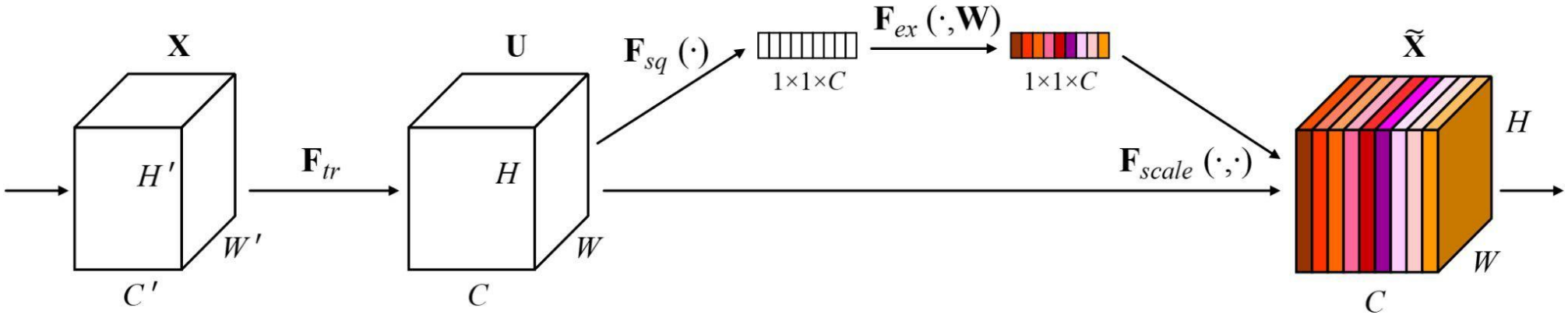
CONVNETS WITH NON-LOCAL OPERATIONS

https://iaml-it.github.io/posts/2021-04-28-transformers-in-vision/

Source: CVPR Tutorial, Beyond CNN https://sites.google.com/view/cvpr-2022-beyond-cnn?pli=1

Squeeze-and-Excitation Networks



Hu et al., 2018

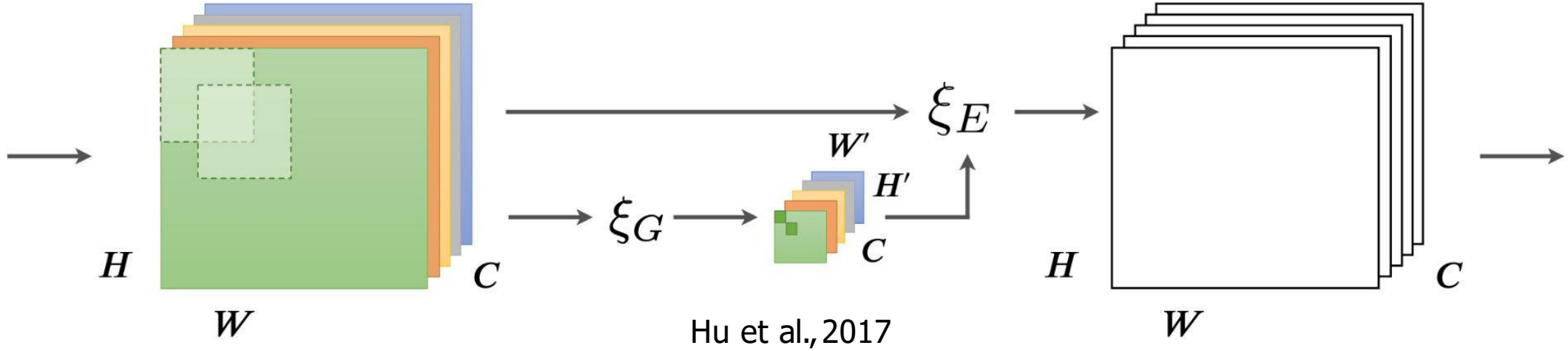
Escape local-only processing with Squeeze-and-Excite block:

- Downsample the data
- Apply (non)linear projection
- Use the result to scale the data

Result: all-to-all communication, but in a very constrained form.

Hu et al., Squeeze-and-Excitation Networks, CVPR 2018

Gather-Excite: Exploiting Feature Context in CNNs



Hu et al., 2017

	top-1 err.	top-5 err.	GFLOPs	#Params
ResNet-101	22.20	6.14	7.57	44.6 M
ResNet-50 (Baseline)	23.30	6.55	3.86	25.6 M
SE	22.12	5.99	3.87	28.1 M
GE-θ^-	22.14	6.24	3.86	25.6 M
GE-θ	22.00	5.87	3.87	31.2 M
GE-θ^+	21.88	5.80	3.87	33.7 M

Hu et al., 2018

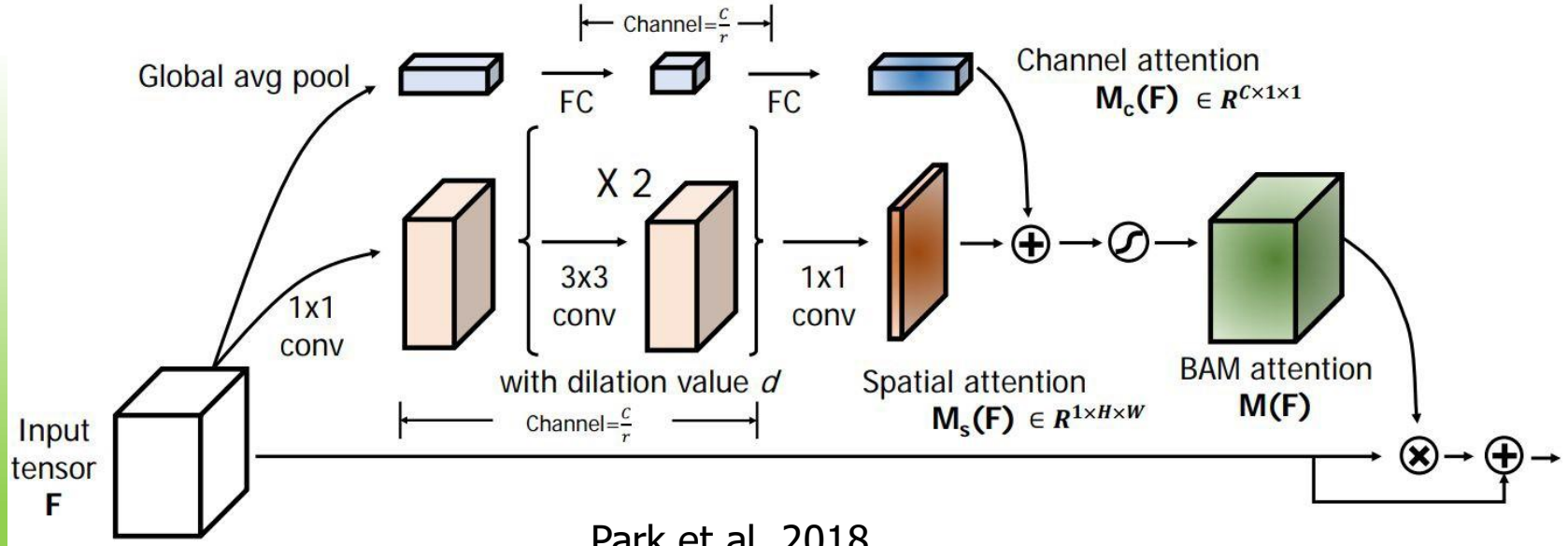
global operation appears to save compute and parameter count

類似SENet，但換成整個feature map

1. 在輸入x上進行卷積或者池化操作生成維度減小的feature map
2. 使用Nearest Neighbor interpolation（最近鄰插值方法）resize到相同shape，經過sigmoid後做點積運算

Hu et al., Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks, NeurIPS 2018

BAM: Bottleneck Attention Module



Park et al., 2018

- Further changes to SE design to make it more flexible
- Leads to lower error "for free"

Maybe consider even more general global ops? 🤔

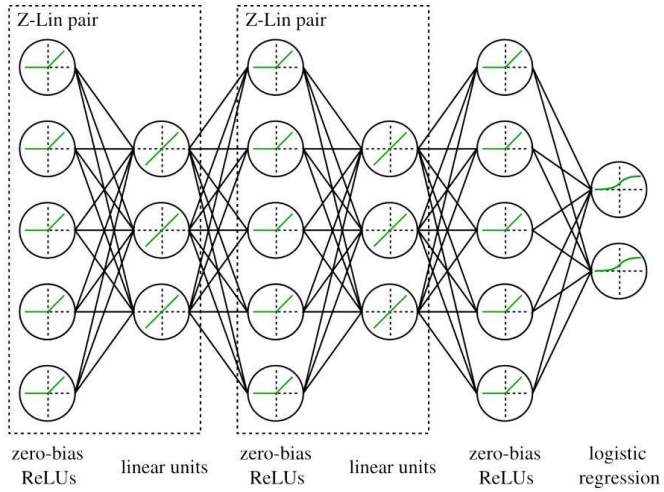
Architecture	Params	GFLOPs	Error	Architecture	Params	GFLOPs	Error
ResNet50	23.71M	1.22	21.49	WideResNet28 (w=8)	23.40M	3.36	20.40
ResNet50 + SE[19]	26.24M	1.23	20.72	WideResNet28 (w=8) + SE[19]	23.58M	3.36	19.85
ResNet50 + BAM	24.07M	1.25	**20.00**	WideResNet28 (w=8) + BAM	23.42M	3.37	**19.06**
PreResNet110	1.73M	0.245	22.22	ResNext29 16x64d	68.25M	9.88	17.25
PreResNet110 + SE[19]	1.93M	0.245	**21.85**	ResNext29 16x64d + SE[19]	68.81M	9.88	16.52
PreResNet110 + BAM	1.73M	0.246	21.96	ResNext29 16x64d + BAM	68.34M	9.9	**16.39**

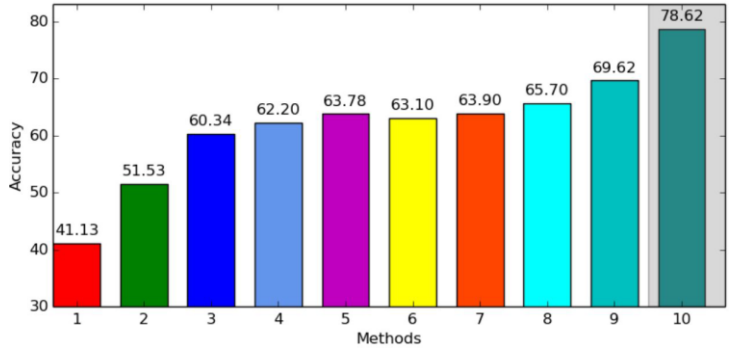
Park et al., 2018

Park et al., Bottleneck attention module, BMVC 2018

MLPs for vision

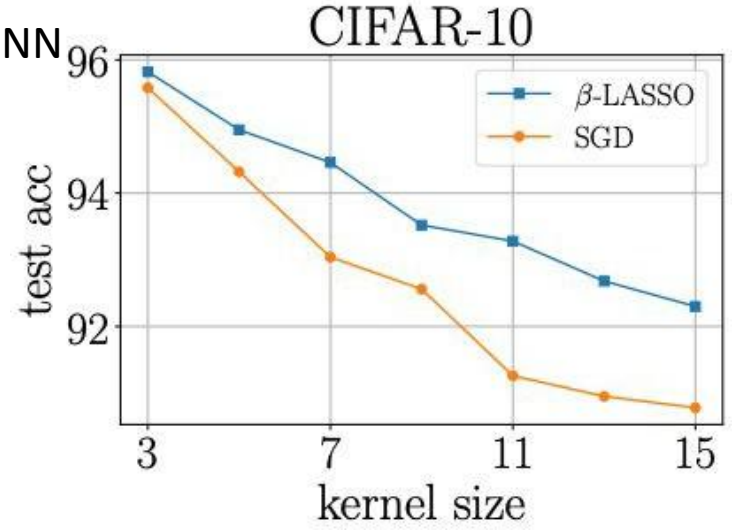
MLP





Lin et al., 2015

CNN



Neyshabur, 2020

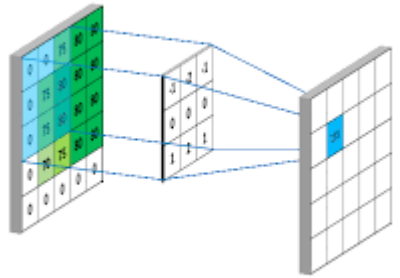
- **An indication that very power architectures become competitive with strong regularization/augmentation schemes**
- **Data size was a limiting factor**

Lin et al., How far can we go without convolution: Improving fully-connected networks, 2015
Neyshabur, Towards Learning Convolutions from Scratch, NeurIPS 2020

VISION TRANSFORMER

Self-attention vs. Convolution in Vision

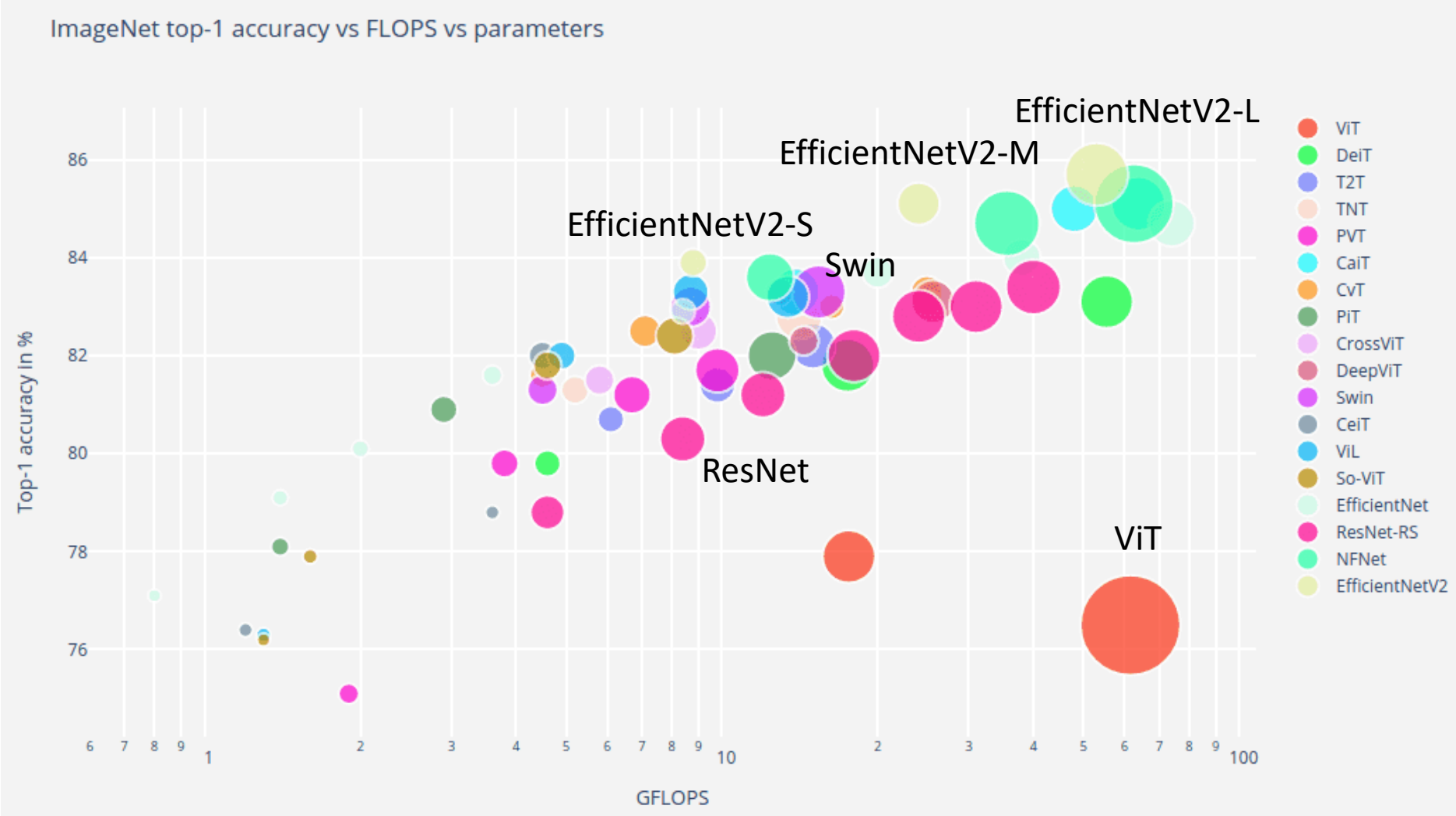
- Each pixel is an input “token”
- Traditionally, convolution is used to integrate pixels
  - Recognize patterns within a small window of pixels
  - Difficult to integrate non-local pixels
  - Have to make network very deep to “see the big picture”



- Self-attention (transformer) also integrates multiple pixels
  - Works when the correlated pixels are non-local
  - E.g. Trunk, tail, legs of an elephant to a whole elephant

CNN vs Transformer

- CNN: inductive biases 歸納偏置 (先驗假設)
  - translation invariance and a locally restricted receptive field
  - invariance means that
    - you can recognize an entity (i.e. object) in an image, even when its appearance or position varies. Translation in computer vision implies that each image pixel has been moved by a fixed amount in a particular direction
  - convolution is
    - a linear local operator. We see only the neighbor values as indicated by the kernel.
- Transformer
  - permutation invariant
  - The bad news is that it cannot process grid-structured data. We need sequences! To this end, we will convert a spatial non-sequential signal to a sequence!



