

Data Science

Lecture 11: Multimodal Data Processing

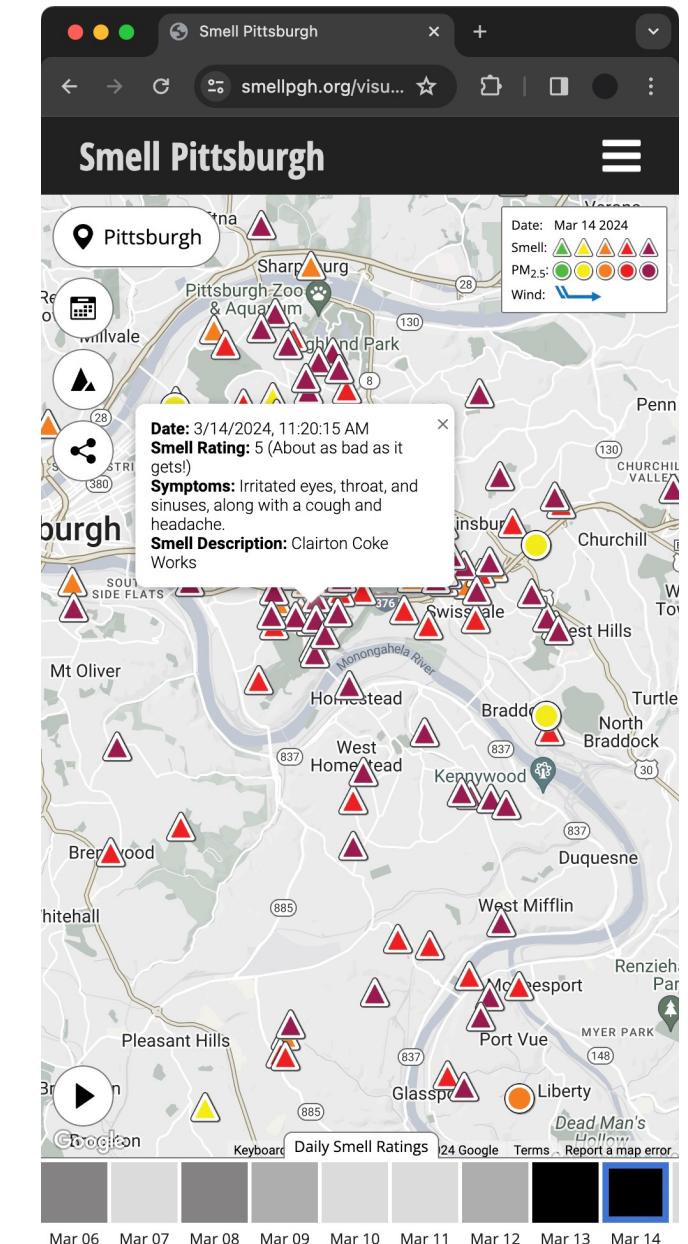
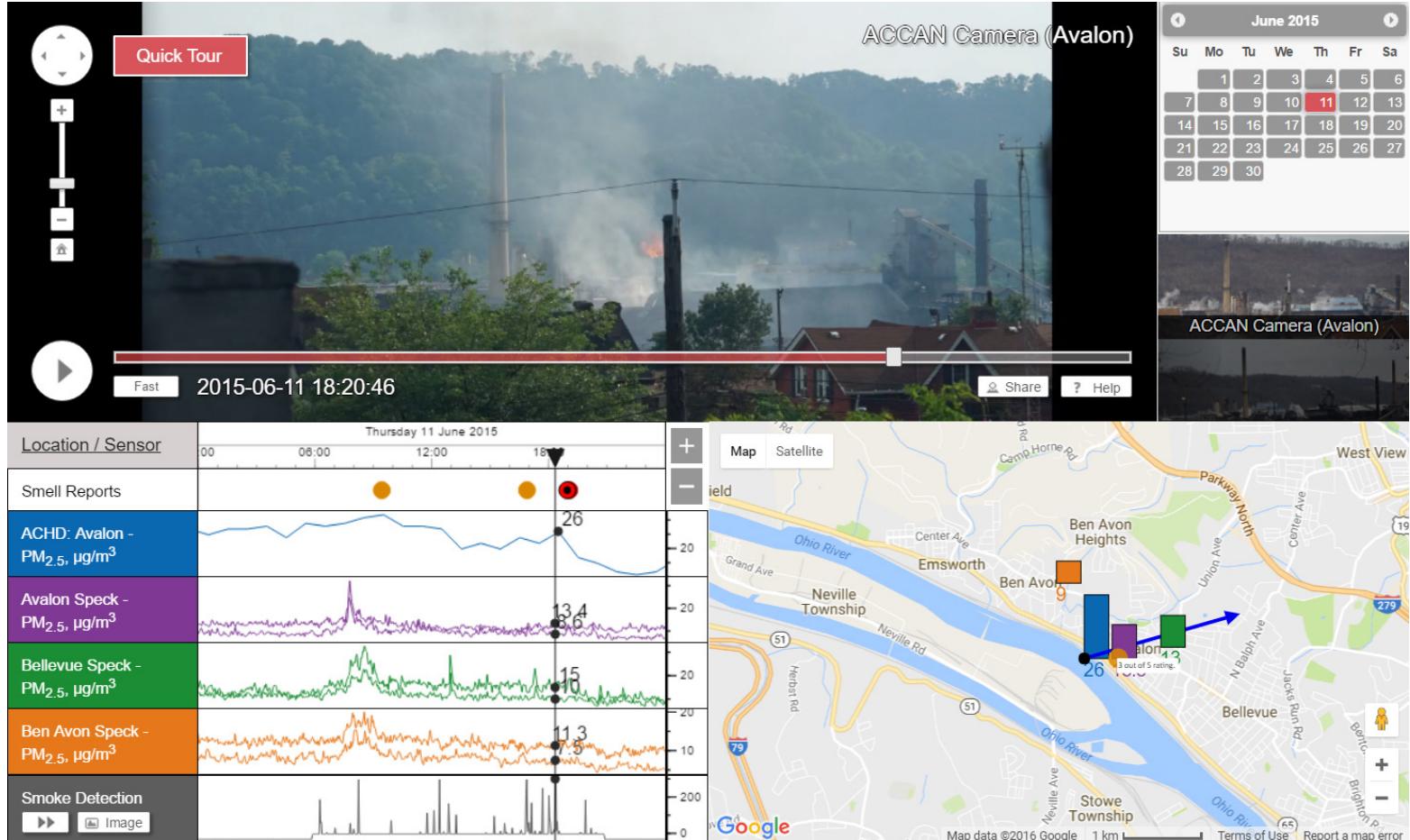


UNIVERSITY
OF AMSTERDAM

Lecturer: Yen-Chia Hsu

Date: Mar 2025

A **modality** means how a natural phenomenon is perceived or expressed. **Multimodal** means having multiple modalities.

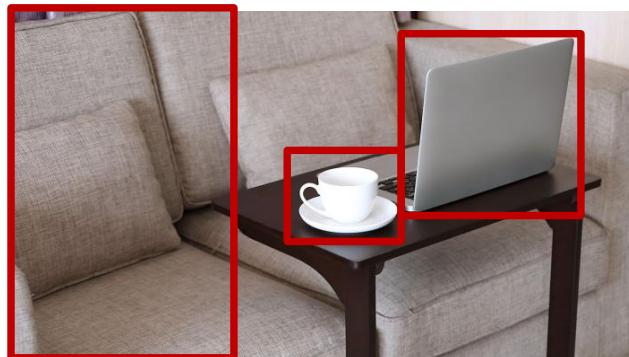
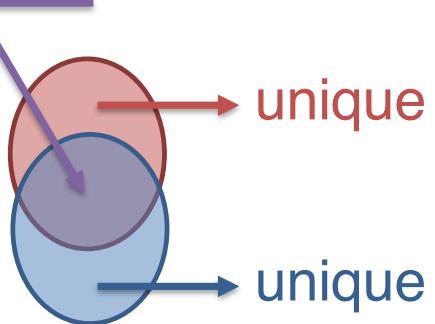
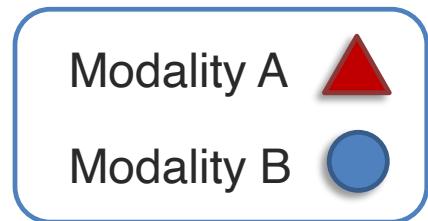


Different modalities can have different characteristics.

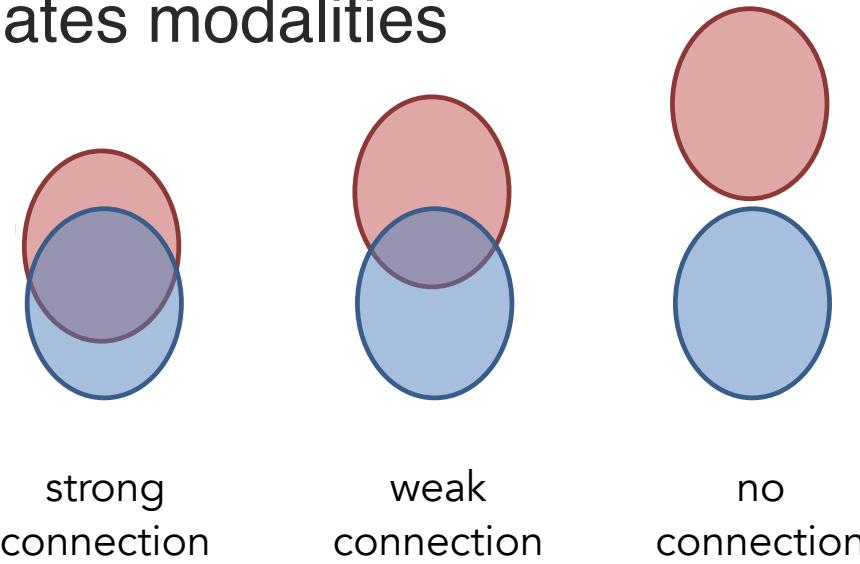
	Modality A (e.g., language)	Modality B (e.g., vision)
1 Element representations: Discrete, continuous, granularity		
2 Element distributions: Density, frequency		
3 Structure: Temporal, spatial, latent, explicit		
4 Information: Abstraction, entropy	$H(\Delta)$	$H(\bullet)$
5 Noise: Uncertainty, noise, missing data		
6 Relevance: Task, context dependence	$\longrightarrow y_1$	$\longrightarrow y_2$

Different modalities can share information with different levels of connections.

