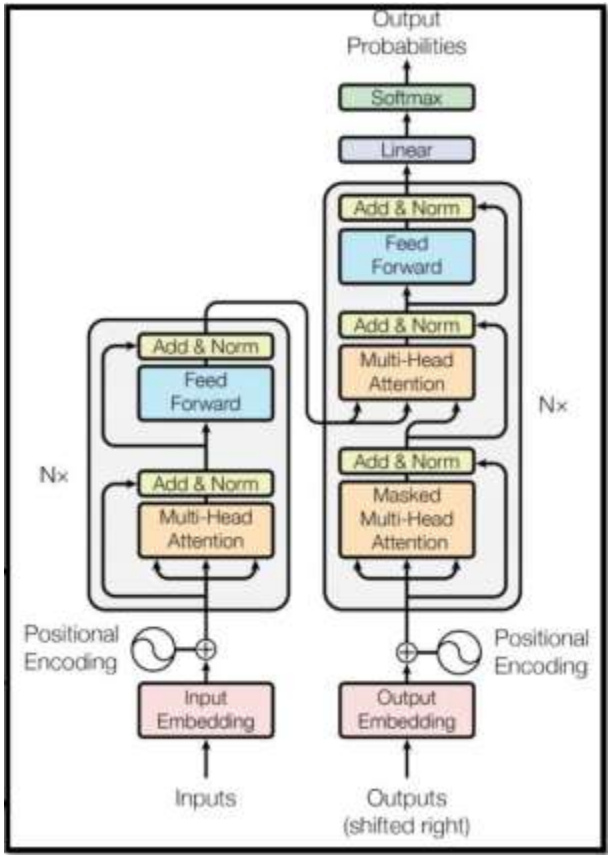
Lecture 09: Ttransformer: Applications

Lan Xu
SIST, ShanghaiTech
Fall, 2023

Transformer

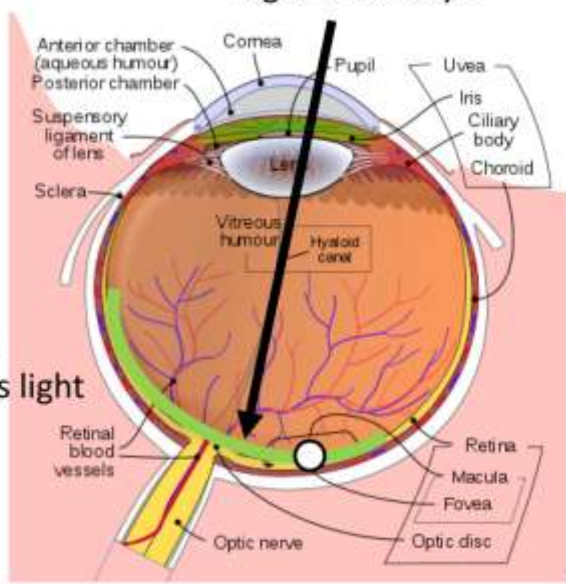
- A new block type in term of encoder-decoder
- Attention only !



Attention Mechanism

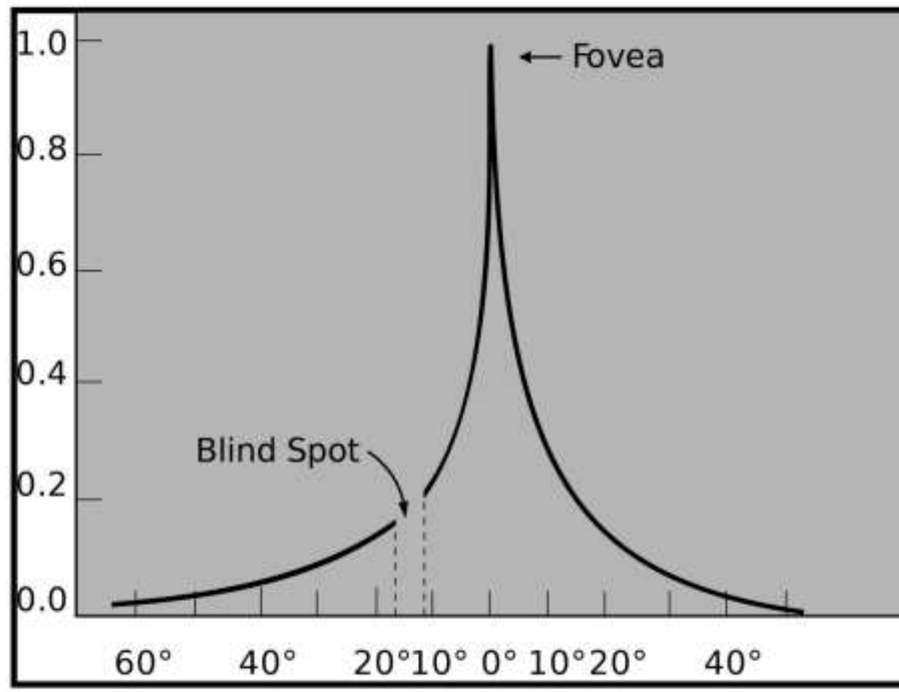
- Human Vision: Fovea

Light enters eye



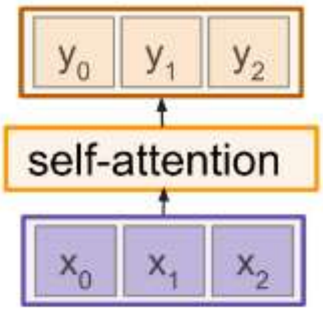
Retina
detects light

The **fovea** is a tiny region of the retina that can see with high acuity



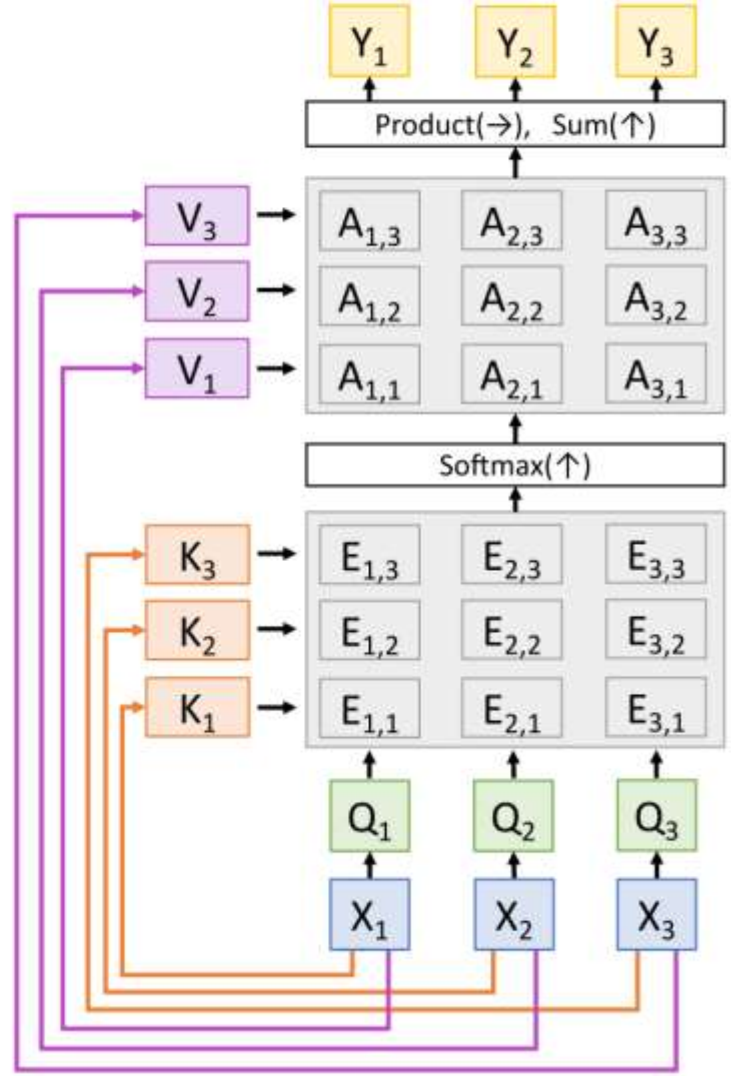
Self-attention Layer Summary

- One query per input vector



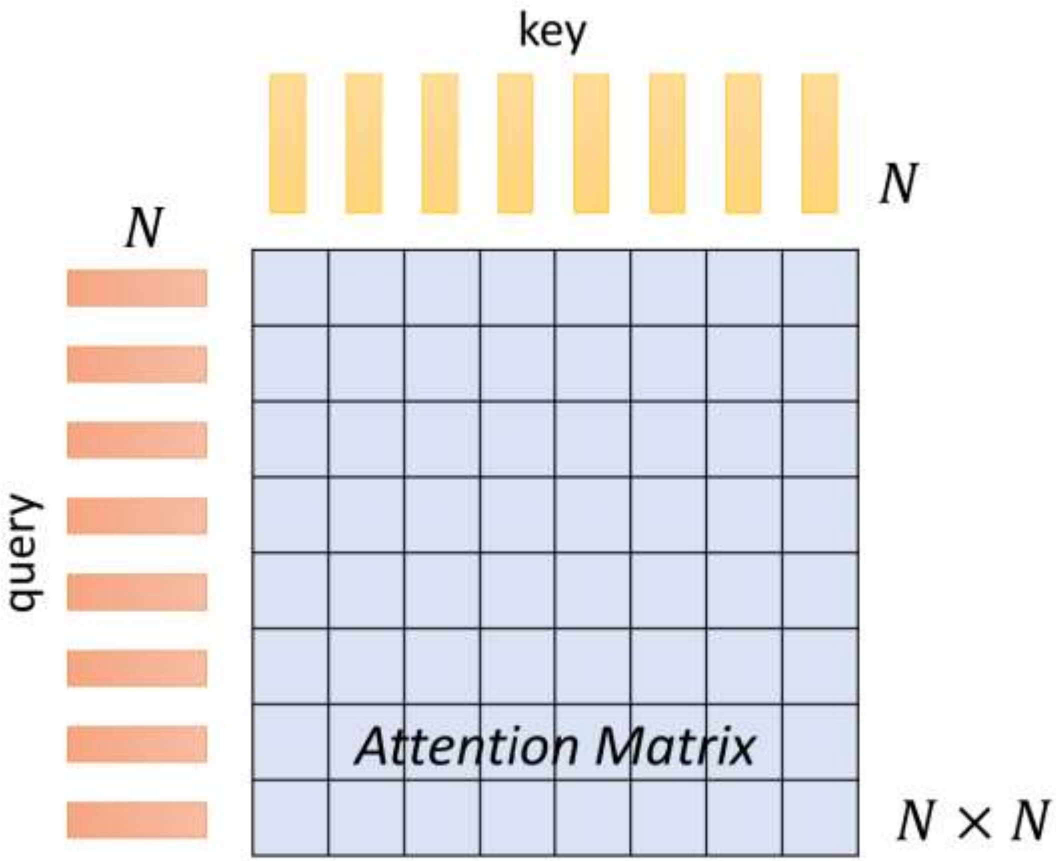
Inputs:
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)
Query matrix: W_Q (Shape: D_X x D_Q)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)
Output vectors: $Y = AV$ (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$



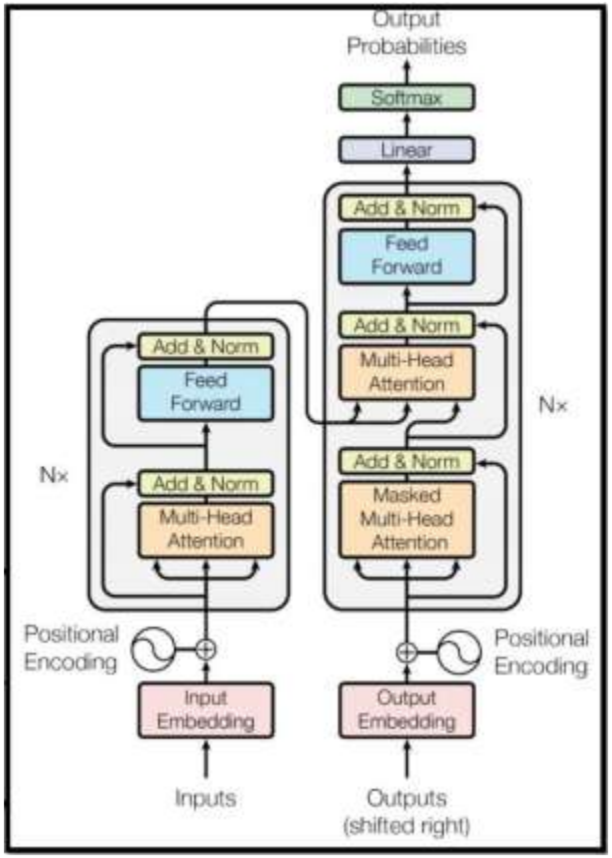
How to make self-attention efficient

- Sequence Length = N



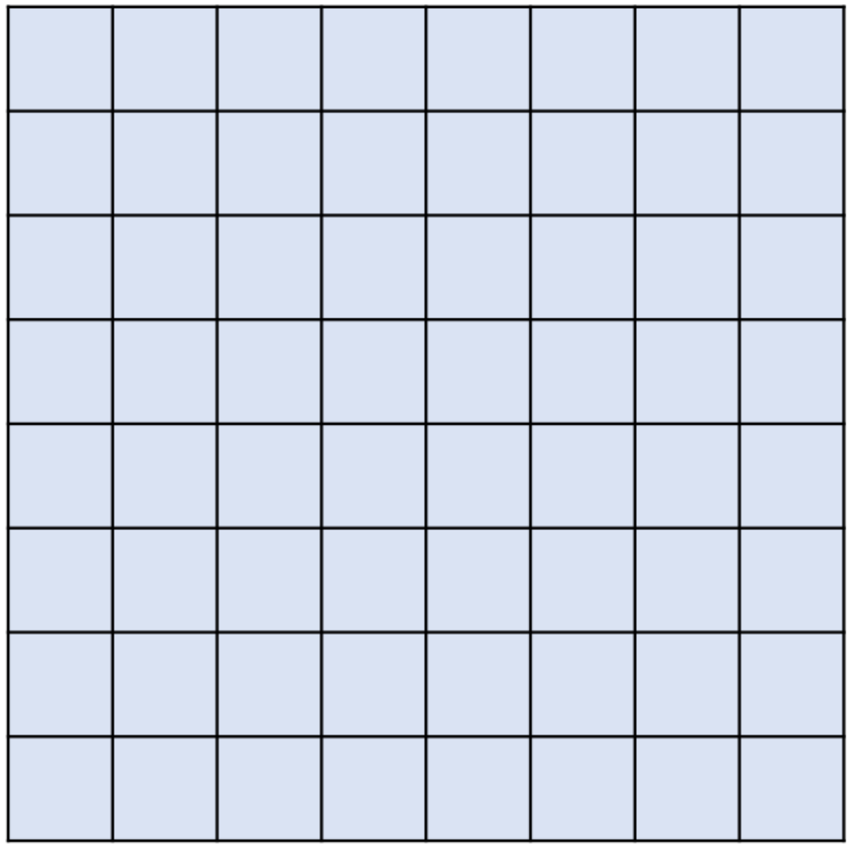
How to make self-attention efficient

- Self-attention is only a module in a larger network.
- Self-attention dominates computation when N is large.



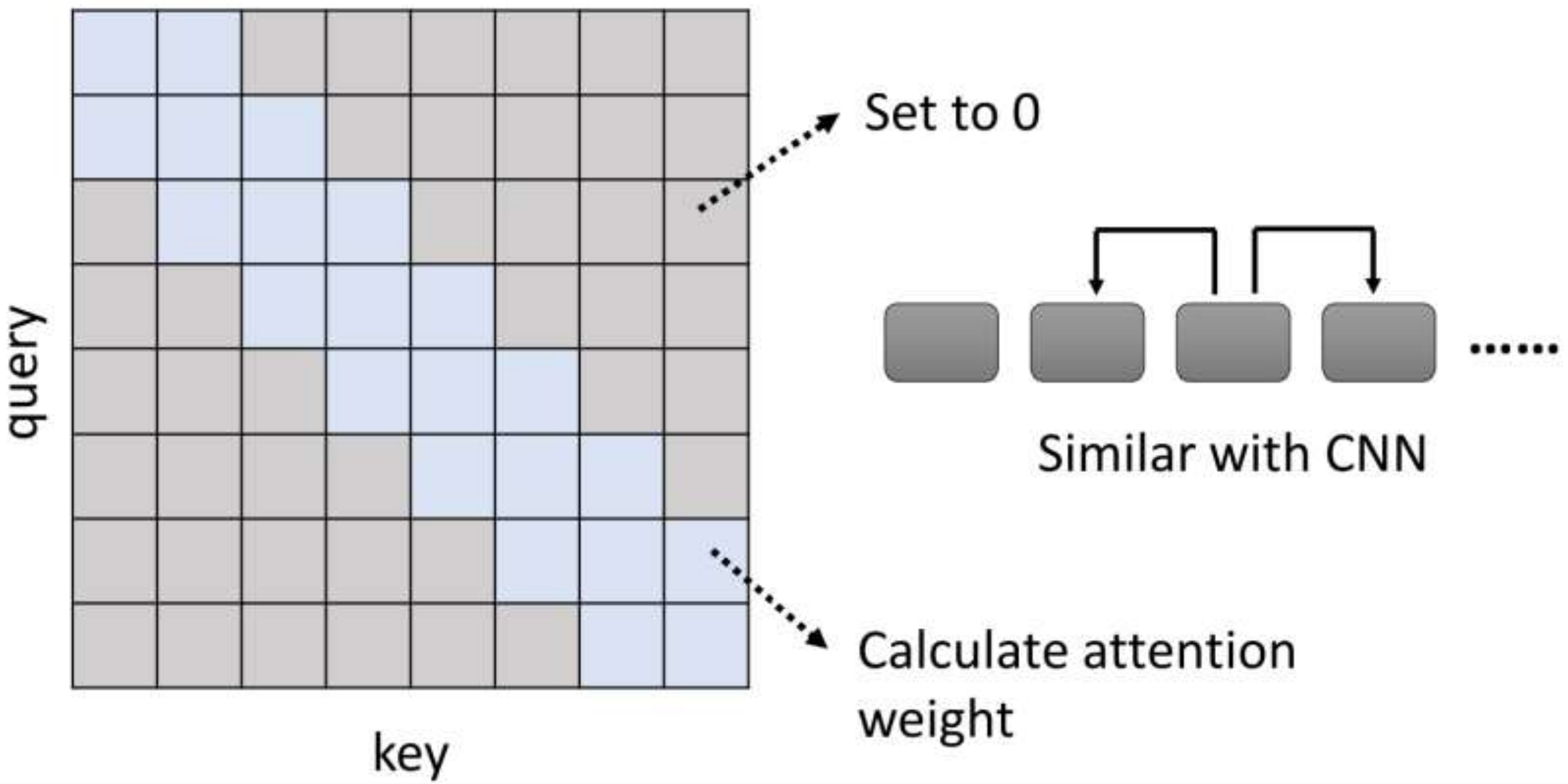
Skip Calculations with Human Knowledge

- Can we fill in some values with human knowledge?



