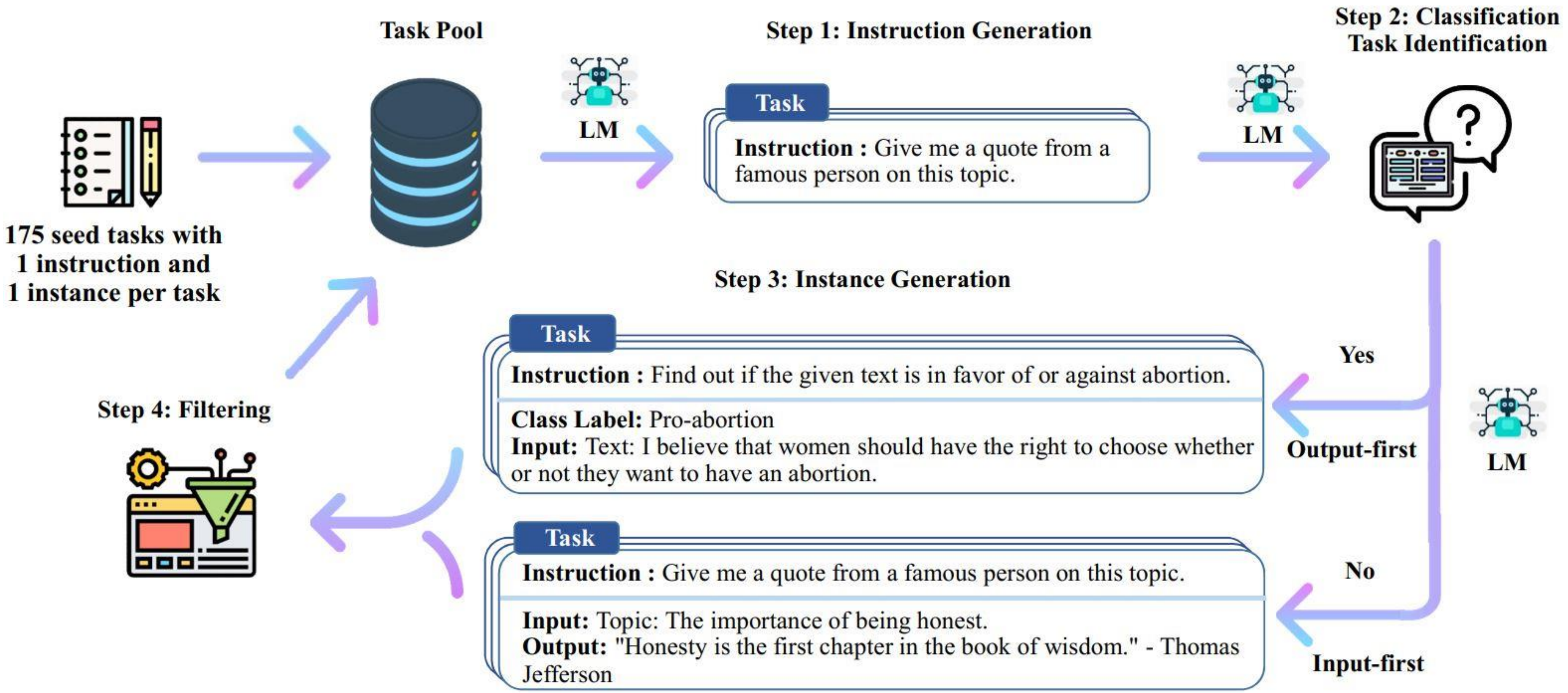
Fine-tuning LLM

- Alpaca
  - The first fine-tuned open-source LLM with less than $600
  - Generate instruction-following examples from text-davinci-003
  - Use the examples to fine-tune LLaMA



Self-Instruct

- Self-Instruct
  - Using a powerful LLM to generate instruction-following samples



Self-Instruct Examples

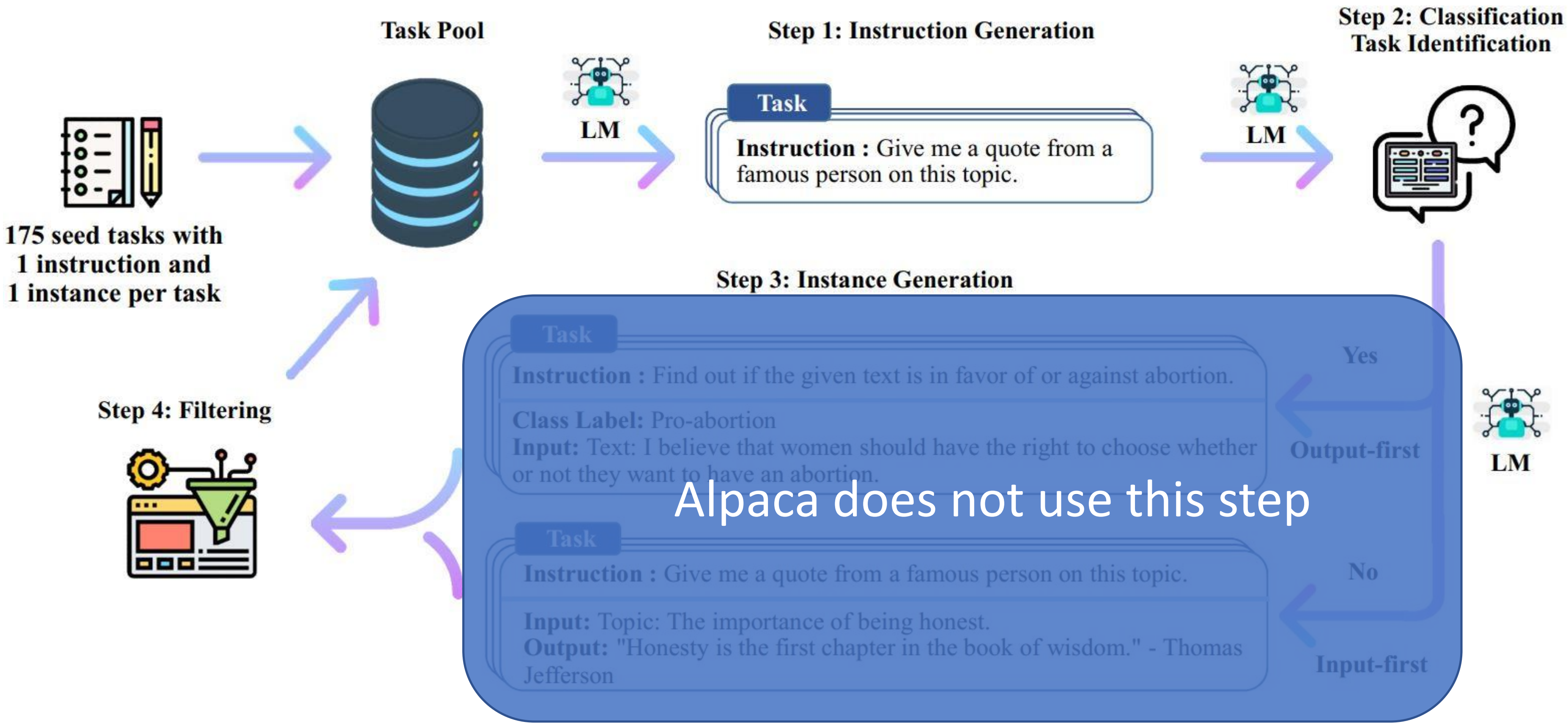
Instruction + Instance Input	Target Output	Validity

