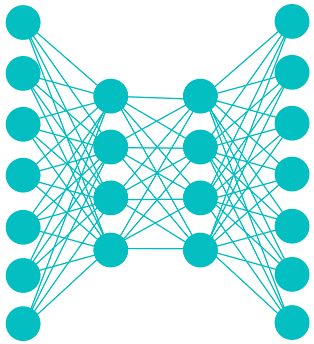
Lecture Notes for

Neural Networks and Machine Learning

Practical Transformers

Vision Transformers



Logistics and Agenda

- Logistics
  - Grading update
- Agenda
  - Paper Presentation
  - Decoder Transformers
  - Vision Transformers
  - Town Hall

Paper Presentation

FaceNet: A Unified Embedding for Face Recognition and Clustering

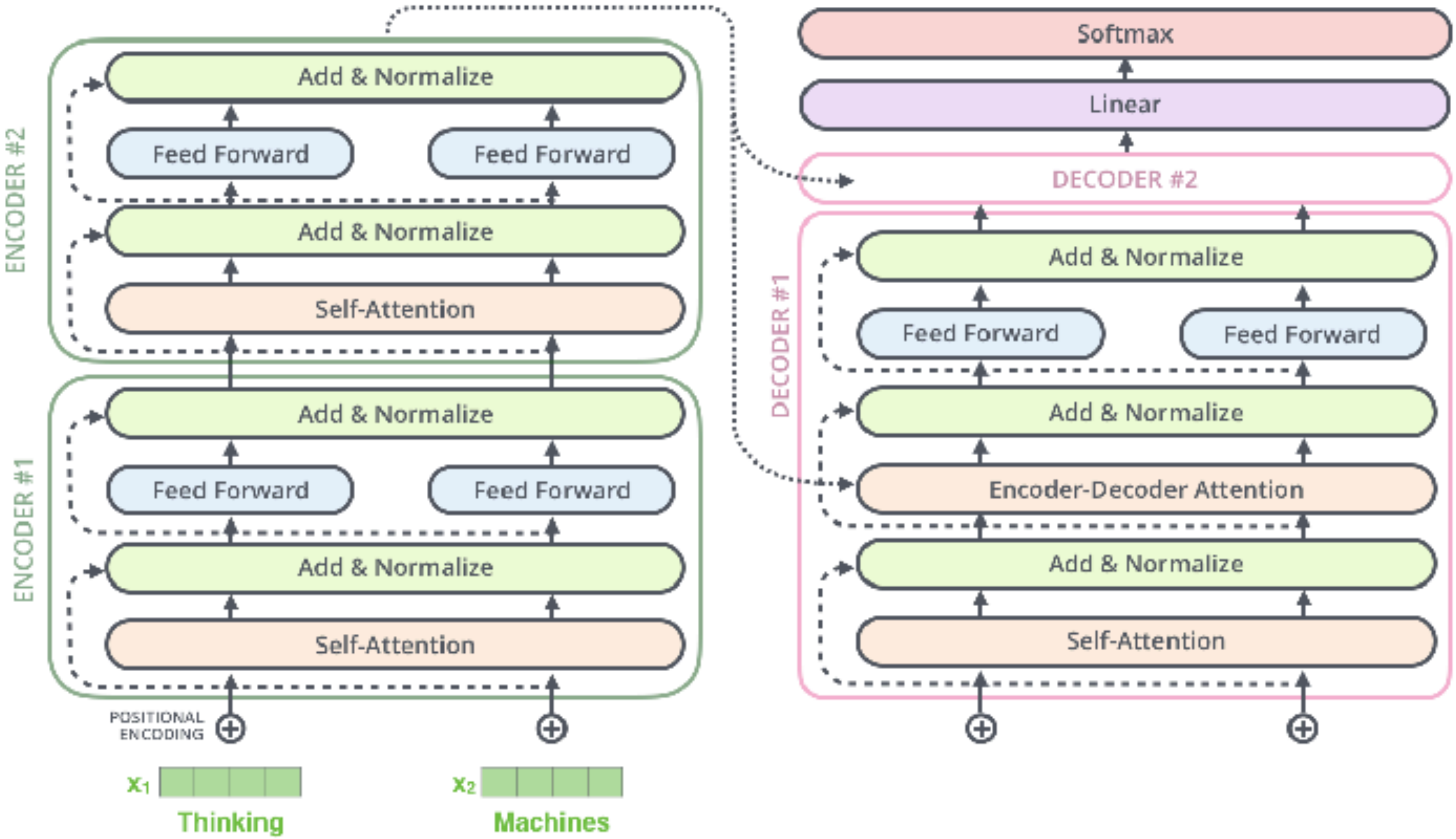
Florian Schroff
fschroff@google.com
Google Inc.

Dmitry Kalenichenko
dkalenichenko@google.com
Google Inc.

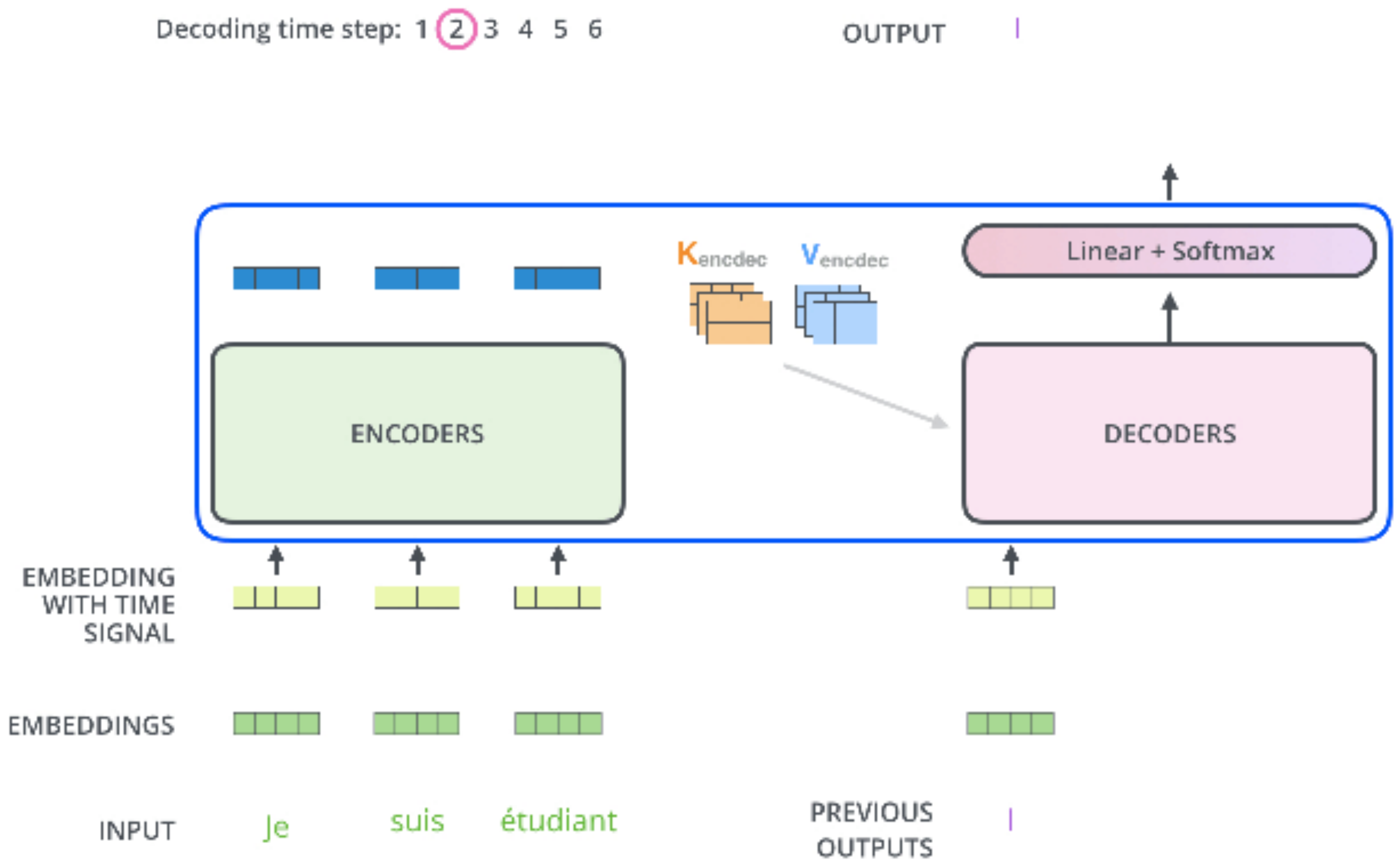
James Philbin
jphilbin@google.com
Google Inc.