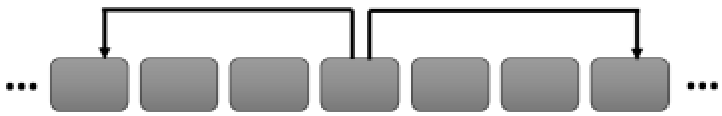
Local Attention/Truncated Attention

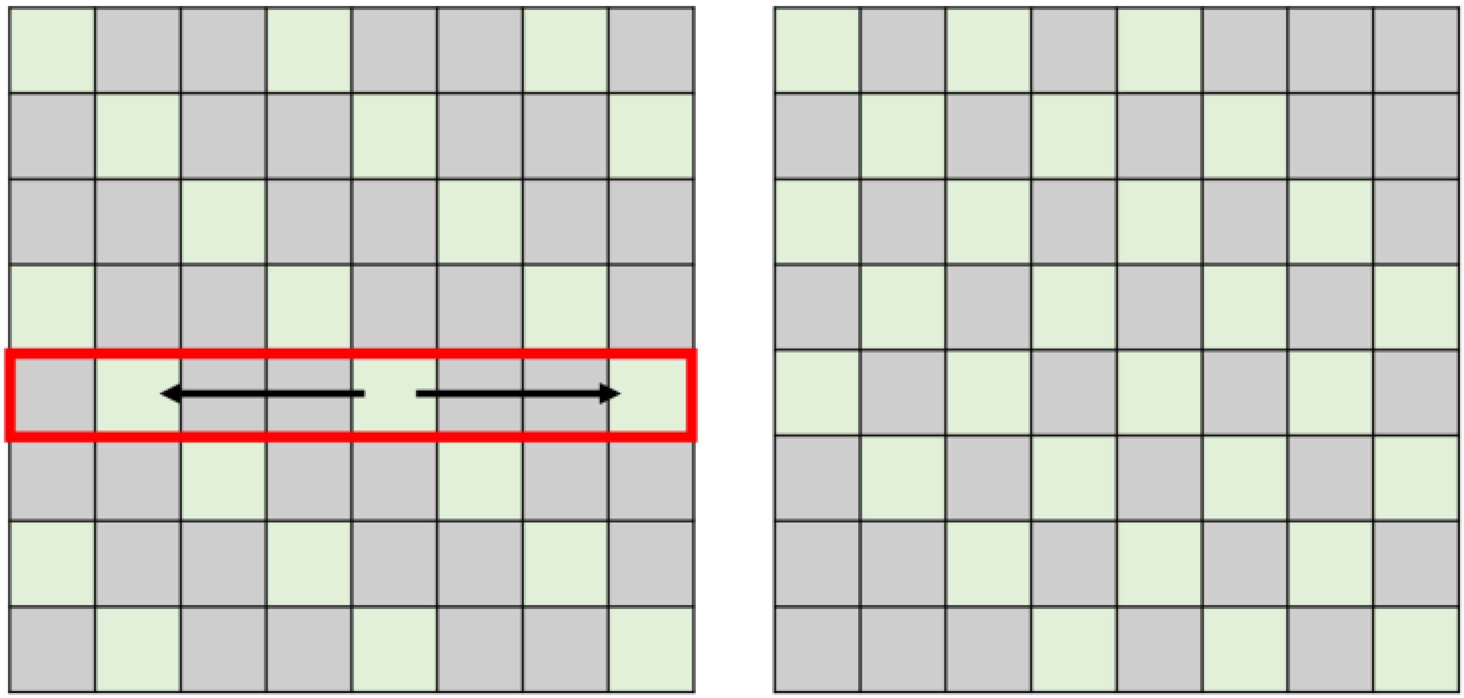
- Make it local
- Similar to CNN, may sacrifice performance



Stride Attention

- Observe non-local regions

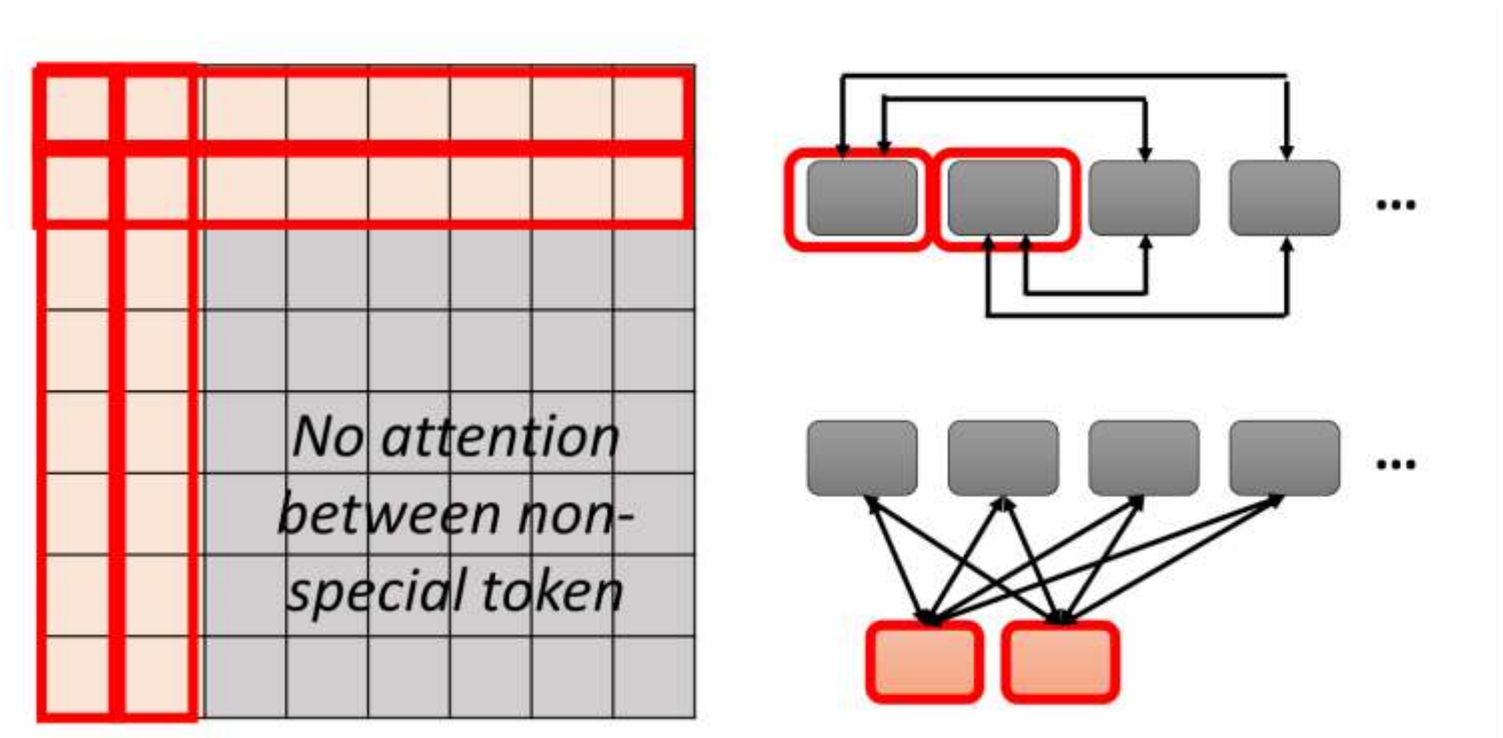




Global Attention

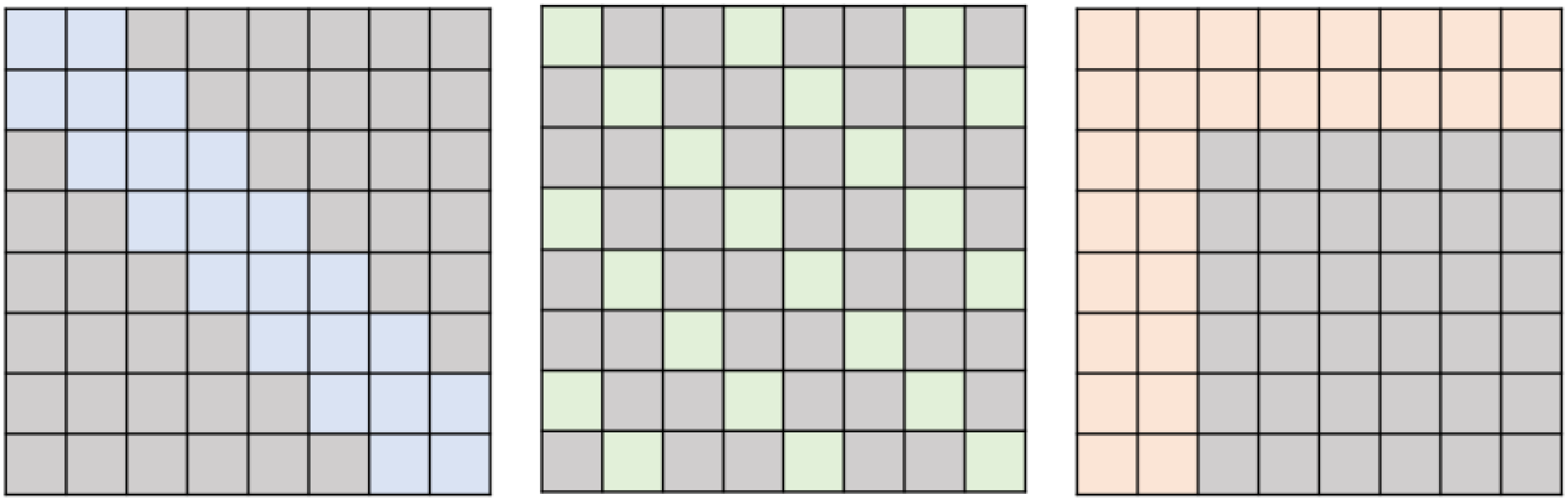
- Add special token into original sequence

- Attend **to** every token → collect global information
- Attend **by** every token → it knows global information



Many Different Choices

- Multi-head attention

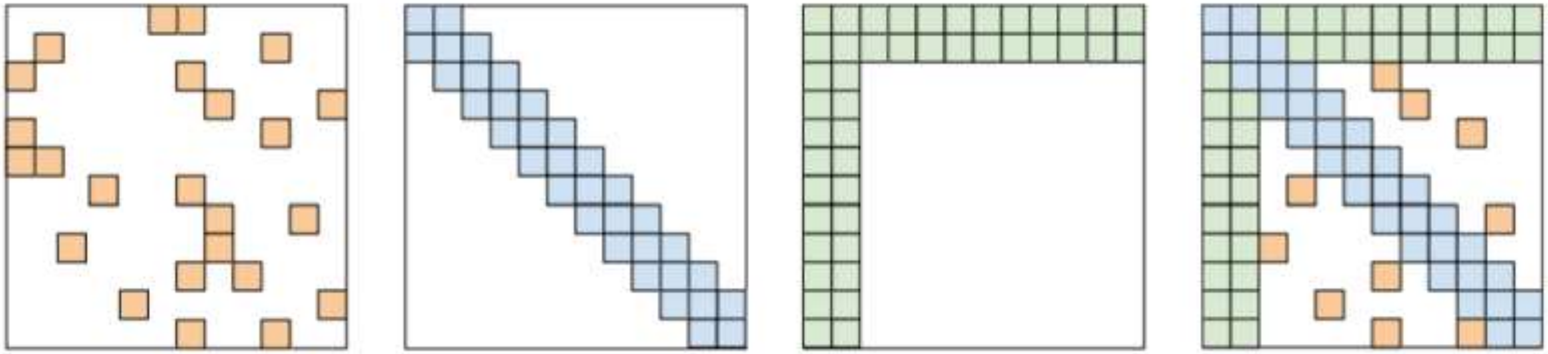


Many Different Choices

- Longformer https://arxiv.org/abs/2004.05150

(b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

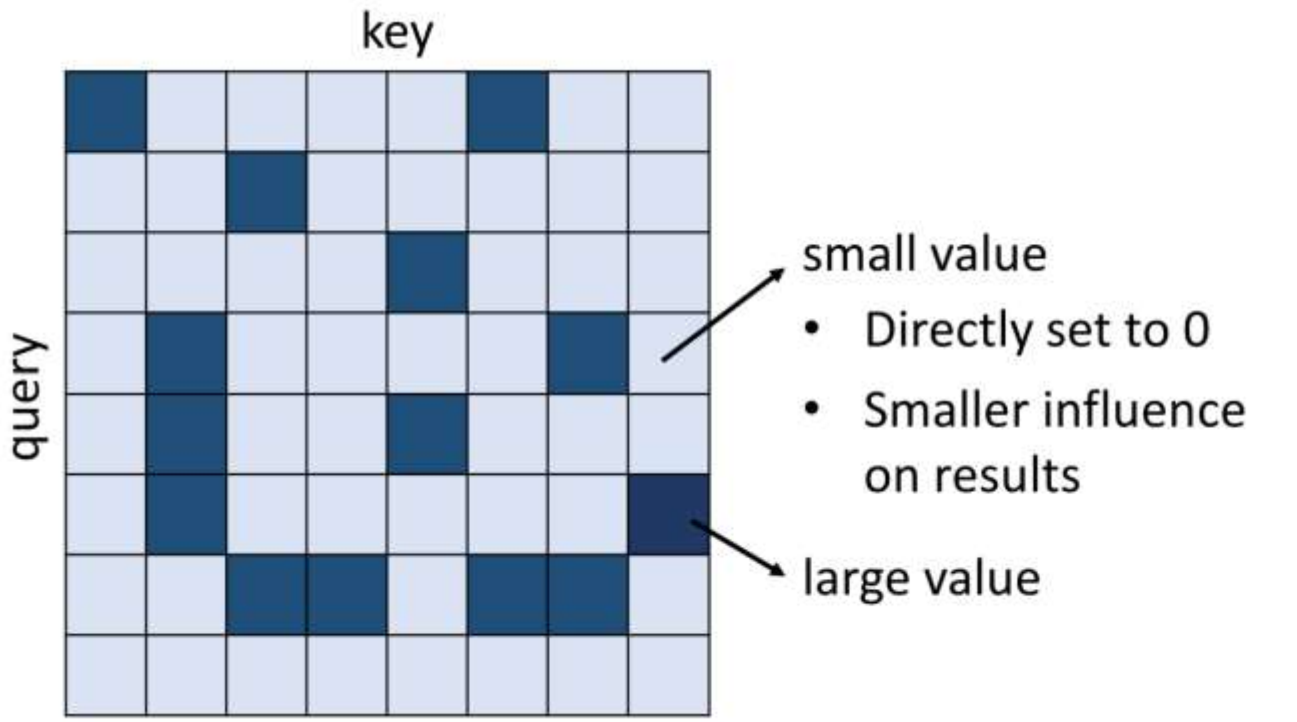
- Big Bird https://arxiv.org/abs/2007.14062



