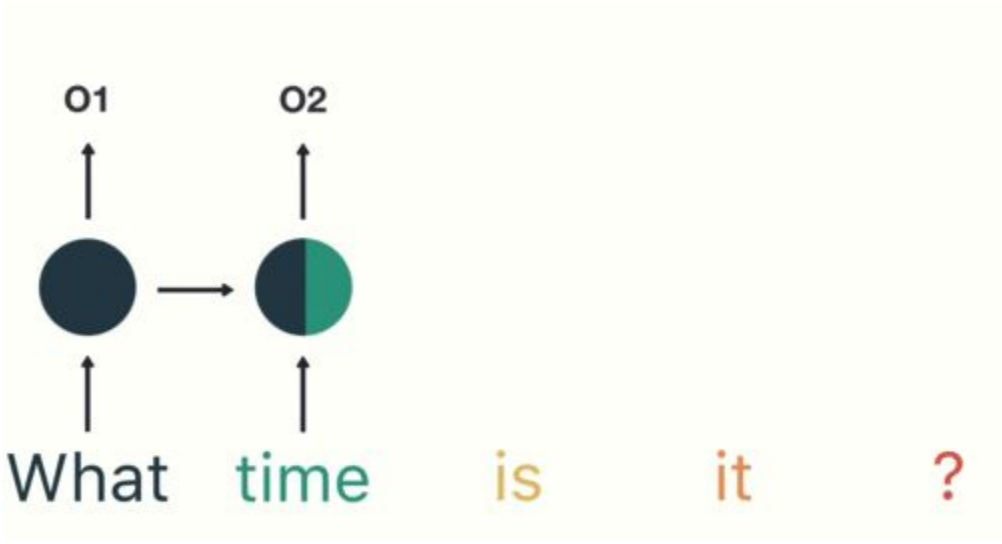
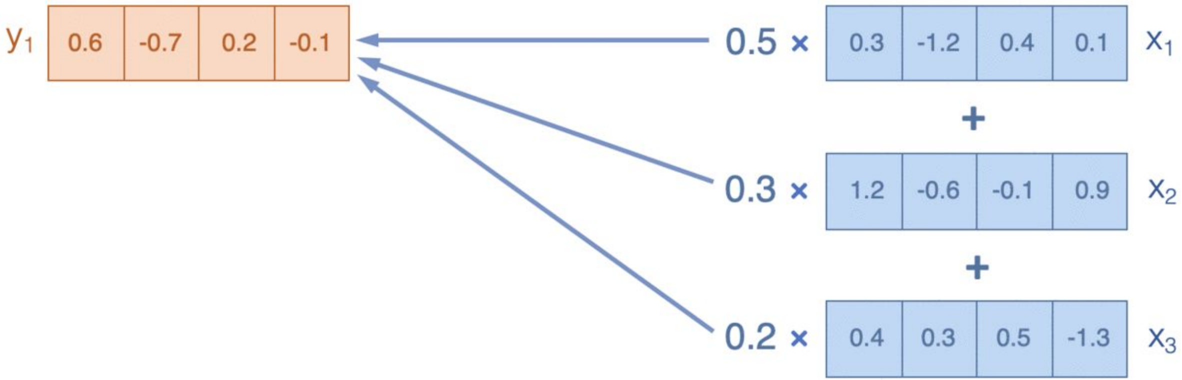
Attention

RNNs



Source: https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

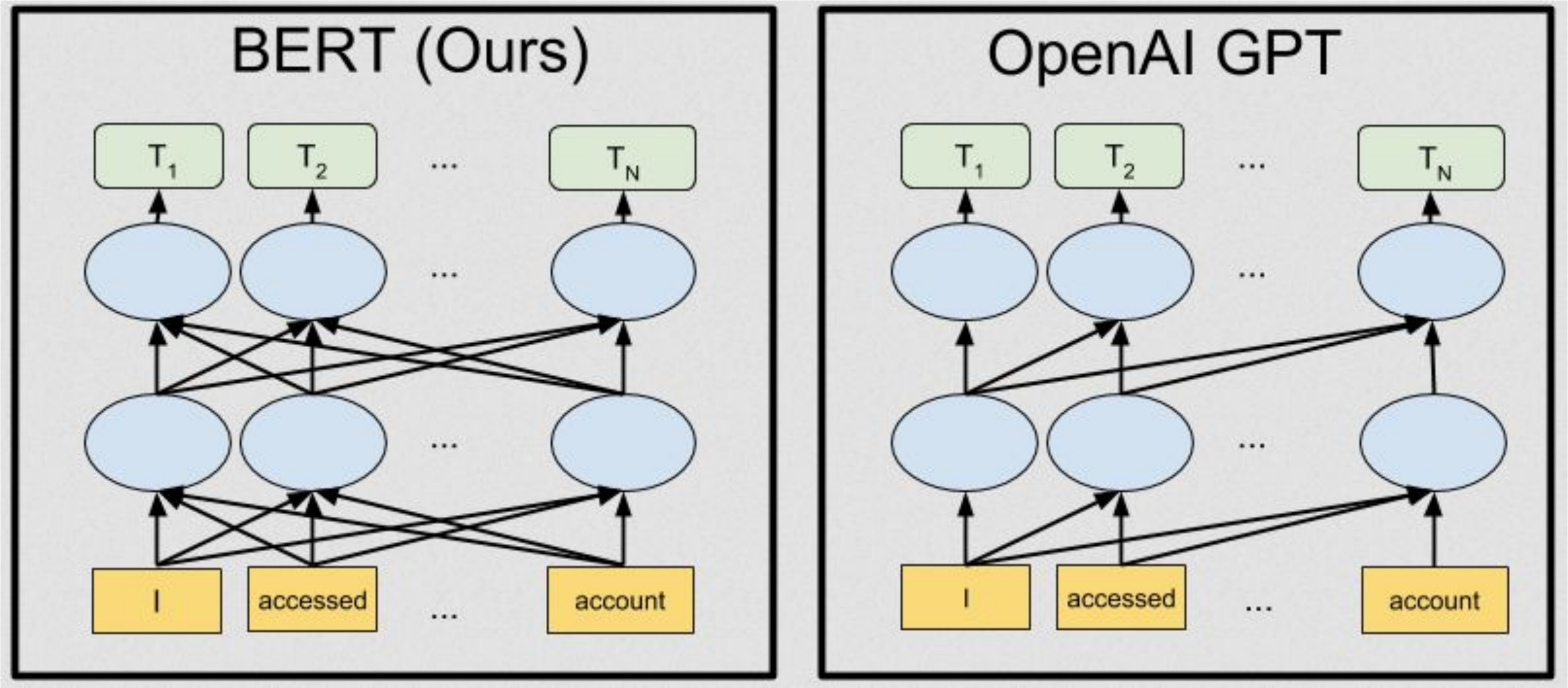
Attention



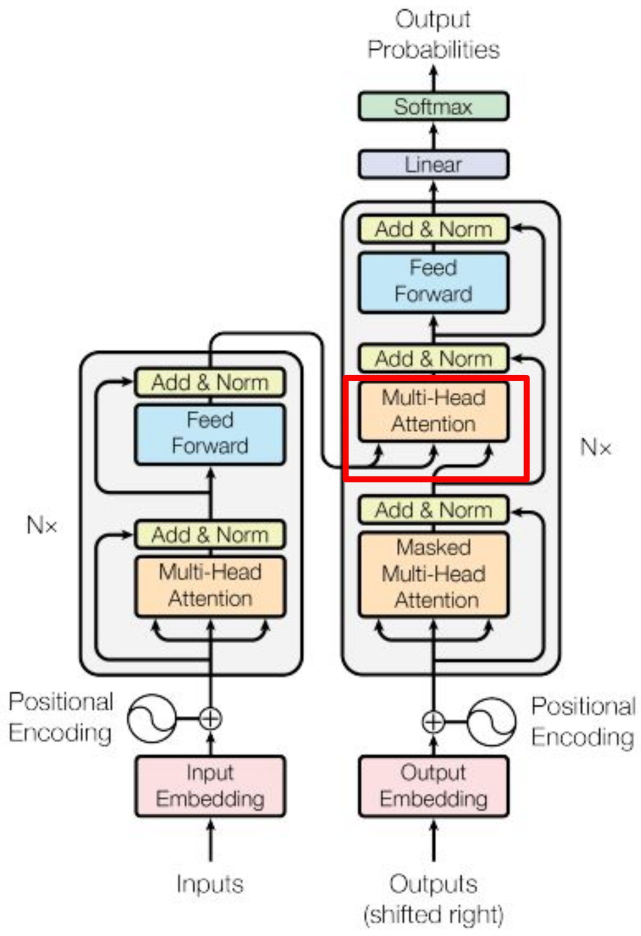
$$y_i = \sum_{j \in S} w_{ij}(Vx_j)$$

$$w_{ij} = \mathrm{softmax}_j(Qx_i \cdot Kx_j)$$

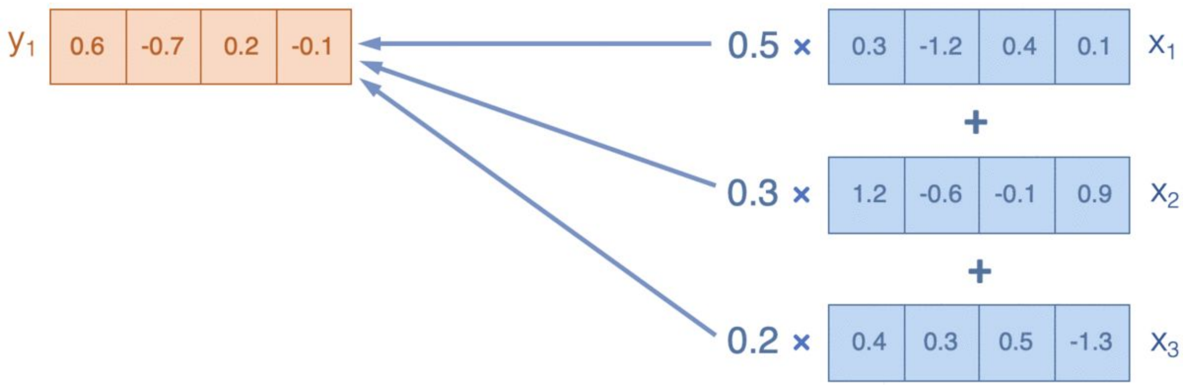
Source: https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1



Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. ***BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding***. 2018



Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit,Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz,and Polosukhin, Illia. ***Attention is all you need***. 2017.




```python
class BertEmbeddings(nn.Module):
    ...
    def forward(input_ids: list[int], position_ids: list[int]):
        inputs_embeds = self.word_embeddings(input_ids)
        position_embeddings = self.position_embeddings(position_ids)
        embeddings = inputs_embeds + position_embeddings
        return embeddings
```

Sinusoidal Encodings

Intuition: Binary Numbers -> Sine/Cos Functions

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	10 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

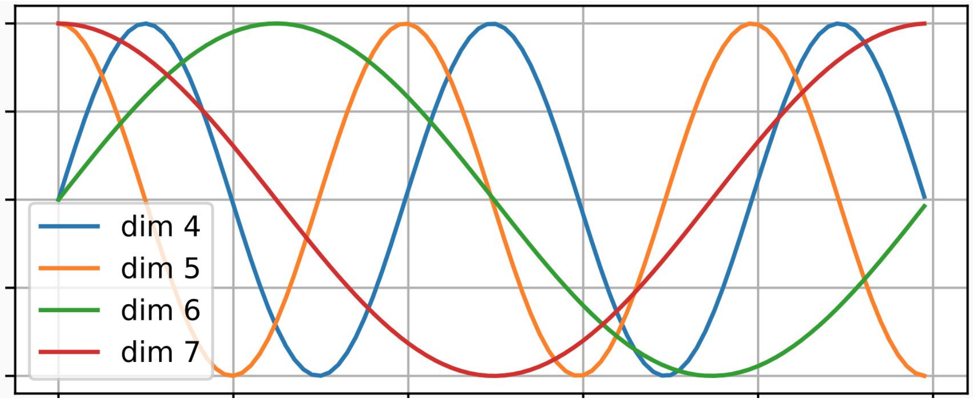


Image Transformer

- Naive solution
  - Each pixel attends to every other pixel
  - Sequence length quickly becomes intractable as resolution increases
- Better solution - Local self-attention
  - Limit sequence length by only attending to a local area of pixels
  - Borrows locality from CNNs
  - Can increase receptive field size without increasing parameters.

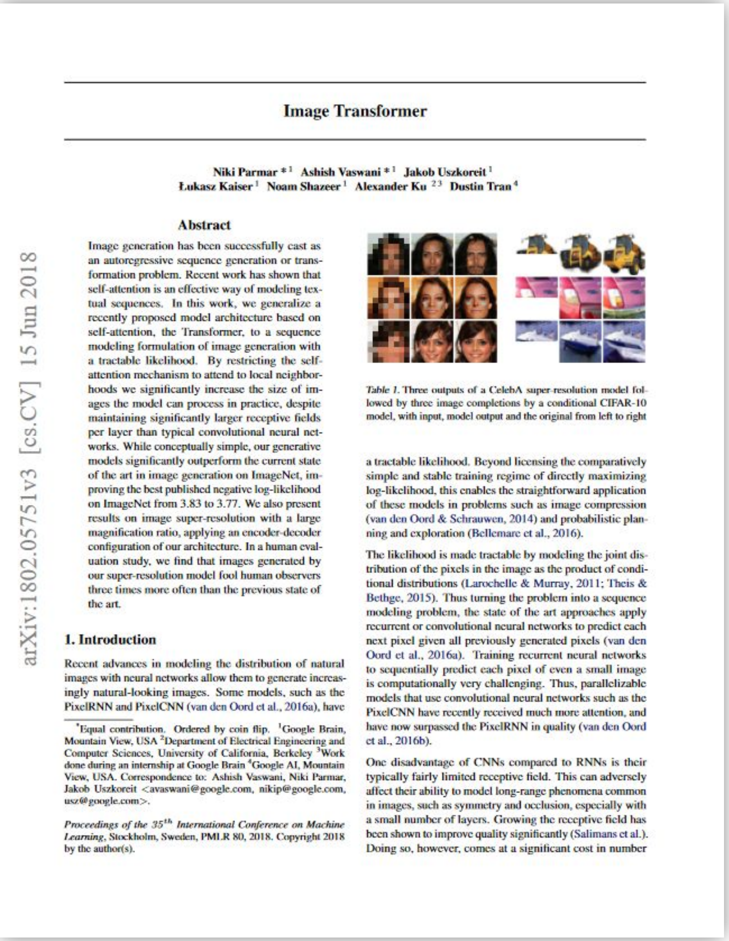


Image Transformer

Niki Parmar * [1] Ashish Vaswani * [1] Jakob Uszkoreit [1]
Łukasz Kaiser [1] Noam Shazeer [1] Alexander Ku [2 3] Dustin Tran [4]

Abstract

Image generation has been successfully cast as an autoregressive sequence generation or transformation problem. Recent work has shown that self-attention is an effective way of modeling textual sequences. In this work, we generalize a recently proposed model architecture based on self-attention, the Transformer, to a sequence modeling formulation of image generation with a tractable likelihood. By restricting the self-attention mechanism to attend to local neighborhoods we significantly increase the size of images the model can process in practice, despite maintaining significantly larger receptive fields per layer than typical convolutional neural networks. While conceptually simple, our generative models significantly outperform the current state of the art in image generation on ImageNet, improving the best published negative log-likelihood on ImageNet from 3.83 to 3.77. We also present results on image super-resolution with a large magnification ratio, applying an encoder-decoder configuration of our architecture. In a human evaluation study, we find that images generated by our super-resolution model fool human observers three times more often than the previous state of the art.

1. Introduction

Recent advances in modeling the distribution of natural images with neural networks allow them to generate increasingly natural-looking images. Some models, such as the PixelRNN and PixelCNN (van den Oord et al., 2016a), have

Table 1. Three outputs of a CelebA super-resolution model followed by three image completions by a conditional CIFAR-10 model, with input, model output and the original from left to right

a tractable likelihood. Beyond licensing the comparatively simple and stable training regime of directly maximizing log-likelihood, this enables the straightforward application of these models in problems such as image compression (van den Oord & Schrauwen, 2014) and probabilistic planning and exploration (Bellemare et al., 2016).

The likelihood is made tractable by modeling the joint distribution of the pixels in the image as the product of conditional distributions (Larochelle & Murray, 2011; Theis & Bethge, 2015). Thus turning the problem into a sequence modeling problem, the state of the art approaches apply recurrent or convolutional neural networks to predict each next pixel given all previously generated pixels (van den Oord et al., 2016a). Training recurrent neural networks to sequentially predict each pixel of even a small image is computationally very challenging. Thus, parallelizable models that use convolutional neural networks such as the PixelCNN have recently received much more attention, and have now surpassed the PixelRNN in quality (van den Oord et al., 2016b).

One disadvantage of CNNs compared to RNNs is their typically fairly limited receptive field. This can adversely affect their ability to model long-range phenomena common in images, such as symmetry and occlusion, especially with a small number of layers. Growing the receptive field has been shown to improve quality significantly (Salimans et al.). Doing so, however, comes at a significant cost in number

*Equal contribution. Ordered by coin flip. [1]Google Brain, Mountain View, USA [2]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley [3]Work done during an internship at Google Brain [4]Google AI, Mountain View, USA. Correspondence to: Ashish Vaswani, Niki Parmar, Jakob Uszkoreit <avaswani@google.com, nikip@google.com, usz@google.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

