Machine Learning - *Temporal* models (69152)

RNNs, Transformers, …

Master in Robotics, Graphics and Computer Vision

Ana C. Murillo

Universidad Zaragoza
1542

Next

- RNN exercise
- **QUESTIONNAIRES** - https://encuestas.unizar.es/english-questionnaires
- Other *temporal* models
  - TCN
  - Transformers

Other *temporal* processing: TCN

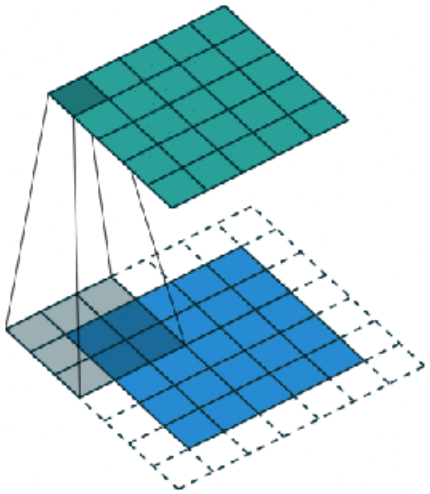
- TCN - **Temporal Convolutional Network**
  - Causal CNN / Convolutional Markov Model

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

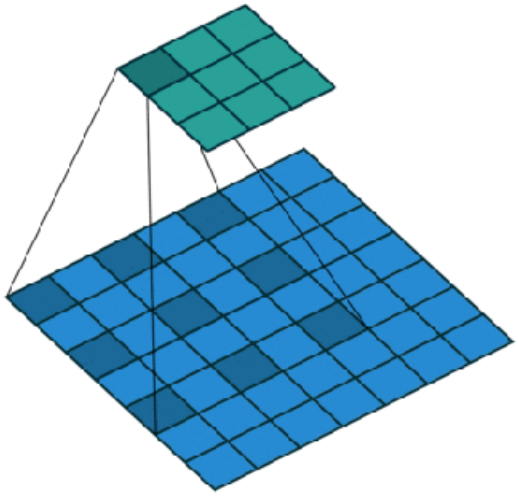
Each sample is conditioned on samples at all previous t

Other *temporal* processing: TCN

- TCN - Temporal Convolutional Network
  - Causal CNN / Convolutional Markov Model
  - 1D s



standard convolution



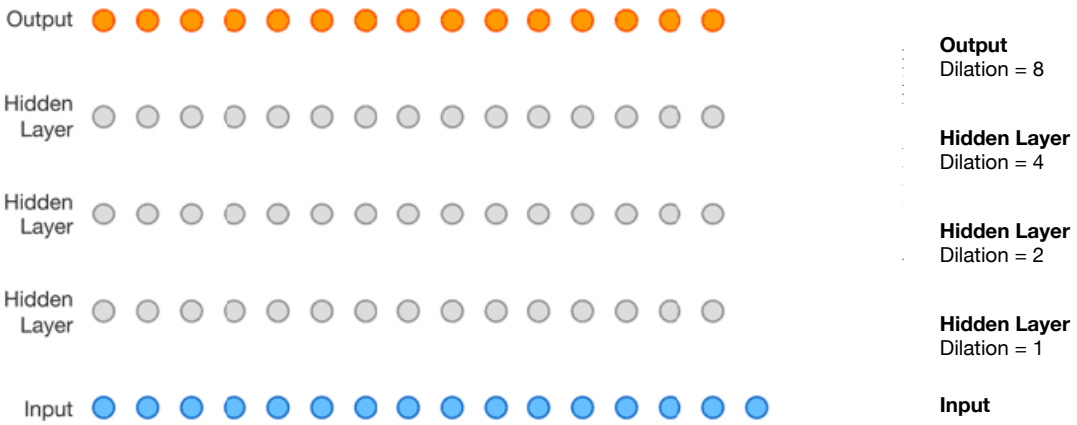
dilated convolution

https://github.com/vdumoulin/conv_arithmetic

Other *temporal* processing: TCN

- TCN - Temporal Convolutional Network
  - Causal CNN / Convolutional Markov Model
  - 1D dilated convolutions
  - **WaveNet**: stack of dilated causal 1D conv layers

https://deepmind.com/blog/article/wavenet-generative-model-raw-audio



Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

Other *temporal* processing

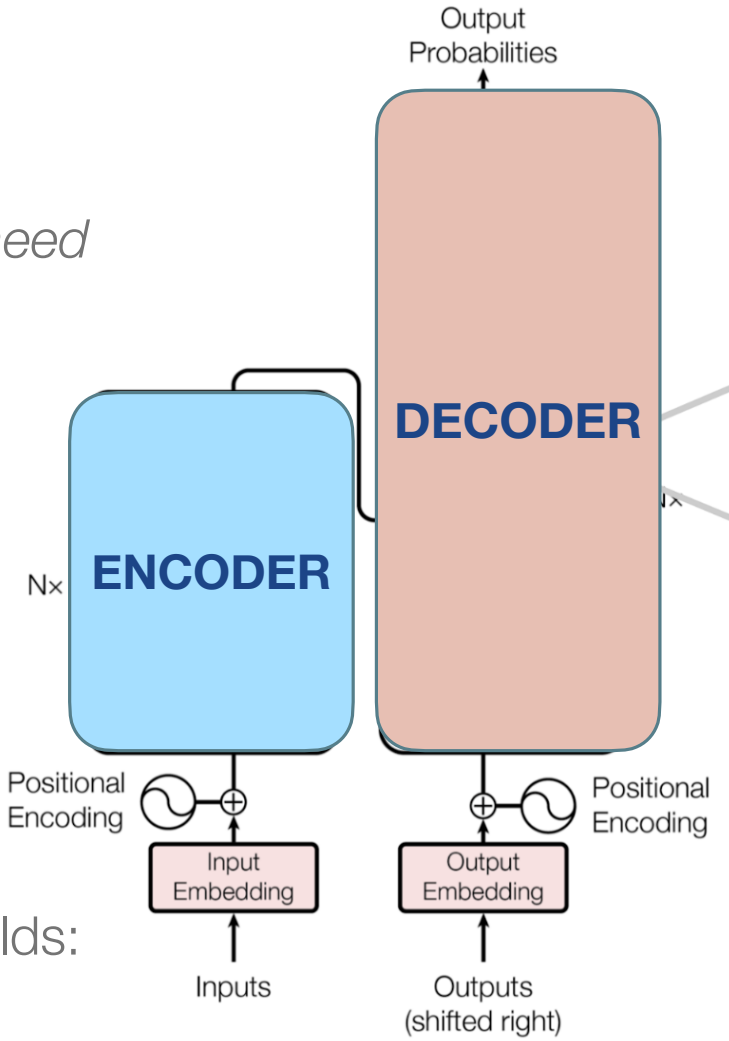
- **Transformers** - Attention
  - *Attention is all you need*

A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers

Transformers: *Attention is all you need*

- Based on *self-attention:* similar to FC layer, but instead of fixed weights, they are adapted depending on the input
- Usually multi-head: multiple sets of learned weights
- Dominant paradigm in many fields: NLP, Computer Vision, …



A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

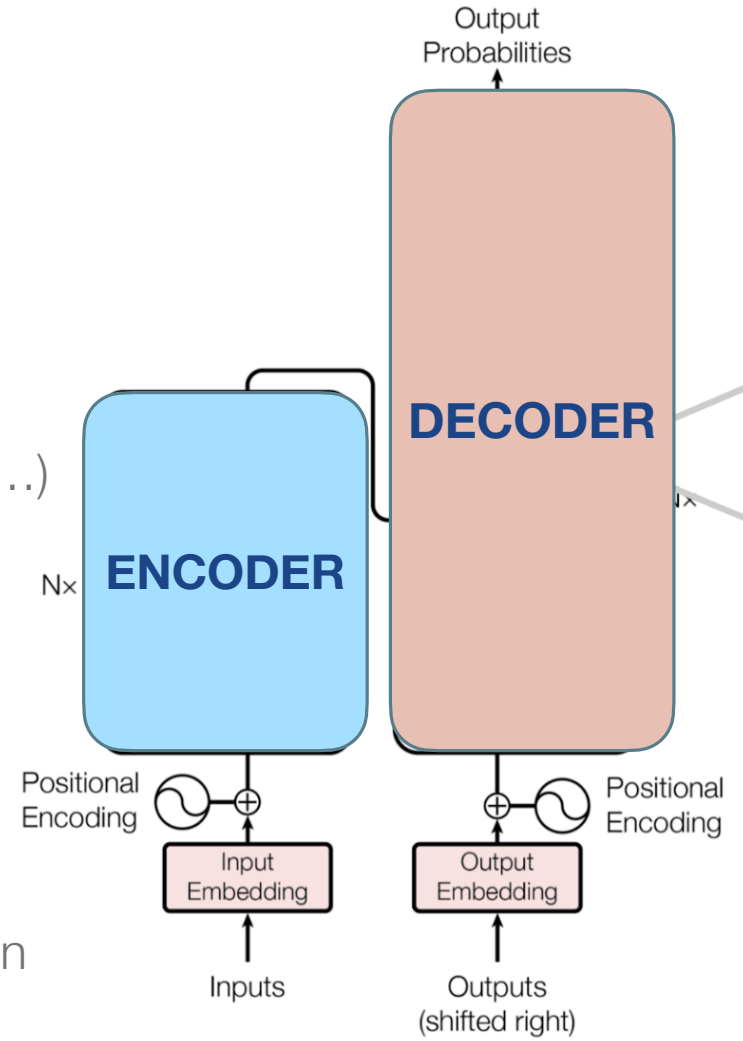
Tranformers

Encoder models (e.g. **BERT**)

- Embeddings from input tokens

Decoder models (e.g. **GPT, GPT-2**, …)

- Left to right autoregressive generation
- Decoder won't know the future during inference —> attention can't use future tokens during training (use mask attention to ignore them)