Instruction: Generate a random password with at least 6 characters.	`def generateRandomPassword():`
	`    password = ""`
	`    while len(password) < 6:`
	`        password += chr(random.randint(65, 90))`
	`    return password`
		Instruction: ✓
		Input: ✓
		Output: ✓

Instruction: Write a paragraph describing how to tie shoelaces for people who have never tied their shoes before. explain each step briefly.	Paragraph: To tie shoelaces, first you need to make a loop with one end of the lace. Then pass the other end through the loop and pull it tight. Next, you need to wrap the lace around the shoe several times. Finally, tuck in the loose end into the wrapped part of the lace.	Instruction: ✓ Input: ✓ Output: ✓
Instruction: Given a word, find out its length and its number of vowels. Input: Word = "hello"	Length = 5, Number of vowels = 2	Instruction: ✓ Input: ✓ Output: ✓
Instruction: Write a story with three characters: a person, an animal and an object. Input: Person: John, a programmer. Animal: A dog. Object: A laptop.	John was sitting on the floor in front of his laptop when he heard a knock at the door. He opened it to find a dog standing there. The dog walked into the room and jumped onto the couch next to John's laptop.	Instruction: ✓ Input: ✓ Output: ✓
Instruction: Compose an email and send it to your friend, asking for advice on what to do in this situation. Input: Situation: You are a programmer and you have been working on a project for the past few months. The deadline is approaching, but you haven't finished yet. Your manager has asked you to stay late every day this week to finish it.	Hi [Friend], I'm having a bit of trouble with my current job. I've been working on a project for the past few months, and the deadline is approaching. My manager has asked me to stay late every day this week to finish it. I don't think I can do that because I have other commitments. What should I do?	Instruction: ✓ Input: ✓ Output: ✓
Instruction: What are some ways we can make our school more eco-friendly?	- Use reusable water bottles instead of disposable plastic ones. - Turn off lights when you leave a room. - Recycle paper, cans and bottles. - Plant trees to absorb carbon dioxide.	Instruction: ✓ Input: ✓ Output: ✓

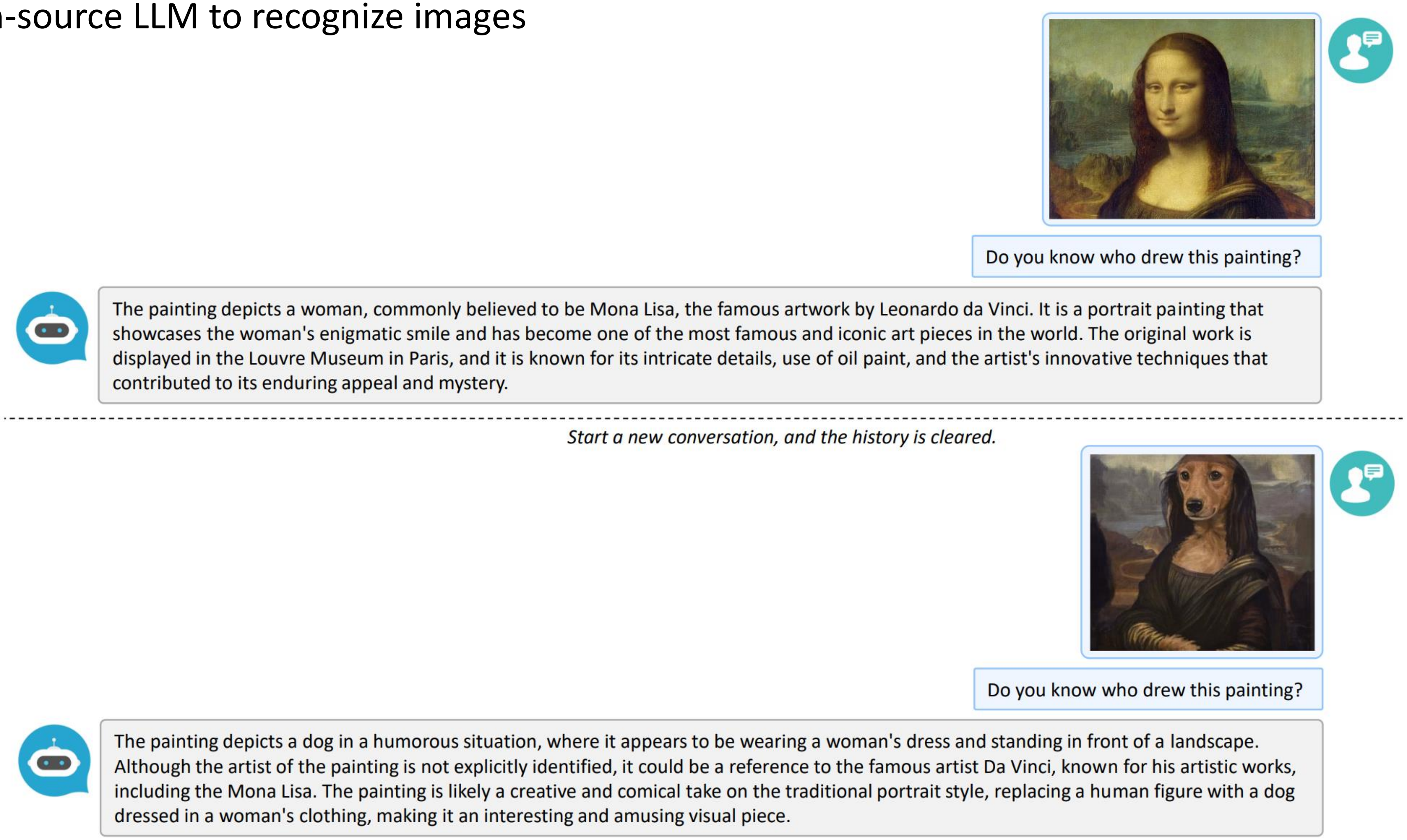
Self-Instruct

- Self-Instruct
  - Using a powerful LLM to generate instruction-following samples



