Attention-Based Models

Sadeep Jayasumana

The "Transformer" Paper

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

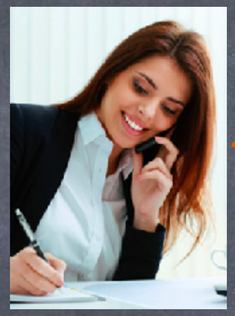
Jakob Uszkoreit* Google Research usz@google.com

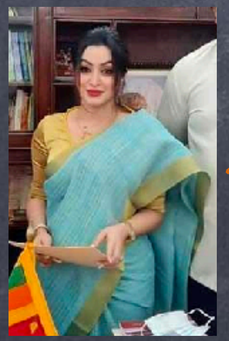
Llion Jones* Google Research llion@google.com

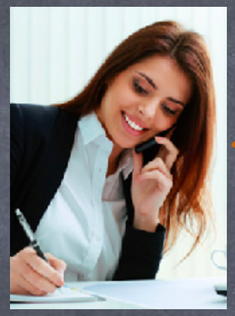
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

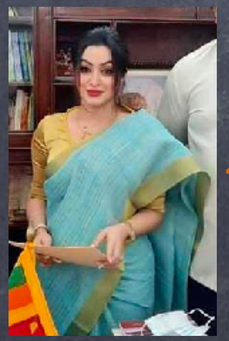
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

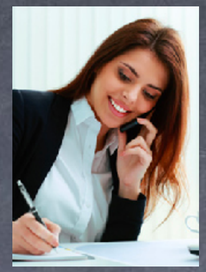
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