Encoder+Decoder Xformer

Transformer: Encoders and Decoders



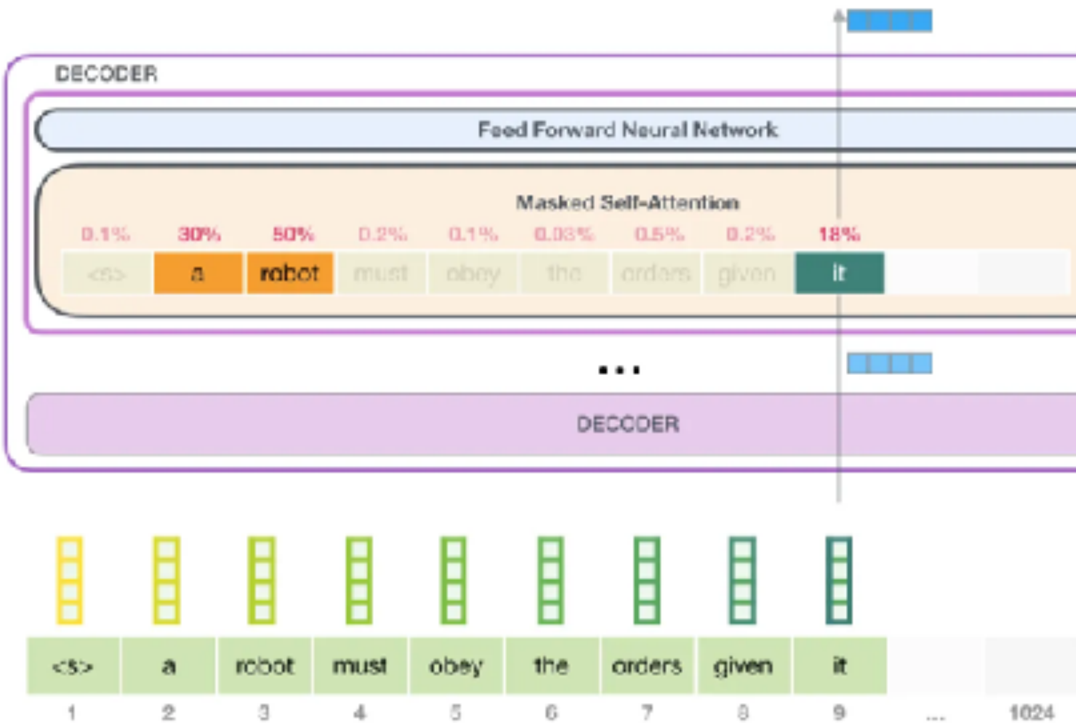
Transformer: Putting it all together



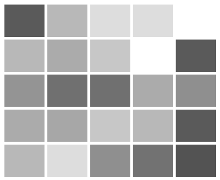
Auto-regressive Transformer

- Decoder only, text encoding happens in attention
- Generative pre-training

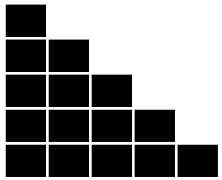
$$\mathcal{L}_1(\mathcal{U}) = \sum_{i\in S} \log P(\underbrace{u_i}_{\text{curr}} \mid \underbrace{u_{i-k},\ldots,u_{i-1}}_{\text{other words}} ; \underbrace{\mathbf{W}}_{\text{params}})$$



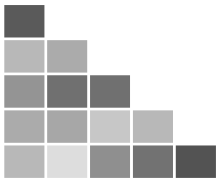
Predict the next word from unlabeled dataset