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

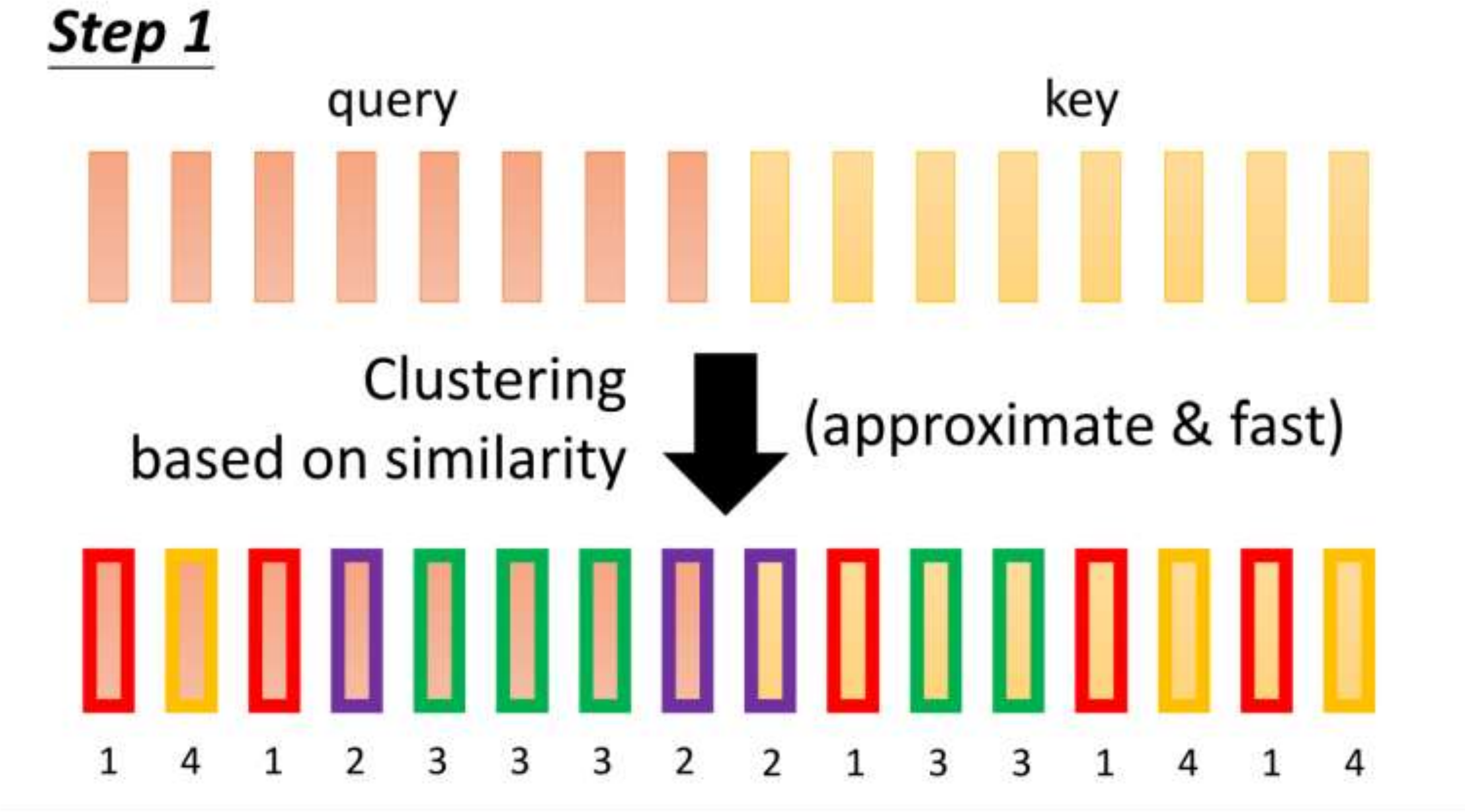
Focus on Critical Parts

- Truncate small value to 0
- Need to quickly estimate the portions with small attention weights



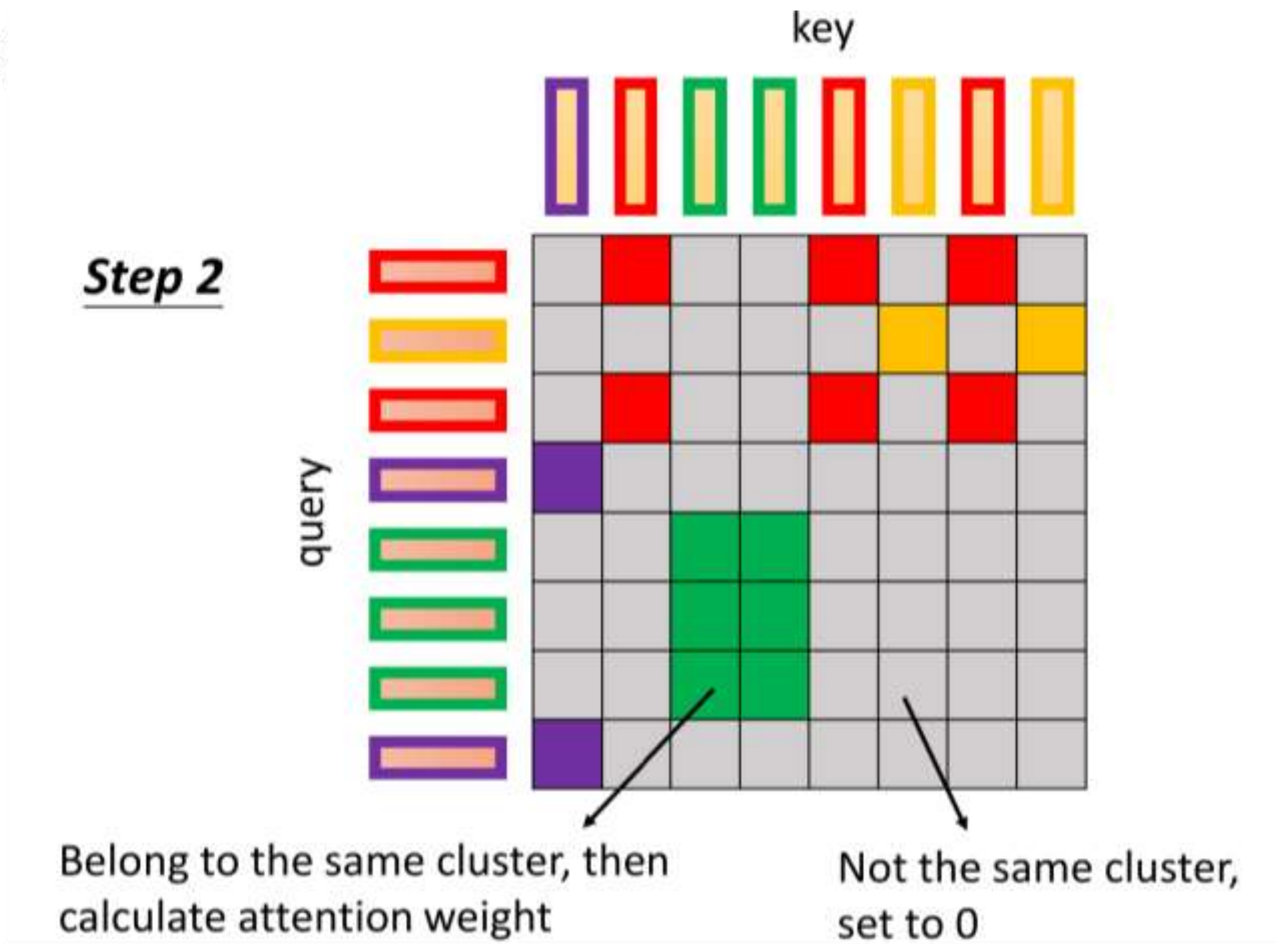
Clustering of Attention Portions

- Reformer: https://openreview.net/forum?id=rkgNKkHtvB
- Routing Transformer: https://arxiv.org/abs/2003.05997



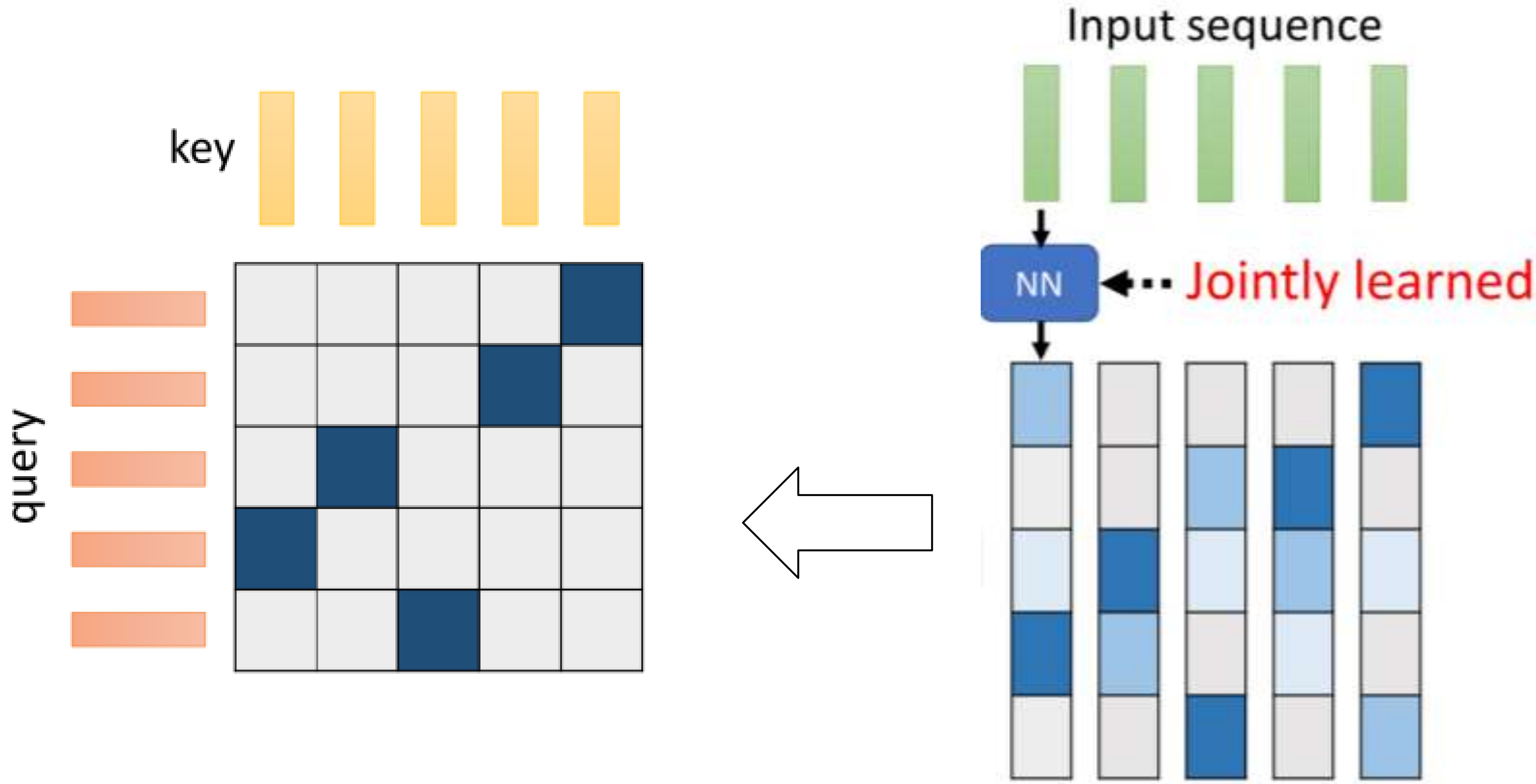
Clustering of Attention Portions

- Reformer: https://openreview.net/forum?id=rkgNKkHtvB
- Routing Transformer: https://arxiv.org/abs/2003.05997



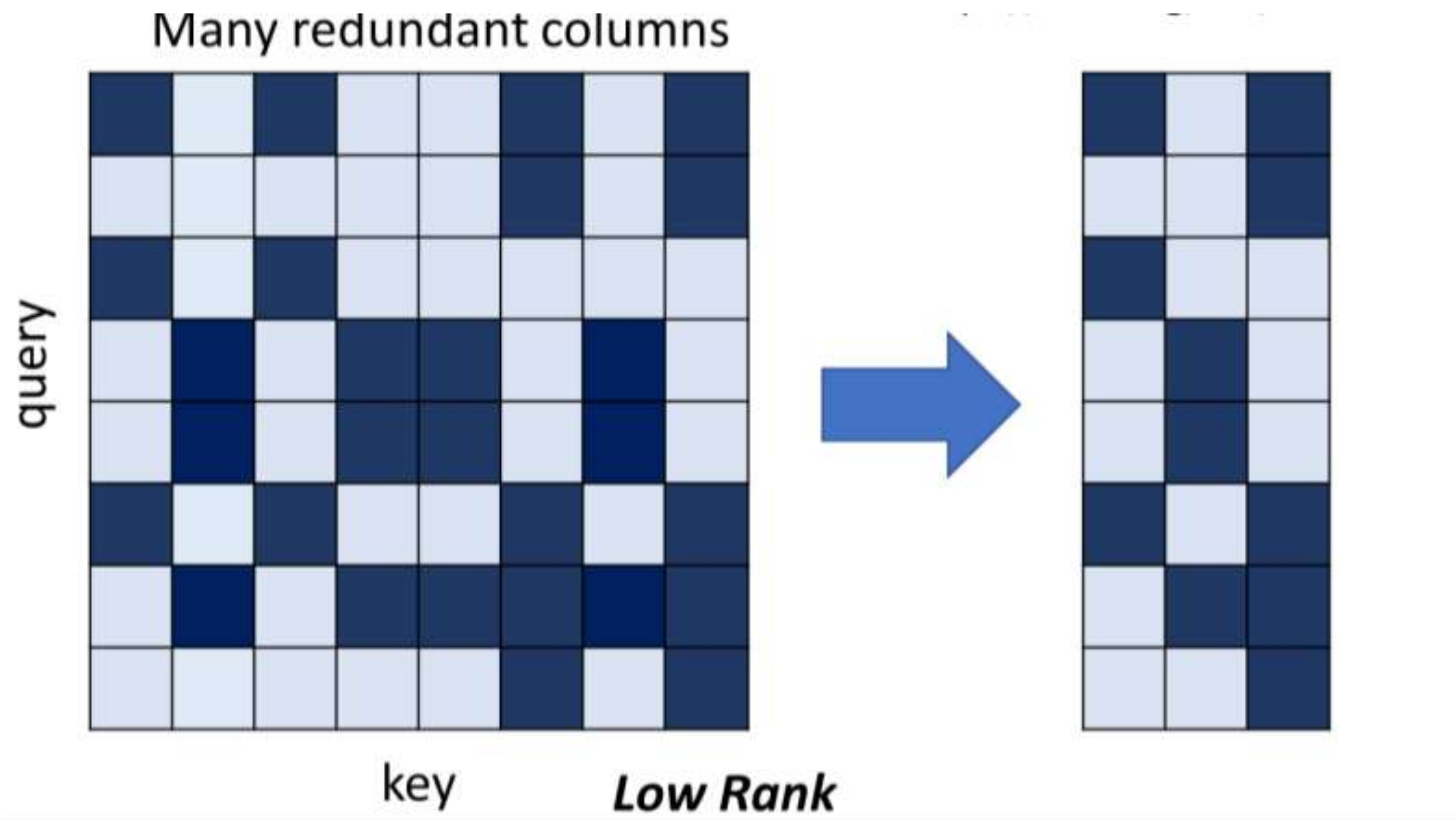
Learnable Patterns

- A grid should be skipped or not is decided by another learned module
- Sinkhorn Sorting Network: https://arxiv.org/abs/2002.11296