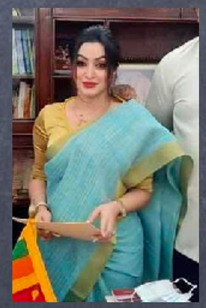


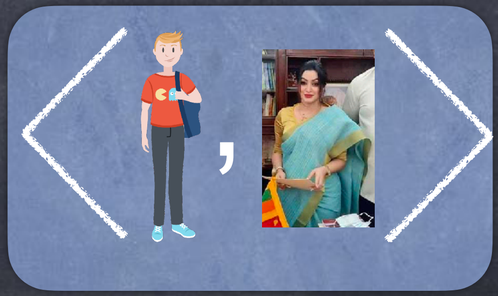


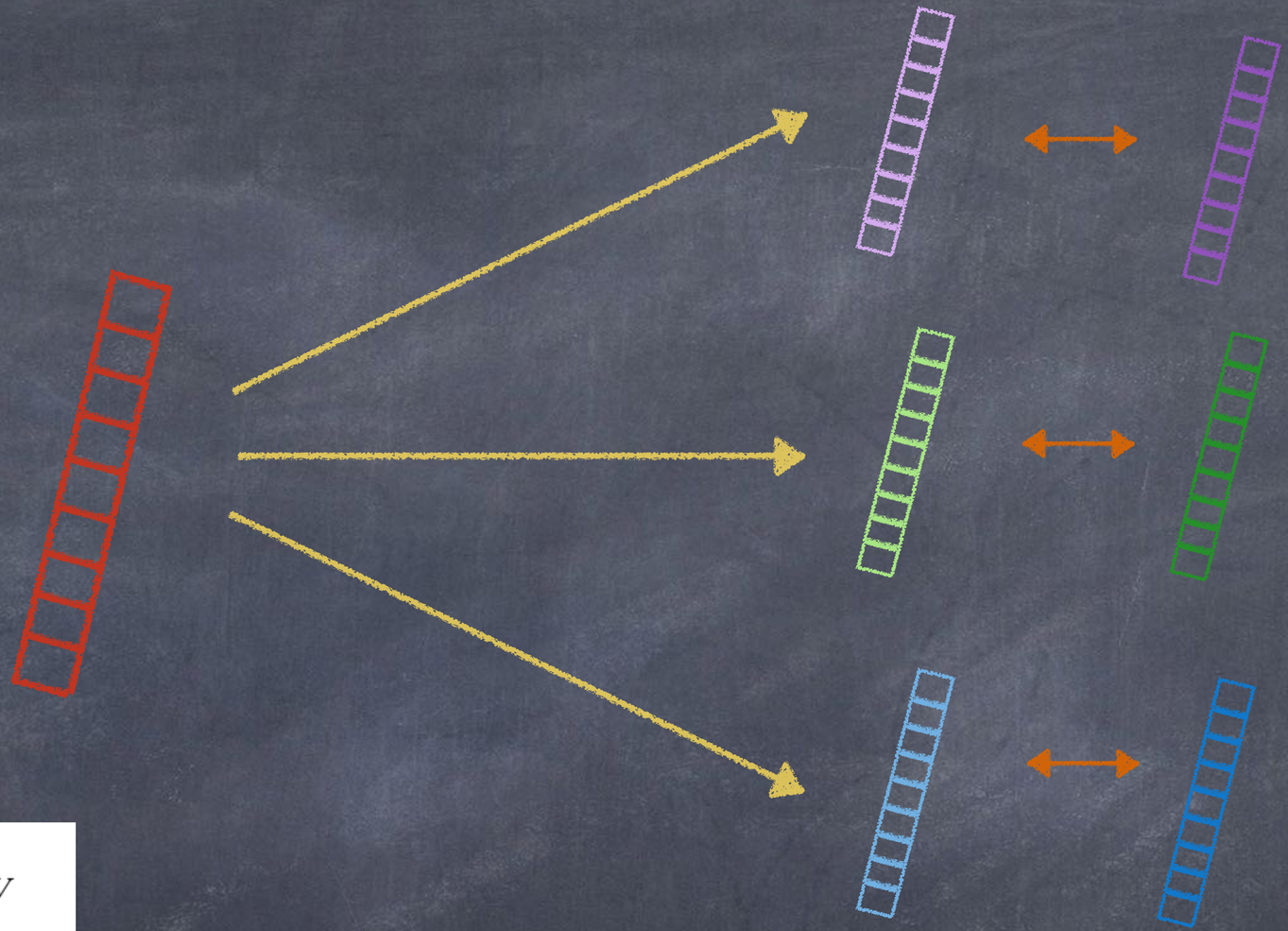




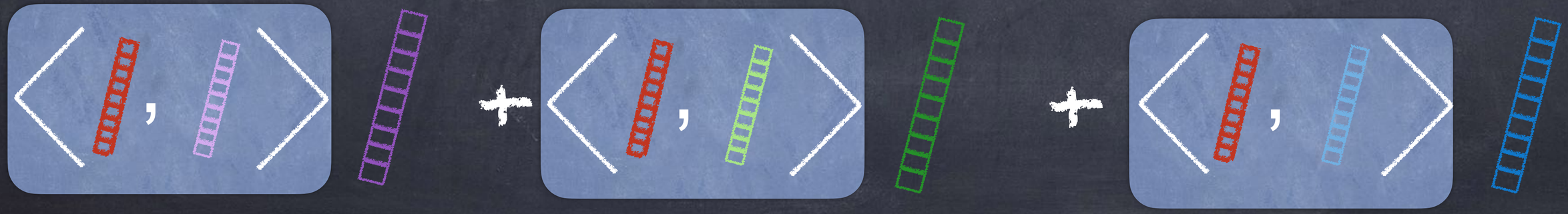








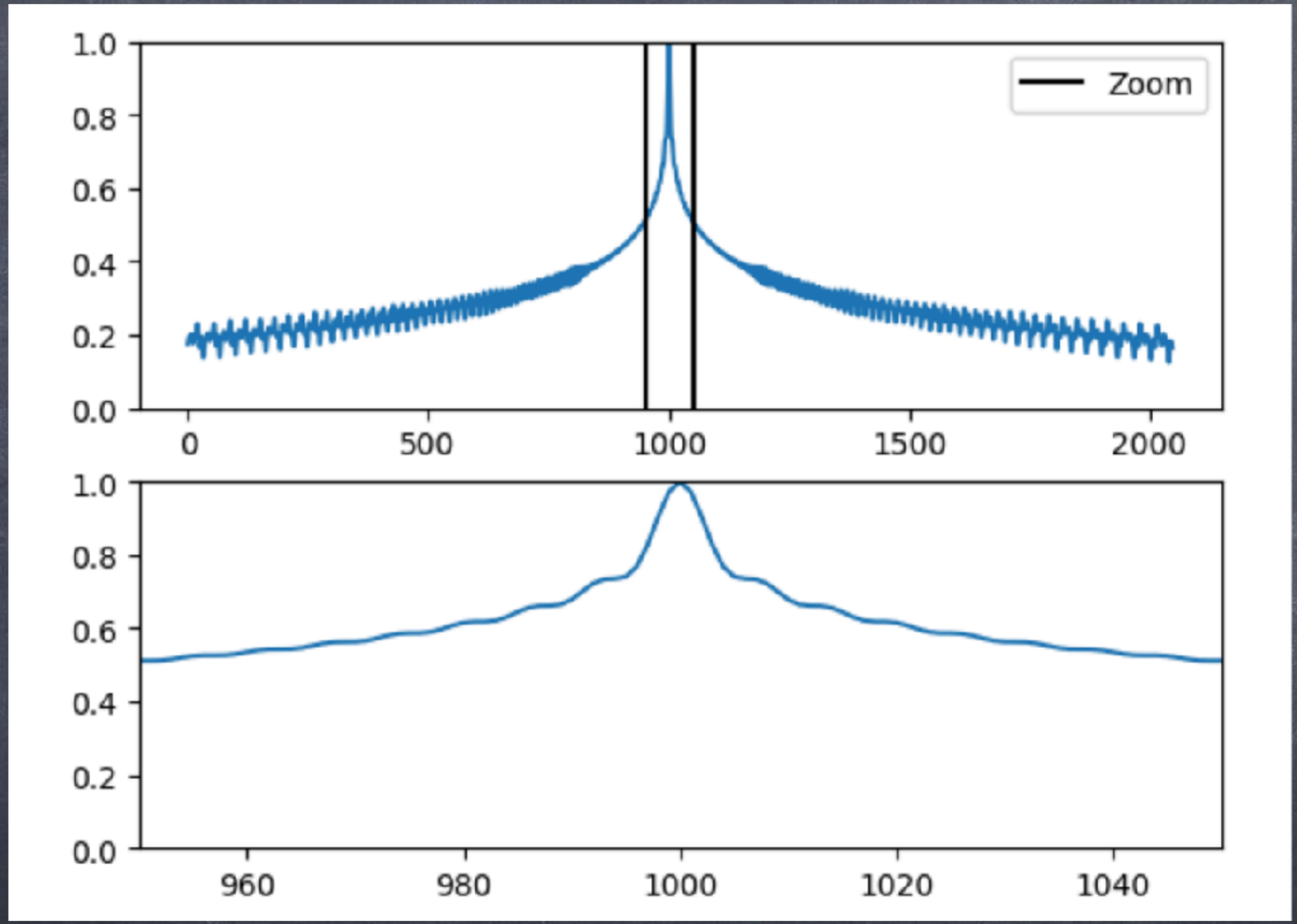
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Input Pipeline