Attention Map



Mask



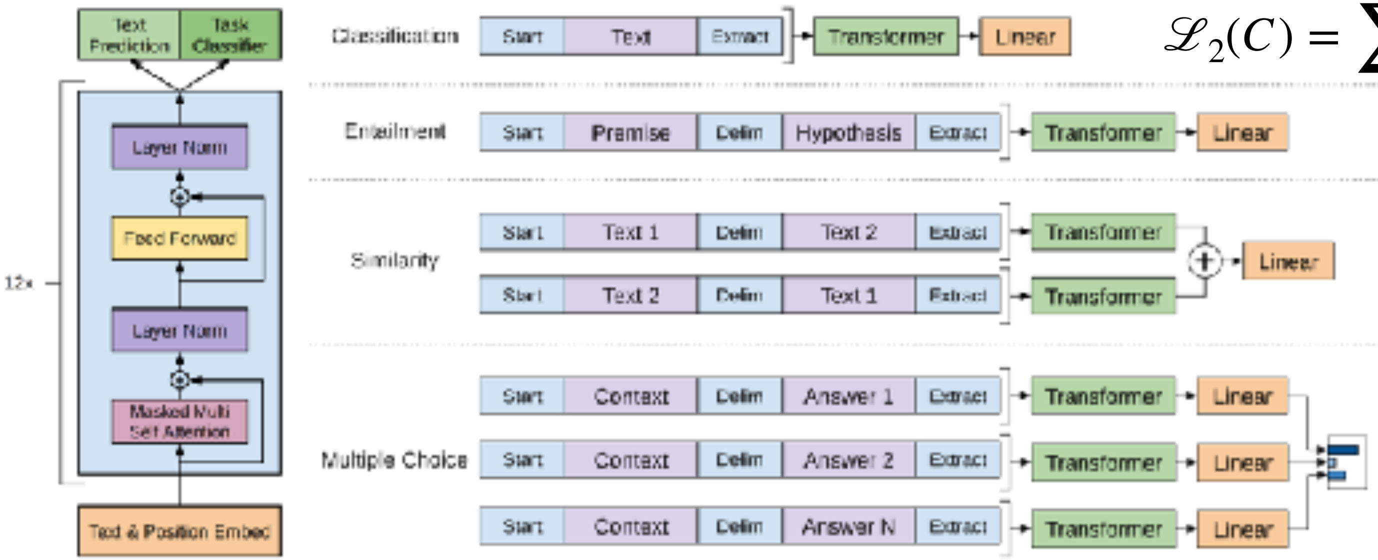
Masked Attention Map

Radford, et al. Improving Language Understanding by Generative Pre-Training, ArXiV 20

Fine Tuning

- Supervised tasks after pre-training, make transformer better through various tasks, trained simultaneously

h_l^m



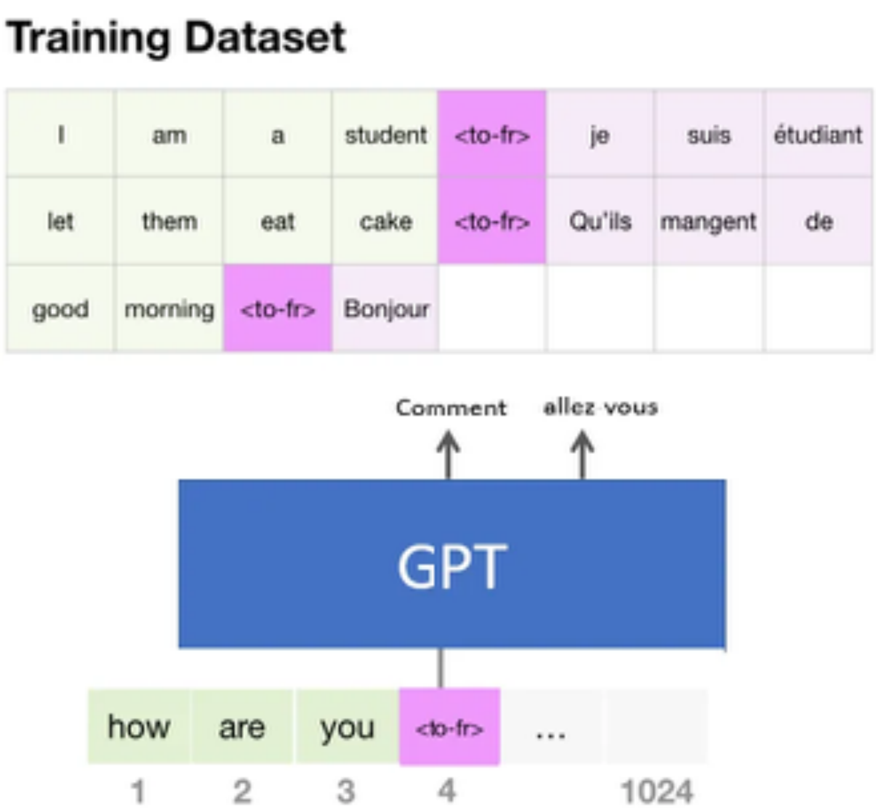
$$\mathscr{L}_2(C) = \sum CE \left[\underbrace{y_c}_{\text{label}} ; \underbrace{\sigma(h_l^m \cdot \mathbf{W}_c)}_{\text{predict}} \right]$$

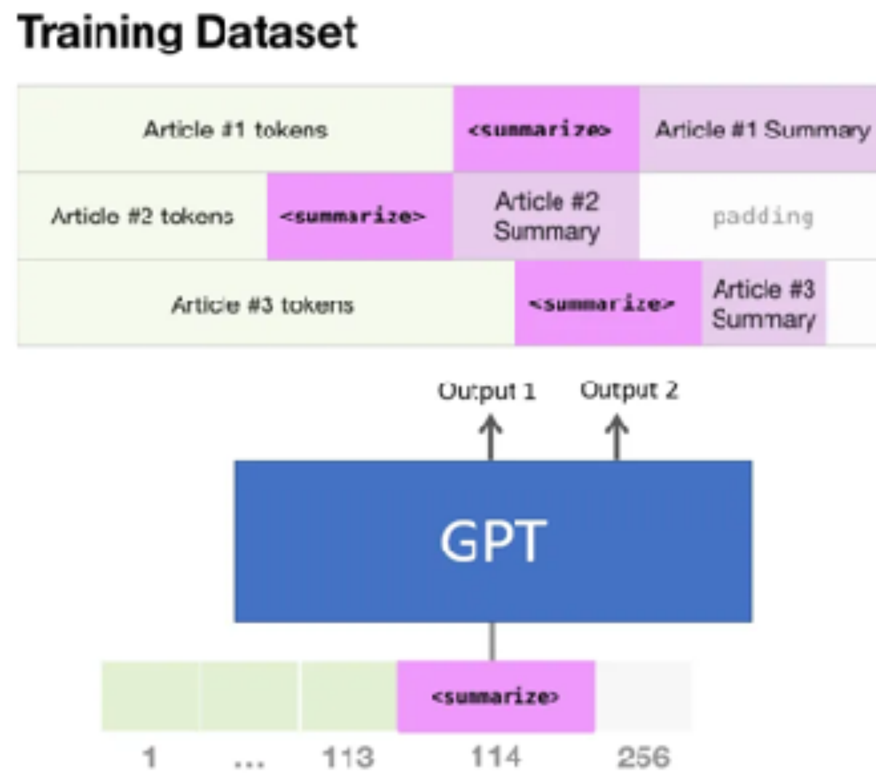
For each task, C

$$\mathscr{L}_{Total} = \sum_{C \in \mathscr{C}} \mathscr{L}_2(C) + \lambda \cdot \mathscr{L}_1(C)$$

Radford, et al. Improving Language Understanding by Generative Pre-Training, ArXiV 20