Connected: Shared information that relates modalities



A **teacup** on the **right** of a **laptop**
in a **clean room**.



The shared information can be connected in different ways.

Association



e.g., correlation,
co-occurrence

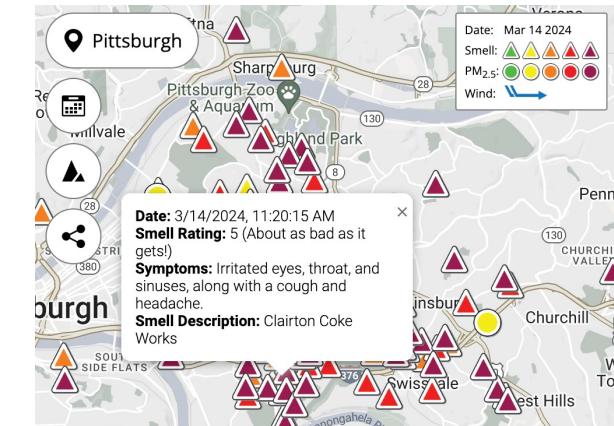


A **teacup** on the **right** of a **laptop**
in a **clean room**.

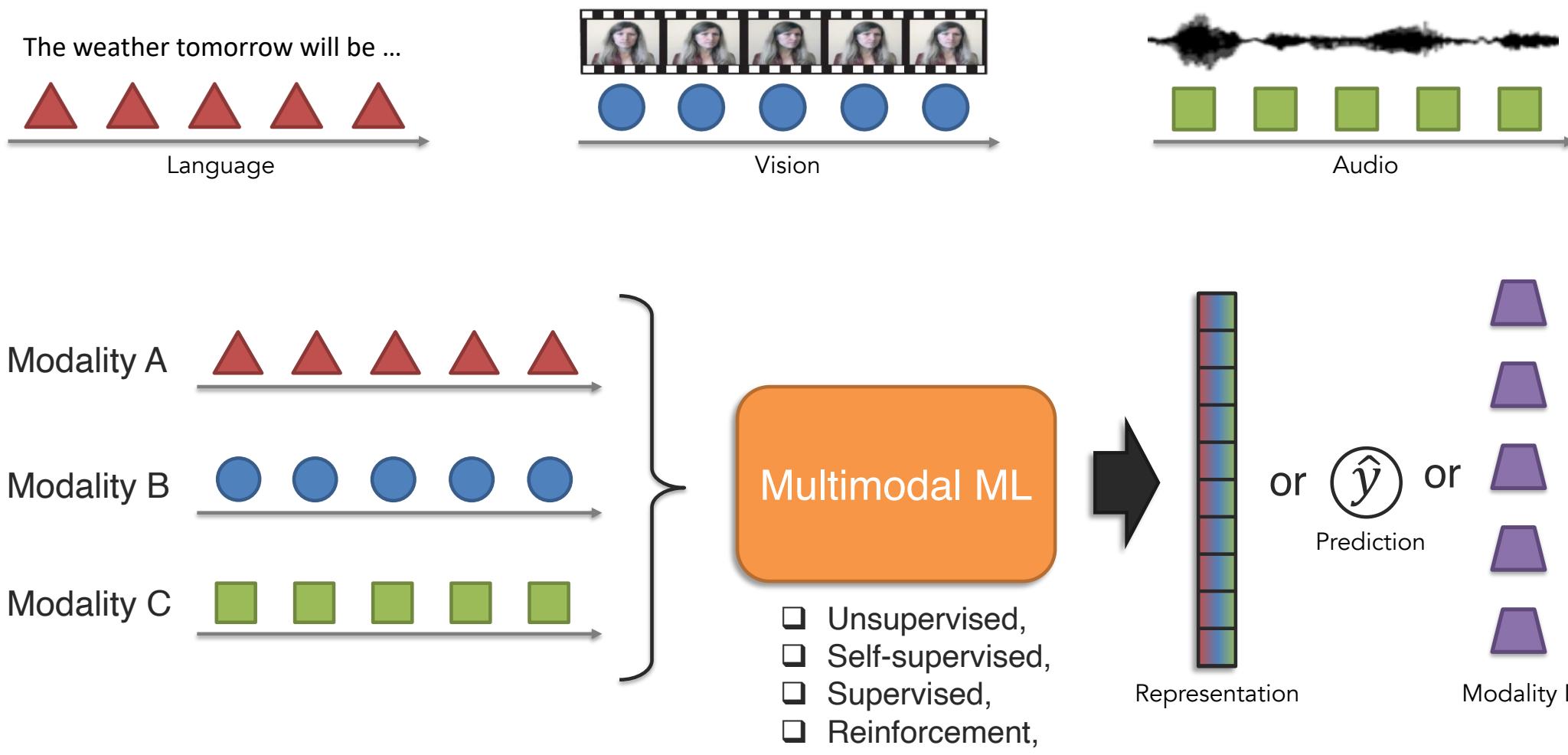
Dependency



e.g., causal,
temporal



Multiple modalities can exist in different parts of the machine learning pipeline.



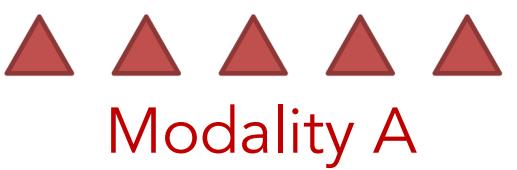
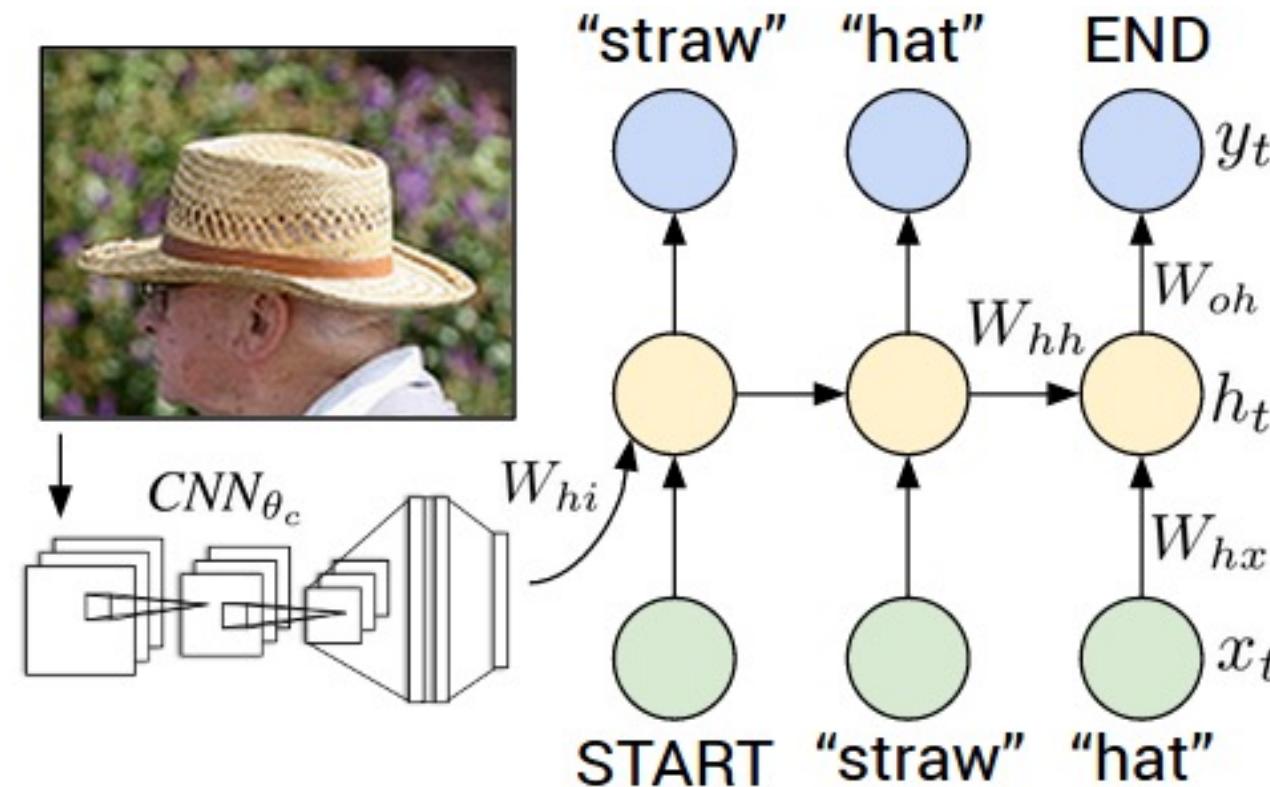
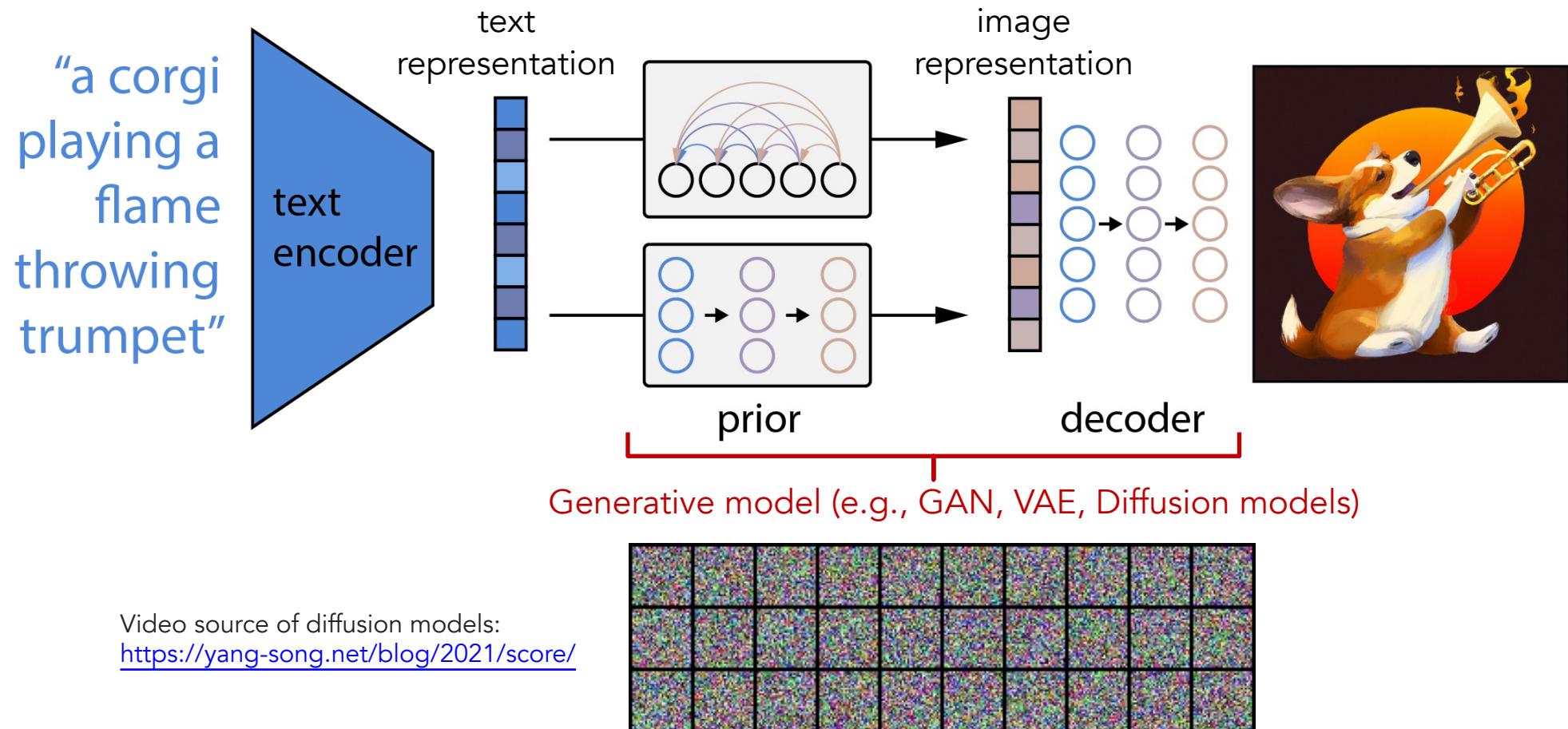