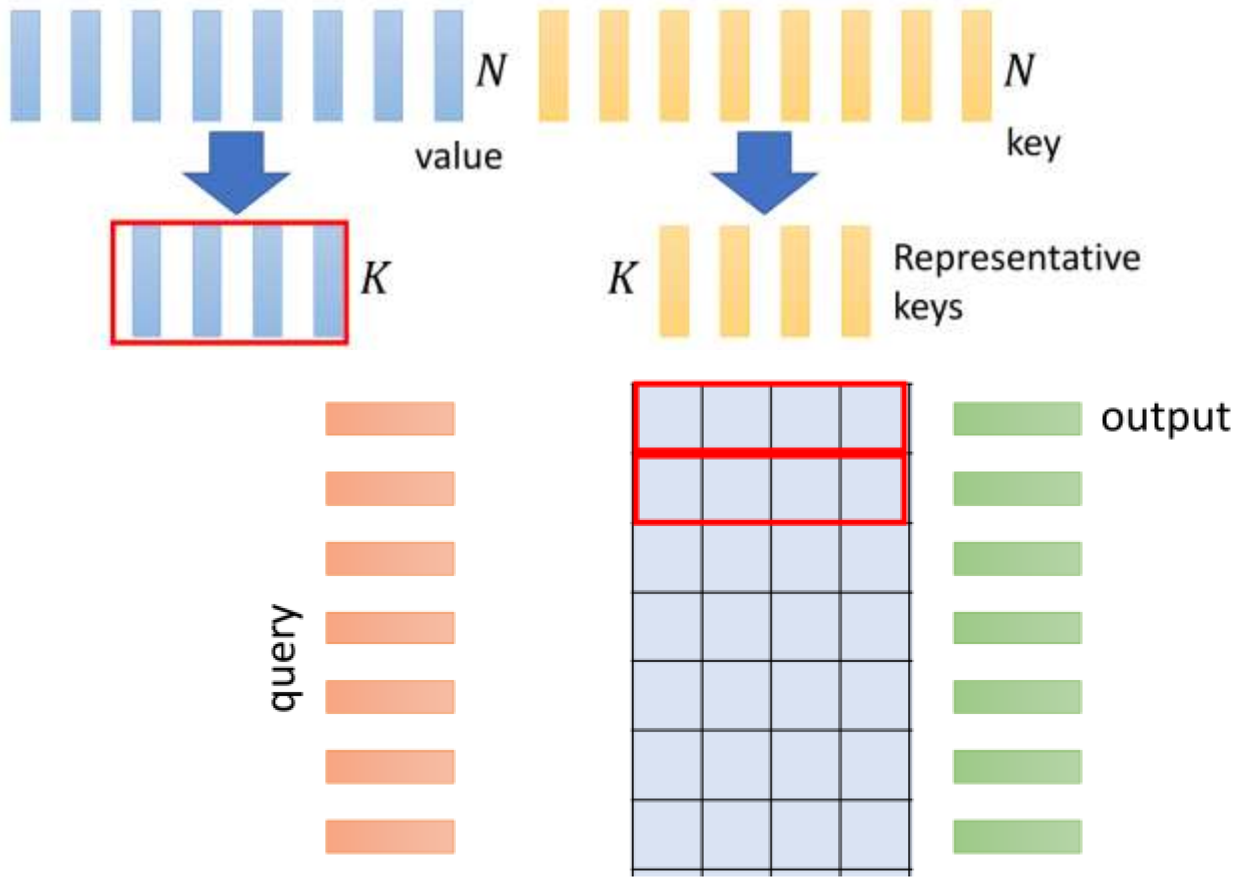
Low Rank property of attention matrix

- Linformer: https://arxiv.org/abs/2006.04768



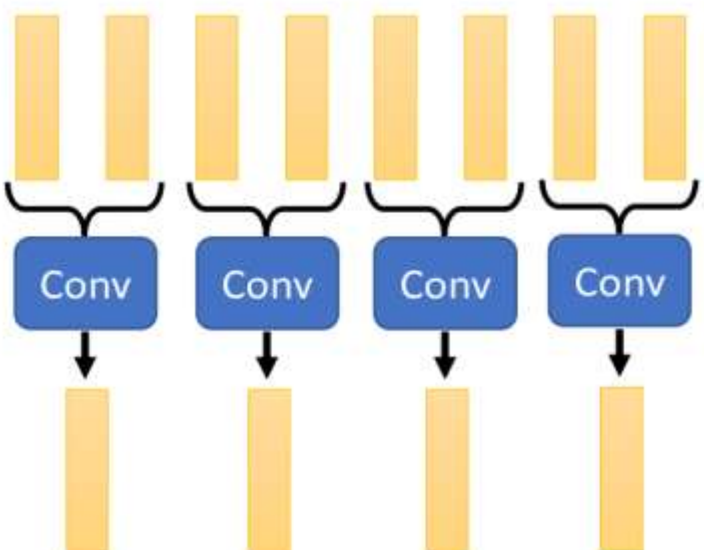
Low Rank property of attention matrix

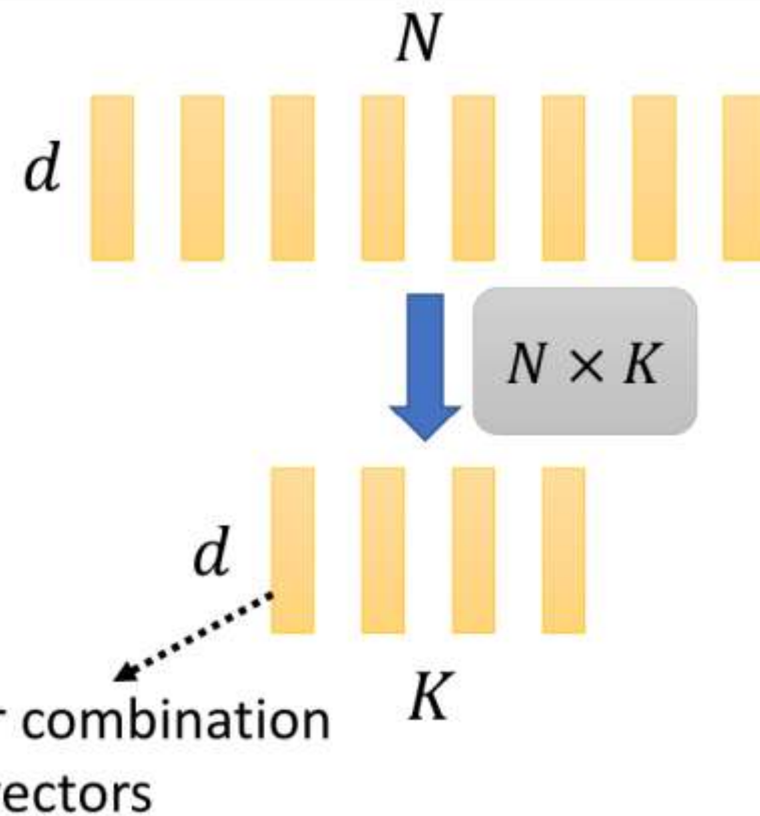
- Can even reduce the number of queries → change output sequence length!



Reduce Number of Keys

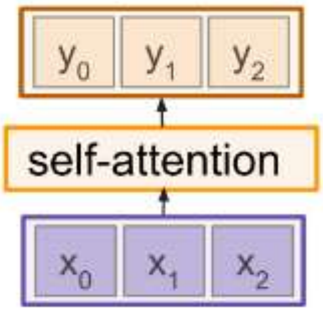
- Compressed Attention: https://arxiv.org/abs/1801.10198
- Linformer: https://arxiv.org/abs/2006.04768

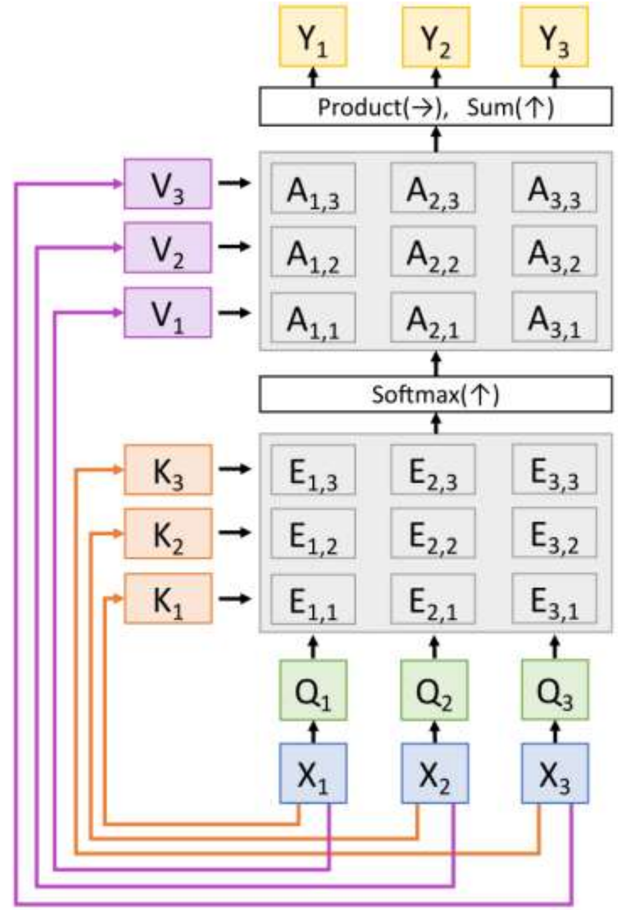




Reduce the matrix calculation

- Attention Mechanism is three-matrix Multiplication



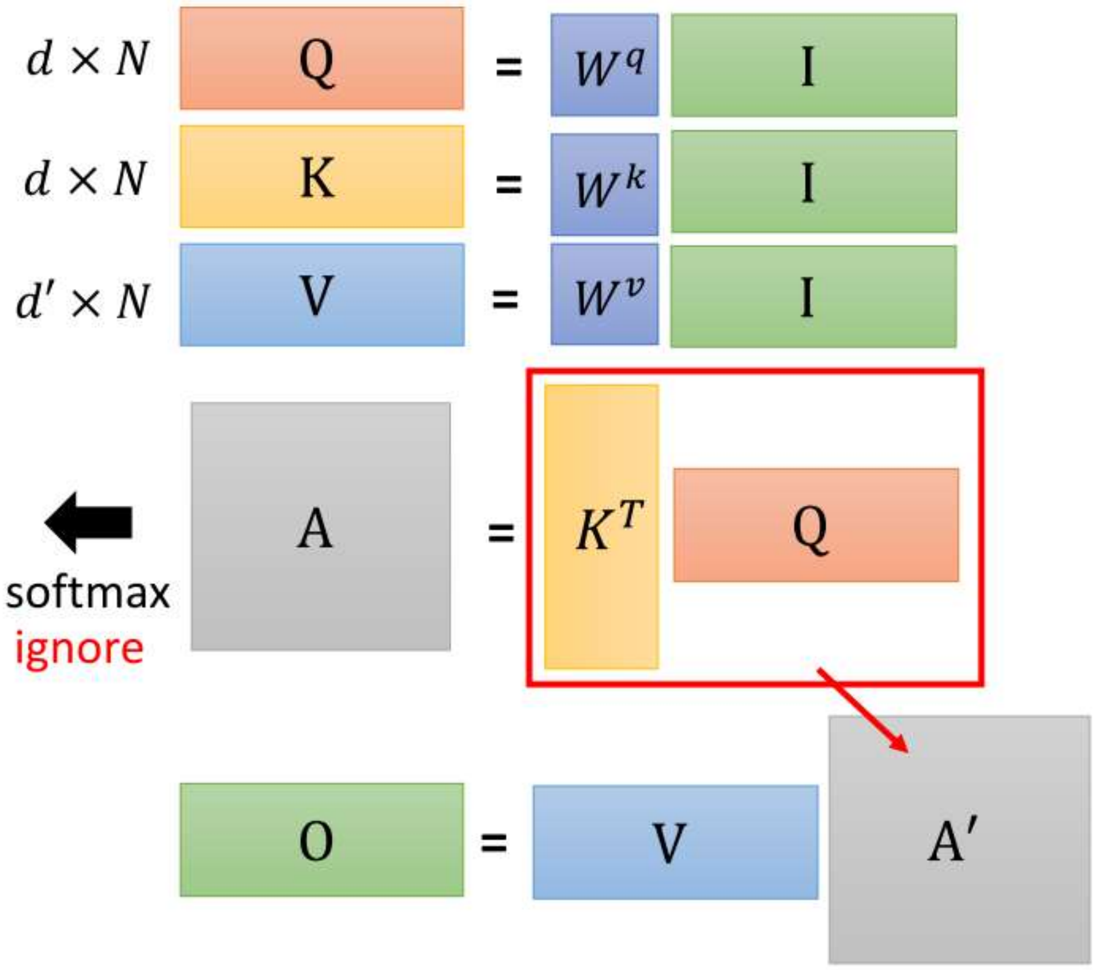


Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \mathrm{sqrt}(D_Q)$
Attention weights: $A = \mathrm{softmax}(E, \mathrm{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