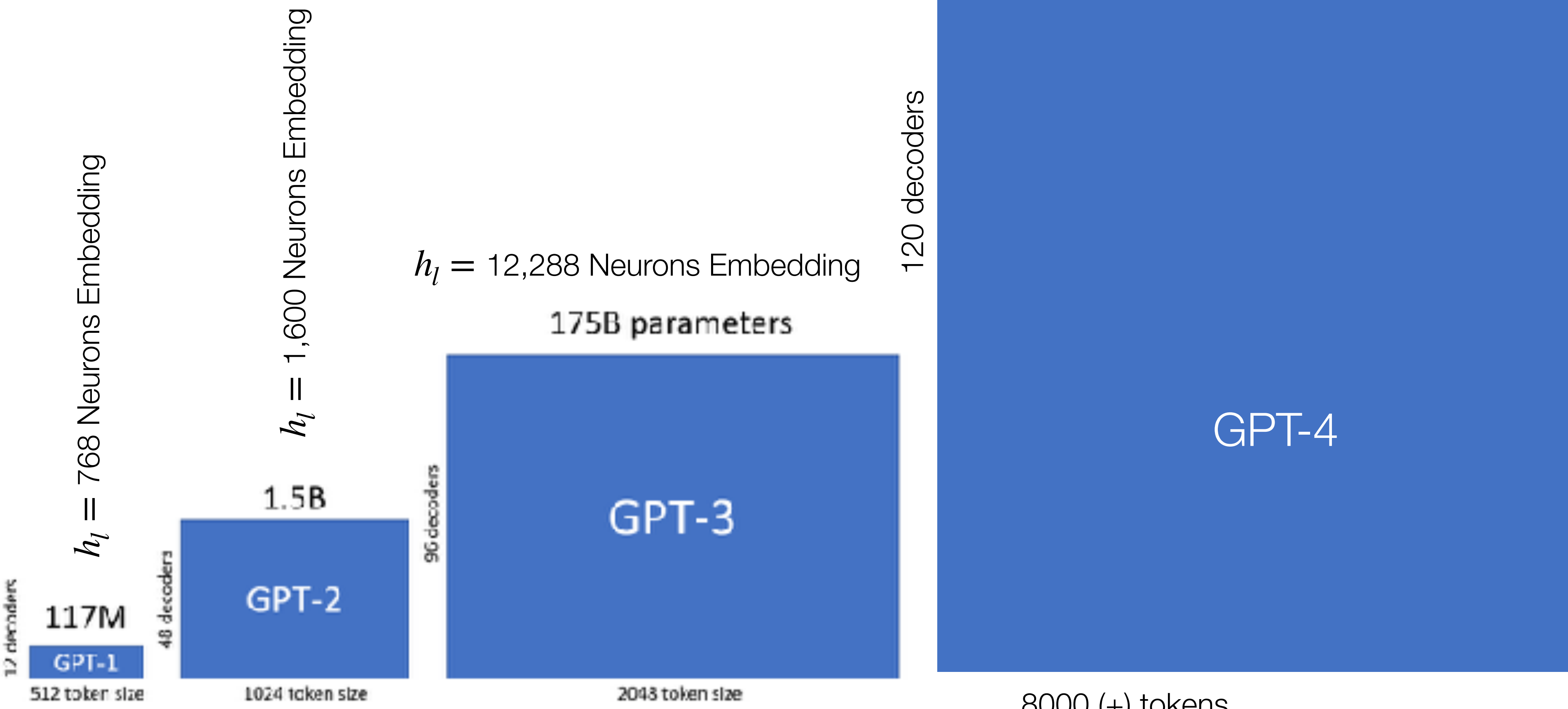
How to label a decoder model?





Fine Tune with "Action" Tokens and training examples.

Size of GPT



8000 (+) tokens
1.7 Trillion parameters
$h_l = $ 20,000 Neurons embedding

Input Images

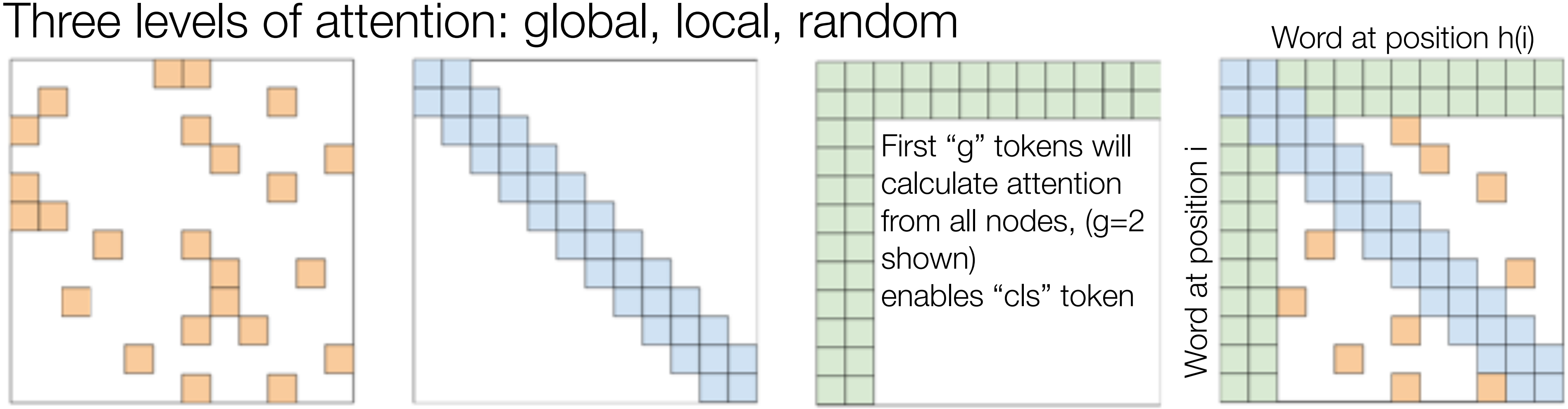
https://medium.com/@YanAIx/step-by-step-into-gpt-70bc4a5d8714

A Variant on Attention, long sequences

- Many works look to make attention more efficient, **BigBird**

$$\mathrm{ATTN}_D(\boldsymbol{X})_i = \boldsymbol{x}_i + \sum_{h=1}^{H} \sigma\left(Q_h(\boldsymbol{x}_i) K_h(\boldsymbol{X}_{N(i)})^T\right) \cdot V_h(\boldsymbol{X}_{N(i)})$$

Three levels of attention: global, local, random



(a) Random attention

Choose a random subset of nodes to calculate attention

(b) Window attention

Choose neighborhood for locally dense attention

(c) Global Attention

(d) BIGBIRD

Zander et al., “Big Bird: Transformers for Longer Sequences” 2021

BigBird Results

https://arxiv.org/pdf/2007.14062.pdf

Question Answering F1 Score
Find portion of passage and evidence

Model	HotpotQA			NaturalQ		TriviaQA		WikiHop
	Ans	Sup	Joint	LA	SA	Full	Verified	MCQ
HGN [26]	**82.2**	88.5	**74.2**	-	-	-	-	-
GSAN	81.6	88.7	73.9	-	-	-	-	-
ReflectionNet [32]	-	-	-	77.1	**64.1**	-	-	-
RikiNet-v2 [61]	-	-	-	76.1	61.3	-	-	-
Fusion-in-Decoder [39]	-	-	-	-	-	84.4	90.3	-
SpanBERT [42]	-	-	-	-	-	79.1	86.6	-
MRC-GCN [87]	-	-	-	-	-	-	-	78.3
MultiHop [14]	-	-	-	-	-	-	-	76.5
Longformer [8]	81.2	88.3	73.2	-	-	77.3	85.3	81.9
BIGBIRD-ETC	81.2	**89.1**	73.6	**77.8**	57.9	**84.5**	**92.4**	**82.3**

Table 3: Fine-tuning results on **Test** set for QA tasks. The Test results (F1 for HotpotQA, Natural Questions, TriviaQA, and Accuracy for WikiHop) have been picked from their respective leaderboard. For each task the top-3 leaders were picked not including BIGBIRD-etc. **For Natural Questions Long Answer (LA), TriviaQA, and WikiHop, BIGBIRD-ETC is the new state-of-the-art**. On HotpotQA we are third in the leaderboard by F1 and second by Exact Match (EM).