Image Captioning takes images as input and then outputs sentences that describe the input images (vision→language).



We can also take text as input and then generate images that match the input text (language→vision).

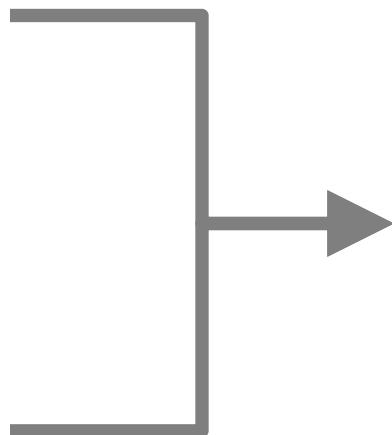




Modality A



Modality B



\hat{y}

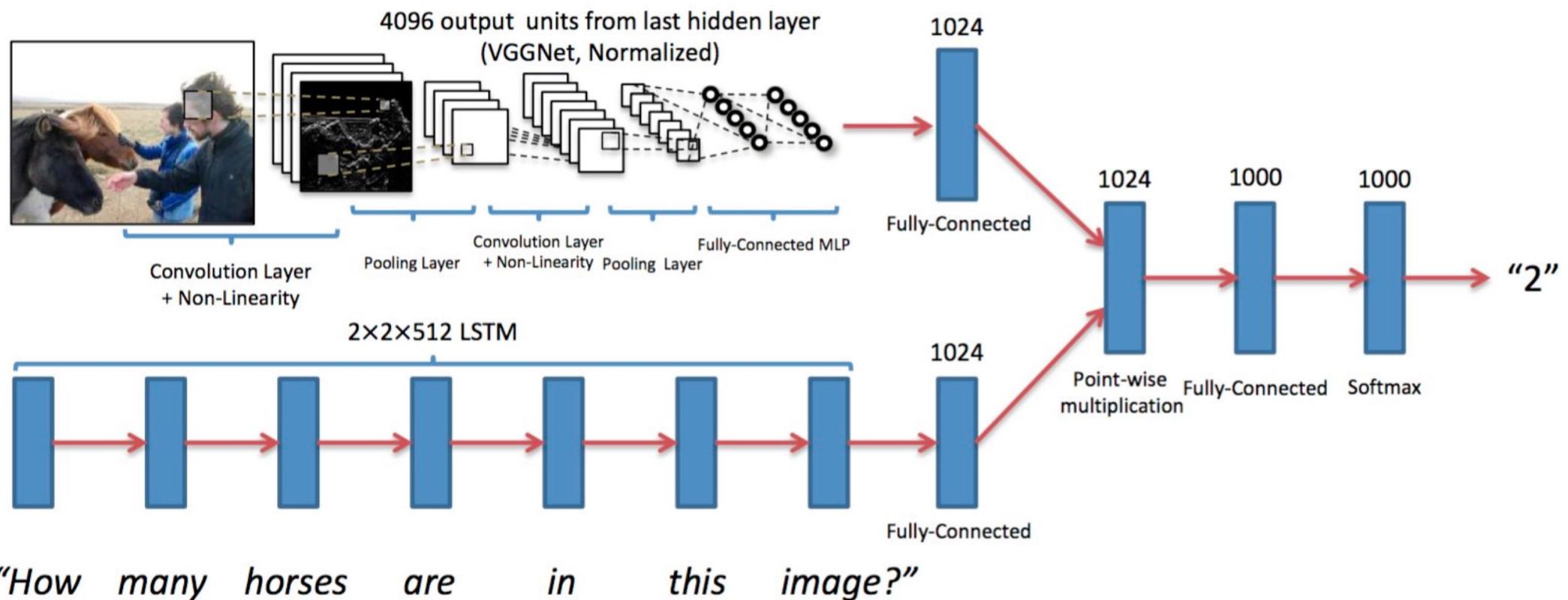
Prediction

OR



Modality C

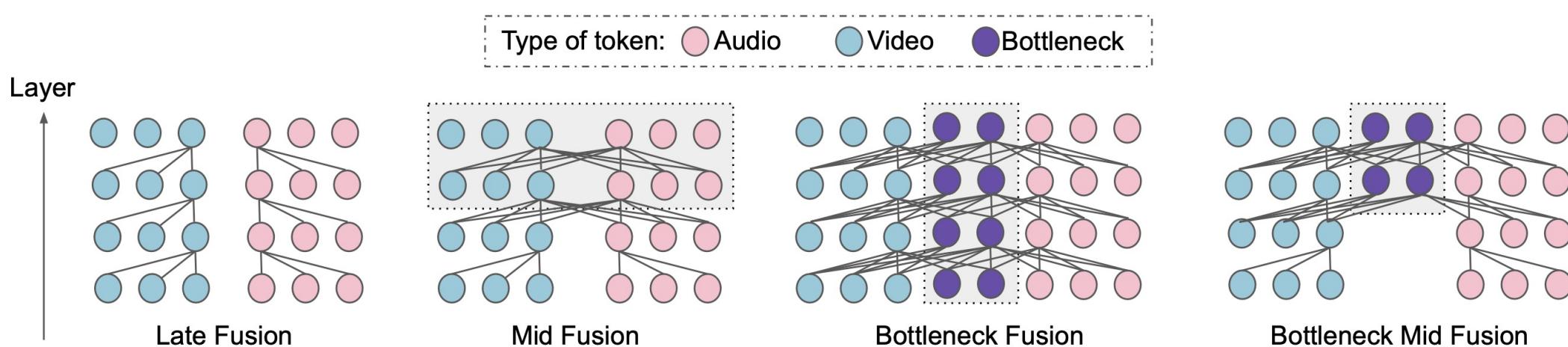
Visual Question Answering takes both images and sentences as input and then outputs a label of text-based multiple-choice answer (**vision+language→label**).



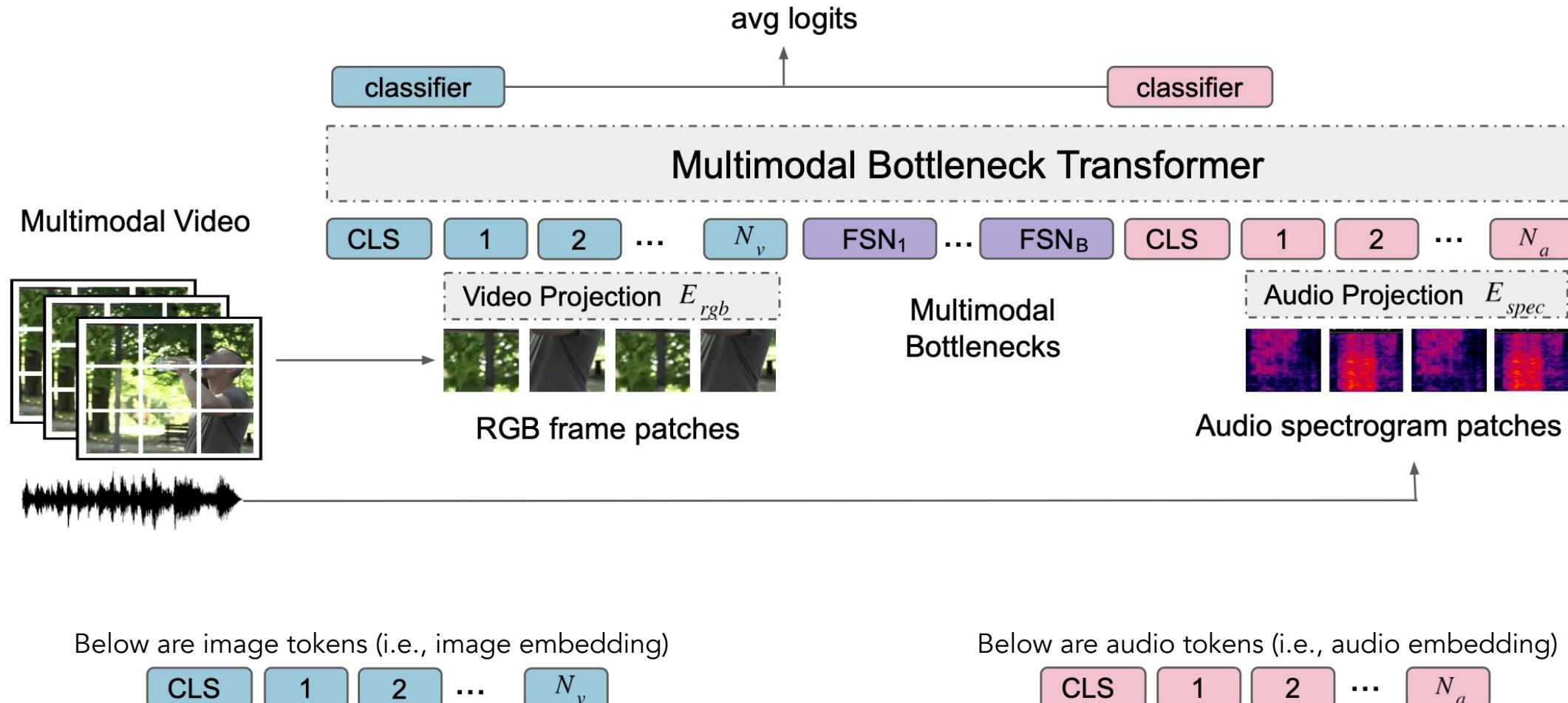
When the input has multiple modalities, we can **fuse the modalities** or explicitly learn their connections in the model architecture.