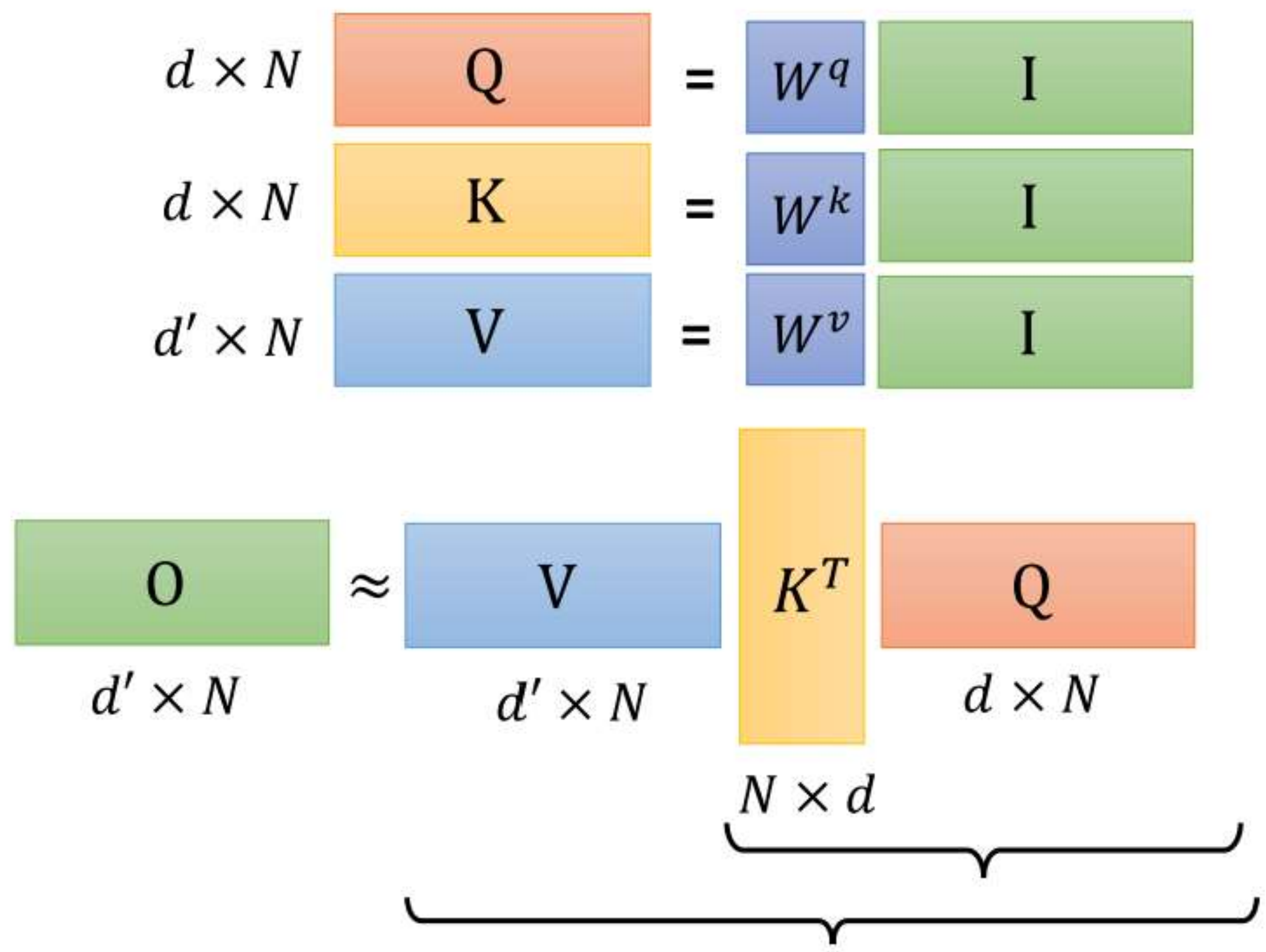
Reduce the matrix calculation

- Attention Mechanism is three-matrix Multiplication



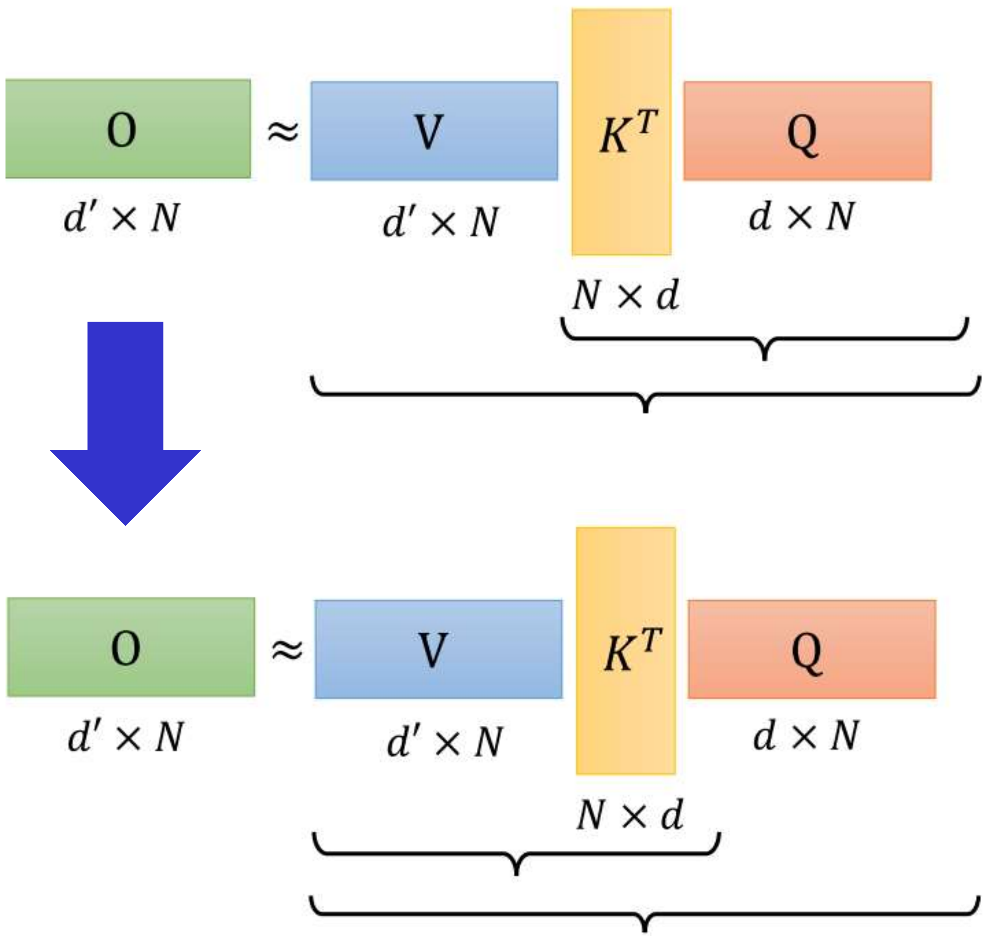
Reduce the matrix calculation

- Attention Mechanism is three-matrix Multiplication



Reduce the matrix calculation

- Attention Mechanism is three-matrix Multiplication



$$\boxed{(d + d')N^2}$$

$N \times d \times N$

$d' \times N \times N$

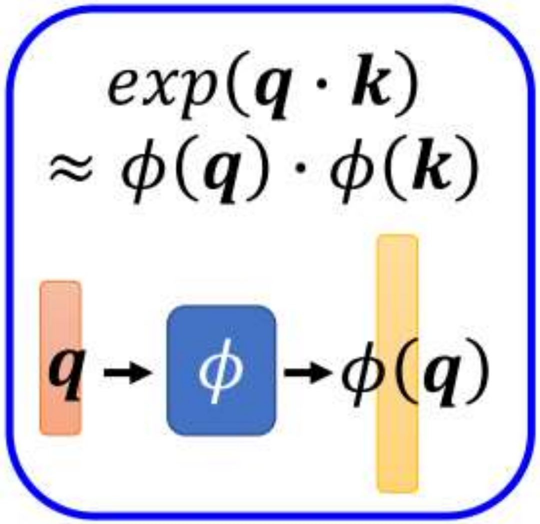
$$\boxed{2d'dN}$$

$d' \times N \times d$

$d' \times d \times N$

Reduce the matrix calculation

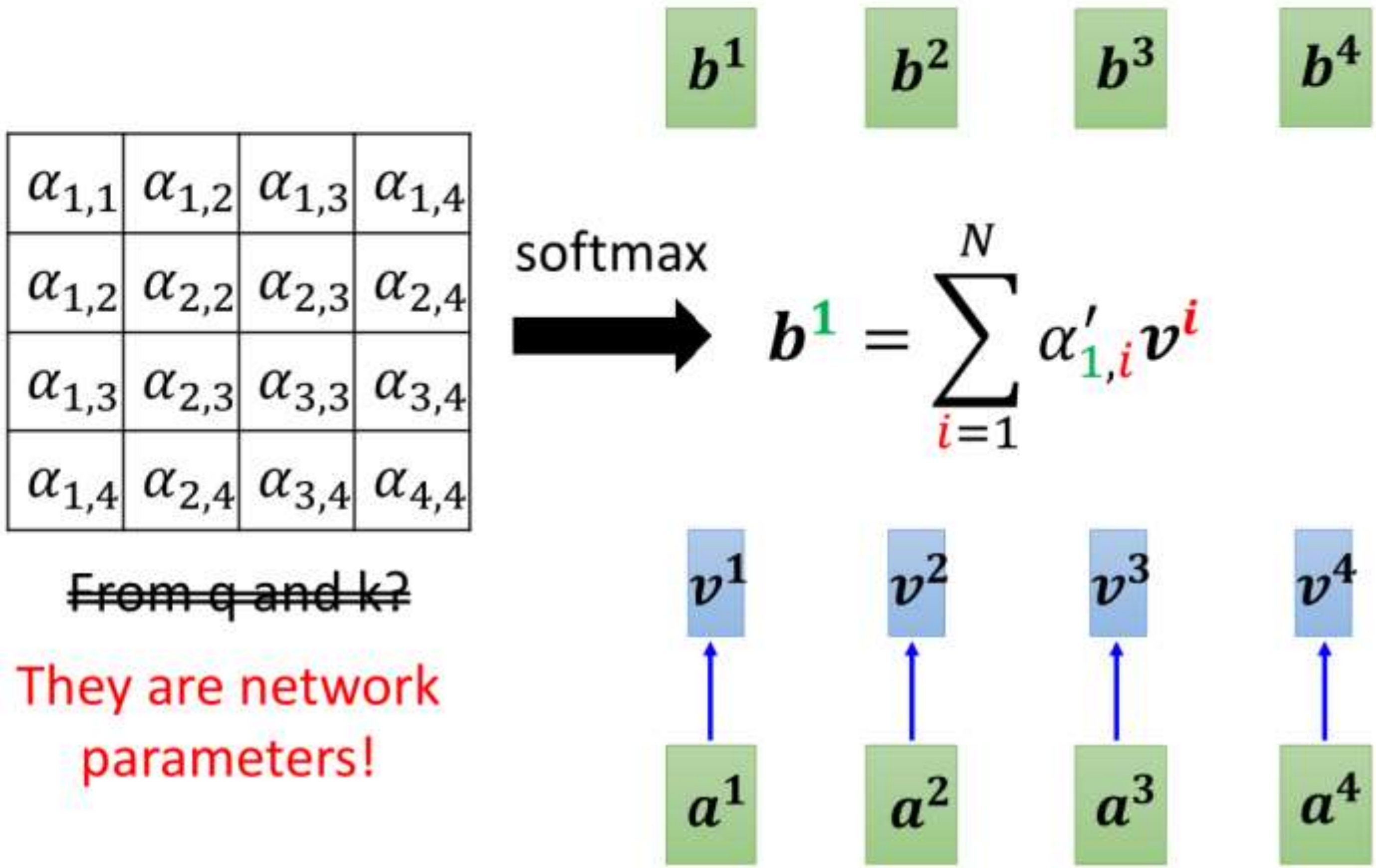
- If put softmax back, more complicated ……
- Linear decomposition



- Efficient attention
  https://arxiv.org/pdf/1812.01243.pdf
- Linear Transformer
  https://linear-transformers.com/
- Random Feature Attention
  https://arxiv.org/pdf/2103.02143.pdf
- Performer
  https://arxiv.org/pdf/2009.14794.pdf

No Q/K to compute attention

- Synthesizer: https://arxiv.org/abs/2005.00743



Attention-free ?

- Fnet: Mixing tokens with fourier transforms

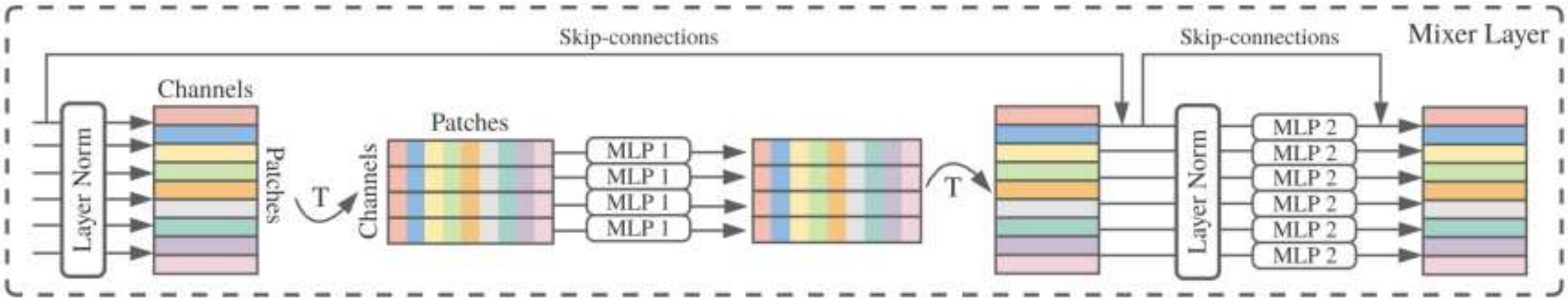
https://arxiv.org/abs/2105.03824

- Pay Attention to MLPs

https://arxiv.org/abs/2105.08050

- MLP-Mixer: An all-MLP Architecture for Vision

https://arxiv.org/abs/2105.01601



Attention Summary

- Human Knowledge

- Local Attention, Big Bird

- Clustering

- Reformer

- Learnable Patterns

- Sinkforn

- Representative Key

- Linformer

- Linear-calculation

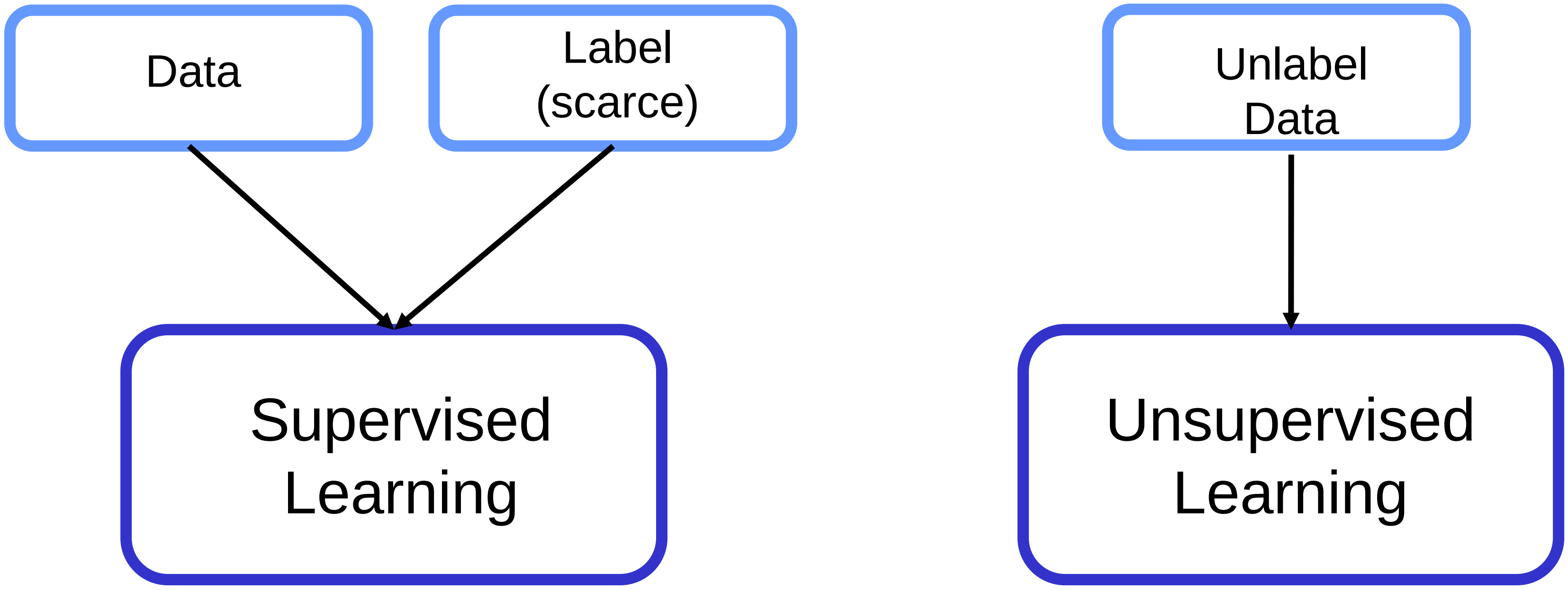
- Linear Transformer, Performer

- New framework

- Synthesizer……

Transformer in NLP

- From the aspect of Unsupervised Learning
- Supervised learning works great and comes with guarantees! But large labeled datasets are hard to find.
- Also learn from unlabeled data?
- Big trove of unlabeled data online!



Transformer in NLP

- Use Autoregressive Generative Models for unsupervised learning!

- “What I cannot **create**, I do not **understand**.”

  ---- Richard Feynman

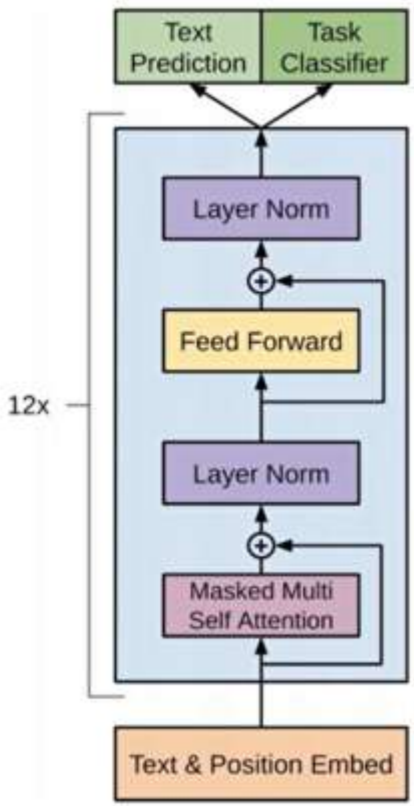
- “What I can **create**, I can also **understand**.”

  ---- Analysis by Synthesis

- Doing very well at next-token prediction requires more than modeling local correlations → perhaps “reasoning”!

Transformer in NLP

- GPT-1 (Radford et al 2018)



DATASET	TASK	SOTA	OURS
SNLI	Textual Entailment	89.3	**89.9**
MNLI Matched	Textual Entailment	80.6	**82.1**
MNLI Mismatched	Textual Entailment	80.1	**81.4**
SciTail	Textual Entailment	83.3	**88.3**
QNLI	Textual Entailment	82.3	**88.1**
RTE	Textual Entailment	**61.7**	56.0
STS-B	Semantic Similarity	81.0	**82.0**
QQP	Semantic Similarity	66.1	**70.3**
MRPC	Semantic Similarity	**86.0**	82.3
RACE	Reading Comprehension	53.3	**59.0**
ROCStories	Commonsense Reasoning	77.6	**86.5**
COPA	Commonsense Reasoning	71.2	**78.6**
SST-2	Sentiment Analysis	**93.2**	91.3
CoLA	Linguistic Acceptability	35.0	**45.4**
GLUE	Multi Task Benchmark	68.9	**72.8**

Transformer in NLP

- GPT-2: Zero-Shot Reading Comprehension

> The 2008 Summer Olympics torch relay was run from March 24 until August 8, 2008.
>
> ...
>
> The relay also included an ascent with the flame to the top of Mount Everest on the border of Nepal and Tibet, China from the Chinese side, which was closed specially for the event.

- Q: And did they climb any mountains?
- A: Everest

Transformer in NLP

- GPT-2: Zero-Shot Summarization

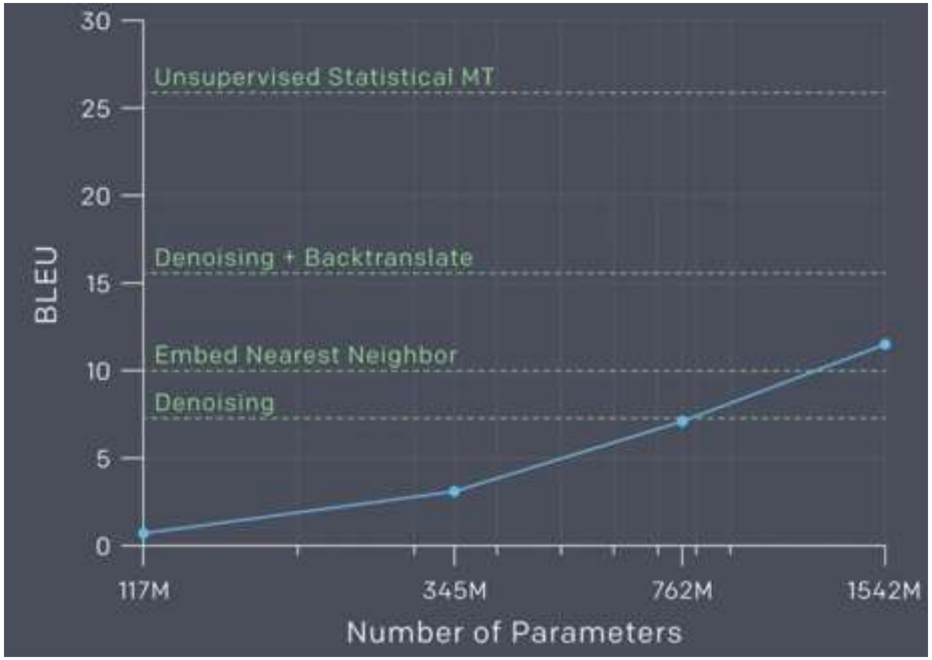
Prehistoric man sketched an incredible array of prehistoric beasts on the rough limestone walls of a cave in modern day France 36,000 years ago...

TLDR: **The original site in Vallon-Pont-D'arc in Southern France is a Unesco World Heritage site and is the oldest known and the best preserved cave decorated by man. The replica cave was built a few miles from the original site in Vallon-Pont-D'Arc in Southern France. The cave contains images of 14 different species of animals including woolly rhinoceros, mammoths, and big cats.**

Transformer in NLP

- GPT-2: Zero-Shot Translation

The sentence *"Un homme a expliqué que l'opération gratuite qu'il avait subie pour soigner une hernie lui permettrait de travailler à nouveau."* translated from French to English, means:
A man told me that the operation gratuity he had been promised would not allow him to travel.