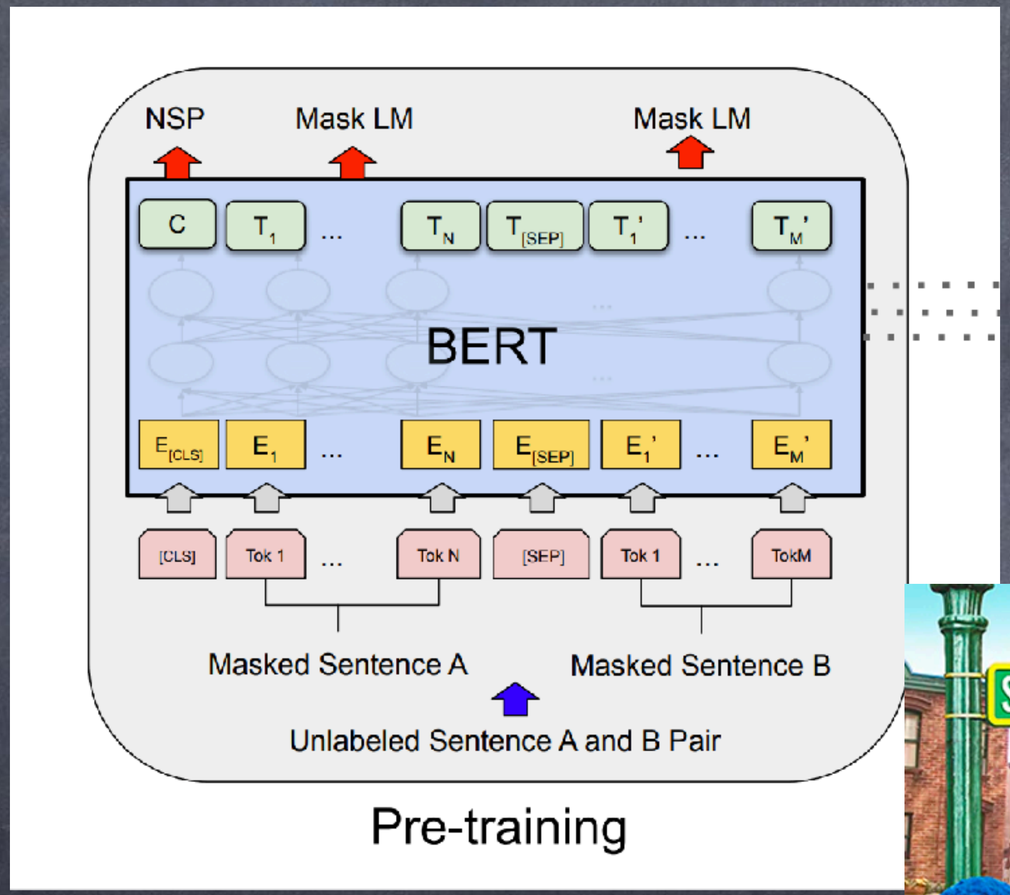
- Tokenization (SentencePiece, WordPiece)
  - Breaks a string into a sequence of tokens
- Lookup
  - Assigns an integer ID to each token using a vocab
- (Input) embedding
  - Assigns a (learnable) vector to each integer ID in the vocab
- Positional encoding
  - Encodes knowledge about the position in the sequence

Positional Encoding



Transformer Encoder Models

MLM Pretraining - BERT



Transformer Decoder Models

Language Generation Example

I went home and ...