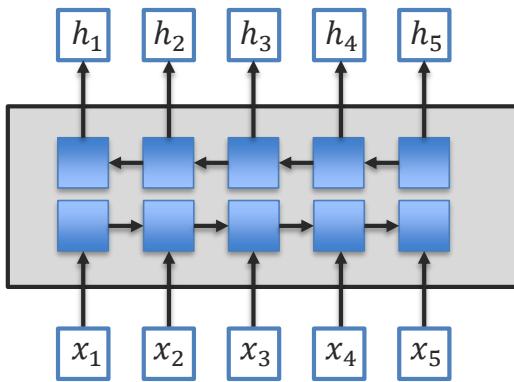
Source: CMU 11-777 MMML Course Lecture 1.1 -- <https://cmu-multicomp-lab.github.io/mmml-course/fall2023/schedule/>



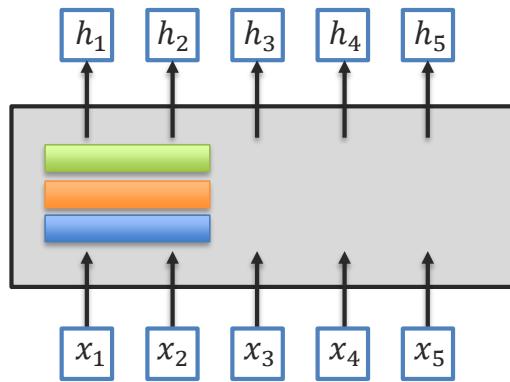
Video Classification can use both video and audio signals to predict output categories
 (vision+audio→labels).



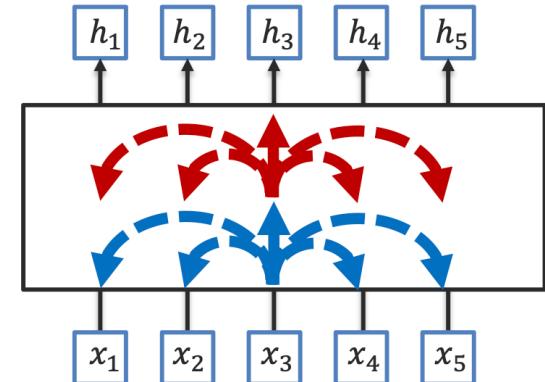
Transformers use **self-attention**, which is a way of encoding sequences to tell how much attention each input should pay attention to the other inputs (including itself).



Bidirectional RNN



Convolution

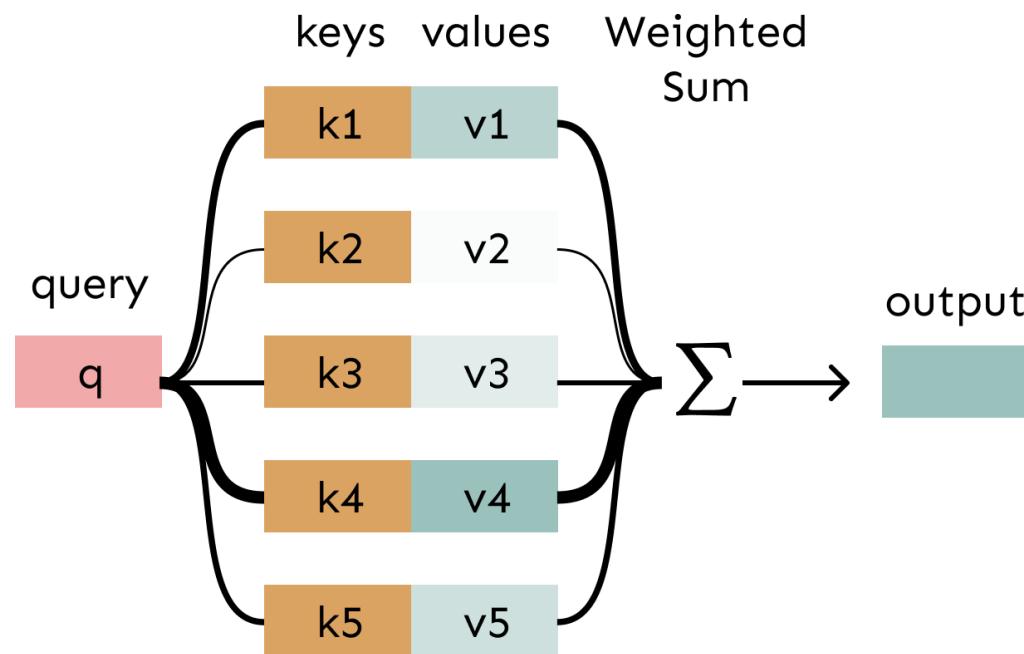


Self-attention

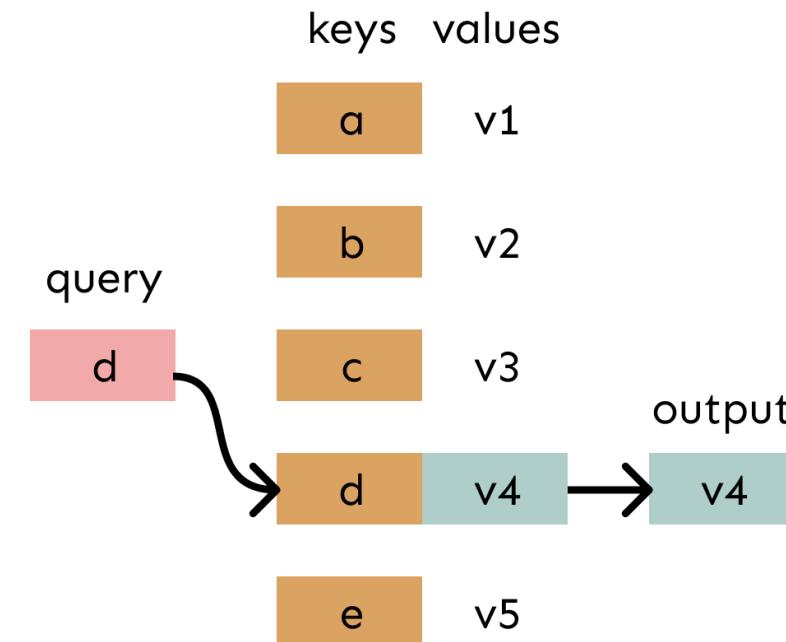
Attention is **weighted** averaging, which lets you do lookups!

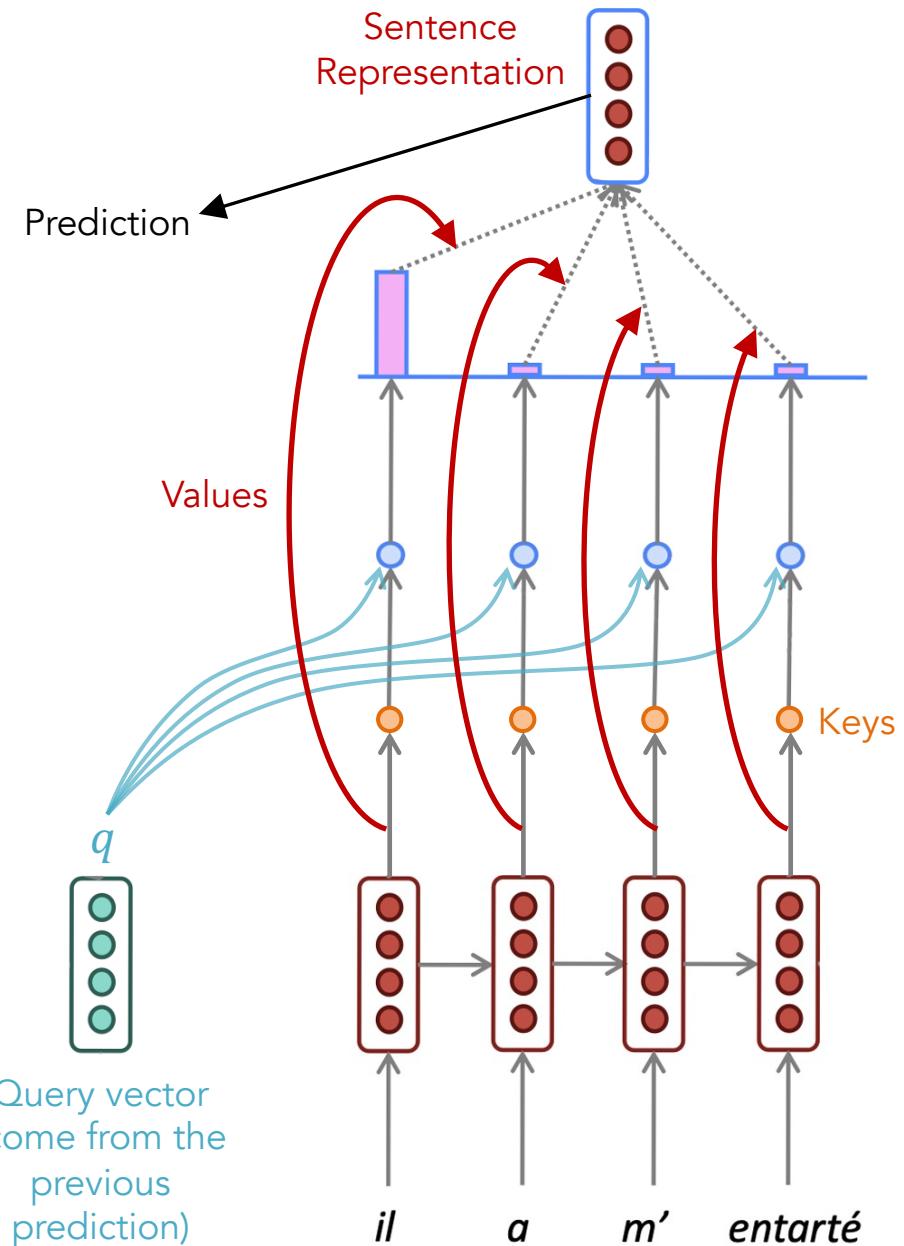
Attention is just a **weighted** average – this is very powerful if the weights are learned!