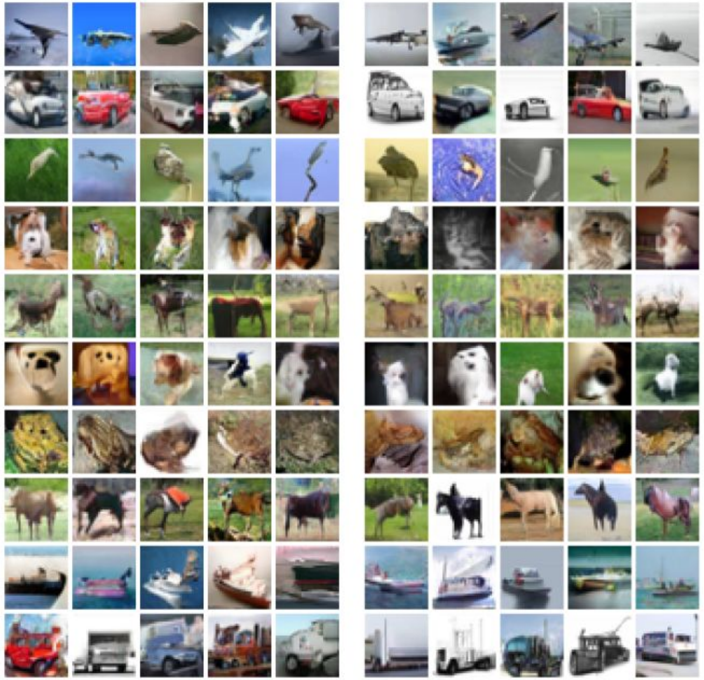
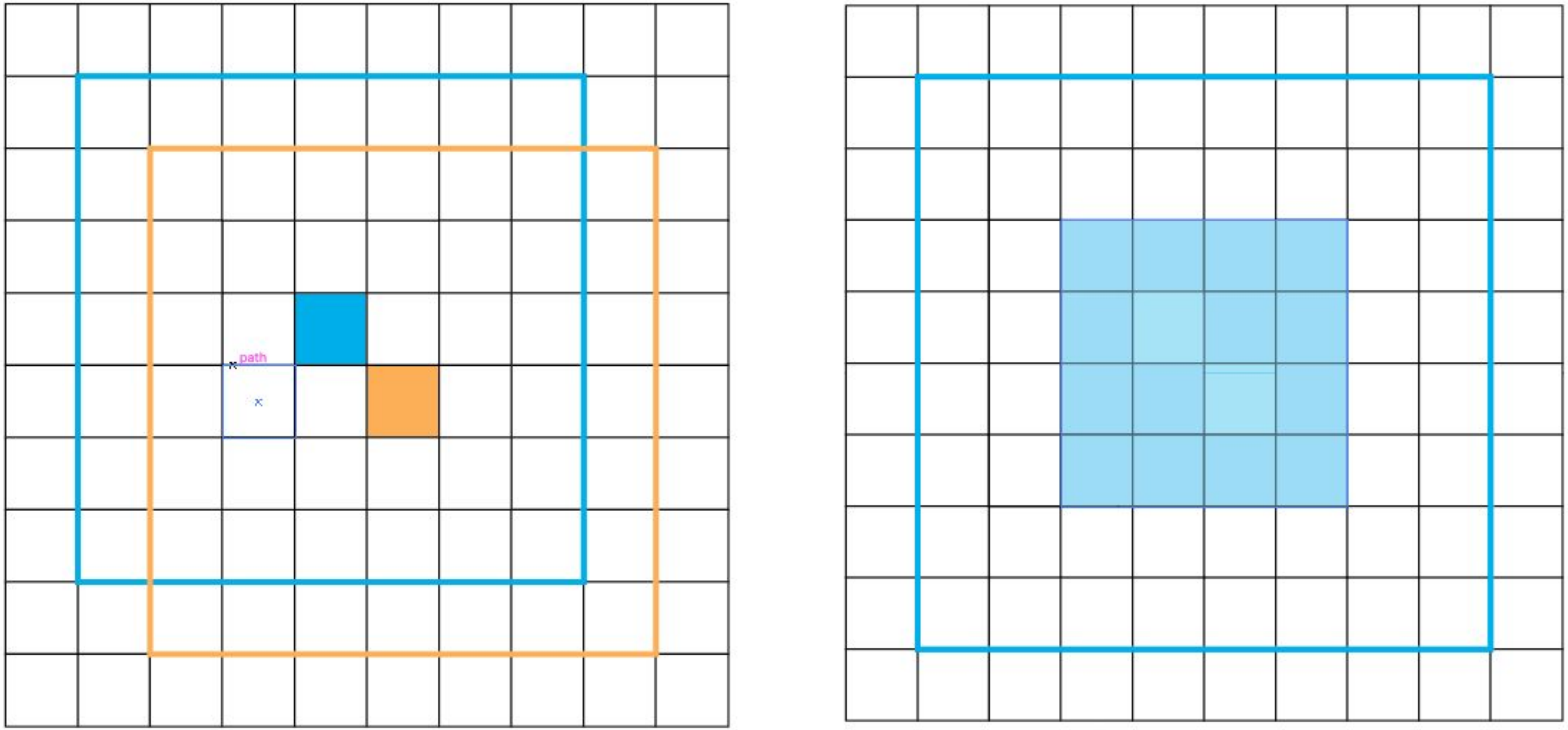
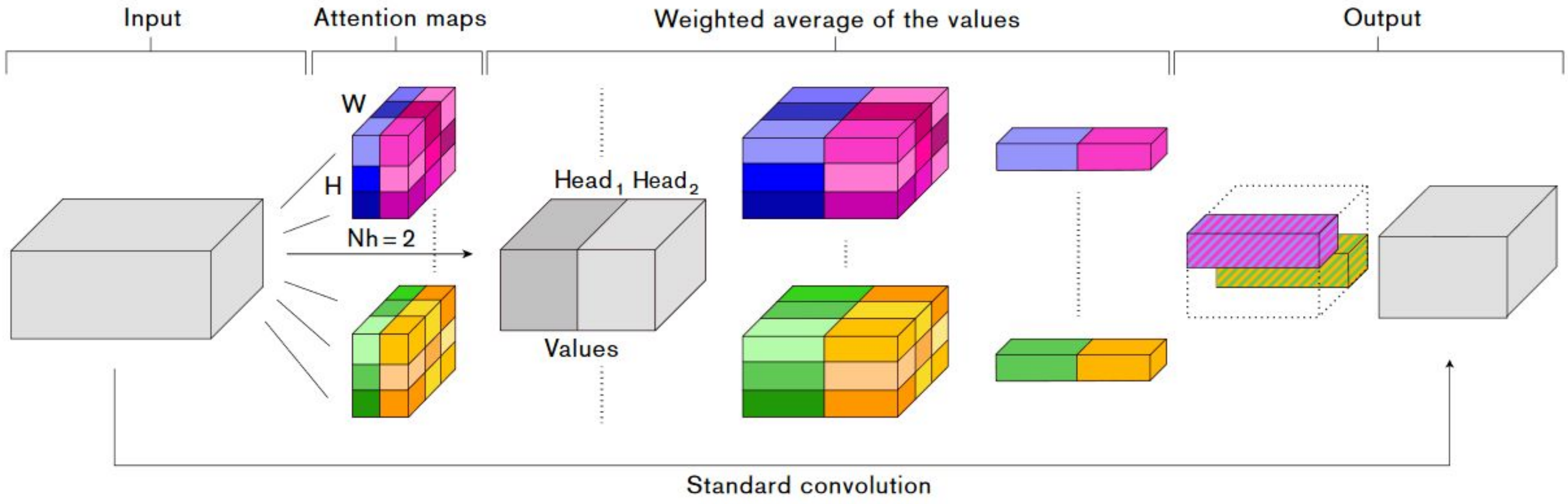


Table 3. Conditional image generations for all CIFAR-10 categories. Images on the left are from a model that achieves 3.03 bits/dim on the test set. Images on the right are from our best non-averaged model with 2.99 bits/dim. Both models are able to generate convincing cars, trucks, and ships. Generated horses, planes, and birds also look reasonable.

Local Attention



Attention Augmented Convolutional Networks



I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens. ***Attention augmented convolutional networks***. In ICCV, 2019.

An Image is Worth 16 x 16 Words

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby

```python
from transformers import BertModel
model = BertModel()
preds = model(image)
```

Recipe for using a transformer

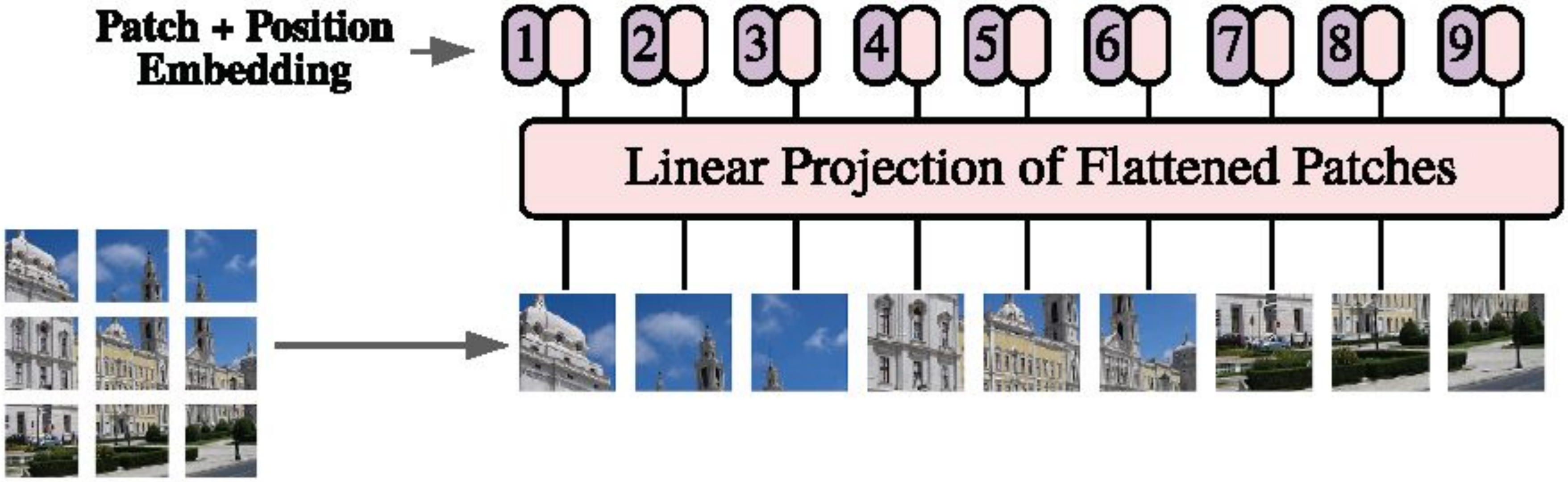
1. Convert your input into a sequence
   (make sure the sequence isn't too long)

2. Embed each element into a feature representation
   (should include information about the element and its position)

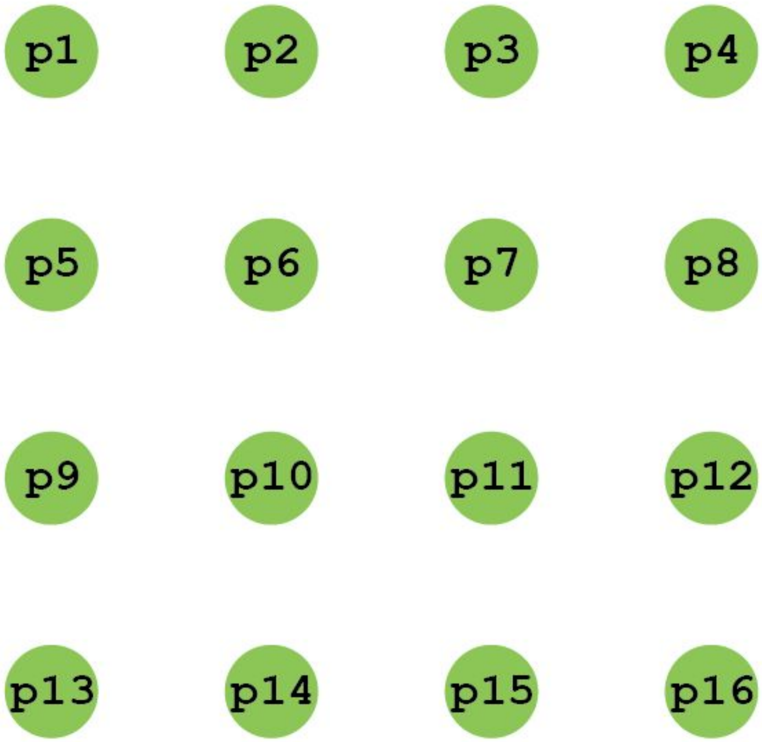
3. Pass the sequence through the transformer