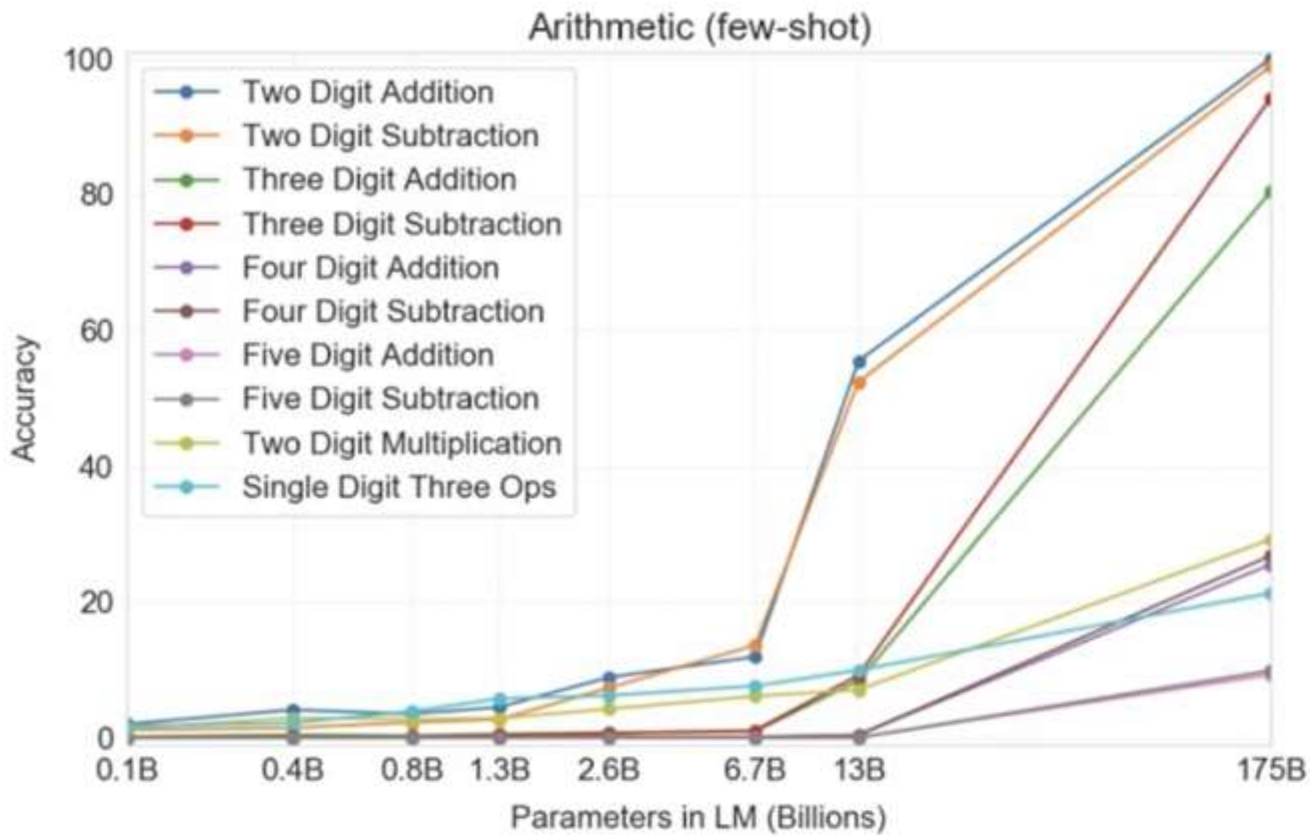
Transformer in NLP

- GPT-3: Language Model Metalearning

Transformer in NLP

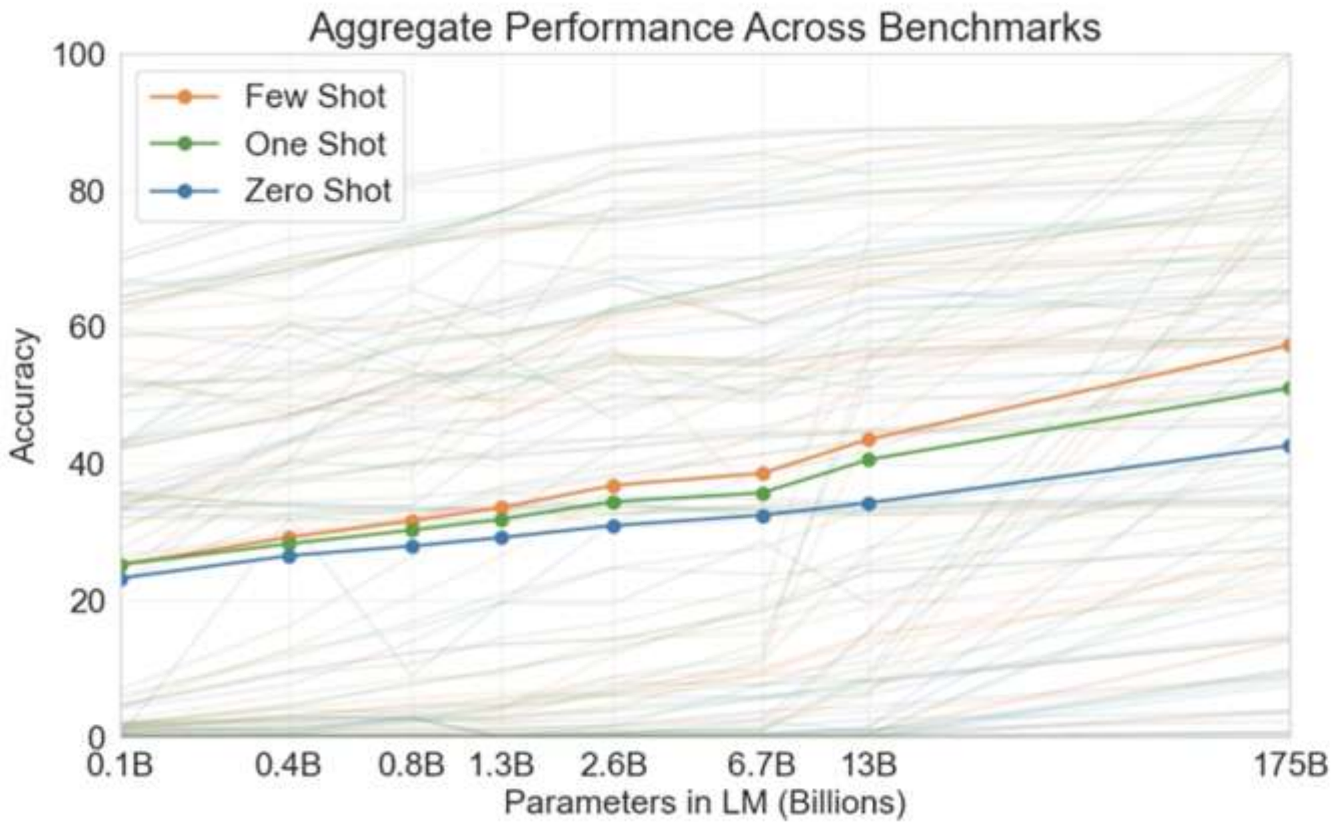
- GPT-3: Few Shot Arithmetic

12+13 = 25. 34+11 = 44. 64 + 30 = 94. 31+41 = **72**.



Transformer in NLP

- GPT-3: General Few Shot Learning
- Autoregressive Language Modeling is universal!



Transformer United

- iGPT (Chen et al 2020): Can we apply GPT to images?

GPT

Next **word** prediction

Pretrain on **Internet**, finetune on **Question Answering, ...**

Zero shot, Fine-tuning

Images

Next **pixel** prediction

Pretrain on **unlabeled ImageNet**, finetune on **CIFAR Classification**

Linear probes, Fine-tuning

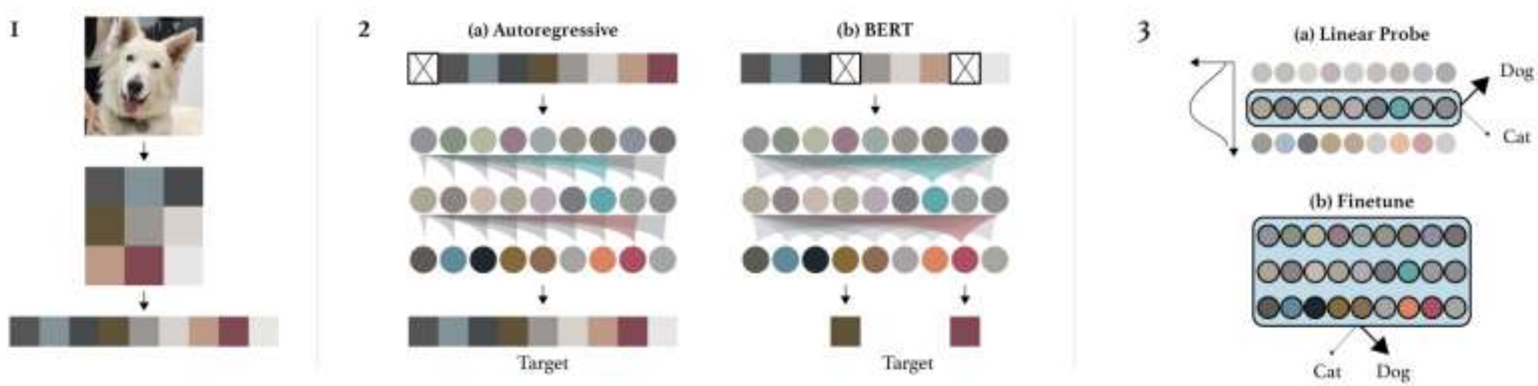


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

Transformer United

- iGPT: Completion

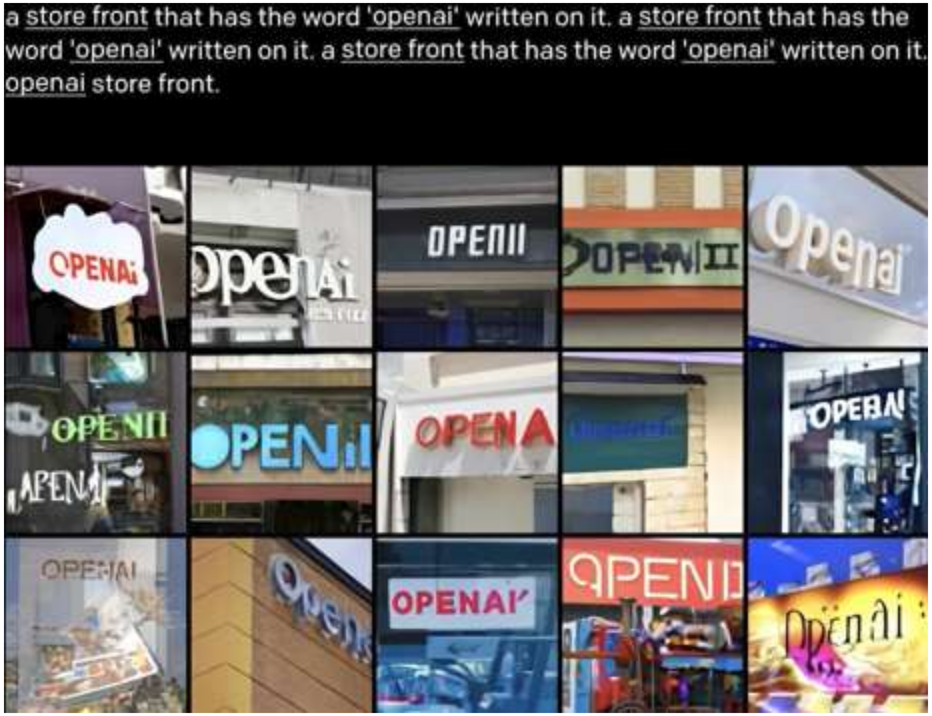
Transformer United

- iGPT: Feature Learning

EVALUATION	MODEL	ACCURACY	PRE-TRAINED ON IMAGENET W/O LABELS	PRE-TRAINED ON IMAGENET W/ LABELS
CIFAR-10 Linear Probe	ResNet-152[50]	94.0		✓
	SimCLR[12]	95.3	✓	
	iGPT-L 32x32	**96.3**	✓	
CIFAR-100 Linear Probe	ResNet-152	78.0		✓
	SimCLR	80.2	✓	
	iGPT-L 32x32	**82.8**	✓	
STL-10 Linear Probe	AMDIM-L[13]	94.2	✓	
	iGPT-L 32x32	**95.5**	✓	
CIFAR-10 Fine-tune	AutoAugment[51]	98.5		
	SimCLR	98.6	✓	
	GPipe[15]	**99.0**		✓
	iGPT-L	**99.0**	✓	
CIFAR-100 Fine-tune	iGPT-L	88.5	✓	
	SimCLR	89.0	✓	
	AutoAugment	89.3		
	EfficientNet[52]	**91.7**		✓

Transformer United

- DALL-E (Ramesh et al 2020): GPT for Text-to-Image
- Simply train a transformer on concat (caption, image)!



Transformer United

- DALL-E: Zero-Shot Image to Image

(a) "the exact same cat on the top as a sketch on the bottom"

(b) "the exact same photo on the top reflected upside-down on the bottom"

(c) "2 panel image of the exact same cat. on the top, a photo of the cat. on the bottom, an extreme close-up view of the cat in the photo."

Transformer United

- DALL-E: Zero-Shot Image to Image

(d) "the exact same cat on the top colored red on the bottom"

(e) "2 panel image of the exact same cat. on the top, a photo of the cat. on the bottom, the cat with sunglasses."

(f) "the exact same cat on the top as a postage stamp on the bottom"

Transformer United

- CodeX: Isn’t Code JUST another modality?

- Why is it worth the effort to train a model on code?

- GPT-3 had a rudimentary ability to write Python code from a docstring or method name, even though there was little code in the training data.
- Functions can be tested with unit tests and an interpreter

Transformer United

- CodeX: The HumanEval Dataset

```
def solution(lst):
   """Given a non-empty list of integers, return the sum of all
of the odd elements that are in even positions.

   Examples
   solution([5, 8, 7, 1]) ==> 12
   solution([3, 3, 3, 3, 3]) ==> 9
   solution([30, 13, 24, 321]) ==>0
   """
```

```
   return sum([x for idx, x in enumerate(lst) if idx%2==0 and
x%2==1])
```

Transformer United

- CodeX: The Pass@K Metric

- Definition: Average probability (over all problems) that at least one of K samples passes unit tests.

- Given **n>=k** samples, where **c** are correct, an unbiased estimator is:

$$\text{pass@}k := \mathop{\mathbb{E}}_{\text{Problems}} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right]$$

Transformer United

- CodeX: Training Details

- Dataset: **159 GB** of code collected from **54 million** repositories

- For efficient training: fine-tuned from GPT-3 models of different sizes

- Extra spaces in tokenizer.

Transformer United

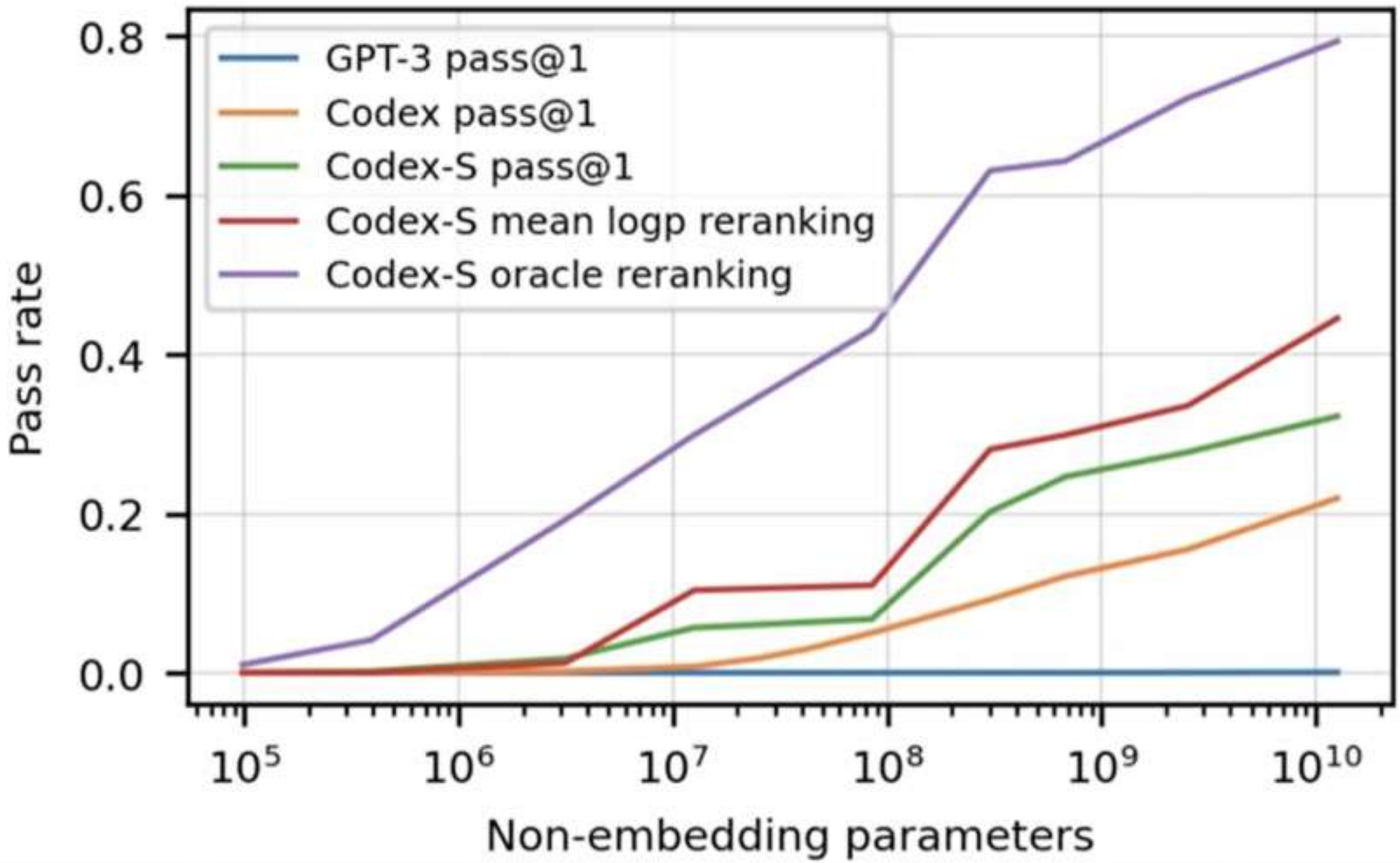
- Training Codex-S

- Goal: Finetune Codex on standalone functions which are correct

- Gather these functions from
  - Competitive programming problems
  - Tracing code execution when running integration tests for projects with CI enabled.

Transformer United

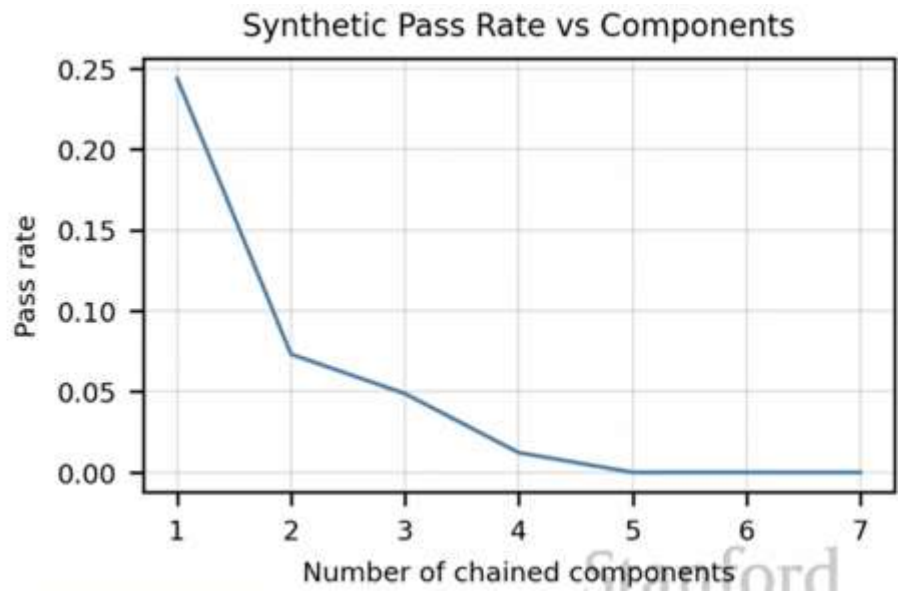
- Codex and Codex-S Performance



Transformer United

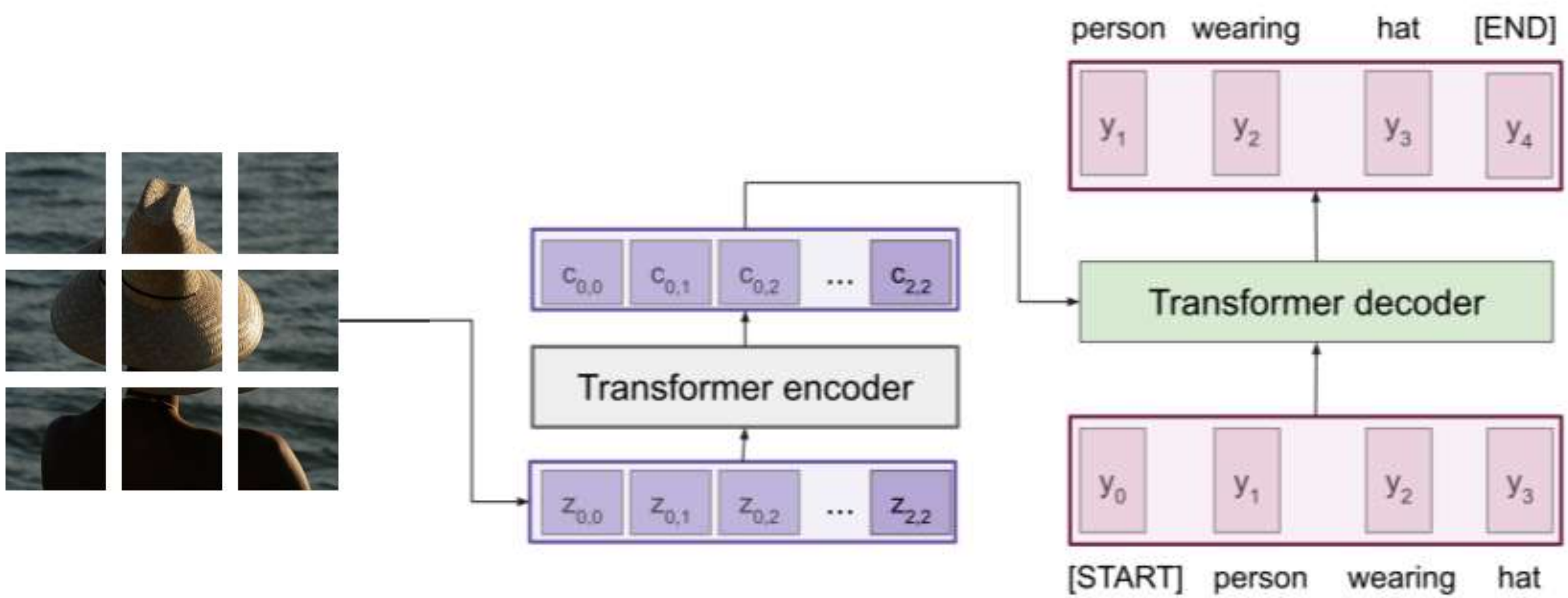
- Codex and Codex-S Limitation
- Binding & Composition

```python
def do_work(x, y, z, w):
    """ Add 3 to y, then subtract 4
    from both x and w. Return the
    product of the four numbers. """
    t = y + 3
    u = x - 4
    v = z * w
    return v
```