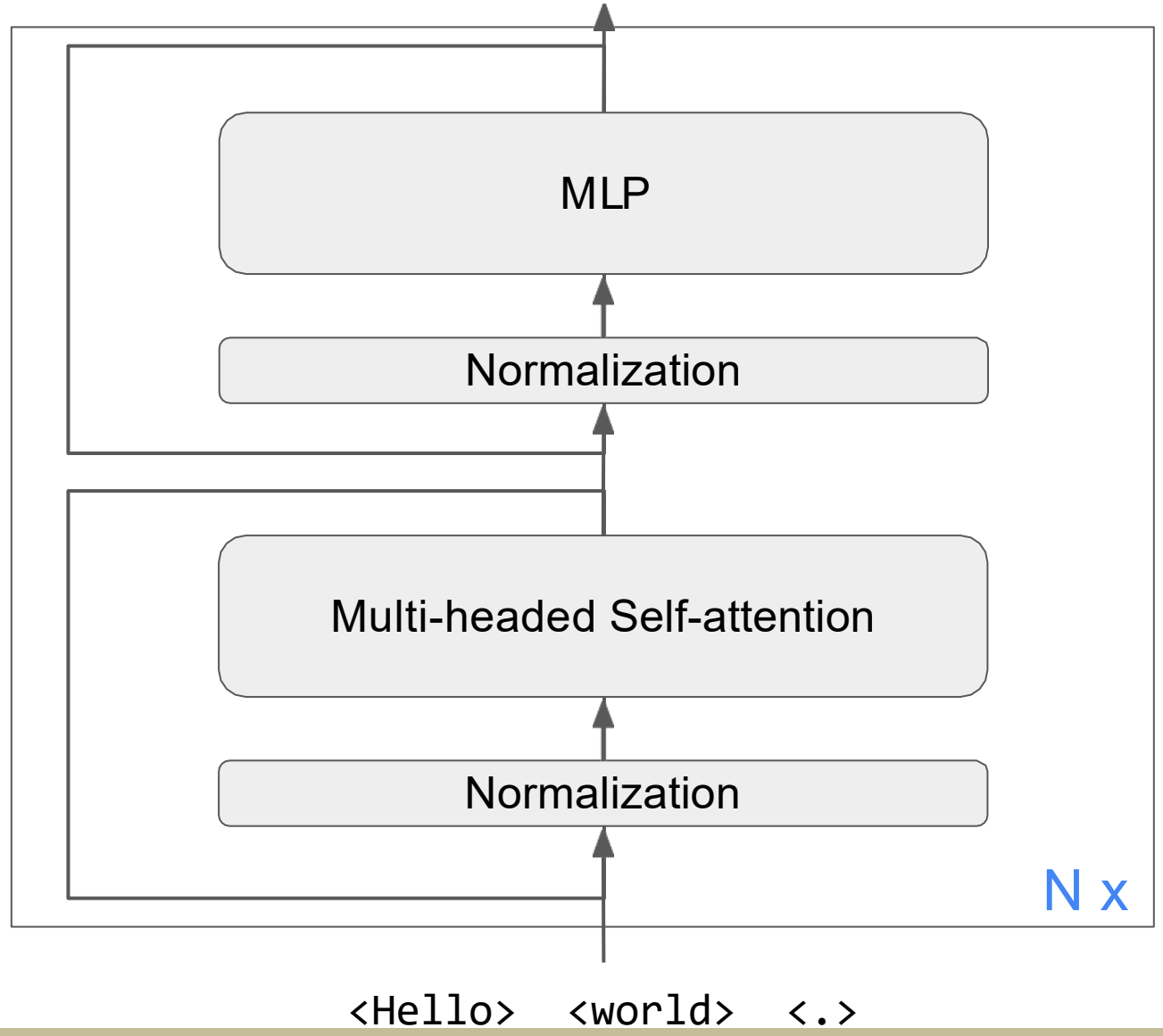
https://theaisummer.com/transformers-computer-vision/ https://iaml-it.github.io/posts/2021-04-28-transformers-in-vision/

- Imagenet accuracy

Rank	Model	Top 1 Accuracy	Number of params	GFLOPs	energy consumption	Extra Training Data	Paper	Code	Result	Year	Tags
1	**OmniVec** (ViT)	92.4%				×	OmniVec: Learning robust representations with cross modal sharing			2023	
2	**CoCa** (finetuned)	91.0%	2100M			×	CoCa: Contrastive Captioners are Image-Text Foundation Models			2022	ALIGN, Transforr, JFT-3B

https://paperswithcode.com/sota/image-classification-on-imagenet

What is a Transformer? (Encoder-only variant)



Two layers,
large hidden dimension

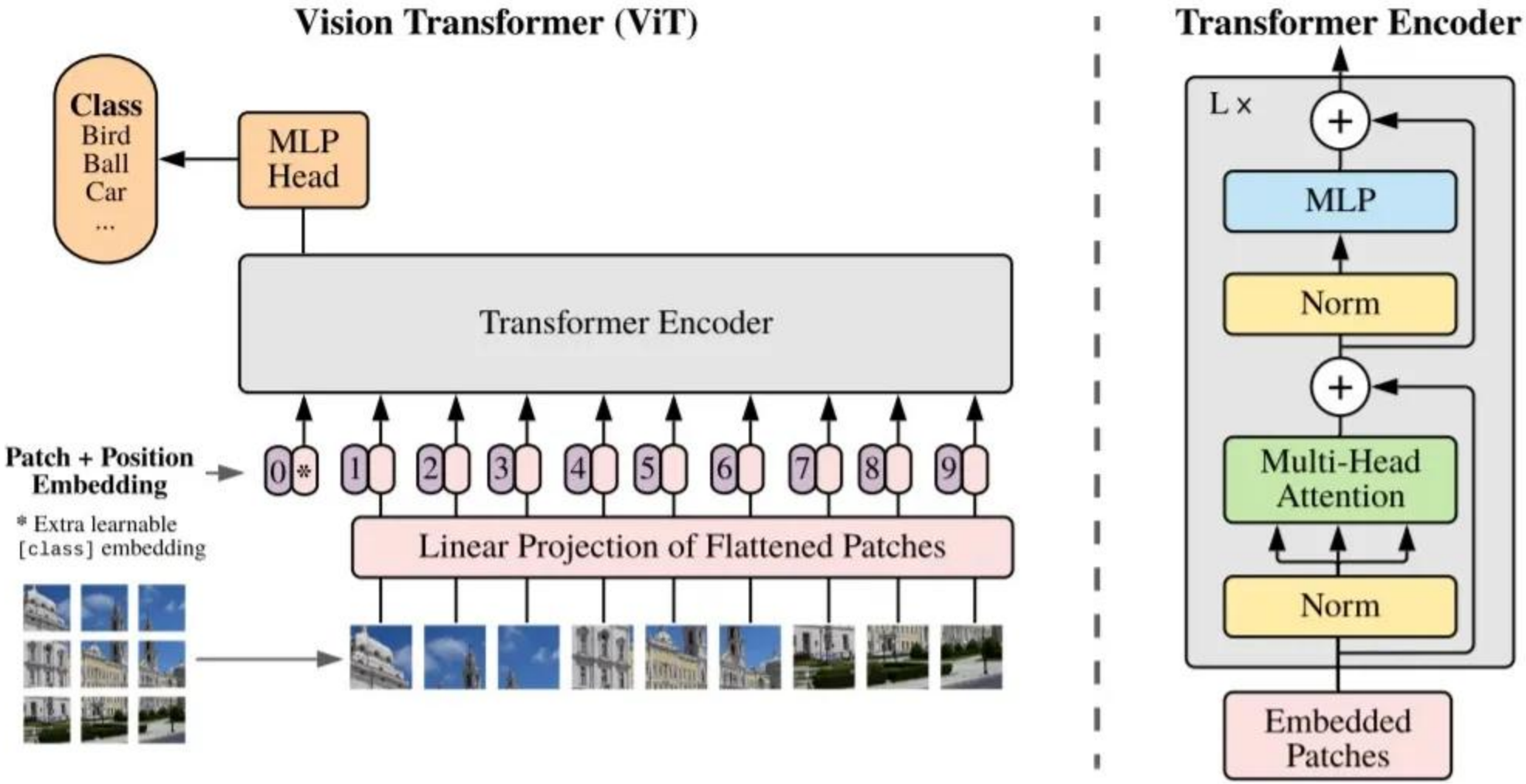
Layer where features for different words interact

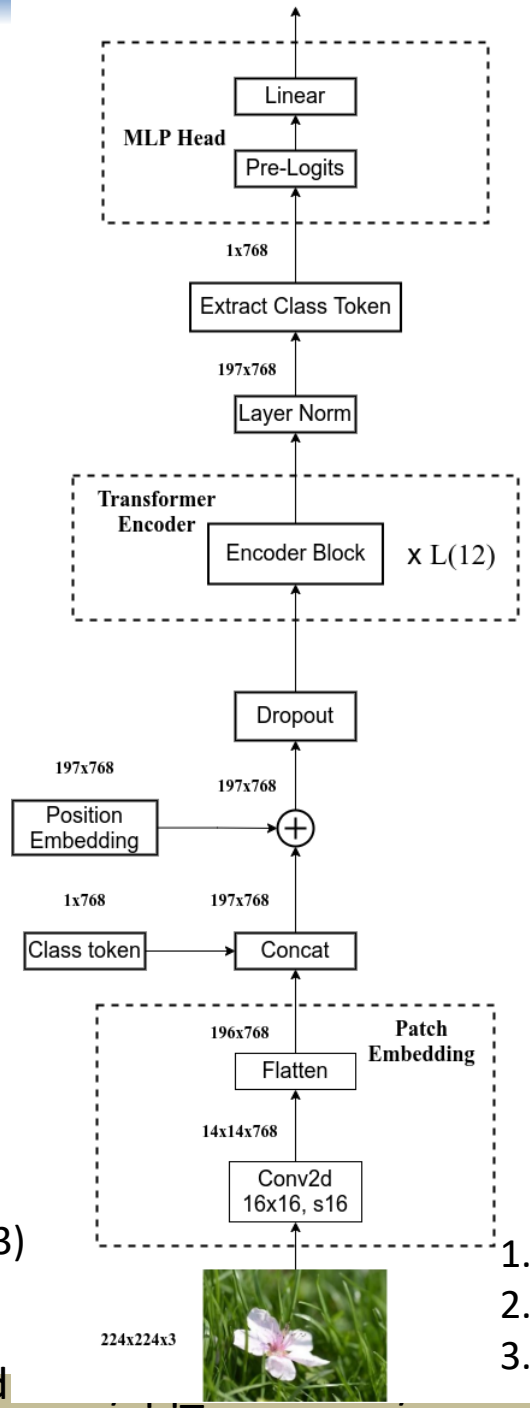
Tokenization
+ word embedding
+ position embeddings
→ Sequence of embedding vectors

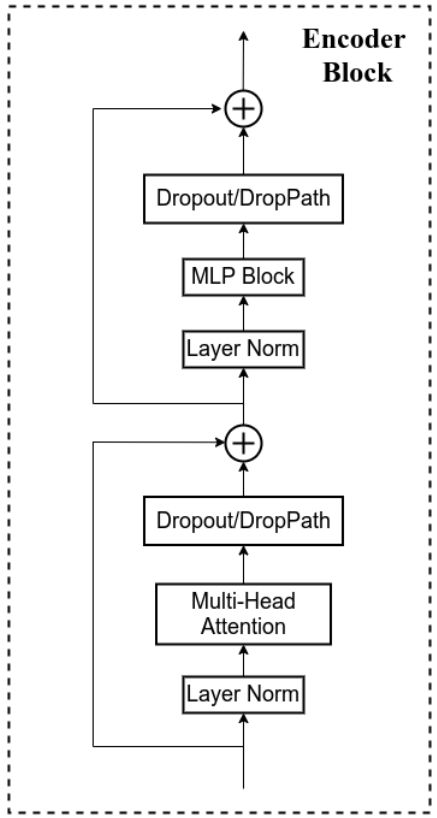
Transformer 用於影像的挑戰與解決方案

- 不改transformer 架構，直接用於image
  - 好處: scalable
- 計算複雜度 $O(n^2)$
  - NLP: n=512, ImageNet: 224x224 =>1-D n=50176
  - ViT: 16x16 Patch, (224/16 =14), n=14x14 = 196
- Lack of inductive bias
  - 大規模數據集預訓練
  - CNN extract features => transformer
- 其他
  - 位置編碼: 1-D/2-D/relative position coding性能差異不大

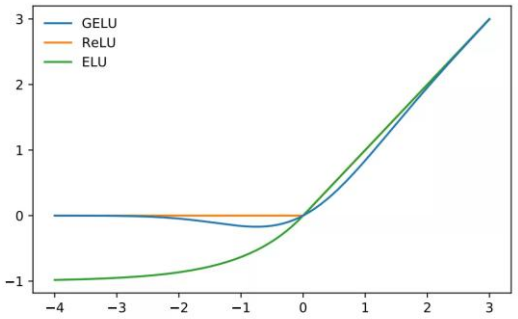
ViT

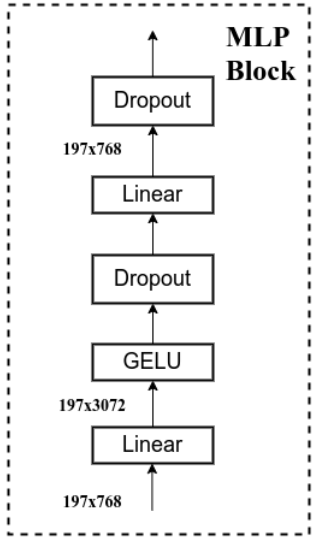

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches





ViT-B/16



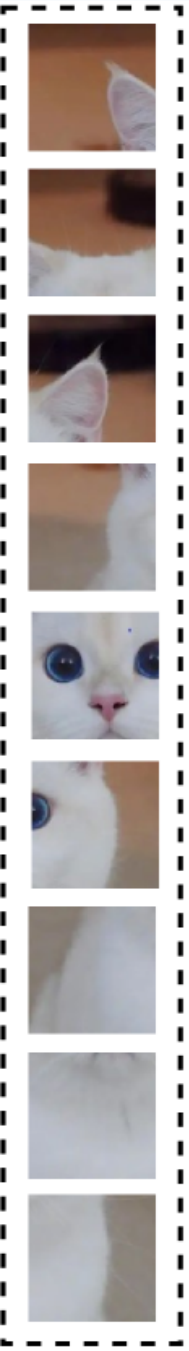


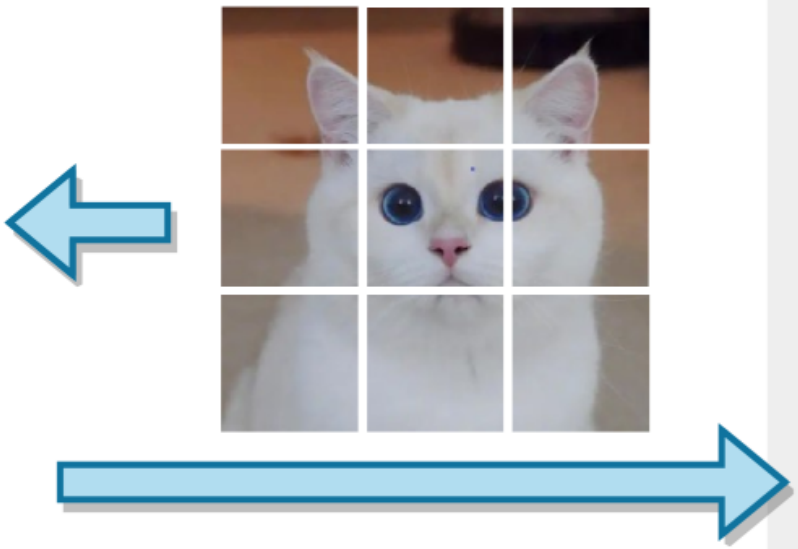
gaussian error linear units

224/16 = 14
=>14x14 (16x16x3)
16x16x3 = 768

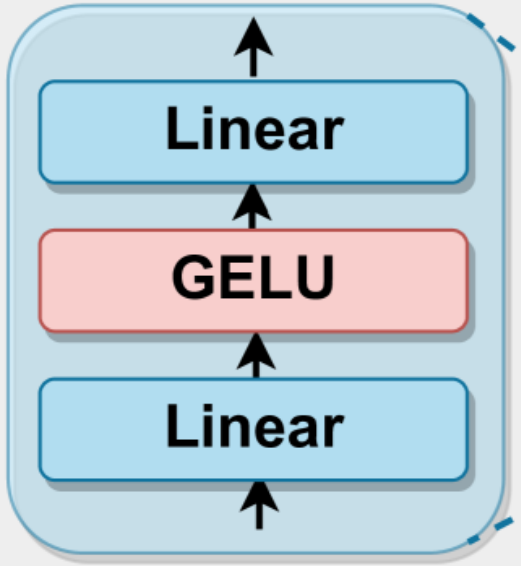
https://blog.csd

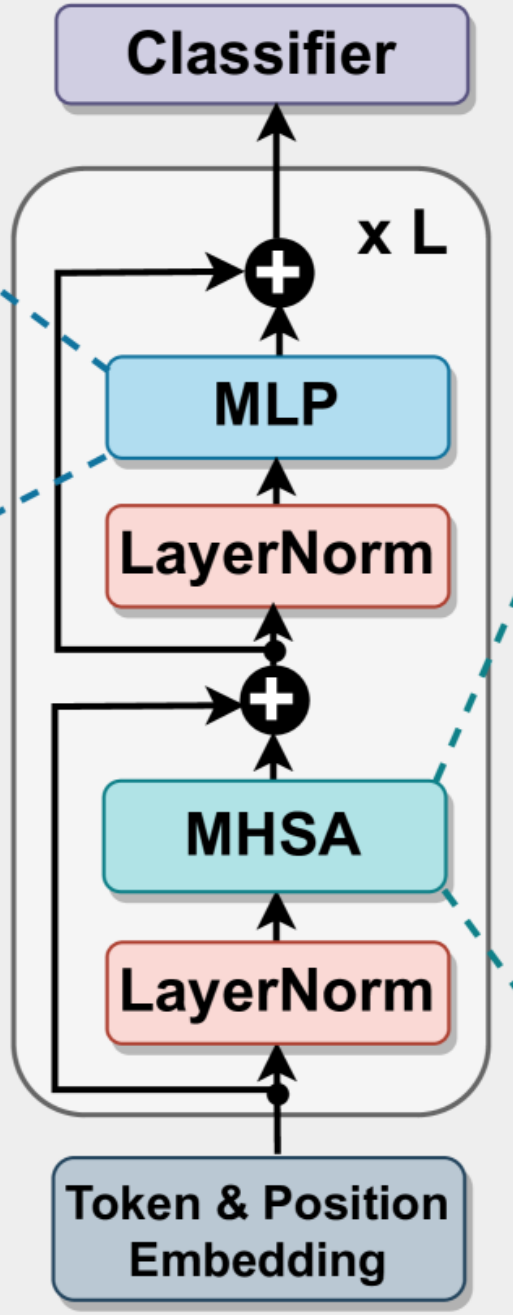
1. Class Tokens are randomly initialized
2. Learnable embedding to accumulate information fro the other tokens in the sequence
3. MLP head which only looks at data from the last layer's Class Token and no other information