DNA Sequence
Bits per character
Encoding

Model	BPC
SRILM [58]	1.57
BERT (sqln. 512)	1.23
BIGBIRD (sqln. 4096)	**1.12**

Table 5: MLM BPC

DNA Fragment
Classification
promoter region

Model	F1
CNNProm [90]	69.7
DeePromoter [71]	95.6
BIGBIRD	**99.9**

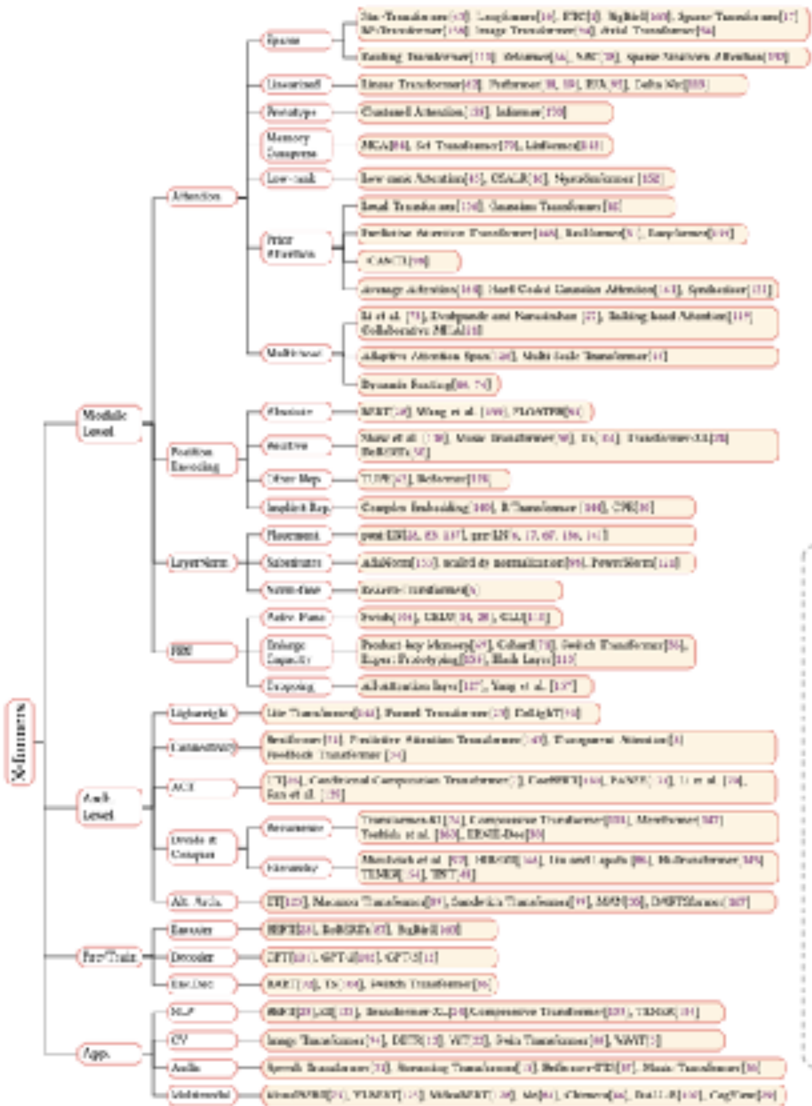
Table 6: Comparison.

Predict non-coding
region effects

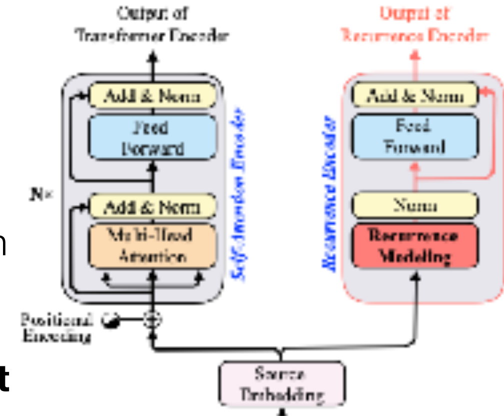
Model	TF	HM	DHS
gkm-SVM [30]	89.6	-	-
DeepSea [109]	95.8	85.6	**92.3**
BIGBIRD	**96.1**	**88.7**	92.1

Table 7: Chromatin-Profile Prediction

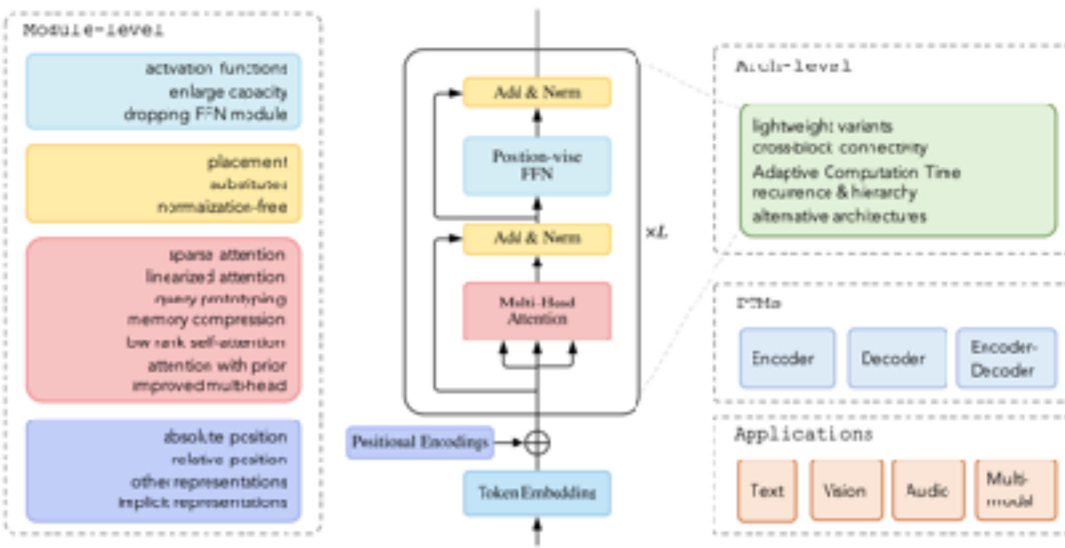
We only have skimmed the surface…



- Architecture Tuning Matters
- Pre-Training Matters
- Sparse Attention Helps with length
- Positional Encoding Doesn't
- Recurrence Might… ?
- **X-formers are NOT just for Text**



