Institute of Electronics
National Yang Ming Chiao Tung University
Hsinchu, Taiwan

AI Training Course Series

Introduction to BERT

Lecture 7-1

Student: Zhi-Kai Xu
Advisor: Juinn-Dar Huang, Ph.D.

August 5, 2024

Outline

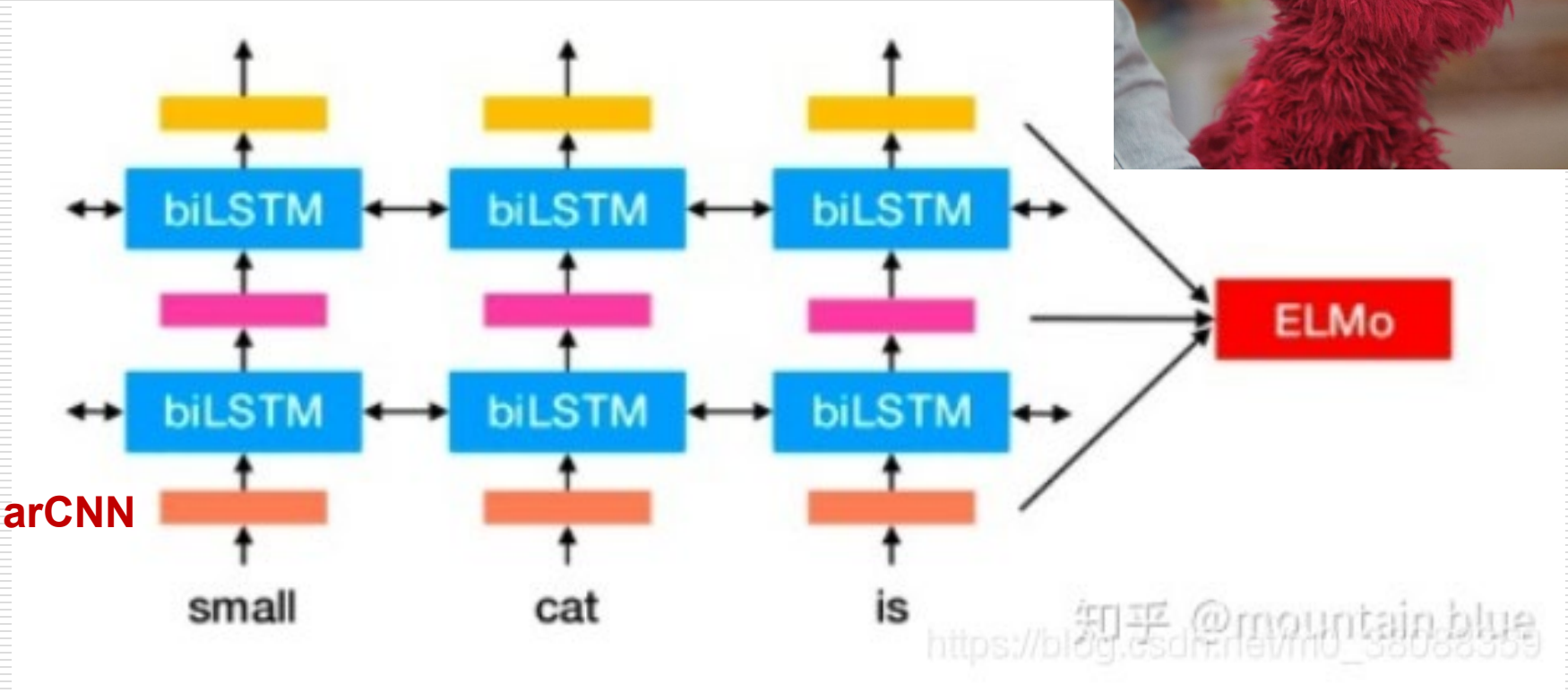
- BERT
  - pretraining (self-supervised learning, masked LM)
  - finetuning (supervised learning, transfer learning)
- Variants of BERT
- Friends of BERT
- Takeaways and references

Pretraining of BERT

Characters at Sesame Street

ELMo

- **ELMo** : **E**mbeddings from **L**anguage **Mo**dels
  Allen Institute for Artificial Intelligence & Univ. of Washington, 2018
  - based on bidirectional LSTMs
  - #parameters = 94M



BERT (1/2)

- Obtain new state-of-the-art results on 11 natural language processing tasks

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Bert: Pre-training of deep bidirectional transformers for language understanding
J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv ..., 2018 - arxiv.org
... We introduce **BERT** and its detailed implementation in this ... For finetuning, the **BERT** model is first initialized with the pre-... A distinctive feature of **BERT** is its unified architecture across ...
☆ Save 99 Cite Cited by 106371 Related articles All 46 versions »

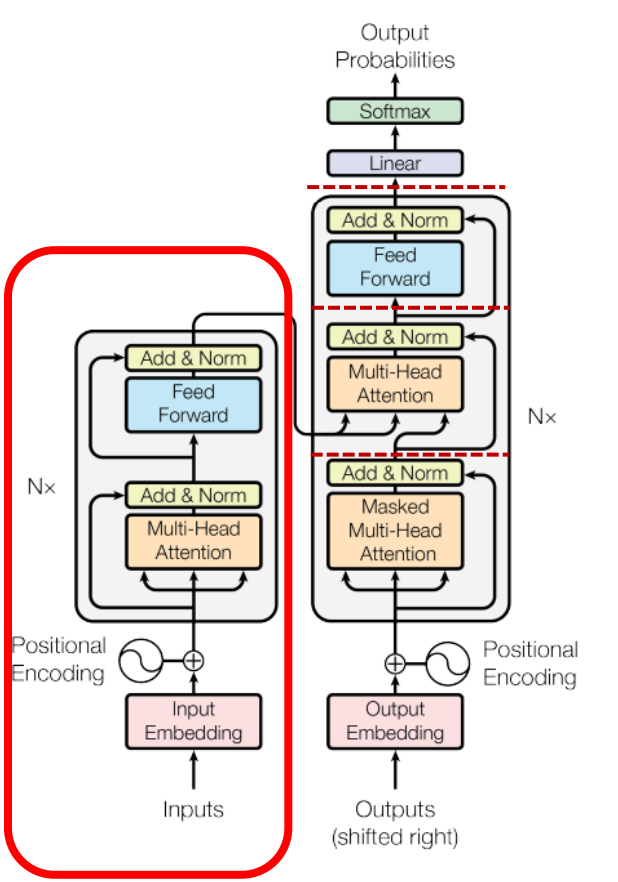
BERT (2/2)

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - proposed by Google
  - v1: Oct. 2018 ; v2: May 2019

- BERT-Base
  - #layers: 12 ; #heads: 12
  - embedding dim: 768
  - #parameters: **110M**

- BERT-Large
  - #layers: 24 ; #heads: 16
  - embedding dim: 1024
  - #parameters: **340M**

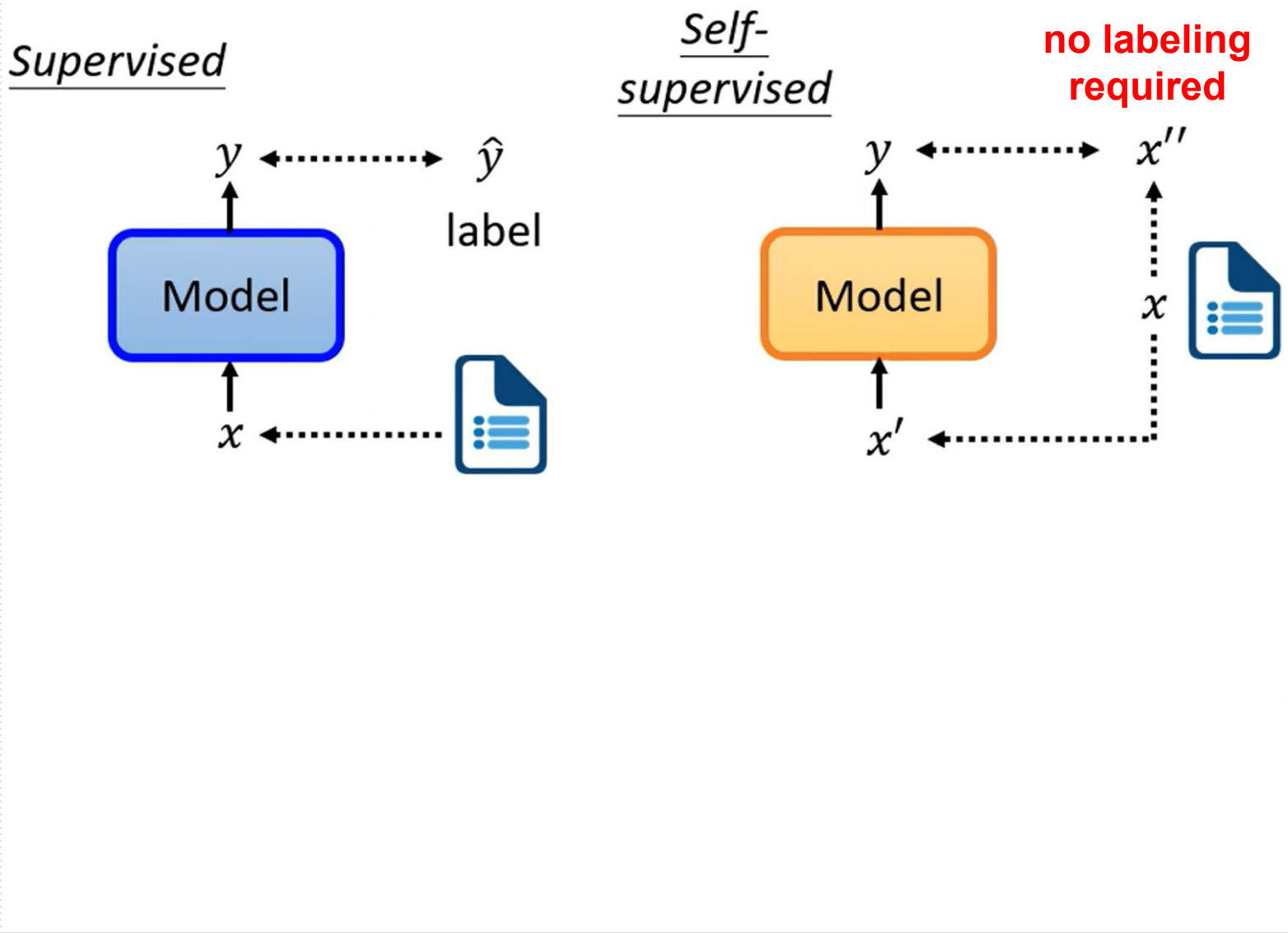
(In both, expansion ratio r = 4 in FFN)



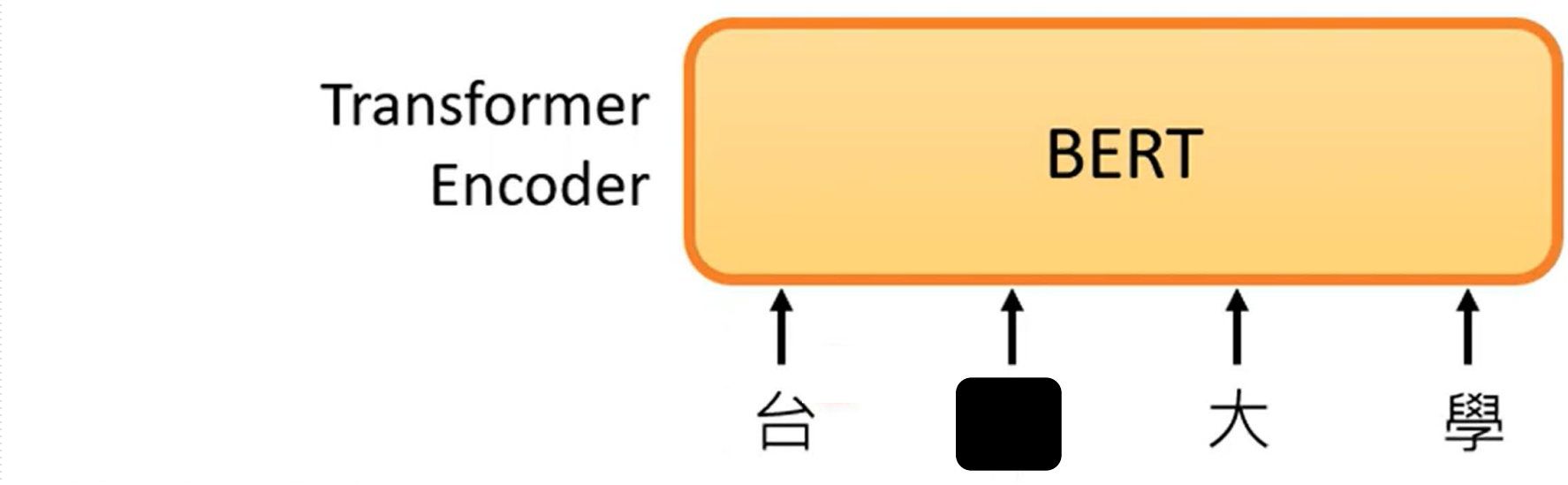
Key Features of BERT

- Self-supervised learning
- Encoder part of Transformer
- First, **pretrain** a large language model (LM)
  - **masked** language model (**MLM**)
- Then, **finetune** the pretrained model for downstream tasks
  - **transfer learning**

Supervised vs. Self-Supervised Learning

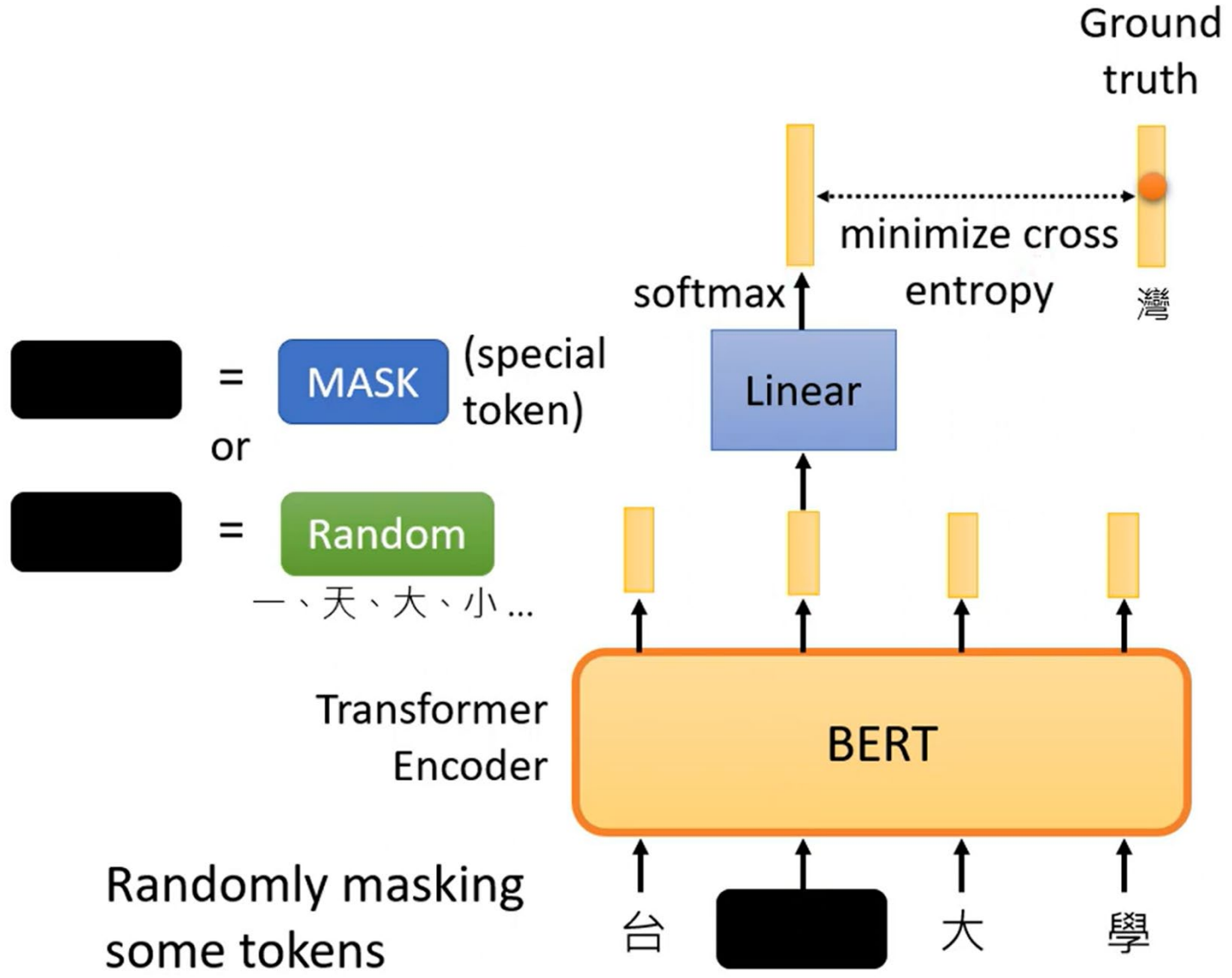


Pretraining Phase – Cloze Test (1/3)

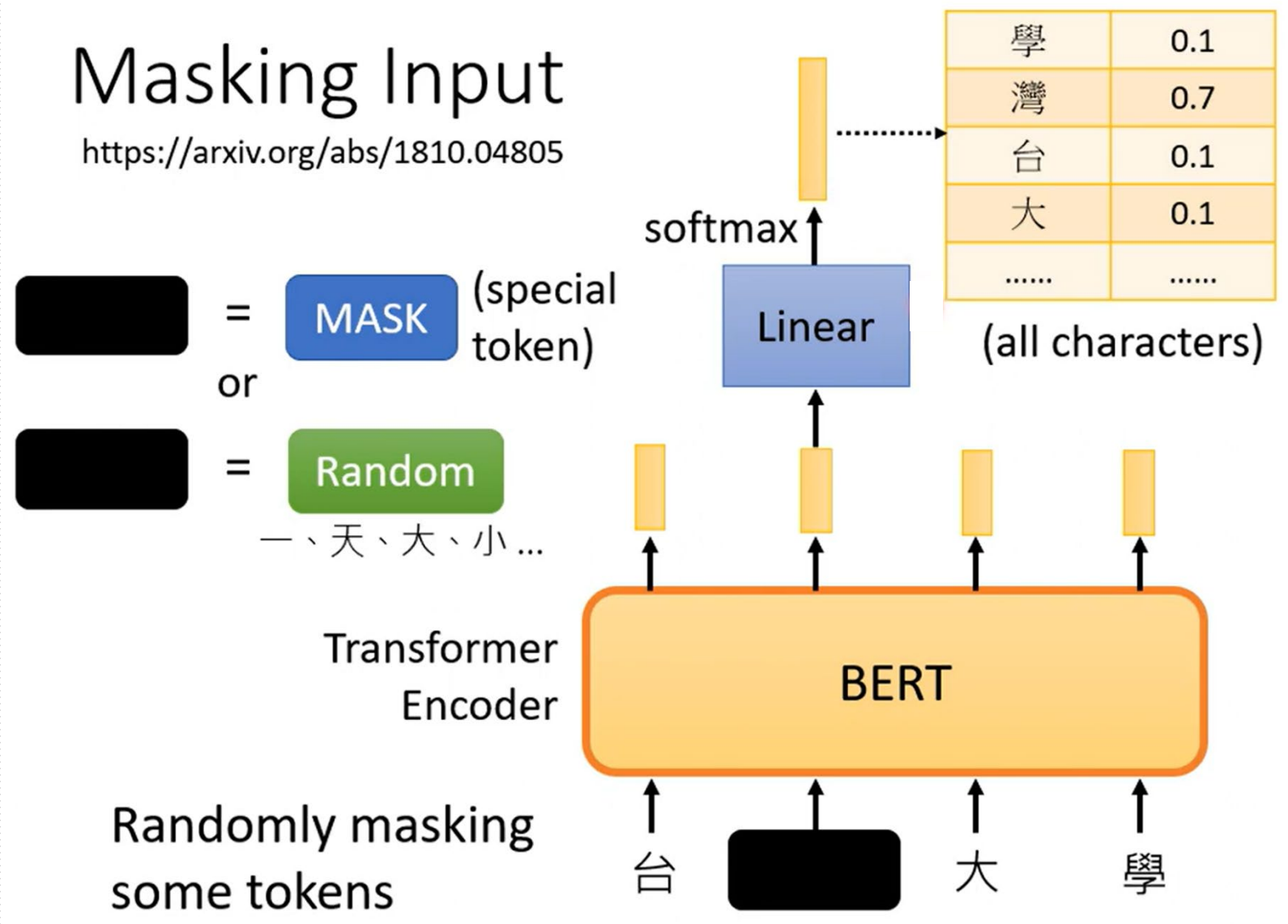


15%, typically

Pretraining Phase – Cloze Test (2/3)

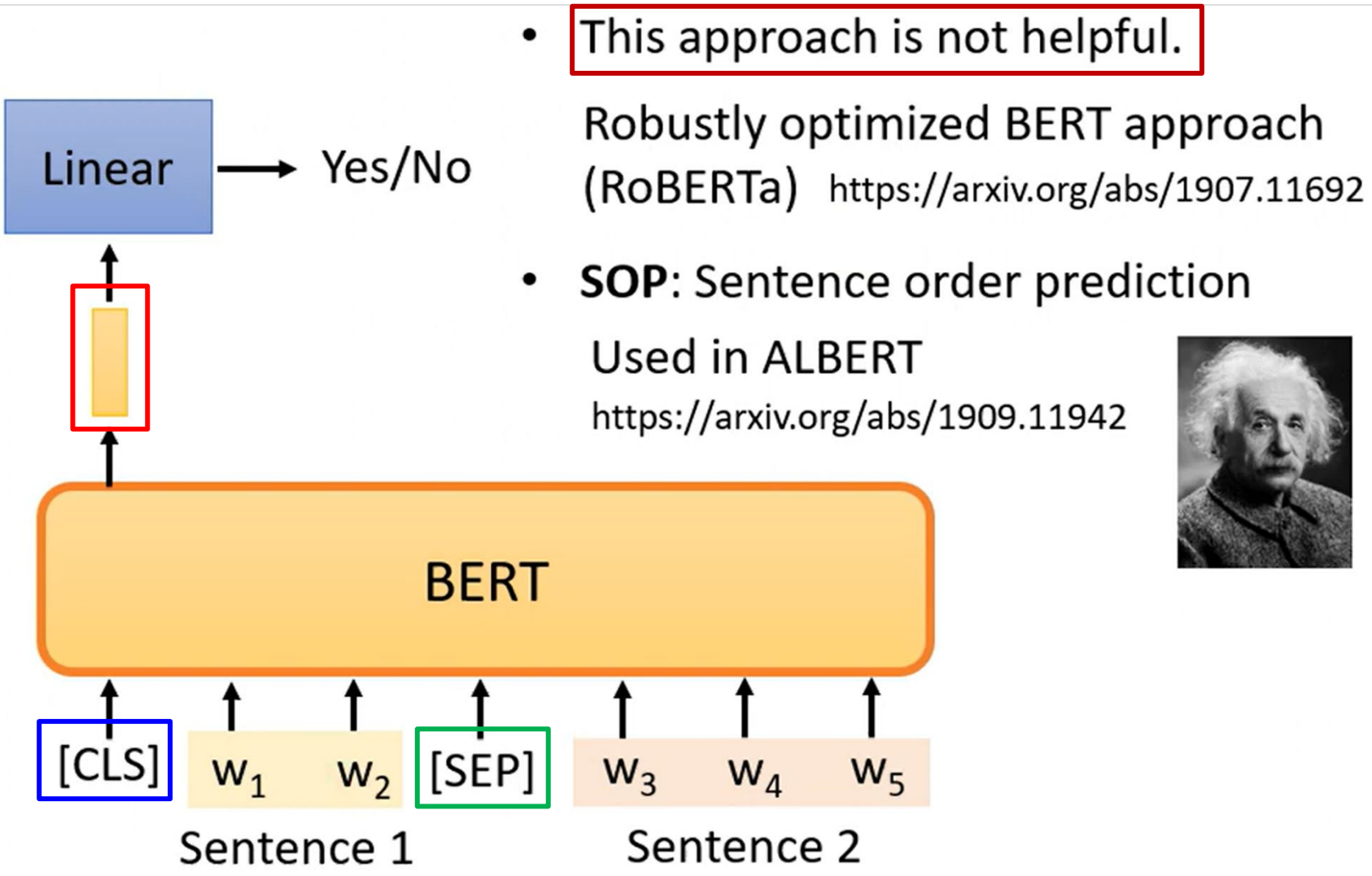


Pretraining Phase – Cloze Test (3/3)



Pretraining Phase – NSP (1/2)

- **NSP**: **N**ext **S**entence **P**rediction



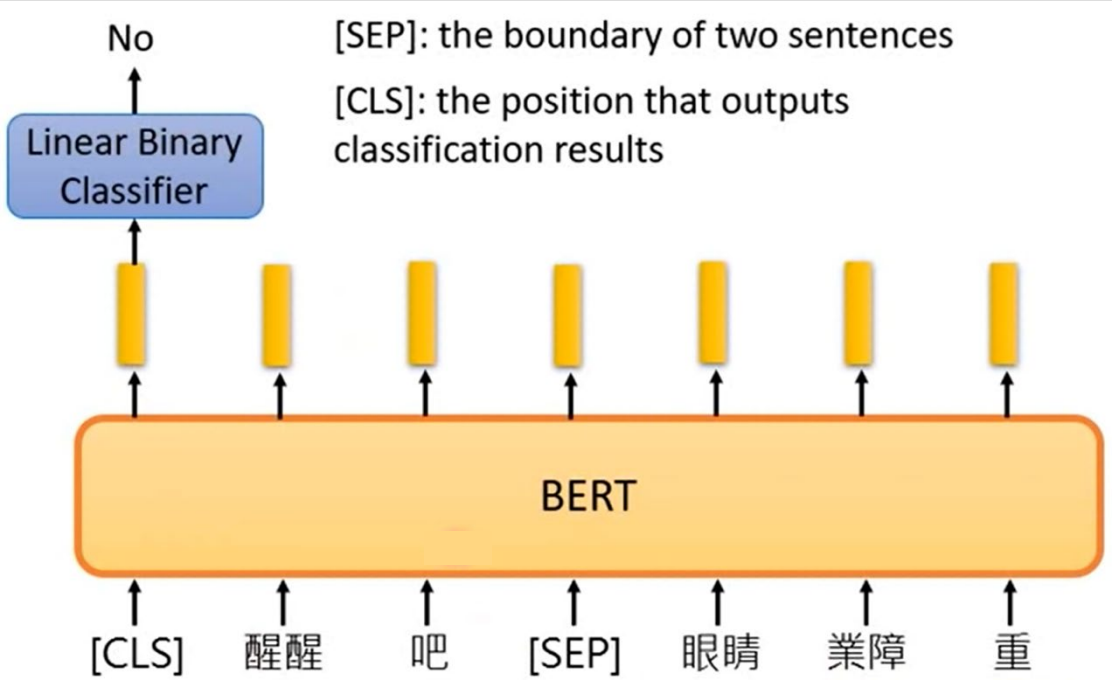
- This approach is not helpful.