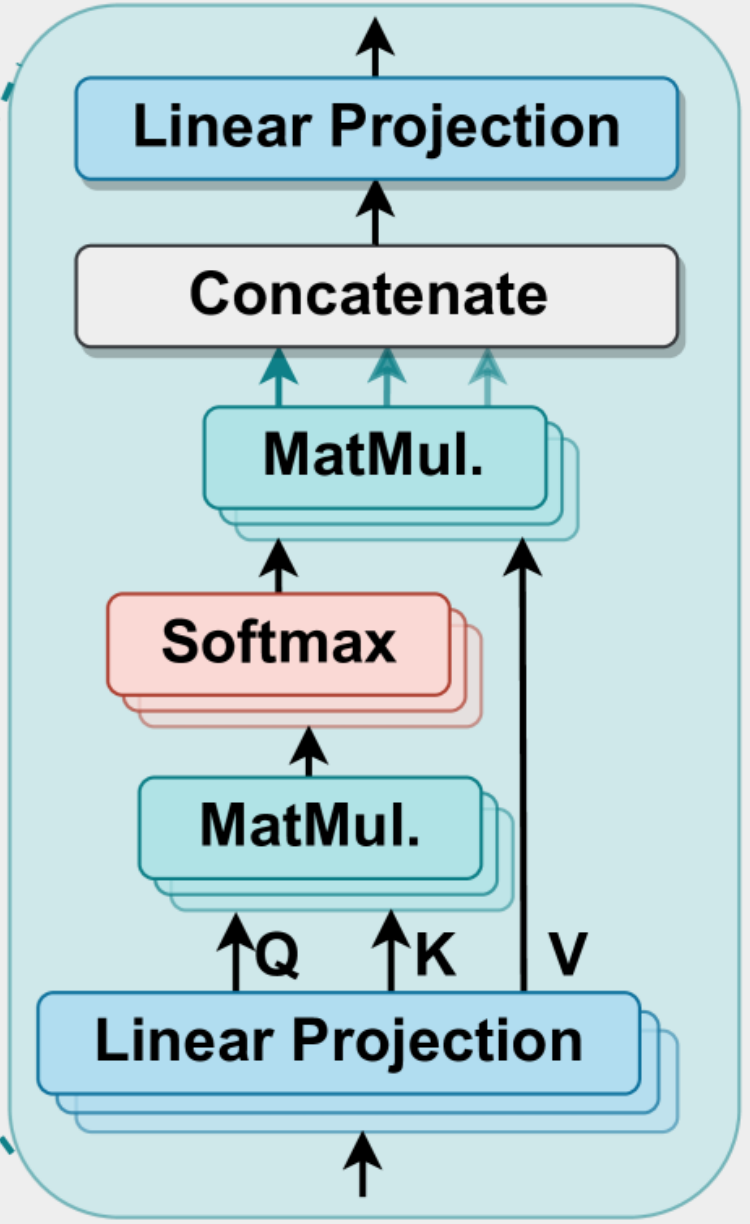


Multi-Layer Perceptron

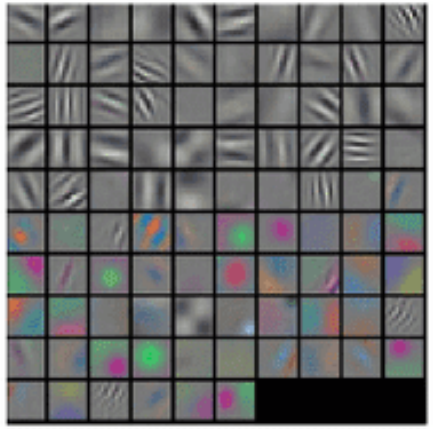




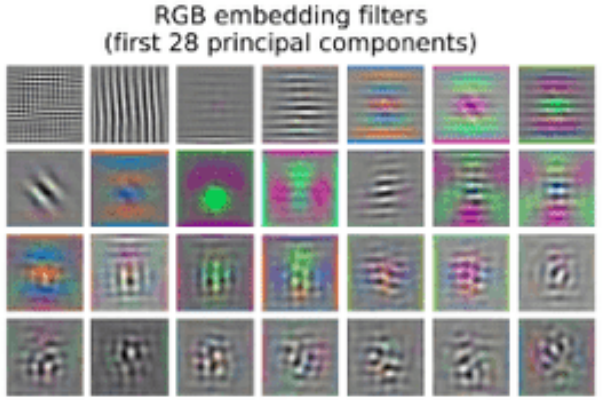
Multi-Head Self-Attention



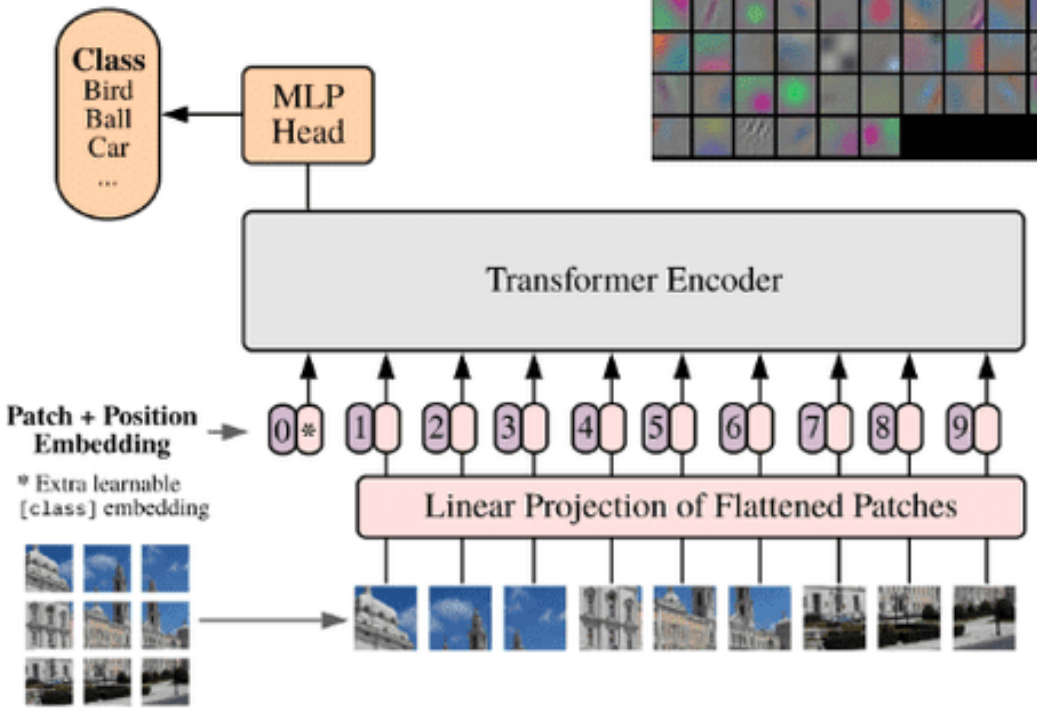
Alexnet 1st conv filters



ViT 1st linear embedding filters



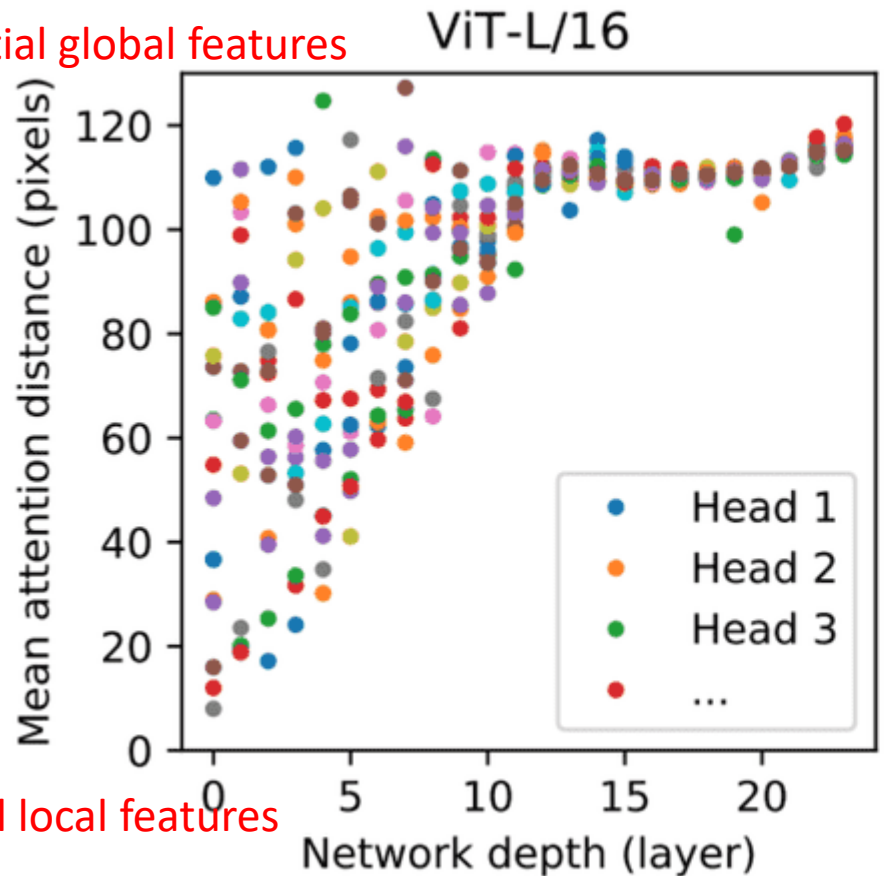
early layer representations may share similar features



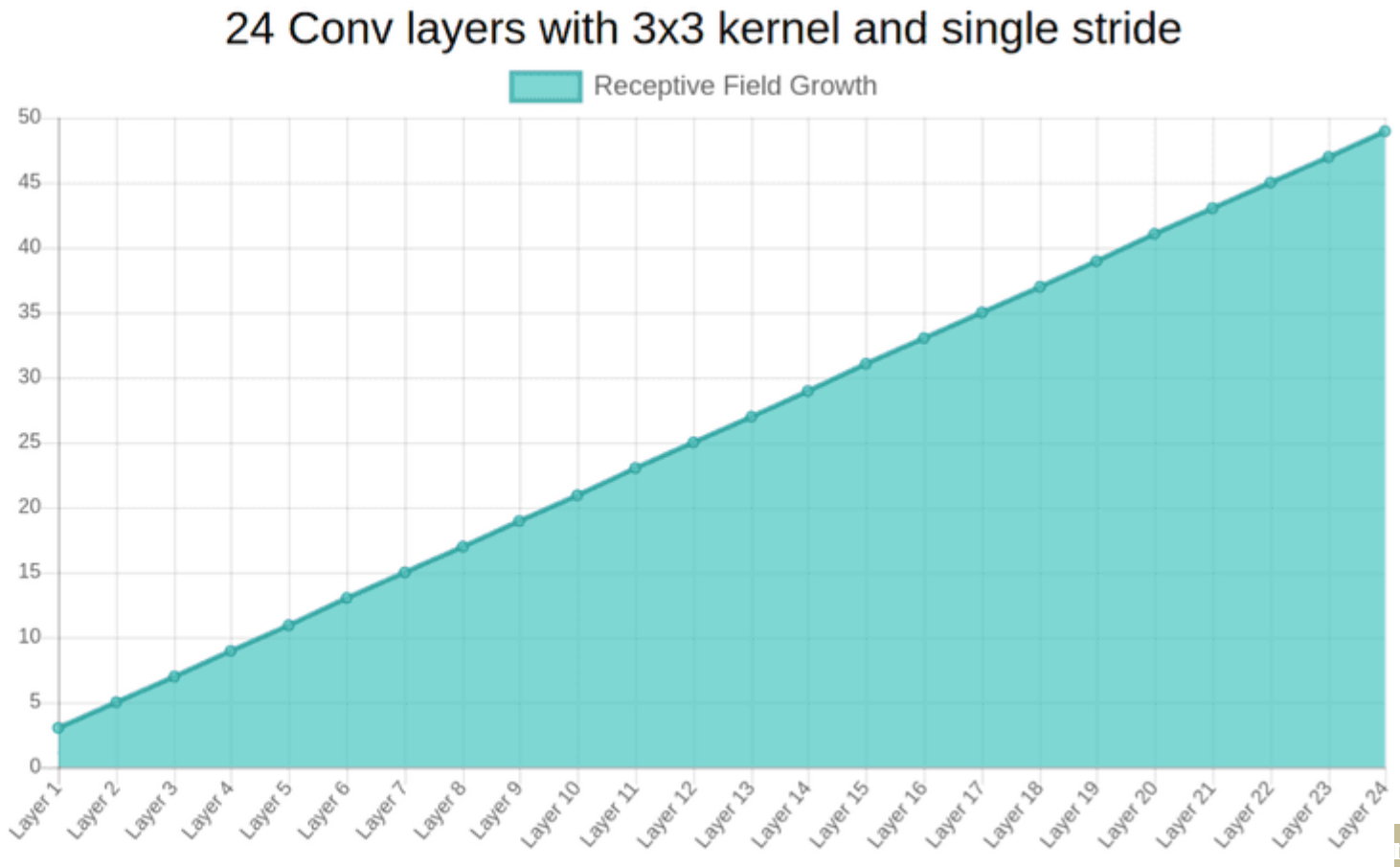
1. Split an image into patches
2. Flatten the patches
3. Produce lower-dimensional linear embeddings from the flattened patches
4. Add positional embeddings
5. Feed the sequence as an input to a standard transformer encoder
6. Pretrain the model with image labels (fully supervised on a huge dataset)
7. Finetune on the downstream dataset for image classification

- How far away are the learned non-local interactions?
  - For patch size P, maximum P*P, which in our case is 128, even from the 1st laver!

Spatial global features



Spatial local features



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

	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

Intriguing Properties of Vision Transformers

- Robust to
  - occlusion, distribution shift, adversarial patch, patch permutation

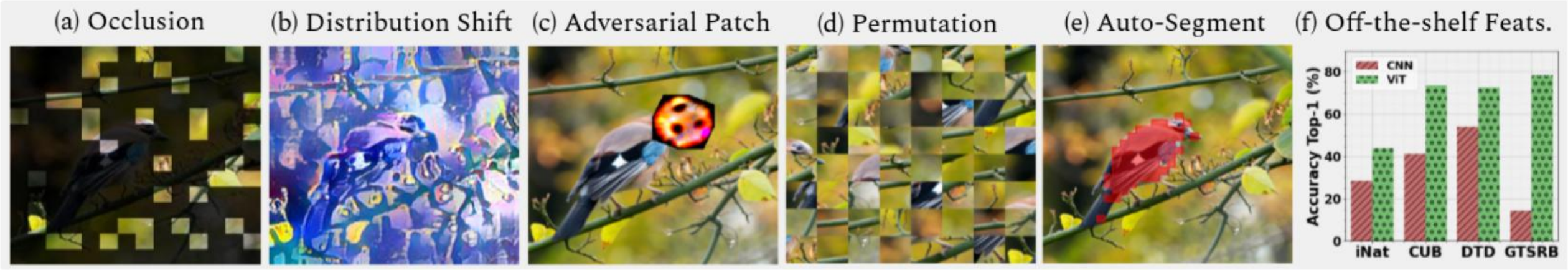


Figure 1: We show intriguing properties of ViT including impressive robustness to (a) severe occlusions, (b) distributional shifts (*e.g.*, stylization to remove texture cues), (c) adversarial perturbations, and (d) patch permutations. Furthermore, our ViT models trained to focus on shape cues can segment foregrounds without any pixel-level supervision (e). Finally, off-the-shelf features from ViT models generalize better than CNNs (f).

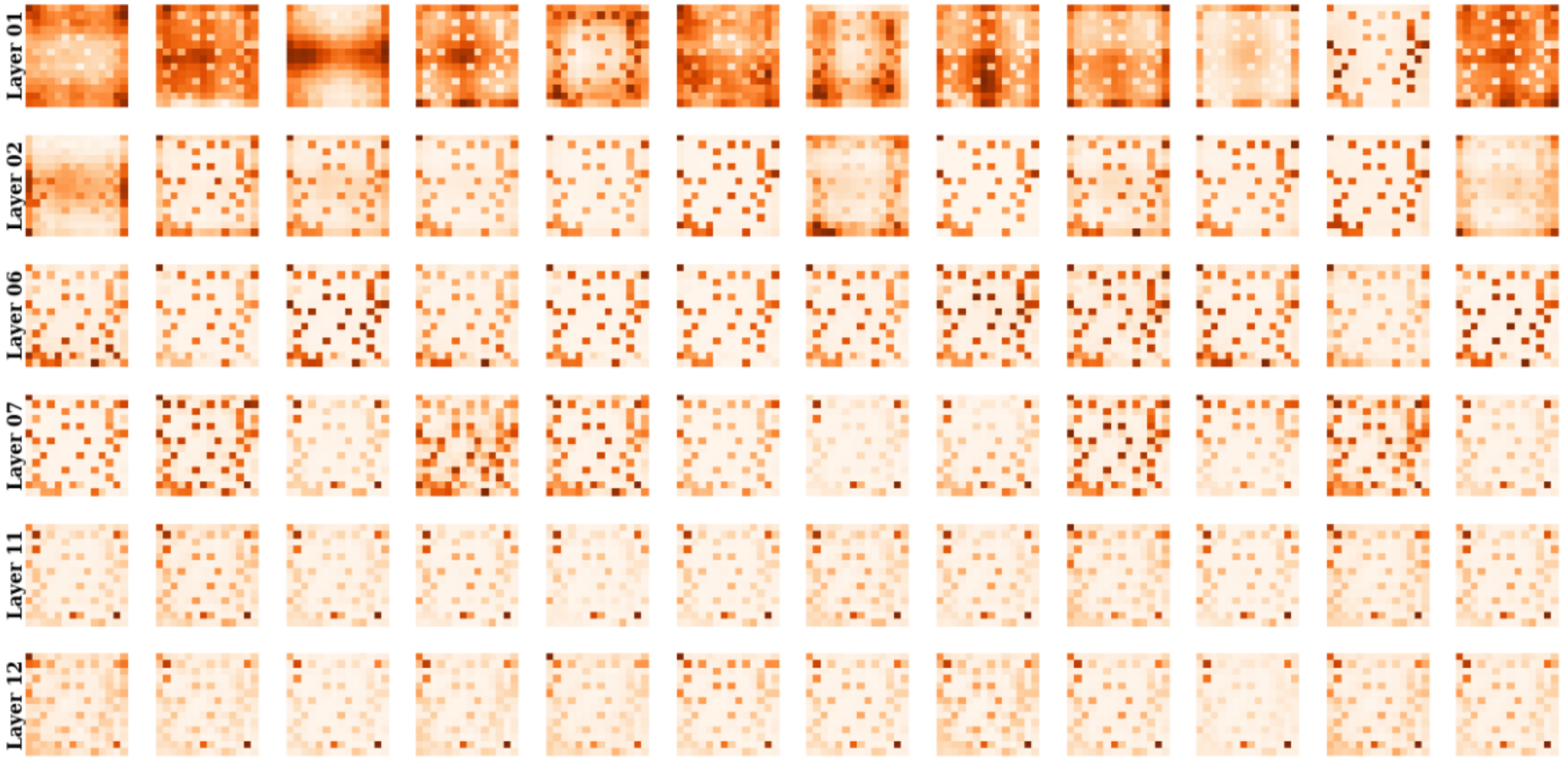


Figure 4: Attention maps (averaged over the entire ImageNet val. set) relevant to each head in multiple layers of an ImageNet pre-trained DeiT-B model. All images are occluded (Random PatchDrop) with the same mask (bottom right). Observe how later layers clearly attend to non-occluded regions of images to make a decision, an evidence of the model's highly dynamic receptive field.

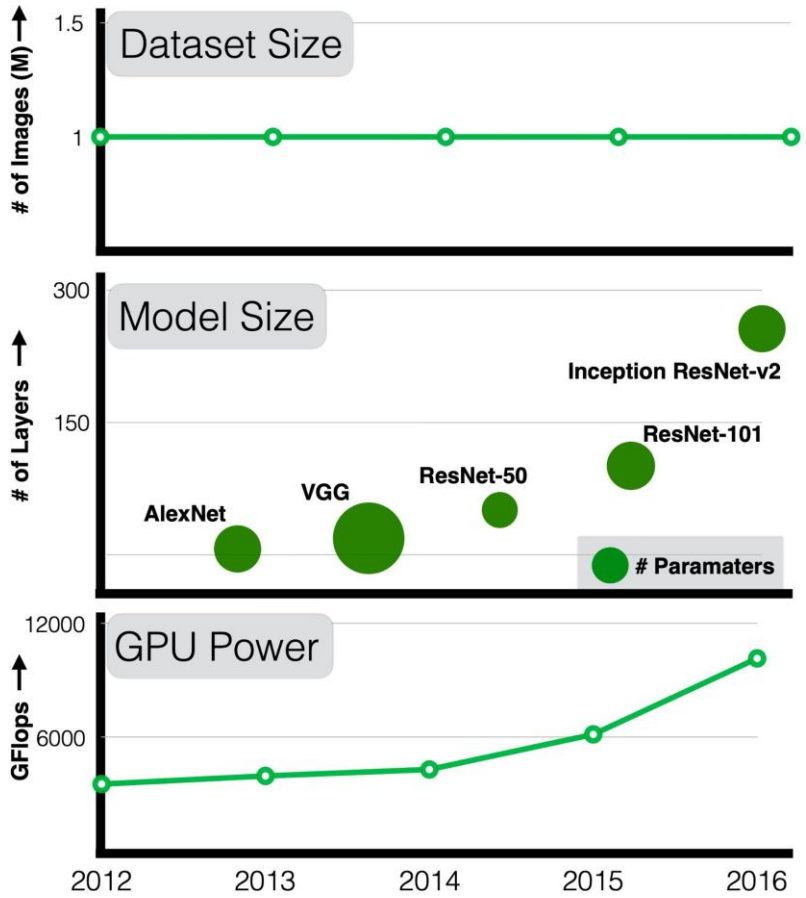
DISCUSSIONS ON VIT AND VARIANTS

Key properties of CNNs & Transformers

Property	CNNs	Vision Transformer
Shift invariance	Mostly	No
Permutation invariance	No	Mostly (at the patch level)
Spatially local processing	Yes 3x3, 5x5, etc. convs	Partially (only patching)
Parameter sharing	Yes	Yes
With increased depth: **Lower resolution,** **Wider (richer) features**	Yes Most have “pyramid” design	No

1. With sufficient data, can Transformers *learn* to overcome lack of intuitive properties (inductive biases)?
2. If so, does it increase training cost? (Is it useful?)