Self-attention: Attending to the same sequence.
Masking: To prevent undesired attentions.

Transformer Encoder-Decoder Models

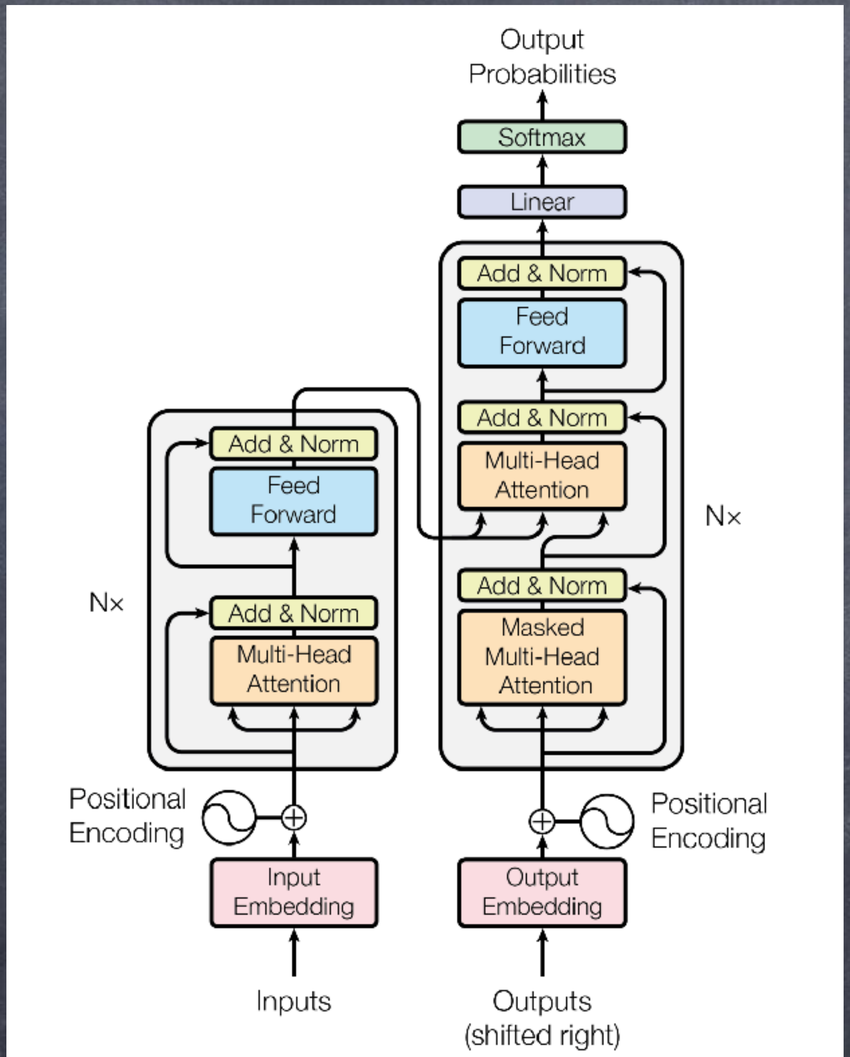


Figure 1: The Transformer - model architecture.

Language Translation Example

I went home

मैं ? ?

Keys

Values

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

Real-world Solutions with Transformers

Real-world Solutions with Transformers

Information Retrieval

Exercise: You are working on a search engine. You have positive (query, document) pairs and negative (query, document) pairs. Design a neural network to retrieve relevant documents given a query. You can assume the existence of an efficient nearest neighbor search algorithm.

Karpukhin et. al., Dense Passage Retrieval for Open-Domain Question Answering

Recommendation Systems

Exercise: You are working on a website that hosts a fixed set of videos. Given the previous watch history of a user, you want to suggest new videos to them. Design a neural network for this purpose. You can assume that the number of videos on your website is O(10,000).

What happens if you want to support new videos that are continuously added by users of the website?