Convert an image into a sequence

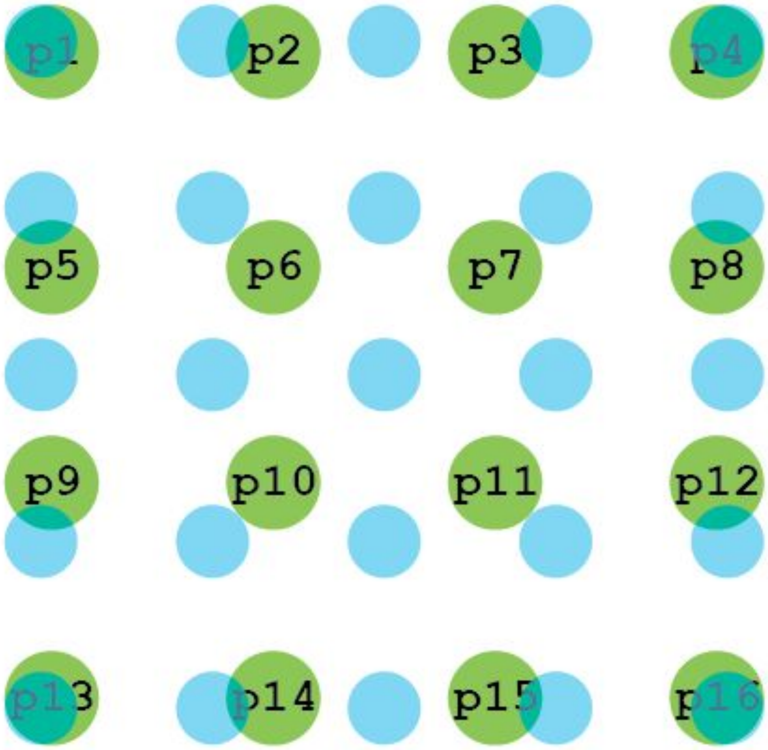
Embed each image patch

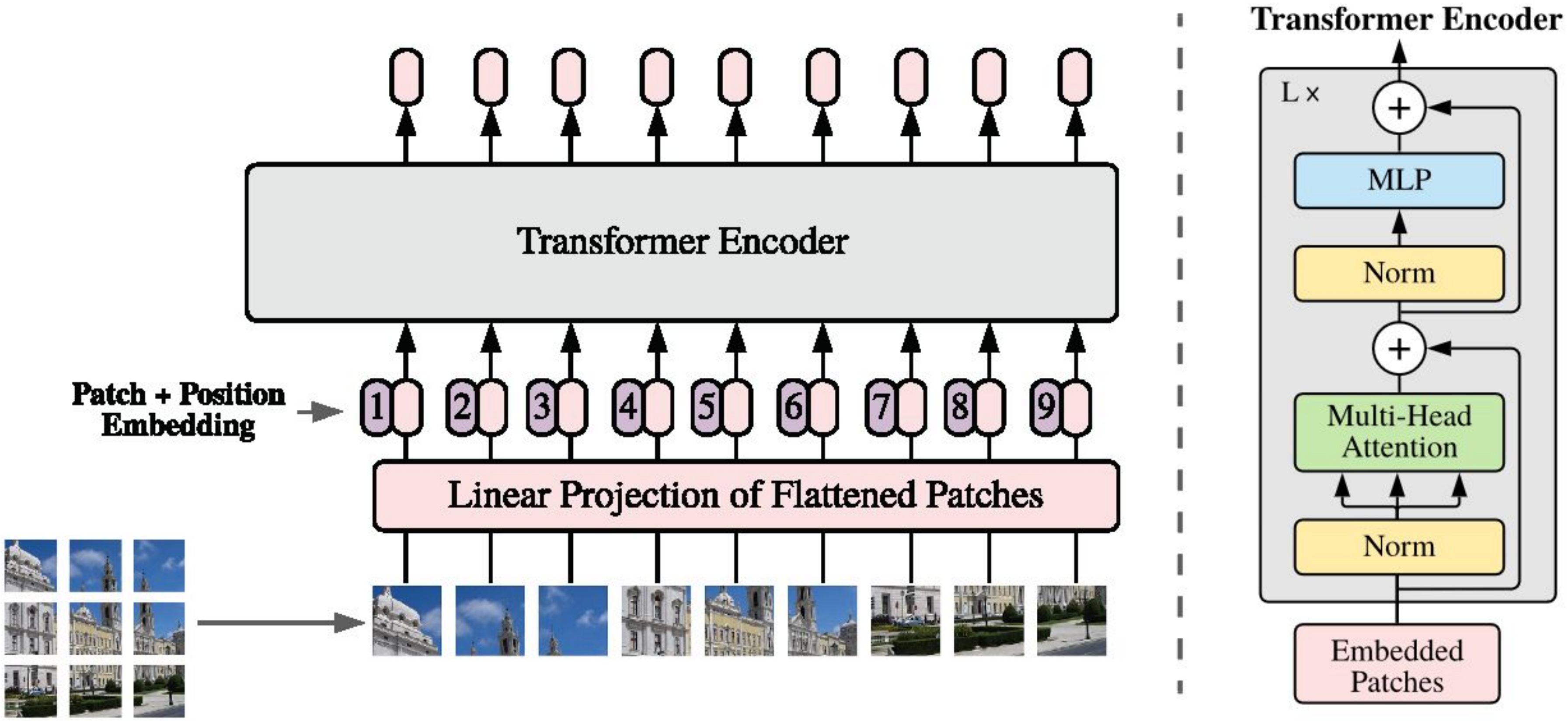


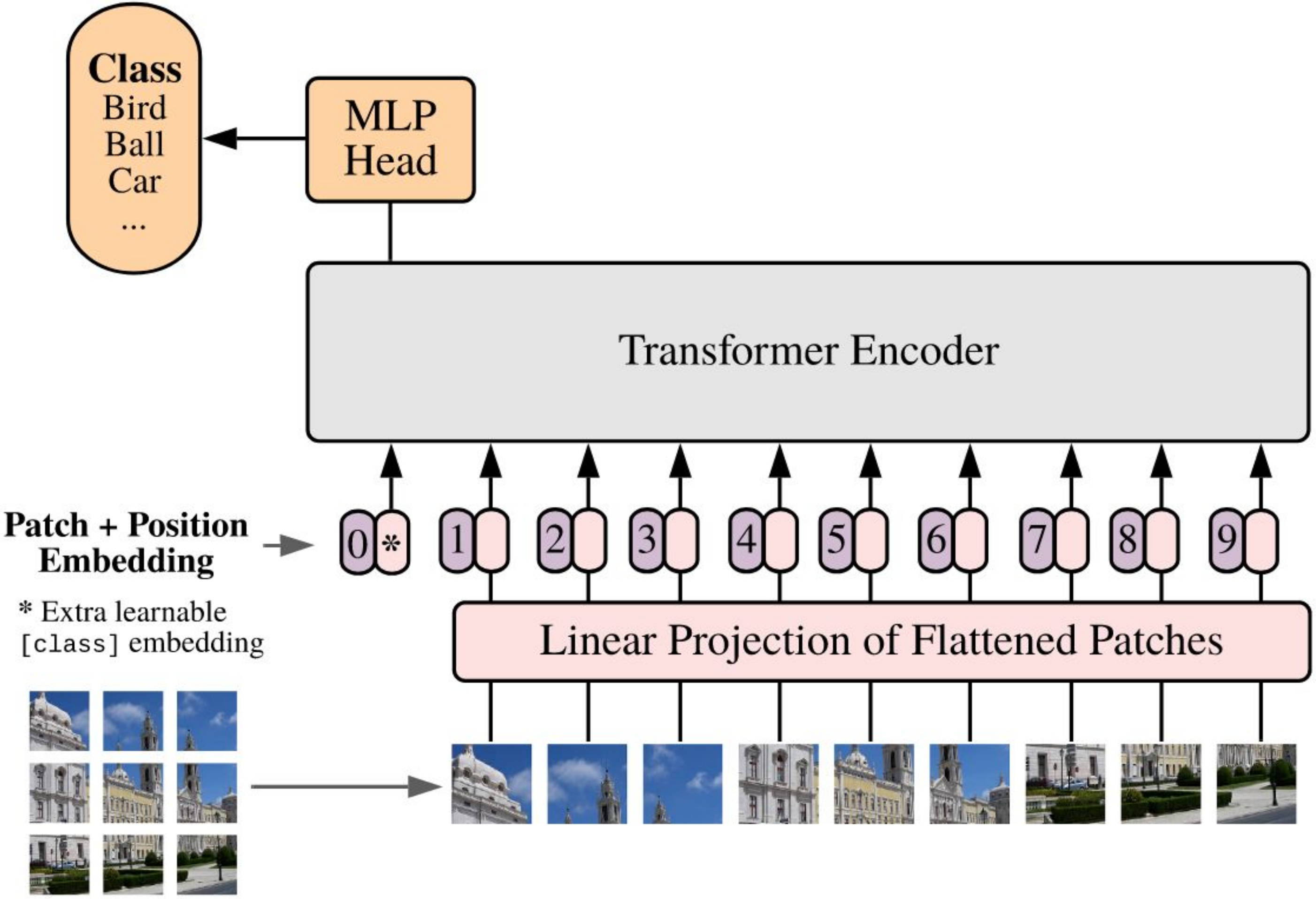
Positional Embeddings



Positional Embeddings



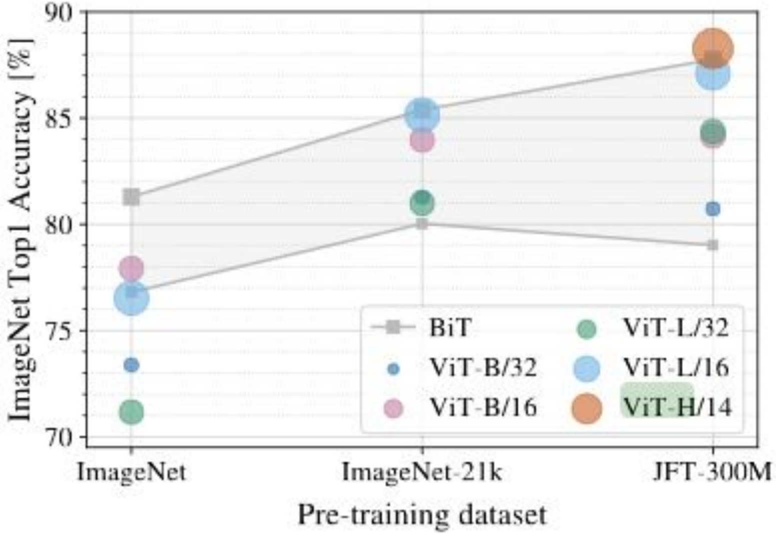

Transformer Encoder
Patch + Position Embedding
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L ×
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches


Class
Bird
Ball
Car
...
MLP
Head
Transformer Encoder
Patch + Position
Embedding
0
*
1
2
3
4
5
6
7
8
9
* Extra learnable
[class] embedding