Rewind: Explorations in training with more images

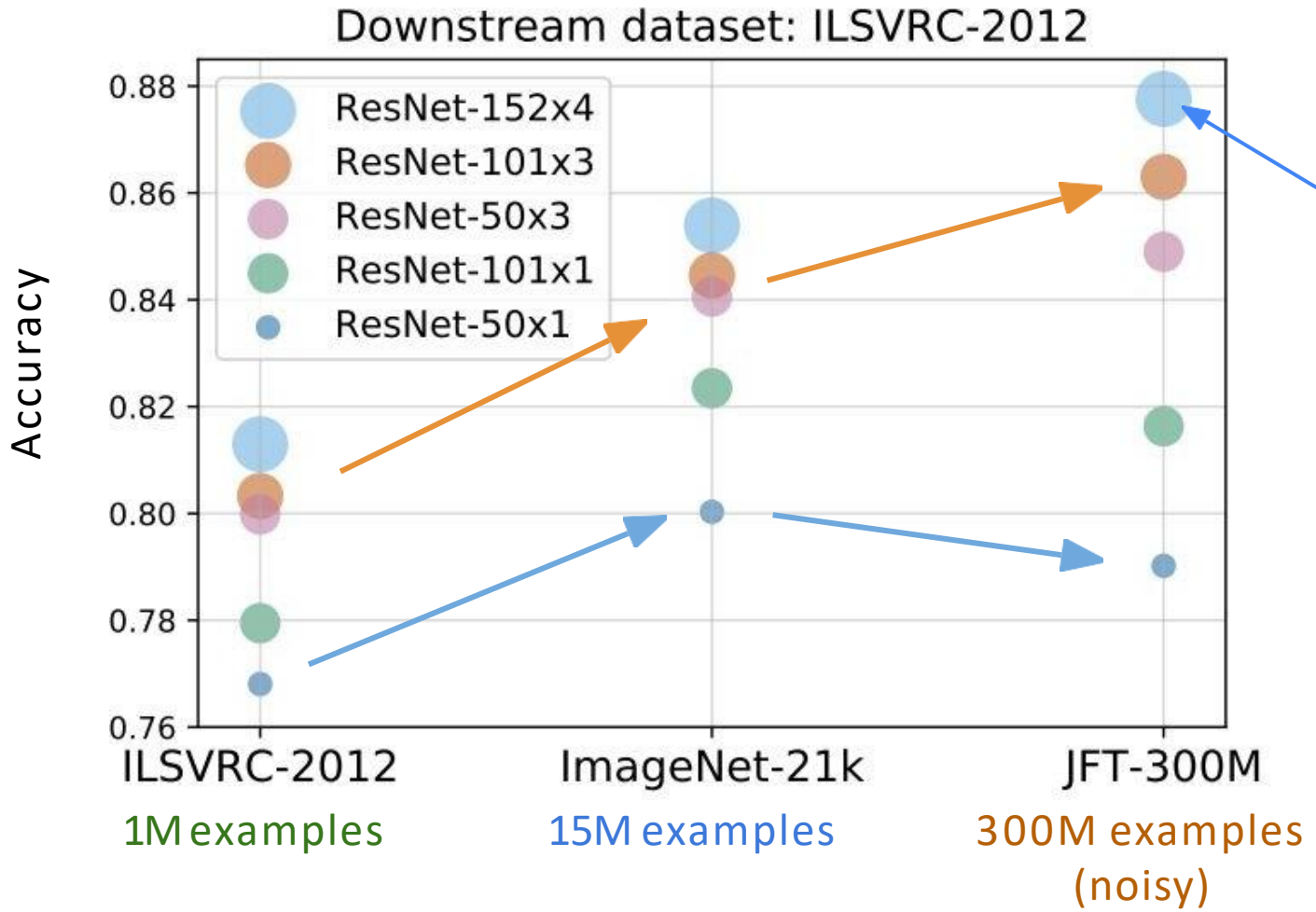


Sun et al. 2017

- Sun et al. obtained 79.2% ImageNet pre-training on 300M weakly-labelled images
- "Exploring the Limits": Pre-train on 3.5B hashtags [Mahajan et al. 2018]
- "Noisy Student": ImageNet pseudo-labelling on 300M images [Xie et al. 2019]
- "Big Transfer": Scaling recipe for transfer [Kolesnikov et al., 2019]

Sun et al., Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, ICCV 2017
Mahajan et al., Exploring the Limits of Weakly Supervised Pretraining, ECCV 2018
Xie et al., Self-training with Noisy Student improves ImageNet classification, CVPR 2020
Kolesnikov et al., Big Transfer (BiT): General Visual Representation Learning, ECCV 2020

Large data is useful iff one uses a high-capacity network



"Big Transfer"
large CNNs

Kolesnikov et al. 2019

Kolesnikov et al., Big Transfer (BiT): General Visual Representation Learning, ECCV 2020

Success in NLP with high-capacity Transformers

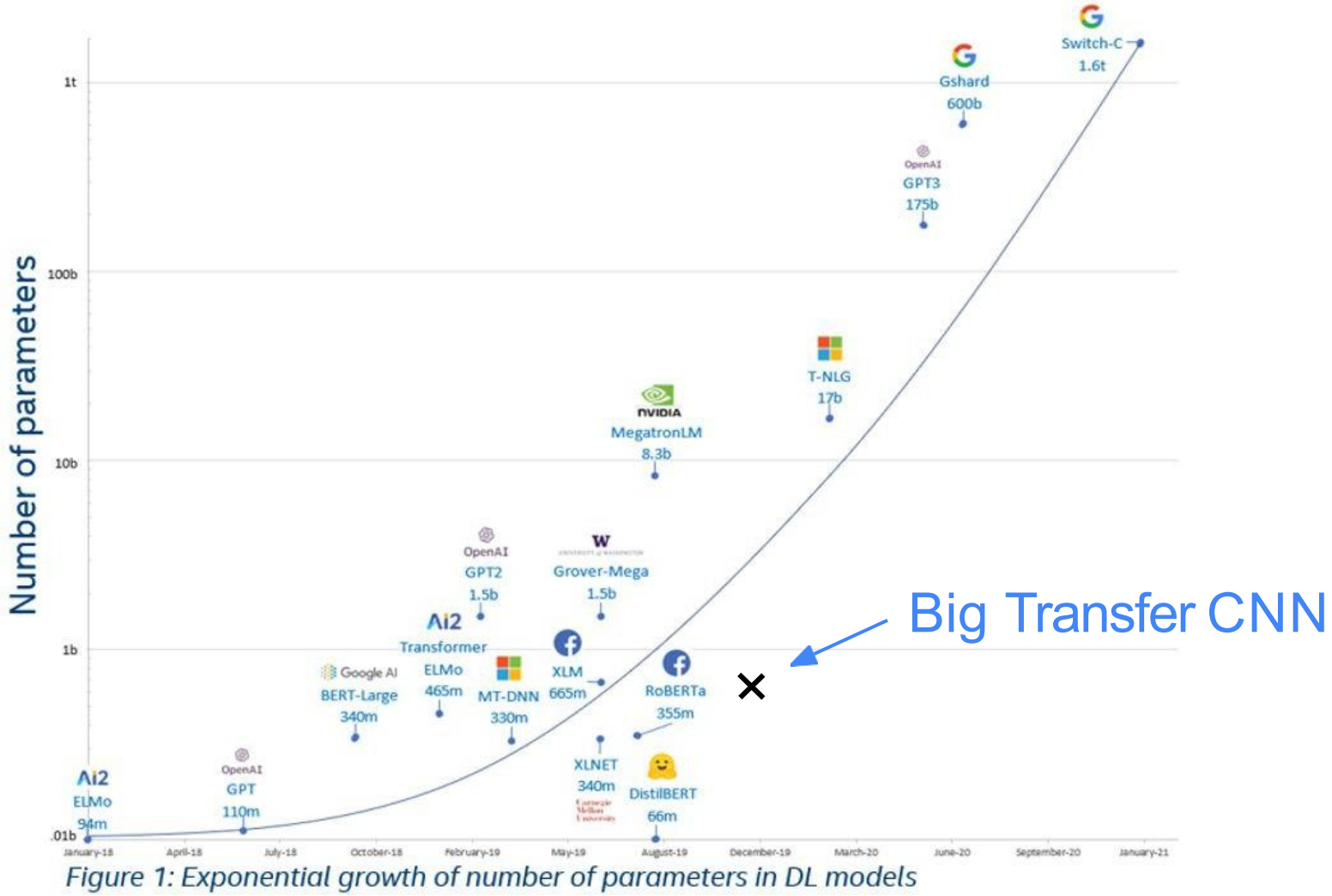
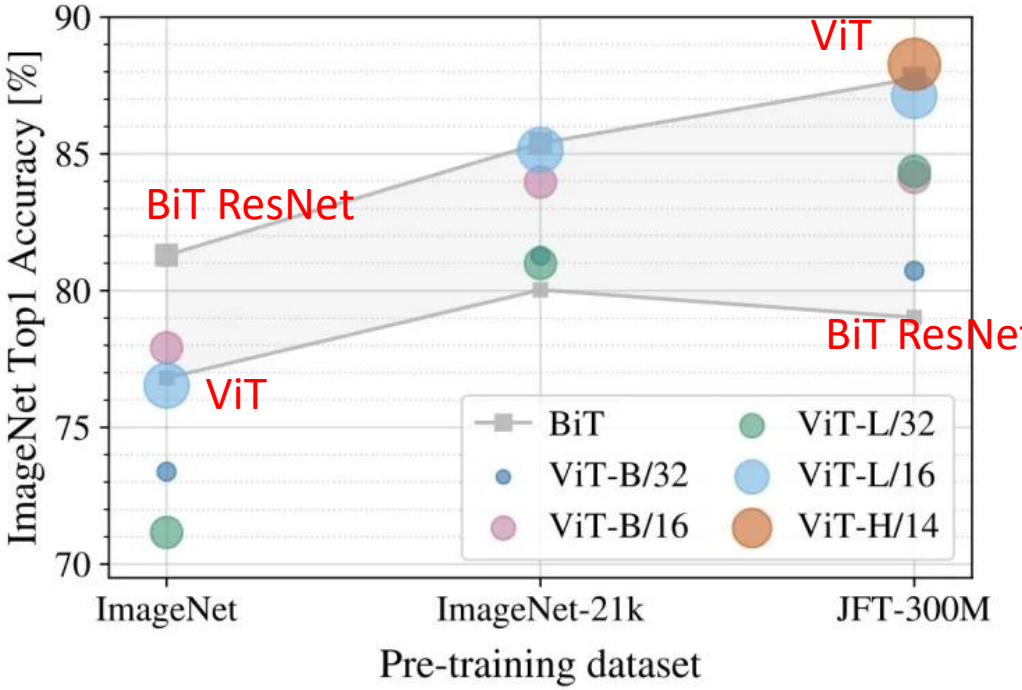


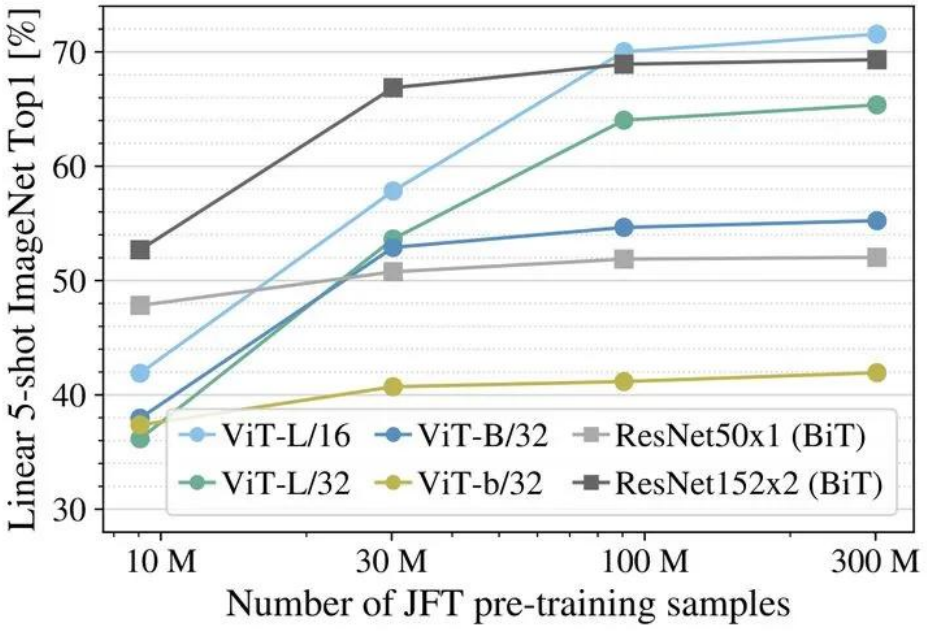
Figure 1: Exponential growth of number of parameters in DL models

Gadi Singer, https://towardsdatascience.com/the-rise-of-cognitive-ai-a29d2b724ccc , based on
Corby Rosset, https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/ and
Sanh et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. EMC^2, 2019

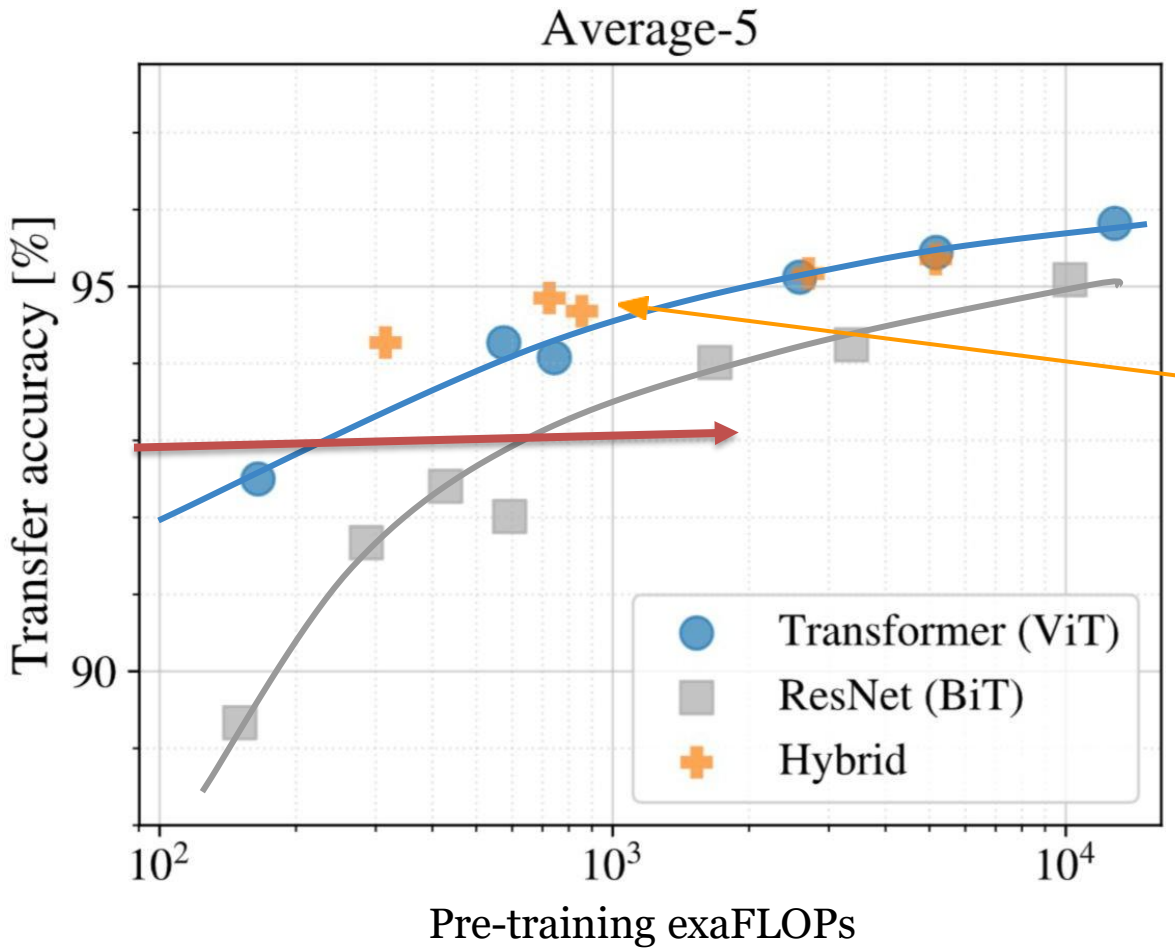
ViT

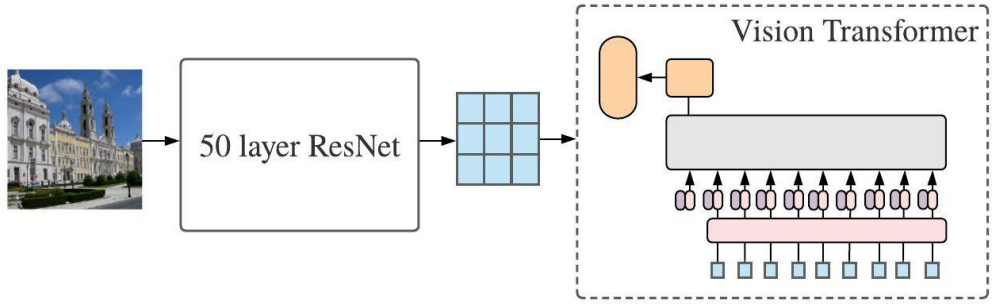
- Input: 16x16x3 patch
- Position embedding
  - Trainable
- Only encoder part of the original transformer
- Training
  - Need massive amount of data (14M images)
    - Overfit for ImageNet
  - 到了JFT 300M，我們才能看到更大的ViT模型全部優勢
  - ResNet對於較小的資料集是有用的，但是對於較大的資料集，像attention一樣學習相關性就足夠了，甚至是更好的選擇
- 原因
  - 由於缺乏像CNN這些歸納偏置，ViT 需要從大量的數據中學習才能獲得良好的性能。 在小規模數據集上，ViT 的性能通常不如 CNN，因為它無法像 CNN 那樣有效地利用圖像的局部性和平移等變性等先驗知識。
  - 用Deit的方法可以解決