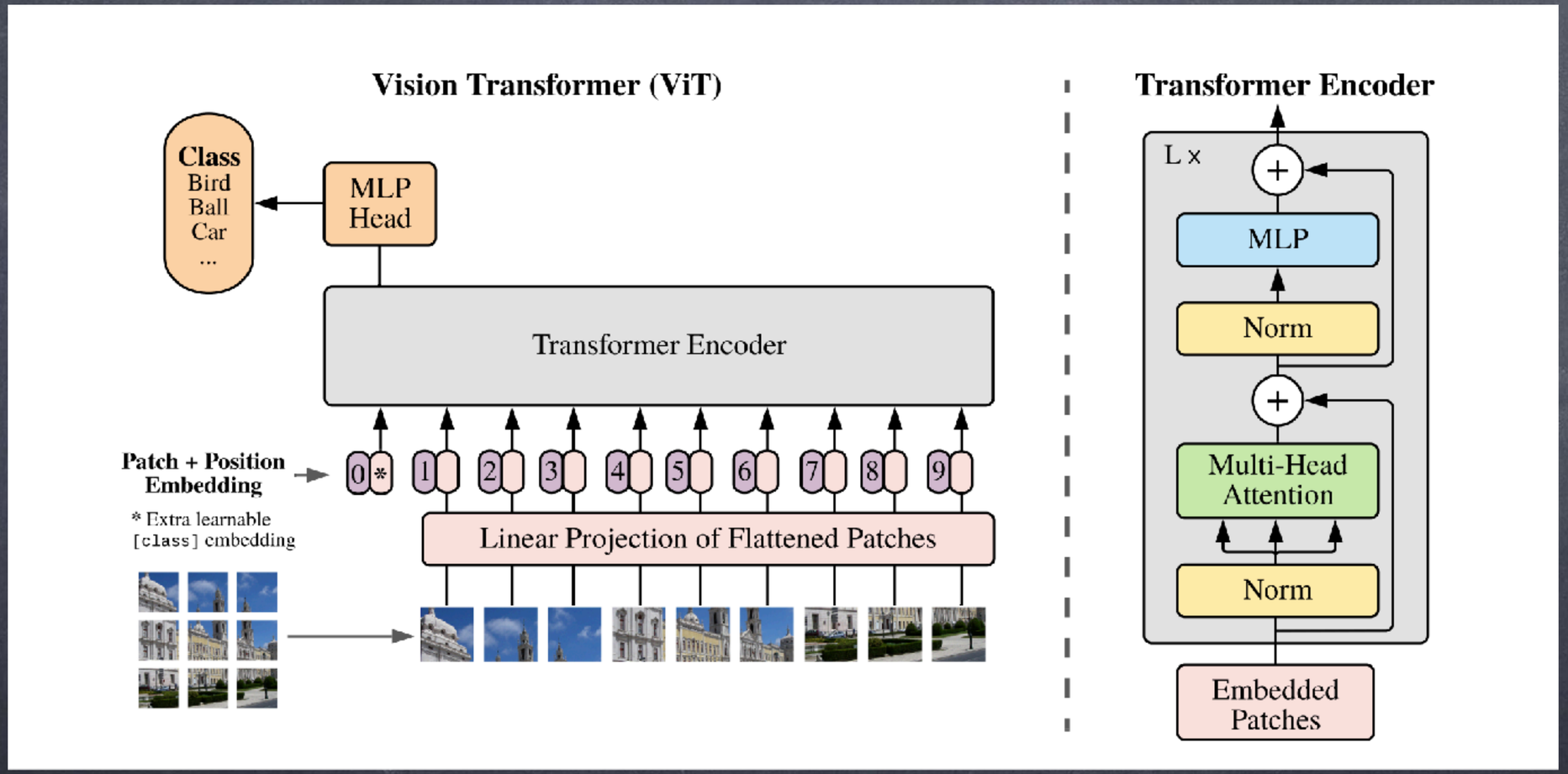
Recommendation Systems

Exercise: You are working on a website that hosts a fixed set of videos. Given the previous watch history of a user, you want to suggest new videos to them. Design a neural network for this purpose. You can assume that the number of videos on your website is O(10,000).

What happens if you want to support new videos that are continuously added by users of the website?

Zang et. al., Next Item Recommendation with Self-Attention.

Vision Transformer (ViT)