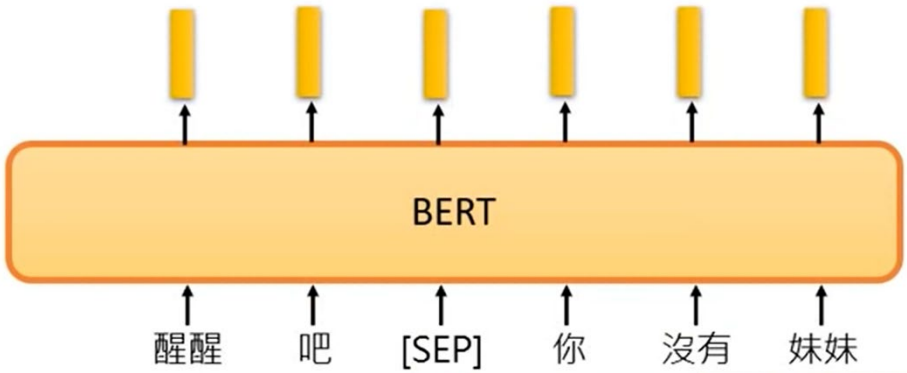
Robustly optimized BERT approach (RoBERTa) https://arxiv.org/abs/1907.11692

- **SOP**: Sentence order prediction

Used in ALBERT
https://arxiv.org/abs/1909.11942

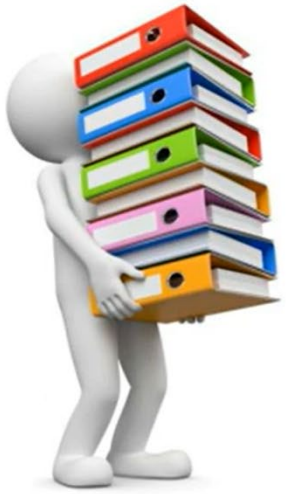
Pretraining Phase – NSP (2/2)





Finetuning of BERT

Pretraining vs. Finetuning



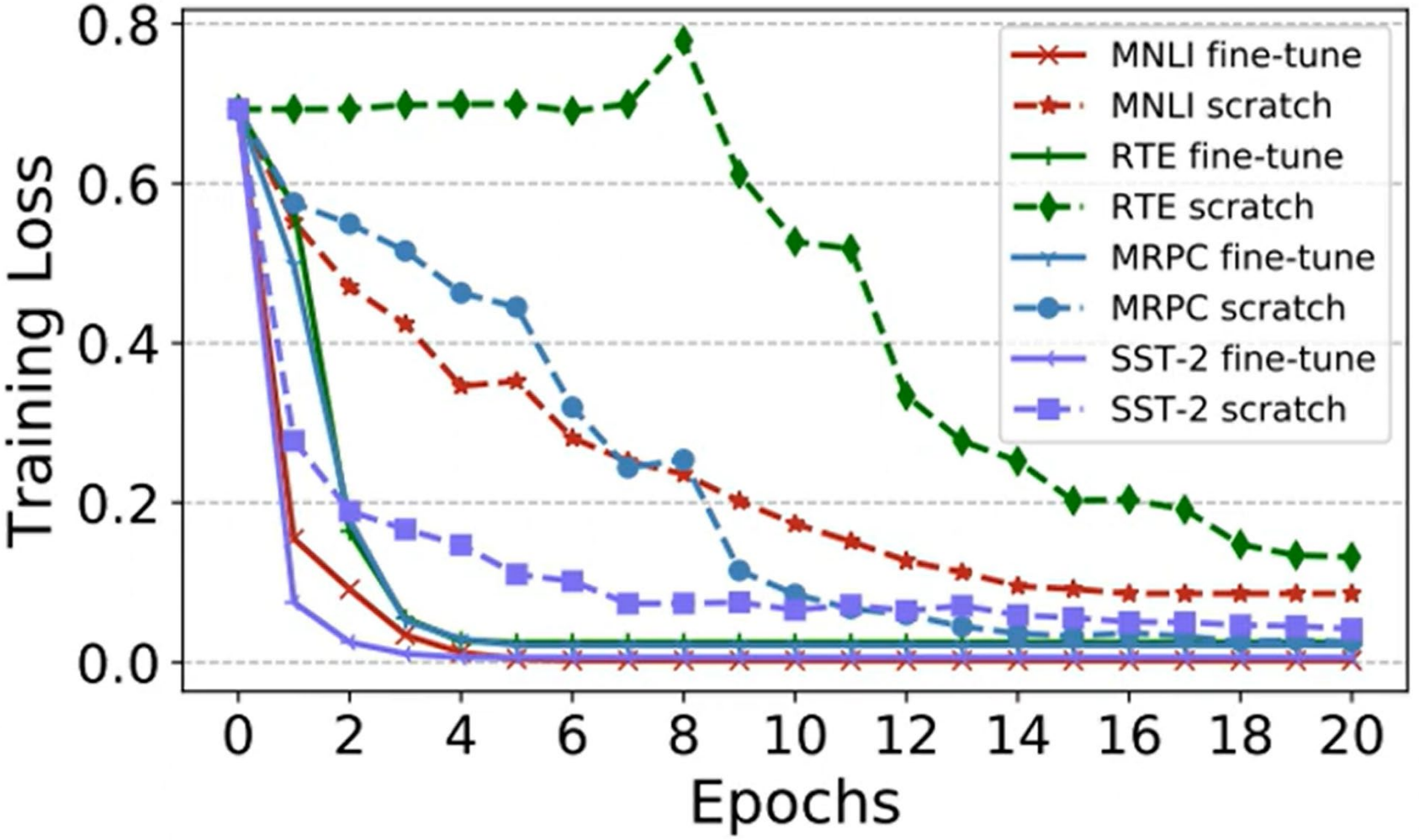
data labeling not required

Self-supervised Learning

BERT

- Masked token prediction
- Next sentence prediction

Pretraining vs. Training-from-Scratch



Source of image: https://arxiv.org/abs/1908.05620

GLUE Benchmark

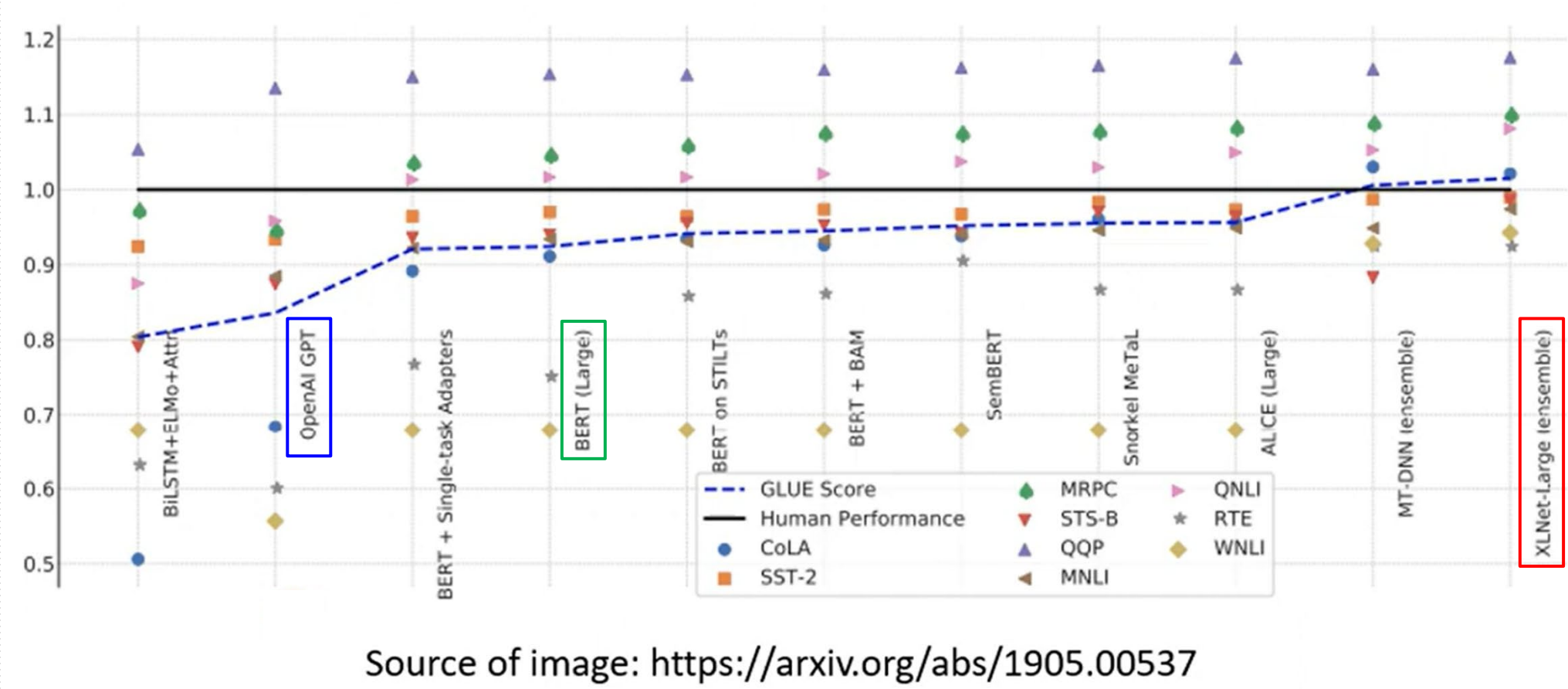
- **G**eneral **L**anguage **U**nderstanding **E**valuation
  https://gluebenchmark.com
  Chinese version, https://www.cluebenchmarks.com

  - Corpus of Linguistic Acceptability (CoLA)
  - Stanford Sentiment Treebank (SST-2)
  - Microsoft Research Paraphrase Corpus (MRPC)
  - Quora Question Pairs (QQP)
  - Semantic Textual Similarity Benchmark (STS-B)
  - Multi-Genre Natural Language Inference (MNLI)
  - Question-answering NLI (QNLI)
  - Recognizing Textual Entailment (RTE)
  - Winograd NLI (WNLI)

9 downstream tasks

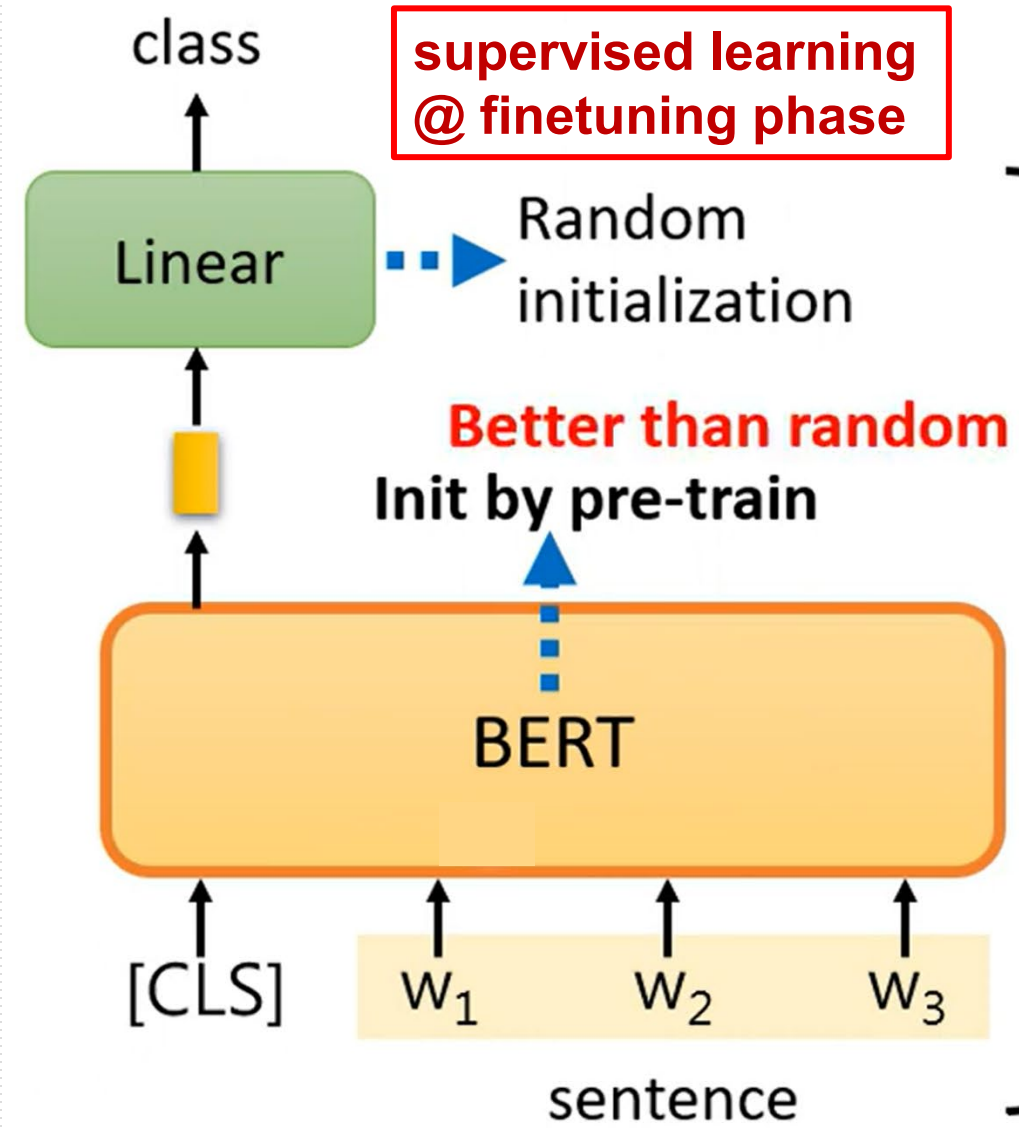
GLUE Scores

- BERT pushes the GLUE score to 80.5, compared to OpenAI GPT, which obtains 72.8 as of the date then



Source of image: https://arxiv.org/abs/1905.00537

Use Case 1 – Sentiment Analysis



supervised learning @ finetuning phase

Input: sequence
output: class

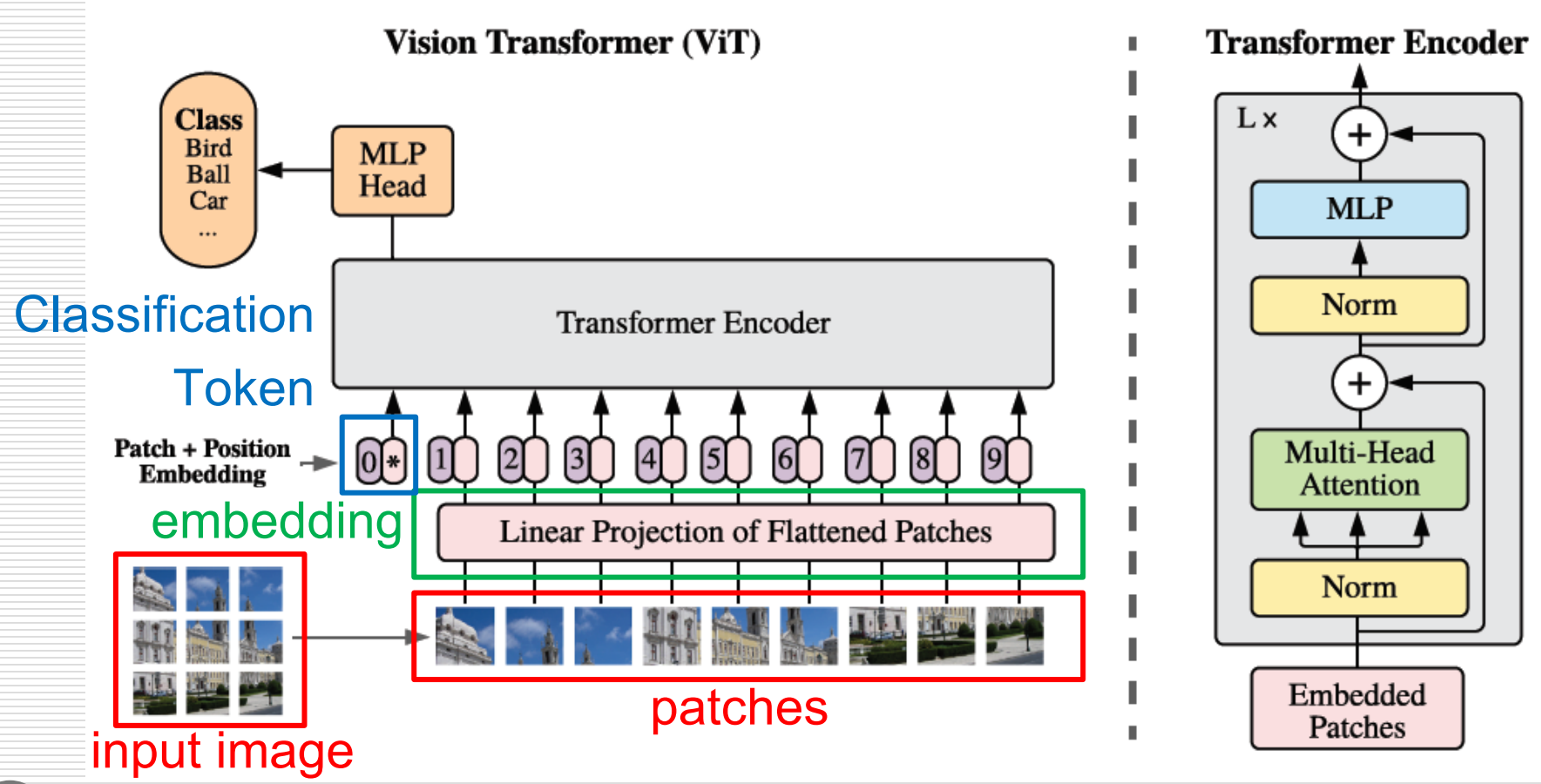
Example: (情感分析)
Sentiment analysis

this is good
↓
positive

This is the model to be learned.

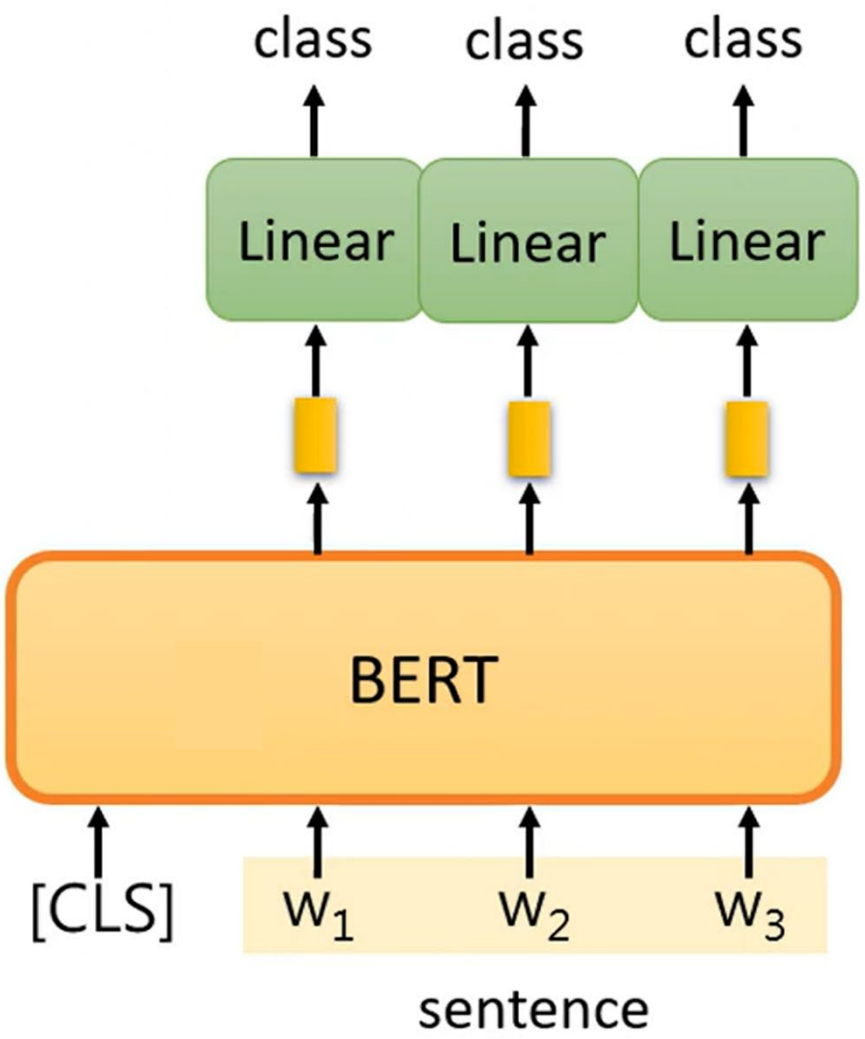
Vision Transformer (ViT)

- Proposed by Google in **2020**
  - a pure Transformer model (**NO** convolutions at all)
  - encoder **ONLY**



Use Case 2 – POS Tagging

- POS Tagging: Part-of-Speech Tagging (詞性標註)

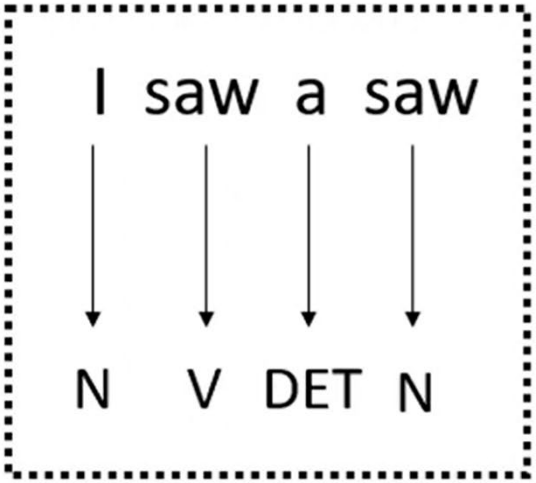


Input: sequence

output: same as input

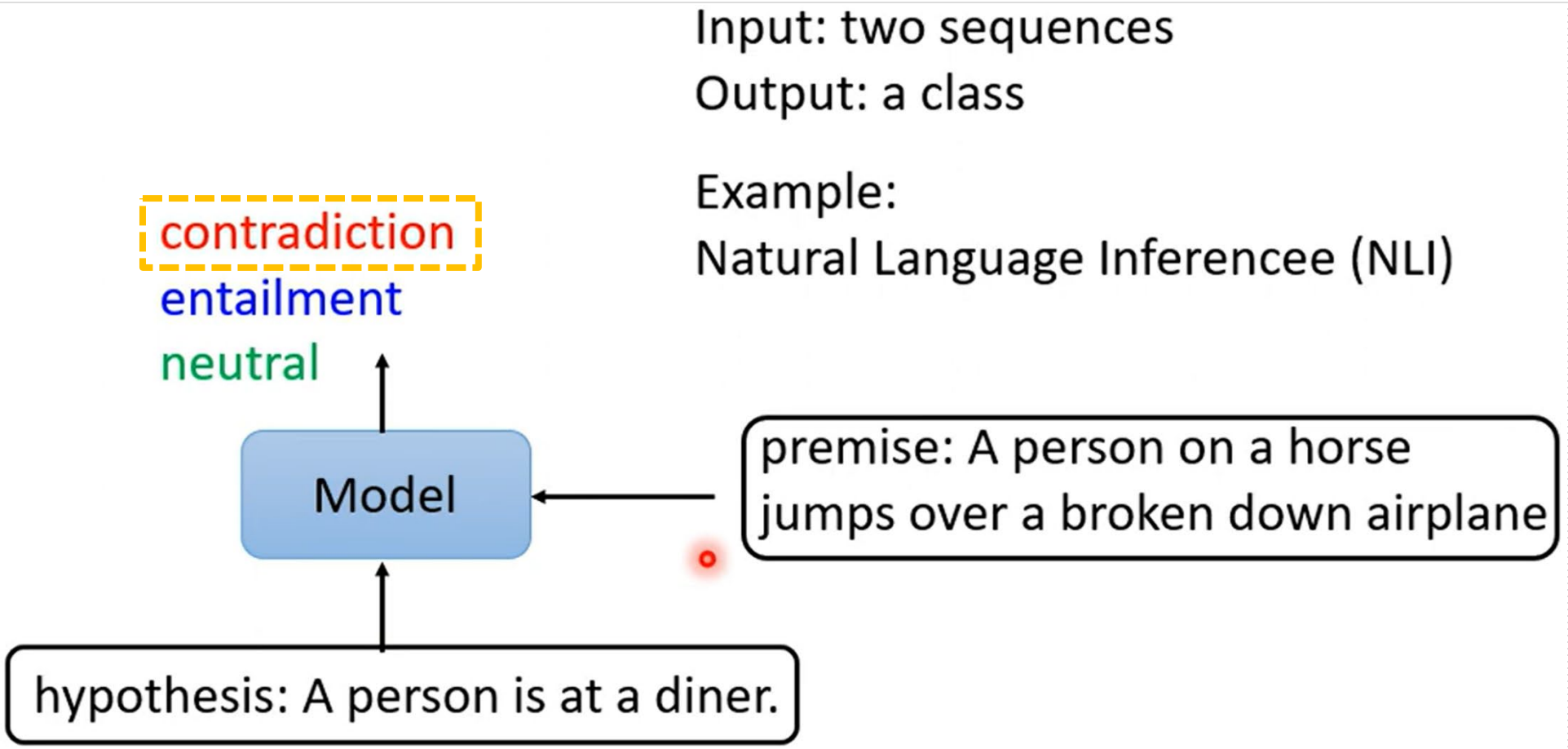
Example:

POS tagging

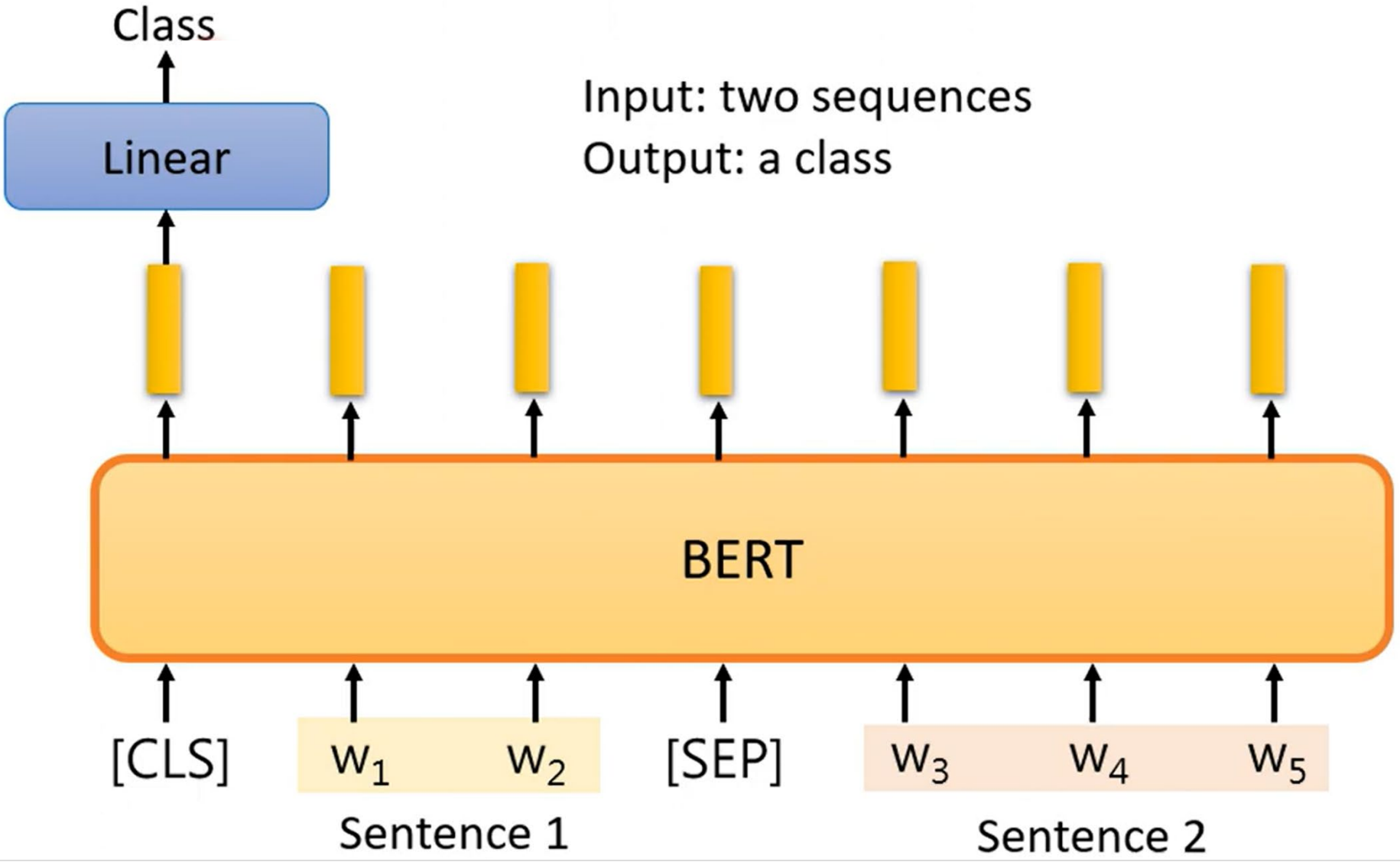


Use Case 3 – NLI (or RTE) (1/2)