Key result : Surprisingly, ViT costs less to pre-train

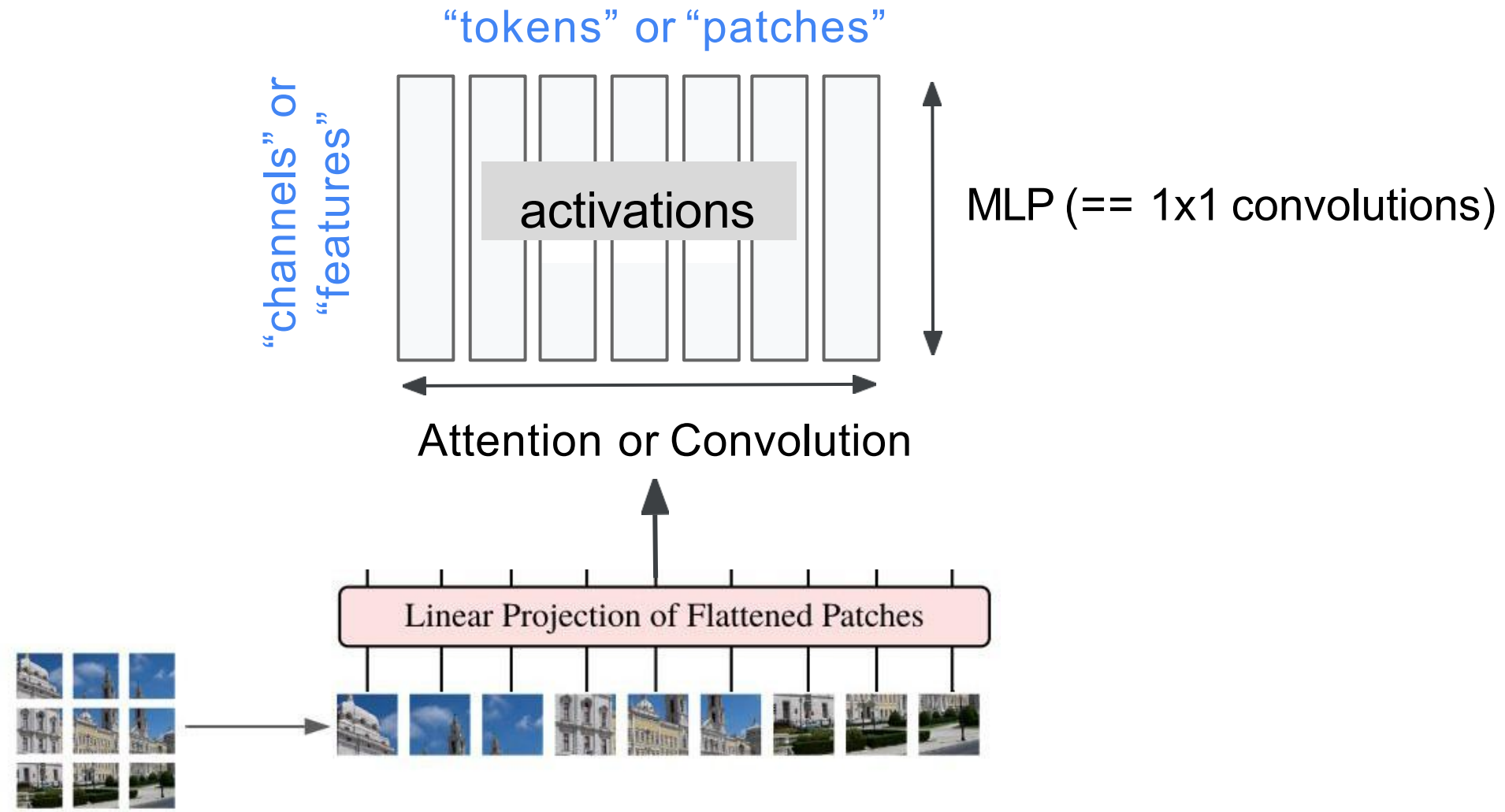




Hybrid effective at smaller scales

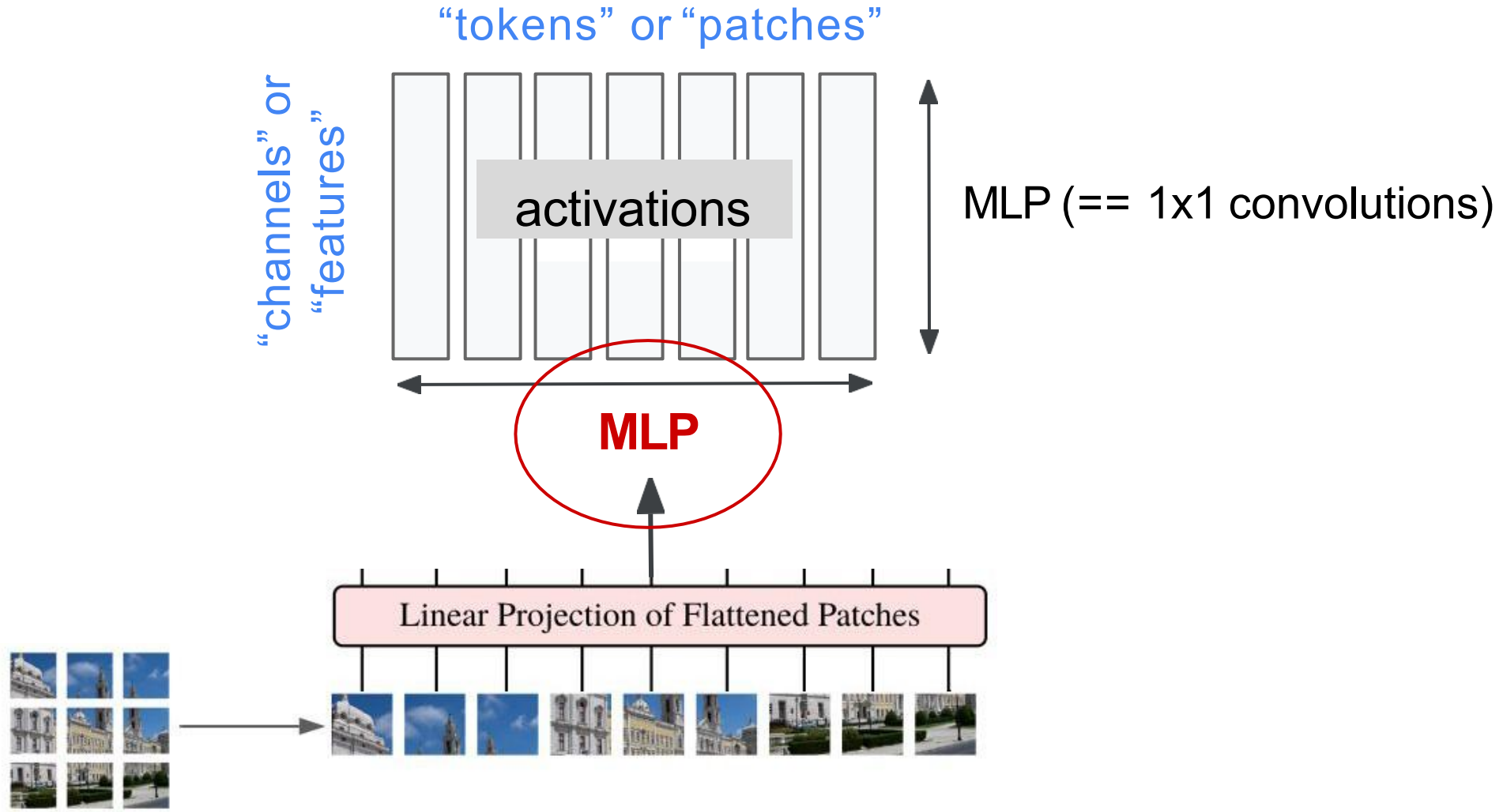
Controlled study: At all cost levels, ViT is cheaper to pre-train than ResNet

Is self-attention the only key ingredient?



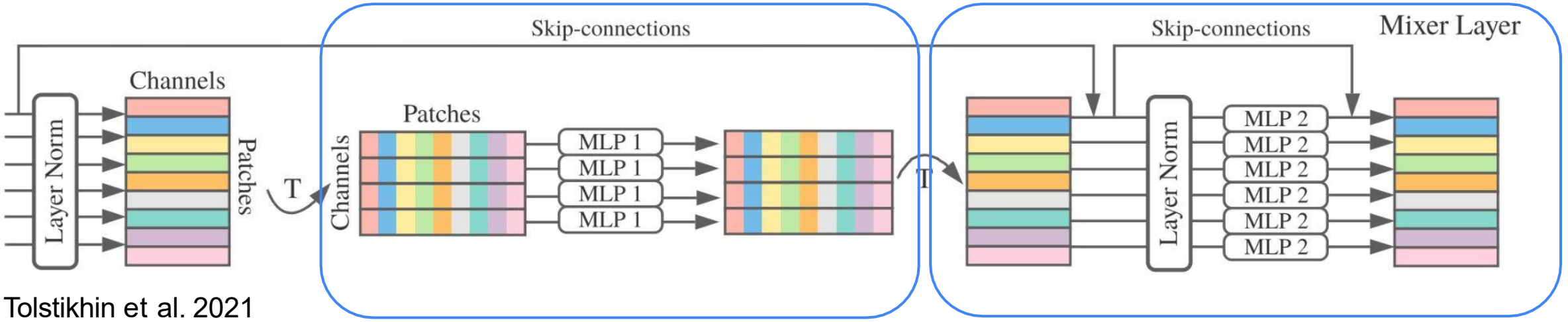
An oversimplified view of vision architectures containing: (1) Feature interactions (2) Spatial interactions

Is self-attention the only key ingredient?



An oversimplified view of vision architectures containing (1) Feature interactions (2) Spatial interactions

Fully-MLP based designs



Tolstikhin et al. 2021

"Transposed" MLP (instead of self-attention)

Standard Transformer MLP layer

- Multiple similar "all MLP" designs proposed concurrently [Tolstikhin et al. 2021, Touvron et al. 2021, Lui et al. 2021]
- MLP-Mixer similar to Vision Transformer (channel MLP instead of S.A. + details)

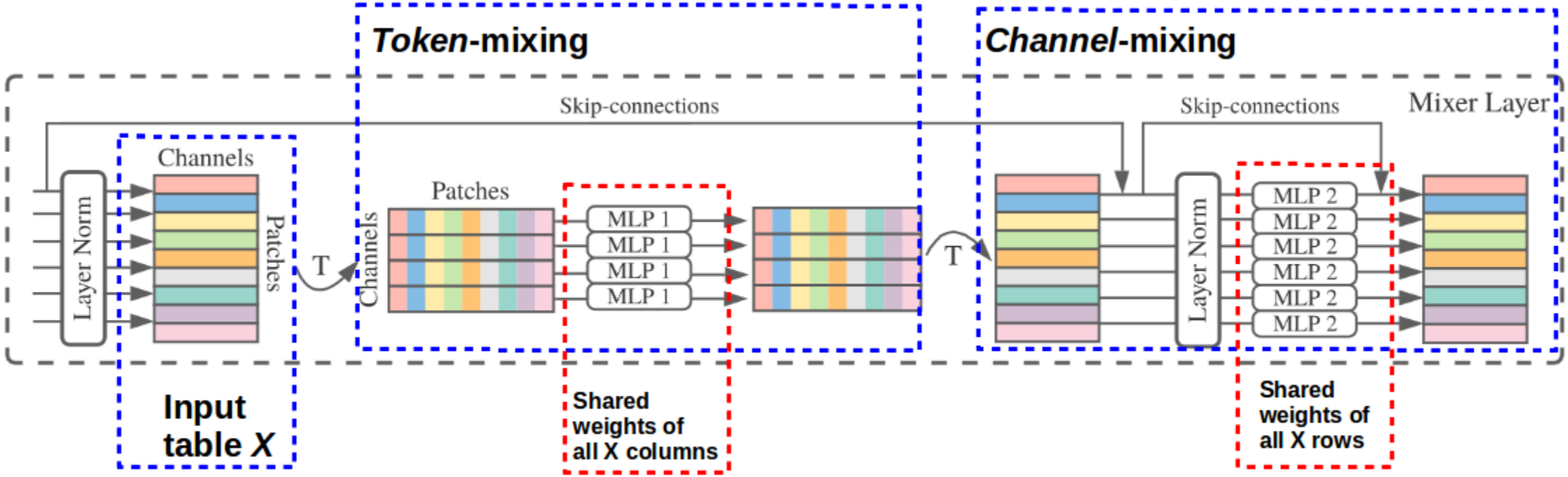
Tolstikhin et al., MLP-Mixer: An all-MLP Architecture for Vision, NeurIPS 2021
Touvron et al., ResMLP: Feedforward networks for image classification with data-efficient training, 2021
Liu et al., Pay Attention to MLPs, arXiv 2021

token-mixing MLPs層相當於廣義的depth-wise convolution

channel-mixing MLPs層相當於1x1 convolution

靠channel-mixing MLPs層結合不同channels的資訊

MLP-Mixer讓這兩種類型的層交替執行



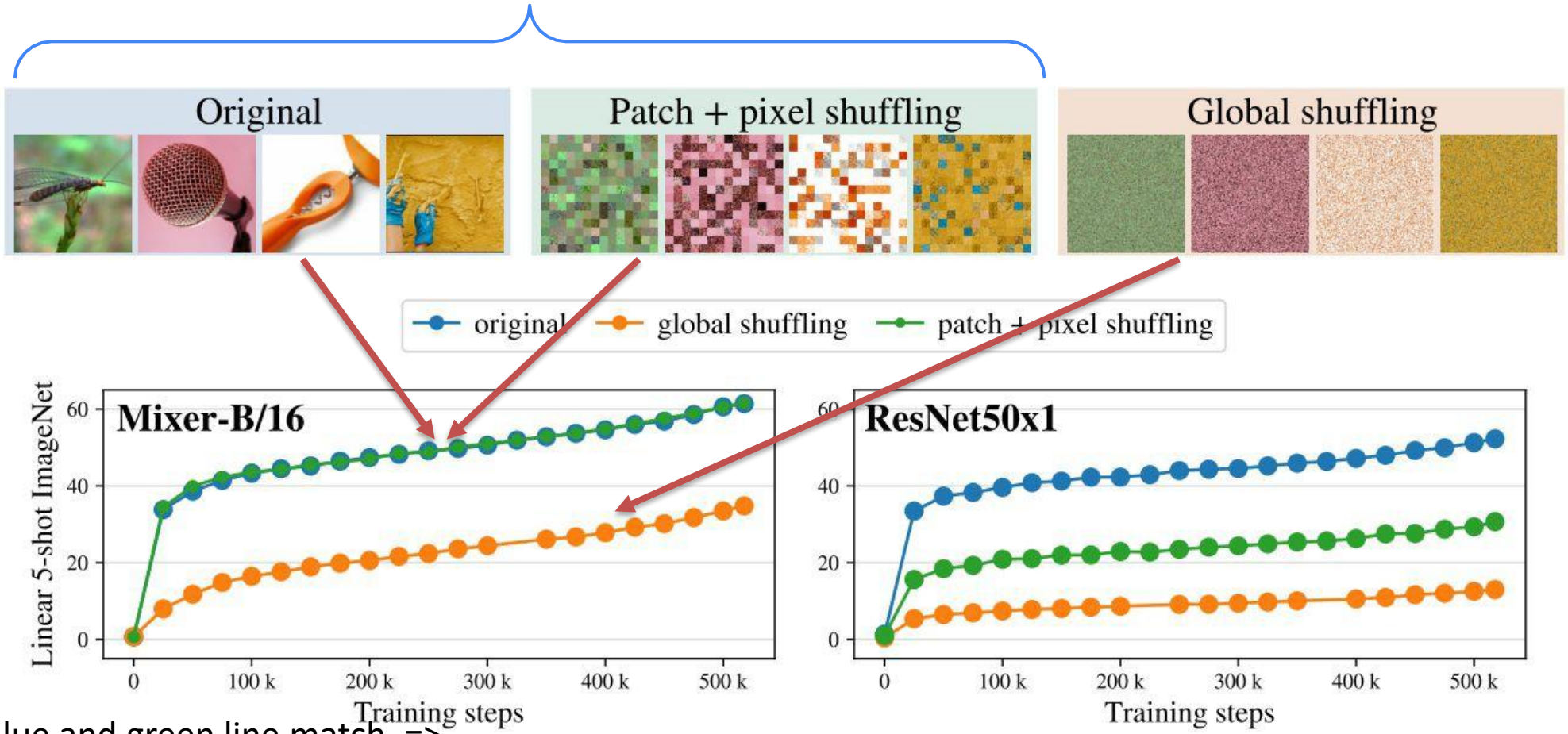
不同patch 同一位置的收集進入MLP

靠token-mixing MLPs層結合不同空間位置的資訊。

T: transpose

Inductive biases

MLP-based models consider these as equivalent

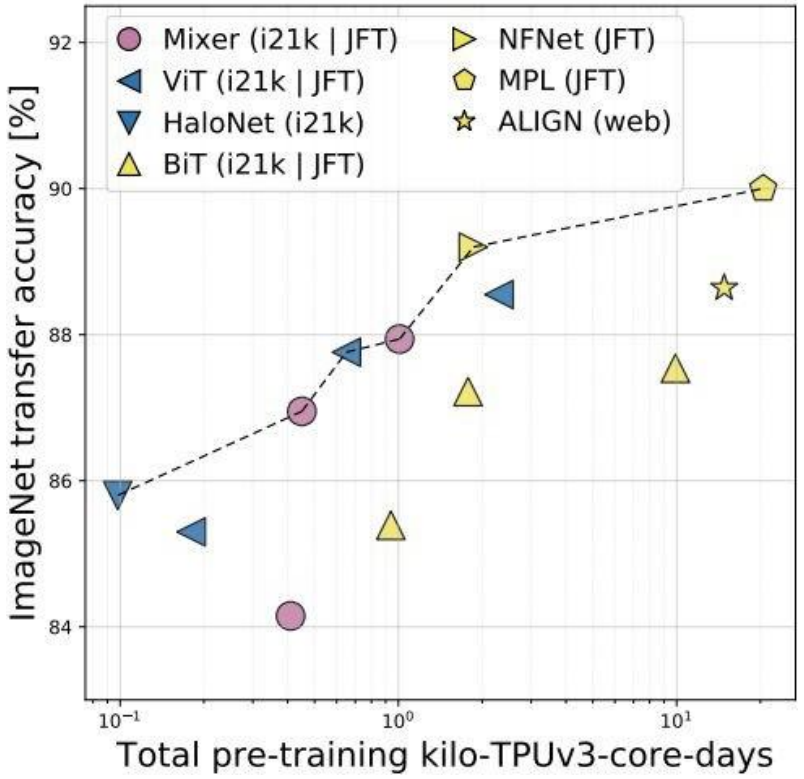


Blue and green line match, =>

Mixer is invariant to the order of patches and pixels within the patches

Tolstikhin et al., MLP-Mixer: An all-MLP Architecture for Vision, NeurIPS 2021

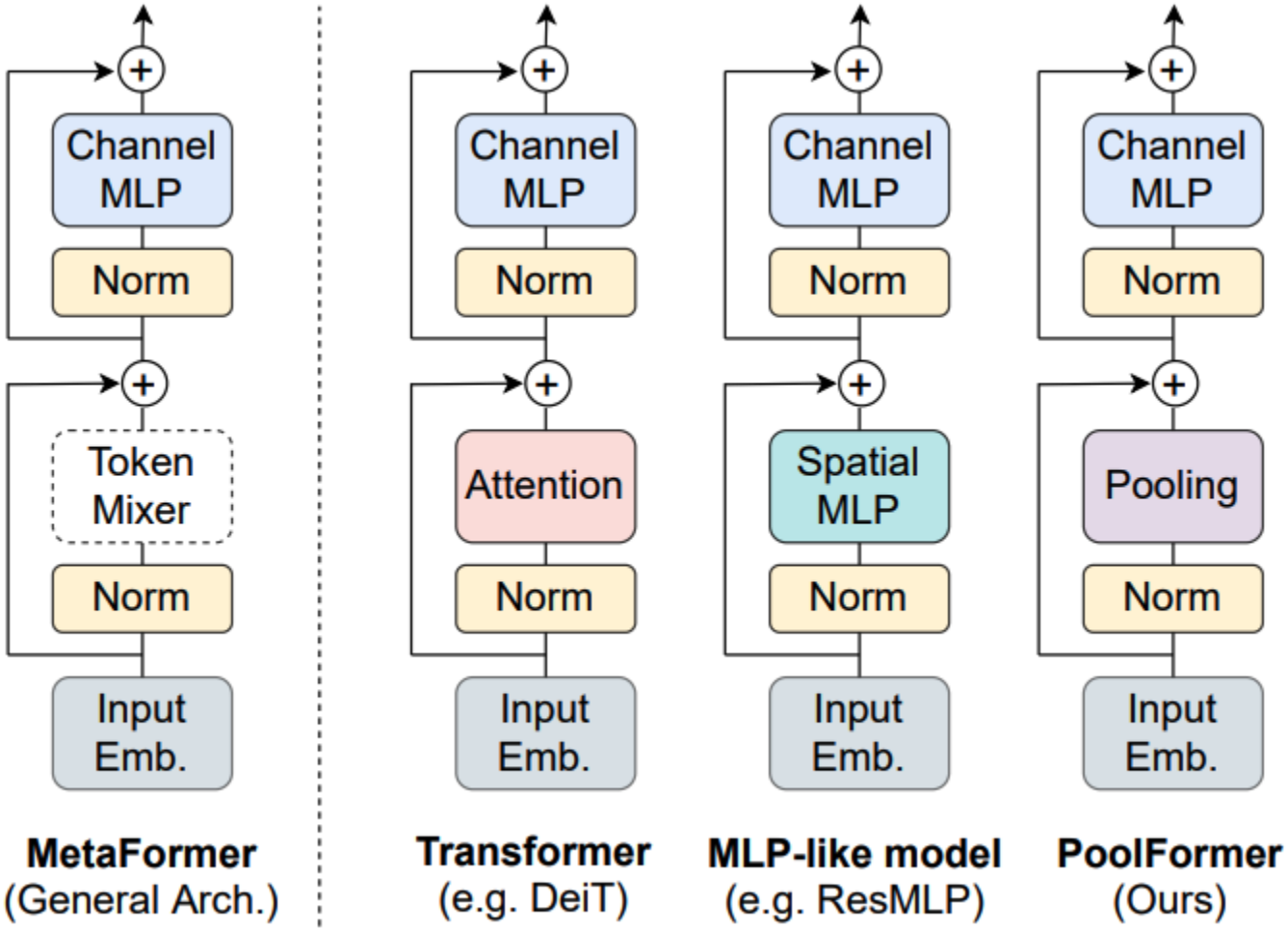
Key Result: MLPs can also learn good features!