Hao et al. 2019

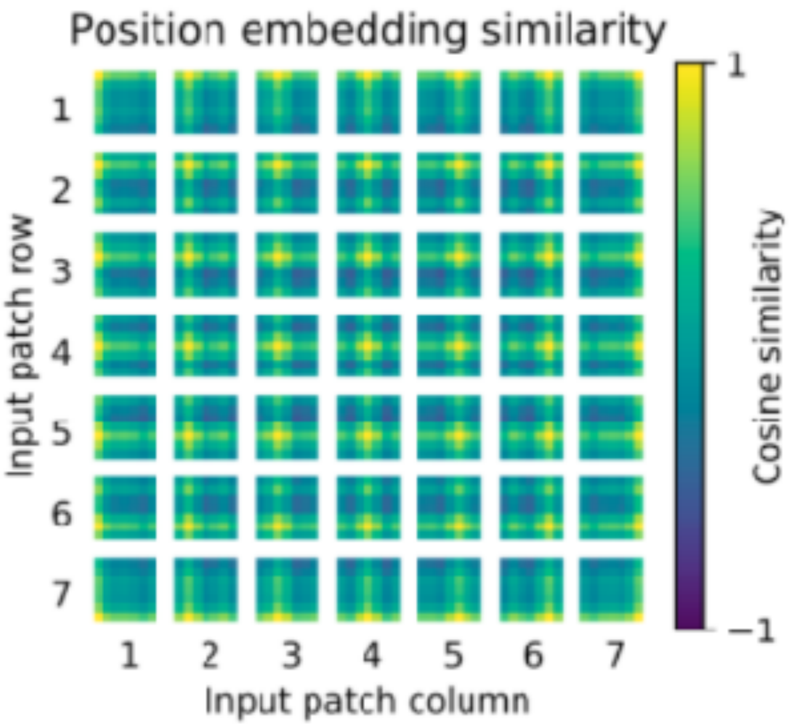


Lin et al "Survey of X-formers, 2021, https://arxiv.org/pdf/2106.04554.pdf

Vision Transformers

Vision Transformers

- Divide image into patches
  - Treat each patch as something to encode separately
  - Flatten each patch
  - Put through dense layer
- Add positional encoding based on position of patch
  - for 7x7 patch, there are 49 positions
- Put into transformer. Same as text transformers …
- **But you need a lot of data**
  - 14M or more images seems to be sweet spot



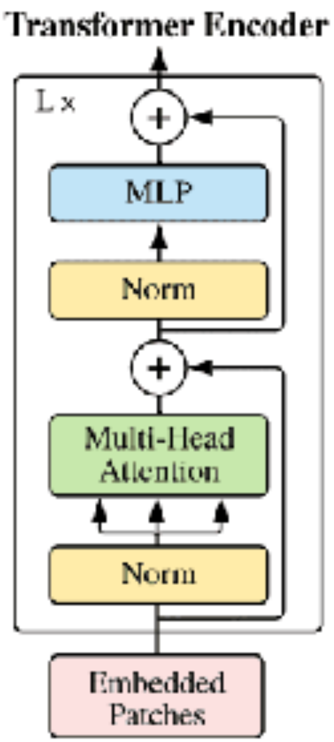
Vision Transformers Video

https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html?m=1

ViT Architectures

- *D* is size of patch embedding
- Uses skip connections (all size *D*)
- Multi-headed self attention (MSA) takes *D* input `patch_embed + pos_embed`
- Main difference in architectures
  - *L* blocks used (*i.e.*, “layers”)
  - *H* heads in each layer (*i.e.*, “heads”)
  - MLP head is final classifier

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E}; \cdots ;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos},$$
$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1},$$
$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell,$$
$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$



Model	Layers	Hidden size *D*	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