Vision Transformer

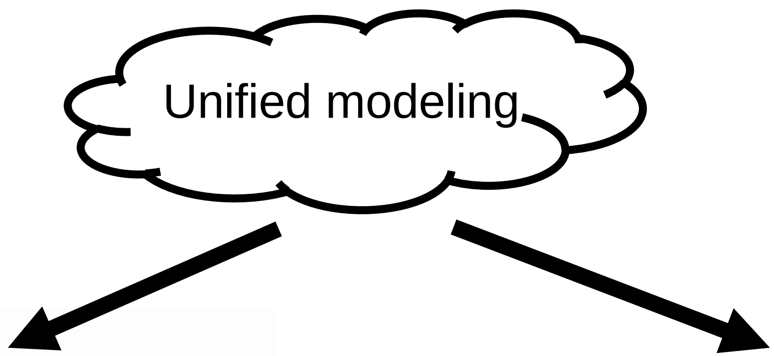
- Image Captioning using ONLY transformers
- Transformers from pixels to language

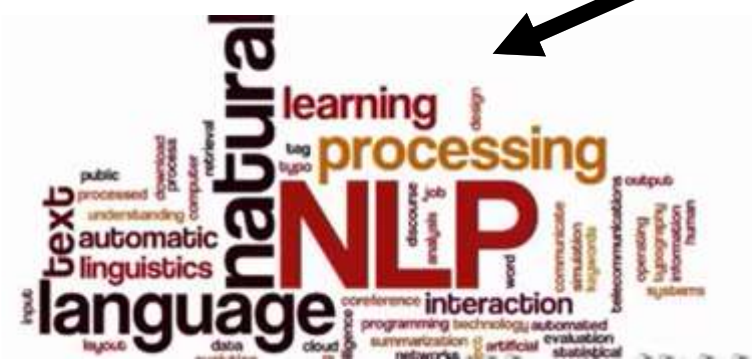


Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale ICLR2021

Vision Transformer

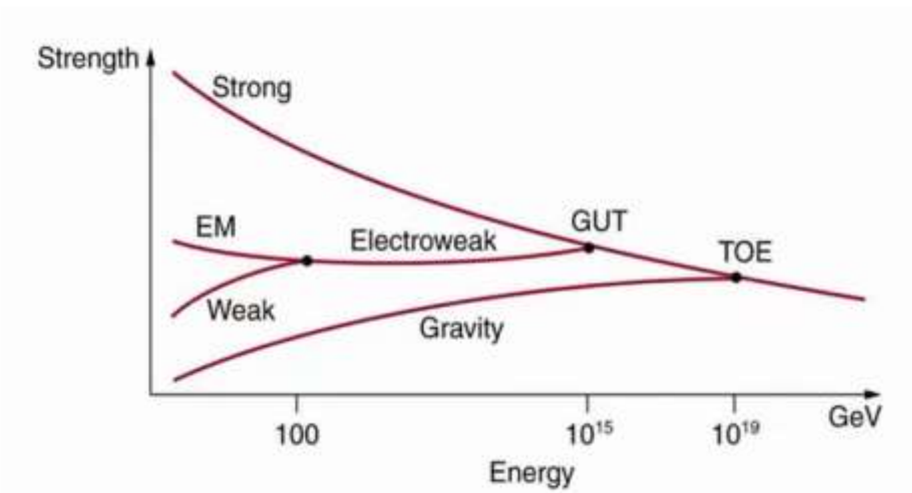
- Motivation: unification story for AI (NLP and CV)
- Beauty; Facilitate joint modeling; Share knowledge deeply



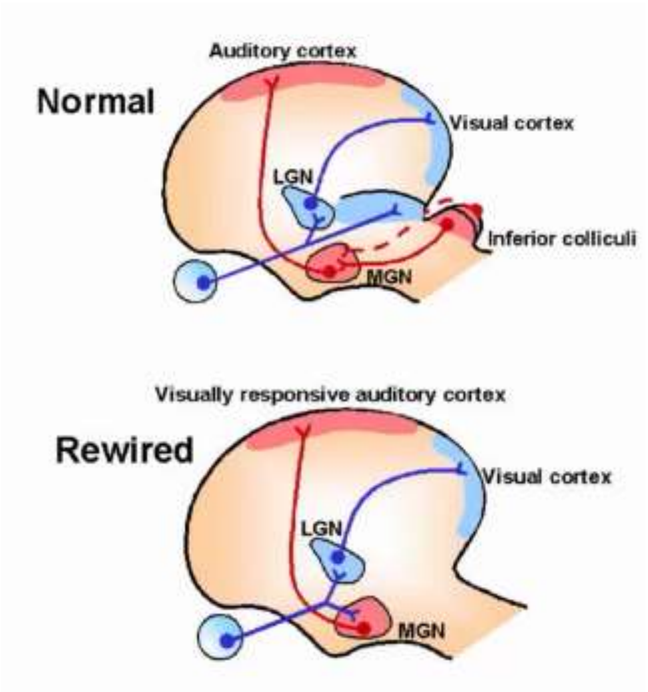


Vision Transformer

- Motivation: unification story for AI (NLP and CV)
- Beauty; Facilitate joint modeling; Share knowledge deeply



Physics



Human neurons

Vision Transformer

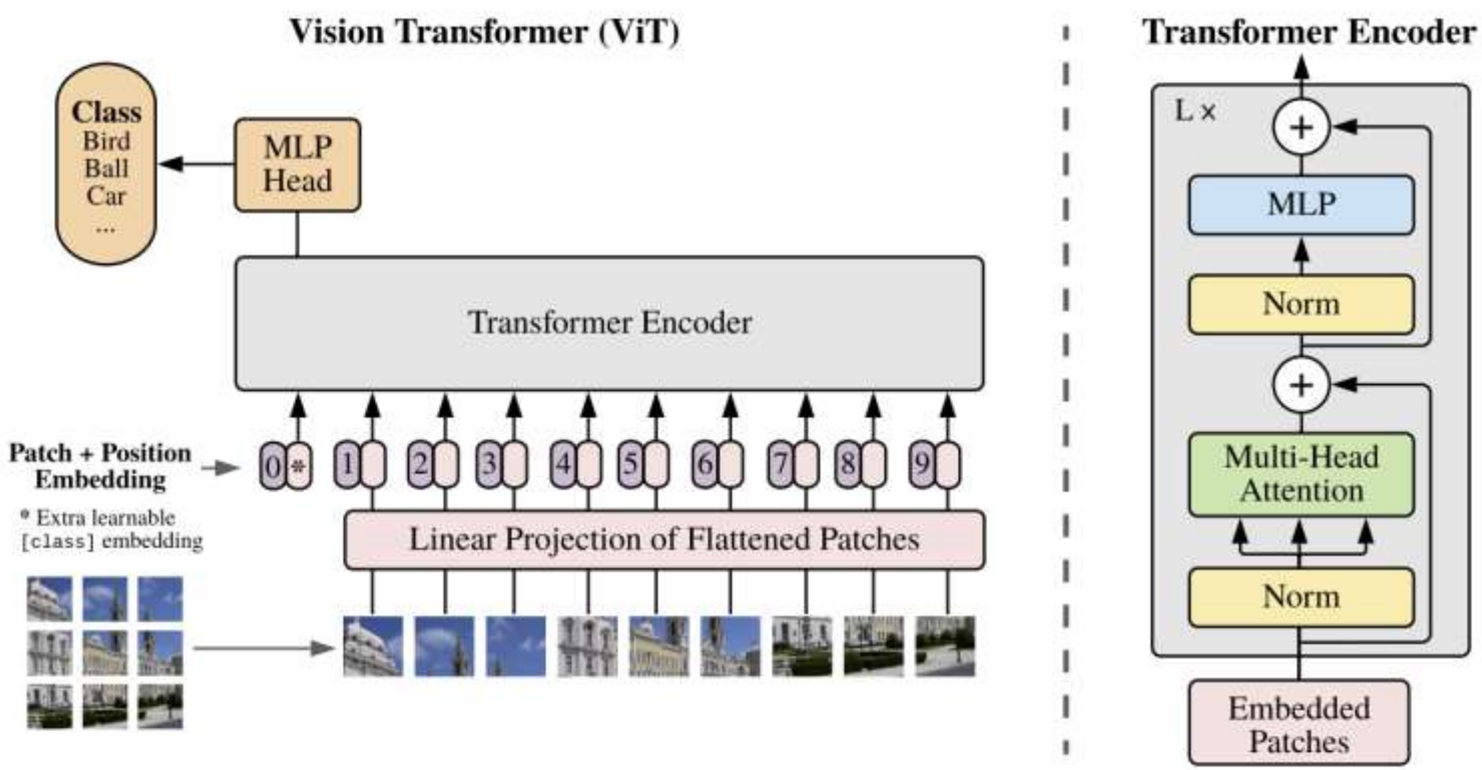
- Image Captioning using ONLY transformers
- Transformers from pixels to language
- SOTA performance on ImageNet-1K

ImageNet-1K image classification

Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR2021

Vision Transformer

- ViT (10/2020)
- SOTA performance on ImageNet-1K image classfication



Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale ICLR2021

Vision Transformer

- Image Captioning using ONLY Transformers
- Vision Transformers vs. ResNets

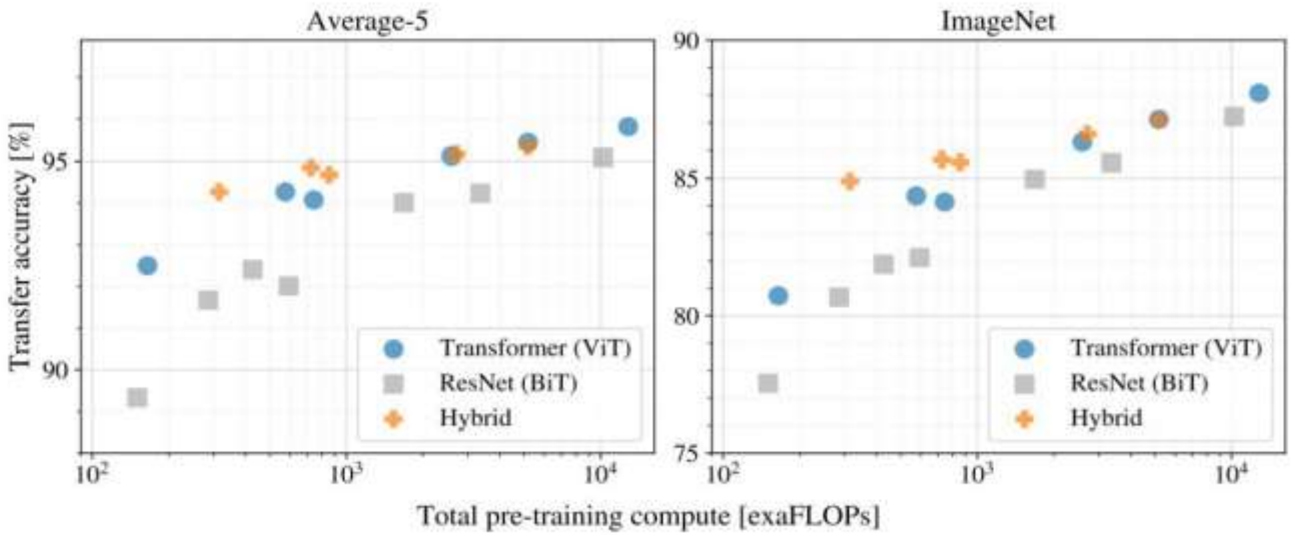
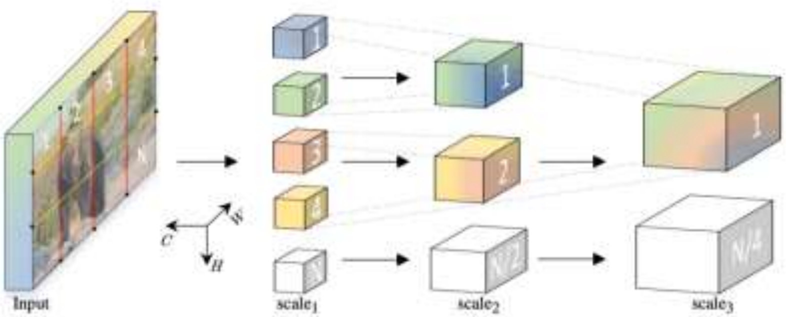


Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

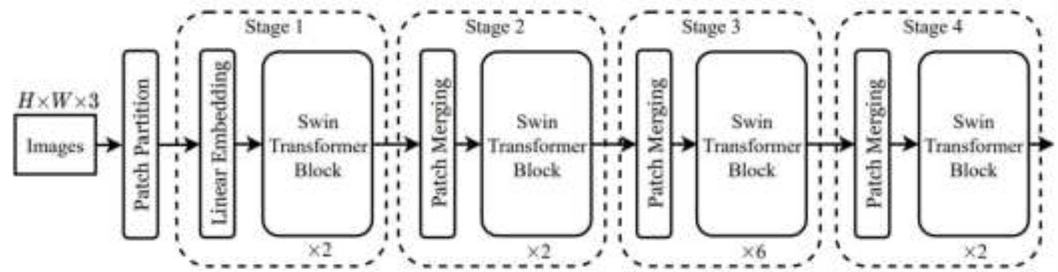
Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR2021

Vision Transformers

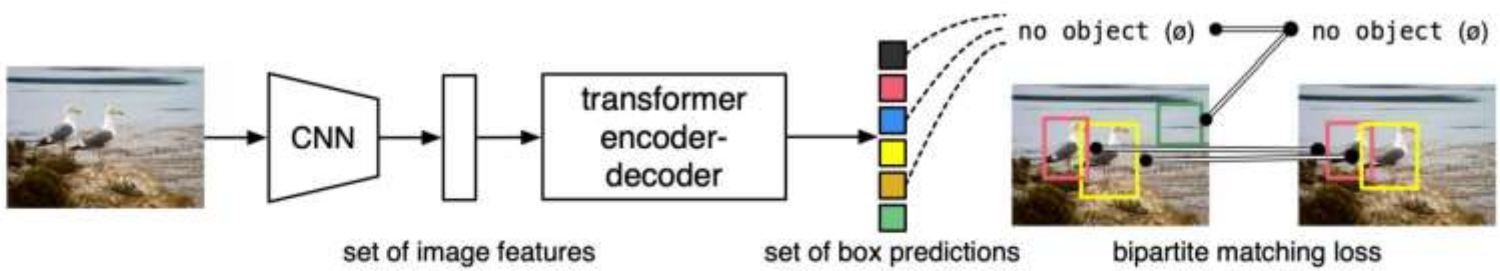
- Still ongoing ……



Fan et al, "Multiscale Vision Transformers", ICCV 2021



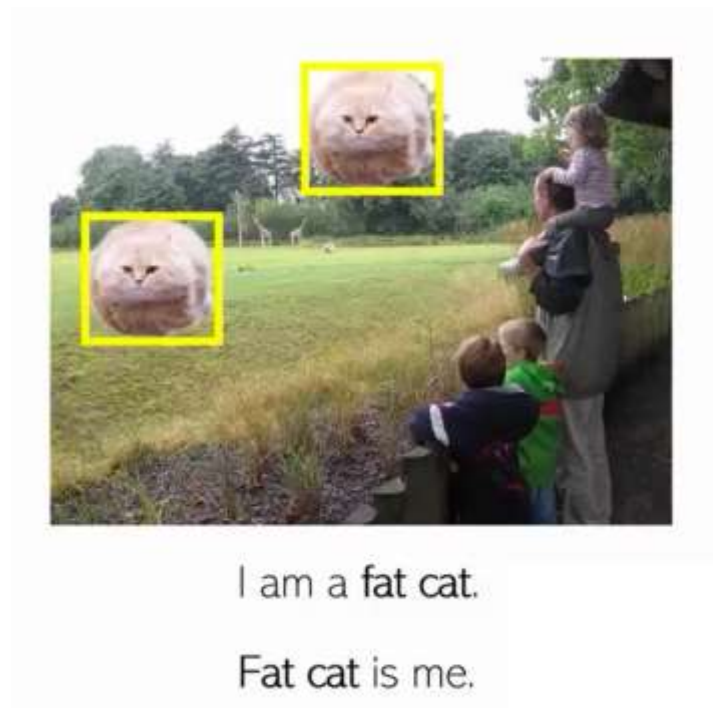
Liu et al, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", CVPR 2021