In **attention**, the **query** matches all **keys** *softly*, to a weight between 0 and 1. The keys' **values** are multiplied by the weights and summed.



In a **lookup table**, we have a table of **keys** that map to **values**. The **query** matches one of the keys, returning its value.





} Step 5: Compute attention-weighted sum of encoder output:

- $\sum_{t=1}^T a_t h_t$

} Step 4: Compute the attention distribution using softmax:

- $[a_1 \ a_2 \ \dots \ a_T] = \text{softmax}([e_1 \ e_2 \ \dots \ e_T])$

} Step 3: Compute attention scores (dot product similarity):

- $e_t = q^T u_t$ q is trainable

} Step 2: Transform encoder outputs (dimension reduction):

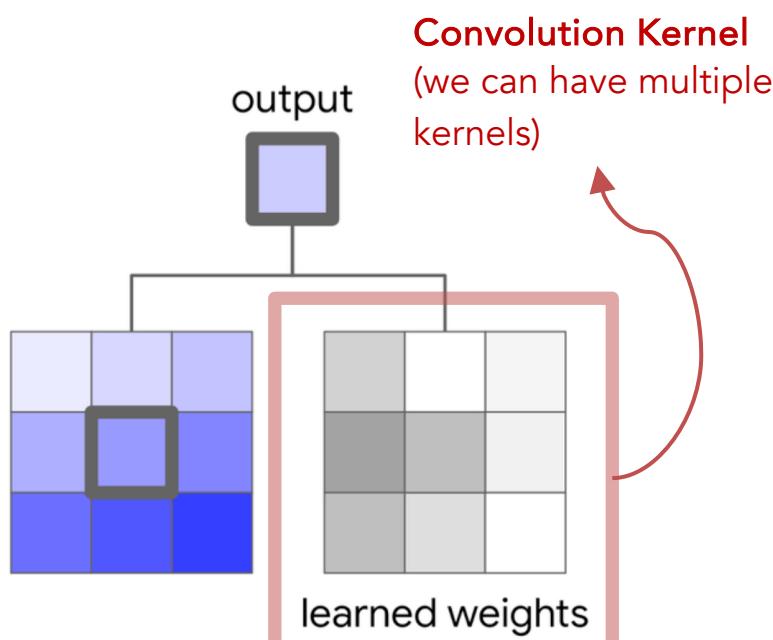
- $u_t = \tanh(W h_t)$ W is trainable

} Step 1: Get the encoder output values (from the RNN):

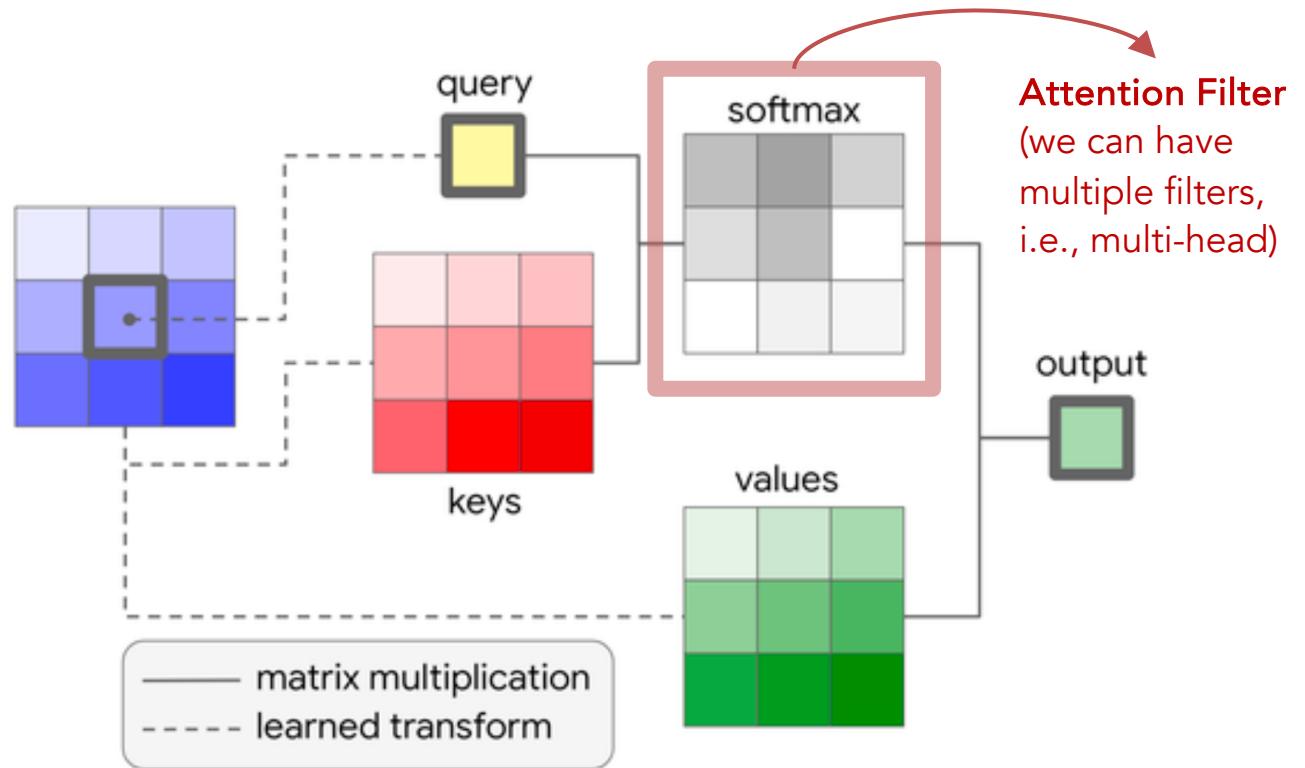
- h_t

There are many ways of doing step 2 and 3

Convolution layers use **fixed weights** (kernels) to filter information. Self-attention layers dynamically compute attention filters to show how well a pixel matches its neighbors.



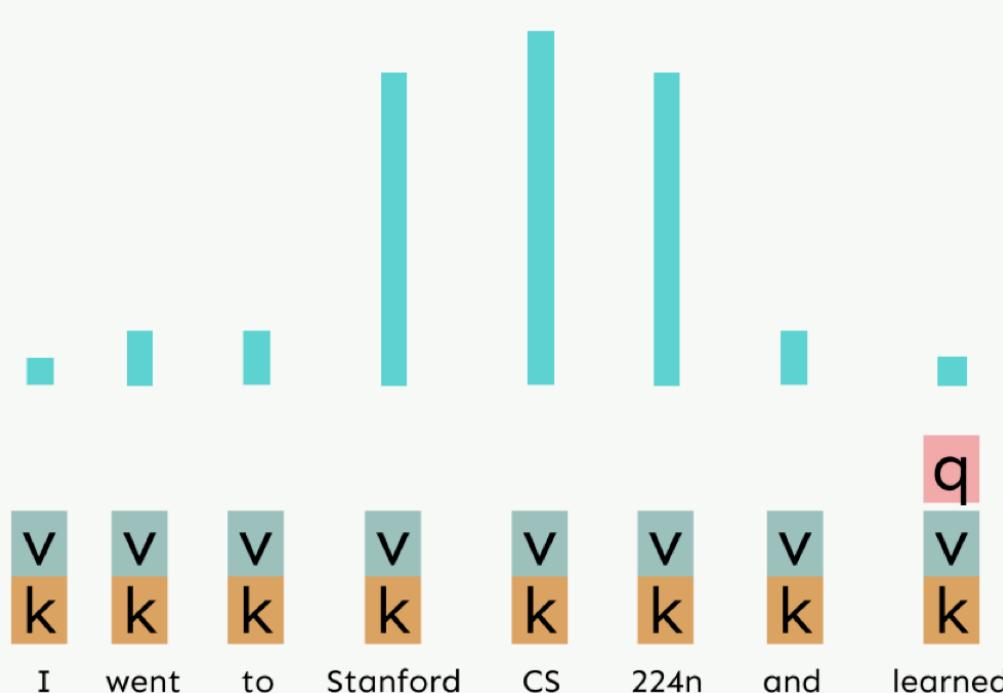
Convolution



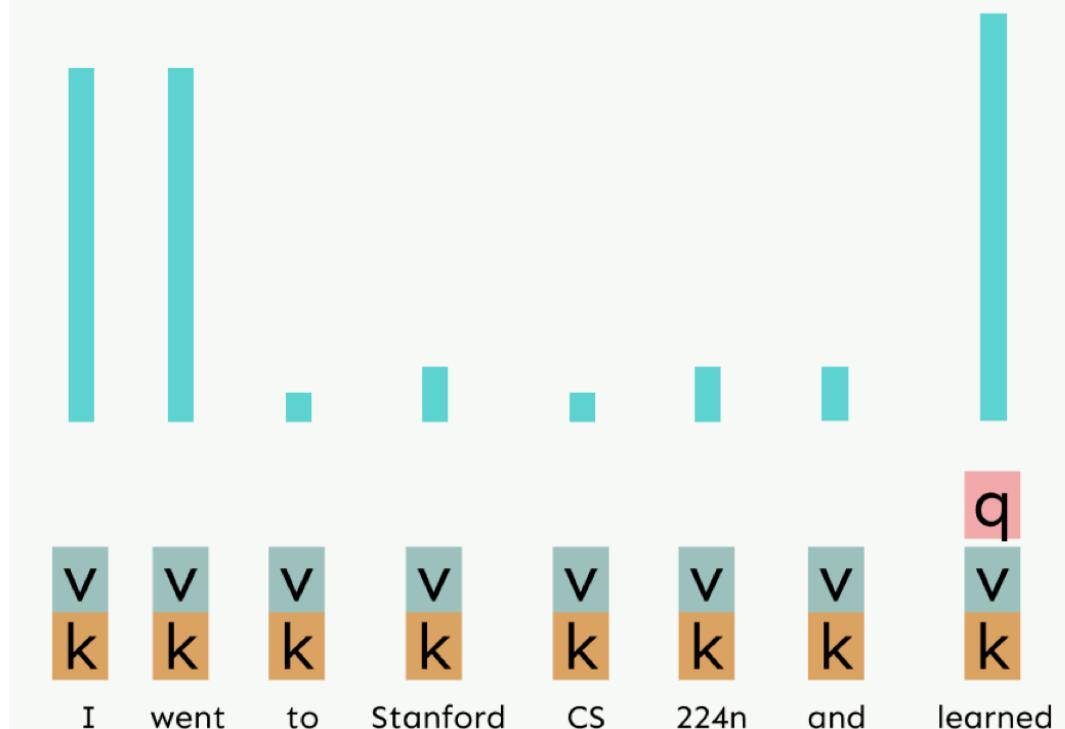
Self-attention

Transformers use **multi-head attention** to look at different aspects of the inputs.