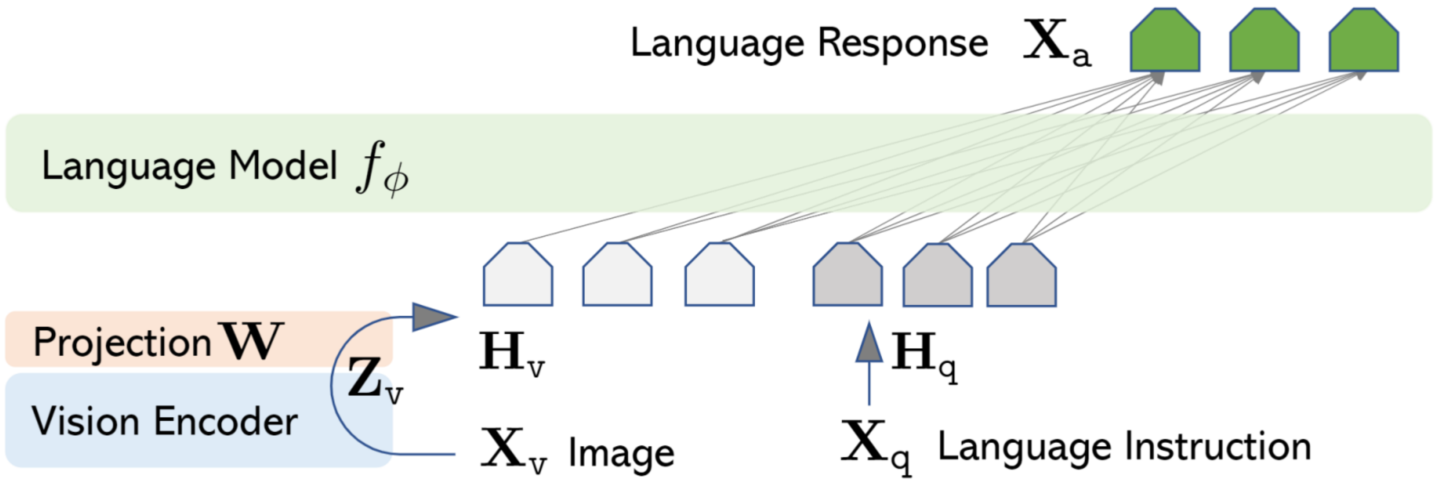
Vision LLM

- LLaVA
  - Finetuned open-source LLM to recognize images



LLaVA Training

- Language decoder: LLaMA (Vicuna)
- Image encoder: CLIP ViT



- **Stage 1: Pre-training for Feature Alignment.** Only the projection matrix is updated, based on a subset of CC3M.
- **Stage 2: Fine-tuning End-to-End.** Both the projection matrix and LLM are updated for two different use senarios:
  - **Visual Chat**: LLaVA is fine-tuned on our generated multimodal instruction-following data for daily user-oriented applications.
  - **Science QA**: LLaVA is fine-tuned on this multimodal reasoning dataset for the science domain.

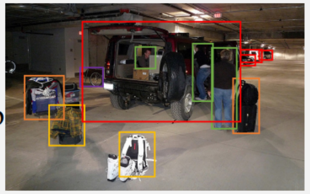
LLaVA Training

- Stage 1
  - Use a subset of Conceptual Captions 3M
    - Simple captions collected from the web, as opposed to MS-COCO
    - Filter out low-frequency noun phrases ➔ Produces 595K pairs
  - All training samples are in the form of:
    - Human: Provide a brief description of the given image. <img1>, <img2>, ... <imgN>
    - Assistant: {Insert a CC3M caption here}

LLaVA Training

- Stage 2
  - Use MS-COCO, which includes
    - 5 captions per image
    - Bounding boxes
  - Use GPT-4 to generate instruction-following samples
    - Includes three types
      - Conversation
      - Detailed description
      - Complex reasoning
    - Total 158K samples

Context type 1: Captions



A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

Context type 2: Boxes

person: [0.681, 0.242, 0.774, 0.694], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

Response type 1: conversation

Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV).
Question: Where is the vehicle parked?
Answer: The vehicle is parked in an underground parking area, likely in a public garage.
Question: What are the people in the image doing?
Answer: The people are trying to fit all of their luggage into the SUV, likely preparing for a trip.

Response type 2: detailed description

The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip.

In the parking area, various luggage items are scattered around the vehicle. There are two backpacks, one located near the left rear wheel and the other closer to the right side of the vehicle. Additionally, there are two suitcases, one on the right side of the car and another further away near the center of the parking area. A bicycle can also be seen on the left side of the vehicle.

Other cars are parked around the main SUV, with one car positioned behind it and slightly to the left, another behind and slightly to the right, and the third car further behind on the right side.

Response type 3: complex reasoning

Question: What challenges do these people face?

Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings to accommodate. They might have to strategize and arrange the luggage efficiently to ensure that everything fits properly into the vehicle. Additionally, they need to consider the comfort of the passengers and visibility while driving, so the placement of the luggage must not obstruct the driver's view or make the passengers uncomfortable during the trip.