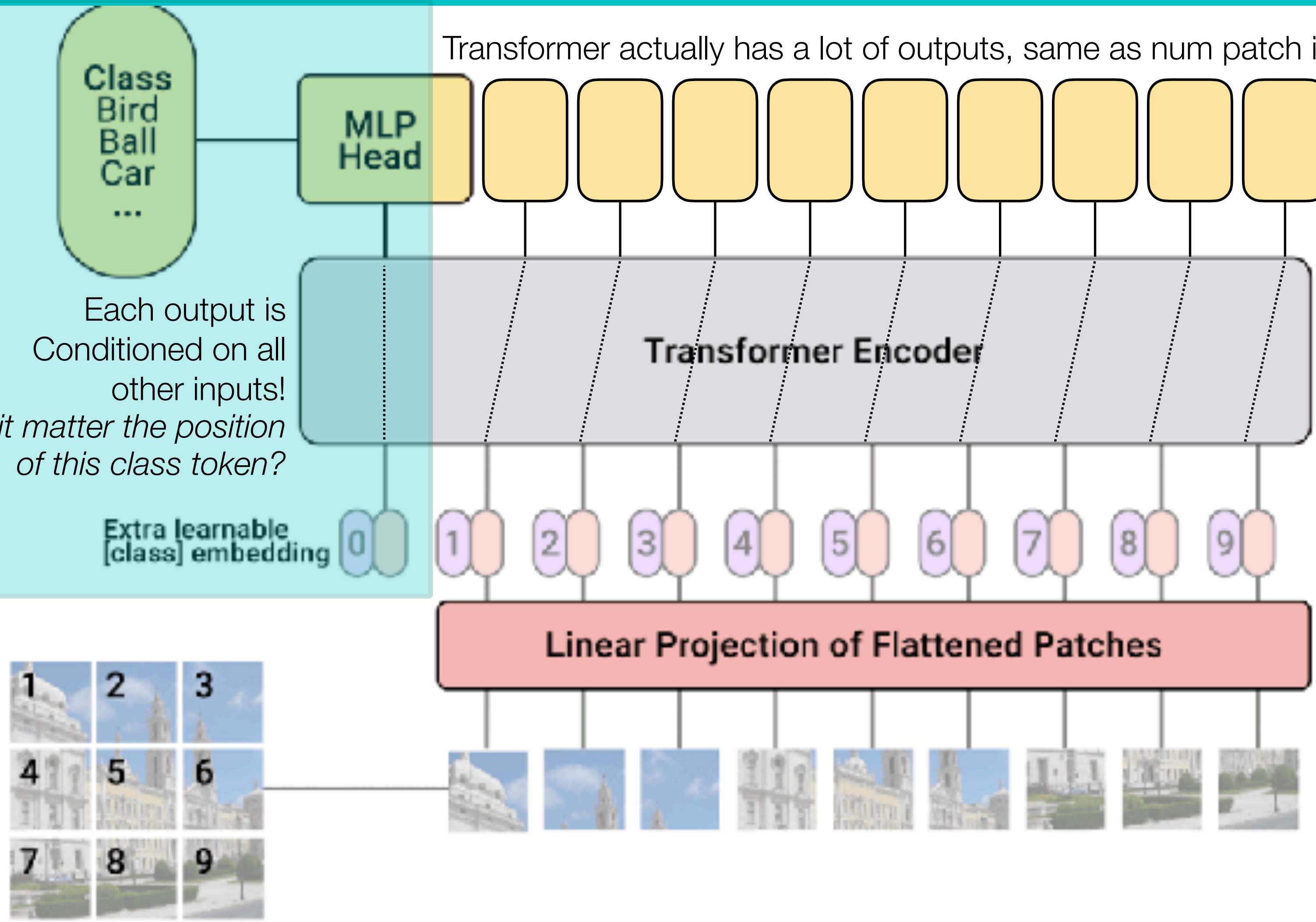
ResNet50: 23M

Dosovitskiy *et al.* (2021) An image is worth 16x16 words: Transformers for Image Recognition at Scale https://arxiv.org/abs/2010.11929

What is the learnable class embedding?

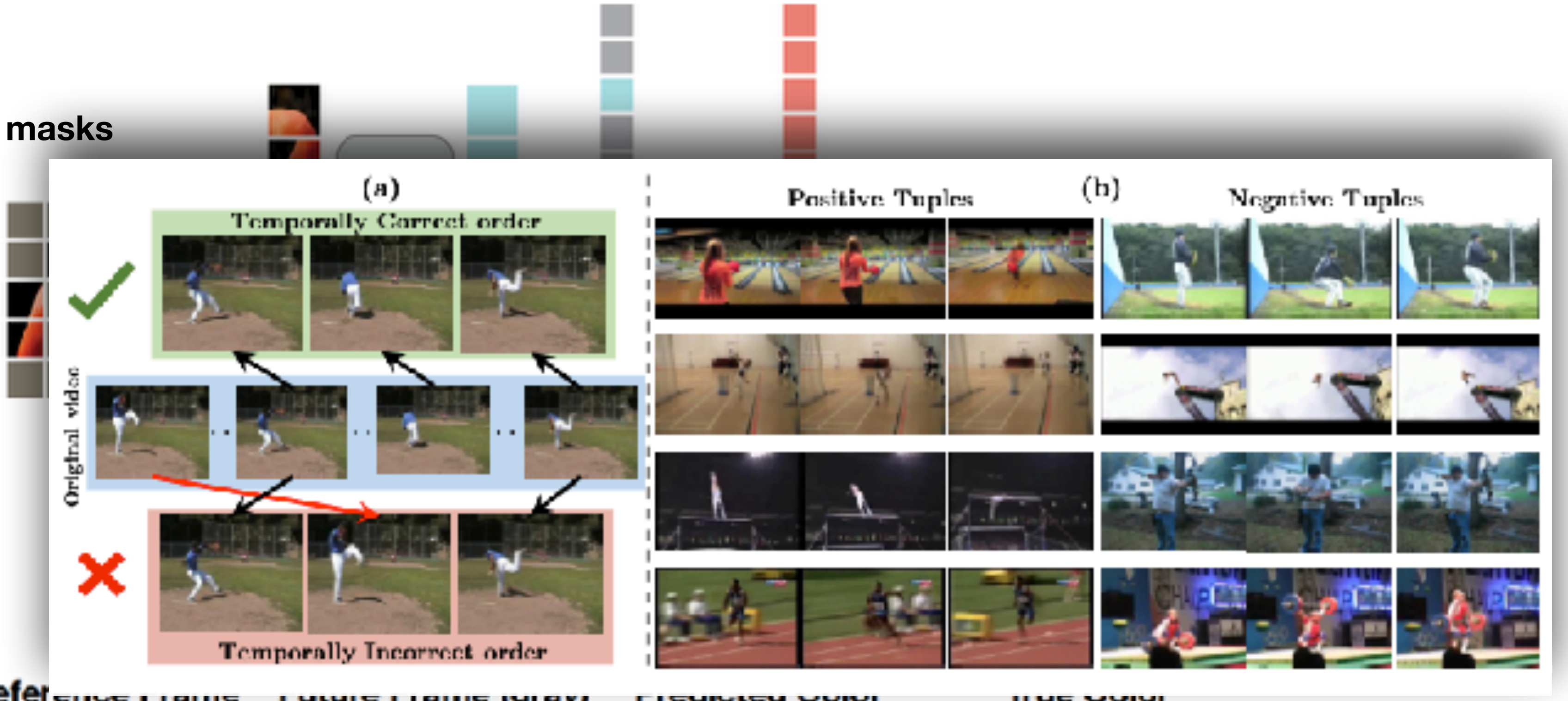
Transformer actually has a lot of outputs, same as num patch inputs

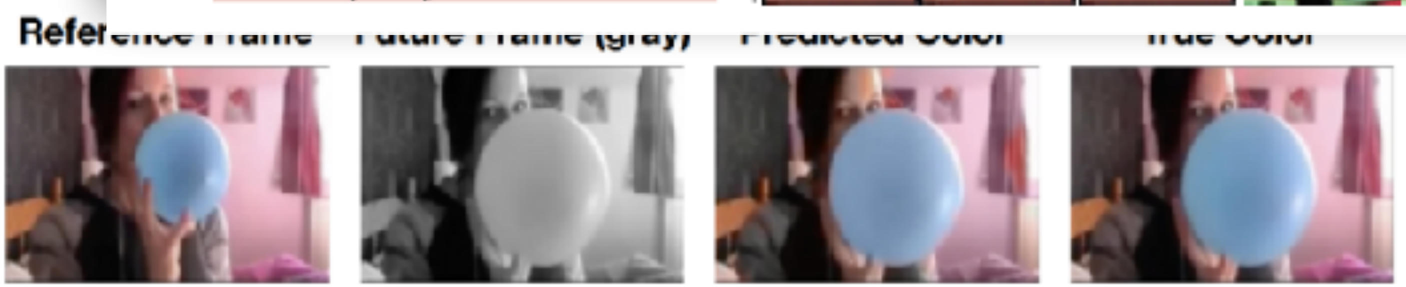
Each output is
Conditioned on all
other inputs!
Does it matter the position
of this class token?