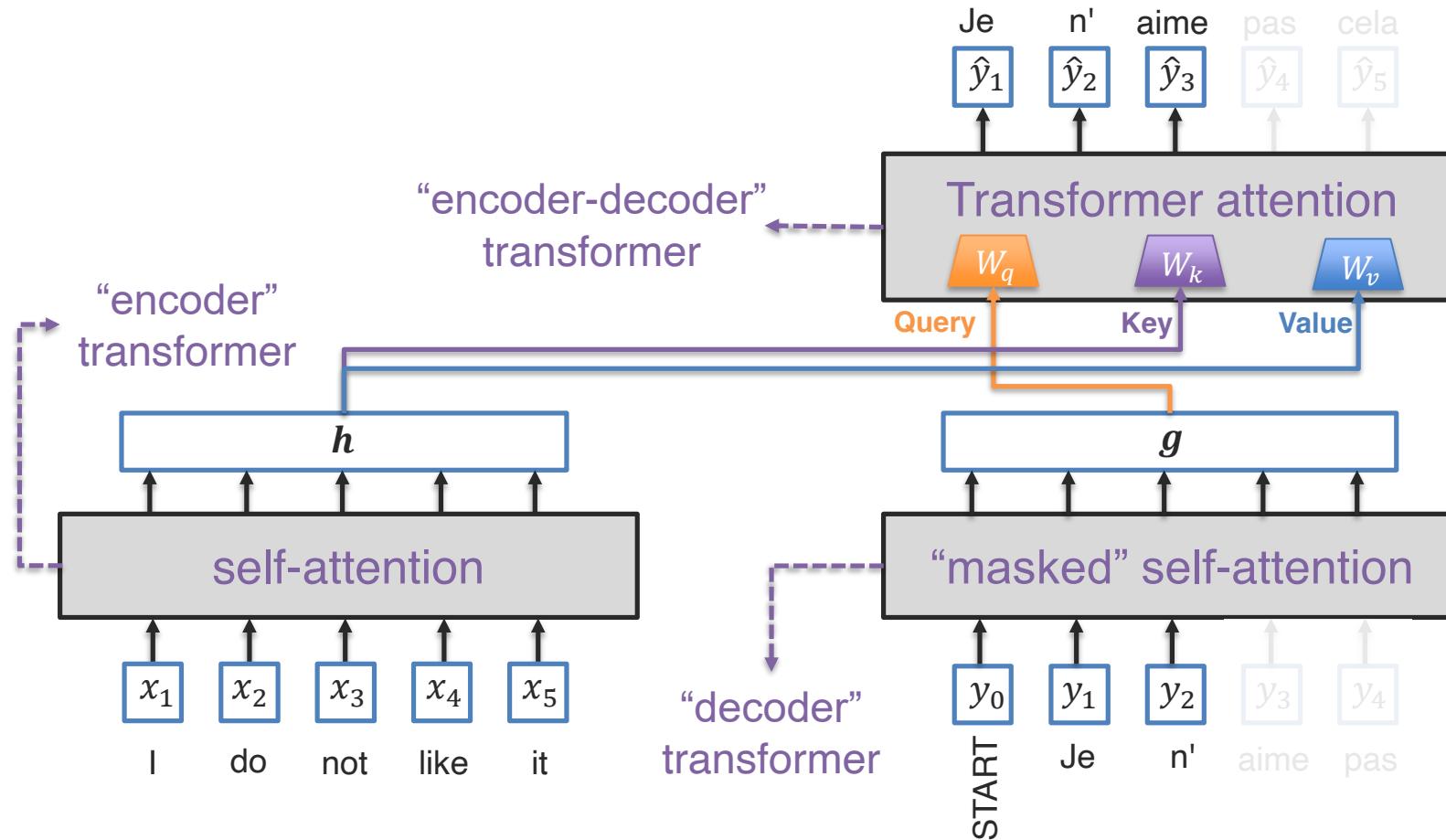
Attention head 1
attends to entities



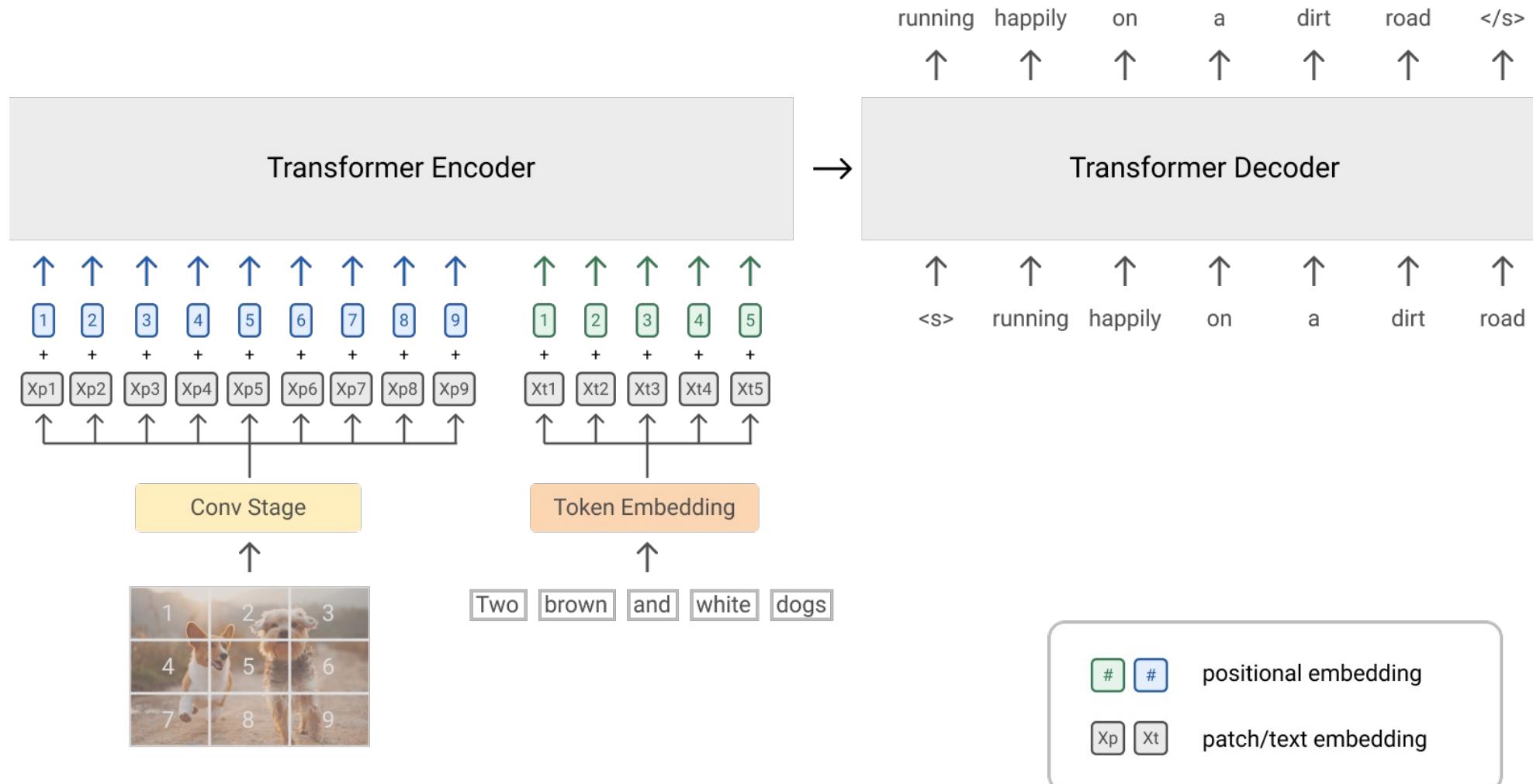
Attention head 2 attends to
syntactically relevant words



Transformers are connected by two self-attention blocks (one for encoder, one for decoder) and an encoder-decoder attention block (similar to the original attention).