Vision Transformer

- Inductive-bias vs large-scale training

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

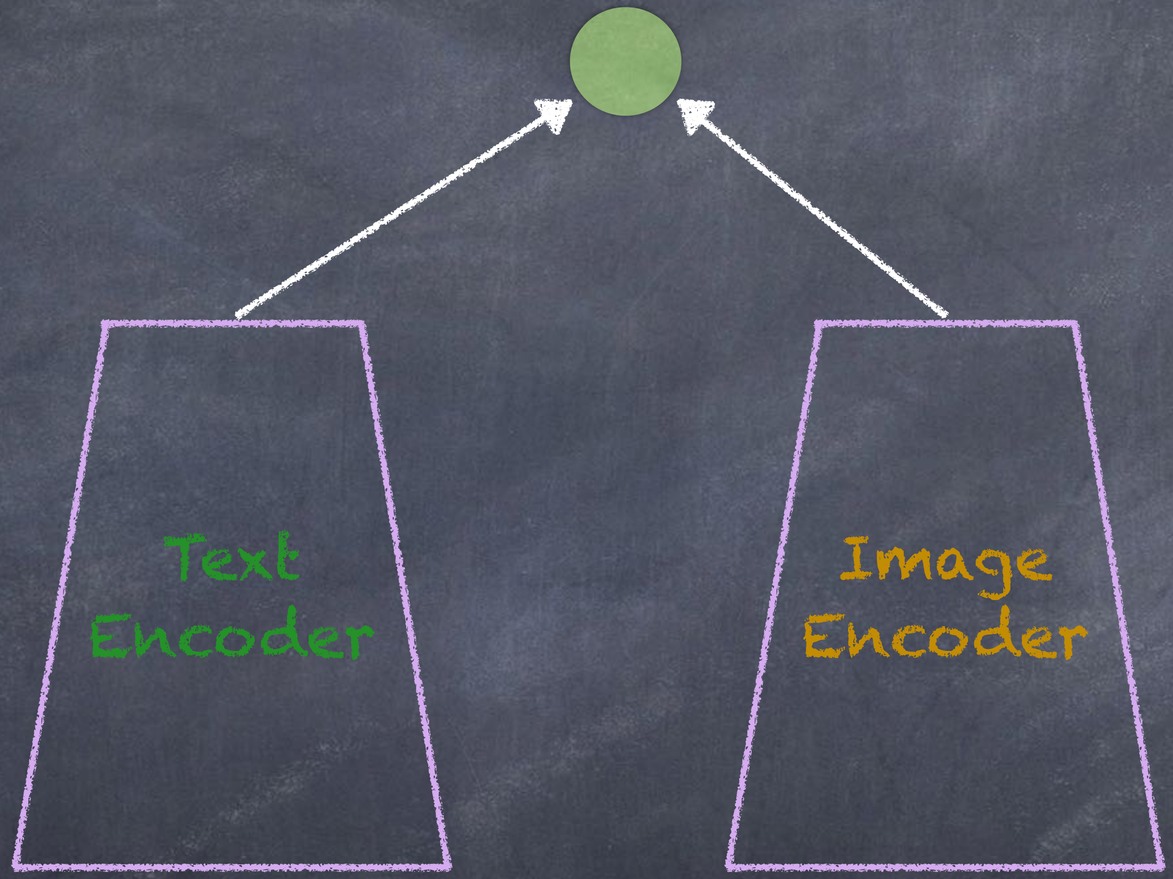
Image-Text Models (CLIP etc.)

A herd of elephants at the edge of a river and in the bushy land nearby.

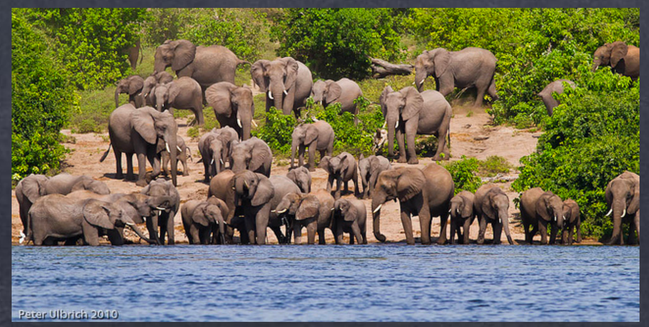
Image-Text Models (CLIP etc.)

A herd of donkeys in their natural habitat.

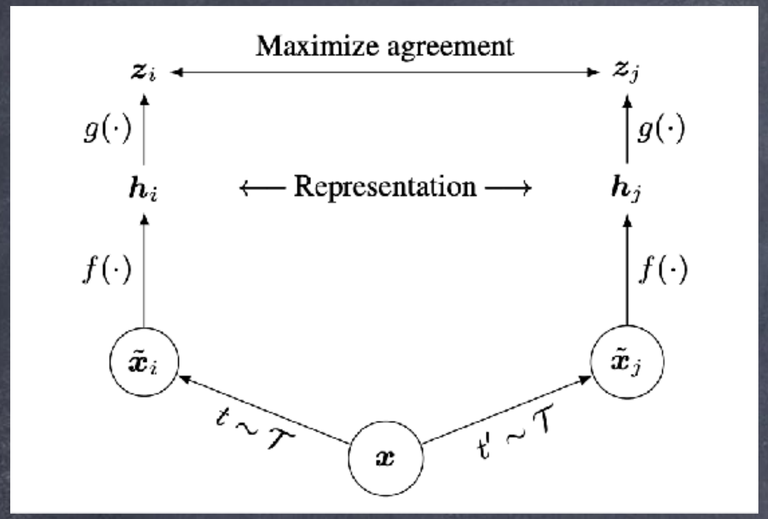
Image-Text Models (CLIP etc.)