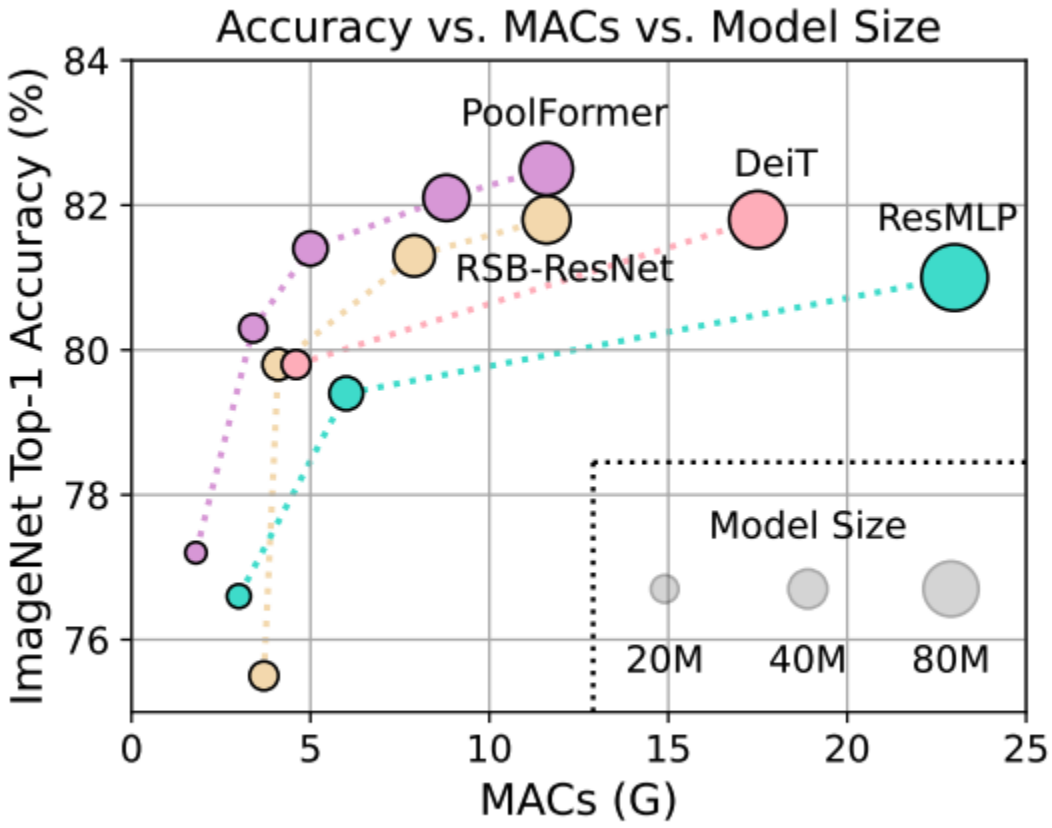
- ResMLP [Touvron el al. 2021] achieve a **81.0% ImageNet** (from scratch), just slightly behind an equivalent Transformer
- MLP-Mixer [Tolstikhin et al. 2021] shows equivalent performance vs. cost to ViT in the a controlled study. Plus similar performance to SOTA at large scale (**87.9% ImageNet**)

Tolstikhin et al., MLP-Mixer: An all-MLP Architecture for Vision, NeurIPS 2021
Touvron et al., ResMLP: Feedforward networks for image classification with data-efficient training, 2021

- Poolformer (metaformer)



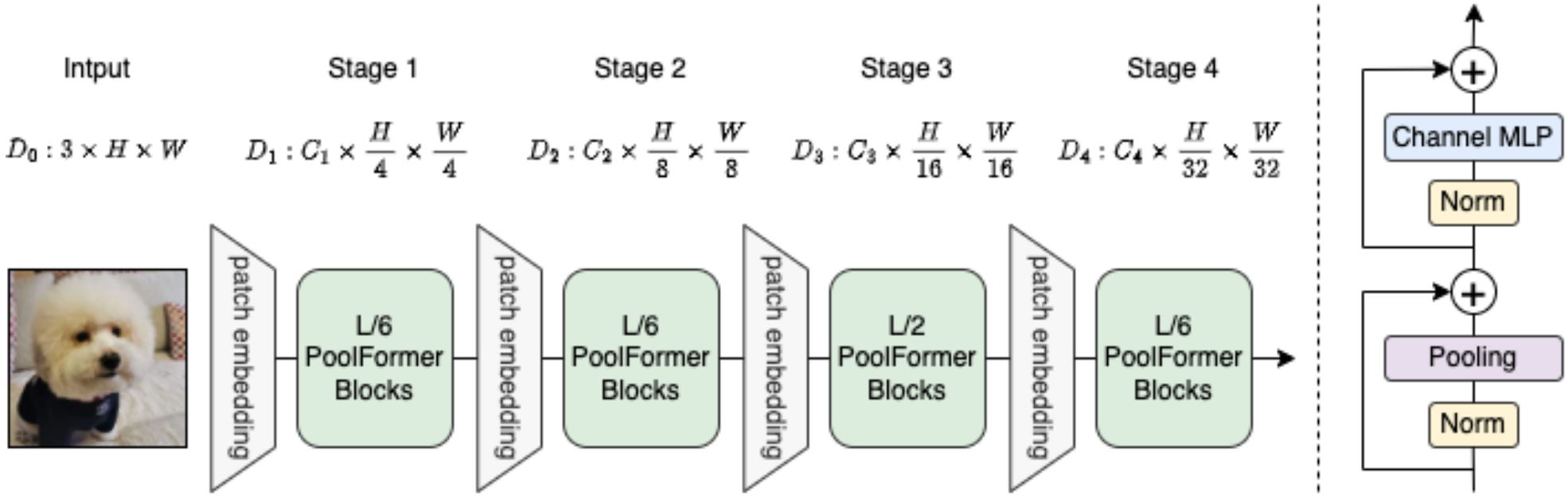
(a)



(b)

Pooling as Attention

- Poolformer

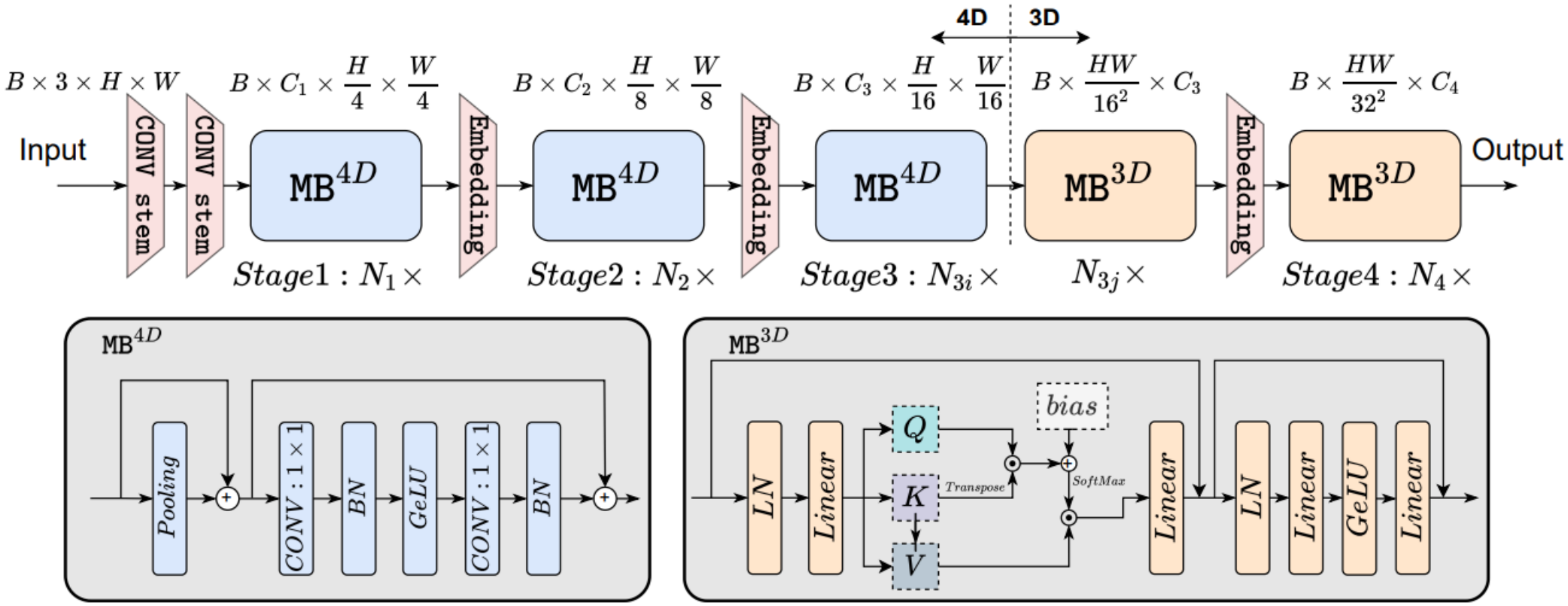


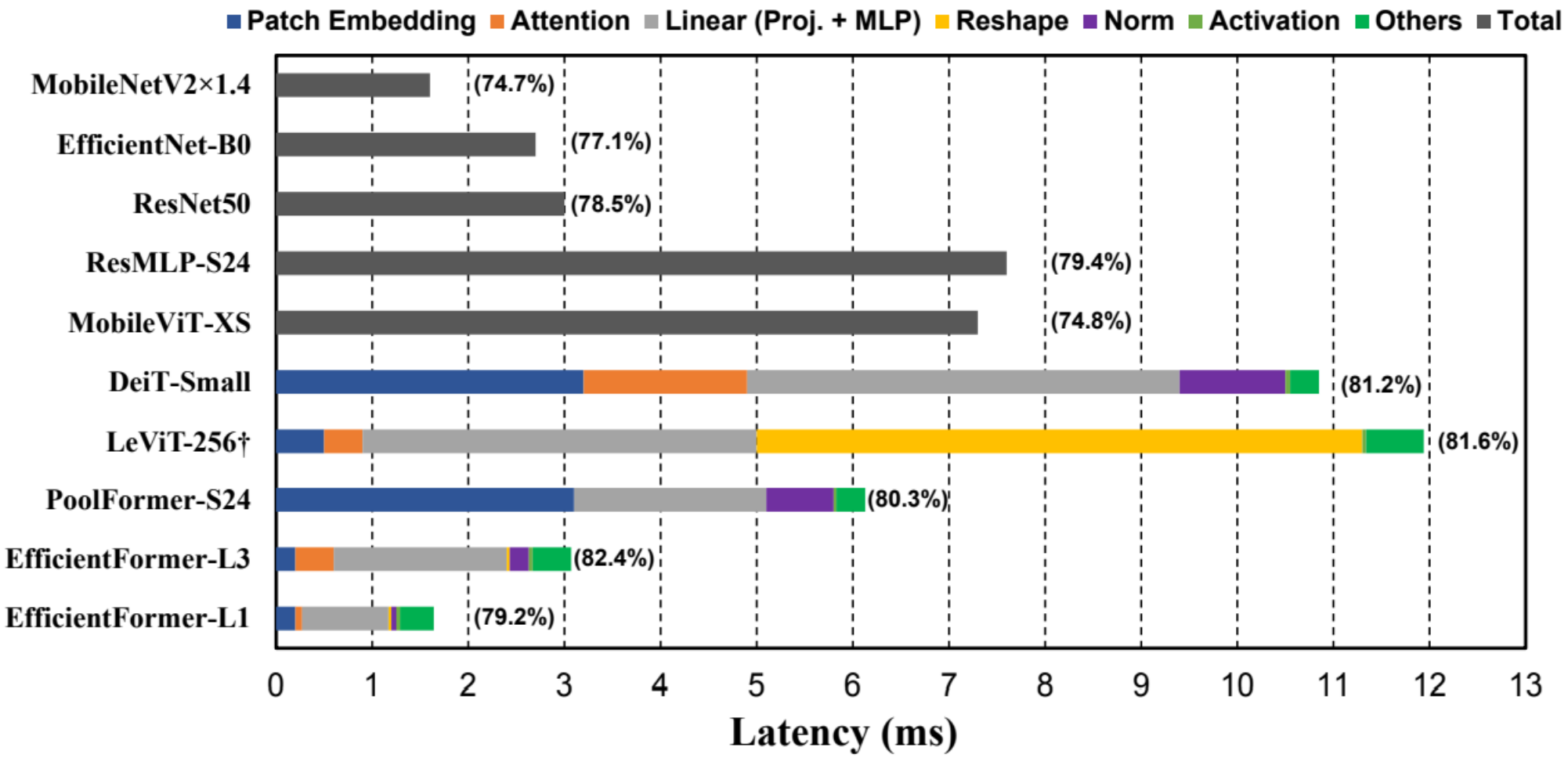
(a) Overall framework with L PoolFormer blocks

(b) PoolFormer block

Pooling as Attention

- EfficientFormer
  - LN-linear => Conv-BN





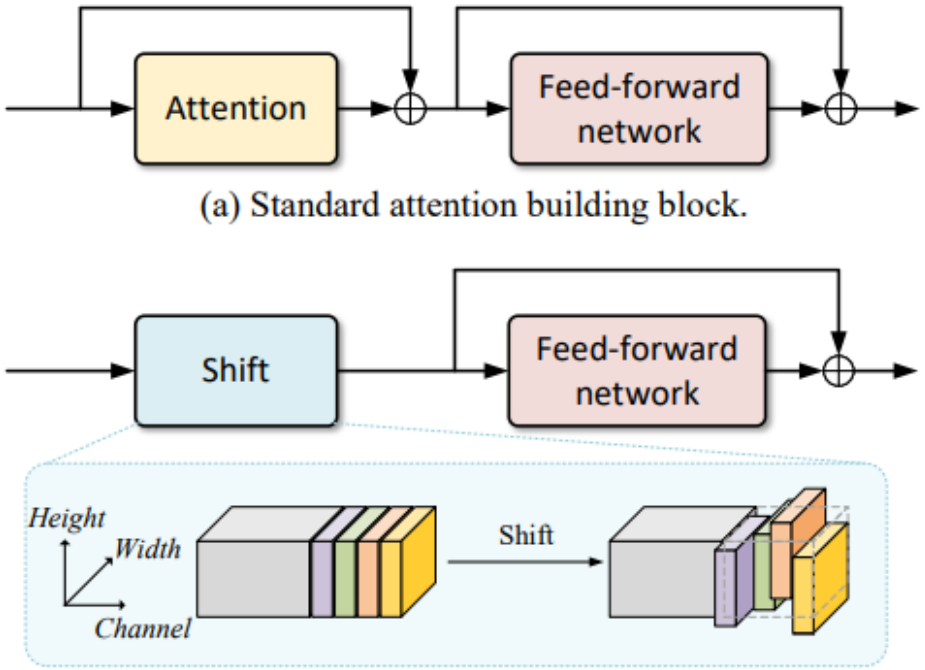
Patch embedding with large kernel and stride is a speed bottleneck on mobile devices.

Observation 2: Consistent feature dimension is important for the choice of token mixer. MHSA is not necessarily a speed bottlen

Observation 3: CONV-BN is more latency-favorable than LN (GN)-Linear and the accuracy drawback is generally acceptable. the latency introduced by LN constitutes around 10% − 20% latency of the whole network.

Observation 4: The latency of nonlinearity is hardware and compiler dependent.

Self Attention => Shift

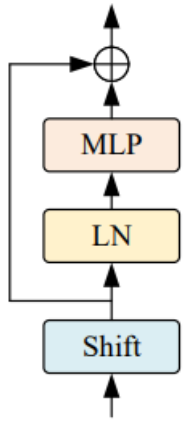


(a) Standard attention building block.

(b) Our shift building block.

Figure 1: An illustration of our shift building block. We propose to replace the attention layer with a simple shift operation in vision transformers. It spatially shifts a small portion of the channels along four directions, and the rest of the channels remain unchanged.

exchange a small portion of the channels between neighboring features.



(b) shift block

ViT-based and MLP-based				
DeiT-S	224^2	22	4.6	79.8
DeiT-B	224^2	86	17.5	81.8
PVT-S	224^2	25	3.8	79.8
PVT-L	224^2	61	9.8	81.7
Swin-T	224^2	29	4.5	81.3
Swin-S	224^2	50	8.7	83.0
Swin-B	224^2	88	15.4	83.5
MLP-Mixer-B/16	224^2	79	-	76.4
gMLP-S	224^2	20	4.5	79.4
gMLP-B	224^2	73	15.8	81.6
S^2-MLP-D	224^2	71	14.0	80.0
S^2-MLP-W	224^2	51	10.5	80.7
AS-MLP-T	224^2	28	4.4	81.3
AS-MLP-S	224^2	50	8.5	83.1
AS-MLP-B	224^2	88	15.2	83.3
Ours				
Shift-T	224^2	28	4.4	81.7
Sfhit-S	224^2	50	8.5	82.8
Sfhit-B	224^2	88	15.2	83.3

Table 2: Comparison with state-of-the-art methods on the ImageNet-1k classification task.

When Shift Operation Meets Vision Transformer:
An Extremely Simple Alternative to Attention Mechanism

SGD ↓ AdamW	ReLU ↓ GELU	BN ↓ LN	90ep ↓ 300ep	ImageNet Top-1 Acc. (%)
				76.4
✓				77.9
✓	✓			78.5
✓	✓	✓		78.4
✓	✓	✓	✓	81.7

Table 6: Ablation analysis on the typical configurations of CNNs and ViTs. We gradually transfer the training configuration from the CNN's setting to the ViT's setting, and investigate how these factors influence the model performances.

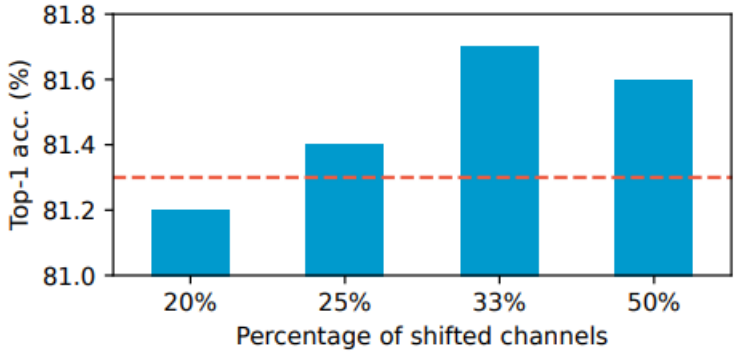
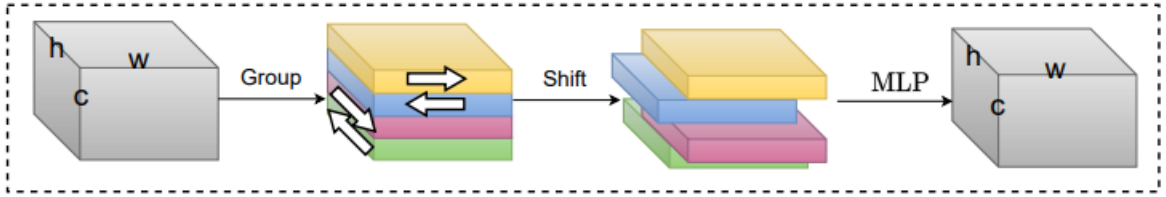
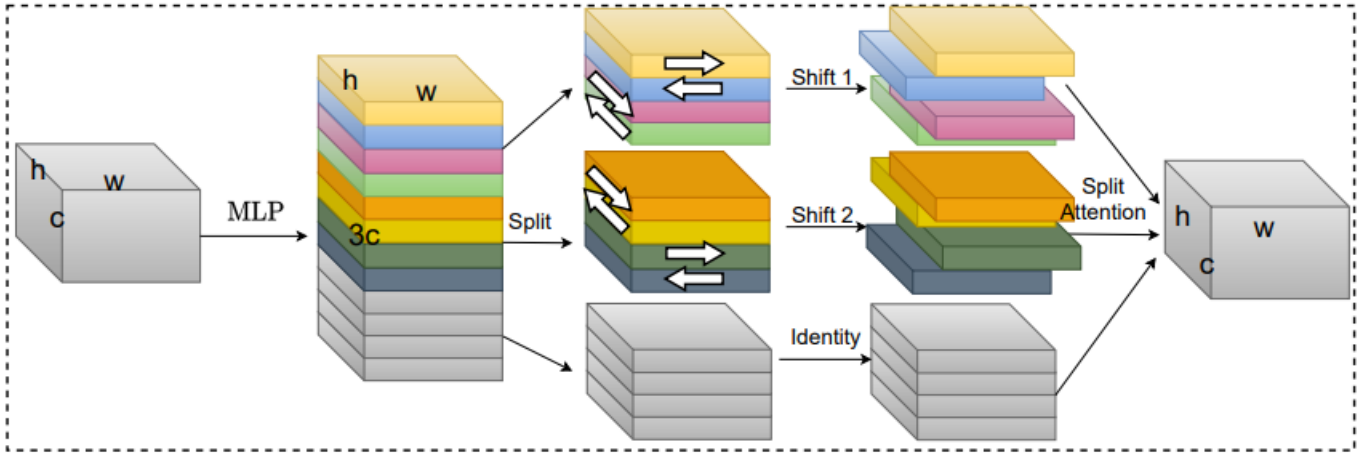


Figure 3: Ablation analysis on the percentage of shifted channels. We plot the top-1 classification accuracy on ImageNet-1k. The red line indicates Swin-T baseline.