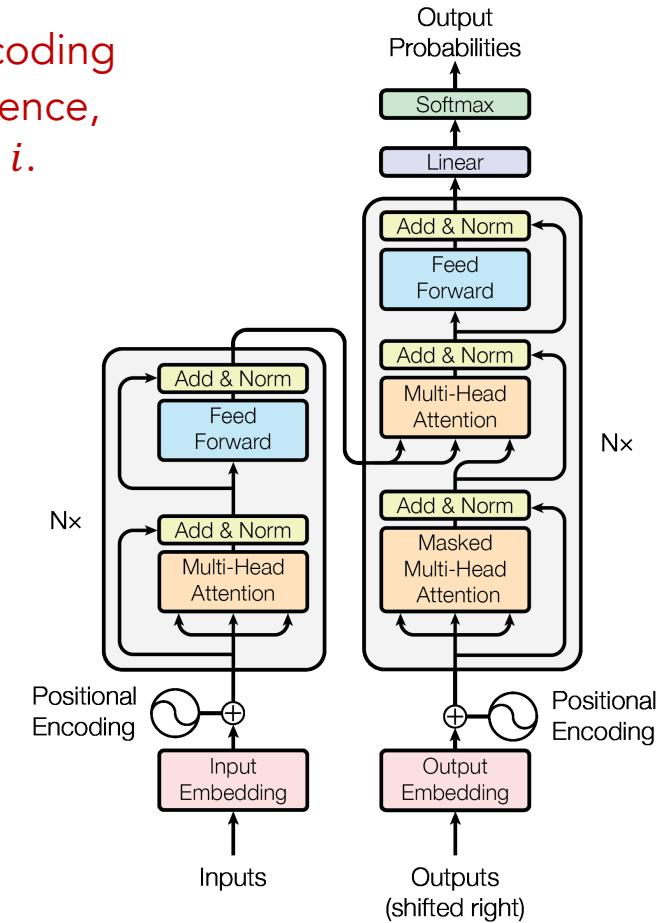
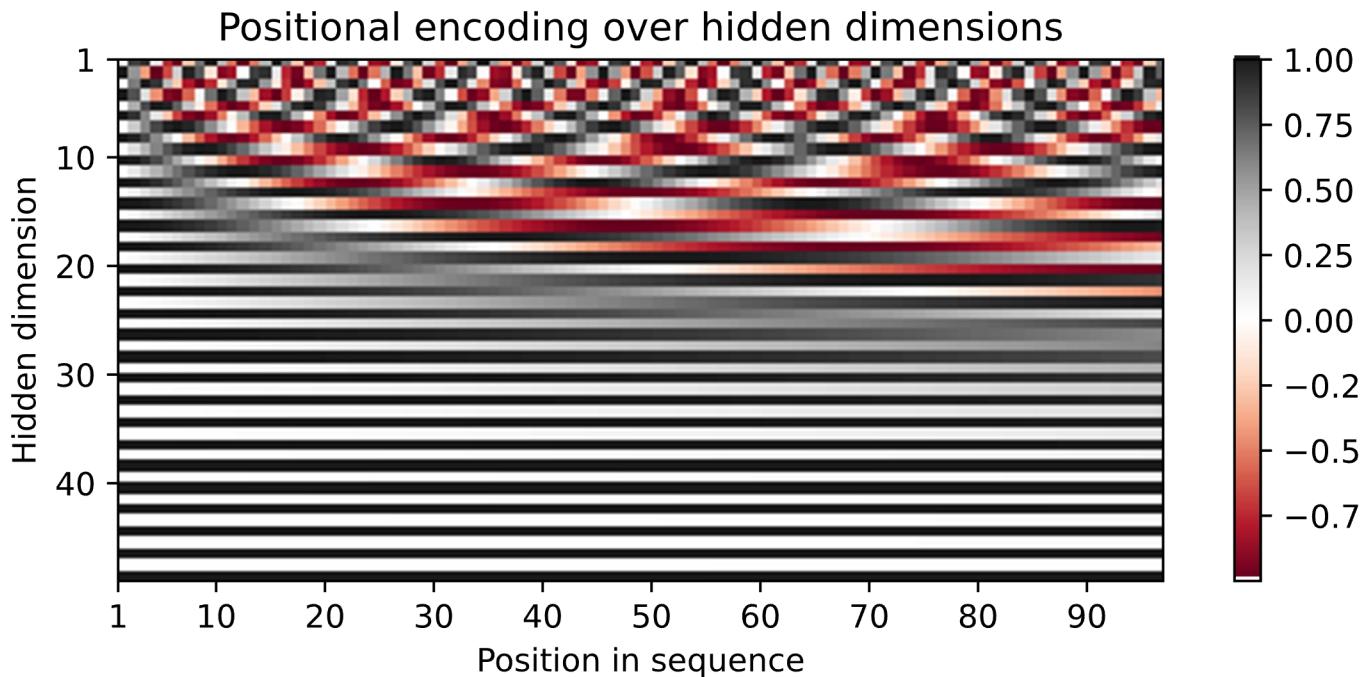
Transformers can work for multiple vision-language tasks (vision+language→language).



Self-attention is permutation invariant (looks at the input as a set of elements). We use **positional encoding** to add the position information to the input embedding vectors.

$$PE_{(pos,i)} = \begin{cases} \sin\left(\frac{pos}{10000^{i/d_{\text{model}}}}\right) & \text{if } i \bmod 2 = 0 \\ \cos\left(\frac{pos}{10000^{(i-1)/d_{\text{model}}}}\right) & \text{otherwise} \end{cases}$$

$PE_{(pos,i)}$ is the position encoding at position pos in the sequence, and hidden dimensionality i .

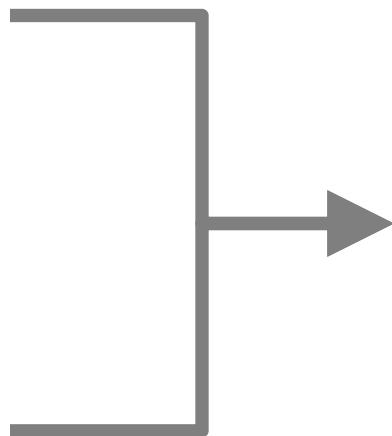




Modality A

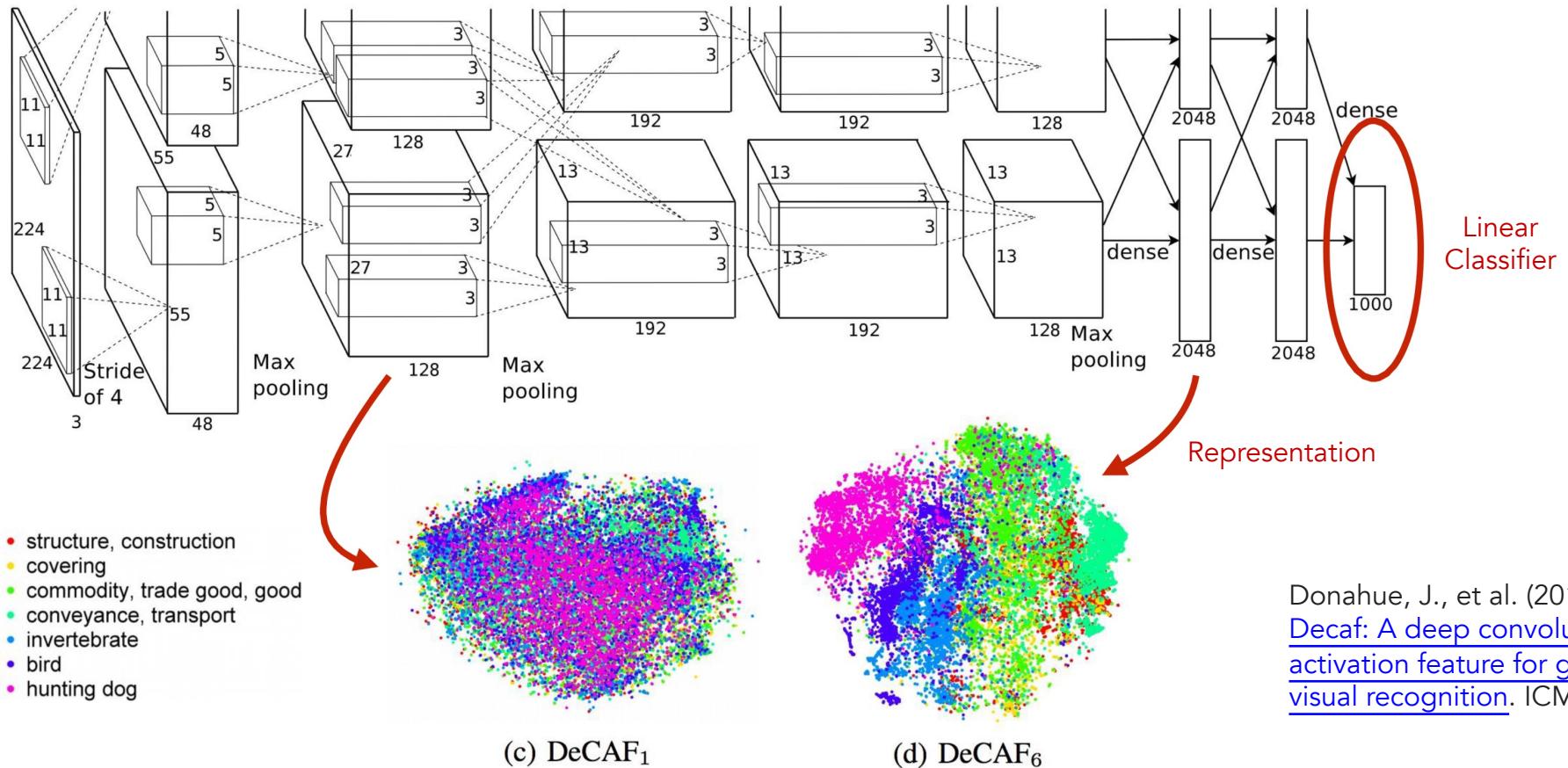


Modality B

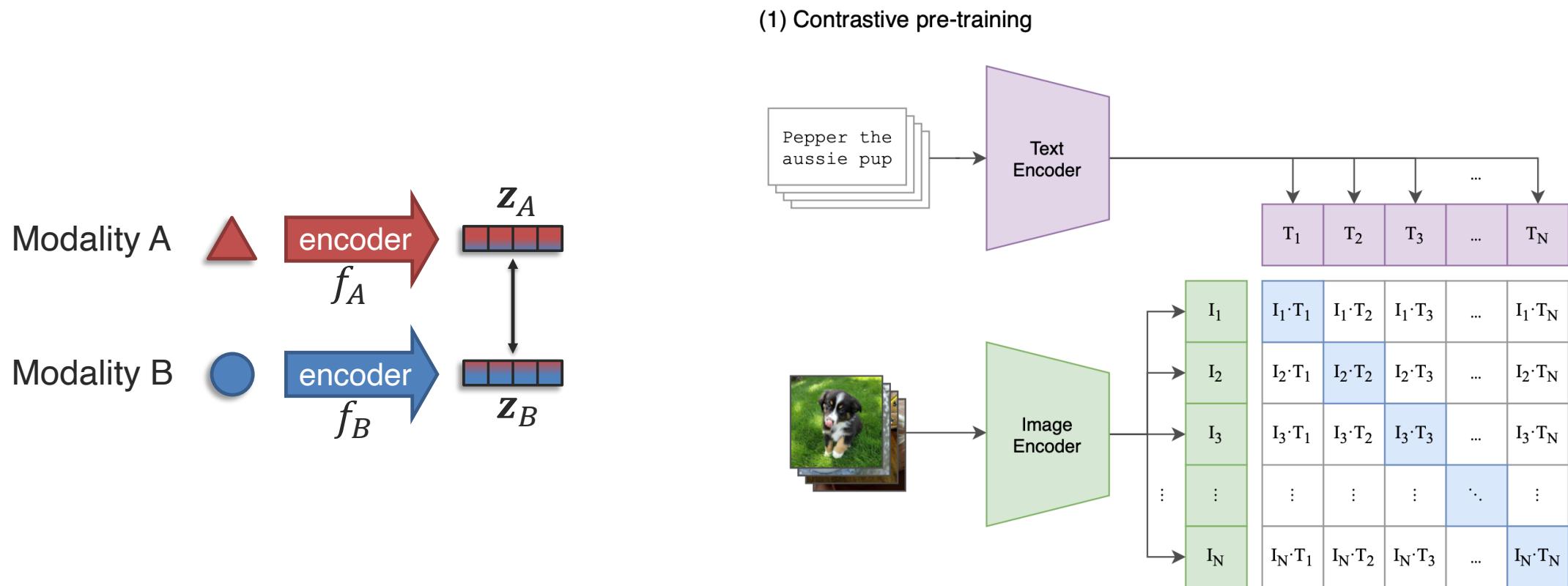


Representation

Instead of completing the task directly, we can also think about how to **learn a good representation** (i.e., embedding) so that a linear classifier can separate the data easily.