- NLI: Natural Langrage Inference
- RTE: Recognizing Textual Entailment



Use Case 3 – NLI (or RTE) (2/2)

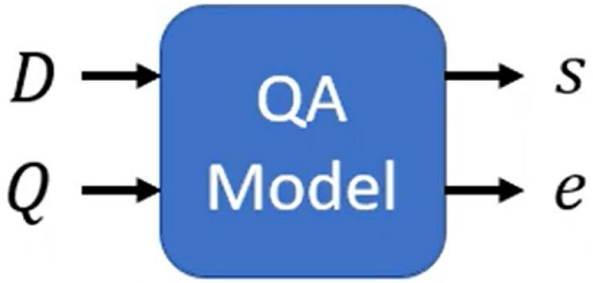


Use Case 4 – Extraction-Based QA (1/3)

- Extraction-based Question Answering (QA)

Document: $D = \{d_1, d_2, \cdots, d_N\}$

Query: $Q = \{q_1, q_2, \cdots, q_M\}$



output: two integers (s, e)

Answer: $A = \{d_s, \cdots, d_e\}$

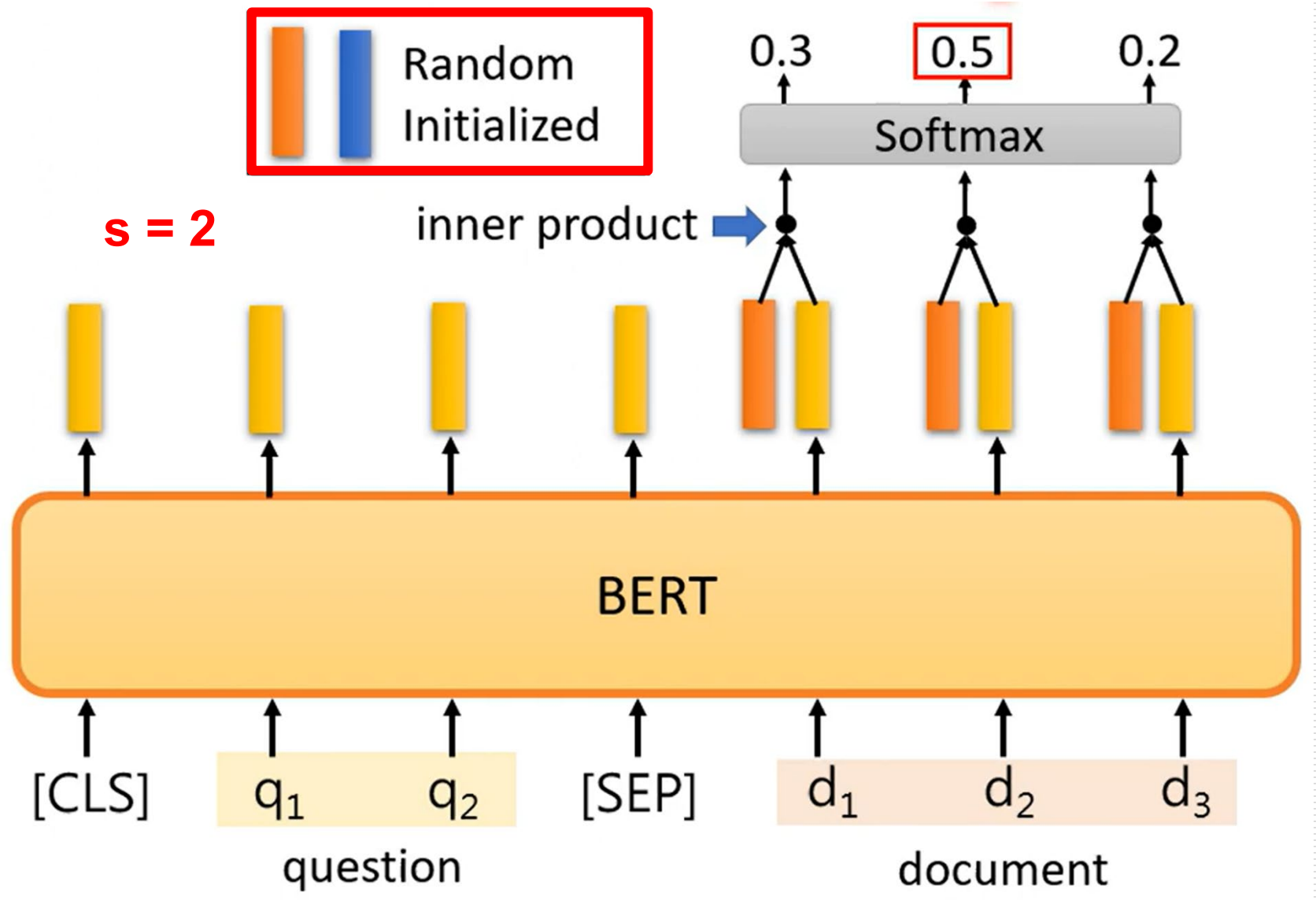
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

Q1: What causes precipitation to fall?
A: gravity

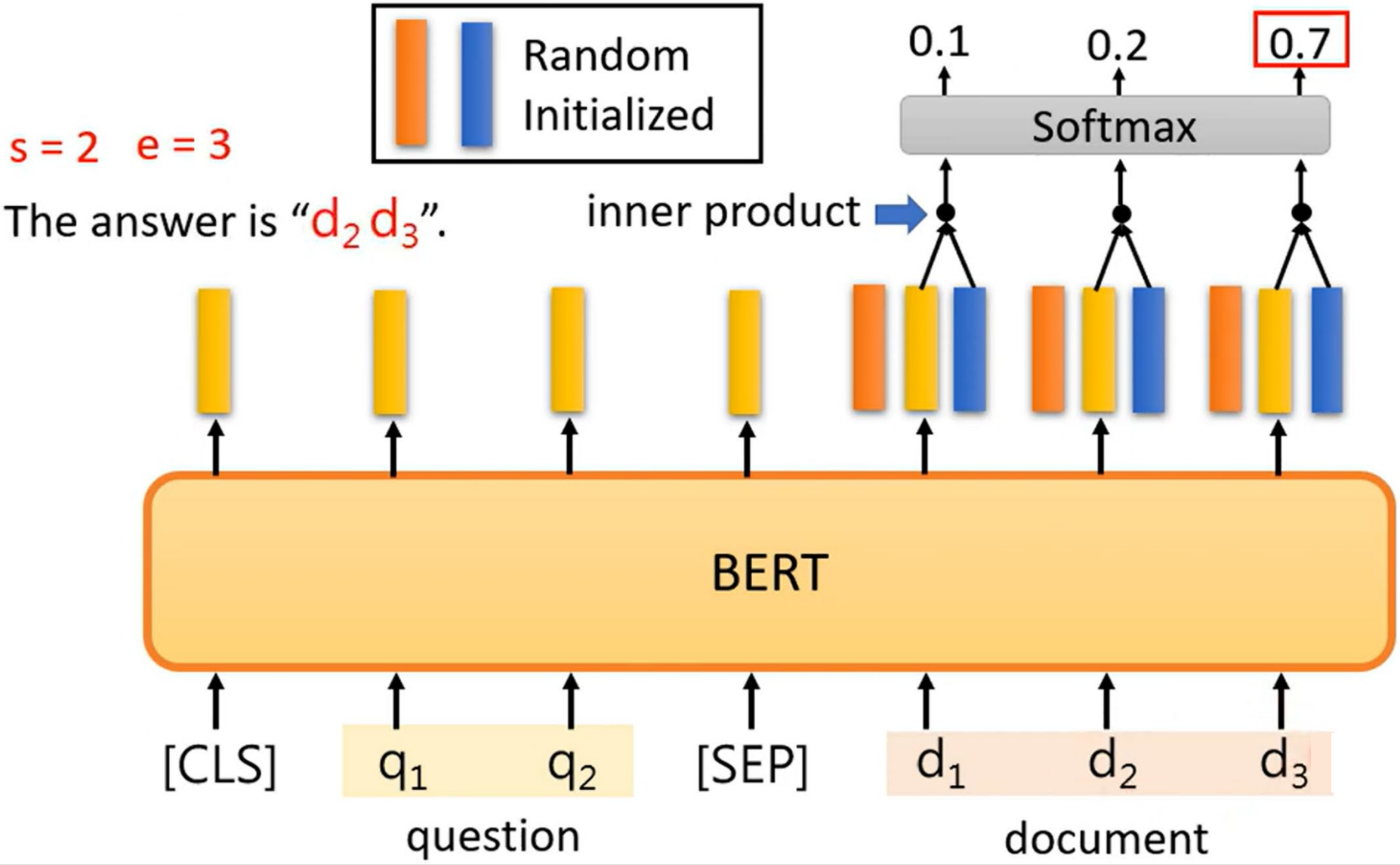
Q2: What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
A: graupel

Q3: Where do water droplets collide with ice crystals to form precipitation?
A: within a cloud

Use Case 4 – Extraction-Based QA (2/3)

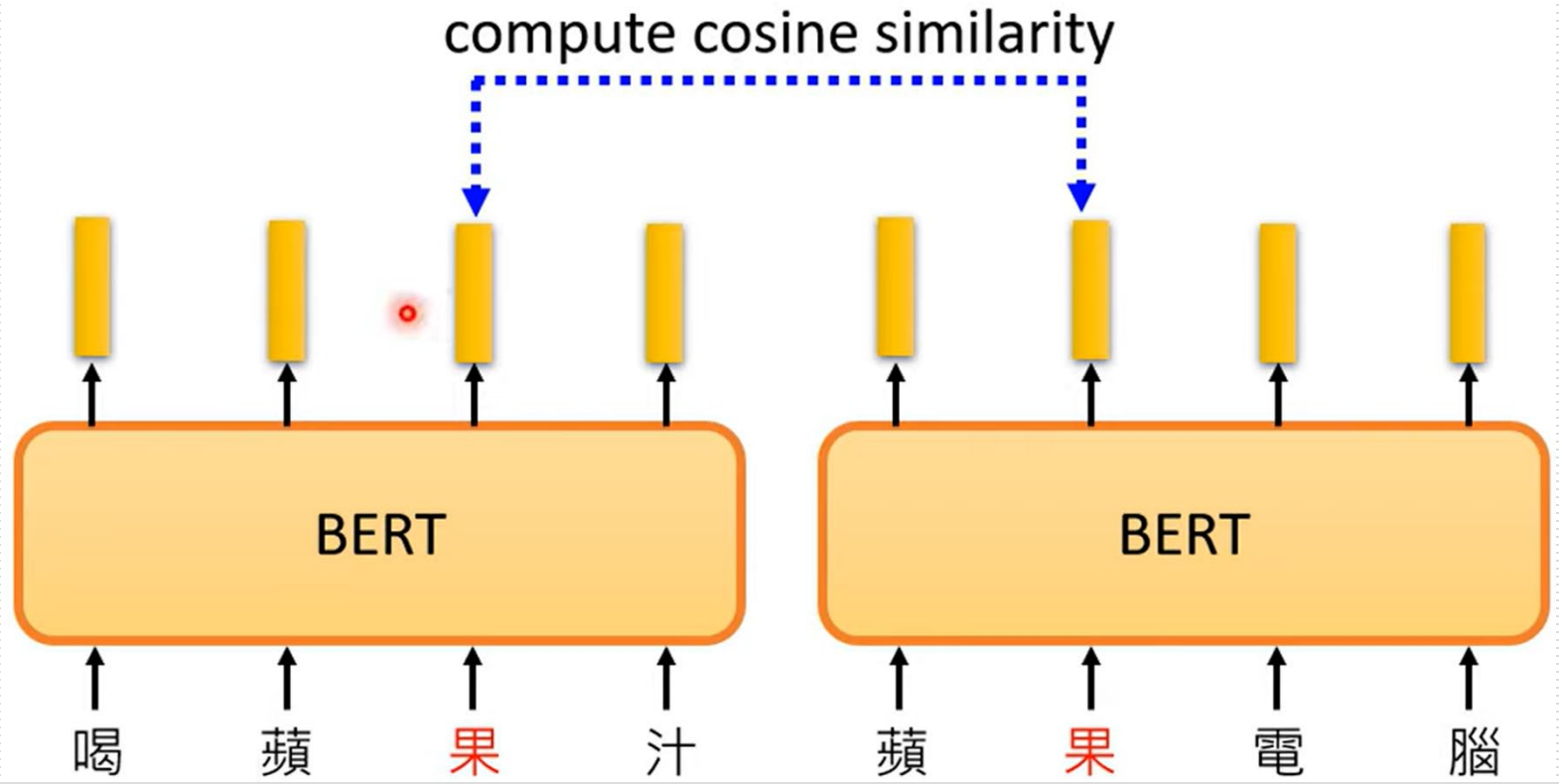


Use Case 4 – Extraction-Based QA (3/3)

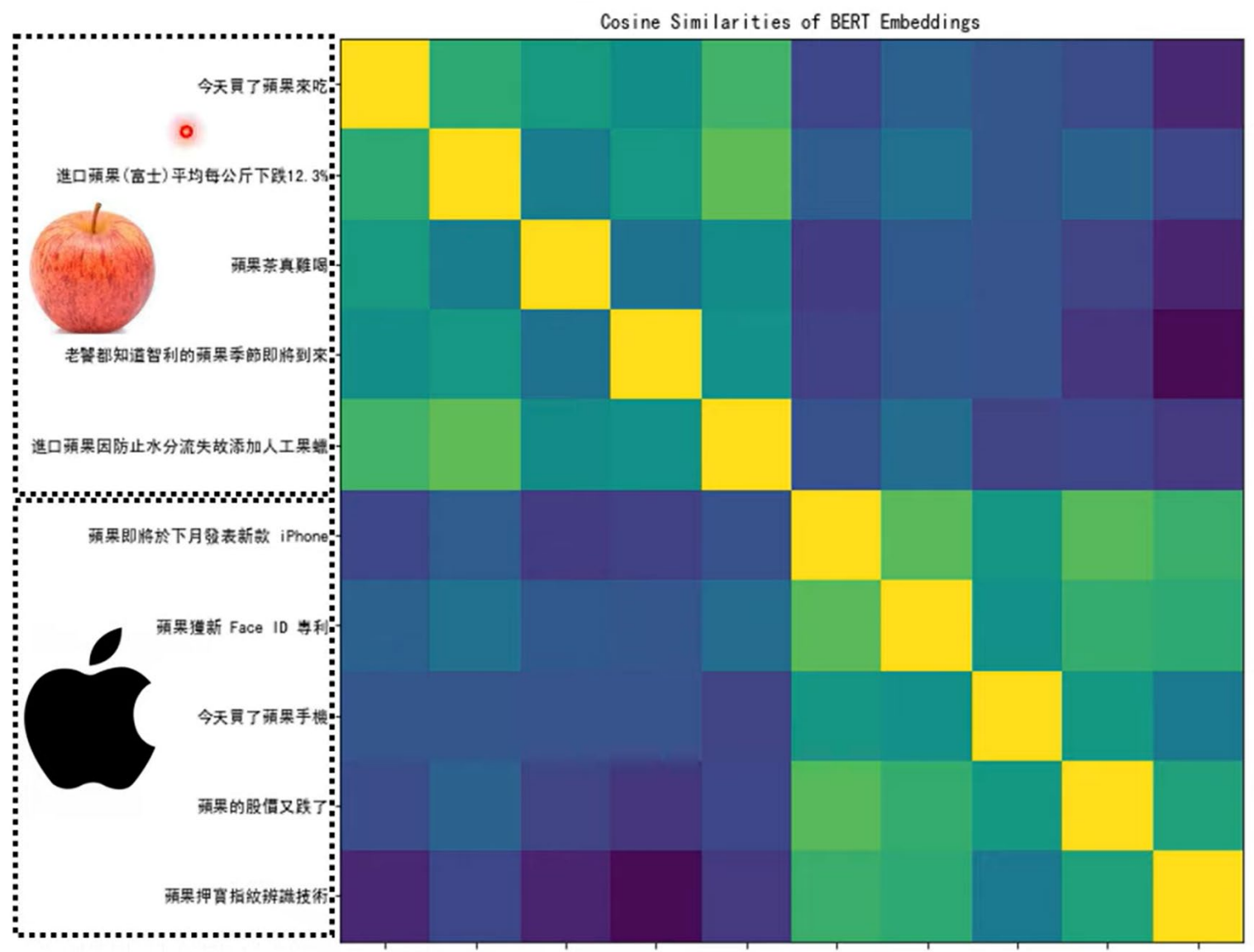


Why Does BERT Work? (1/3)

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$



Why Does BERT Work? (2/3)

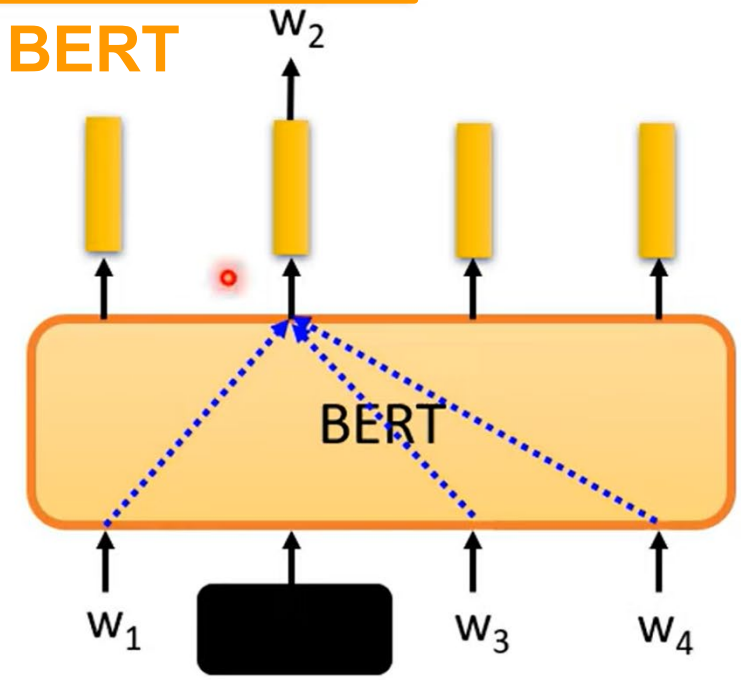


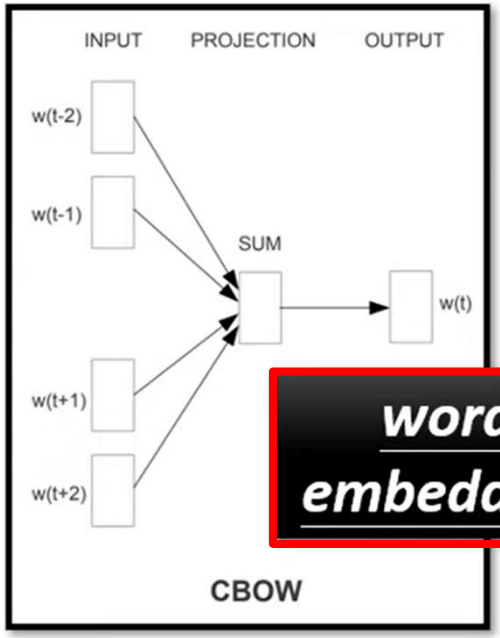
Why Does BERT Work? (3/3)

John Rupert Firth

You shall know a word by the company it keeps

Contextualized word embedding

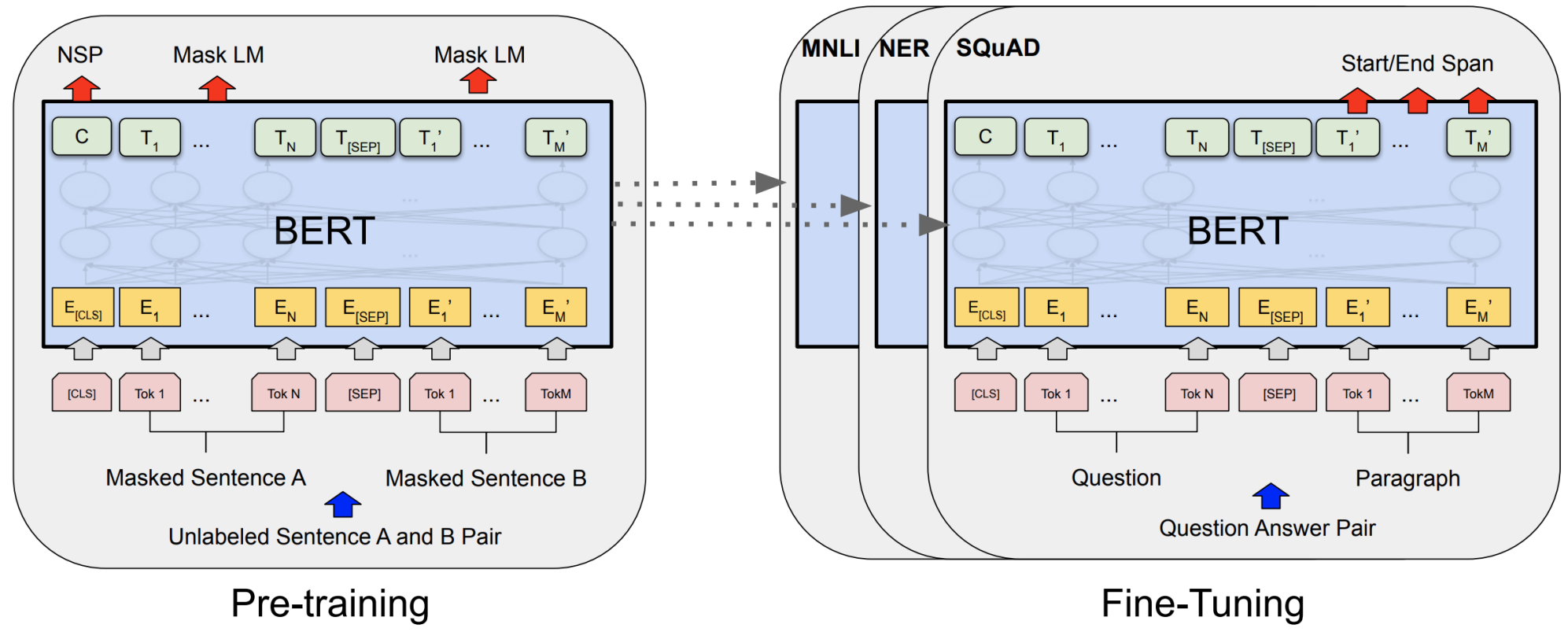




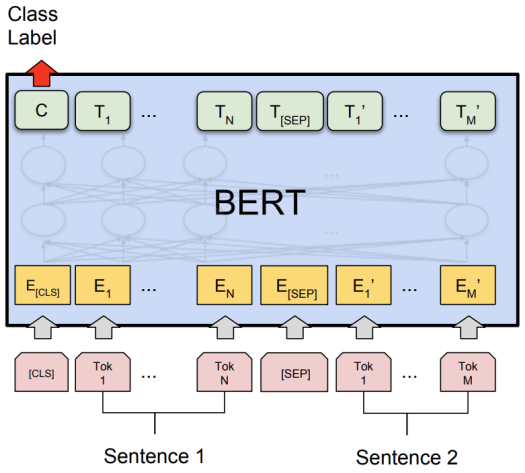
word embedding

Continuous Bag-of-Words (CBOW), 2013

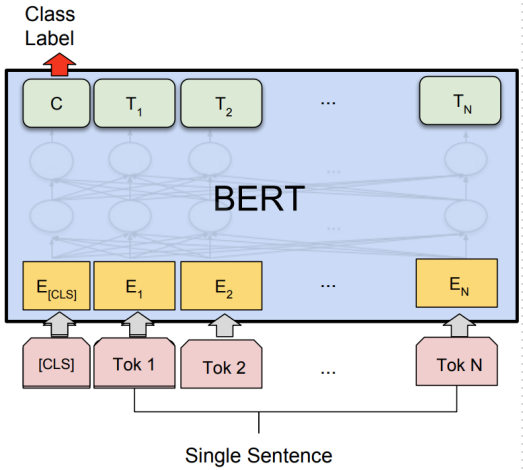
Quick Review (1/3)



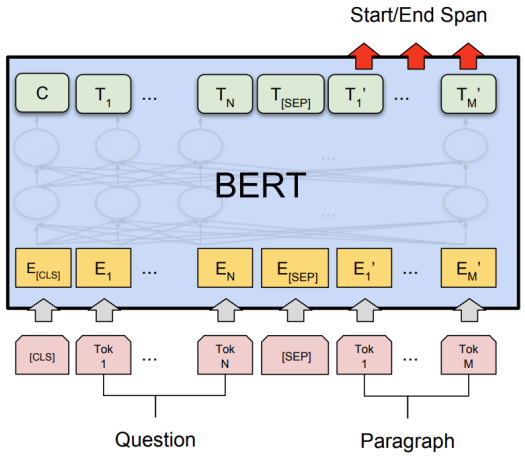
Quick Review (2/3)



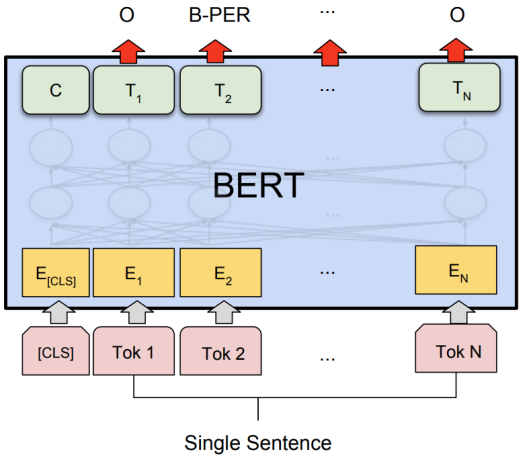
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