AI504: Programming for Artificial Intelligence

Week 15: Image-Text Multimodal Learning

Edward Choi

Grad School of AI

edwardchoi@kaist.ac.kr

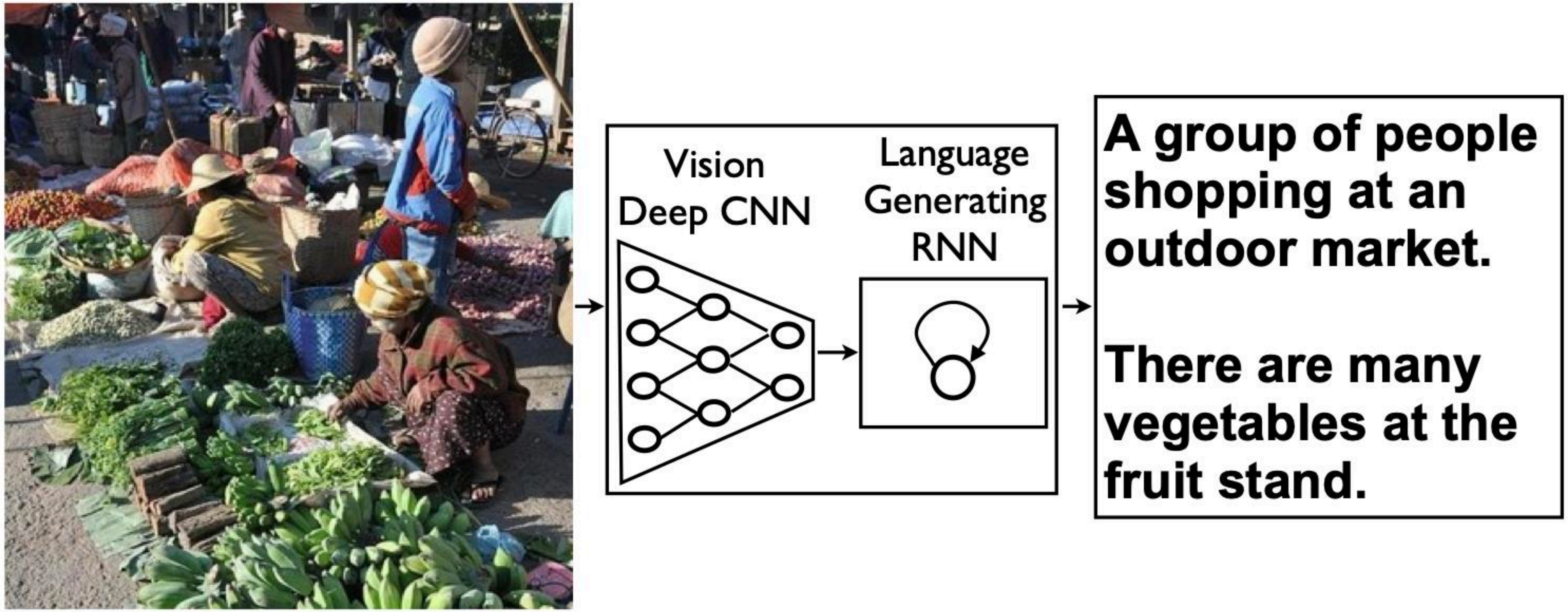
Show and Tell

Show and Tell

- Show and Tell: A Neural Image Caption Generator
  - Vinyals et al. CVPR 2015
- First paper to perform neural image captioning without any domain knowledge
  - No object detection, language modeling, description templates
  - Not text ranking, but pure generation
  - End-to-end training