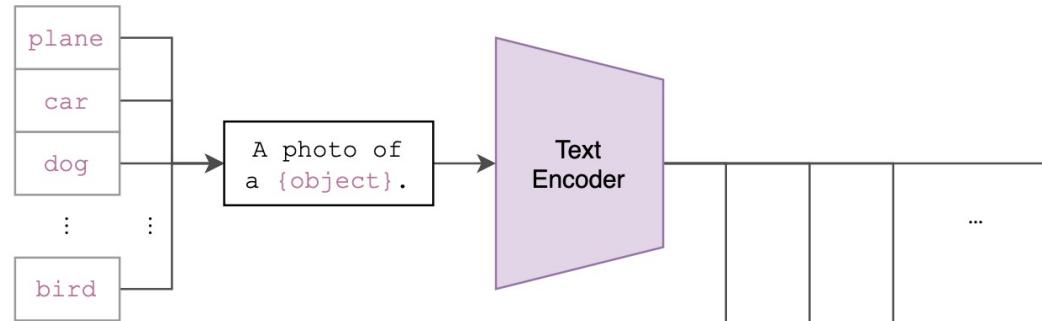


The CLIP model learns a joint text-image representation using a large number of image and text pairs (**vision+language→representation**).

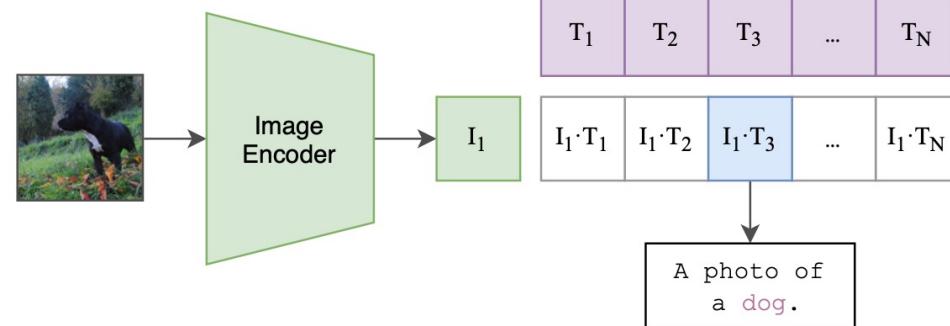


We can use the learned CLIP embedding to perform **zero-shot prediction** by taking the label with the largest similarity score between the label text and the image.

(2) Create dataset classifier from label text

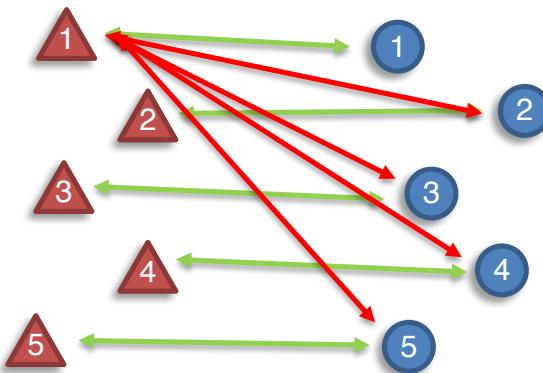


(3) Use for zero-shot prediction



Contrastive Learning brings positive pairs closer and pushes negative pairs far apart.

Paired data: $\{\triangle, \circ\}$
(e.g., images and text descriptions)



↔ Positive pairs
↔ Negative pairs

Simple contrastive loss:

$$\max\{0, \alpha + sim(\mathbf{z}_A, \mathbf{z}_B^+) - sim(\mathbf{z}_A, \mathbf{z}_B^-)\}$$

positive pairs

negative pair

Similarity functions are often cosine similarity

Popular contrastive loss: InfoNCE

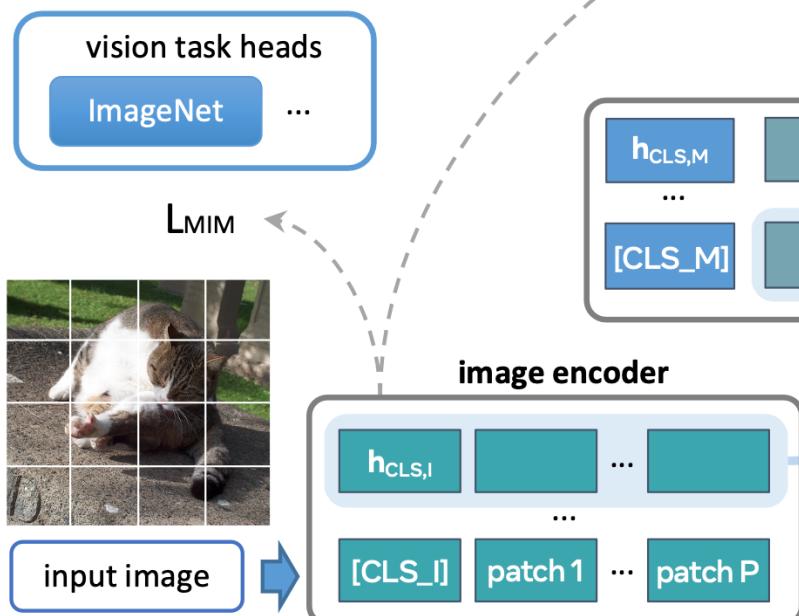
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log \frac{sim(\mathbf{z}_A^l, \mathbf{z}_B^l)}{\sum_{j=1}^N sim(\mathbf{z}_A^i, \mathbf{z}_B^j)}$$

Similarity function can be cosine similarity

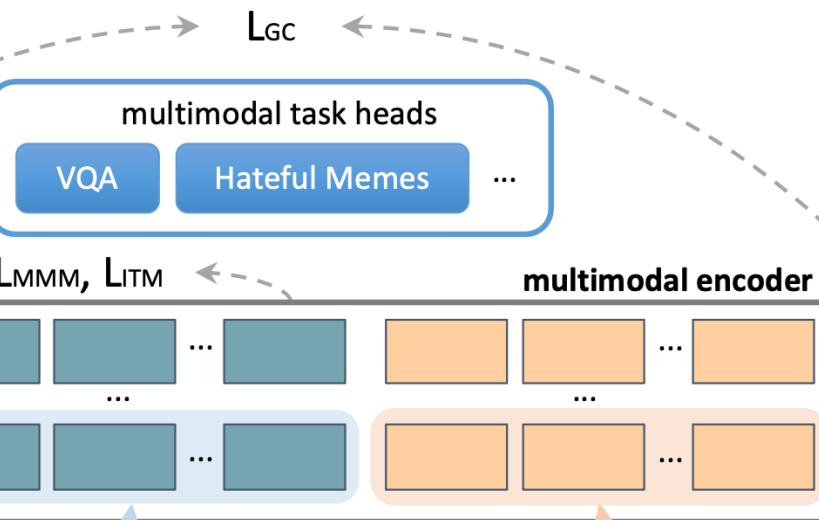
positive pairs
negative pairs and positive pairs

One active research area is **foundation models**, which work for both unimodal (e.g., image/text classification) and multimodal tasks (e.g., visual question answering).

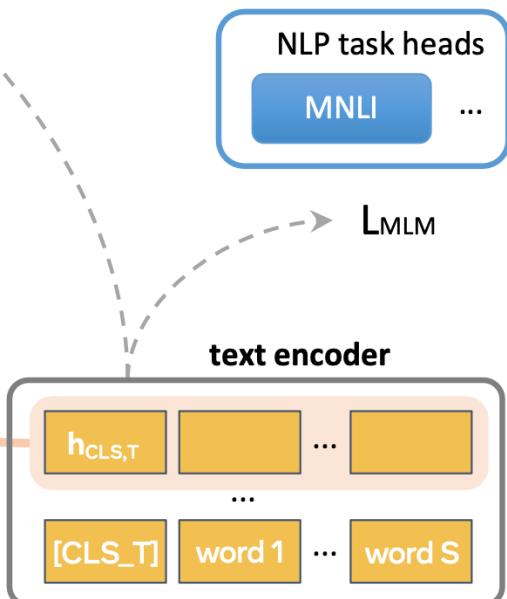
MIM, MLM, and MMM Loss:
Hide some data (i.e., masked)
and predict the hidden parts
(image, text, or both)



GC Loss: Contrastive Learning



ITM Loss: Create a set of image-text pairs and predict if an input image matches an input text



This cat was wonderful! He
was making his daily cleaning
on an ancient grave as to say
“I am the boss here!”

Take-Away Messages

- Multimodal means having multiple modalities that represent multiple natural phenomena.
- Multiple modalities can exist in different parts of the machine learning pipeline.
- We can fuse the modalities or explicitly learn their connections in the model architecture.
- Self-attention is a way of encoding sequences to tells how much attention each input should pay attention to the other inputs (including itself).
- We can also think about how to learn a good representation (i.e., embedding) so that a linear classifier can separate the data easily.
- Contrastive Learning brings positive pairs closer and pushes negative pairs far apart.



Questions?