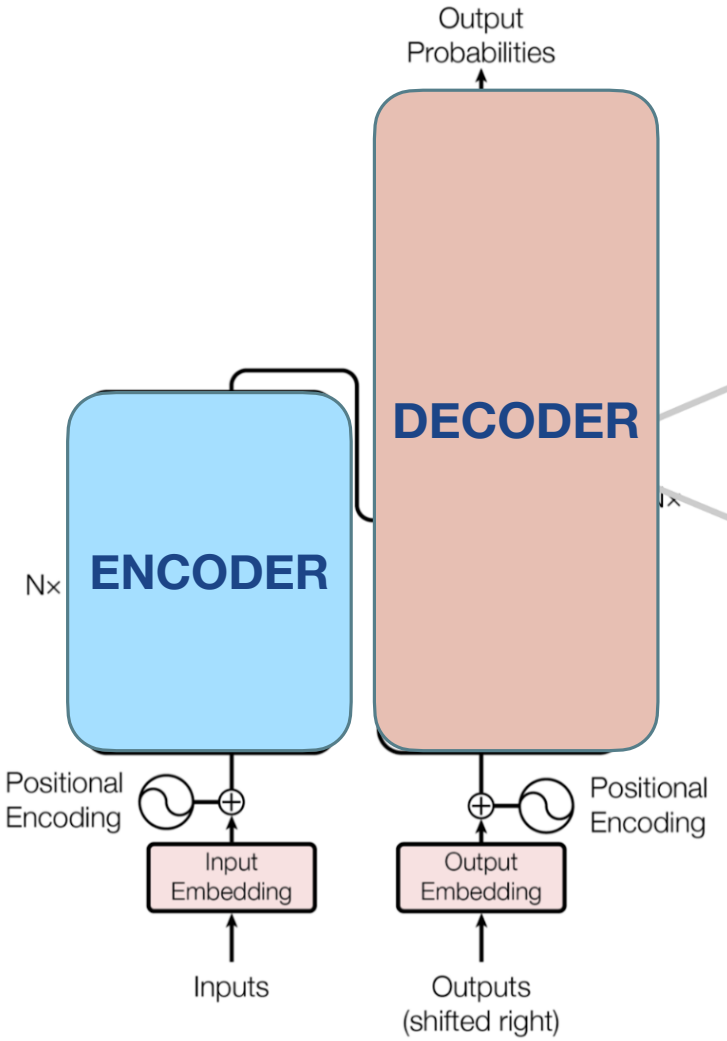


A. Vaswani et al. "Attention Is All You Need". In: NIPS. 2017.

Transformers

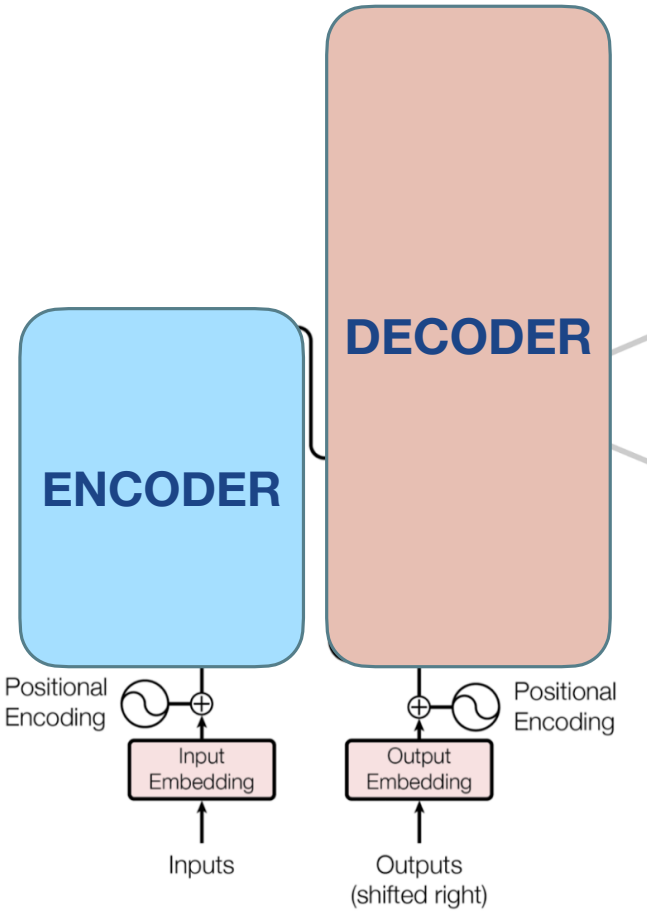
Encoder-Decoder models (T5)

- Seq2seq models
- *Cross attention:*
  Queries come from the previous decoder layer, but Keys and Values come from encoder output



A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers



Query? Key? Value?

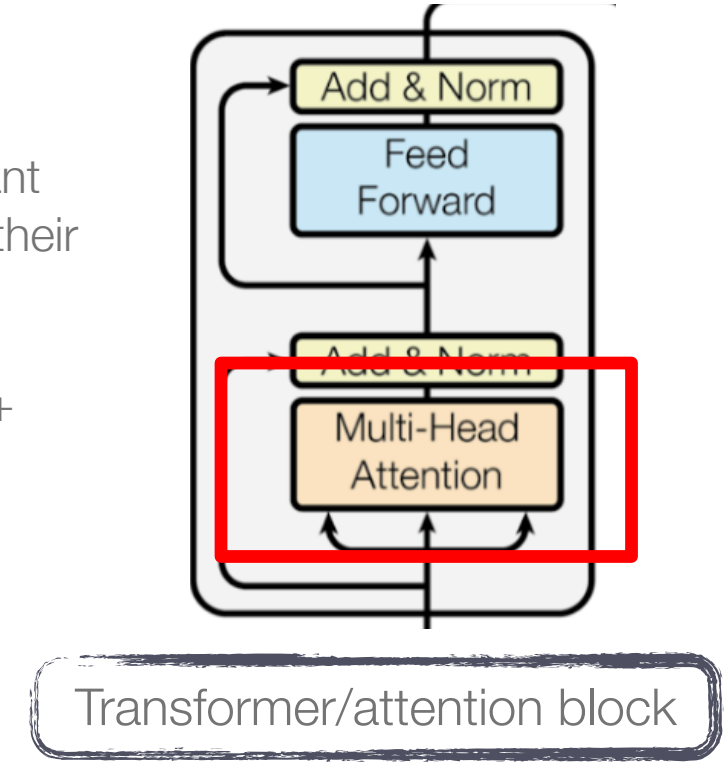
A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers: *transformer or attention block*

GOAL: augment each token representation based on relationship to other tokens

For each token (represented with a **KEY**, **QUERY**, and **VALUE**):

- Use **QUERY** to find the most relevant token(s) in the sequence based on their respective **KEY**(s)
- Calculate *importance* (dot product + softmax)
- *Weighted sum* of relevant token(s) **VALUE**(s) based on importance:
  —> new token representation

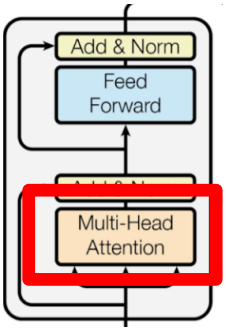


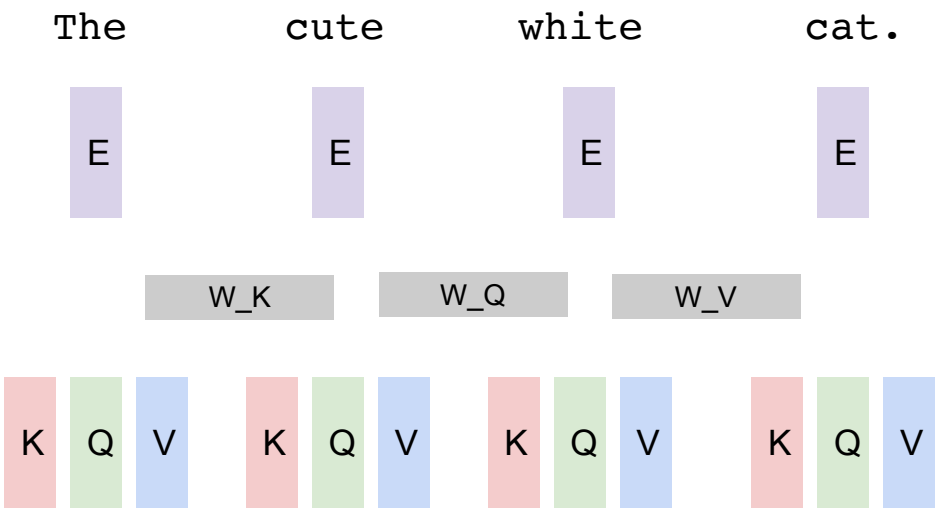
A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers: attention block

- **KEY**, **QUERY**, and **VALUE** representations are linear projections of initial token representation

- token embedding vector is multiplied matrices W_K, W_Q and W_V respectively (W are learned parameters)

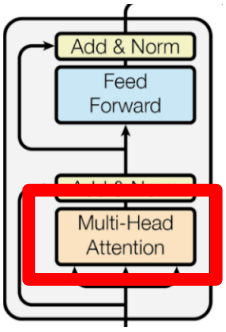




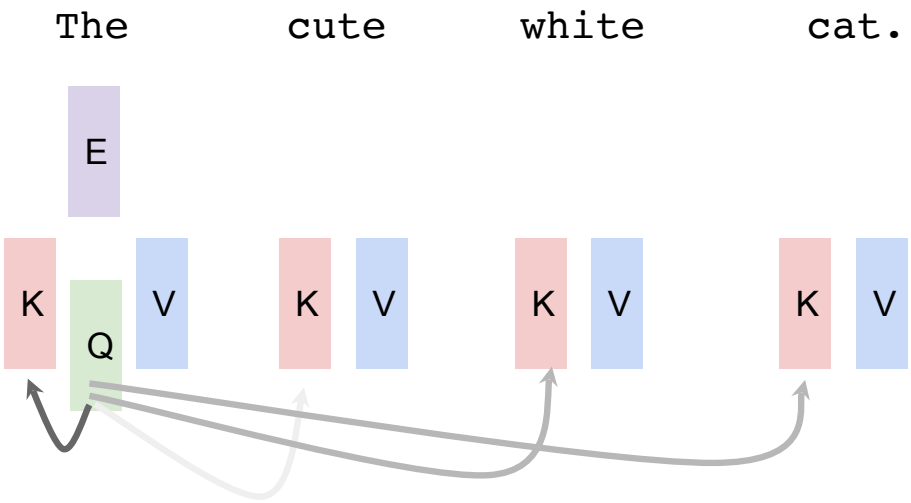
A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers: attention block

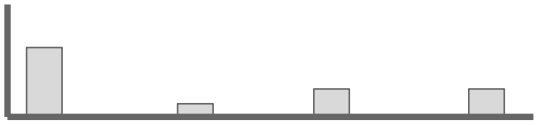
Example: representation for the first token



- *Most relevant tokens* based on Q: Dot products between **Q and Ks**, and *Softmax* on the dot product results