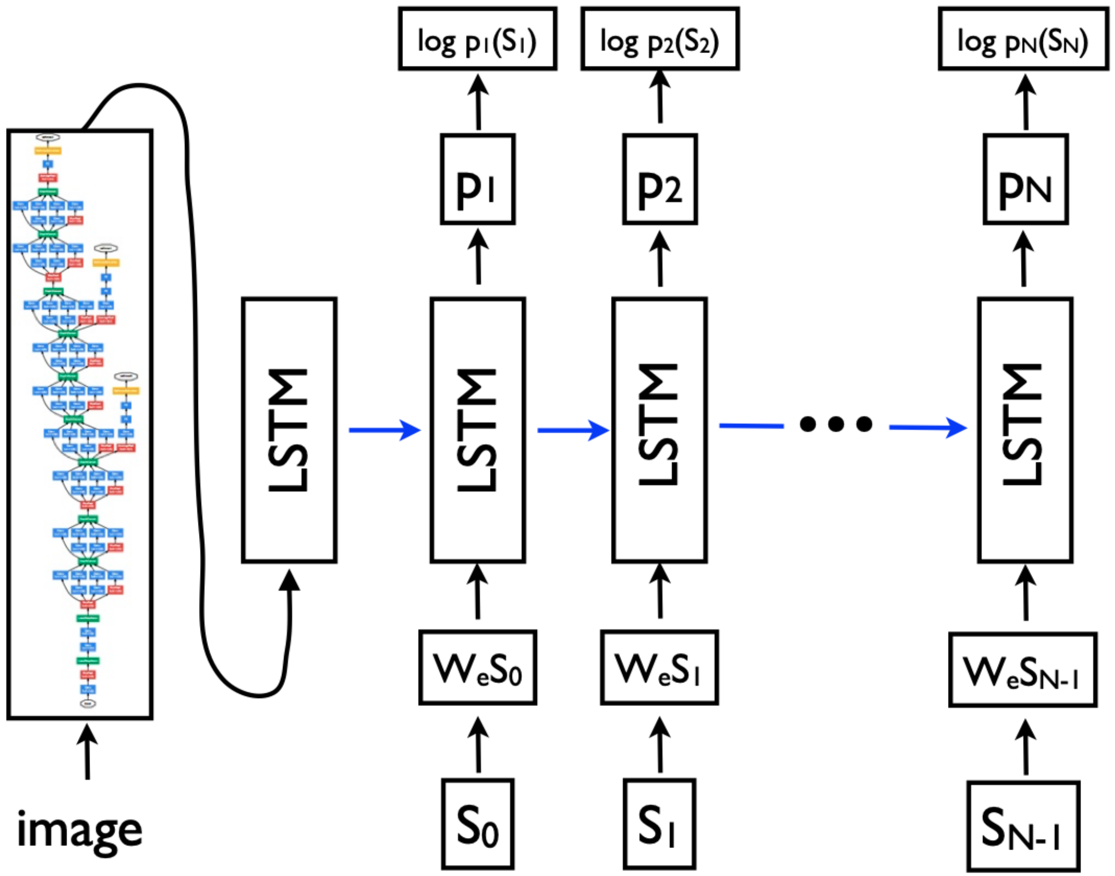
Show and Tell

- Very simple architecture



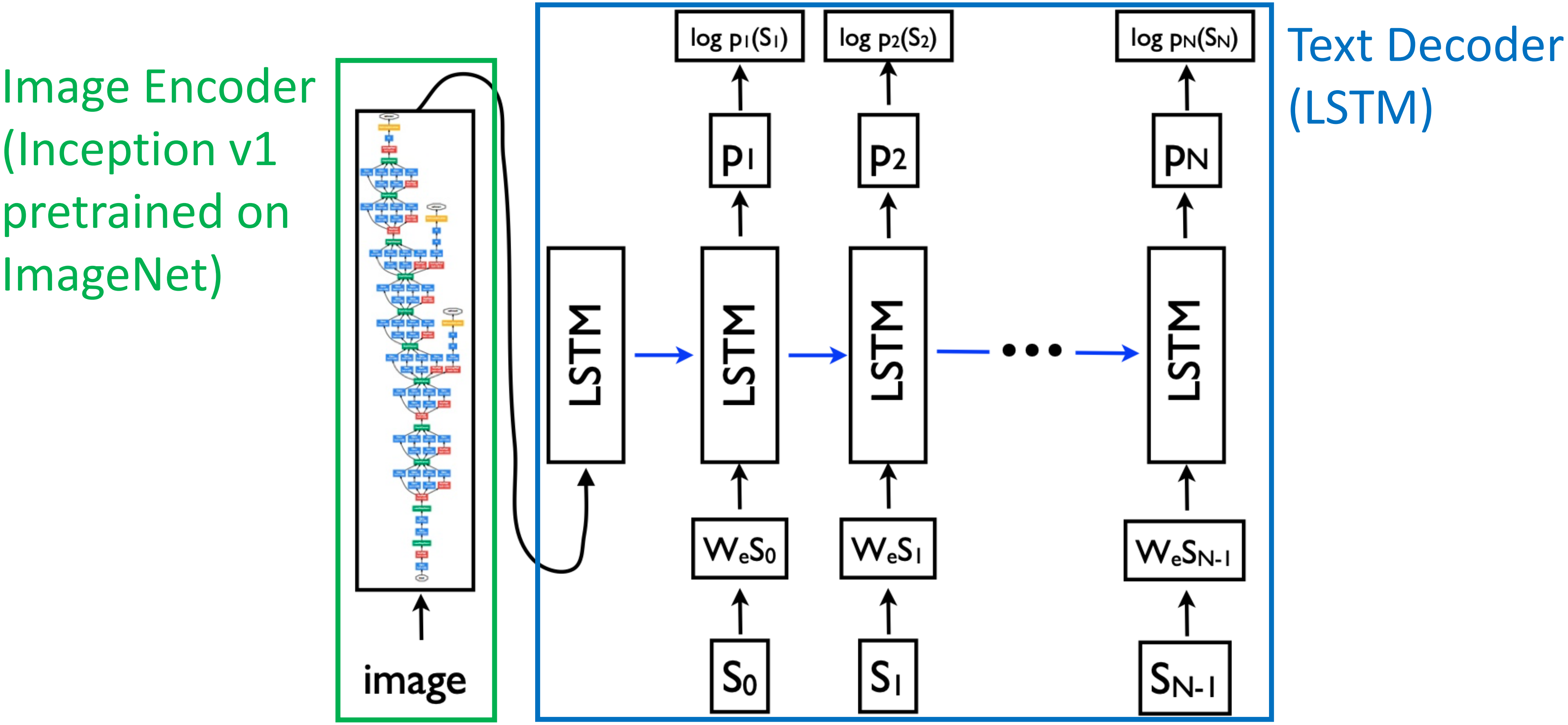
Show and Tell

- A bit more detailed architecture depiction



Show and Tell

- A bit more detailed architecture depiction

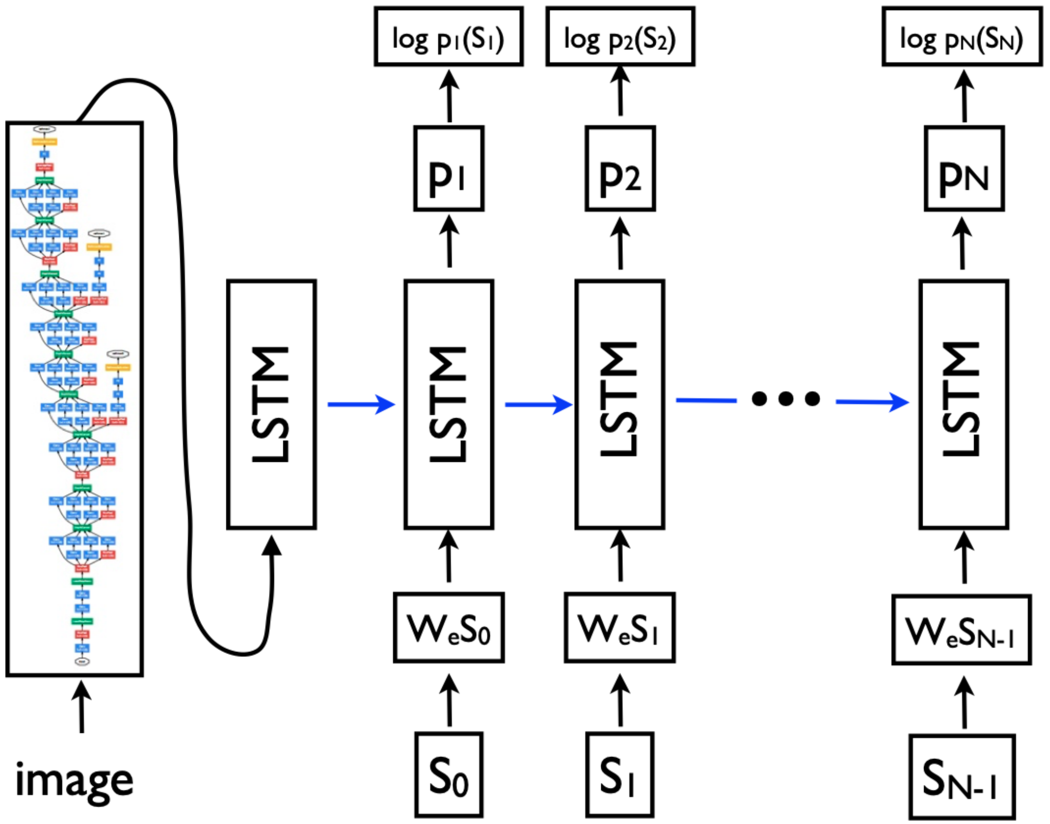


Show and Tell

- Each S_i is predicted based on p_i
  - S_i = Softmax($W_s p_i$ + b)
- Each p_i is derived based on p_{i-1}, S_{i-1}
  - p_i = RNN(p_{i-1}, $W_e S_{i-1}$)