Pre-training for ViT

Fill in masks

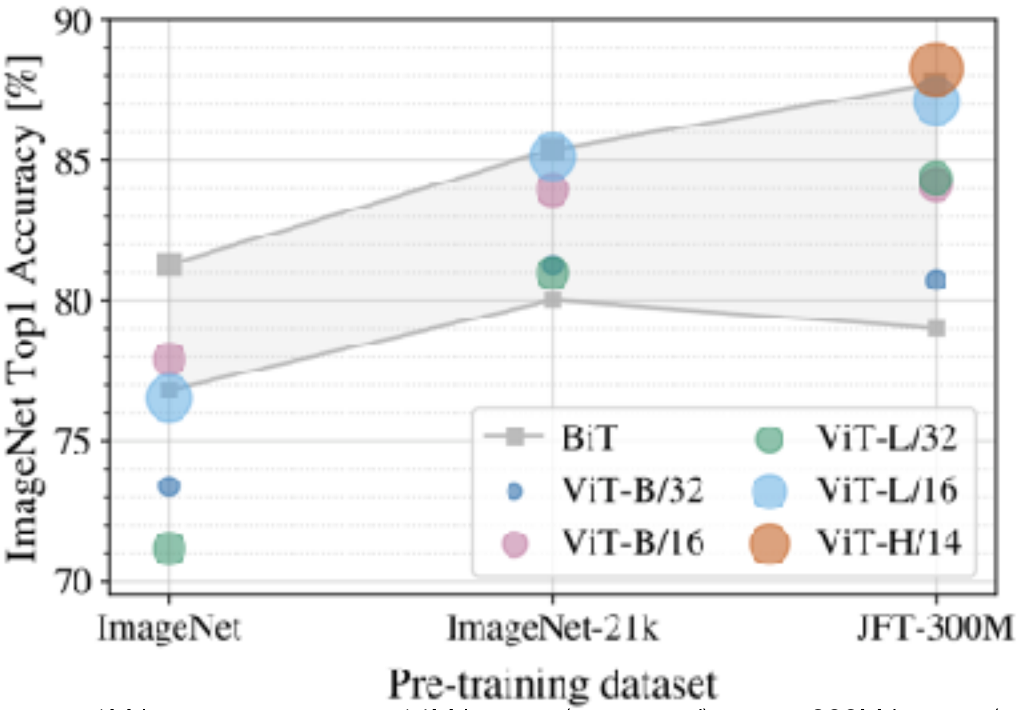




https://www.fast.ai/2020/01/13/self_supervised/

Fine tuning: Do they work?

- Yes, but you need to do some work
- Less than 14M images for pre-training? Use ResNet.



~1M Images
1k classes

~14M Images (part. crawl)
21k classes

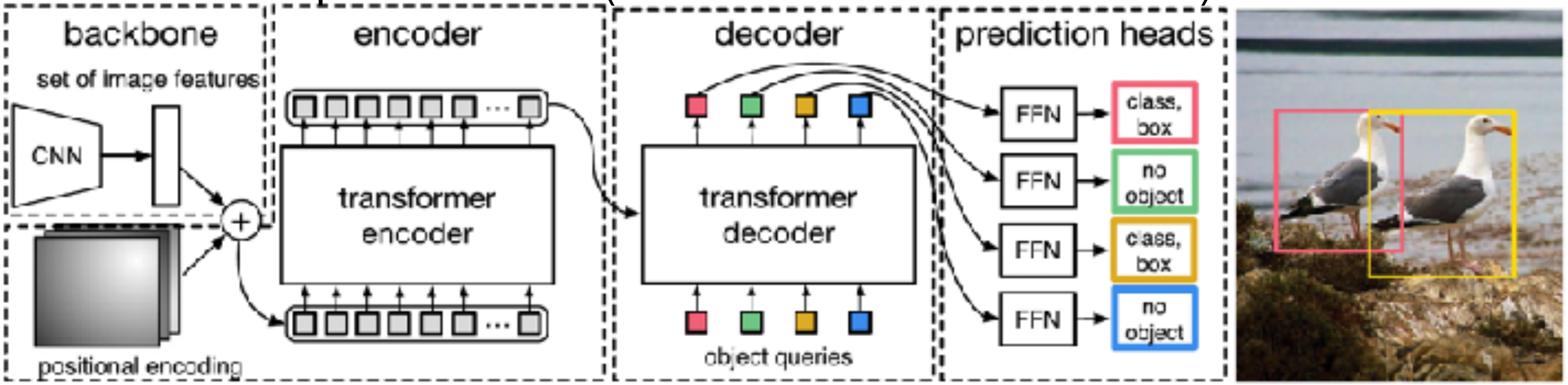
~300M Images (crawled)
18k classes

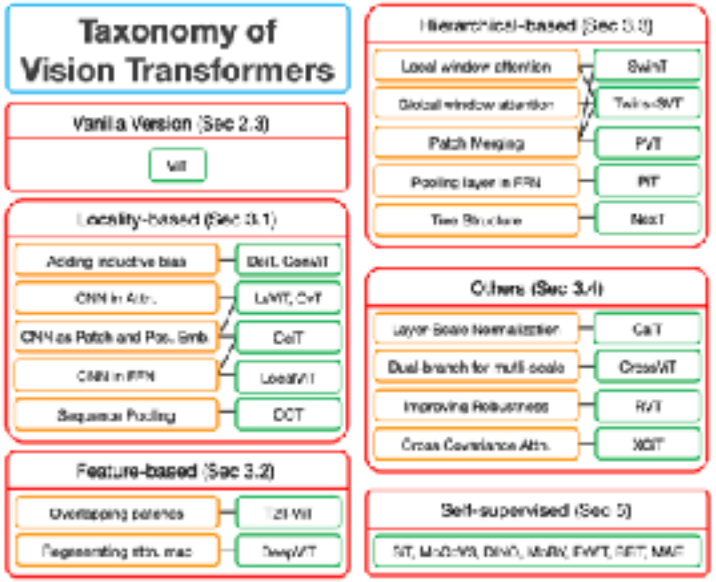
Transfer Learning From Huge ViT

	ViT-H	Previous SOTA
ImageNet	88.55	88.5
ImageNet-ReaL	90.72	90.55
Cifar-10	99.50	99.37
Cifar-100	94.55	93.51
Pets	97.56	96.62
Flowers	99.68	99.63

Many Variants of the ViT

- One example: DETR (Detection Transformer)





- ViT is still an ongoing area of research
- Input Patch Structure (overlap)
- Efficient Attention, Cross Attn.
- Methods or SSL
- Image/text generation

Transformer Town Hall

Hugging Face Text transformers: https://huggingface.co/transformers/v3.3.1/pretrained_models.html
Hugging Face ViT: https://huggingface.co/docs/transformers/model_doc/vit
Keras text Transformers: https://keras.io/guides/keras_nlp/transformer_pretraining/
Keras ViT: https://github.com/faustomorales/vit-keras