A herd of elephants at the edge of a river and in the bushy land nearby.



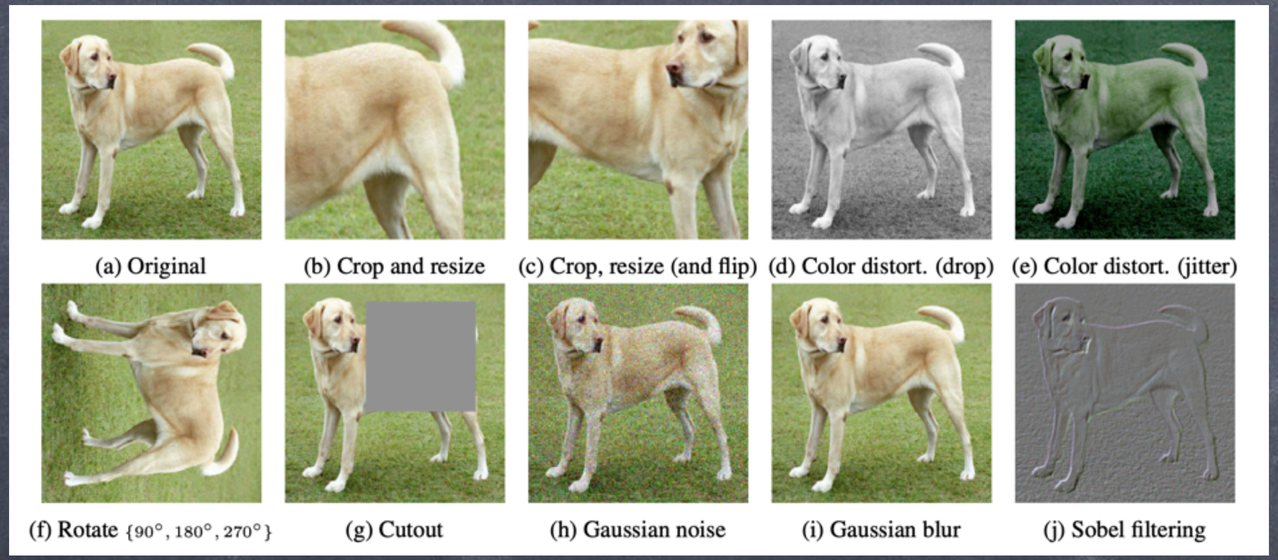
Contrastive Learning (SimCLR)



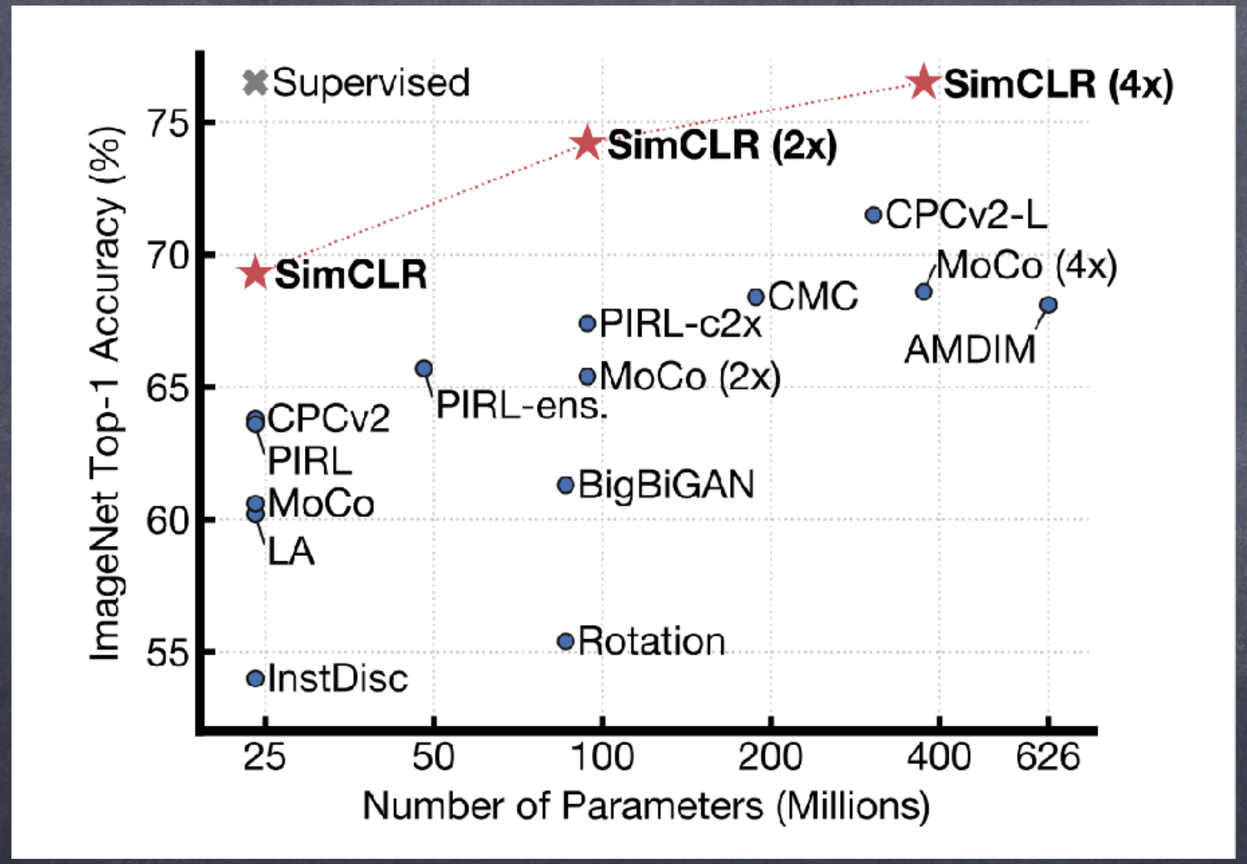
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)},$$