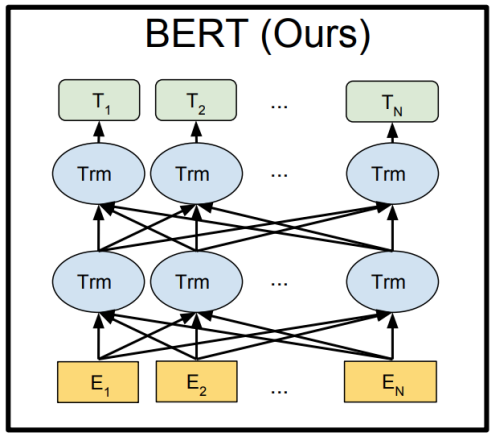


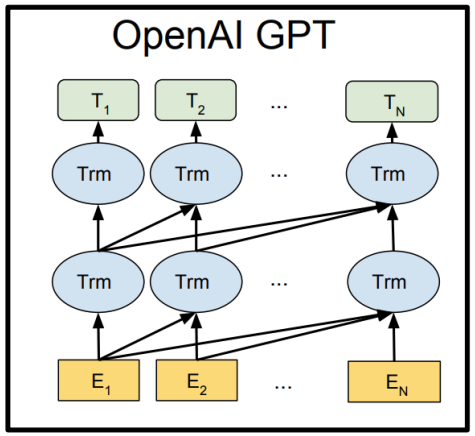
(c) Question Answering Tasks: SQuAD v1.1

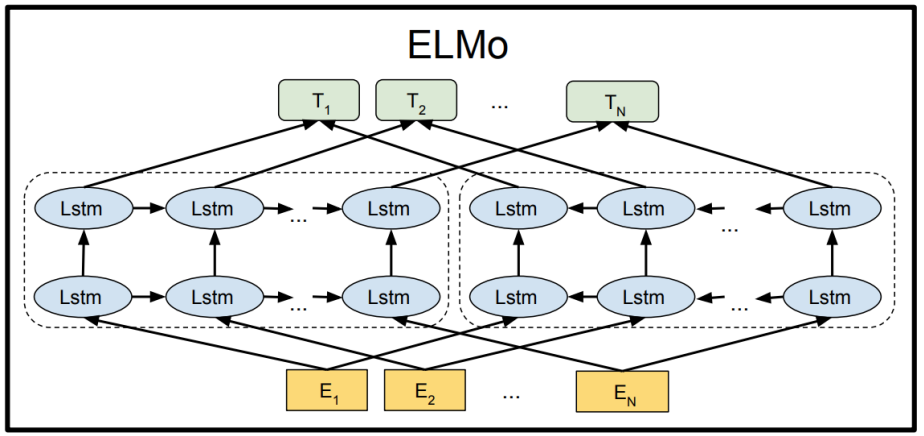


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Quick Review (3/3)







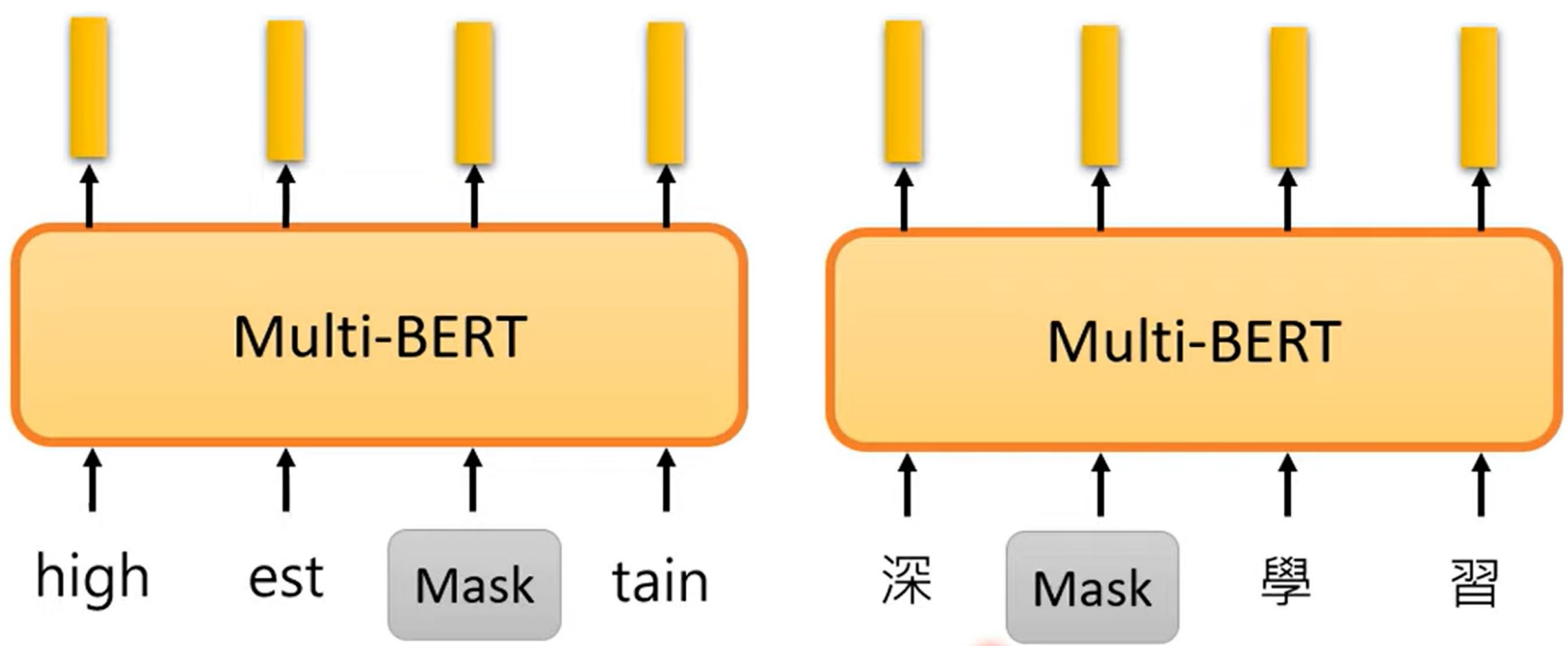
Masked LM (MLM)

Causal LM (CLM)

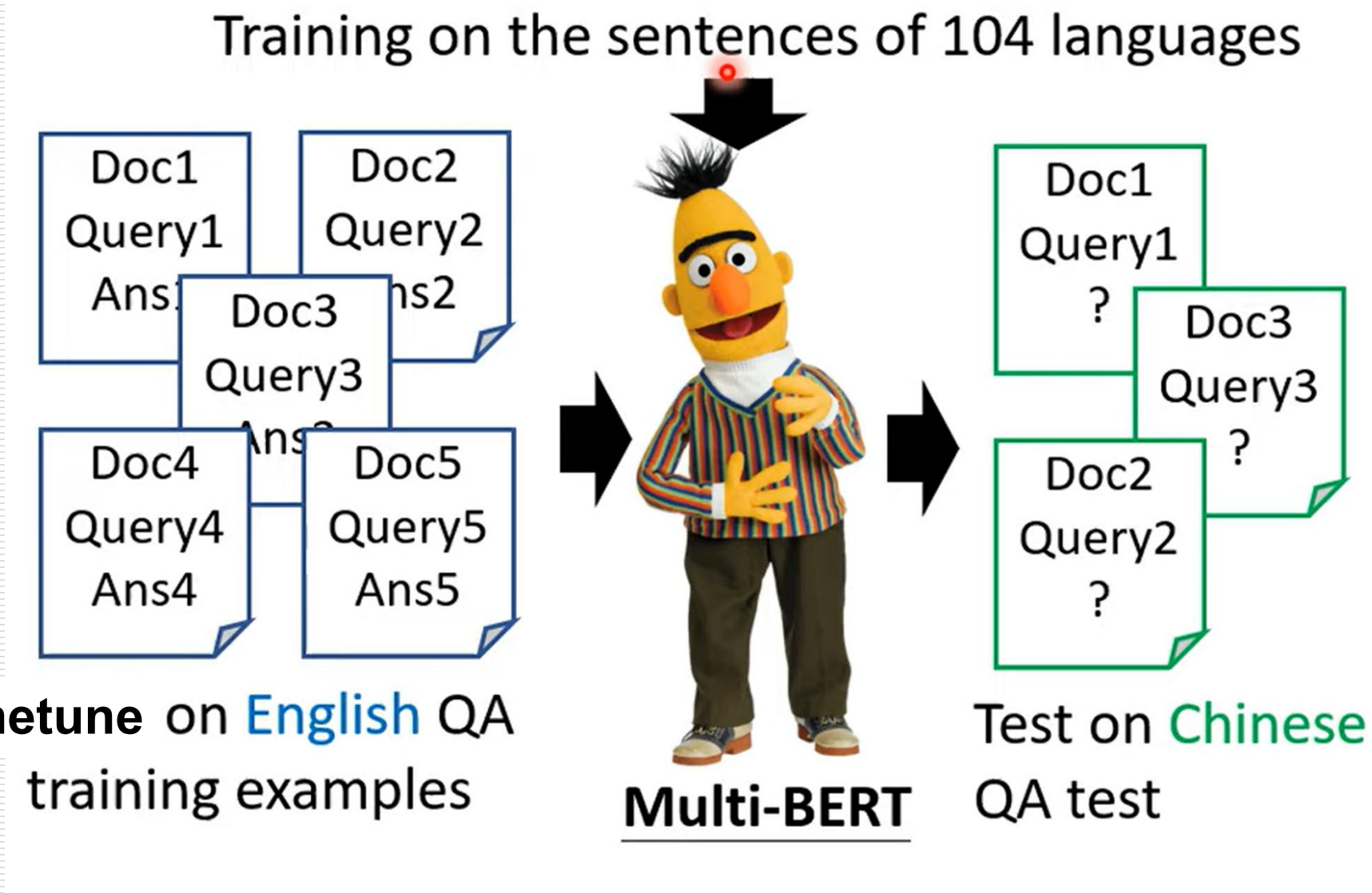
Variants of BERT
and
Friends of BERT

Multilingual BERT

- Train BERT with 104 languages



Zero-Shot Reading Comprehension (1/2)



Zero-Shot Reading Comprehension (2/2)

- English: SQuAD, Chinese: DRCD

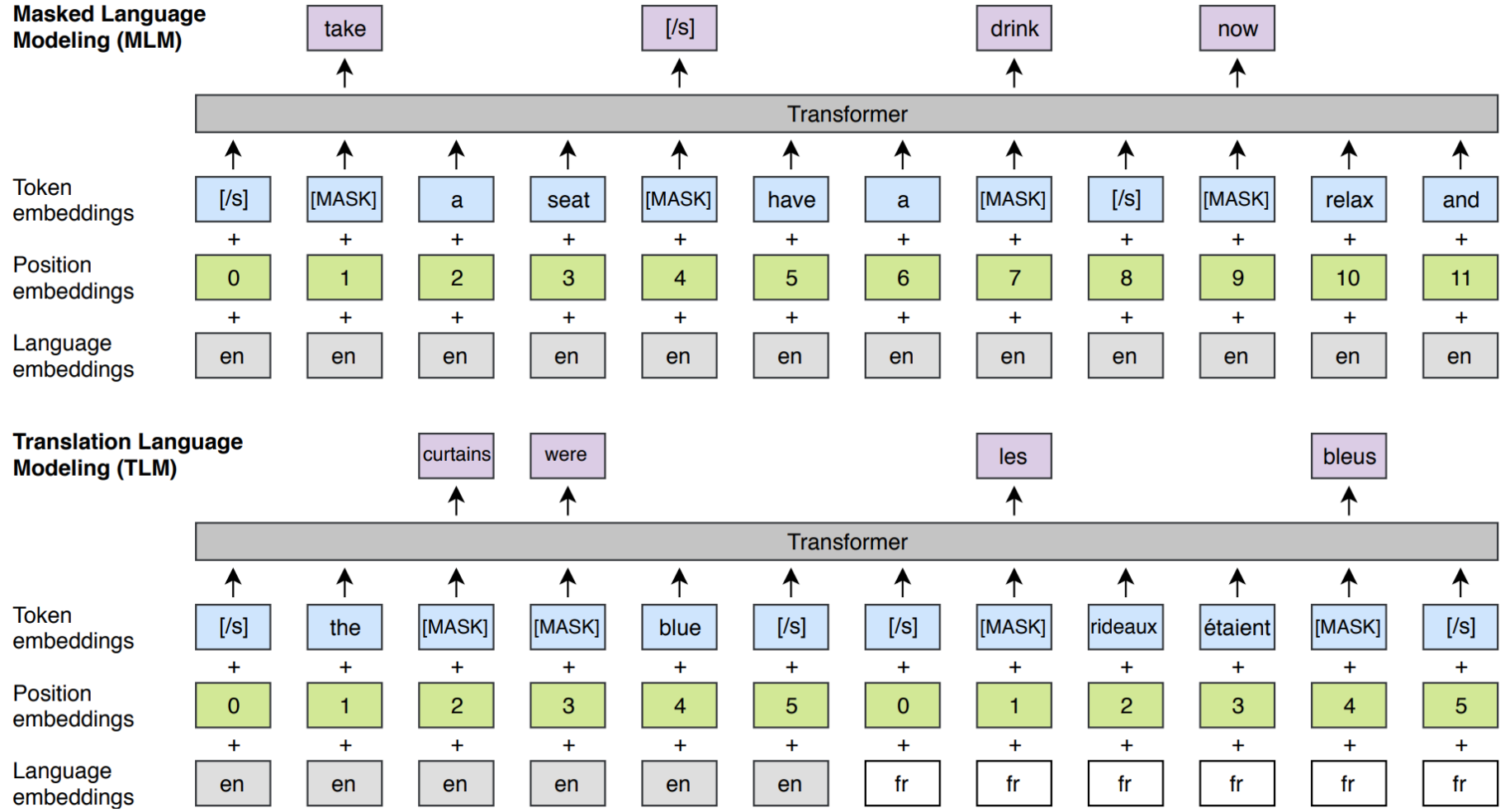
Model	Pre-train	Fine-tune	Test	EM	F1
QANet	none	Chinese	Chinese	66.1	78.1
BERT	Chinese	Chinese		82.0	89.1
	104 languages	Chinese		81.2	88.7
		English		63.3	78.8
		Chinese + English		82.6	90.1

F1 score of Human performance is 93.30%

This work is done by 劉記良、許宗嫄
https://arxiv.org/abs/1909.09587

XLM – Cross-Lingual Language Model

- "Cross-lingual Language Model Pretraining," arXiv'19, NIPS'19, by Facebook



Economic BERT-Based Models

- DistilBERT, NIPS'19, Hugging Face
- TinyBERT, arXiv'19, Huawei Noah's Ark
- MobileBERT, ACL'20, CMU + Google
- **Q8**BERT, **Quantized 8-bit** BERT, NIPS'19, Intel
- **ALBERT**, **A L**ite **BERT**, Google
  - arXiv'19, ICLR'20

Poor People's BERT

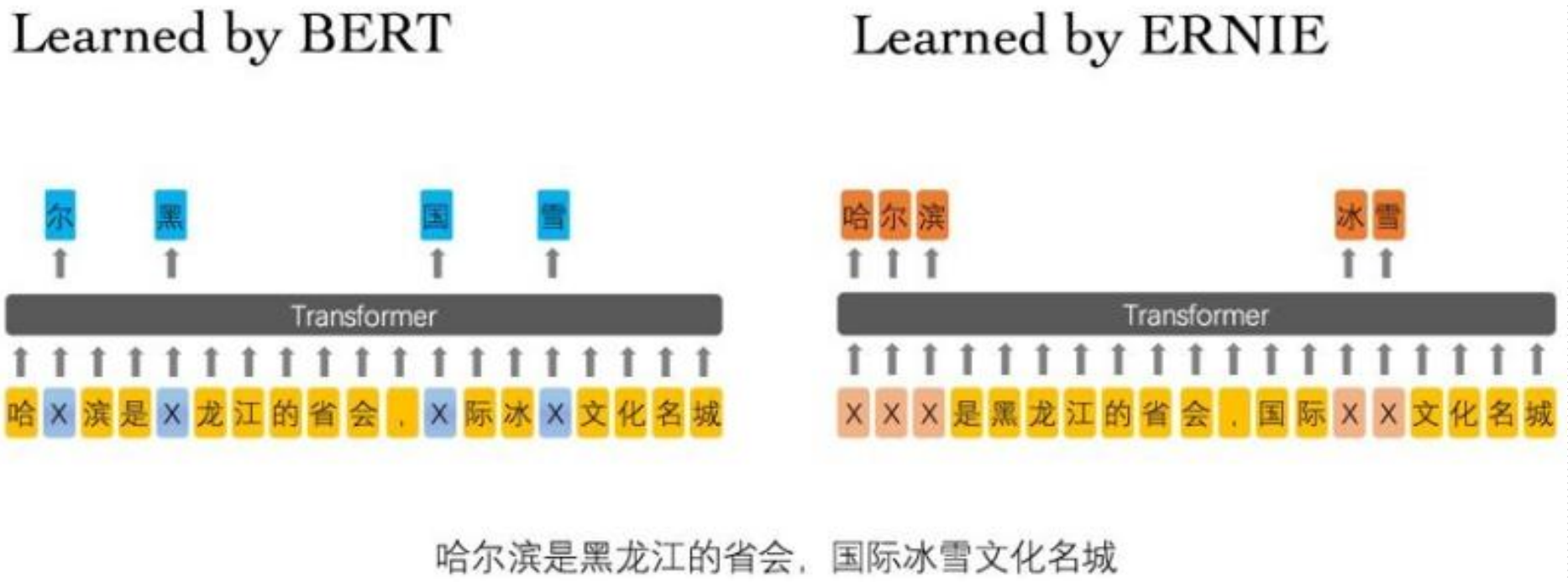
RoBERTa – A Well-Trained BERT

- RoBERTa, Facebook, 2019
  A **R**obustly **O**ptimized **BERT** Pretraining **A**pproach
  - the original BERT was significantly **under-trained**
  - bigger batch size and longer training time in pretraining
  - a larger training dataset (10x) in pretraining
  - pretraining in longer sentences
  - no NSP in pretraining

BERTの逆襲!!

ERNIE

- **ERNIE**: (年輕人，1 2 3 4 6 7 8)
  Enhanced **R**epresentation through
  K**n**owledge Int**e**gration, by 百度, 2019

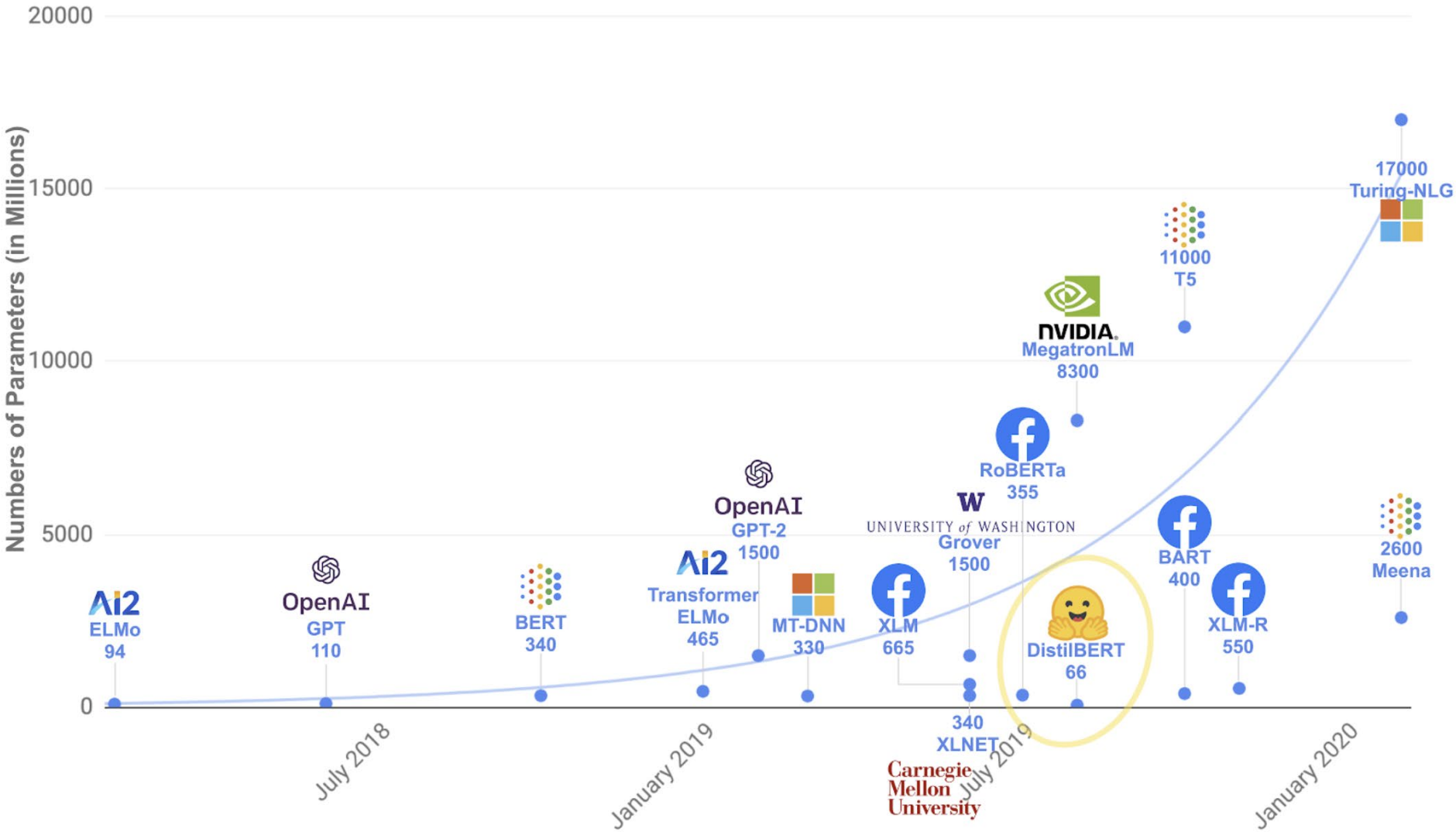


Grover and Big Bird

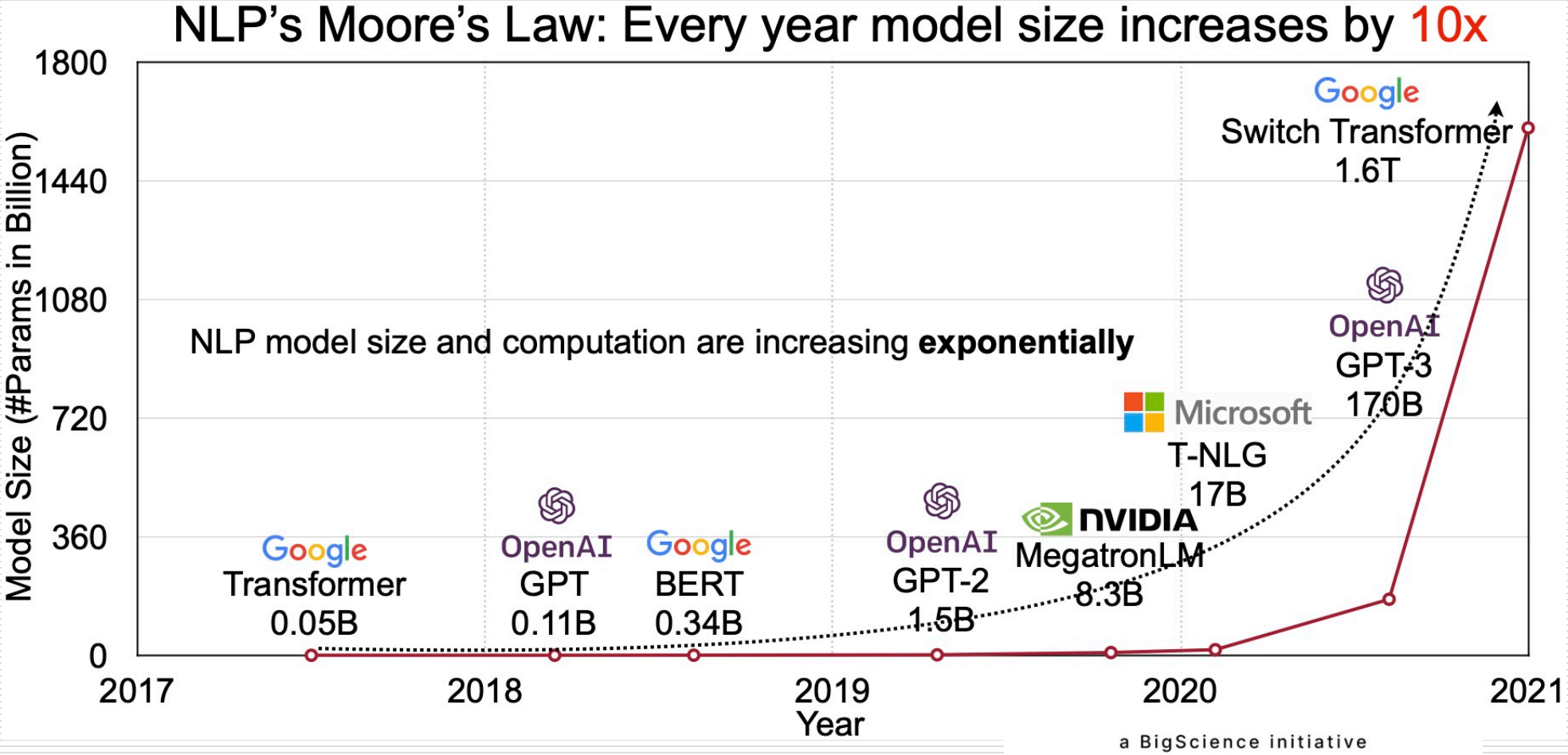
- **Grover**:
  Generating a**R**ticles by **O**nly **V**iewing m**E**tadata **R**ecords
  "Defending Against Neural Fake News,"
  Allen Institute for Artificial Intelligence & Univ. of Washington, 2019

- Big Bird: Transformers for Longer Sequences, Google, 2020
  (放棄治療)

Model Size (1/3)



Model Size (2/3)