Linear Projection of Flattened Patches

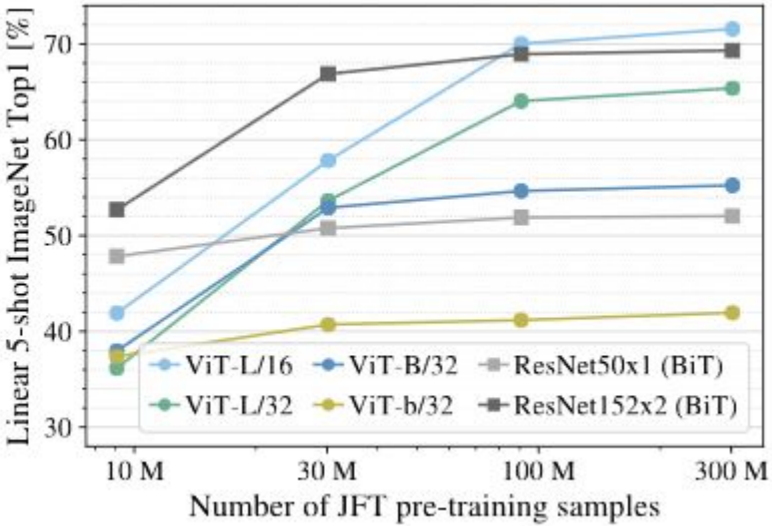
Results

	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

Results



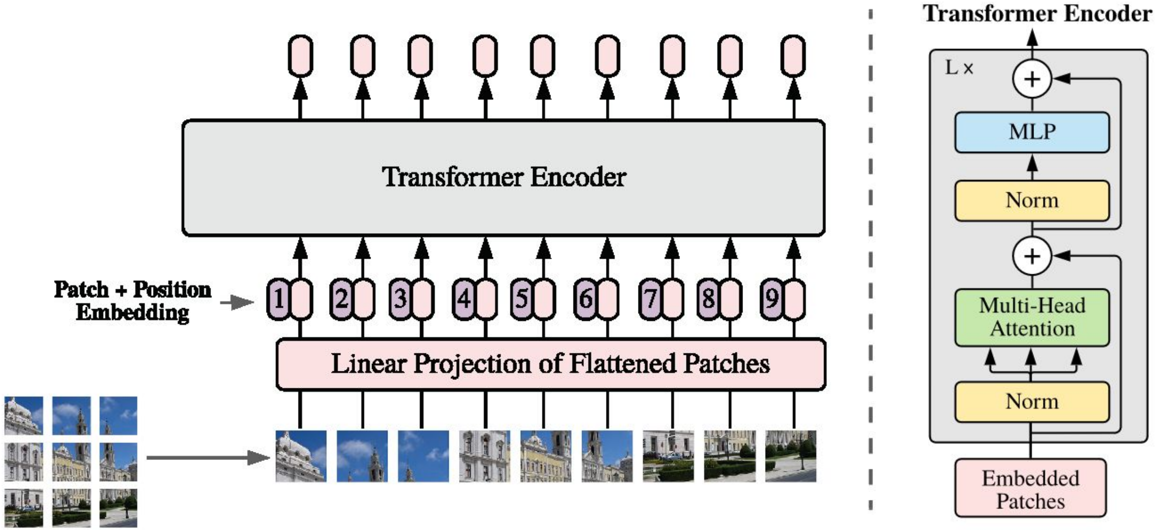
ImageNet Top1 Accuracy [%]
90
85
80
75
70
ImageNet
ImageNet-21k
JFT-300M
Pre-training dataset
BiT
ViT-B/32
ViT-B/16
ViT-L/32
ViT-L/16
ViT-H/14

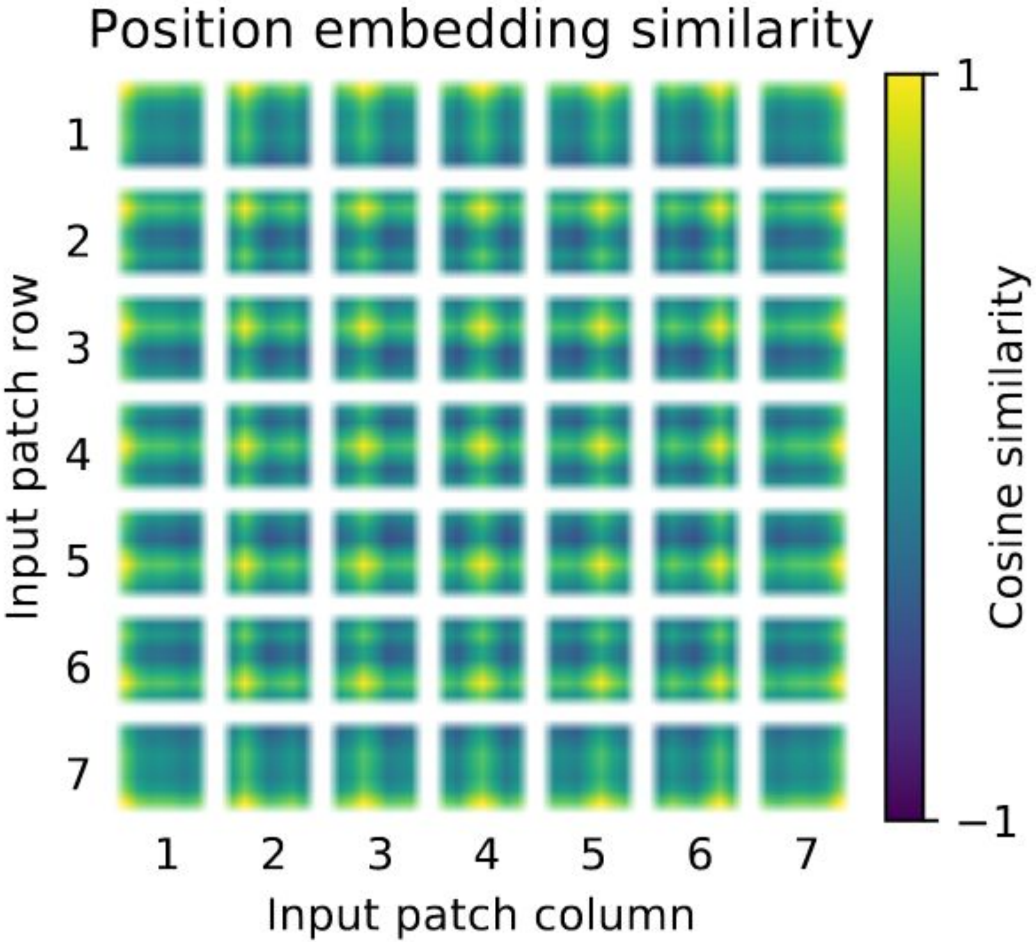


Linear 5-shot ImageNet Top1 [%]
70
60
50
40
30
10 M
30 M
100 M
300 M
Number of JFT pre-training samples
ViT-L/16
ViT-B/32
ResNet50x1 (BiT)
ViT-L/32
ViT-b/32
ResNet152x2 (BiT)