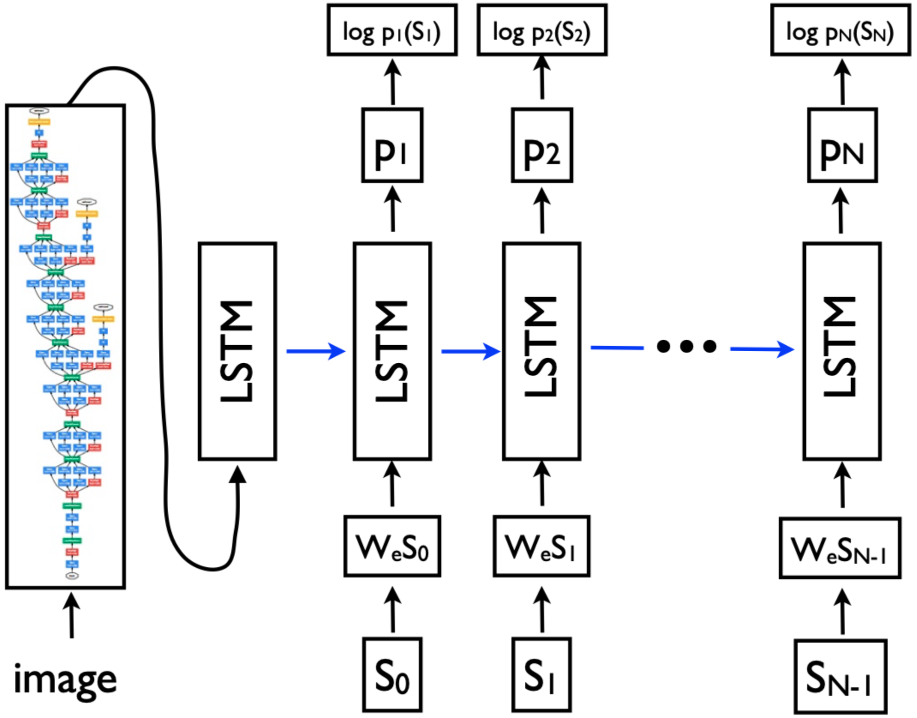
- W_e = Word embedding
- S_{-1} = CNN(Image)

- S_0:<START>, S_N:<END>



Show and Tell

- **Some technical details**
- 512 embedding size & RNN size
  - Output of CNN is also 512-dimensional
- Image embedding is “fed” into LSTM at time -1
  - Not used to initialized the LSTM hidden vector.
  - Hidden layers are probably initialized to 0
- Pretrained word embeddings didn’t help much
  - Specifically, Word2Vec
- Beam search is used with beam size 20
- Trained with negative log-likelihood



Popular Datasets

Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [6]	-	-	1000
Flickr8k [26]	6000	1000	1000
Flickr30k [33]	28000	1000	1000
MSCOCO [20]	82783	40504	40775
SBU [24]	1M	-	-

Model Performance

Metric	BLEU-4	METEOR	CIDER
NIC	**27.7**	**23.7**	**85.5**
Random	4.6	9.0	5.1
Nearest Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

Table 1. Scores on the MSCOCO development set.

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Table 2. BLEU-1 scores. We only report previous work results when available. SOTA stands for the current state-of-the-art.

Evaluation Results (grouped by human rating)

Text-to-Image

Text-to-Image

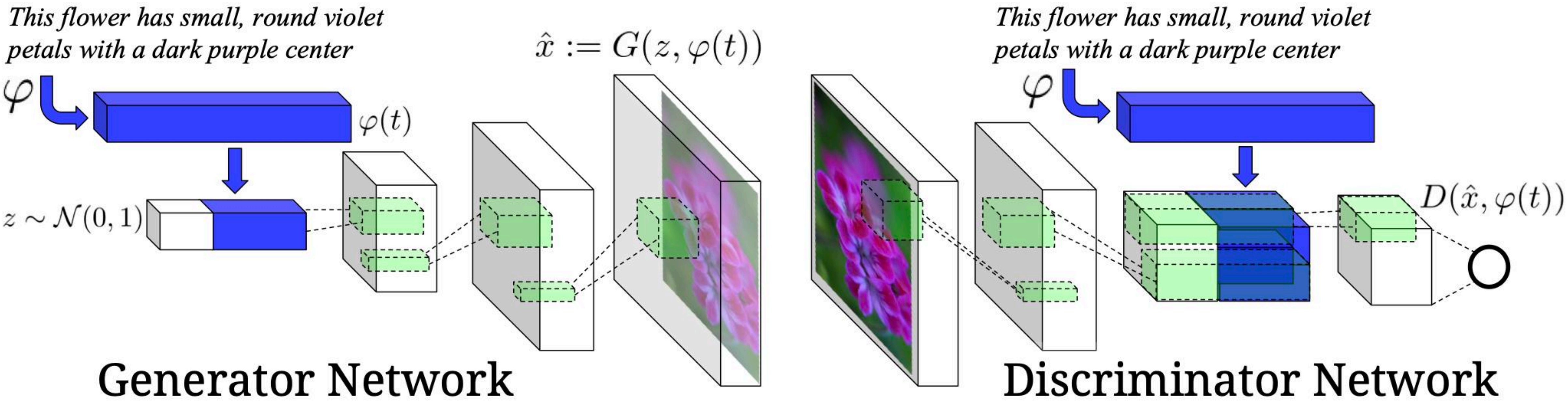
- Generative Adversarial Text to Image Synthesis
  - Reed et al. ICML 2016
- Text-conditioned image generation with GAN

this small bird has a pink breast and crown, and black primaries and secondaries.

the flower has petals that are bright pinkish purple with white stigma

Model Architecture

- Encode text with RNN
- Decode (i.e. generate) image with GAN
  - Use deconvolution (like DC-GAN) to upsample.