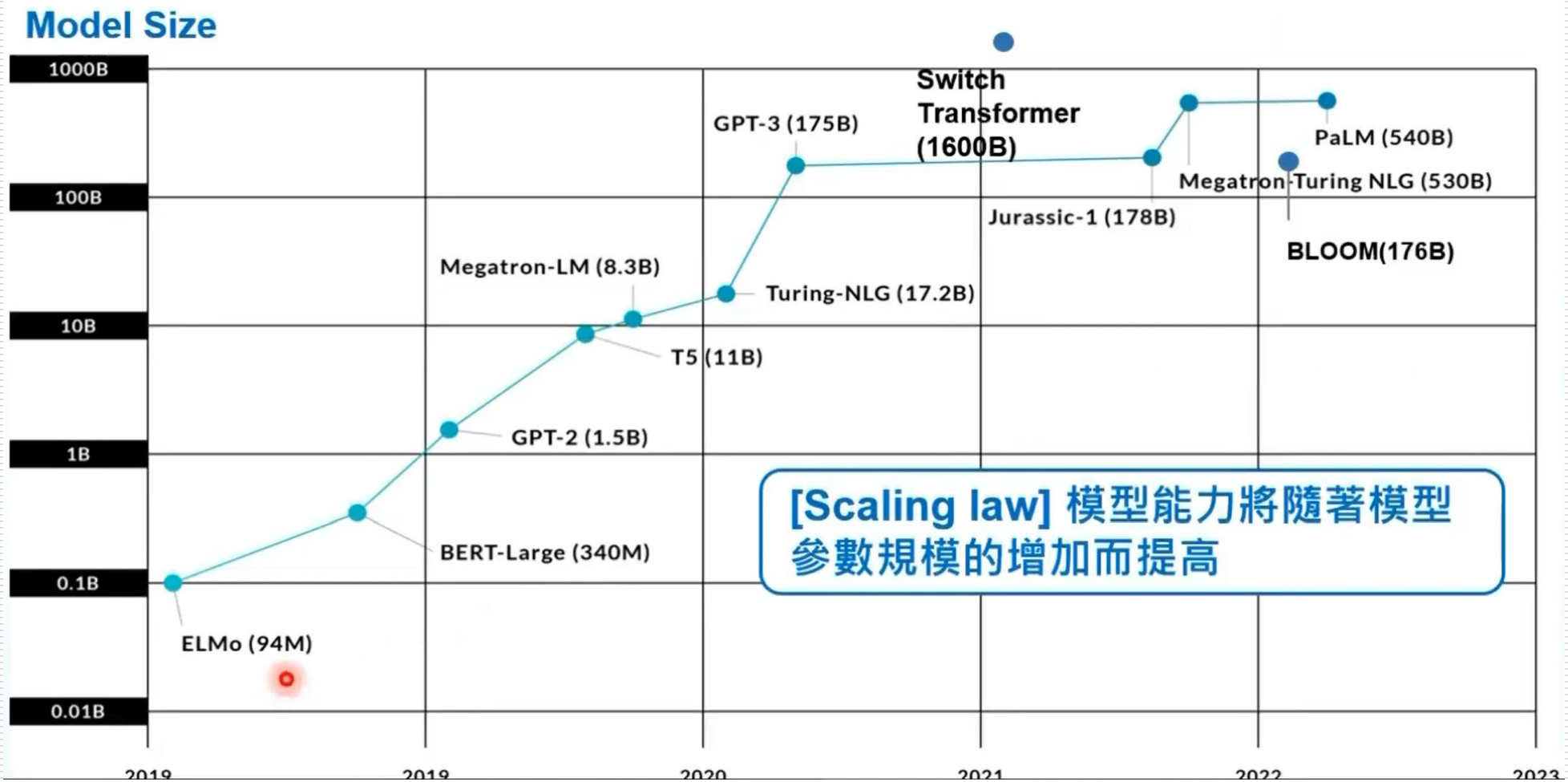
Hugging Face's BLOOM LM
BigScience Large Open-science Open-access Multilingual Language Model, 2022

a BigScience initiative

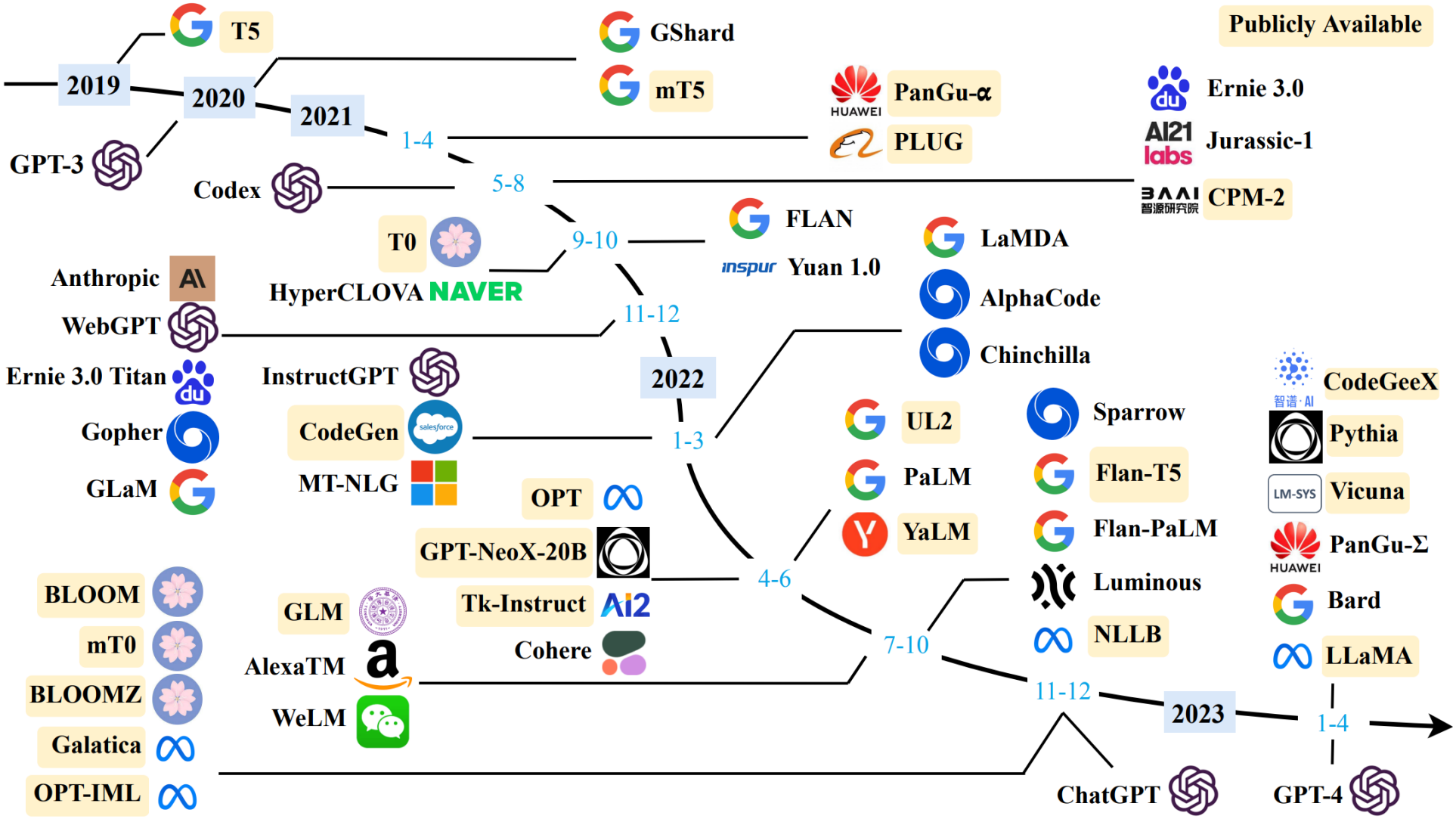
BLOOM

176B params · 59 languages · Open-access

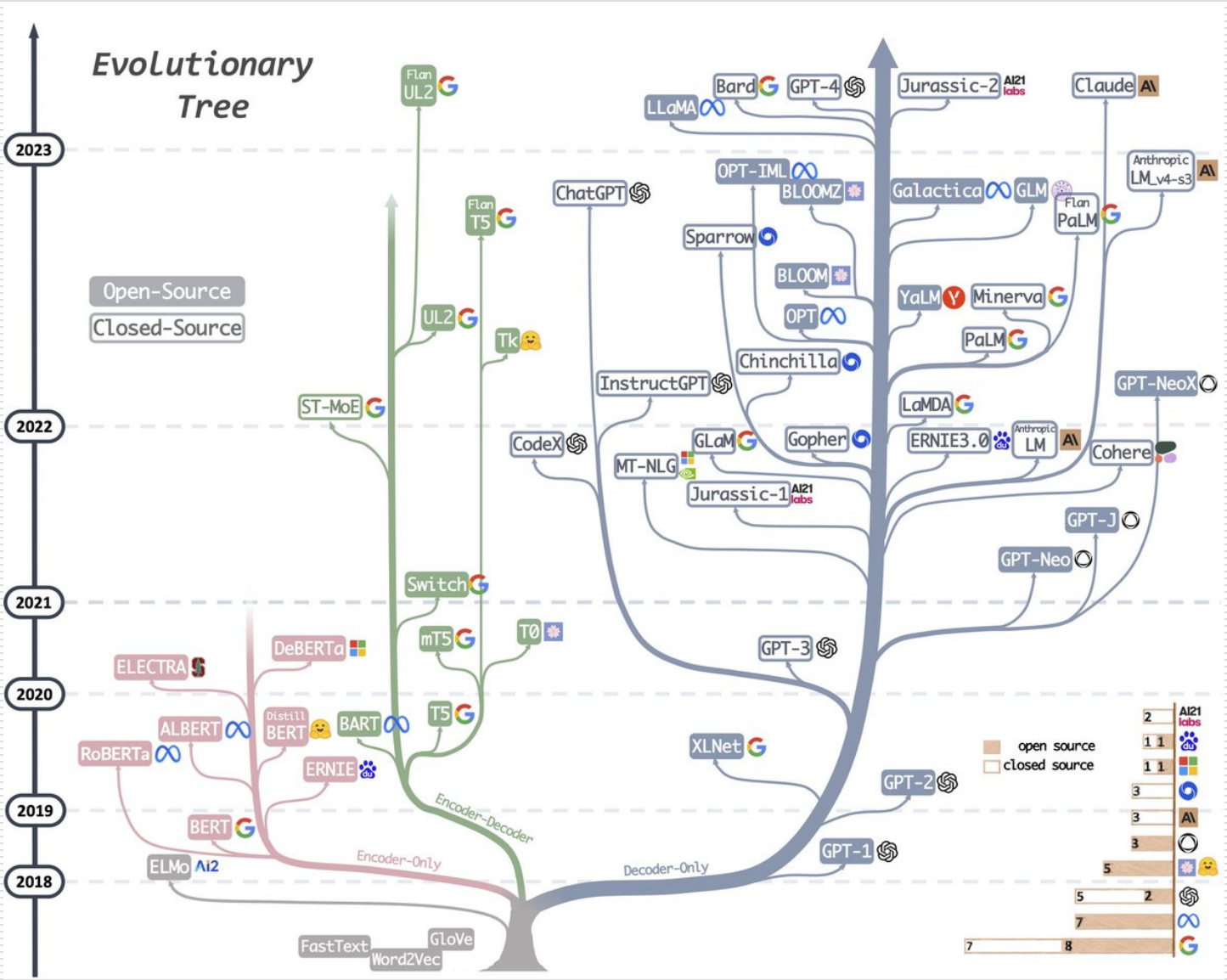
Model Size (3/3)



LLMs (Model Size > 10B)



LLM Evolutionary Tree by Y. LeCun



Takeaways

- **BERT**:
  Bidirectional **Encoder** Representations from **Transformers**
- **Pretrain-then-Finetune** methodology
- **Self-supervised learning** in pretraining phase
- Masked language modeling (**MLM**) is used in pretraining
- **Transfer learning** in finetuning phase
- Know famous variants and friends of BERT

More About BERT (1/2)

- 台大李宏毅教授的解說影片
  - https://youtu.be/e422eloJ0W4，【機器學習2021】自督導式學習 (一)
  - https://youtu.be/gh0hewYkjgo，【機器學習2021】自督導式學習 (二)
  - https://youtu.be/ExXA05i8DEQ，【機器學習2021】自督導式學習 (三)
  - https://youtu.be/WY_E0Sd4K80 ，【機器學習2021】自督導式學習 (四)
  - https://youtu.be/1_gRK9EIQpc，【DLHLP 2020】BERT and its family (1)
  - https://youtu.be/ Bywo7m6ySIk，【DLHLP 2020】BERT and its family(2)
  - https://youtu.be/8rDN1jUI82g，【DLHLP 2020】Multilingual BERT
  - https://youtu.be/UYPa347-DdE，【ELMO, BERT, GPT】, 2019

More About BERT (2/2)

- 台大李宏毅教授的 ELMo, BERT, GPT 解說文章
  - https://hackmd.io/@shaoeChen/Bky0Cnx7L
- Lee Meng (李孟)：“進擊的BERT：NLP界的巨人之力與**遷移學習**”
  - https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html
- “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” – ArXiv paper
  - https://arxiv.org/abs/1810.04805

Institute of Electronics
National Yang Ming Chiao Tung University
Hsinchu, Taiwan

AI Training Course Series

Transformer-Based Image Classification

Lecture 7-2

Architecture Development & Algorithm Refinement for Intelligent Computing

Student: Meng-Hsun Hsieh
Advisor: Juinn-Dar Huang, Ph.D.

August 5, 2024

Outline

- Review: Vision Transformer (ViT)
- DeiT
- Swin Transformer
- CvT
- CSWin Transformer
- Pale Transformer
- Exercise

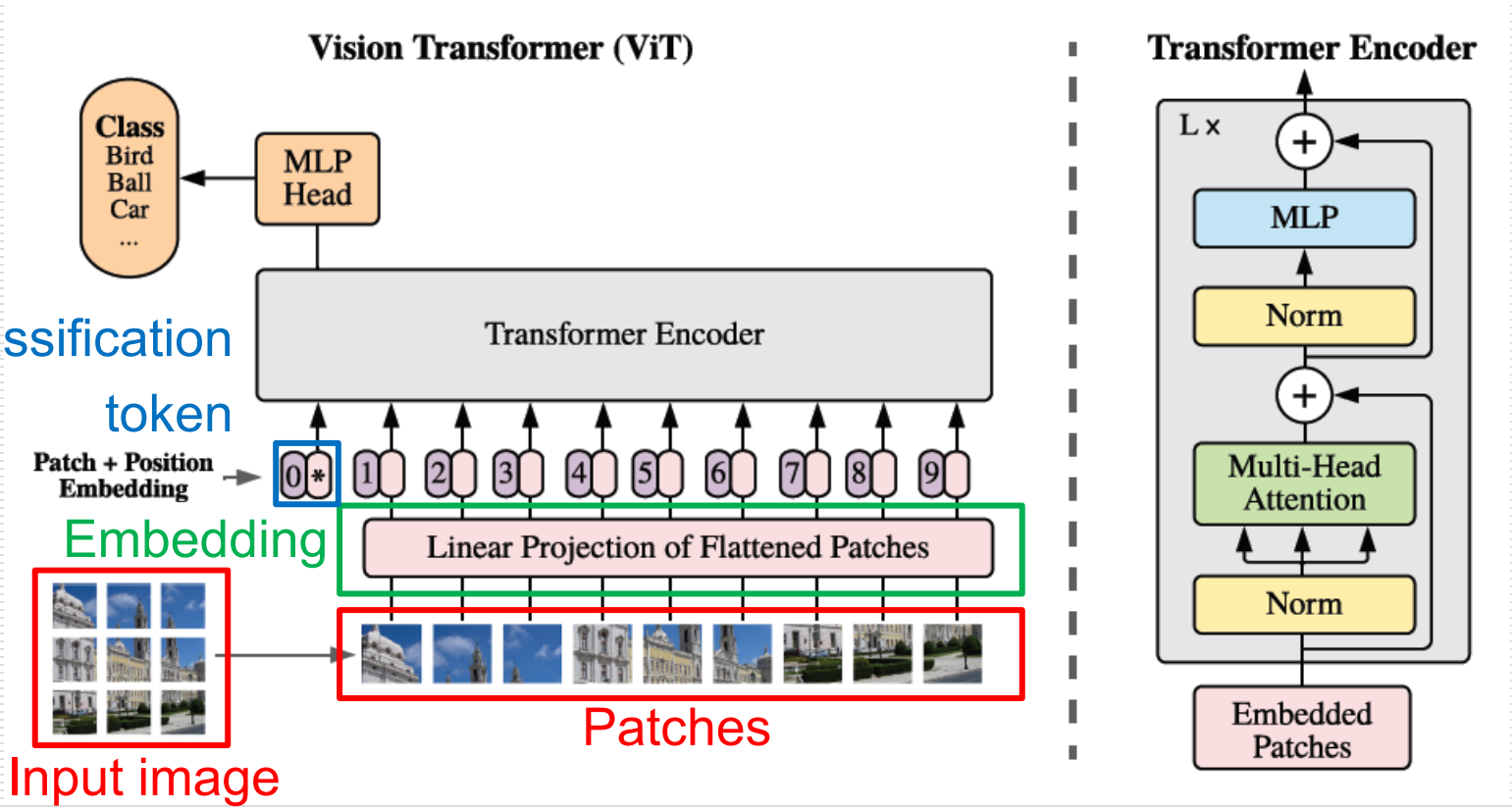
ViT

Google, 2020

(An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale)

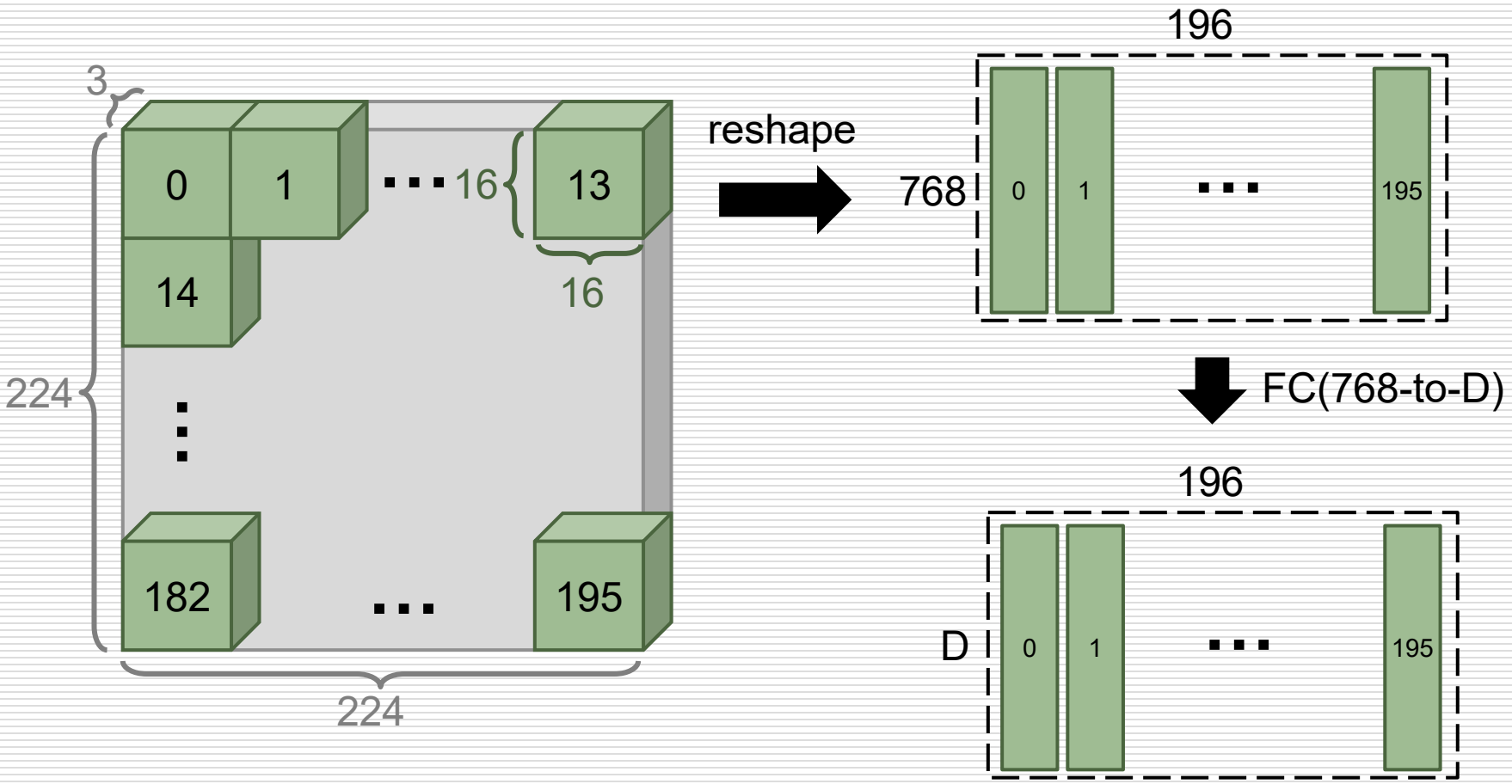
Review: Vision Transformer (ViT)

- Inspired by the success in NLP
  - ➔ applying Transformer directly to images
  - ➔ pure transformer model (no convolution)



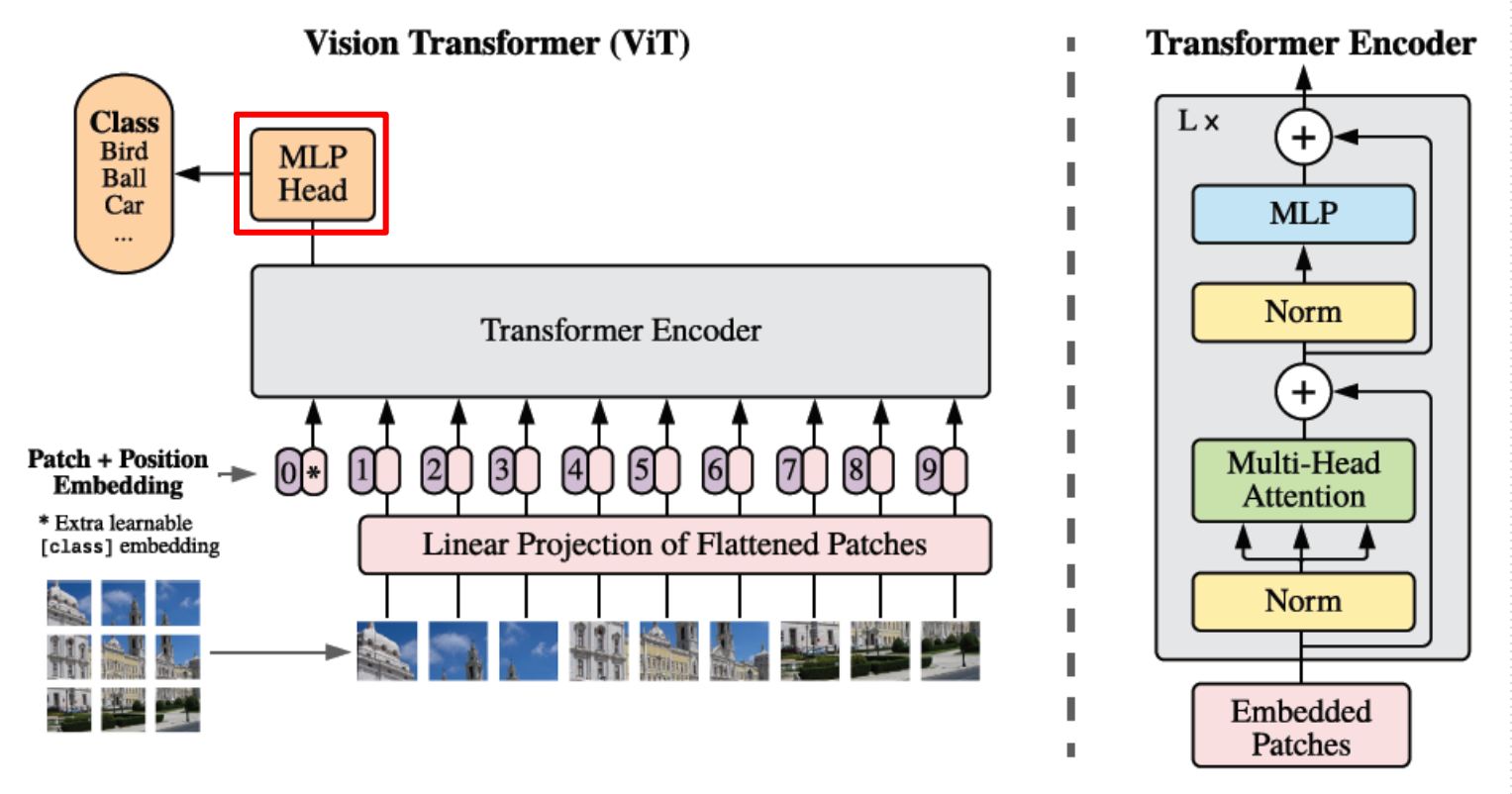
ViT – Patch Embedding

- Split an image into patches (or tokens)
  - D: hidden dimension



ViT – Classifier

- Add an extra learnable “classification token”
- Use MLP Head as classifier
  - FC (D-to-1000) @ ImageNet



ViT – Datasets

- Pre-train (Large datasets)
  - ImageNet: 1.2M images ← most commonly used
  - ImageNet-21k: 14M images
  - JET-300M: 300M images (private, $$$)
- Fine-tune
  - ImageNet
  - ImageNet ReaL
  - CIFAR-10, CIFAR-100
  - Oxford-IIIT Pets
  - Oxford Flowers-102
  - VTAB (19 tasks)

Experiment Results (1/2)

- Pre-train ViT on large datasets, and fine-tune to smaller downstream tasks

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

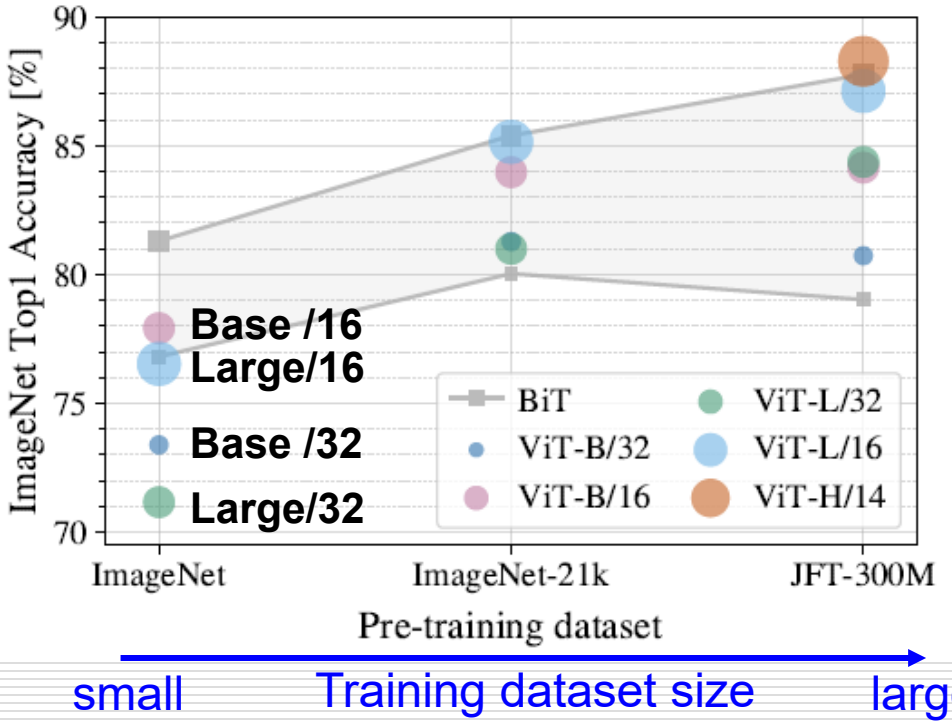
Table 1: Details of Vision Transformer model variants.