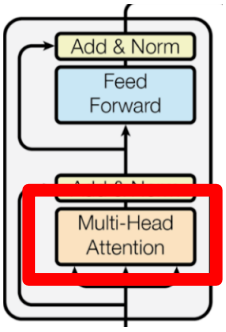


Relevance weight for each input token

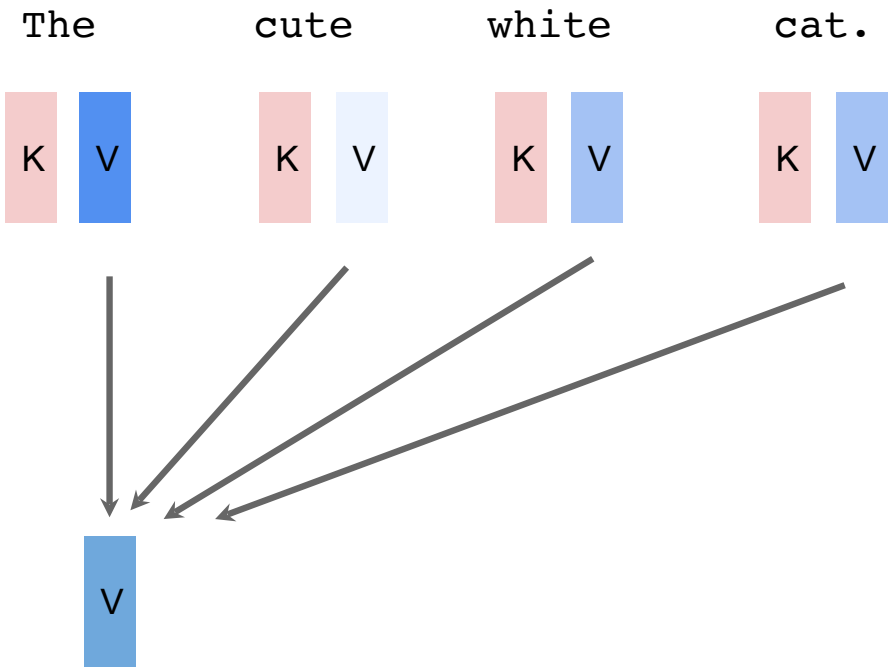


Transformers: attention block

Example: representation for the first token



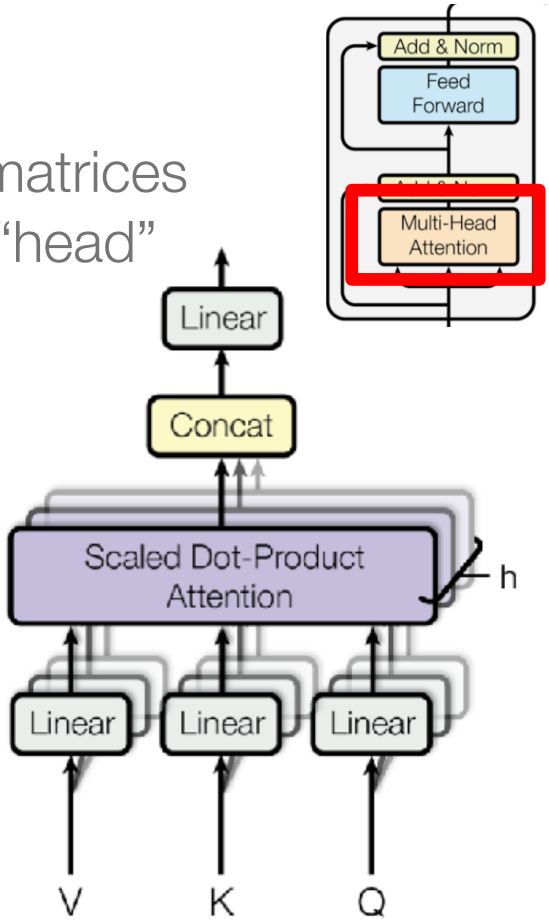
- *Weighted sum* of **Vs**



Transformers: attention block

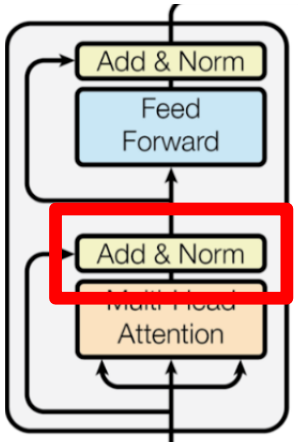
Multi-head Attention (h “heads”):

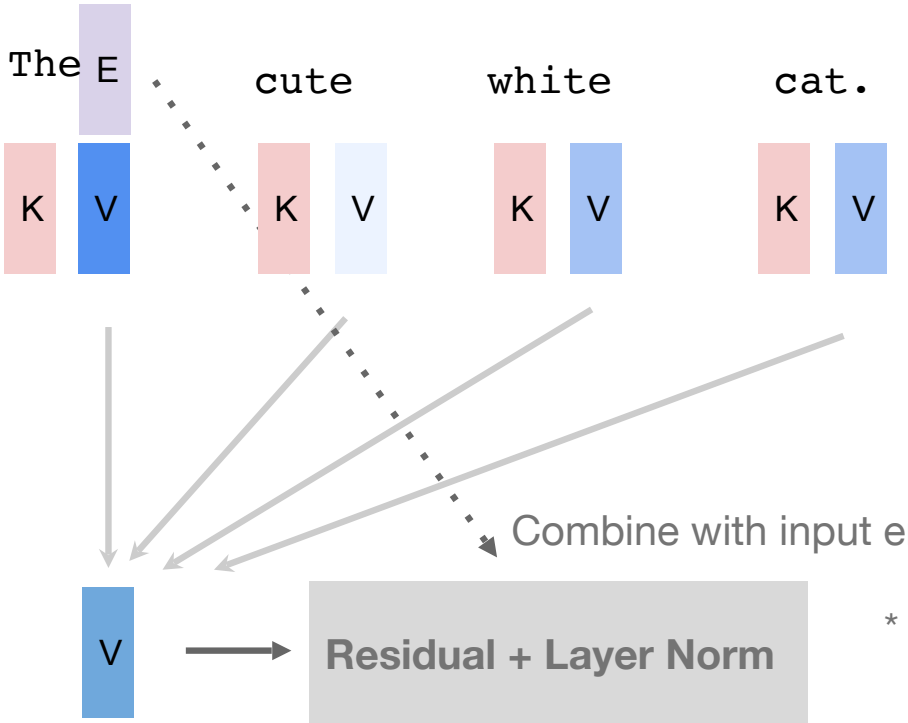
- Use different sets of projection matrices (W_K, W_Q and W_V), one per “head”
- Process each “head”, in parallel
- Concatenate all h values and re-project them to get the final embedding



Transformers: attention block

Add & Norm: Residual connection
+ Layer Normalization*



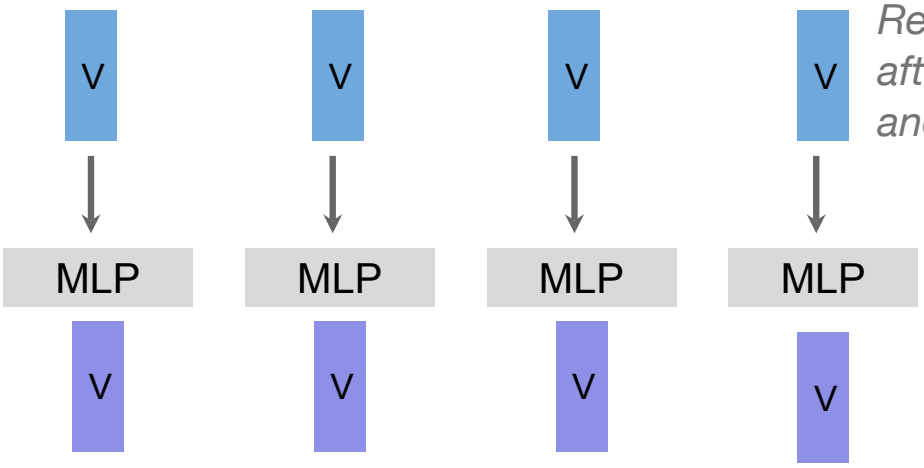


Combine with input embeddings (Residual)

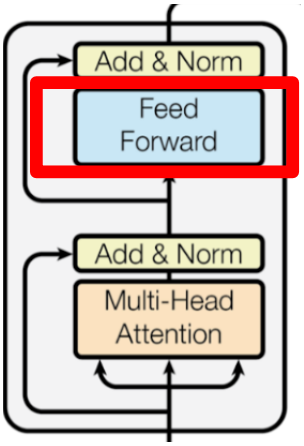
* batch normalization with statistics computed for each sample

Transformers: attention block

Feed Forward: Two layer MLPs applied individually to each token embedding



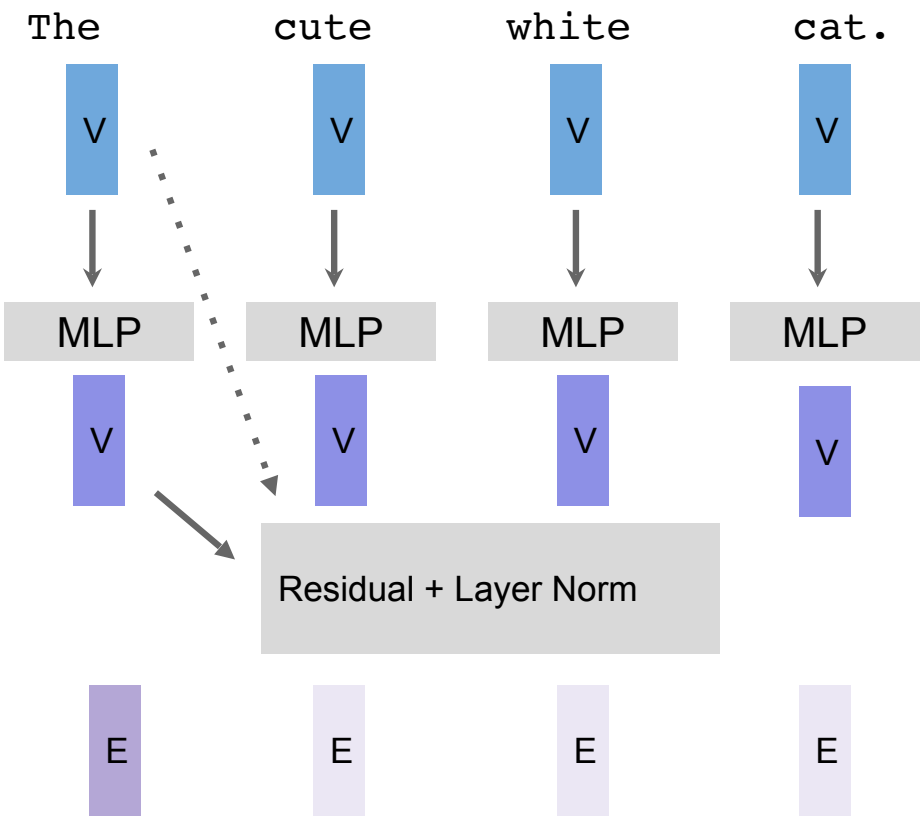
Representations after Multi-Head and Add&Norm

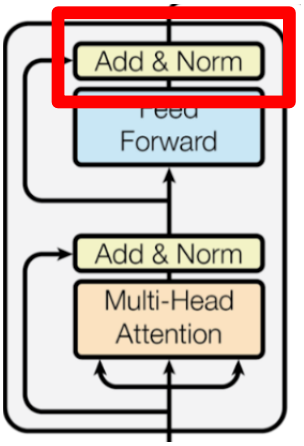


- Attention: “recombining” input vectors
- MLP: adds non-linearities and linear layers to “mix” channel values