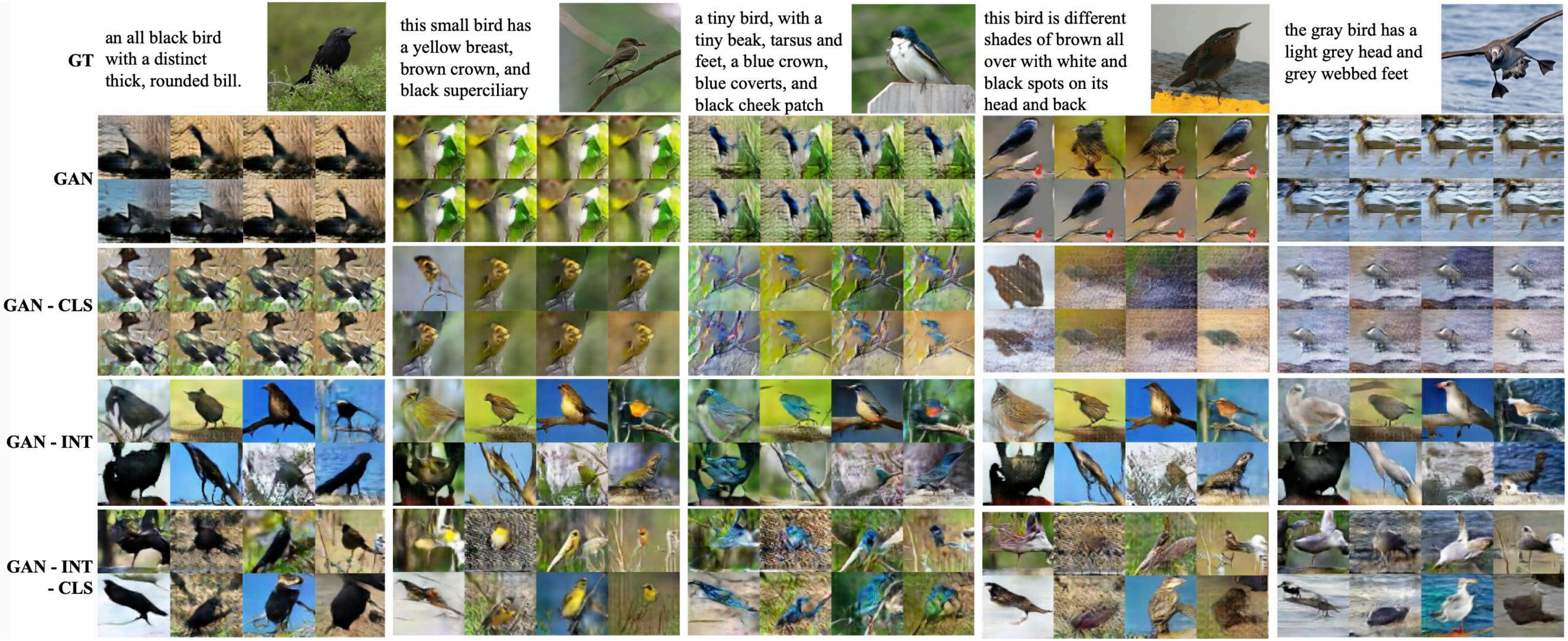
Training Strategy

- Discriminator’s job is complicated
  - Real image with right text? ➔ Real!
  - Fake image with right text? ➔ Fake!
  - Real image with wrong text? ➔ Fake!
  - Fake image with wrong text? ➔ Fake!
- Discriminator is fed three cases
  - Real image, right text
  - Real image, wrong text
  - Fake image, right text

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mis-matching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