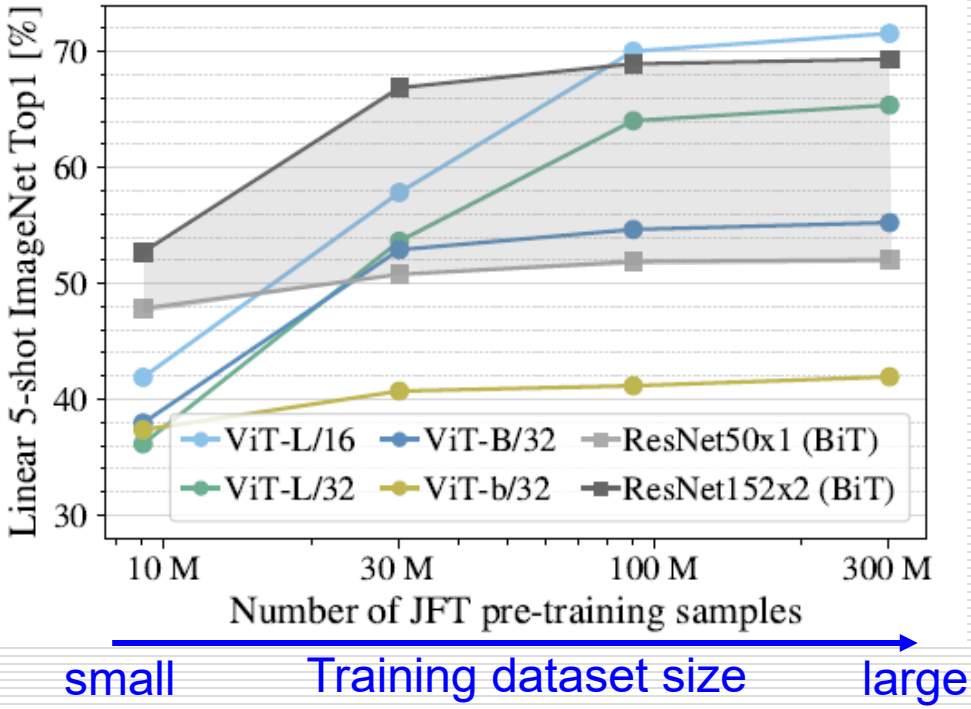
CNN-based model

	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

Experiment Results (2/2)

- For smaller dataset: ResNet > ViT
- For larger dataset: ViT > ResNet





Summary of ViT

- Apply Transformer-based model on vision task

- Drawbacks
  - need huge dataset to train
  - huge model size

  ➔ extensive computing resources for training

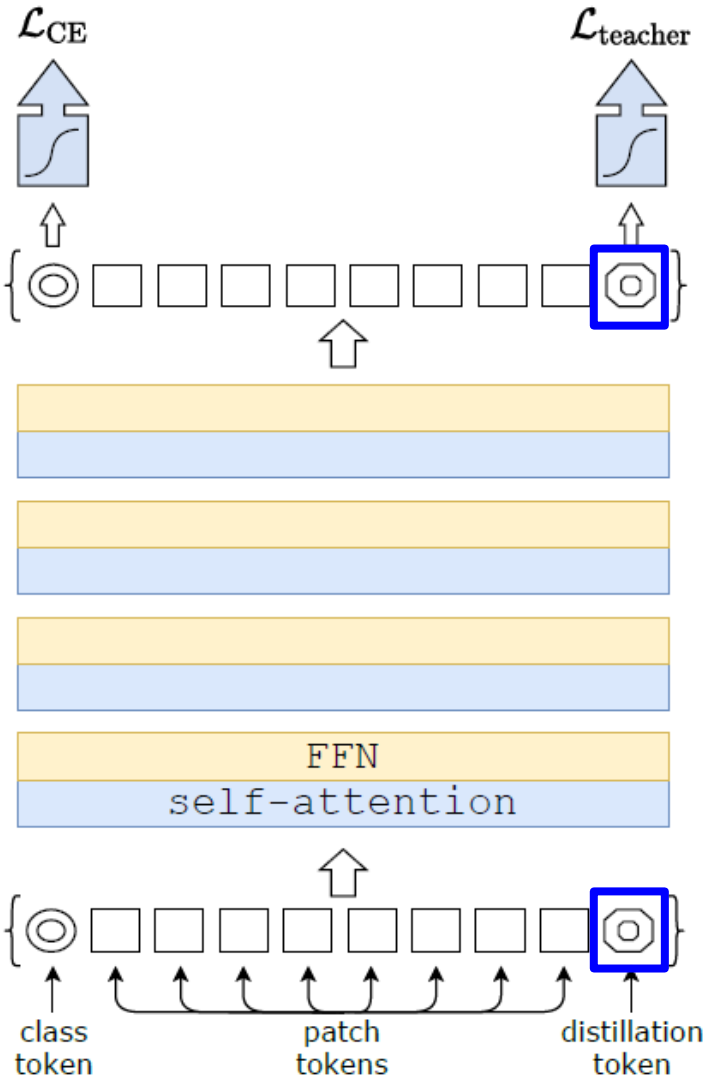
DeiT

Facebook, 2020

(Training Data-efficient Image Transformers & Distillation Through Attention)

DeiT

- Add an extra “distillation token”
  - teacher-student strategy
  - teacher: CNN
  - student: Transformer



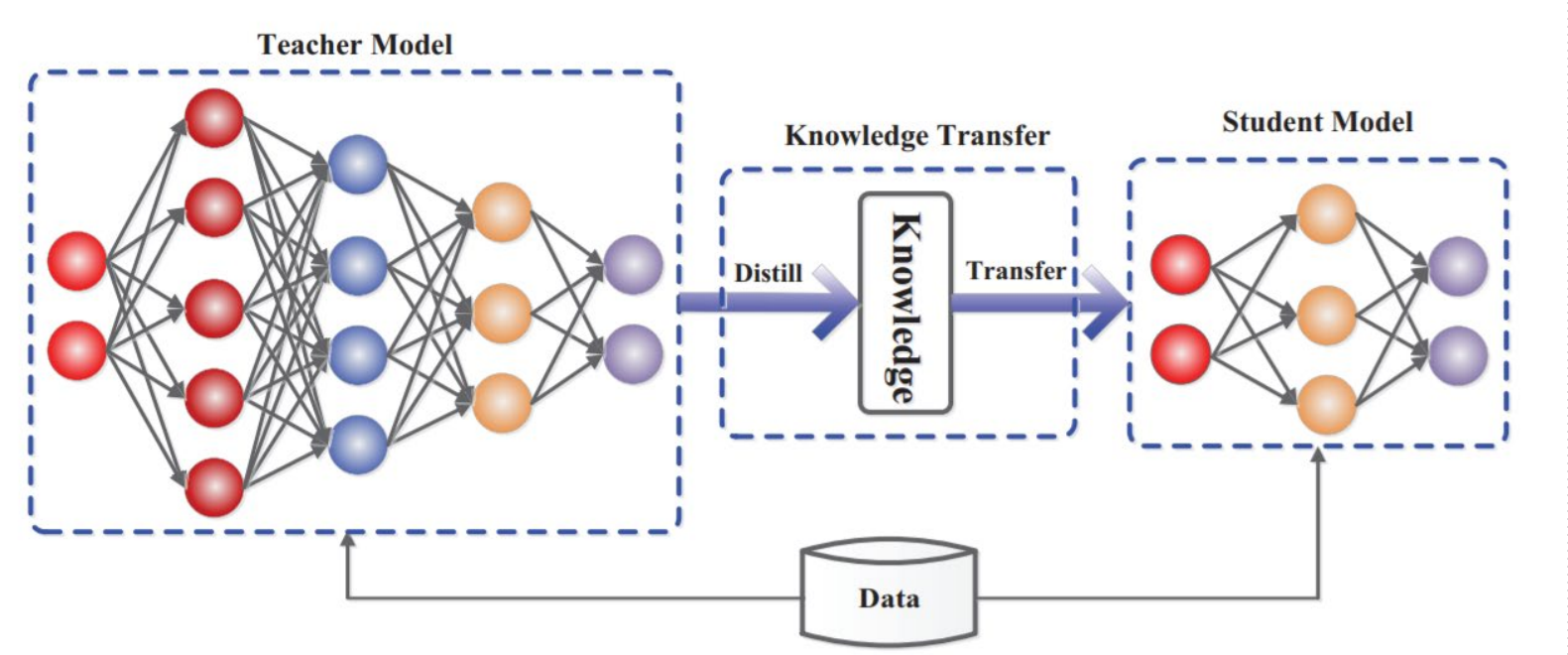
Distillation (1/3)

- Knowledge distillation:

teacher $\xrightarrow{\text{特徵(knowledge)}}$ student

(complex/deep) (simple/shallow)

(CNN) (Transformer)



Distillation (2/3)

- The teacher's supervision takes into account the effects of the data augmentation during training

Ground truth: dog

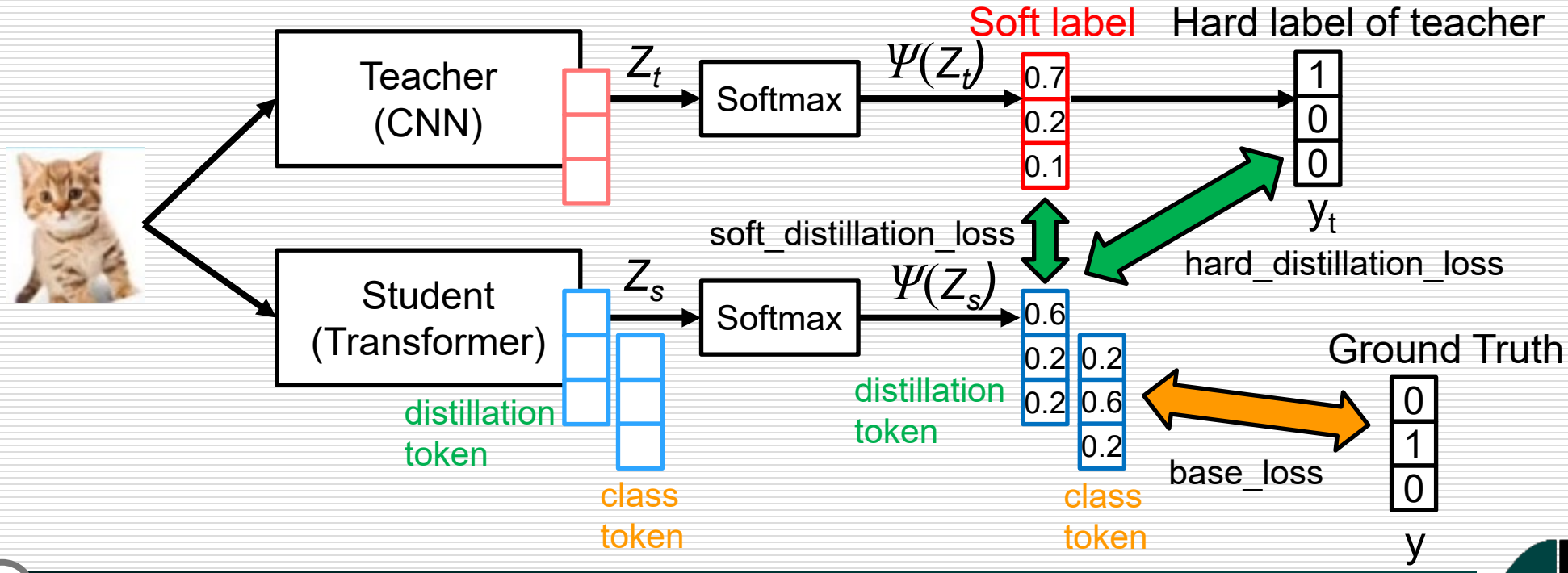
Data Augmentation (crop)

Ground truth: dog 😐

Teacher output: cat ☺
(better label for student model)

Distillation (3/3)

- A ***student*** model get “soft” labels coming from a strong ***teacher*** network
- It can be regarded as a form of compression of the teacher model into a smaller student model



Loss Function

- Soft distillation
  - $\mathcal{L}_{global} = (1-\lambda)\underbrace{\mathcal{L}_{CE}(\psi(Z_s), y)}_{base\_loss} + \underbrace{\lambda\tau^2 KL(\psi(Z_s/\tau), \psi(Z_t/\tau))}_{soft\_distillation\_loss}$
- Hard-label distillation
  - $\mathcal{L}_{global}^{hardDistill} = \frac{1}{2}\underbrace{\mathcal{L}_{CE}(\psi(Z_s), y)}_{base\_loss} + \frac{1}{2}\underbrace{\mathcal{L}_{CE}(\psi(Z_s), y_t)}_{hard\_distillation\_loss}$

Experiment Results (1/3)

- DeiT-(Ti/S/B)
  - (Tiny/Small/Base)
  - teacher: RegNetY-16GF (Facebook 2020)
- DeiT-B has the same architecture as the ViT-B

Model	ViT model	embedding dimension	#heads	#layers	#params	training resolution	throughput (im/sec)
DeiT-Ti	N/A	192	3	12	5M	224	2536
DeiT-S	N/A	384	6	12	22M	224	940
DeiT-B	ViT-B	768	12	12	86M	224	292

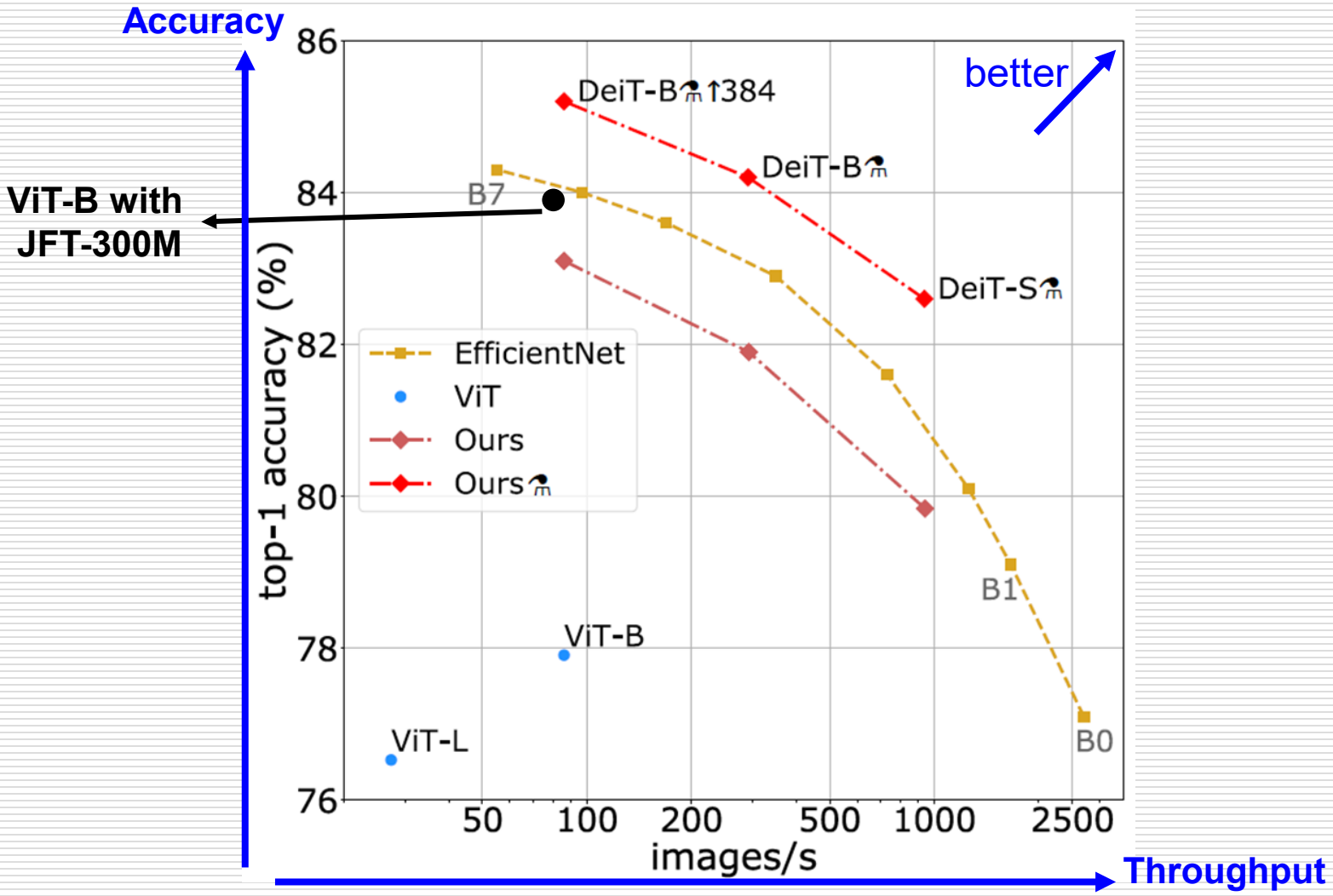
Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Experiment Results (2/3)

Network	#param.	image size	throughput (image/s)	ImNet top-1	Real top-1	V2 top-1
Transformers						
ViT-B/16 [15]	86M	384^2	85.9	77.9	83.6	-
ViT-L/16 [15]	307M	384^2	27.3	76.5	82.2	-
DeiT-Ti	5M	224^2	2536.5	72.2	80.1	60.4
DeiT-S	22M	224^2	940.4	79.8	85.7	68.5
DeiT-B	86M	224^2	292.3	81.8	86.7	71.5
DeiT-B↑384	86M	384^2	85.9	83.1	87.7	72.4
DeiT-Ti⚗	6M	224^2	2529.5	74.5	82.1	62.9
DeiT-S⚗	22M	224^2	936.2	81.2	86.8	70.0
DeiT-B⚗	87M	224^2	290.9	83.4	88.3	73.2
DeiT-Ti⚗ / 1000 epochs	6M	224^2	2529.5	76.6	83.9	65.4
DeiT-S⚗ / 1000 epochs	22M	224^2	936.2	82.6	87.8	71.7
DeiT-B⚗ / 1000 epochs	87M	224^2	290.9	84.2	88.7	73.9
DeiT-B⚗ ↑384	87M	384^2	85.8	84.5	89.0	74.8
DeiT-B⚗ ↑384 / 1000 epochs	87M	384^2	85.8	85.2	89.3	75.2

Smaller, Faster, Higher Accuracy!

Experiment Results (3/3)



Summary of DeiT

- Using better training procedure
  - distillation, data augmentation
- Benefits
  - smaller model size
  - lower training cost (only ImageNet-1K)
  - better performance

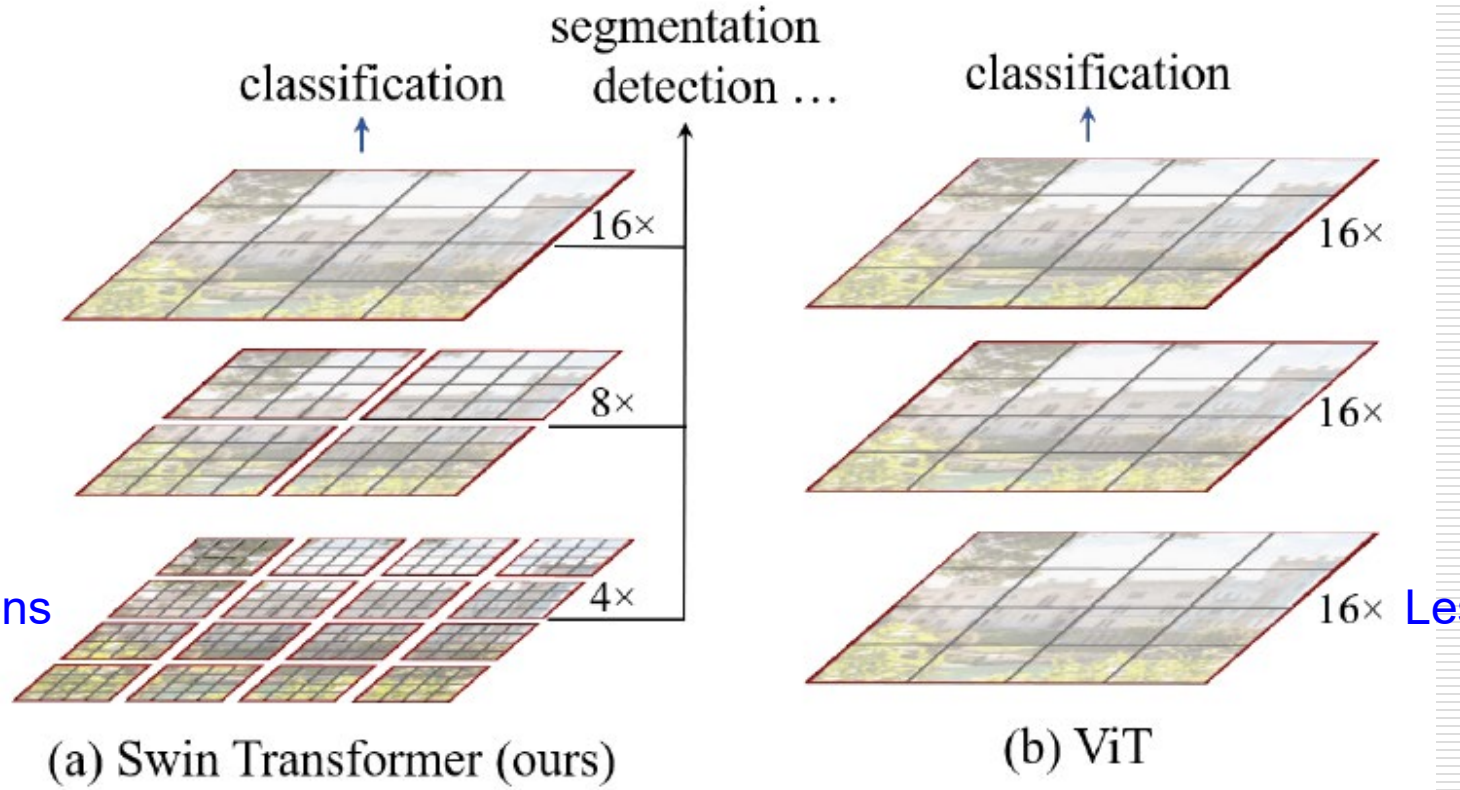
Swin Transformer

Microsoft, 2021

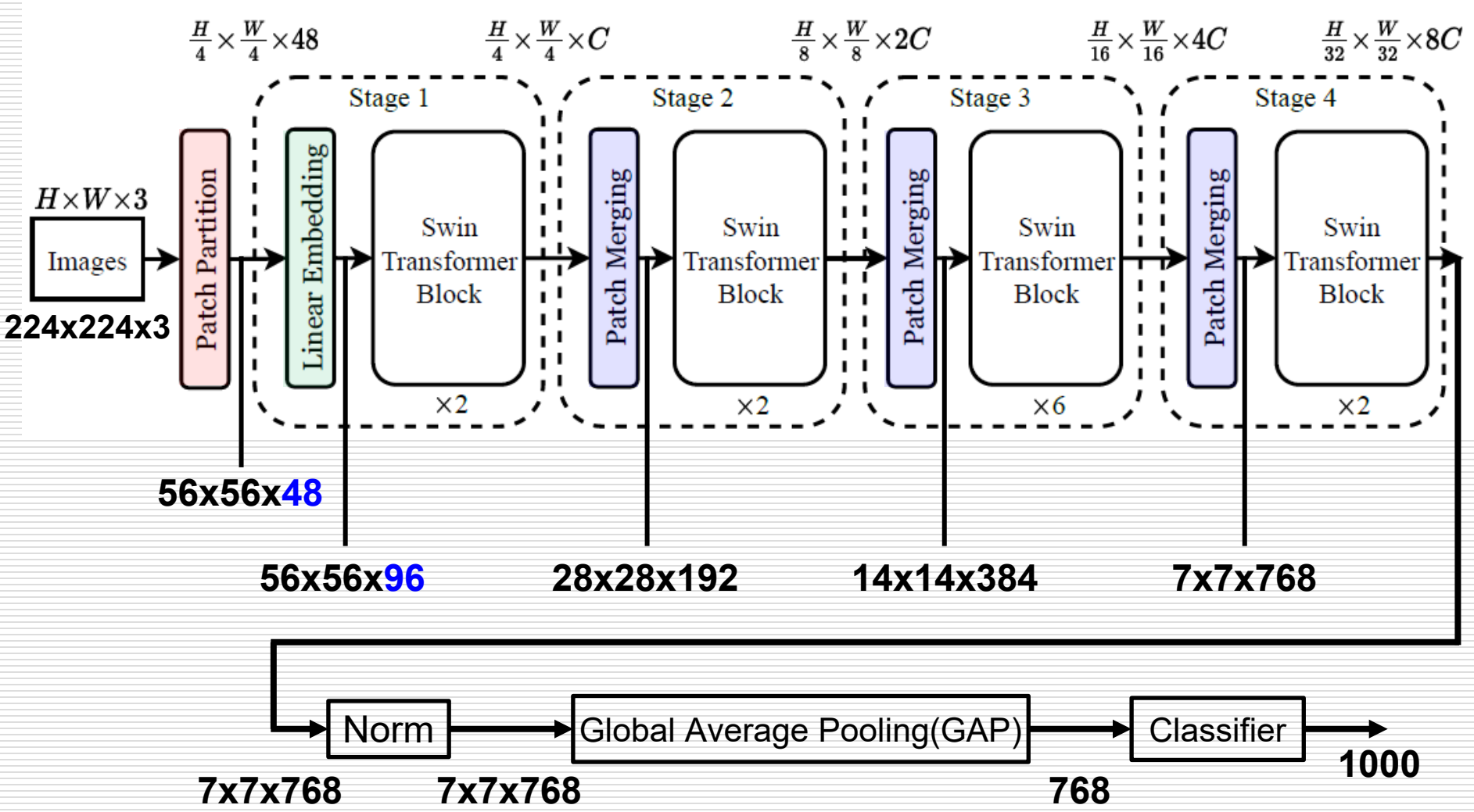
(Swin Transformer: Hierarchical Vision Transformer using Shifted Windows)

Swin

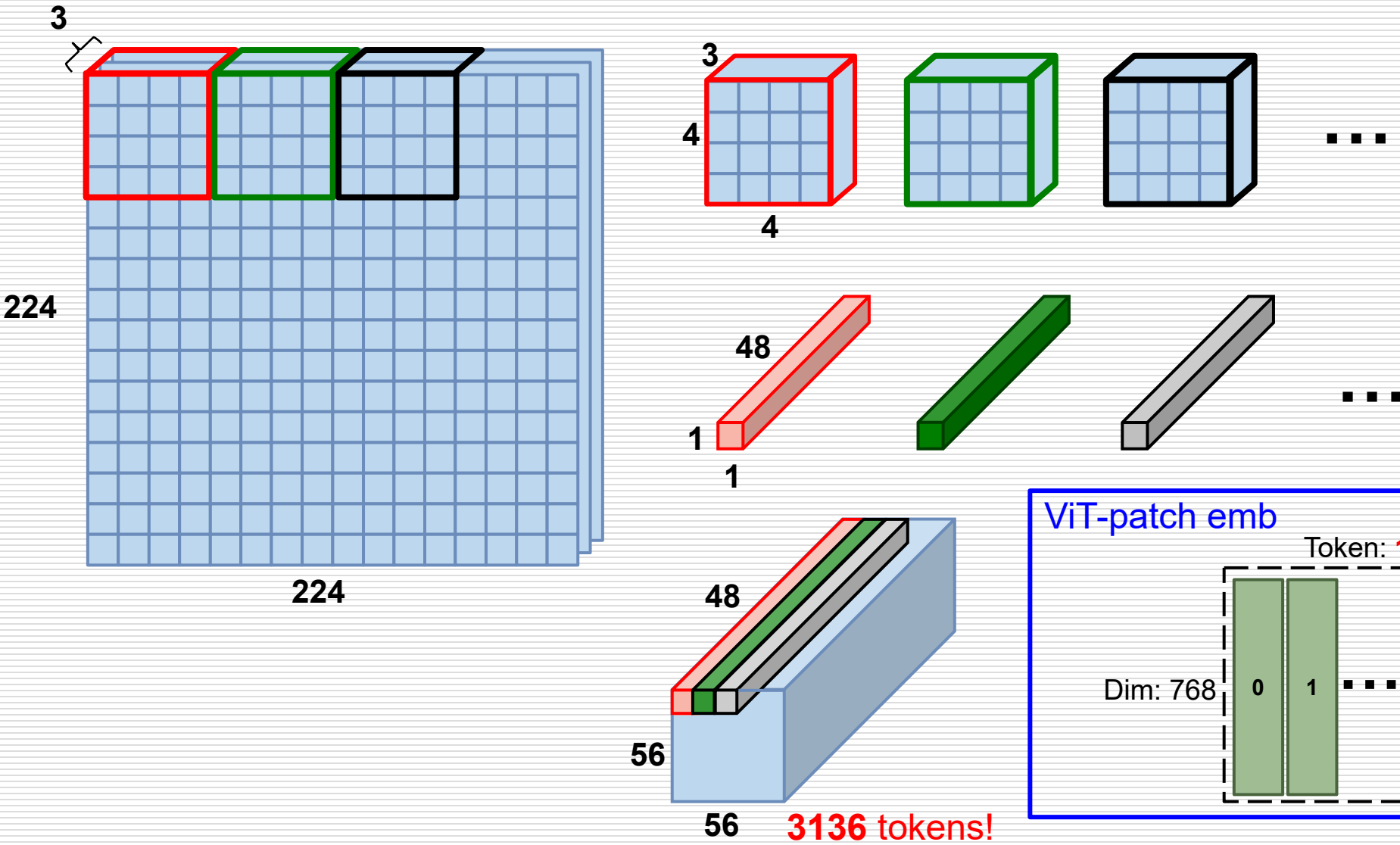
- Building hierarchical feature maps
  - more tokens in early stage
  - merging tokens between stages