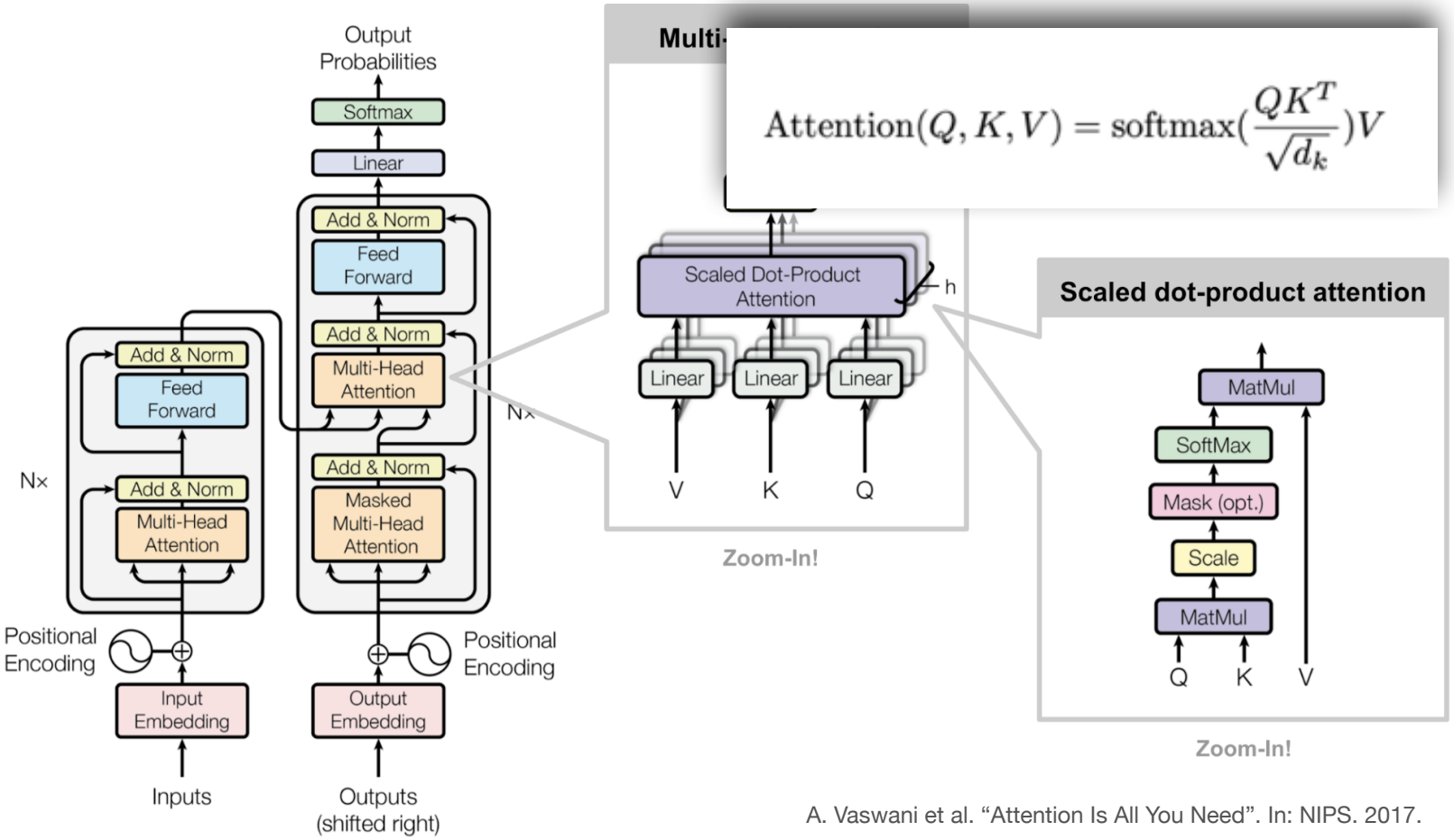
Transformers: attention block

Add & Norm:





Transformers: attention block



A. Vaswani et al. “Attention Is All You Need”. In: NIPS. 2017.

Transformers: attention block

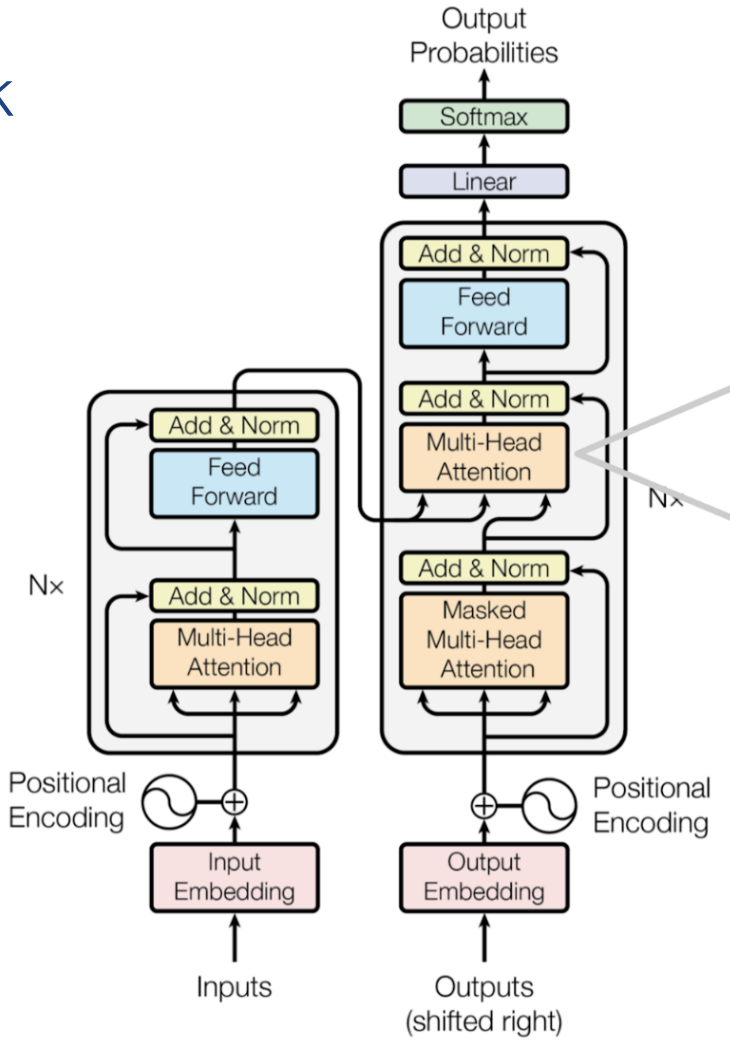
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- Attention *is just* really large matrix multiplications!
- Many operations …
- … but super parallelizable on GPUs/TPUs

Transformers: attention block

- "***self-attention***": keys and values are produced from the same source as queries.

- "***cross-attention***": keys and values come from some other, external source (e.g. an encoder module), than the queries

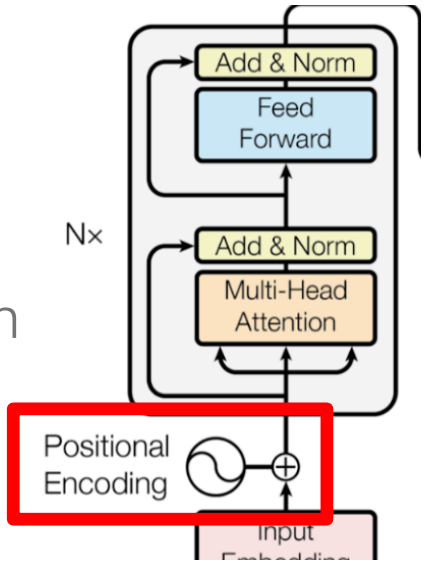


Transformers

Positional Embedding

Add position embeddings to the input so each token can keep track of its index in the sequence. Different variations:

- Sinusoidal embeddings (a vector based on a fixed equation)
- Learned embeddings



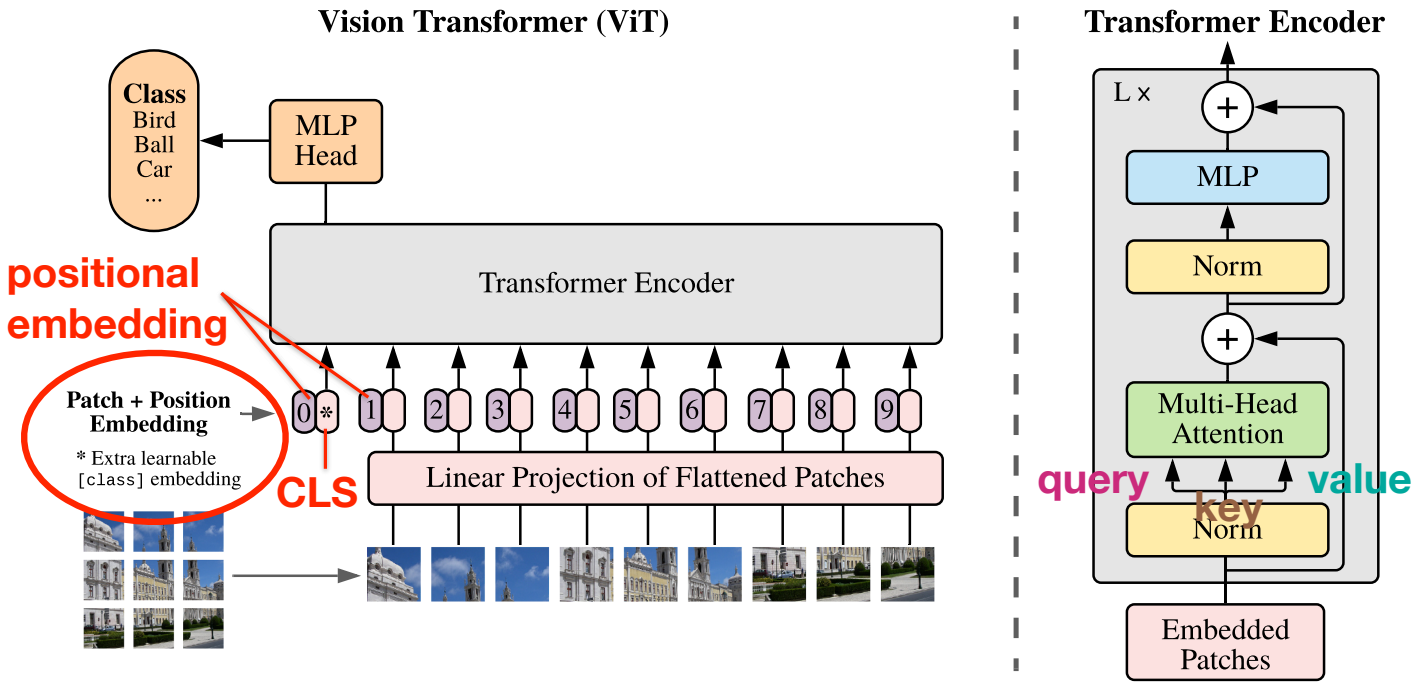
Transformers

CLS Token

- Often have a dummy token attached to beginning of input
- Use the output representation of that token for classification/other downstream tasks (e.g. pass it into a feed forward network to predict)

Visual Transformers

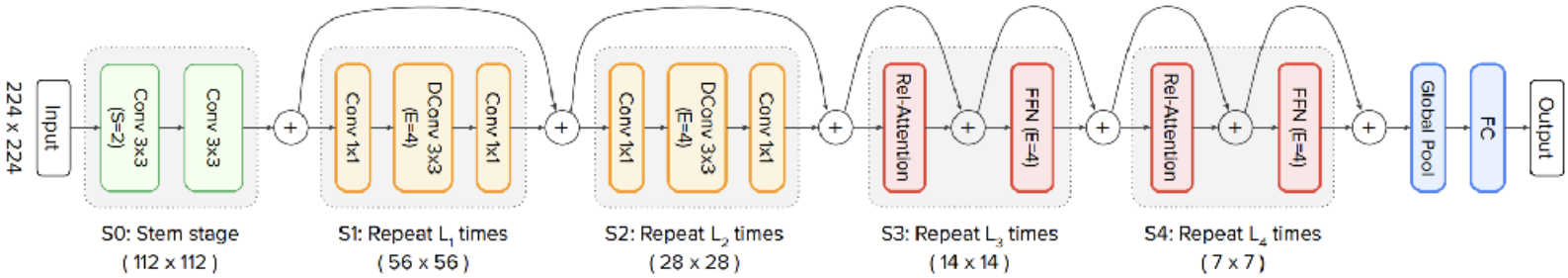
- The **Vision Transformer**: patches + positional encoding



Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: ICLR. 2021.