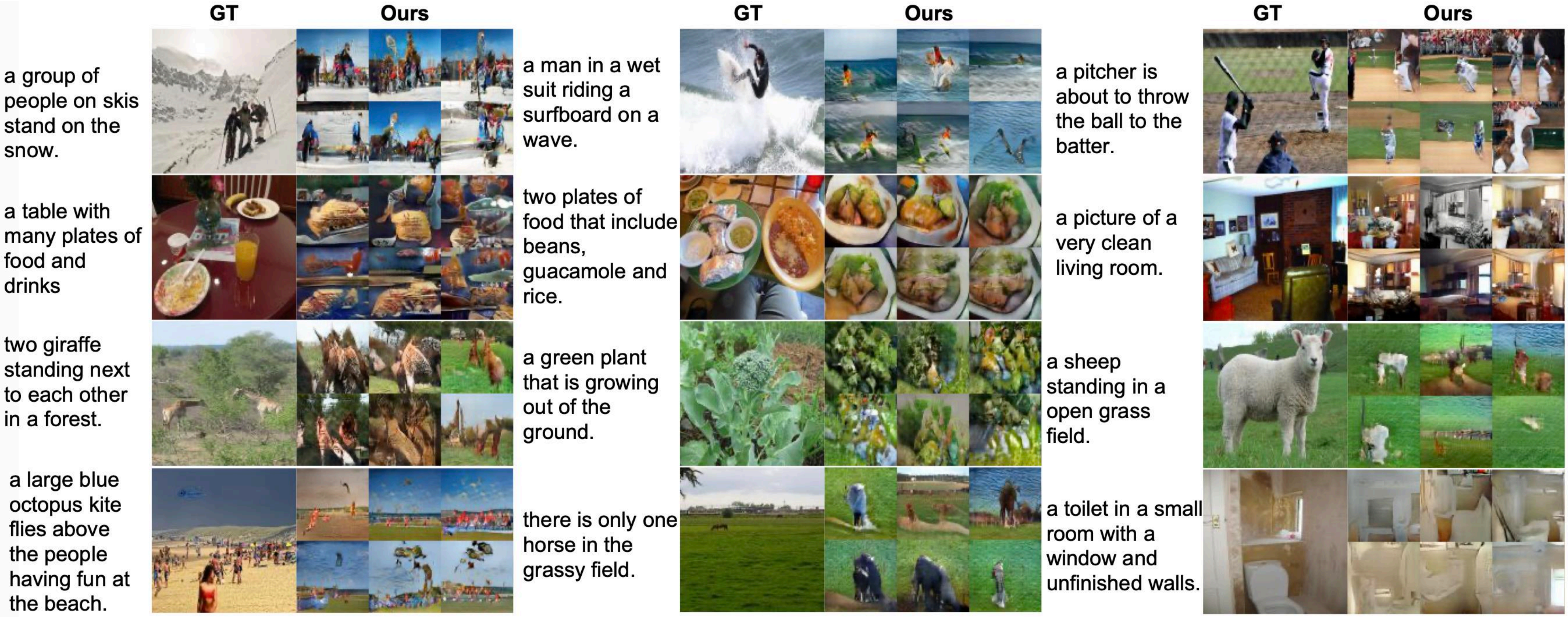
Examples


GT
an all black bird with a distinct thick, rounded bill.
this small bird has a yellow breast, brown crown, and black superciliary
a tiny bird, with a tiny beak, tarsus and feet, a blue crown, blue coverts, and black cheek patch
this bird is different shades of brown all over with white and black spots on its head and back
the gray bird has a light grey head and grey webbed feet
GAN
GAN - CLS
GAN - INT
GAN - INT - CLS

Examples


GT
this flower is white and pink in color, with petals that have veins.
these flowers have petals that start off white in color and end in a dark purple towards the tips.
bright droopy yellow petals with burgundy streaks, and a yellow stigma.
a flower with long pink petals and raised orange stamen.
the flower shown has a blue petals with a white pistil in the center
GAN
GAN - CLS
GAN - INT
GAN - INT - CLS

Examples


GT
Ours
a group of people on skis stand on the snow.
a table with many plates of food and drinks
two giraffe standing next to each other in a forest.
a large blue octopus kite flies above the people having fun at the beach.
GT
Ours
a man in a wet suit riding a surfboard on a wave.
two plates of food that include beans, guacamole and rice.
a green plant that is growing out of the ground.
there is only one horse in the grassy field.
GT
Ours
a pitcher is about to throw the ball to the batter.
a picture of a very clean living room.
a sheep standing in a open grass field.
a toilet in a small room with a window and unfinished walls.

DALL-E

- Zero-Shot Text-to-Image Generation
  - Ramesh et al. (OpenAI), 2021
- Purely based on Transformers + Vector Quantization
  - No GAN, no VAE
  - 64 layers, 62 attention heads, 12 billion params
  - 250 million text-image pairs collected from the Internet

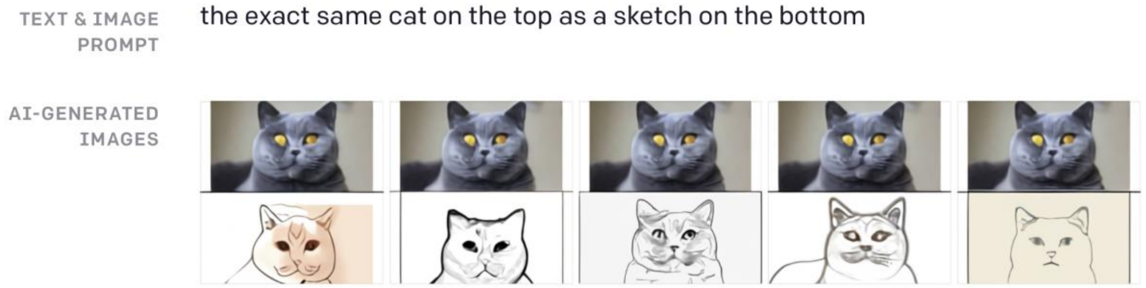
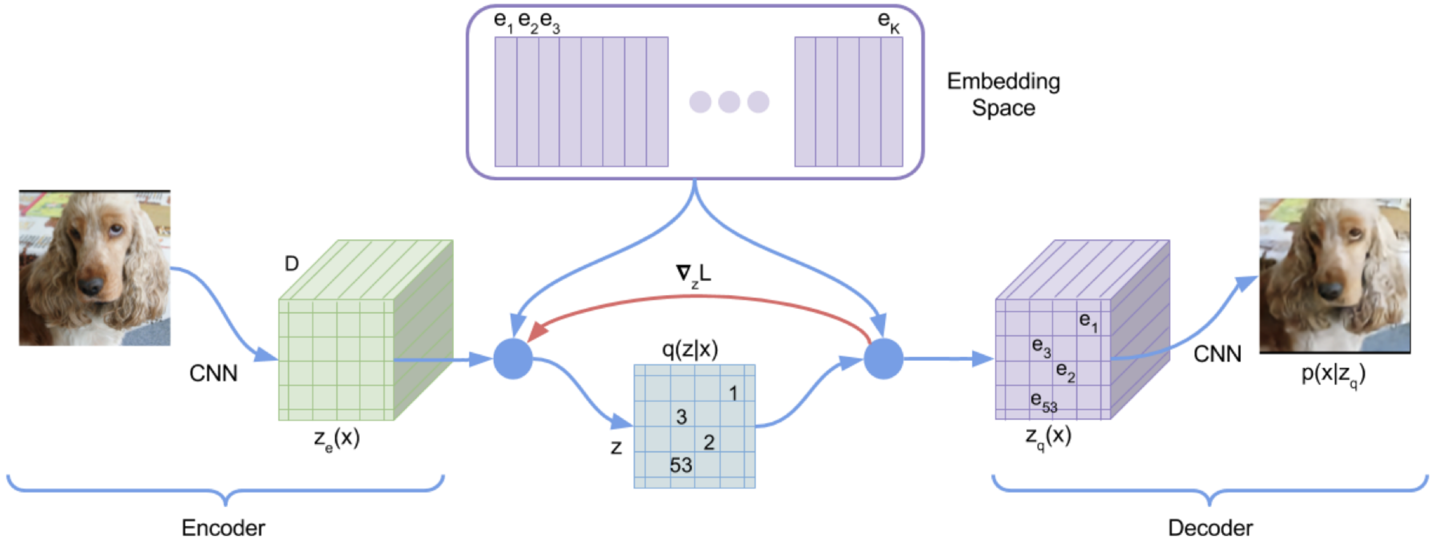


Image Tokens

- Use "vector quantization"
  - "Neural Discrete Representation Learning", van den Oord et al. (DeepMind), 2017
- Replace each image feature with a image token
  - There is a predefined dictionary of image tokens
  - Now an image can be represented as a sequence of tokens (like text!)



DALL-E Architecture