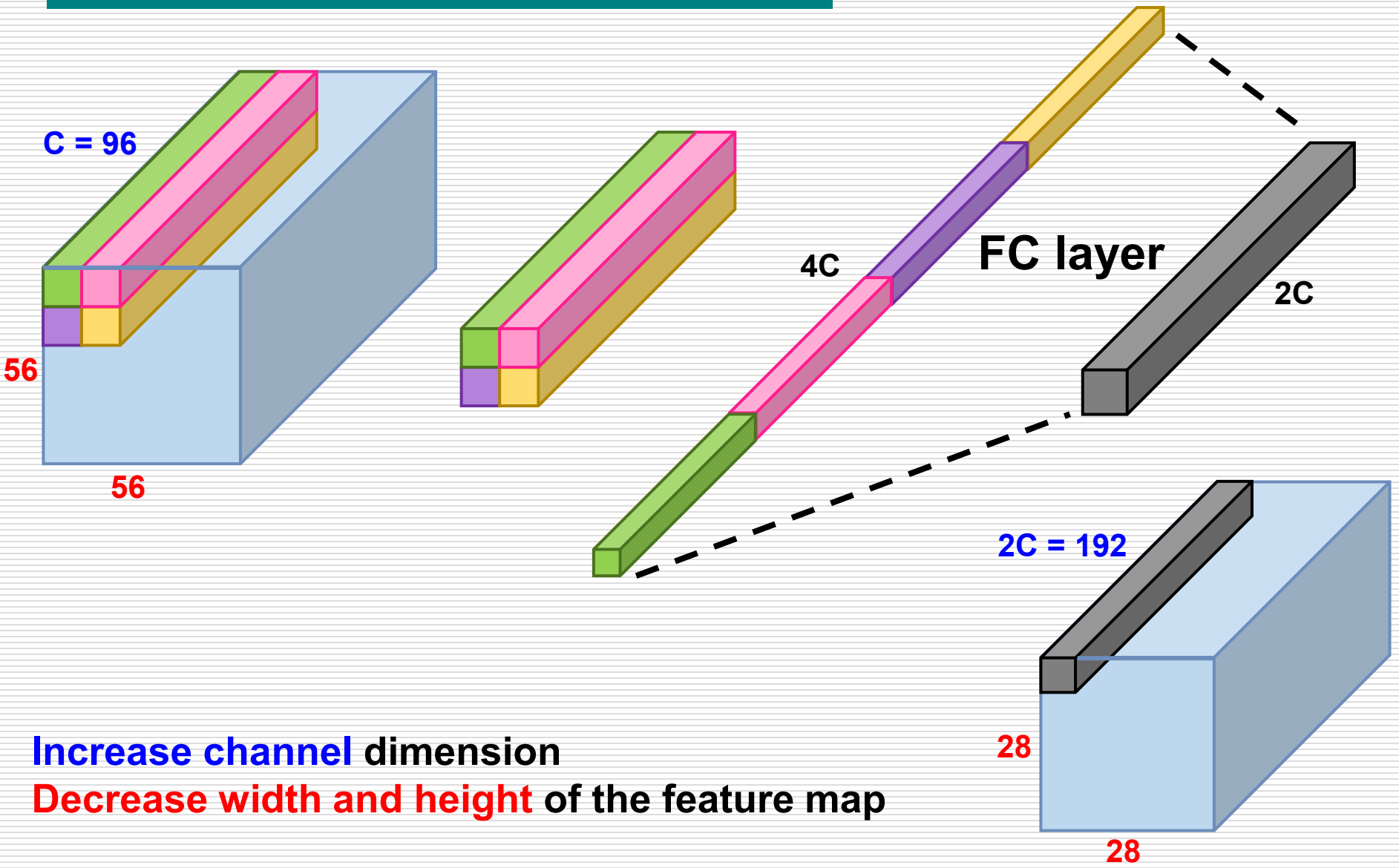
Model Architecture



Patch Partition



Patch Merging

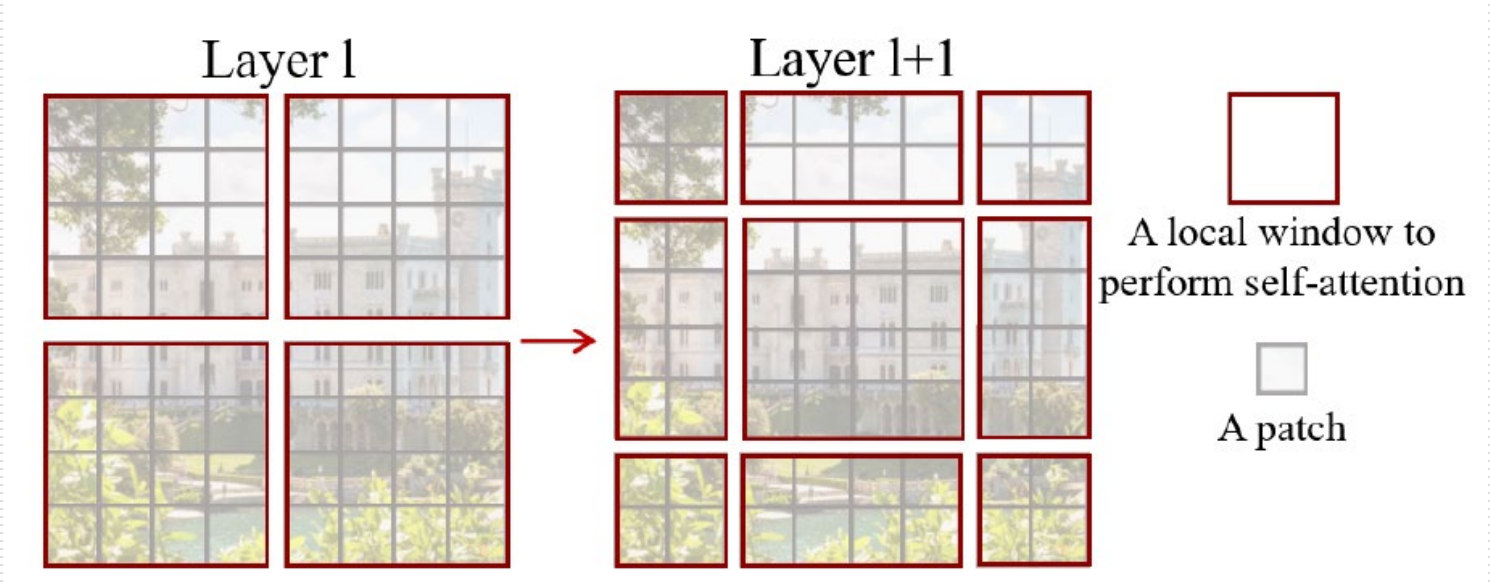


Increase channel dimension

Decrease width and height of the feature map

Window-Based MSA

- Perform attention only on tokens that belong to the same window ➔ reduce computation cost
- The shifted windows crosses the boundaries of the previous windows ➔ better model performance

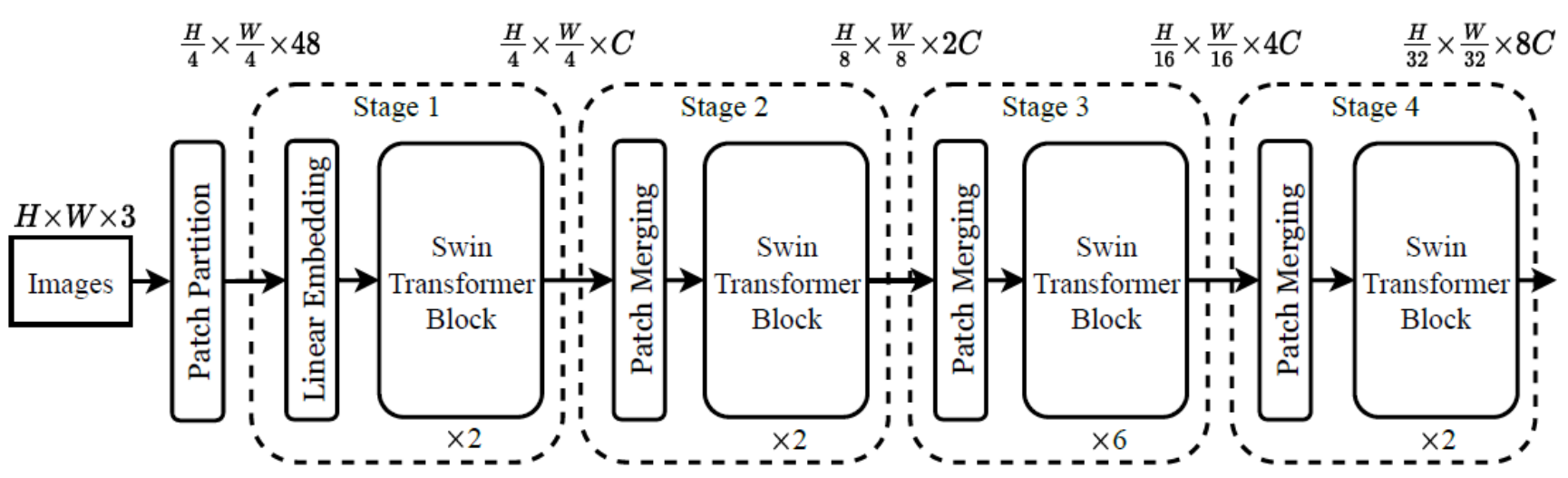


W-MSA (Regular Window MSA) **SW-MSA (Shifted Window MSA)**

Variant Architecture

- Window size $M = 7$
- MLP expansion ratio $\alpha = 4$

Tiny	• Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
Small	• Swin-S: $C = 96$, layer numbers = $\{2, 2, 18, 2\}$
Base	• Swin-B: $C = 128$, layer numbers = $\{2, 2, 18, 2\}$
Large	• Swin-L: $C = 192$, layer numbers = $\{2, 2, 18, 2\}$



Computational Complexity

- $\Omega(\mathrm{MSA}) = 4hwC^2 + 2(hw)^2C$
  - quadratic to patch number hw

- $\Omega(\mathrm{W{-}MSA}) = \left(\frac{h}{M} \times \frac{w}{M}\right) \times (4M^2C^2 + 2M^4C)$
  $= 4hwC^2 + 2M^2hwC$
  - linear to patch number hw when M is fixed

Experiment Results

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8 / **81.2***
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8 / **83.4***
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1 / **84.5***
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

*** With teacher model**

Summary of Swin

- Present a Window-based Attention
  - hierarchical feature representation
  - linear computational complexity with respect to input image size
  - better performance compared to DeiT

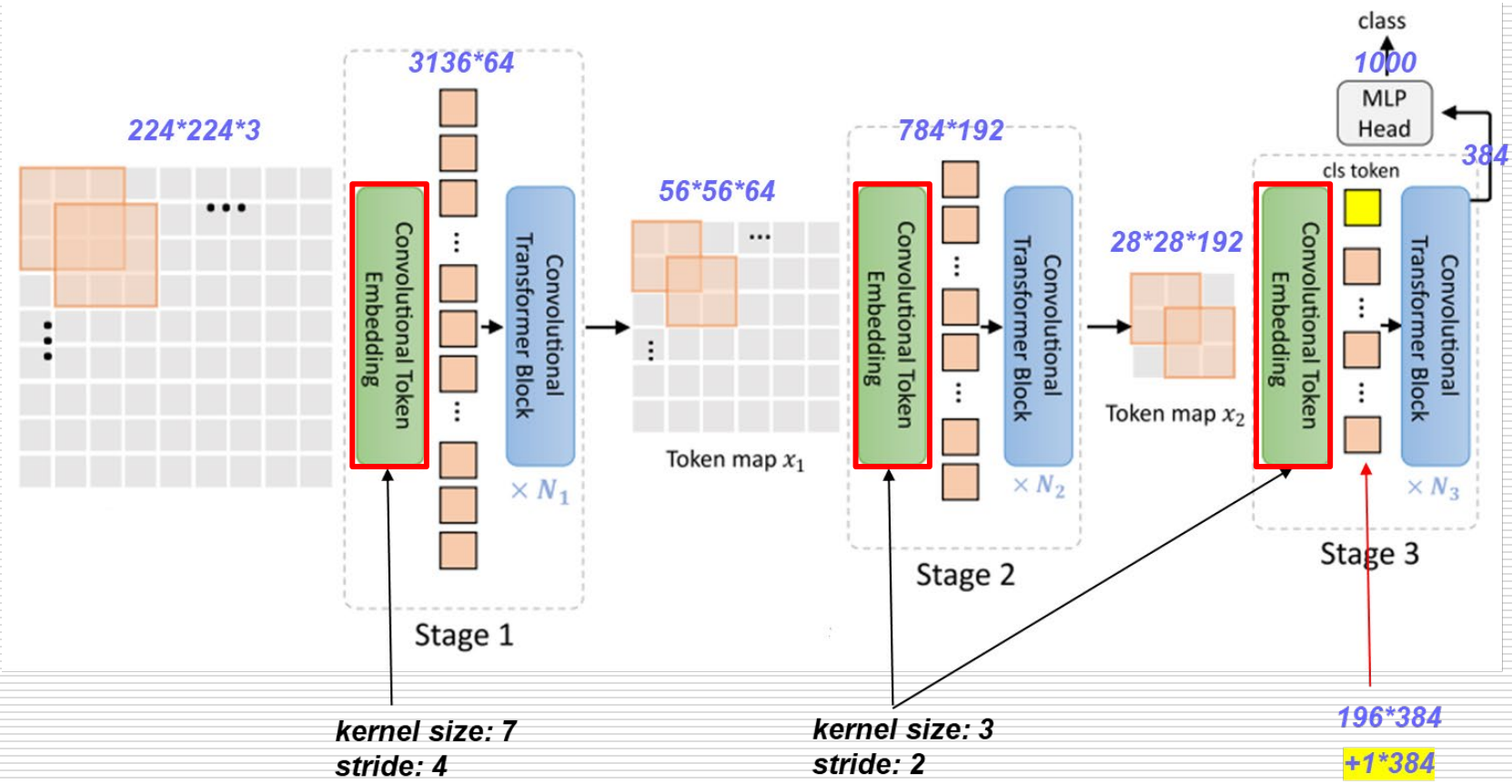
CvT

Microsoft, 2021

(CvT: Introducing Convolutions to Vision Transformers)

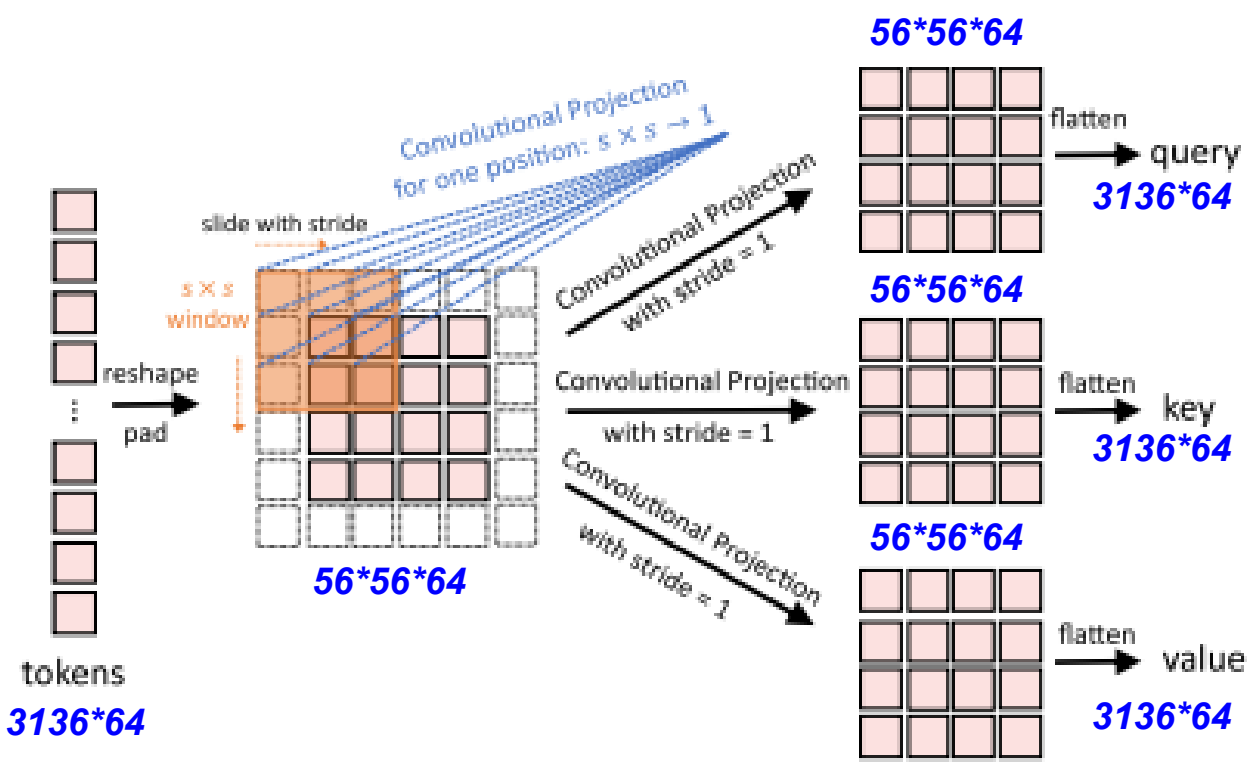
CvT

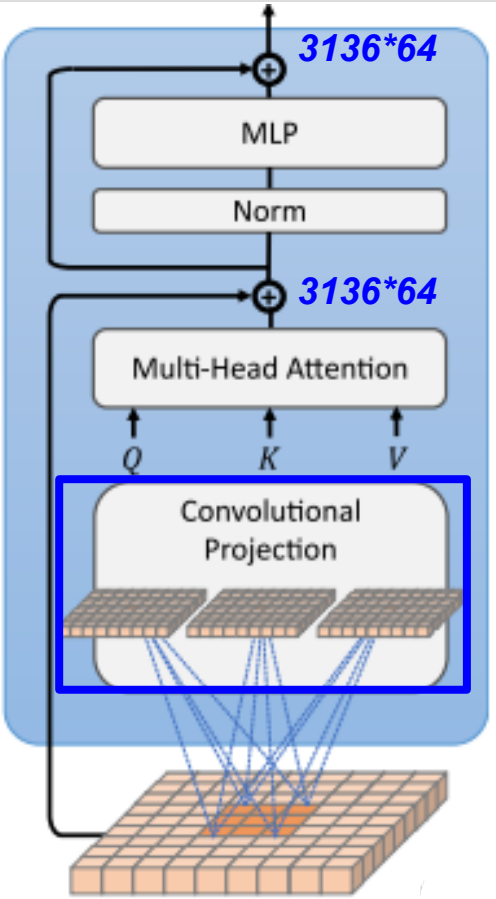
- Fusing Convolution in Transformer
  - hierarchical structure



Convolutional Transformer Block

- Replacing QKV linear projection with depth-wise convolution

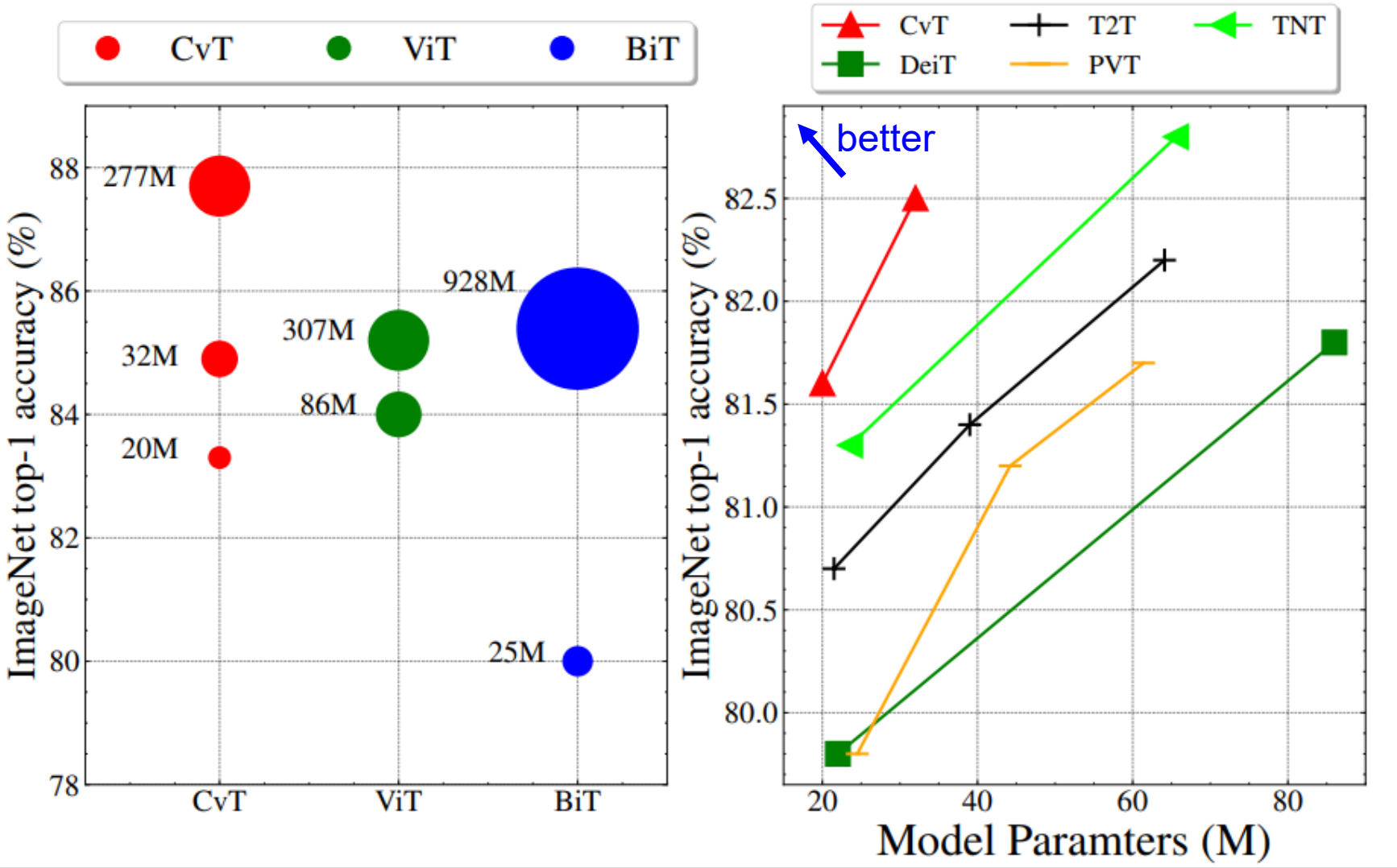




Experiment Results (1/2)

Method Type	Network	#Param. (M)	image size	FLOPs (G)	ImageNet top-1 (%)	Real top-1 (%)	V2 top-1 (%)
Convolutional Networks	ResNet-50 [15]	25	224^2	4.1	76.2	82.5	63.3
	ResNet-101 [15]	45	224^2	7.9	77.4	83.7	65.7
	ResNet-152 [15]	60	224^2	11	78.3	84.1	67.0
Transformers	ViT-B/16 [11]	86	384^2	55.5	77.9	83.6	–
	ViT-L/16 [11]	307	384^2	191.1	76.5	82.2	–
	DeiT-S [30][arxiv 2020]	22	224^2	4.6	79.8	85.7	68.5
	DeiT-B [30][arxiv 2020]	86	224^2	17.6	81.8	86.7	71.5
	PVT-Small [34][arxiv 2021]	25	224^2	3.8	79.8	–	–
	PVT-Medium [34][arxiv 2021]	44	224^2	6.7	81.2	–	–
	PVT-Large [34][arxiv 2021]	61	224^2	9.8	81.7	–	–
	T2T-ViT$_t$-14 [41][arxiv 2021]	22	224^2	6.1	80.7	–	–
	T2T-ViT$_t$-19 [41][arxiv 2021]	39	224^2	9.8	81.4	–	–
	T2T-ViT$_t$-24 [41][arxiv 2021]	64	224^2	15.0	82.2	–	–
	TNT-S [14][arxiv 2021]	24	224^2	5.2	81.3	–	–
	TNT-B [14][arxiv 2021]	66	224^2	14.1	82.8	–	–
Convolutional Transformers	**Ours:** CvT-13	20	224^2	4.5	81.6	86.7	70.4
	Ours: CvT-21	32	224^2	7.1	82.5	87.2	71.3
	Ours: CvT-13$_{\uparrow 384}$	20	384^2	16.3	83.0	87.9	71.9
	Ours: CvT-21$_{\uparrow 384}$	32	384^2	24.9	**83.3**	**87.7**	**71.9**

Experiment Results (2/2)



Ablation Study – Projection Method

- Replacing the Linear Projection with Convolutional Projection improves the performance

Method	Conv. Projection			Imagenet top-1 (%)
	Stage 1	Stage 2	Stage 3	
a				80.6
b	✓			80.8
c	✓	✓		81.0
d	✓	✓	✓	81.6
#Blocks	1	2	10	

Summary of CvT

- Introducing convolutions into the Vision Transformer architecture
  - Convolution + Transformer

- Achieving superior performance
  - convolutional token embedding
  - convolutional projection