Visual Transformers

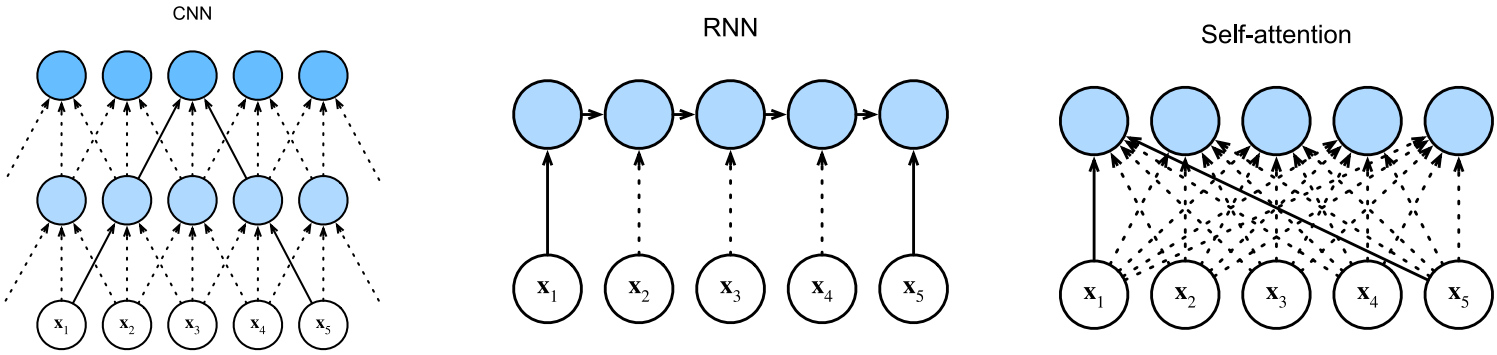
- The **Vision Transformer:** later architectures used a conv layer instead of patches + linear projections for token embeddings
  - Transformers applied later, where the input is smaller
  - Input dimension to each block is downsampled



"CoAtNet: Marrying Convolution and Attention for All Data Sizes"

Transformers vs CNN vs RNN

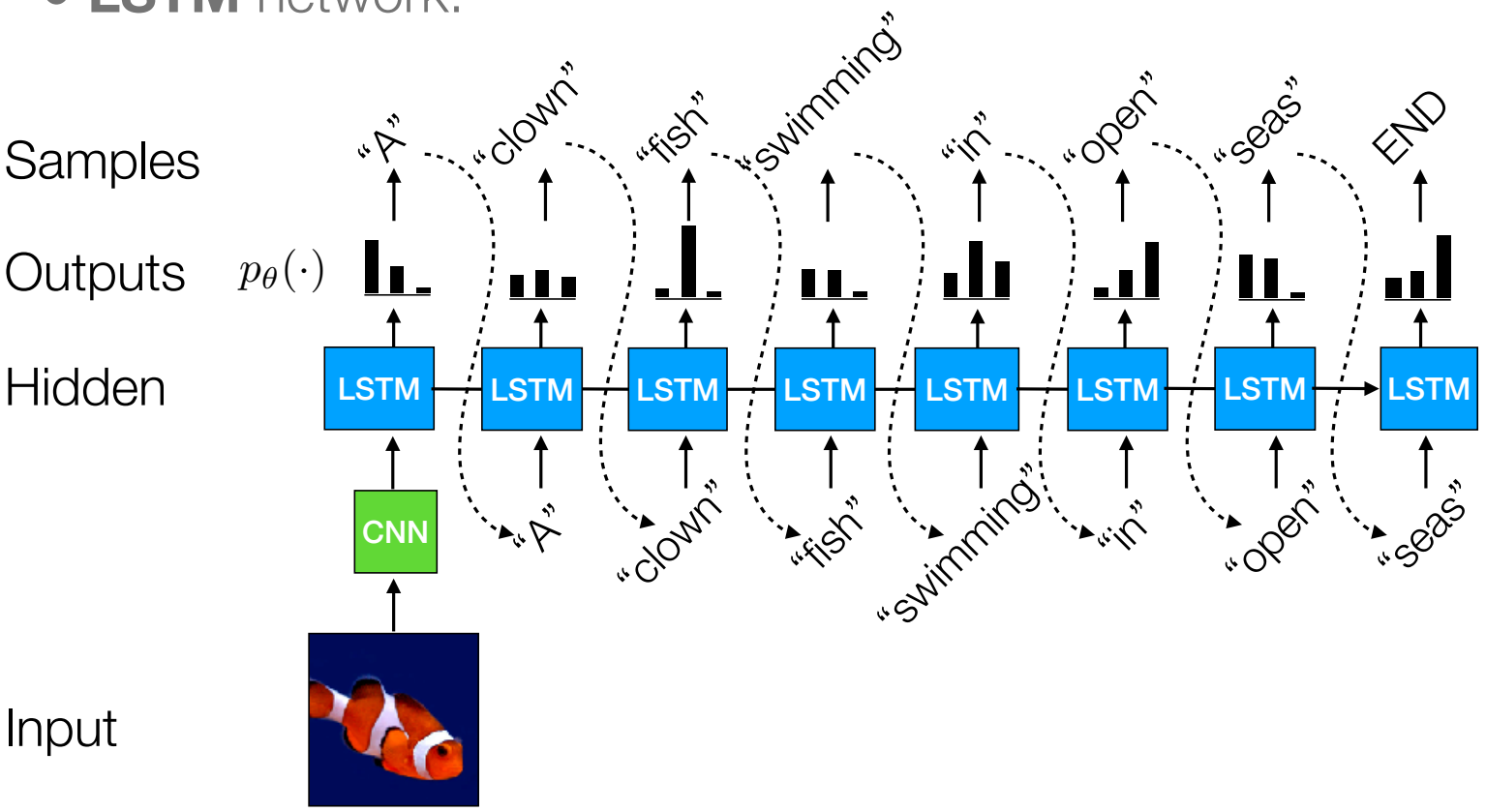
- ***1D CNN - RNN - Self-attention***
  - different tradeoffs speed - expressive power (maximum path length between any two inputs)
  - self-attention: every output is connected to every input
  - Differently from RNN, Transformers can process all input at once (instead of seq.)
  - Differently from CNN, Transformers do not have certain “bias” (as the assumptions encoded in CNNs about *nearby* more important than *further away*)



A. Zhang, Z. Lipton, M. Li, and A. Smola. Dive into deep learning. 2020.

Transformers vs RNN

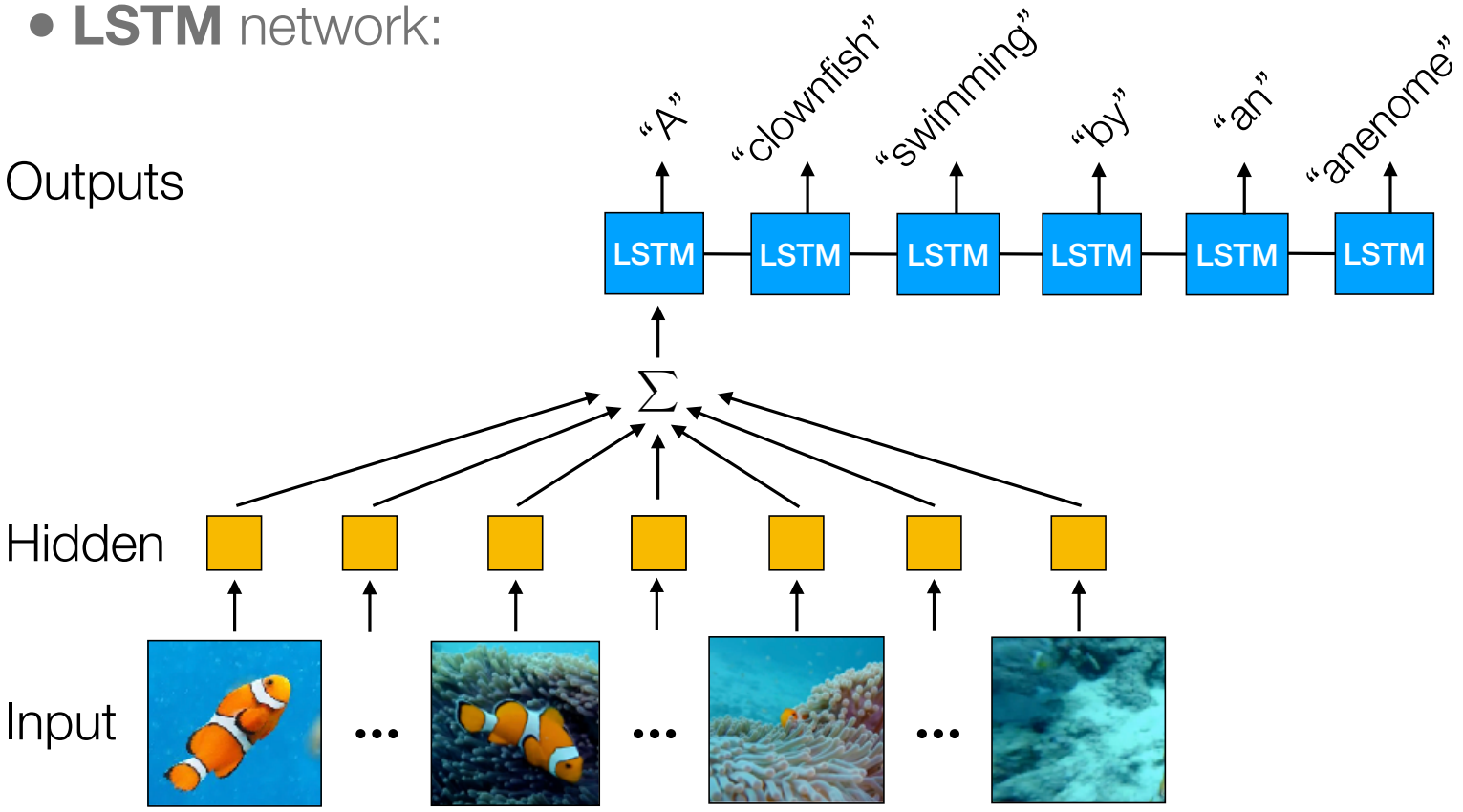
- **LSTM** network:



slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

Transformers vs RNN

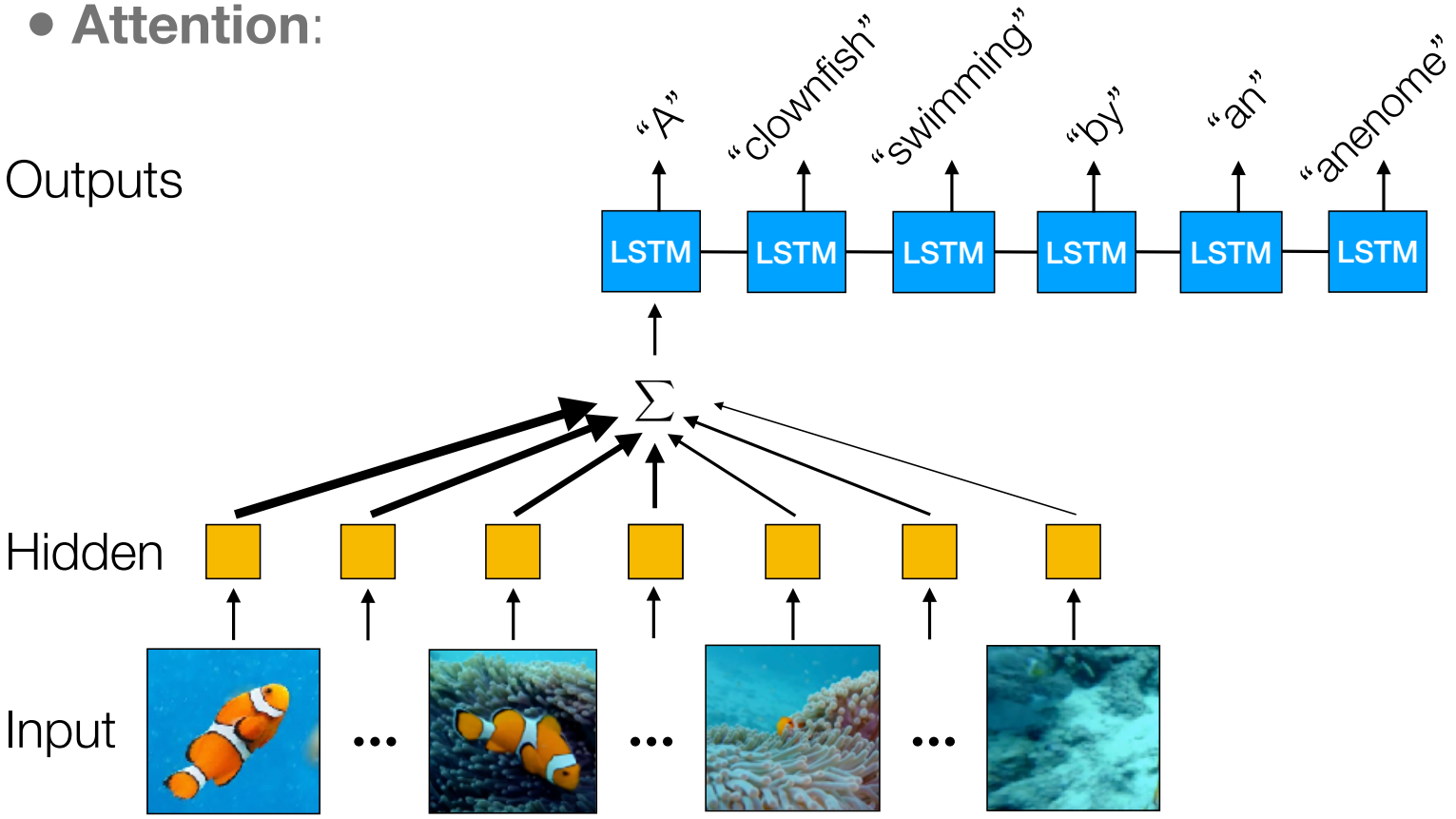
- **LSTM** network:



slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

Transformers vs RNN

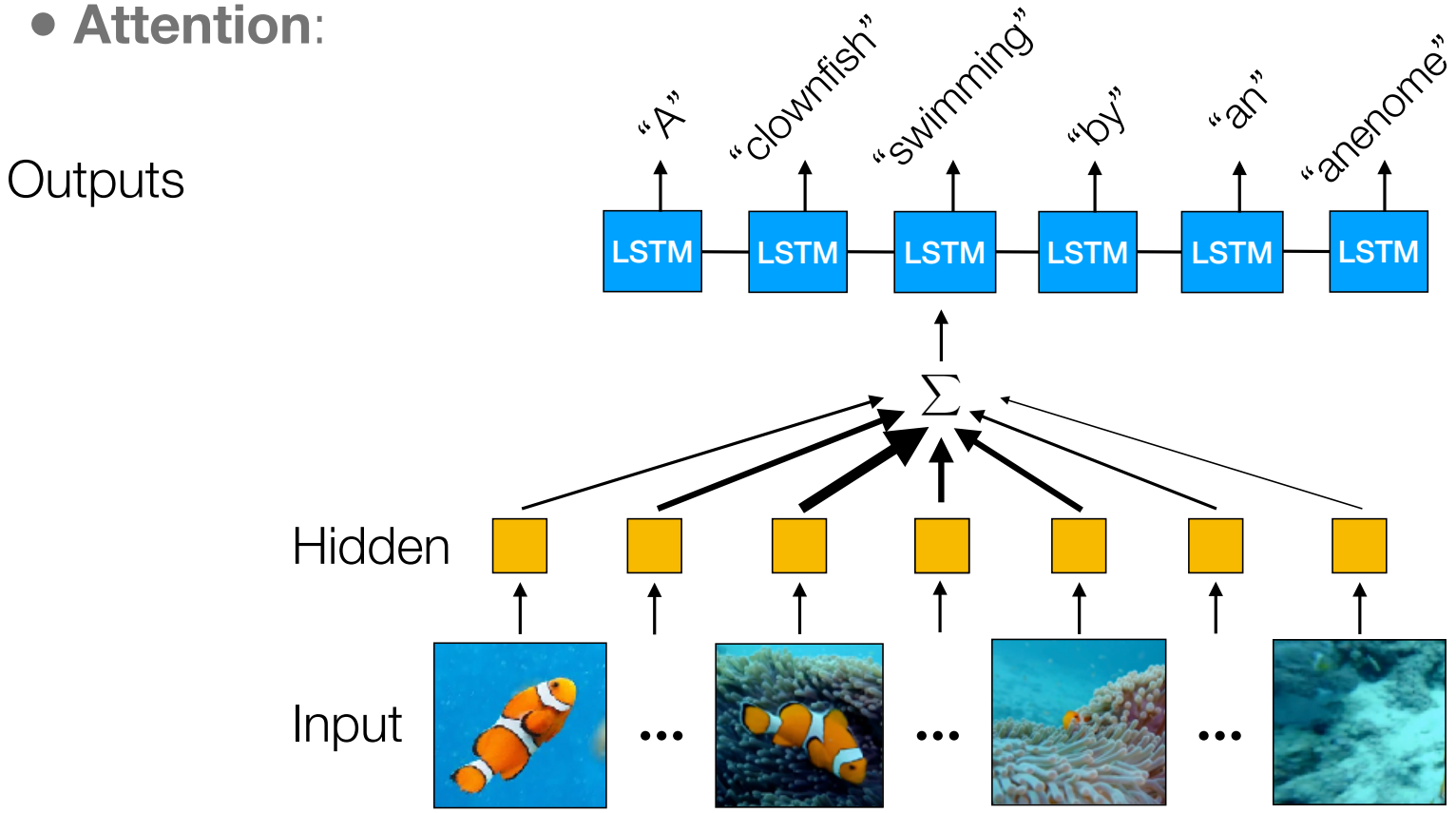
- **Attention**:



slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

Transformers vs RNN

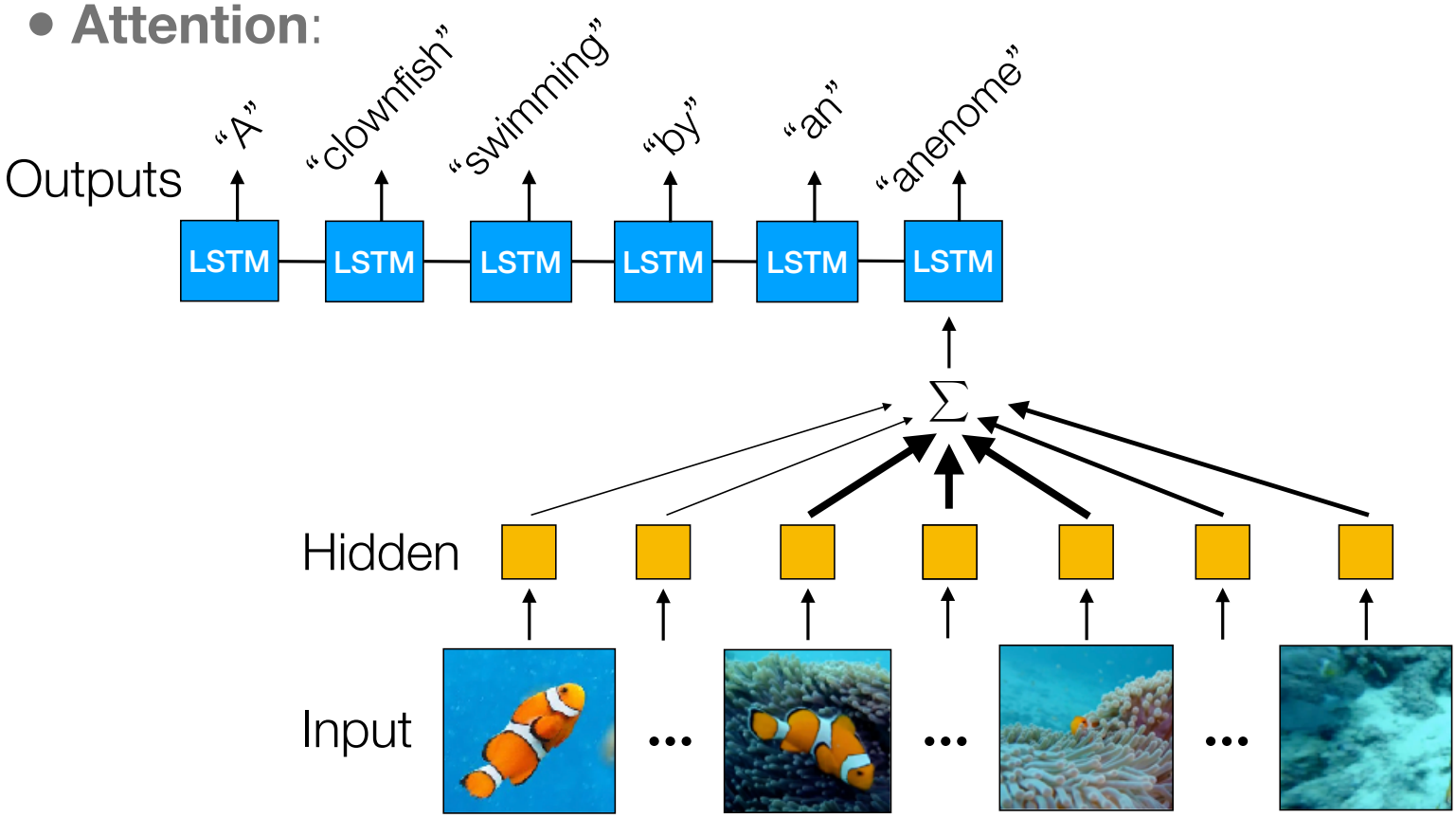
- **Attention**:



slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

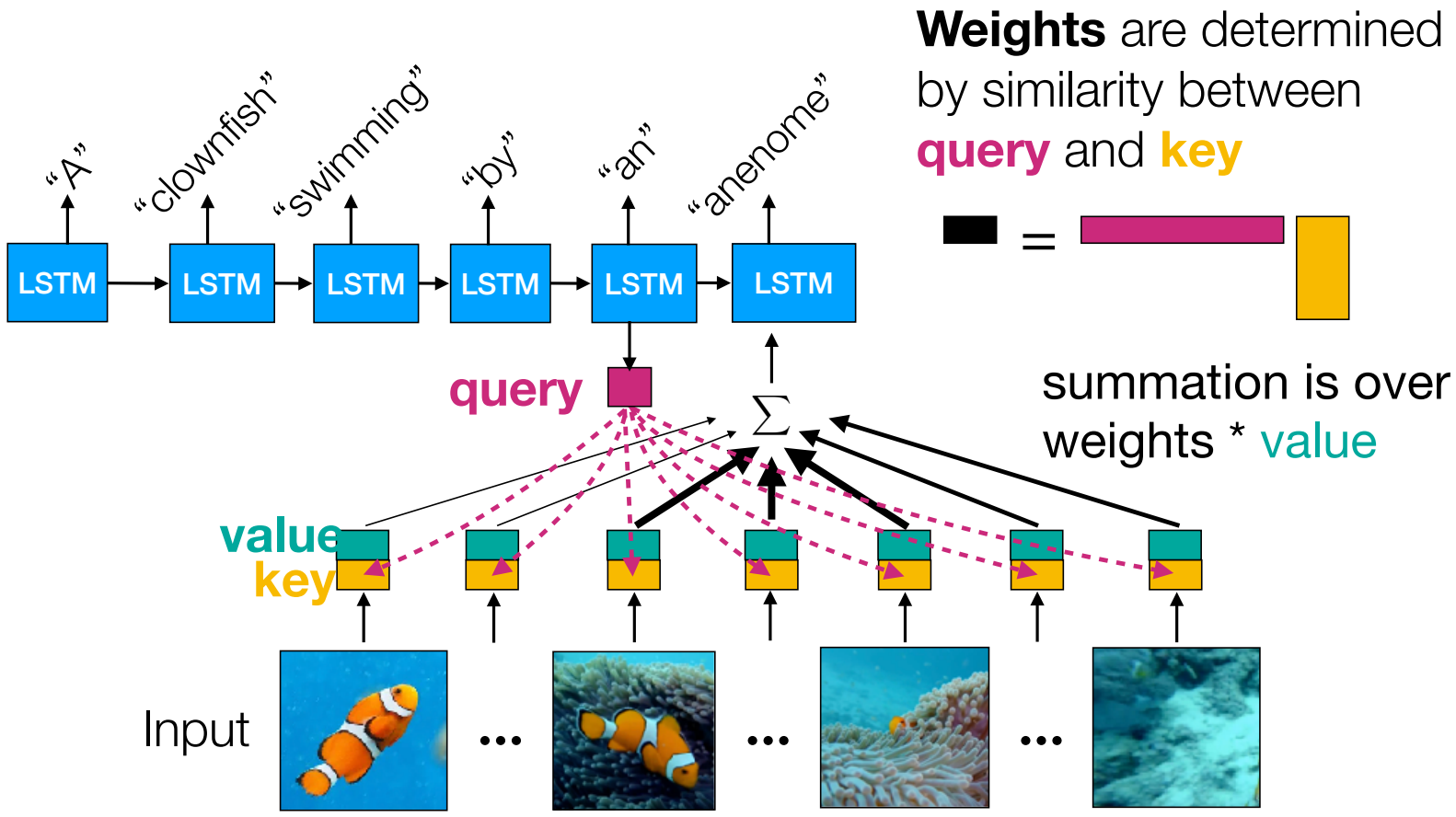
Transformers vs RNN

- **Attention**:



slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

Transformers vs RNN

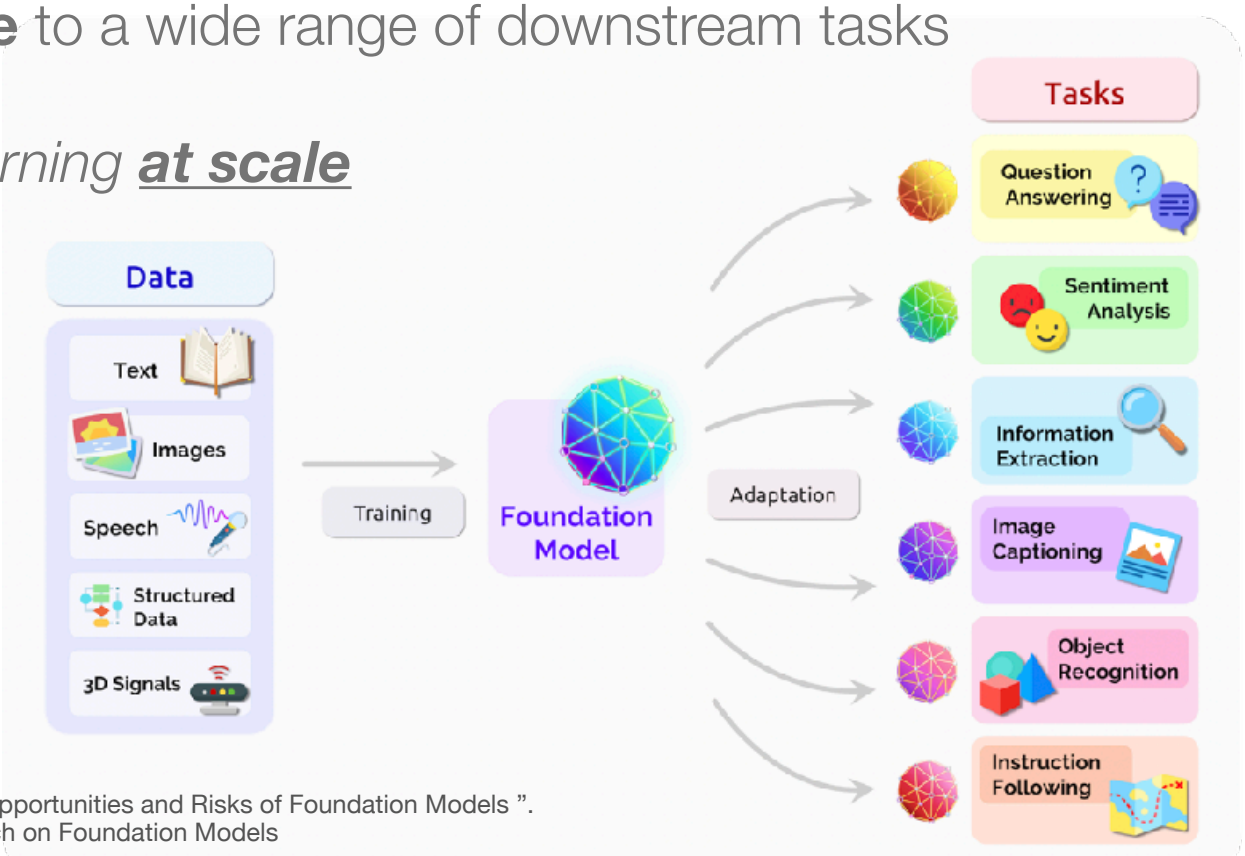


slides from: B. Freeman. P. Isola. *Advances in Computer Vision. MIT. 2021*

Foundation models

- Trained on broad **very (very) large scale** datasets and **adaptable** to a wide range of downstream tasks

Transfer learning ***at scale***

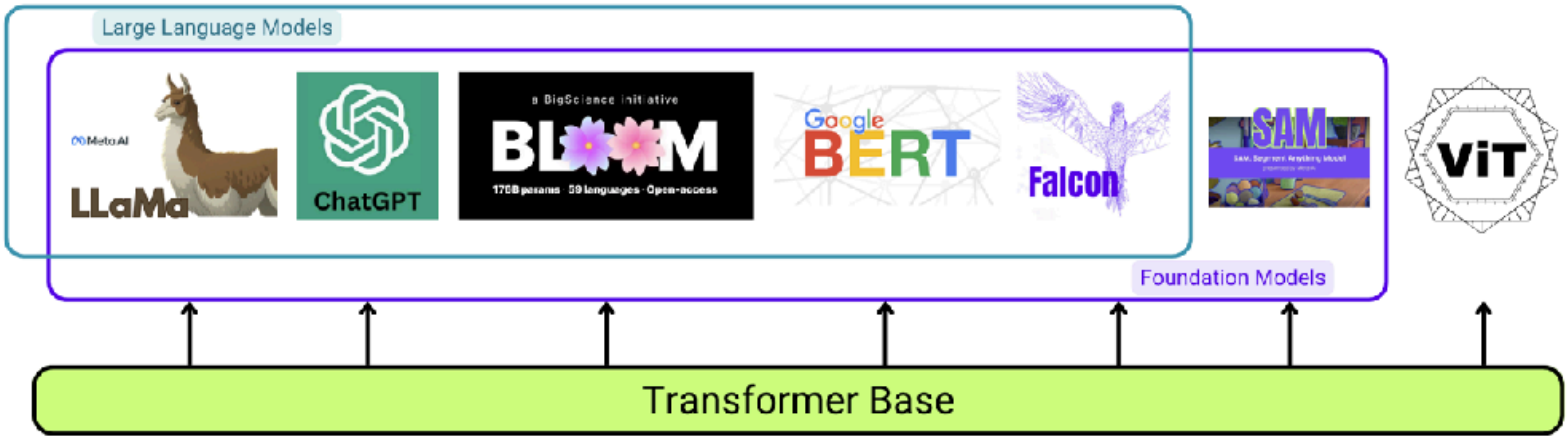


R. Bommasani et al. “On the Opportunities and Risks of Foundation Models ”. arXiv. 2022. Center for Research on Foundation Models

Foundation models

- Trained on broad **very (very) large scale** datasets and **adaptable** to a wide range of downstream tasks

Transfer learning ***at scale***



https://www.comet.com/site/blog/explainable-ai-for-transformers/

Foundation models

- **Pros:**
  - **plenty application domains & Impressive results**
- **Cons:**
  - **Very (VERY) resource demanding** (GPT-3 has 175 billion params)
  - **New legal, ethic challenges** (all AI apps inherit bias from a few generic models)

Foundation models

- Key ingredients of existing examples (not “new”):
  - **Self-supervised learning**
- and new ones that exist today:
  - Improved hardware
  - **Transformer-based architectures**
  - Availability of data

Foundation models

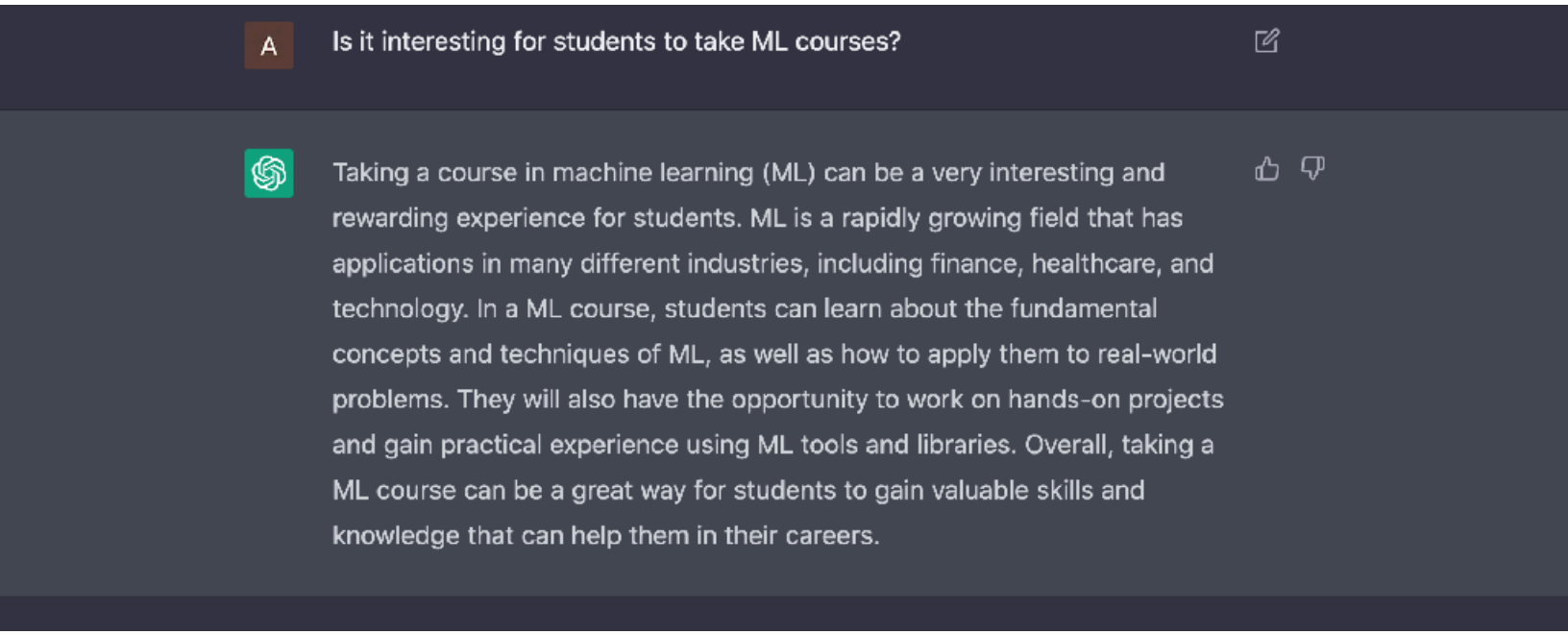
- **BERT** (Bidirectional Encoder Representations from Transformers)
  - NLP, Google 2018 (currently applied in English-language queries at Google)
  - **transformer** language model with a variable number of encoder layers and self-attention heads
  - trained from unlabeled data from the **BooksCorpus**[4] - 800M words and English **Wikipedia** - 2,500M words.
  - pretrained on two tasks: ***language modeling*** (some tokens masked; trained to predict them from context) and ***next sentence prediction*** (predict if certain sentence was probable or not given the first).