S2-MLPv2: Improved Spatial-Shift MLP Architecture for Vision



(a) Spatial-shift operations in S^2-MLP

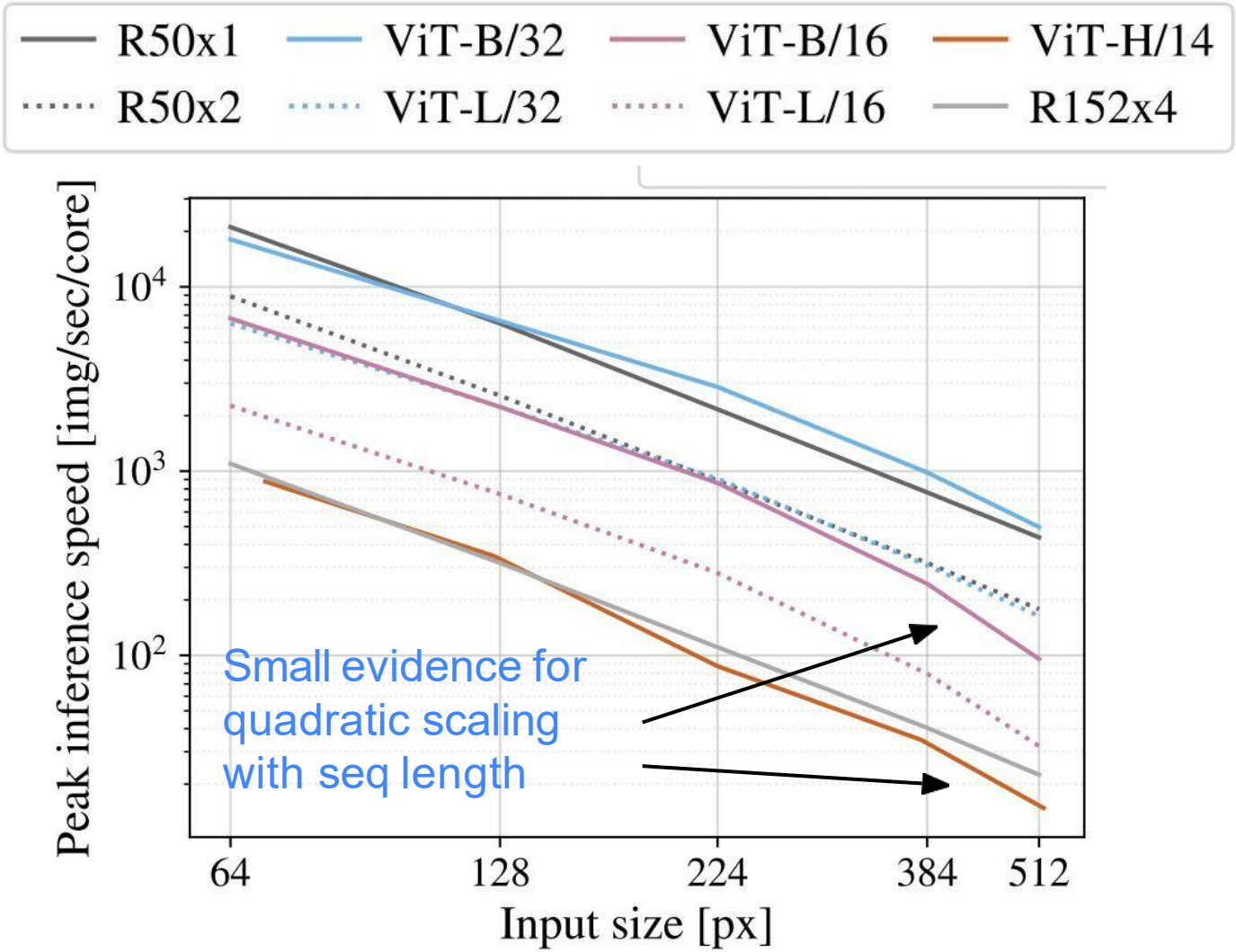


(b) Spatial-shift operations in S^2-MLPv2

Figure 1: Comparisons between the spatial-shift operations in S^2-MLP (Yu et al., 2021b) and the proposed S^2-MLPv2. In S^2-MLP, the channels are equally divided into four parts, and each part shifts along different directions. An MLP is conducted on the shifted channels. In contrast, in S^2-MLPv2, the c-channel feature map is expanded into the $3c$-channel feature map. Then the expanded map is equally split into three parts along the channel dimension. For each part, we conduct different spatial-shift operations. Then the shifted parts are merged through the split-attention operation (Zhang et al., 2020) to generate the c-channel feature map.

Aside: Practical caveats regarding resolution

- Resizing
  - Fully-convolutional nets applicable to any resolution
  - Transformers require resizing position embeddings (straightforward)
  - MLPs require custom re-initialization & finetuning (less straightforward)
- Quadratic scaling of attention
  - Generally not an issue for classification
  - Trivial solution: larger patches



Dosovitskiy et al. 2020

Summary so far

- Transformers can learn great features for vision (with data)
- Attention not the only key ingredient (but useful)
- Useful inductive biases not always intuitive
- ***Do such architectures only work in the large transfer-learning regime?***

Evolutions

1. Data-efficient (pre-)training
2. The ViT-CNN spectrum
3. Scaling up

(Pre-)Training with less data

- Data augmentation
- Better regularization
- Distillation (CNN Teacher)
- Optimization

(Pre-)Training with less data

Methods	ViT-B [15]	DeiT-B
Epochs	300	300
Batch size	4096	1024
Optimizer	AdamW	AdamW
learning rate	0.003	$0.0005 \times \frac{\text{batchsize}}{512}$
Learning rate decay	cosine	cosine
Weight decay	0.3	0.05
Warmup epochs	3.4	5
Label smoothing ε	✗	0.1
Dropout	0.1	✗
Stoch. Depth	✗	0.1
Repeated Aug	✗	✓
Gradient Clip.	✓	✗
Rand Augment	✗	9/0.5
Mixup prob.	✗	0.8
Cutmix prob.	✗	1.0
Erasing prob.	✗	0.25

Augmentation and regularization strategies of Touvron et al. 2021

- Touvron et al. 2021: "DeiT" recipe (aug+reg+distillation) obtain competitive ImageNet from-scratch performance (85.2% using "Base" sized network, 83.1% w/o distillation)
- **Important:** When evaluating ***architectures***, be very careful to control for augmentation/regularization

Touvron et al., Training data-efficient image transformers & distillation through attention, ICML 2021

(Pre-)Training with less data

orig

aug

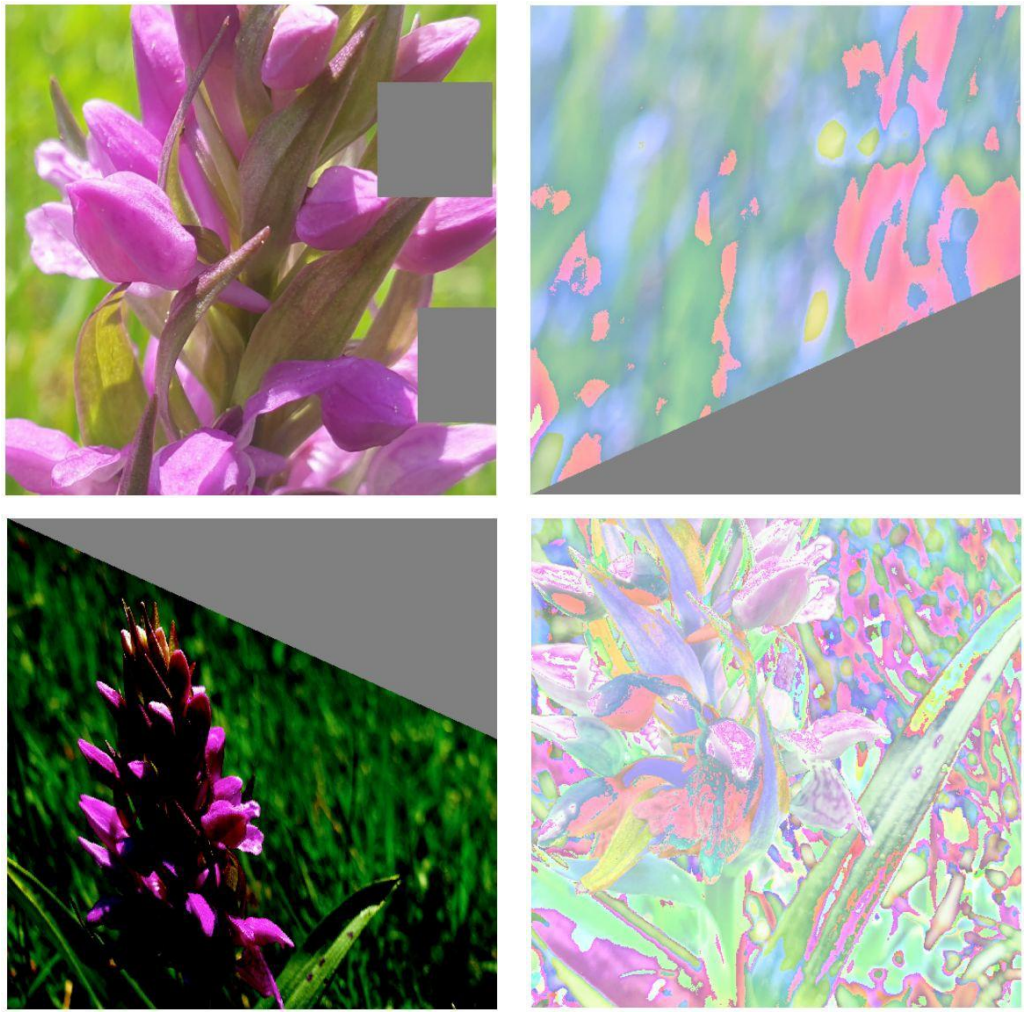


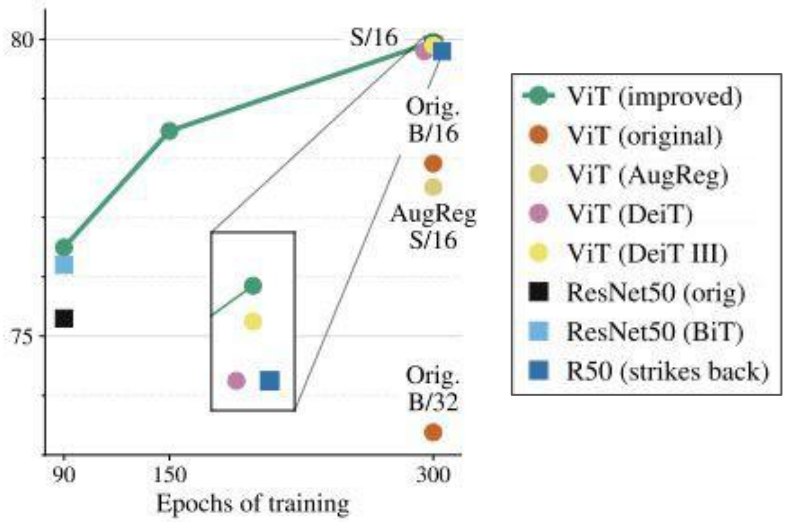
image by Neil Houlsby & Marie Escudié

(Pre-)Training with less data: Simplification

Procedure → / Reference	Previous approaches: ViT [13]	Steiner et al. [42]	DeiT [48]	Wightman et al. [57]	Ours: ImNet-1k	ImNet-21k Pretrain.	ImNet-21k Finetune.
Batch size	4096	4096	1024	2048	2048	2048	2048
Optimizer	AdamW	AdamW	AdamW	LAMB	LAMB	LAMB	LAMB
LR	3.10^{-3}	3.10^{-3}	1.10^{-3}	5.10^{-3}	3.10^{-3}	3.10^{-3}	3.10^{-4}
LR decay	cosine	cosine	cosine	cosine	cosine	cosine	cosine
Weight decay	0.1	0.3	0.05	0.02	0.02	0.02	0.02
Warmup epochs	3.4	3.4	5	5	5	5	5
Label smoothing ε	0.1	0.1	0.1	✗	✗	0.1	0.1
Dropout	✓	✓	✗	✗	✗	✗	✗
Stoch. Depth	✗	✓	✓	✓	✓	✓	✓
Repeated Aug	✗	✗	✓	✓	✓	✗	✗
Gradient Clip.	1.0	1.0	✗	1.0	1.0	1.0	1.0
H. flip	✓	✓	✓	✓	✓	✓	✓
RRC	✓	✓	✓	✓	✓	✗	✗
Rand Augment	✗	Adapt.	9/0.5	7/0.5	✗	✗	✗
3 Augment (ours)	✗	✗	✗	✗	✓	✓	✓
LayerScale	✗	✗	✗	✗	✓	✓	✓
Mixup alpha	✗	Adapt.	0.8	0.2	0.8	✗	✗
Cutmix alpha	✗	✗	1.0	1.0	1.0	1.0	1.0
Erasing prob.	✗	✗	0.25	✗	✗	✗	✗
ColorJitter	✗	✗	✗	✗	0.3	0.3	0.3
Test crop ratio	0.875	0.875	0.875	0.95	1.0	1.0	1.0
Loss	CE	CE	CE	BCE	BCE	CE	CE

Summary of different training setups
[Touvron et al. 2022]

- Touvron et al. 2022 improve/simplify DeiT recipe based on advances in ResNet training (85.8% ImageNet)
- Beyer et al. 2022, with the right *small modifications*, ViT-Small achieves 80%, comparable to DeiT at the same scale

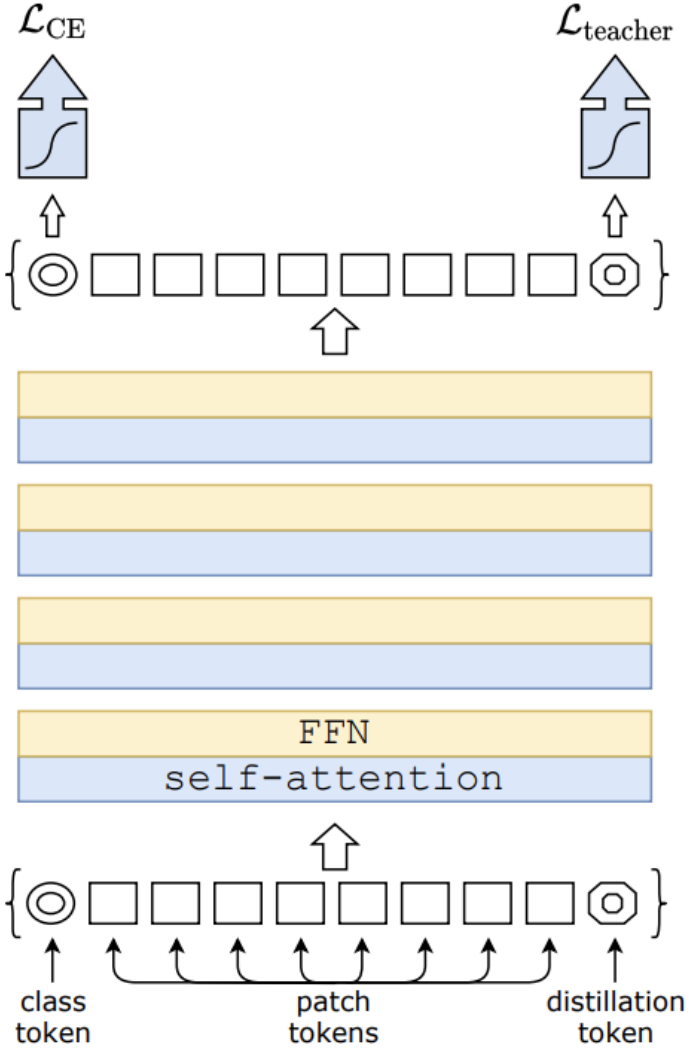


[Beyer et al. 2022]

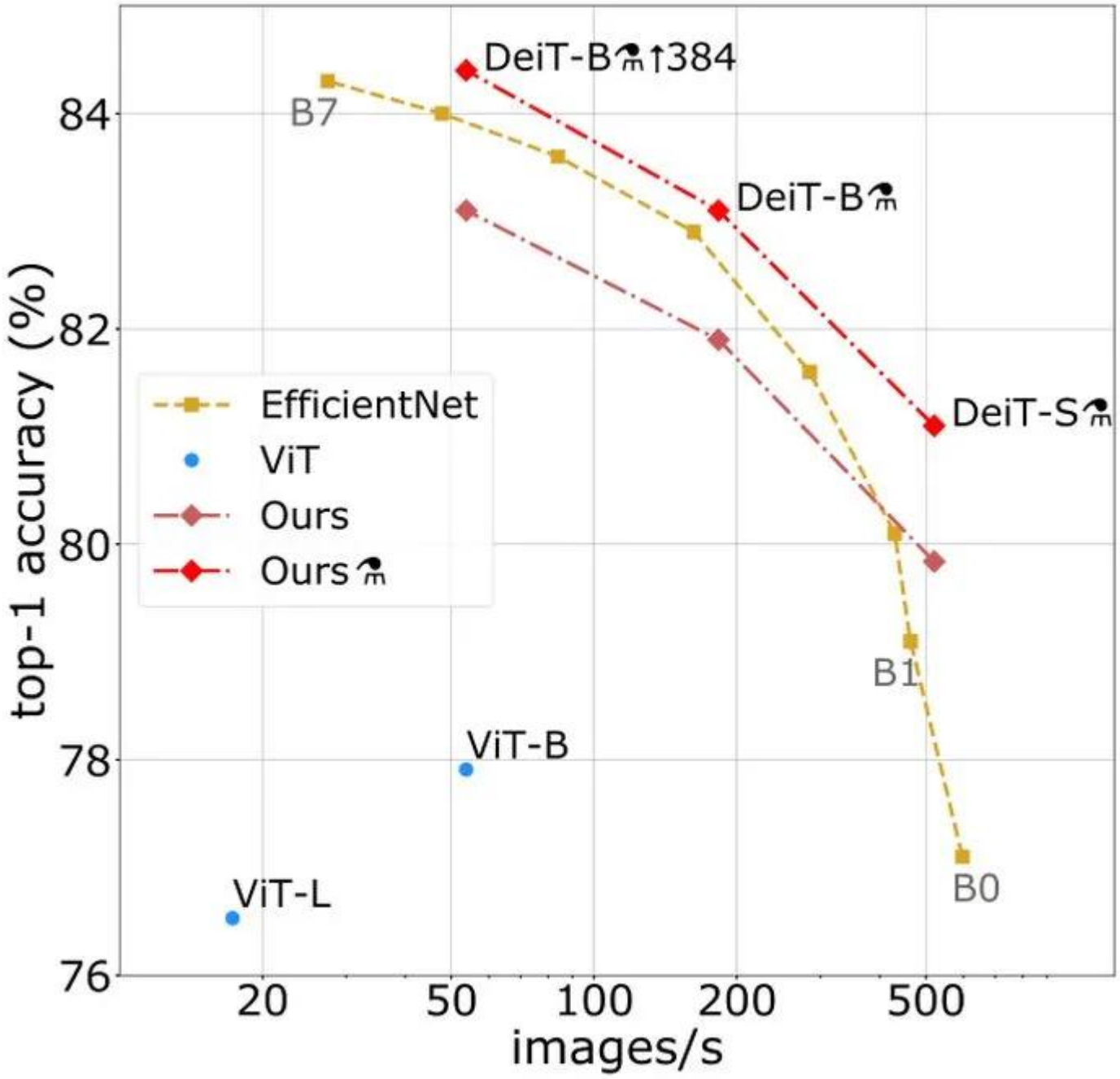
Touvron et al., DeiT III: Revenge of the ViT, arXiv 2022
Beyer et al., Better plain ViT baselines for ImageNet-1k, arXiv 2022