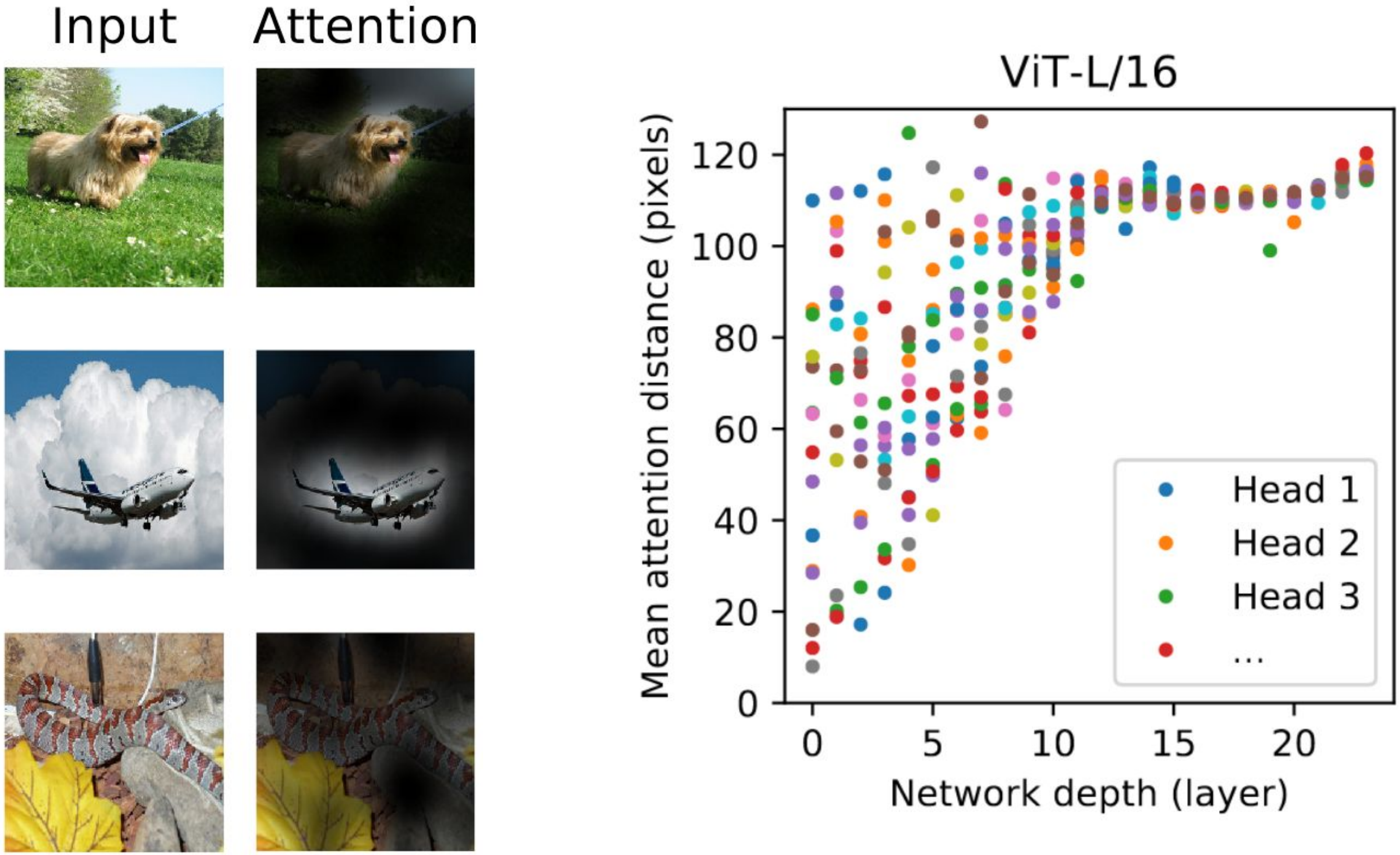
Compared to CNNs

1. Not restricted to a local region. Each layer globally attends to the image.
2. No more translation invariance/equivariance





Position embedding similarity
Input patch row
Input patch column
1
2
3
4
5
6
7
Cosine similarity
1
−1


Input
Attention
ViT-L/16
Mean attention distance (pixels)
0
20
40
60
80
100
120
0
5
10
15
20
Network depth (layer)
Head 1
Head 2
Head 3
...

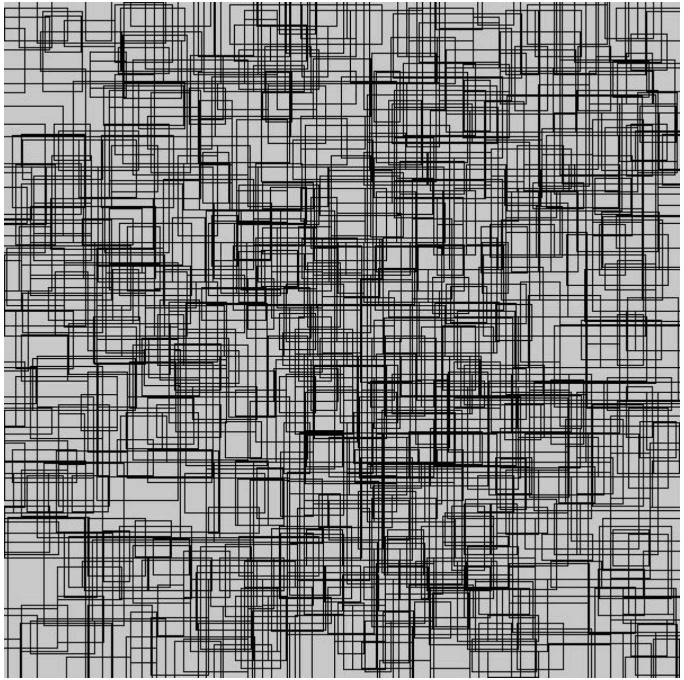
Discussion

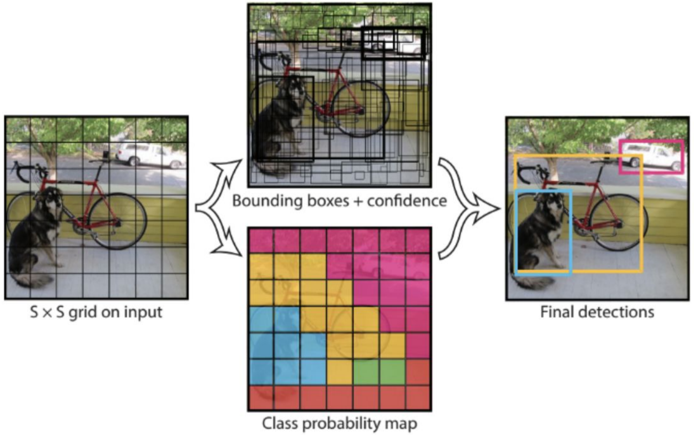
- How important is locality?
- Is this a good way of encoding position?
- Will transformers surpass CNNs as state-of-the art?

End-to-End Object Detection with Transformers

Nicolas Carion*, Francisco Massa*, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoriuko

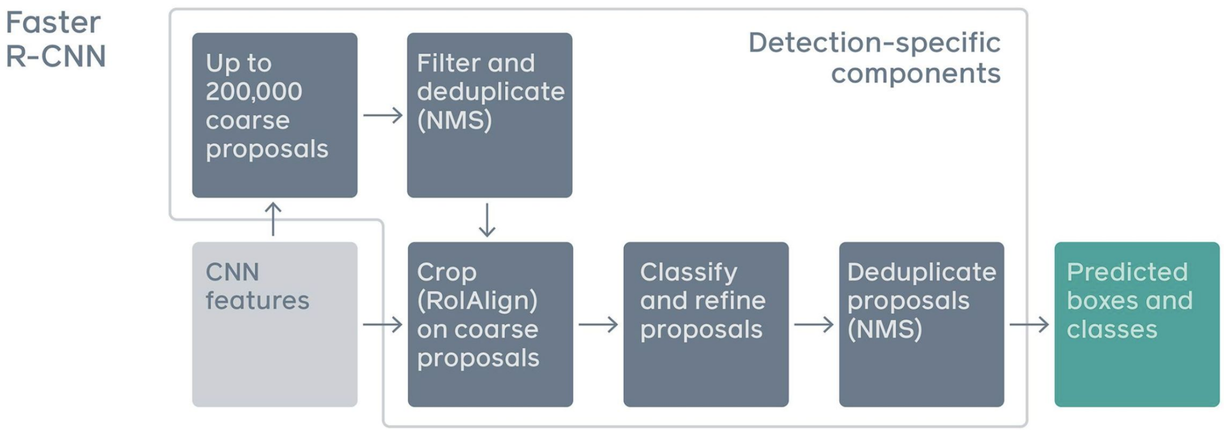
Object Detection Background: Anchors





YOLO

Faster R-CNN



Faster
R-CNN
Detection-specific
components
Up to
200,000
coarse
proposals
Filter and
deduplicate
(NMS)
CNN
features
Crop
(RoIAlign)
on coarse
proposals
Classify
and refine
proposals
Deduplicate
proposals
(NMS)
Predicted
boxes and
classes

Object Detection Background: Non-Maximal Suppression

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
    $B_{nms} \leftarrow \emptyset$
    for $b_i \in B$ **do**
        $discard \leftarrow \text{False}$
        for $b_j \in B$ **do**
            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then**
                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**
                    $discard \leftarrow \text{True}$
        if not $discard$ **then**
            $B_{nms} \leftarrow B_{nms} \cup b_i$
    return B_{nms}

Set Prediction

- No canonical deep learning model to directly predict sets
  - Closest idea is multilabel classification
- Difficulty in avoiding near-duplicates
  - Usually solved through post-processing (non-maximal suppression)
- Usual solution to design a loss based on the Hungarian Algorithm
  - Find bipartite matching between ground truth and prediction
  - Enforces permutation-invariance

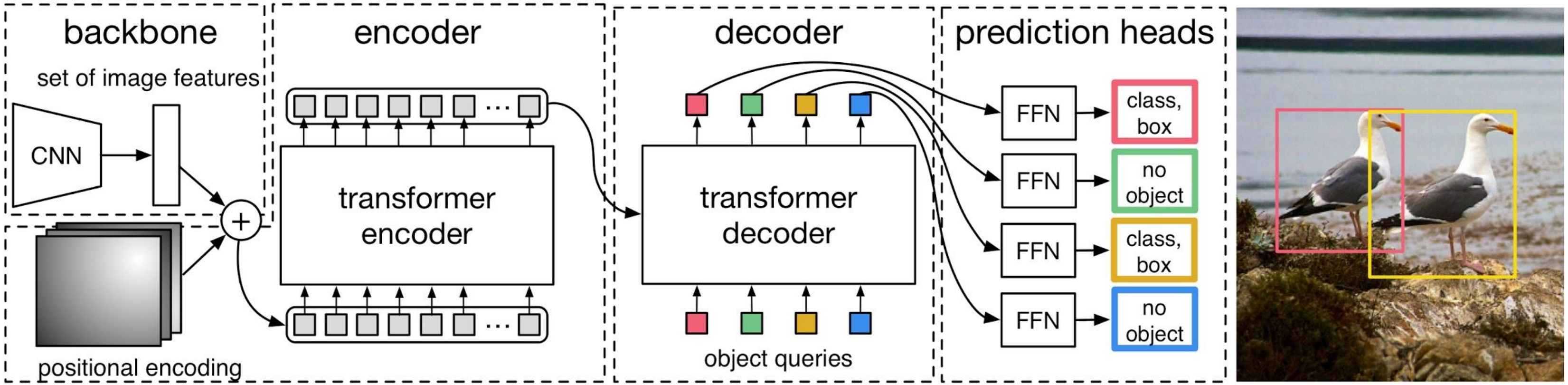
Hungarian Algorithm

- Polynomial time combinatorial optimization algorithm
- Given n workers and tasks, and an n×n matrix containing the cost of assigning each worker to a task, find the cost minimizing assignment.