CSWin Transformer

Microsoft, 2021

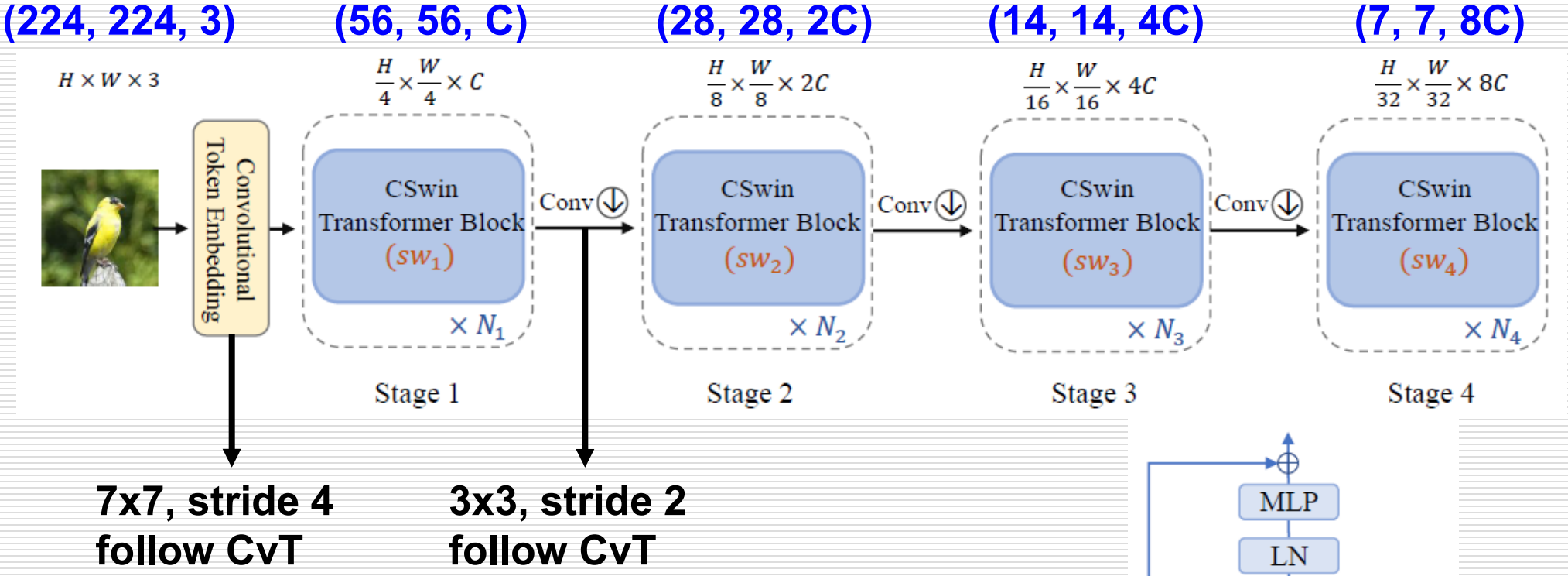
(CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows)

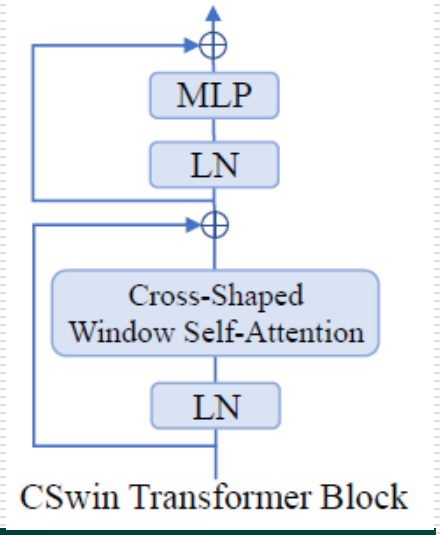
CSwin

- A challenging issue in transformer design
  - global self-attention is very expensive to compute
  - local self-attention limits the field of interactions of each token

➔ **C**ross-**S**haped **Win**dow self-attention

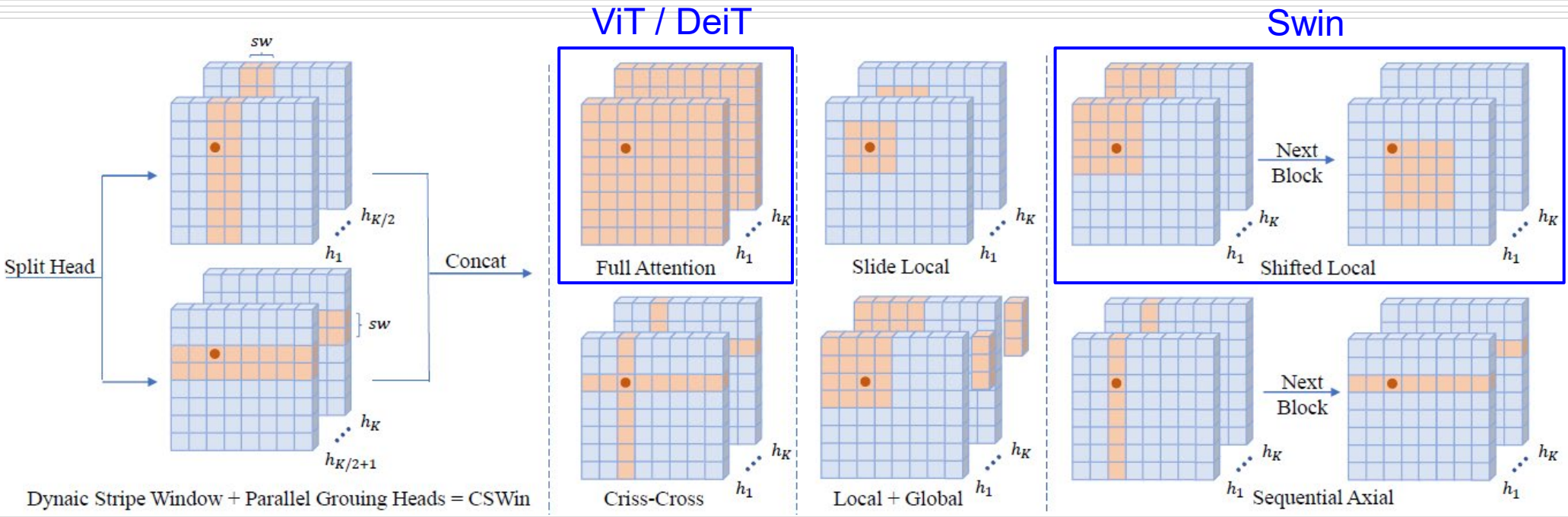
Model Architecture





Cross-Shaped Window Self-Attention

- Split multi-heads into groups
- Perform self-attention in horizontal/vertical **stripes**
- Adjust stripe width (sw) at different stages (1-2-7-7)



Positional Encoding

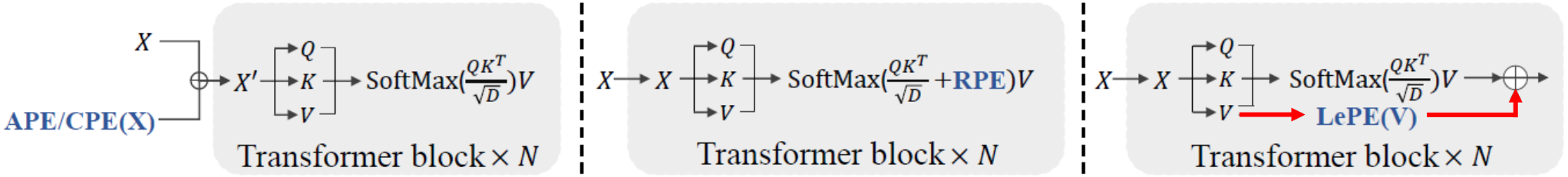
- Self-attention ignores the token positional information
  - positional encoding is widely used in Transformers to add positional information back

- PE are often defined as the sinusoidal functions of a series or the **learnable parameters**
  - sinusoidal functions: original transformer [1]
  - **learnable parameters:** BERT, ViT, DeiT, etc.

[1] Ashish Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017.

Locally-Enhanced Positional Encoding

- APE/RPE: Absolute/Relative Positional Encoding
  - NOT friendly to varying input size (need interpolation)
- CPE: Conditional Positional Encoding
  - use Conv layer to generate PE
  - arbitrary input size
- LePE
  - incorporate information within each Transformer block
  - arbitrary input size



APE: ViT, DeiT
CPE: CPVT, Pale

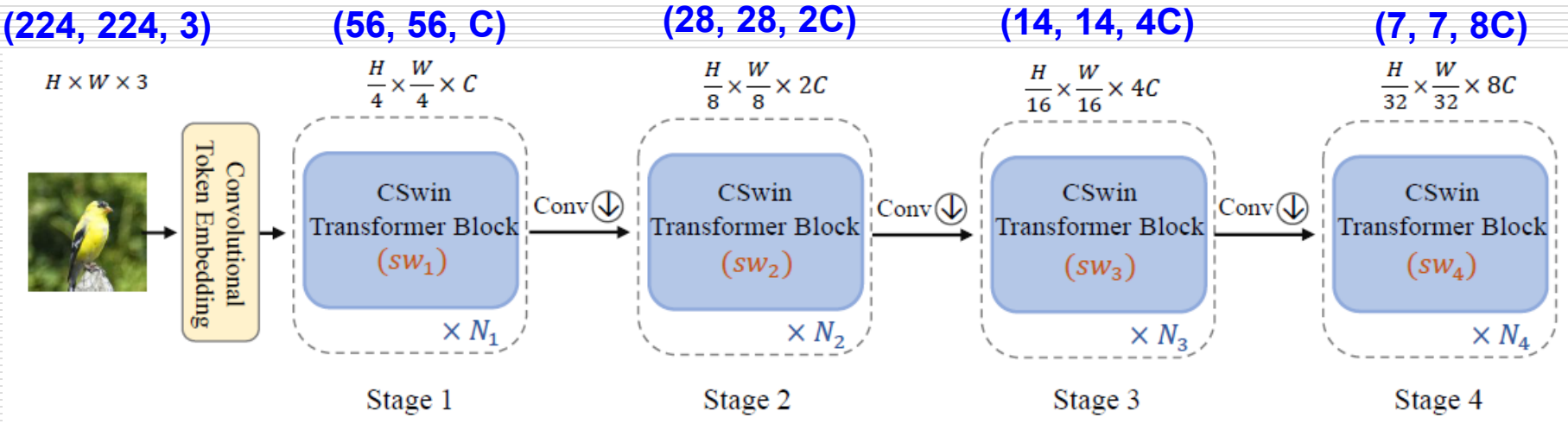
RPE: Swin

LePE: CSWin

Variant Architecture

- Apply global attention at the last stage ($sw = 7$)

Models	#Dim	#Blocks	sw	#heads	#Param.	FLOPs
CSWin-T	64	1,2,21,1	1,2,7,7	2,4,8,16	23M	4.3G
CSWin-S	64	2,4,32,2	1,2,7,7	2,4,8,16	35M	6.9G
CSWin-B	96	2,4,32,2	1,2,7,7	4,8,16,32	78M	15.0G
CSWin-L	144	2,4,32,2	1,2,7,7	6,12,24,48	173M	31.5G



Computational Complexity

- $\Omega(\mathrm{MSA}) = 4hwC^2 + 2(hw)^2C$
  - quadratic to patch number hw

- $\Omega(\mathrm{W{-}MSA}) = 4hwC^2 + 2M^2hwC$
  - linear to patch number hw when M is fixed

- $\Omega(\mathrm{CSWin}) = 4hwC^2 + sw(h+w)hwC$
  - smaller sw for early stages (higher resolution)
  - larger sw for later stages (lower resolution)

Experiment Results – ImageNet-1K

Method	Image Size	#Param.	FLOPs	Throughput	Top-1
Eff-B4 [51]	380^2	19M	4.2G	349/s	**82.9**
Eff-B5 [51]	456^2	30M	9.9G	169/s	**83.6**
Eff-B6 [51]	528^2	43M	19.0G	96/s	84.0
DeiT-S [53]	224^2	22M	4.6G	940/s	79.8
DeiT-B [53]	224^2	87M	17.5G	292/s	81.8
DeiT-B [53]	384^2	86M	55.4G	85/s	83.1
PVT-S [58]	224^2	25M	3.8G	820/s	79.8
PVT-M [58]	224^2	44M	6.7G	526/s	81.2
PVT-L [58]	224^2	61M	9.8G	367/s	81.7
$T2T_t$-14 [66]	224^2	22M	6.1G	–	81.7
$T2T_t$-19 [66]	224^2	39M	9.8G	–	82.2
$T2T_t$-24 [66]	224^2	64M	15.0G	–	82.6
CvT-13 [60]	224^2	20M	4.5G	–	81.6
CvT-21 [60]	224^2	32M	7.1G	–	82.5
CvT-21 [60]	384^2	32M	24.9G	–	83.3
Swin-T [38]	224^2	29M	4.5G	755/s	81.3
Swin-S [38]	224^2	50M	8.7G	437/s	83.0
Swin-B [38]	224^2	88M	15.4G	278/s	83.3
Swin-B [38]	384^2	88M	47.0G	85/s	84.2
CSWin-T	224^2	23M	4.3G	701/s	82.7
CSWin-S	224^2	35M	6.9G	437/s	**83.6**
CSWin-B	224^2	78M	15.0G	250/s	**84.2**
CSWin-B	384^2	78M	47.0G		**85.4**

Summary of CSwin

- Present the **Cross-Shaped Window** Self-Attention
  - can enlarge the attention area
  - with subtle extra computation cost

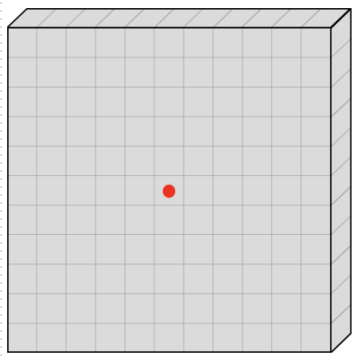
Pale Transformer

Baidu, 2021

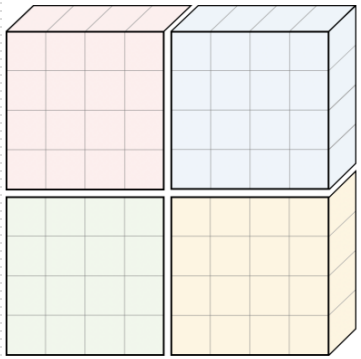
(Pale Transformer: A General Vision Transformer Backbone with Pale-Shaped Windows)

Pale Transformer

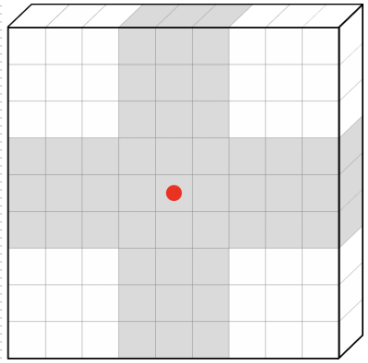
- Propose Pale-Shaped Window self-attention mechanism
  - benefit from the larger receptive fields



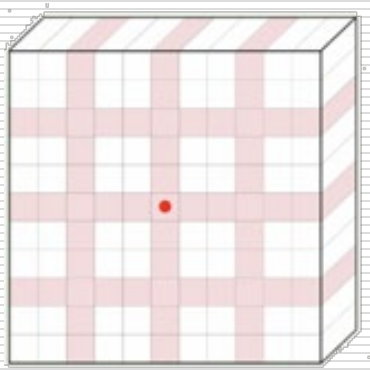
Global Self-Attention



Shifted Window



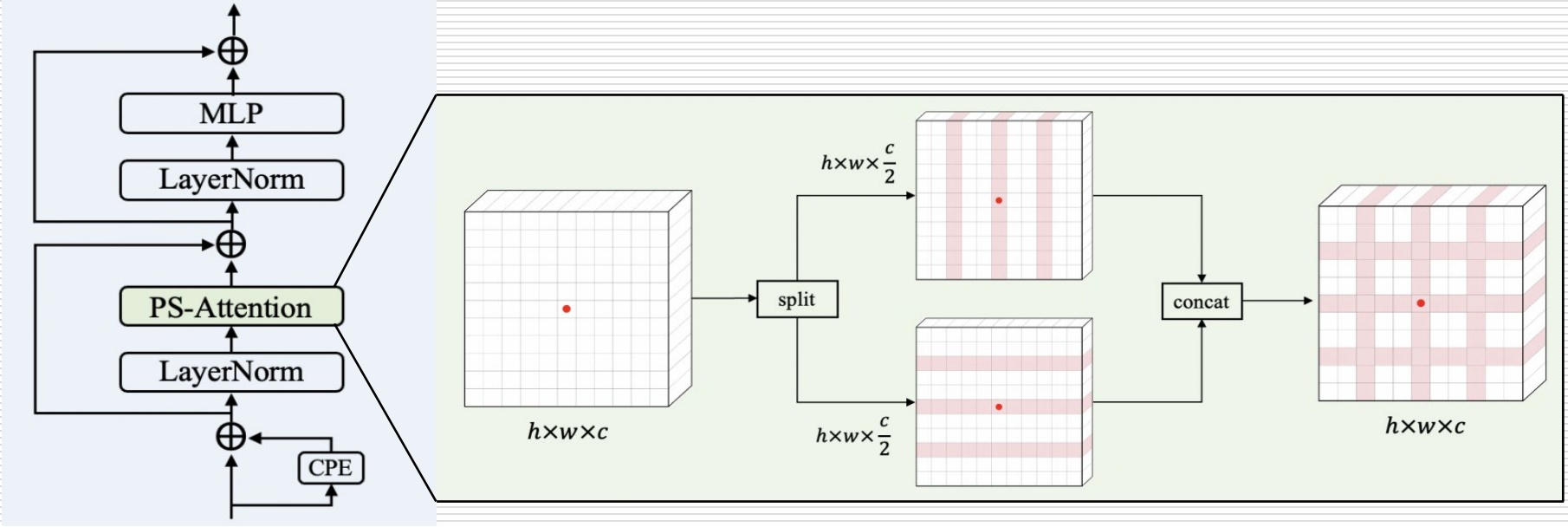
Cross-Shaped Window



Pale-Shaped Window

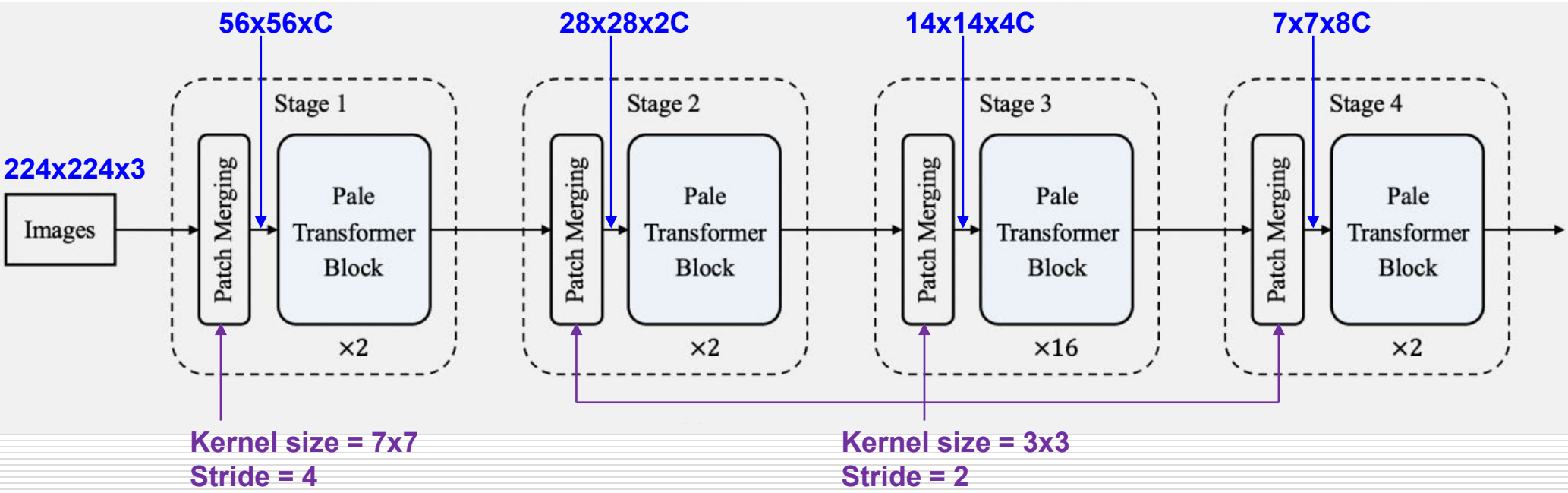
Pale Transformer Block

- Split multi-head into two groups
  - vertical stripes
  - horizontal stripes
- Replace QKV linear projection with depth-wise separable convolution



Model Architecture

- **Hierarchical** backbone with 4 stages



Variant Architecture

- Main differences are embedding dimension and head numbers

Stage	Output Stride	Layer	Pale-T	Pale-S	Pale-B
1	4	Patch Merging	$P_1 = 4$, $C_1 = 64$	$P_1 = 4$, $C_1 = 96$	$P_1 = 4$, $C_1 = 128$
		Pale Transformer Block	$\begin{bmatrix} S_1 = 7 \\ H_1 = 2 \\ R_1 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_1 = 7 \\ H_1 = 2 \\ R_1 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_1 = 7 \\ H_1 = 4 \\ R_1 = 4 \end{bmatrix} \times 2$
2	8	Patch Merging	$P_2 = 2$, $C_2 = 128$	$P_2 = 2$, $C_2 = 192$	$P_2 = 2$, $C_2 = 256$
		Pale Transformer Block	$\begin{bmatrix} S_2 = 7 \\ H_2 = 4 \\ R_2 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_2 = 7 \\ H_2 = 4 \\ R_2 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_2 = 7 \\ H_2 = 8 \\ R_2 = 4 \end{bmatrix} \times 2$
3	16	Patch Merging	$P_3 = 2$, $C_3 = 256$	$P_3 = 2$, $C_3 = 384$	$P_3 = 2$, $C_3 = 512$
		Pale Transformer Block	$\begin{bmatrix} S_3 = 7 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 16$	$\begin{bmatrix} S_3 = 7 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 16$	$\begin{bmatrix} S_3 = 7 \\ H_3 = 16 \\ R_3 = 4 \end{bmatrix} \times 16$
4	32	Patch Merging	$P_4 = 2$, $C_4 = 512$	$P_4 = 2$, $C_4 = 768$	$P_4 = 2$, $C_4 = 1024$
		Pale Transformer Block	$\begin{bmatrix} S_4 = 7 \\ H_4 = 16 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_4 = 7 \\ H_4 = 16 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} S_4 = 7 \\ H_4 = 32 \\ R_4 = 4 \end{bmatrix} \times 2$

Computational Complexity

- $\Omega(\mathrm{MSA}) = 4hwC^2 + 2(hw)^2C$
  - quadratic to patch number hw
- $\Omega(\mathrm{W{-}MSA}) = 4hwC^2 + 2M^2hwC$
  - linear to patch number hw when M is fixed
- $\Omega(\mathrm{CSWin}) = 4hwC^2 + sw(h+w)hwC$
  - smaller sw for early stages (higher resolution)
  - larger sw for later stages (lower resolution)
- $\Omega(\mathrm{Pale}) = 4hwC^2 + (s_ch + s_rw + 27)hwC$

Experiment Results – ImageNet-1K

Backbone	Params	FLOPs	Top-1 (%)
DeiT-S (Touvron et al. 2021)	22M	4.6G	79.8
PVT-S (Wang et al. 2021b)	25M	3.8G	79.8
T2T-14 (Yuan et al. 2021a)	22M	6.1G	80.7
Swin-T (Liu et al. 2021)	29M	4.5G	81.3
Twins-SVT-S (Chu et al. 2021a)	24M	2.8G	81.3
CvT-13 (Wu et al. 2021a)	20M	4.5G	81.6
CSWin-T (Dong et al. 2021)	23M	4.3G	82.7
Pale-T (ours)	22M	4.2G	**83.4**
PVT-M (Wang et al. 2021b)	44M	6.7G	81.2
T2T-19 (Yuan et al. 2021a)	39M	9.8G	81.4
CvT-21 (Wu et al. 2021a)	32M	7.1G	82.5
Swin-S (Liu et al. 2021)	50M	8.7G	83.0
Twins-SVT-B (Chu et al. 2021a)	56M	8.3G	83.1
CSWin-S (Dong et al. 2021)	35M	6.9G	83.6
Refined-ViT-S (Zhou et al. 2021)	25M	7.2G	83.6
VOLO-D1* (Yuan et al. 2021b)	27M	6.8G	84.2
Pale-S (ours)	48M	9.0G	**84.3**

Backbone	Params	FLOPs	Top-1 (%)
DeiT-B (Touvron et al. 2021)	86M	17.5G	81.8
T2T-24 (Yuan et al. 2021a)	64M	15.0G	82.2
Swin-B (Liu et al. 2021)	88M	15.4G	83.3
Twins-SVT-L (Chu et al. 2021a)	99M	14.8G	83.3
Focal-B (Yang et al. 2021)	90M	16.0G	83.8
Shuffle-B (Huang et al. 2021)	88M	15.6G	84.0
CSWin-B (Dong et al. 2021)	78M	15.0G	84.2
Refined-ViT-M (Zhou et al. 2021)	55M	13.5G	84.6
VOLO-D2* (Yuan et al. 2021b)	59M	14.1G	85.2
Pale-B (ours)	85M	15.6G	**84.9**

Summary of Pale Transformer

- Proposed Pale-Shaped Window self-attention mechanism
  - **larger receptive fields**
  - better model performance

- Potential issue
  - **irregular** matrix computation

  ➔ hard to accelerate on hardware (GPU/ASIC accelerator)

Takeaways

- ViT is the first work that applies self-attention to the CV task
- Its variants attempt to use local attention to reduce the computational cost
- Although the window-based methods perform well, executing them on hardware is not efficient due to their irregular operation

Exercise (1/2)

- Download the example file from FB club
  - Copy it to your colab
  - Build a ViT model for image classification on FashionMNIST
  - Report #MACs and #parameters of a single transformer encoder layer of this ViT model by manual calculations and provide detailed calculation steps
    - #MACs: calculate linear layer (matrix multiplication) only
- Specifications
  - #Encoder layer = 4, #Head = 4, Patch size: 4x4
  - Embedding dimension = 64, MLP expansion ratio = 4
  - Do not modify the number of epochs
  - Do not modify valid_size
  - Overall Test Accuracy ≥ 84%

Exercise (2/2)

- Deadline: 8/12 (Mon.) 23:59

- Submit code (.ipynb) and report (.pdf) to:
- https://docs.google.com/forms/d/e/1FAIpQLSedwqioz4mIfwEqGGR-AuHFBh_1srHs8uCPnpfckp0-xq77WA/viewform?usp=sf_link
  - HW7_帳號.ipynb
  - HW7_帳號.pdf
    - Overall test accuracy screenshot (60%)
    - #MACs calculation (20%)
    - #parameters calculation (20%)

```
Test Loss: 0.358496

Test Accuracy of Class     0: 83.00% (830/1000)
Test Accuracy of Class     1: 96.80% (968/1000)
Test Accuracy of Class     2: 78.30% (783/1000)
Test Accuracy of Class     3: 88.70% (887/1000)
Test Accuracy of Class     4: 79.50% (795/1000)
Test Accuracy of Class     5: 92.90% (929/1000)
Test Accuracy of Class     6: 65.90% (659/1000)
Test Accuracy of Class     7: 93.40% (934/1000)
Test Accuracy of Class     8: 96.00% (960/1000)
Test Accuracy of Class     9: 94.90% (949/1000)

Test Accuracy (Overall): 86.94% (8694/10000)
```

Thank you