Paper: A Simple Framework for Contrastive Learning of Visual Representations

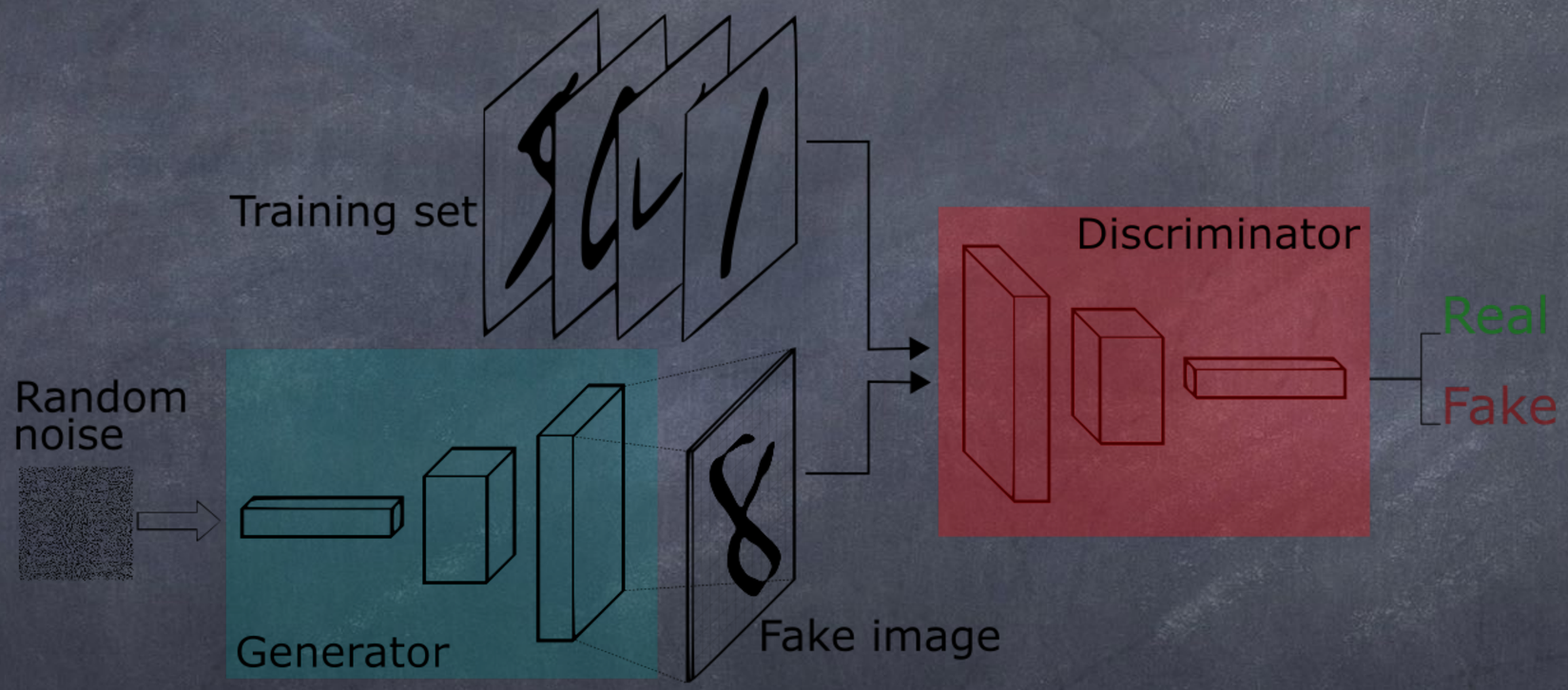
Data Augmentation in SimCLR



SimCLR Results



Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Thank you!