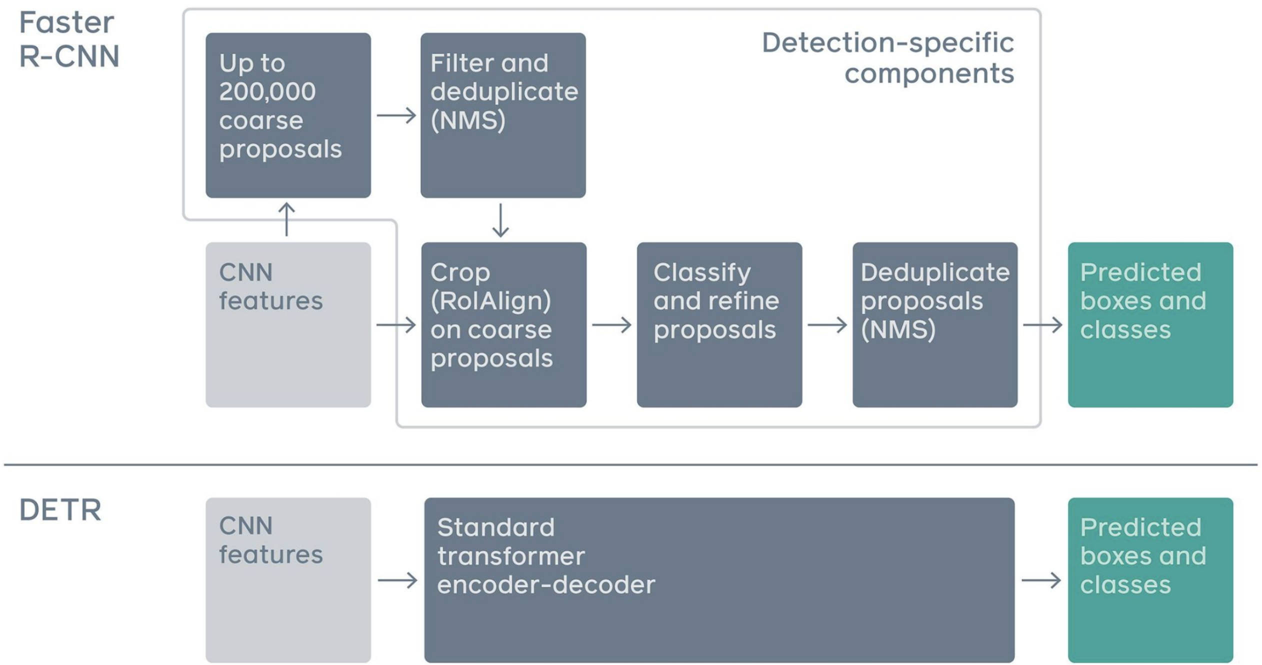
	Clean bathroom	Sweep floors	Wash windows
Paul	\$2	\$3	\$3
Dave	\$3	\$2	\$3
Chris	\$3	\$3	\$2

Minimum cost = \$6, Paul cleans the bathroom, Dave sweeps the floors, and Chris washes the windows.

DEtection **TR**ansformer High Level Architecture

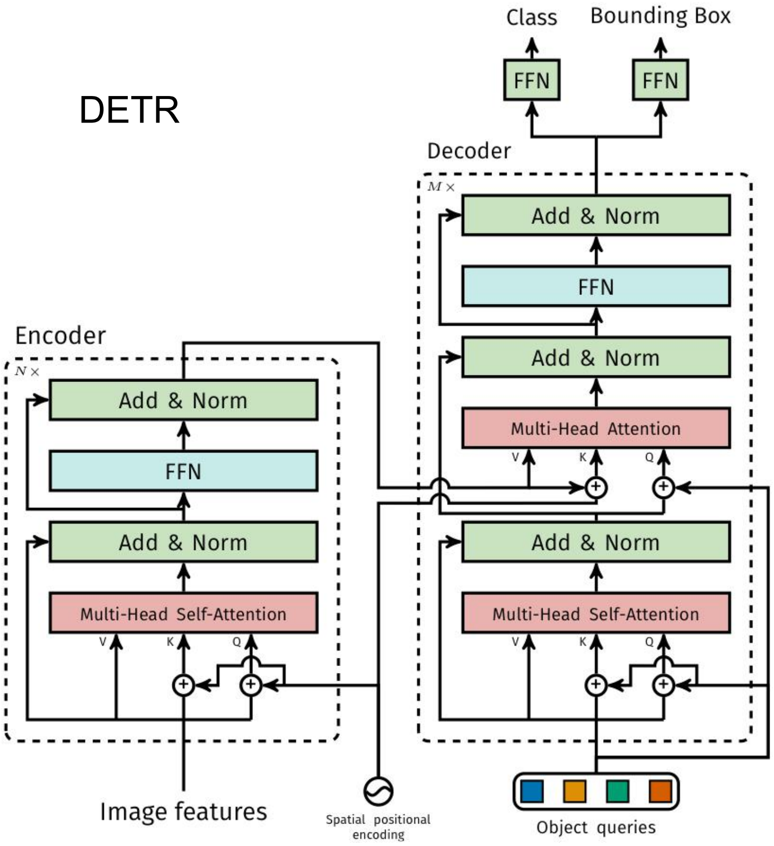


DETR vs Faster R-CNN



DETR Transformers

DETR



Attention is All You Need

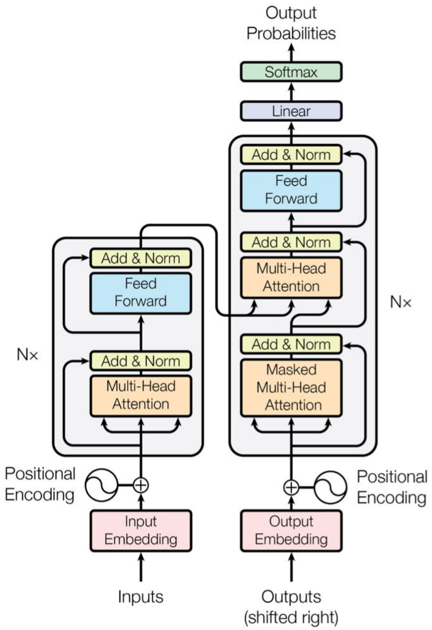


Figure 1: The Transformer - model architecture.

Spatial Positioning

spatial pos. enc. encoder	decoder	output pos. enc. decoder	AP	Δ	AP_{50}	Δ
none	none	learned at input	32.8	-7.8	55.2	-6.5
sine at input	sine at input	learned at input	39.2	-1.4	60.0	-1.6
learned at attn.	learned at attn.	learned at attn.	39.6	-1.0	60.7	-0.9
none	sine at attn.	learned at attn.	39.3	-1.3	60.3	-1.4
sine at attn.	sine at attn.	learned at attn.	**40.6**	-	**61.6**	-

Biparate Loss

Pairwise matching cost between predicted and ground truth objects

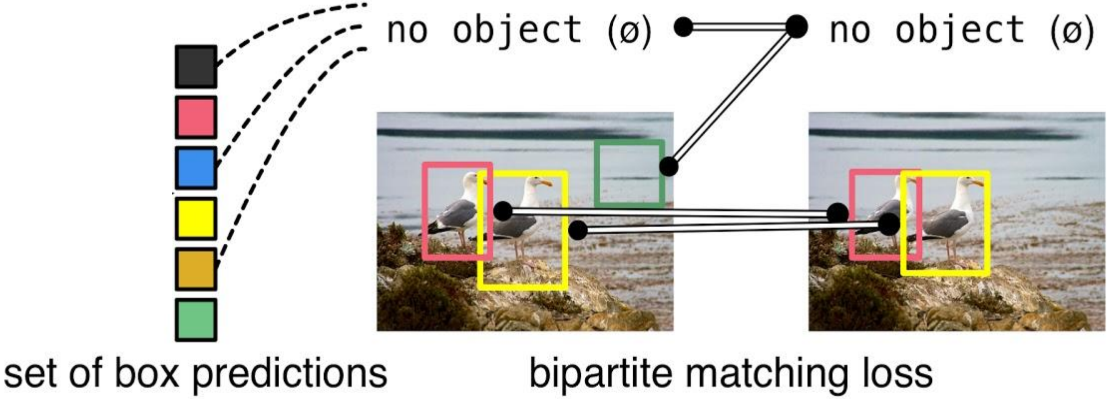
$$\boxed{\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)}),}$$

$y_i = (c_i , b_i)$

c_i is the class target

b_i is a vector that defines ground truth box center coordinates and normalized width and height

Biparate Loss



LMatch

Class prediction

$$\text{LMatch} = \boxed{-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i)} + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \Big] \ ,$$