Foundation models

- **GPT-3/3.5/4**: *Generative Pre-trained Transformer*
  - NLP, openAI. GPT3 2020; GPT4 March 2023. Generate human-like text
  - **GPT3**: **transformer** network with 2048 token long context and 175 billion params (800 Gb of storage). Largest at the time.
    - trained on ***common crawl*** *(60%), webtext, books, wikipedia (3%)* to *predict what the next token* is
  - **GPT4**: pre-training using both public data and "data licensed" from other parties. Then, fine-tuned with reinforcement learning feedback from humans and AI for human alignment and policy compliance. (tech details not public
    - Also takes image inputs

Foundation models

- **GPT-3.5** application - *GPTChat 2022*



https://chat.openai.com/chat

Foundation models

NLP + generative image model. **Images from "*prompts*"**

- **DALL-E** (openAI 2021/22. Based on GPT-3)
- **Stable diffusion** (Stability AI & CompVis LMU 2022. Public code and model. **Diffusion model**: VAE + U-Net + text encoder.)
- **Midjourney** (Midjourney team, 2022)

Foundation models

- Dall-e GPT3.5 examples

“photorealistic happy polar bear with Christmas hat playing a Christmas music instrument in a card with the University of Zaragoza logo on the top right corner”

“Photorealistic image with "El Pilar" cathedral in Viena next to Belvedere on a snowy day with no people in front but pigeons”

Foundation models

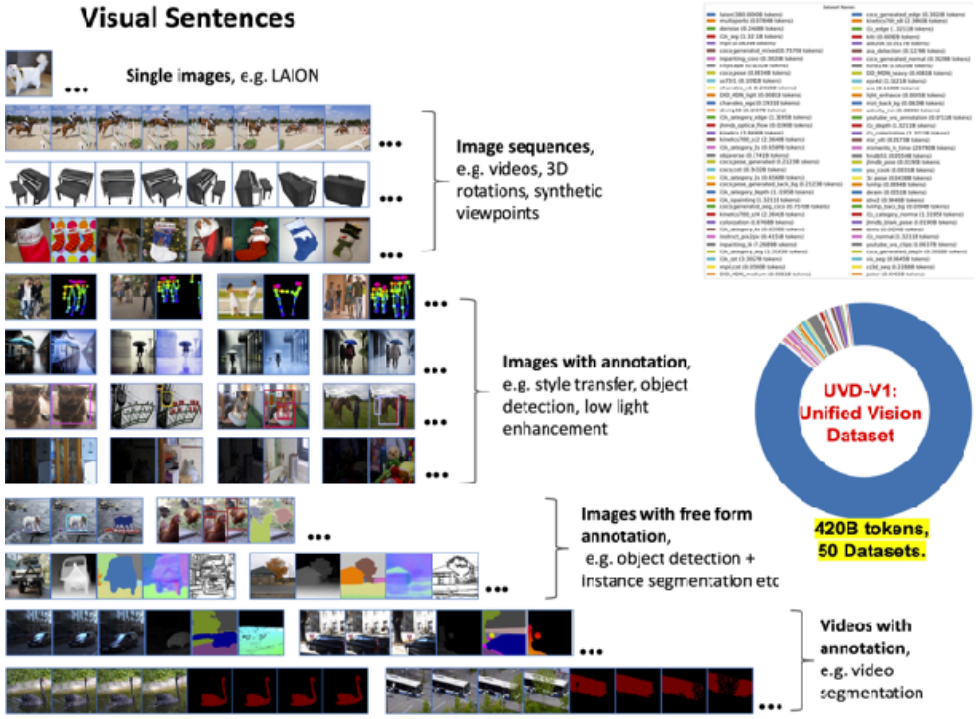
- Dall-e GPT3.5 examples, and GPT4 …

"photorealistic happy polar bear with Christmas hat playing a Christmas music instrument in a card with the University of Zaragoza logo on the top right corner"

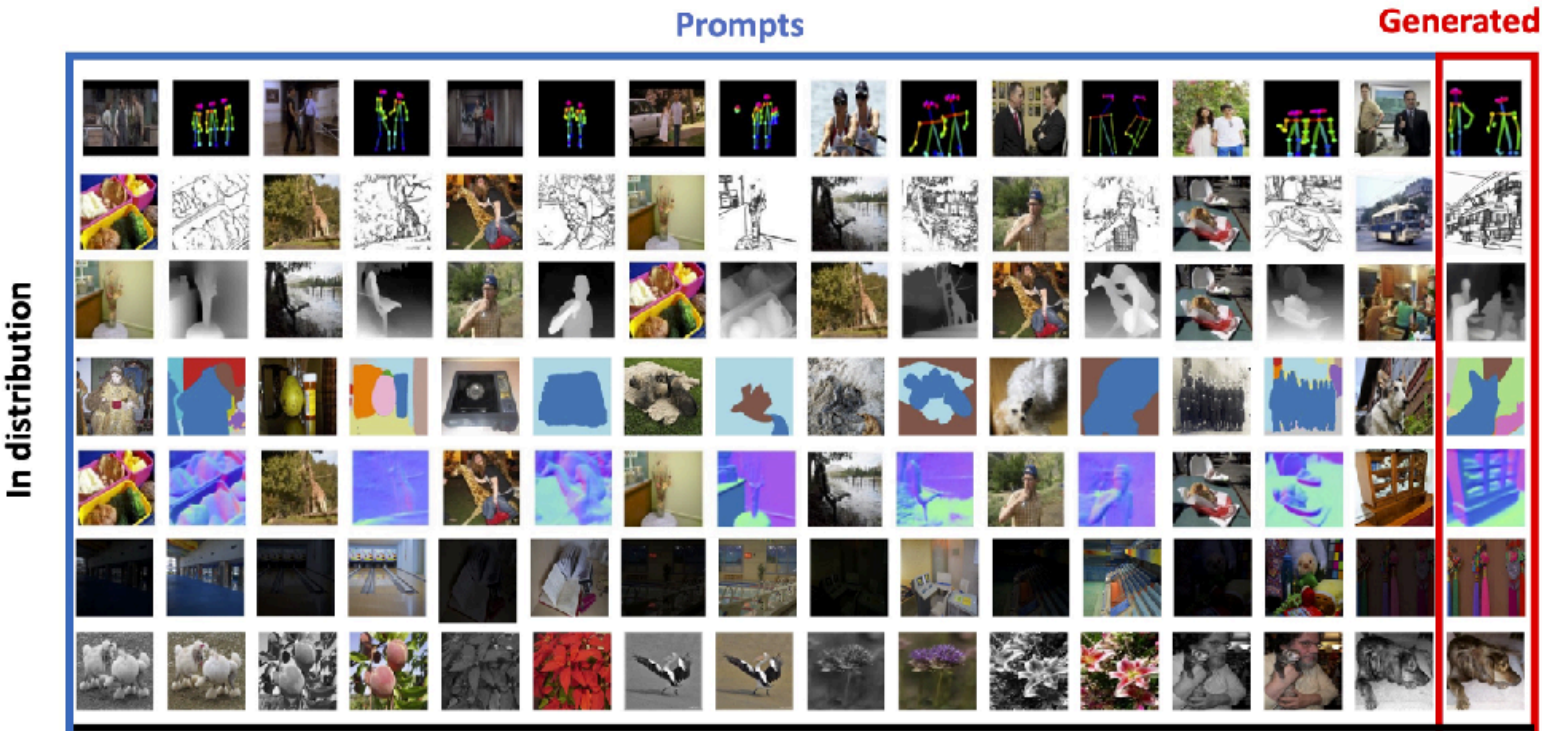
"Photorealistic image wi
Viena next to Belveder
no people in fr

Foundation models: Large Vision Models

- *Sequential Modeling Enables Scalable Learning for Large Vision Models*



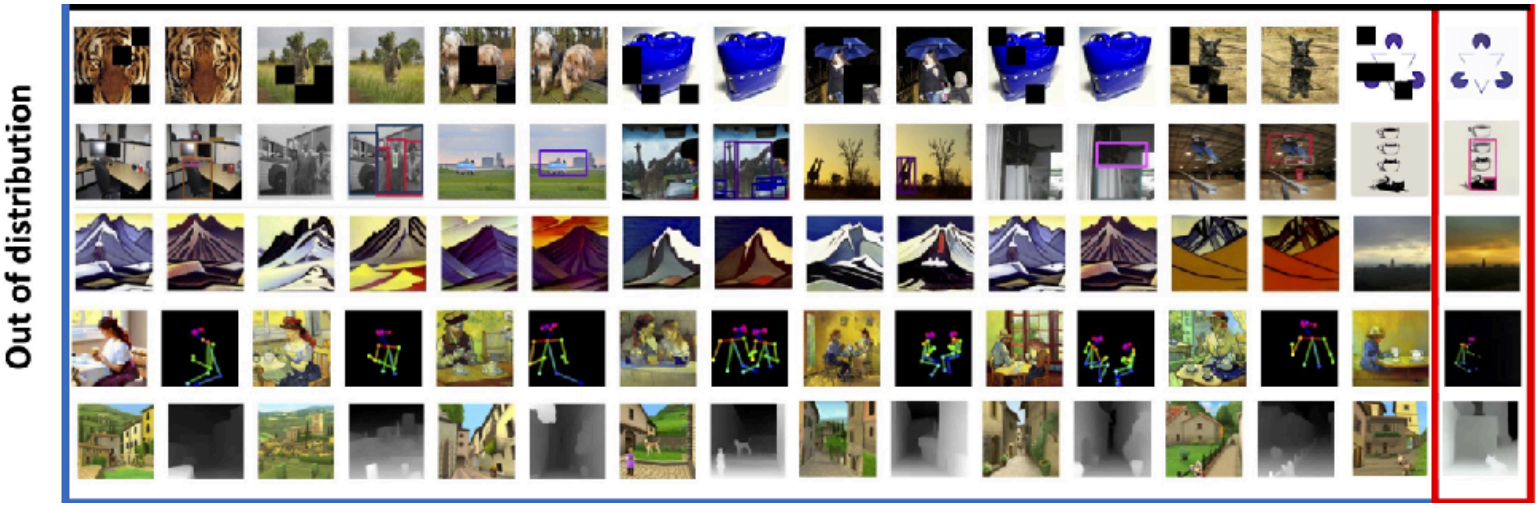
Bai, Yutong, et al. "Sequential Modeling Enables Scalable Learning for Large Vision Models." *arXiv preprint arXiv:2312.00785* (2023).



- Every row prompt: sequence of images interleaved with annotations, followed by a query.

- Last image: predicted by the model

Foundation models



- Every row prompt: sequence of images interleaved with annotations, followed by a query.

- Last image: predicted by the model

Next

- LAB 5
- Efficiency & DL
- Wrap-up

Bibliography - Resources for some of the materials today

- Stanford classes on deep learning for Computer Vision (http://cs231n.stanford.edu) and Deep Learning (https://cs230.stanford.edu/)
- Berkeley course - CS 198-126: Deep Learning for Visual Data (https://ml-berkeley.notion.site/CS-198-126-Deep-Learning-for-Visual-Data-c77ce2526cb4460b8b02dc56a86bc426)
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org
- *Computer Vision: Algorithms and Applications*. 2nd Edition. Richard Szeliski. https://szeliski.org/Book/
- *Probabilistic Machine Learning: An introduction*. K. Murphy. MIT Press, 2022, probml.ai