Carion et al, "End-to-End Object Detection with Transformers", ECCV 2020

Swin Transformer

- Problem of ViT: don't consider the difference between textual and visual signals



Liu et al, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", ICCV2021

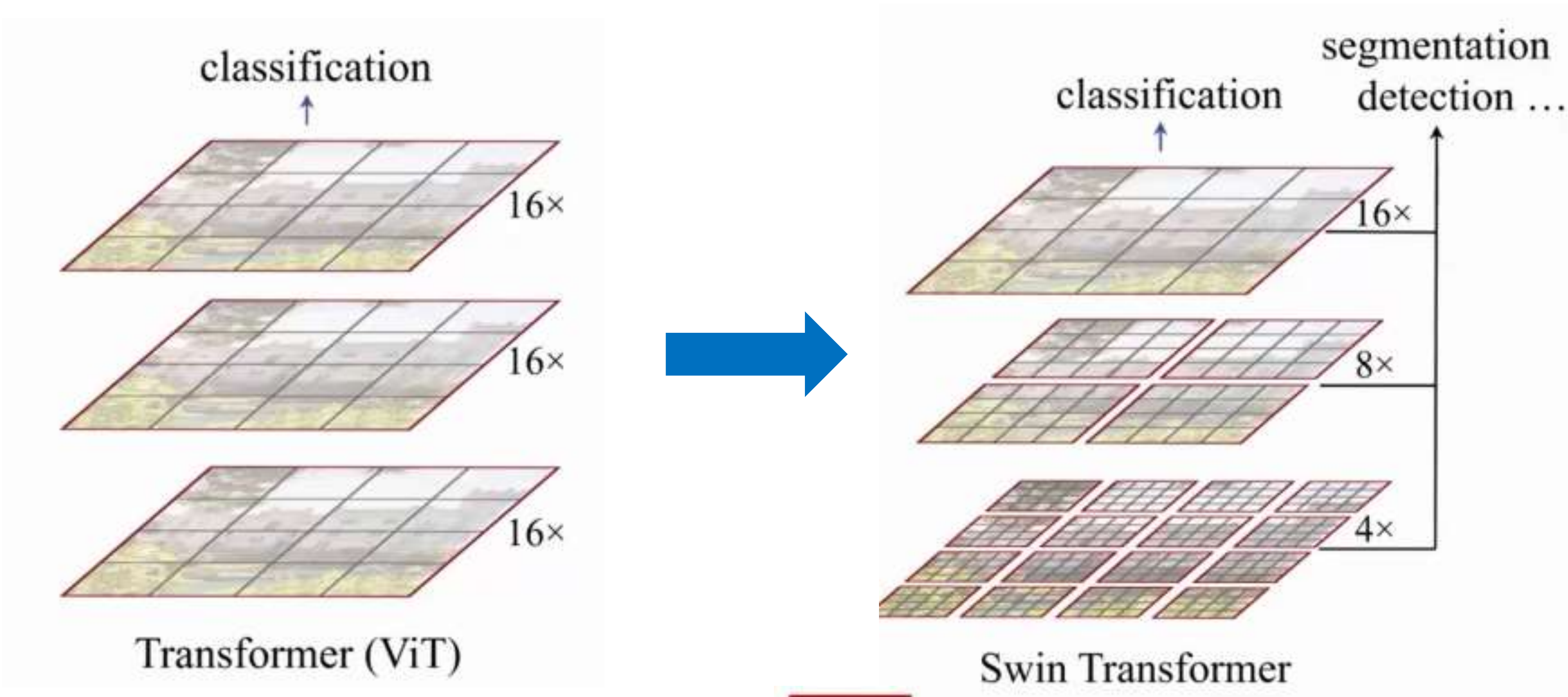
Swin Transformer

- Problem of ViT: mainly for image classification

Liu et al, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", ICCV2021

Swin Transformer

- Reconsider the good priors for visual signals
- Hierarchy / Locality / Translation invariance

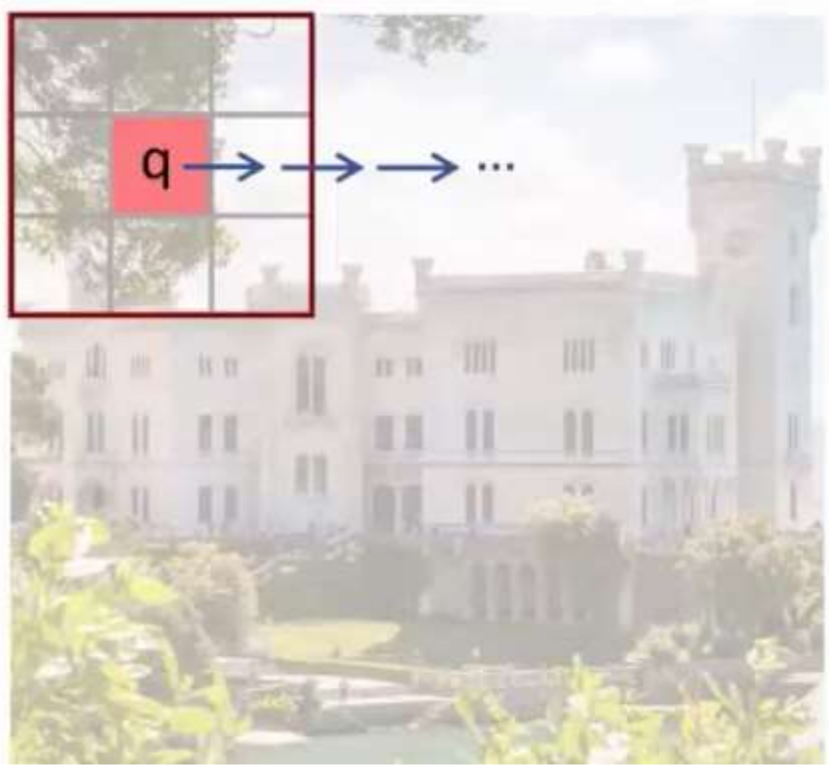


Patch/Feature bin

Computation scope of self-attention

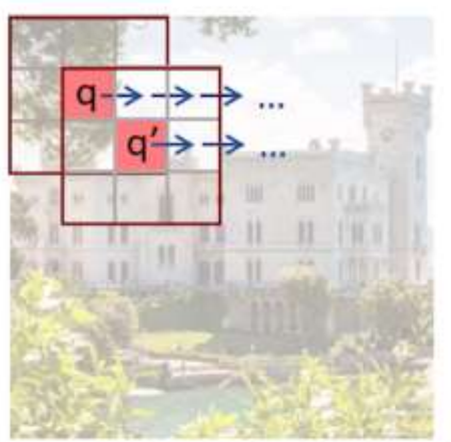
Swin Transformer

- How about sliding window as in CNN?
- Slow in real computation → Different queries use different key sets

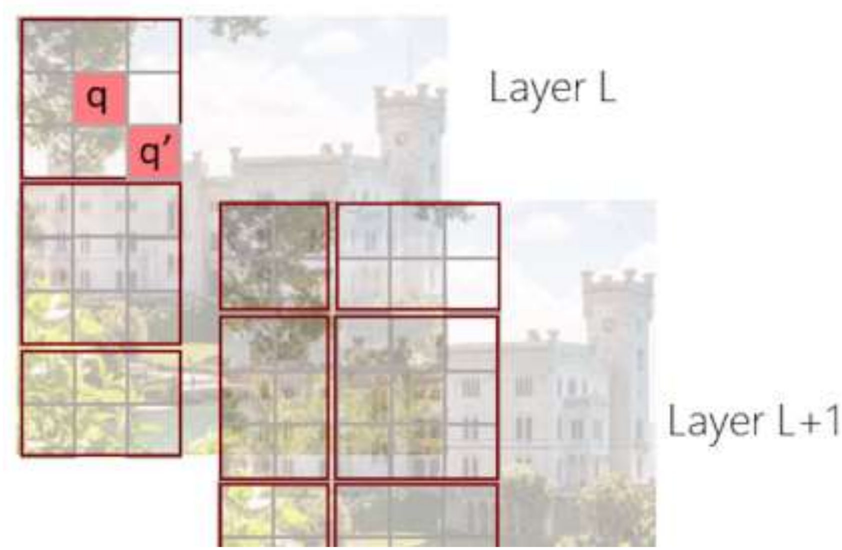


Swin Transformer

- Key idea: locality by **S**hifted **win**dows
- Non-overlapped windows (faster real speed than sliding windows)
- Windows are shifted in the next layer



Sliding window
LR-Net, ICCV2019



Shifted window
Swin, ICCV2021

Swin Transformer

- SOTA performance on a variety of tasks

- Backbone-level comparison
  - Performs consistently better than CNN on various object detectors and various model sizes (+3~4.5 mAP)

(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS	
Cascade	R-50	46.3	64.3	50.5	82M	739G	18.0	
Mask R-CNN	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3	+4.2
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3	
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3	+3.7
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6	
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0	+3.5
Sparse	R-50	44.5	63.4	48.2	106M	166G	21.0	
R-CNN	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4	+3.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS	
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4	
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0	
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3	+4.2
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8	
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0	+3.7
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4	
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6	+3.6

Swin Transformer

- SOTA performance on a variety of tasks

Swin Transformer

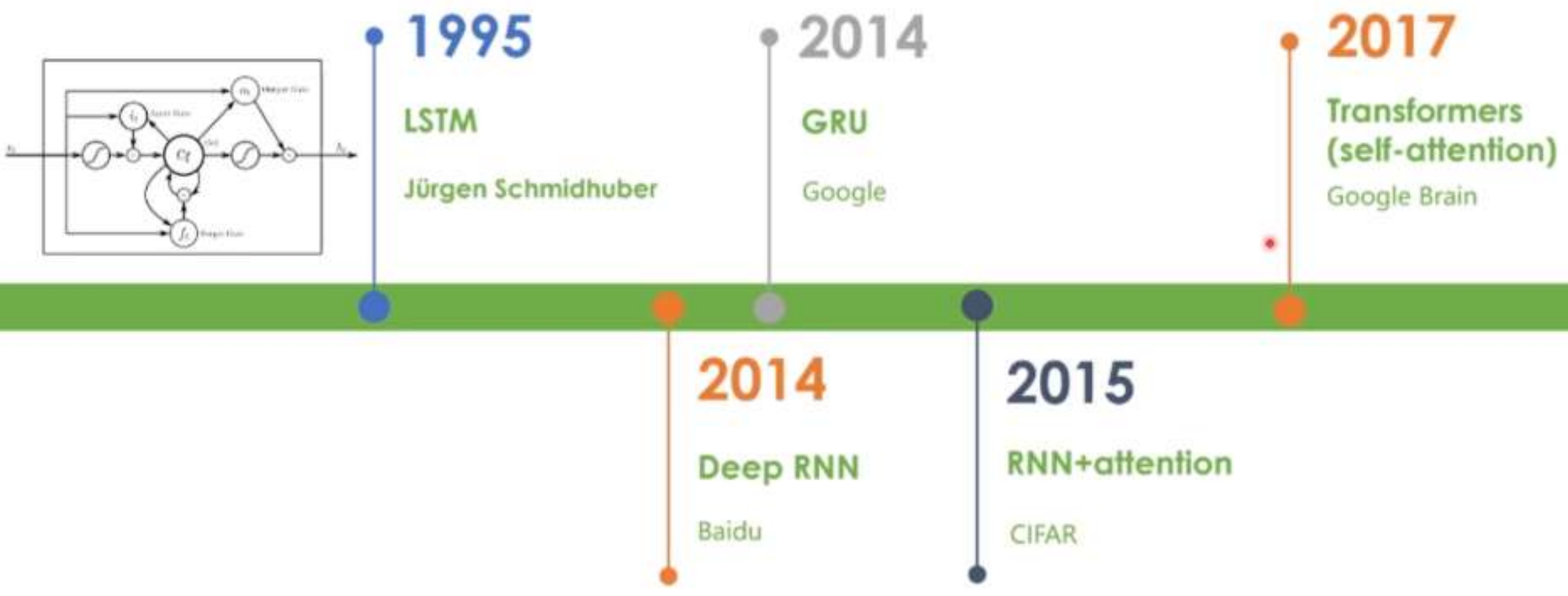
- SOTA performance on a variety of tasks

1	**SwinV2-G** (UperNet)	59.9	✓	Swin Transformer V2: Scaling Up Capacity and Resolution
2	**SeMask** (SeMask Swin-L MSFaPN-Mask2Former)	58.2	✓	SeMask: Semantically Masked Transformers for Semantic Segmentation
	SeMask (SeMask Swin-L FaPN-Mask2Former)	58.0	✓	SeMask: Semantically Masked Transformers for Semantic Segmentation
4	**SeMask** (SeMask Swin-L Mask2Former)	57.5	✓	SeMask: Semantically Masked Transformers for Semantic Segmentation
5	**BEiT-L** (ViT+UperNet, ImageNet-22k pretrain)	57.0	✓	BEiT: BERT Pre-Training of Image Transformers
6	**FaPN** (MaskFormer, Swin-L, ImageNet-22k pretrain)	56.7		
7	**SeMask** (SeMask Swin-L MaskFormer)	56.2		

1	**SwinV2-G** (HTC++)	63.1						✓	Swin Transformer V2: Scaling Up Capacity and Resolution	2021	Swin-Transformer
2	**Florence-CoSwin-H**	62.4						✓	Florence: A New Foundation Model for Computer Vision	2021	Swin-Transformer
3	**GLIP** (Swin-L, multi-scale)	61.5	79.5	67.7	45.3	64.9	75.0	✓	Grounded Language-Image Pre-training	2021	multiscale Vision Language Dynamic Head BERT-Base
4	**Soft Teacher + Swin-L** (HTC++, multi-scale)	61.3						✓	End-to-End Semi-Supervised Object Detection with Soft Teacher	2021	multiscale Swin-Transformer

Vision Transformers

- Recall the model evolution in NLP or sequential data

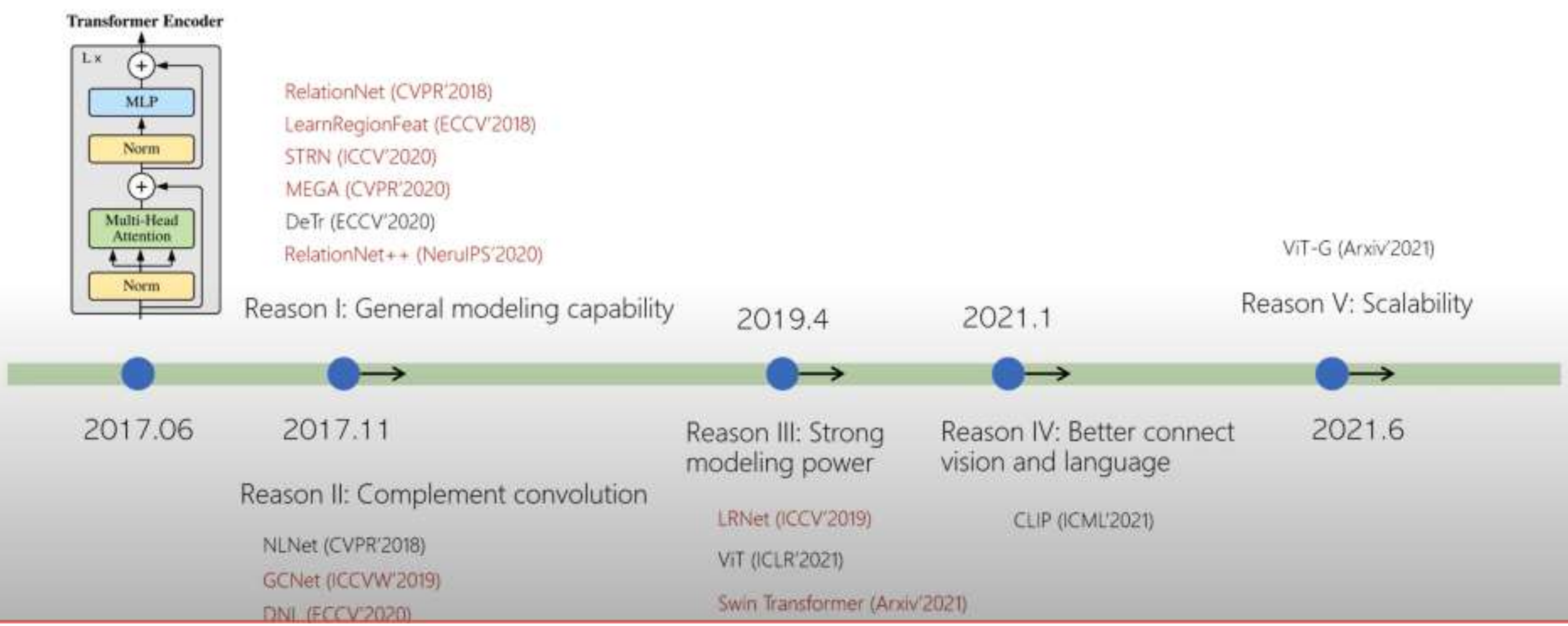


Vision Transformers

- Can NLP/CV share the same basic modules?

Vision Transformers

- Still unleash the power of Transformer in CV



Summary

- Transformer Applications
  - NLP, Vision
  - Cross/Multi- Modality

- Next time:
  - Prediction Problem