LMatch

Box Loss

LMatch = $-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i)$ + $\boxed{\mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i))}\Big]$,

$\mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$ = $\lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1$

Biparate Loss

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)}),$$

Negative log-likelihood class prediction

$$\mathcal{L}_{\mathrm{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[\boxed{-\log \hat{p}_{\hat{\sigma}(i)}(c_i)} + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\mathrm{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right] ,$$

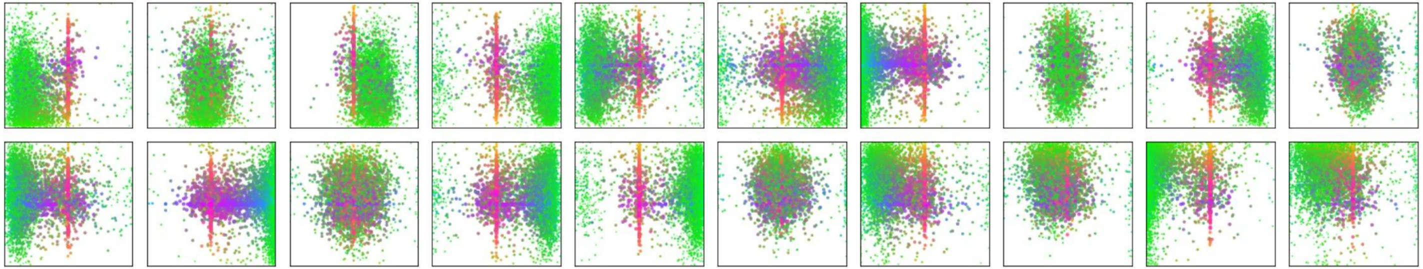
Biparate Loss

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)}),$$

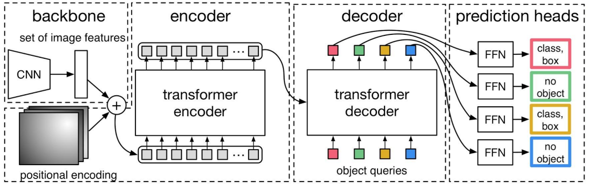
Box Loss

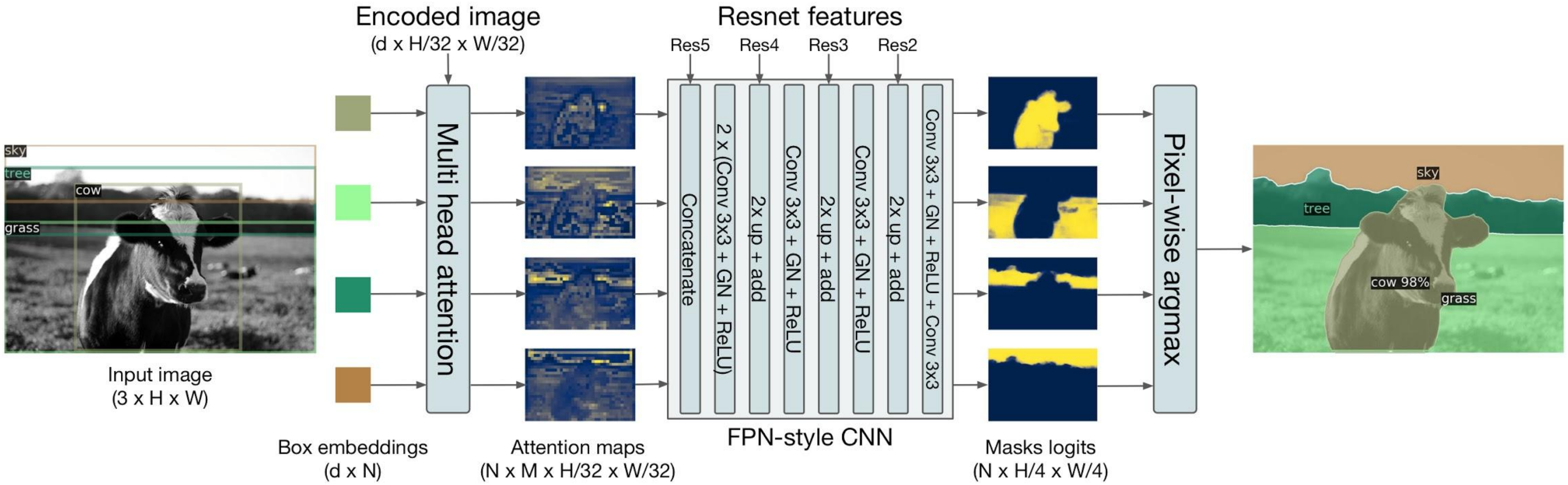
$$\mathcal{L}_{\mathrm{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \boxed{\mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\mathrm{box}}(b_i, \hat{b}_{\hat{\sigma}}(i))} \right] ,$$

Learned Query Embeddings



Panoptic Segmentation 🐄

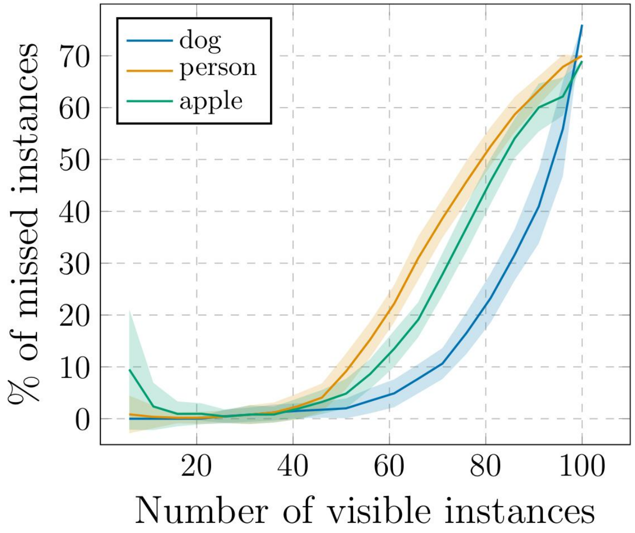




Quantitative Results

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Quantitative Results



DETR Qualitative Results

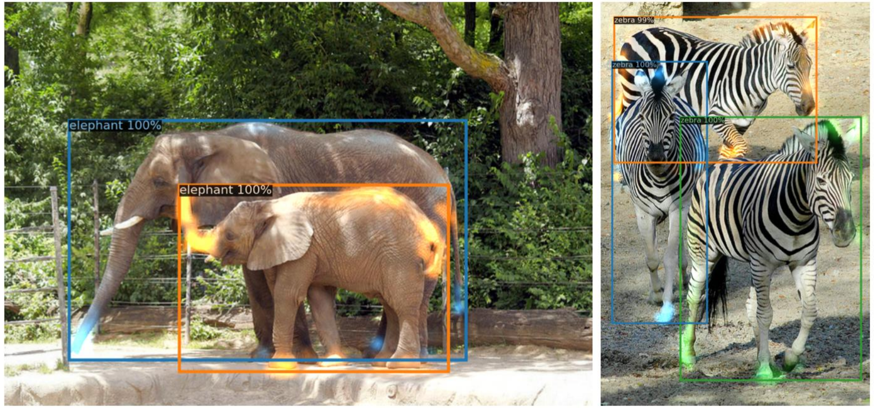
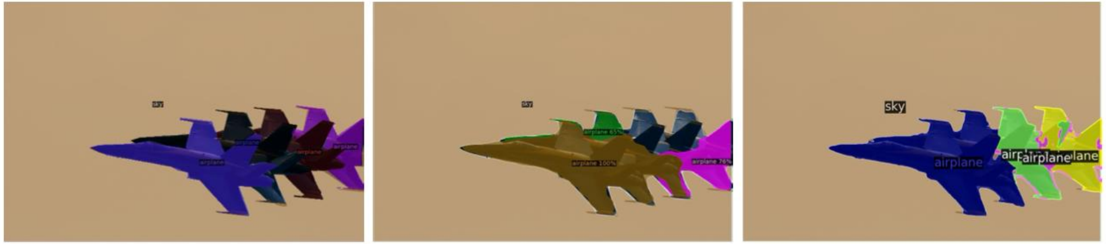
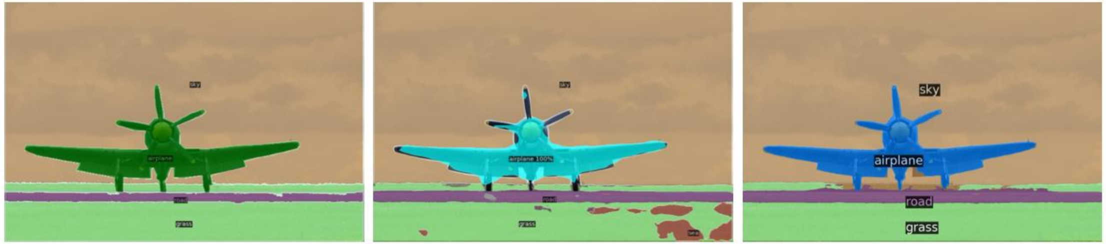


Fig. 6: Visualizing decoder attention for every predicted object (images from COCO `val` set). Predictions are made with DETR-DC5 model. Attention scores are coded with different colors for different objects. Decoder typically attends to object extremities, such as legs and heads. Best viewed in color.

DETR Qualitative Results



(a) Failure case with overlapping objects. PanopticFPN misses one plane entirely, while DETR fails to accurately segment 3 of them.



(b) `Things` masks are predicted at full resolution, which allows sharper boundaries than PanopticFPN

Fig. 11: Comparison of panoptic predictions. From left to right: Ground truth, PanopticFPN with ResNet 101, DETR with ResNet 101

Final Thoughts and Discussion

- Is an attention based model more explainable than a traditional CNN?
- Is the absence of locality with attention based models a bug a feature or both?
- Are we approaching a unified approach for NLP and computer vision tasks?