Transformer+Distillation：DeiT

- Problems of ViT
  - transformers do not generalize well when trained on insufficient amounts of data
- DeiT: training strategy
  - Distillation (Distillation loss)
  - 通過引入了一個distillation token，然後在self-attention layers中跟class token，patch token不斷交互。
  - 它跟左下角的class token很像，唯一的區別在於，class token的目標是跟真實的label一致，而distillation token是要跟teacher model預測的label一致。

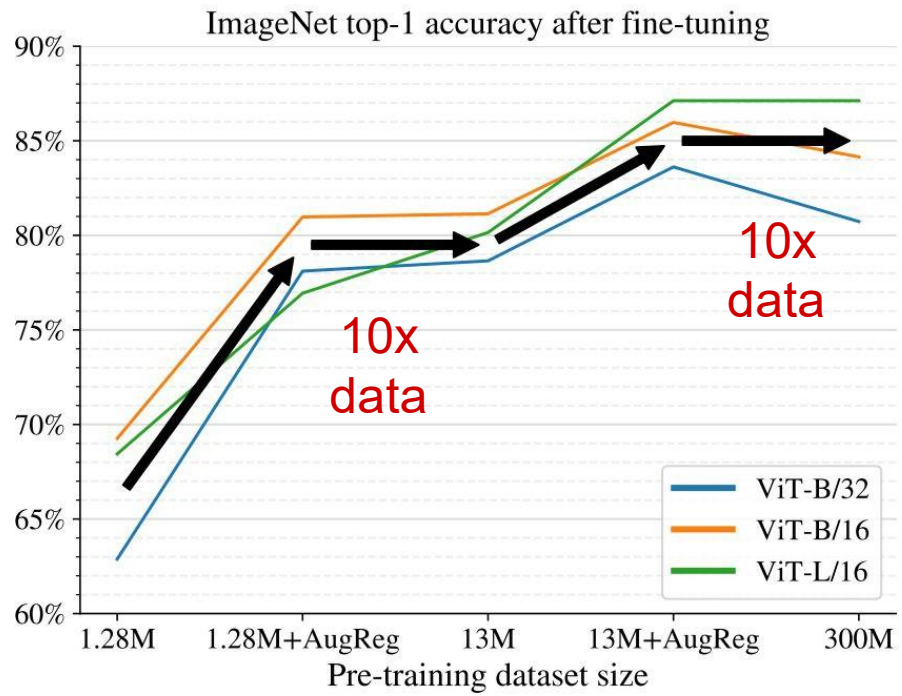


- DeiT
  - No MLP head but a linear classifier
  - ViT 在小資料集上的性能不如使用CNN網路EfficientNet，但是跟ViT結構相同，僅僅是使用更好的訓練策略的DeiT比ViT的性能已經有了很大的提升，
  - 在此基礎上，再加上蒸餾(distillation) 操作，性能超過了EfficientNet。



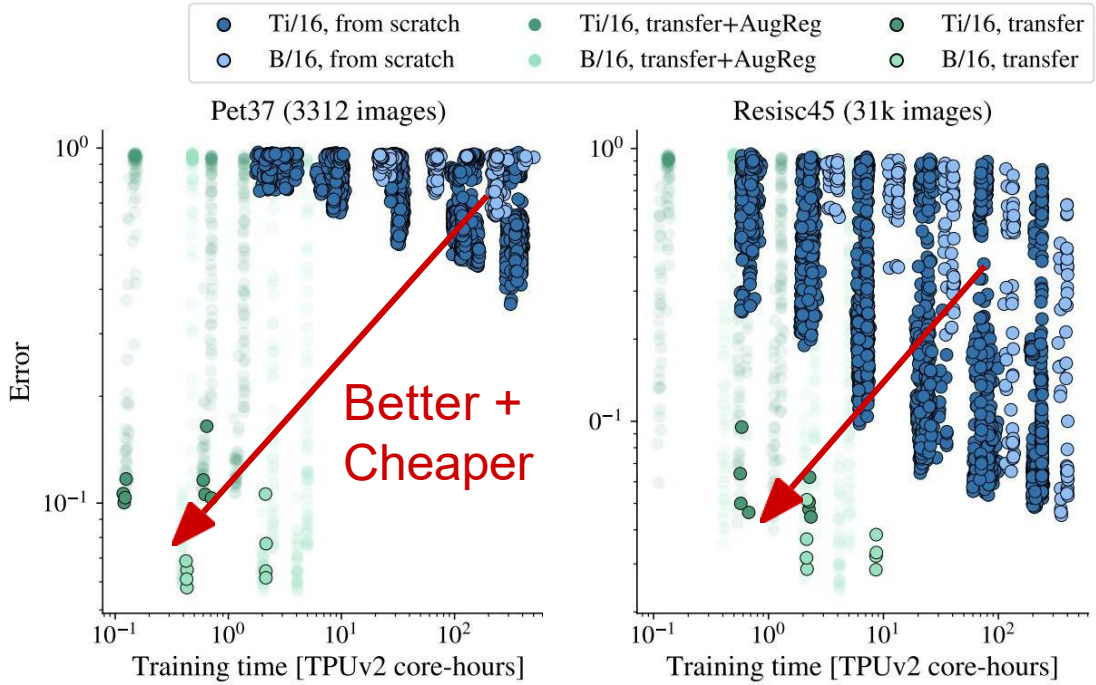
Regularization/Augmentation in other data regimes

Reg/aug & pre-training



Additional reg/aug makes up for (approx.) 10x pre-training data

Low data regime



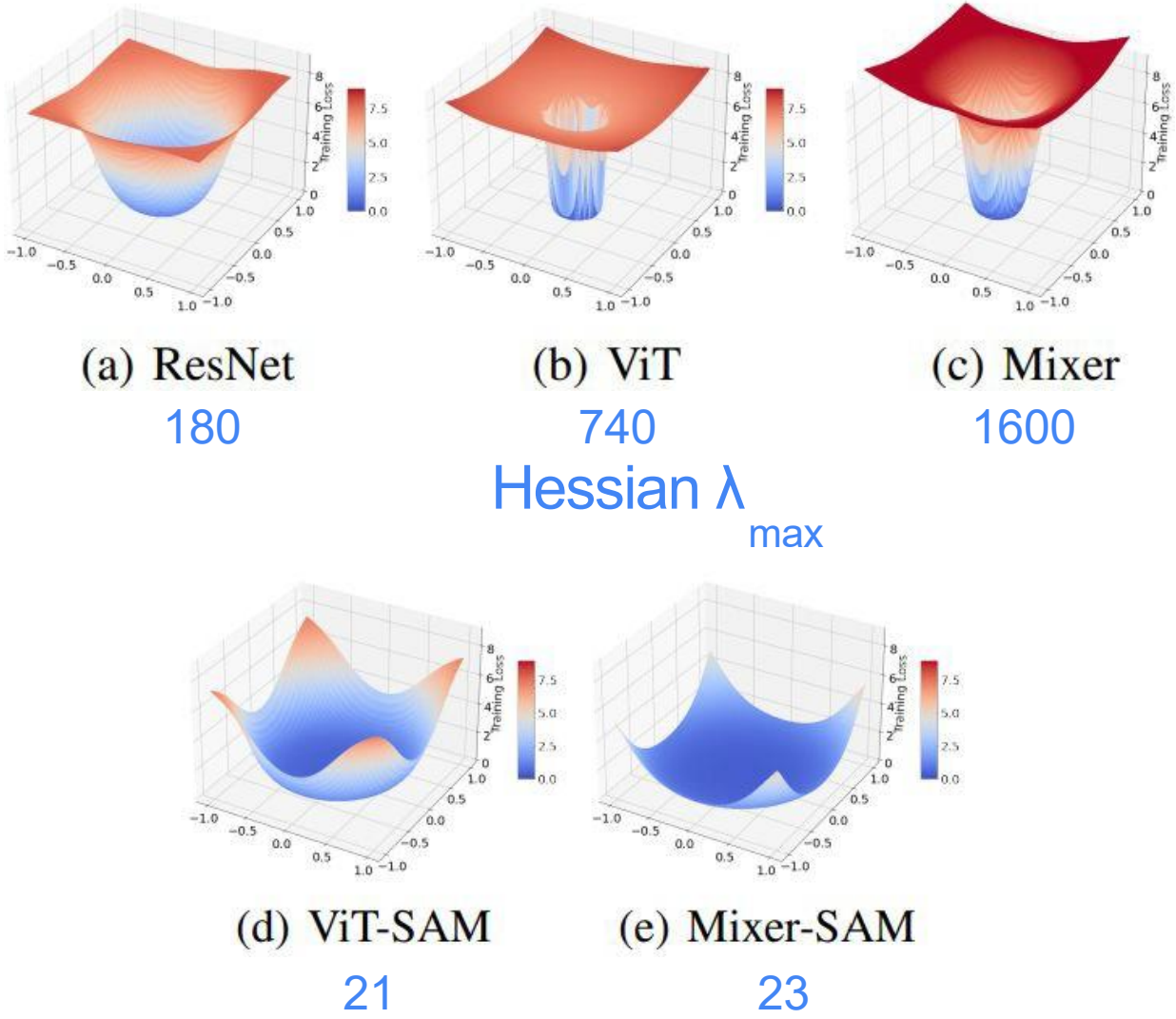
Even with tricks, for ***small*** datasets, transfer trains faster and finds good hyperparameters more easily

Steiner et al., How to train your ViT? Data, Augmentation and Regularization in Vision Transformers, TMLR 2022

Optimization

Chen et al. 2022

- Observation: When trained on ImageNet alone, ViT/Mixer converge to much sharper mimima
- Train with SAM (Sharpness Aware Minimum) optimizer
- ImageNet from scratch + *vanilla training setup* (crop, weight decay, dropout)
  - 80% ViT (from 75%)
  - 77% MLP-Mixer (from 66%)



[Chen et al. 2022]

Chen et al., When Vision Transformers outperform ResNets without pre-training or strong data augmentations, ICLR 2022

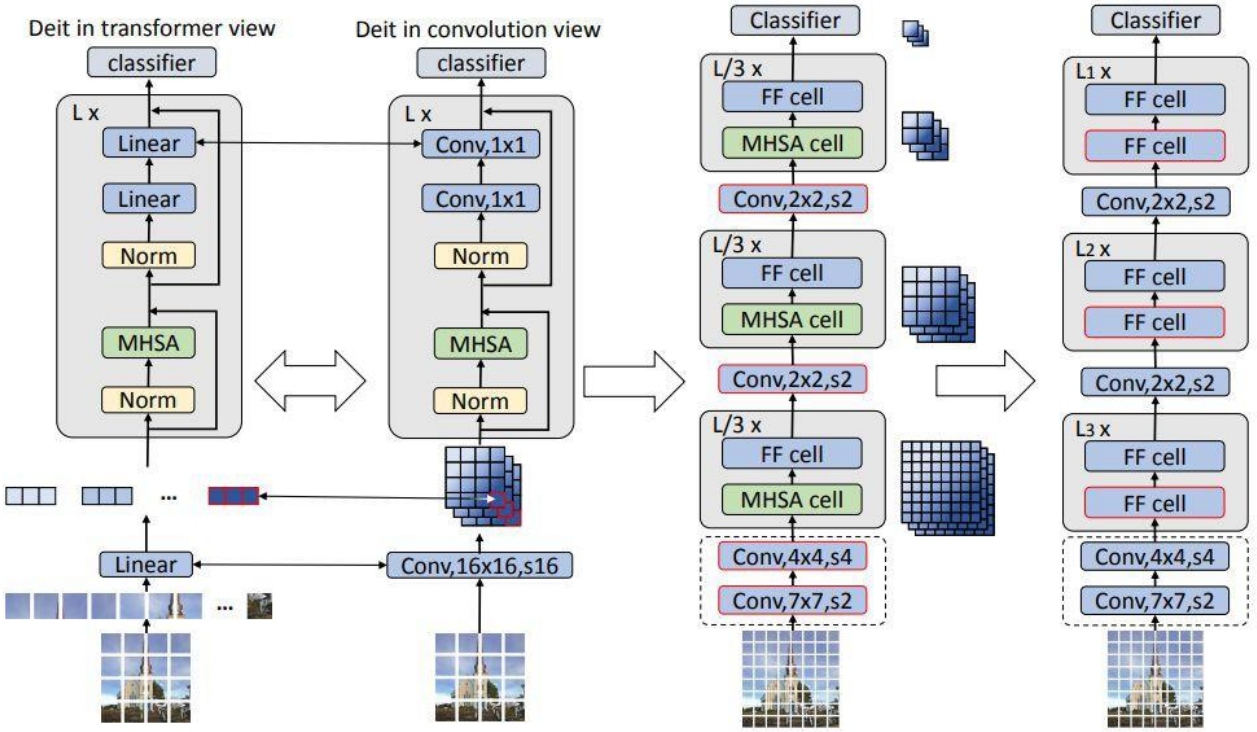
Summary so far

- Vision Transformers/MLPs can learn great features for vision
- Pre-training can be replaced/complimented with augmentation/regularization
  - Chen et al 2017: 79% ImageNet w/ 300M image pre-training using ResNet-101
  - Beyer et al. 2022: 80% ImageNet from scratch using ViT-S/16
- Caveat: some methods specialized to ImageNet training from scratch

ViT-CNN Spectrum

Can one improve performance (accuracy, training time, stability, data efficiency) by combining the best of both worlds?

ViT-CNN Spectrum

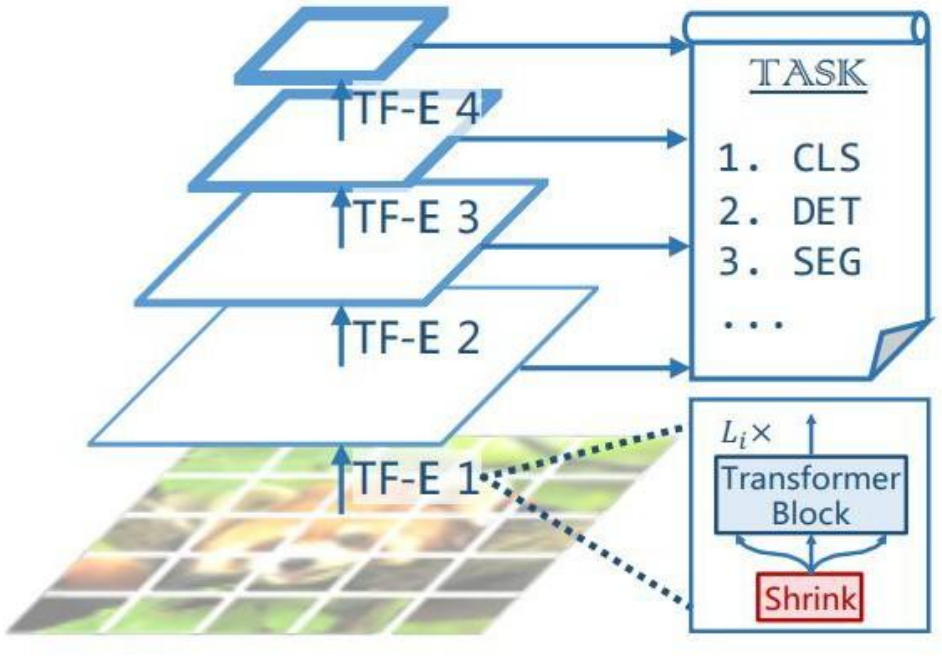


ViT-CNN spectrum from VisFormer [Chen et al. 2021]
A similar decomposition in Liu et al. [2022]

1. Isotropic / Pyramidal (staged)
2. Stem (root)
3. Head
4. Self-attention / NxN convolution
5. Many details:
   normalization layer,
   pointwise nonlinearity,
   position embedding,
   location of skips,
   width ratios

Chen et al., Visformer: The Vision-friendly Transformer, ICCV 2021
Liu et al., A ConvNet for the 2020s, arXiv 2022

ViT-CNN Spectrum: Pyramidal ViTs



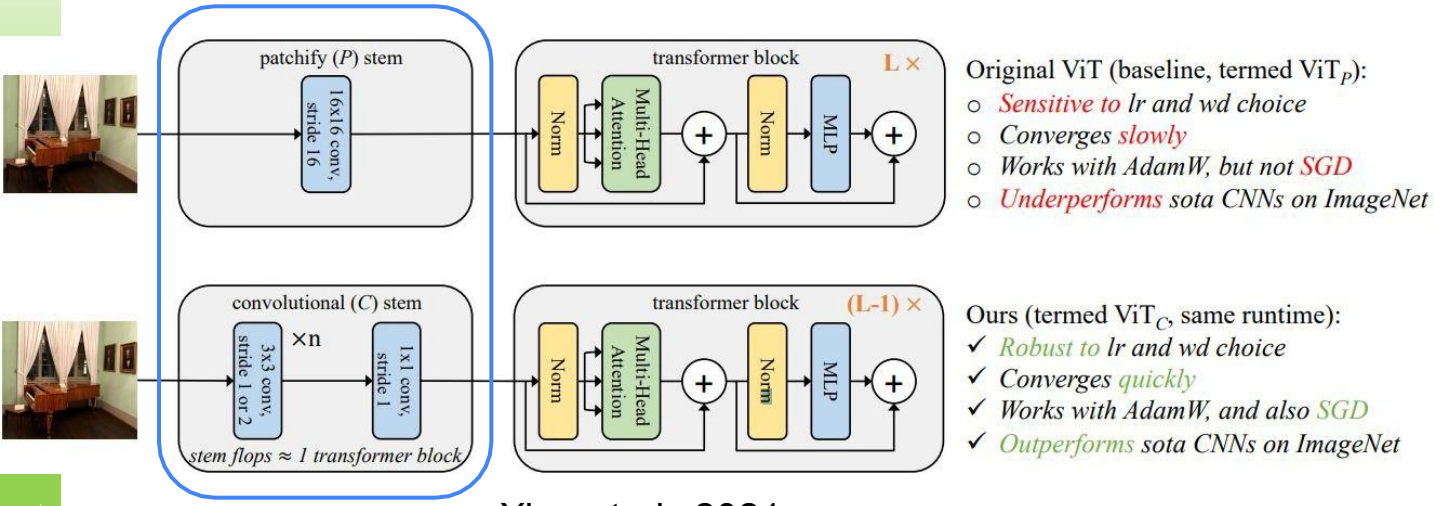
[Wang et al. 2021] Pyramid Vision Transformer

- Transformers have uniform width (isotropic), CNNs *mostly* built from "stages"
- T2T-ViT, PiT, PvT, PvT v2, Swin Transformer, CvT, Multiscale ViT
- Pro: Improvement in accuracy (/fewer parameters) at small/medium scale. *Potentially most useful for high-res tasks.*
- Con: Increased complexity and hyperparameters

Yuan et al, Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet, ICCV 2021
Heo et al., Rethinking Spatial Dimensions of Vision Transformers, ICCV 2021
Wang et al., Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions, ICCV 2021
Ze et al., Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021
Wu et al., CvT: Introducing convolutions to vision transformers, ICCV 2021
Fan et al., Multiscale Vision Transformers, ICCV 2021
Wang et al. PVT v2: Improved baselines with Pyramid Vision Transformer, arXiv 2021

ViT-CNN Spectrum: Root/Stem

"patching" →
a few convs



Xiao et al., 2021

- 16x16 patching + projection equivalent to conv w/ kernel=16x16, stride=16x16
  - → Use another convolutional structure (e.g. CvT, VisFormer, LeViT)
- "Early convolutions help transformers see better" shows various benefits e.g. *can train with SGD* [Xiao et al. 2021]
- Pro: Simple change, worth it for classification (unless prevents reuse of pre-trained weights)
- Con: Problematic for mask-based self-supervision

Xiao et al., Early Convolutions Help Transformers See Better, NeurIPS 2021
Chen et al., Visformer: The Vision-friendly Transformer, ICCV 2021
Wu et al., CvT: Introducing convolutions to vision transformers, ICCV 2021
Graham et al., LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference, ICCV 2021

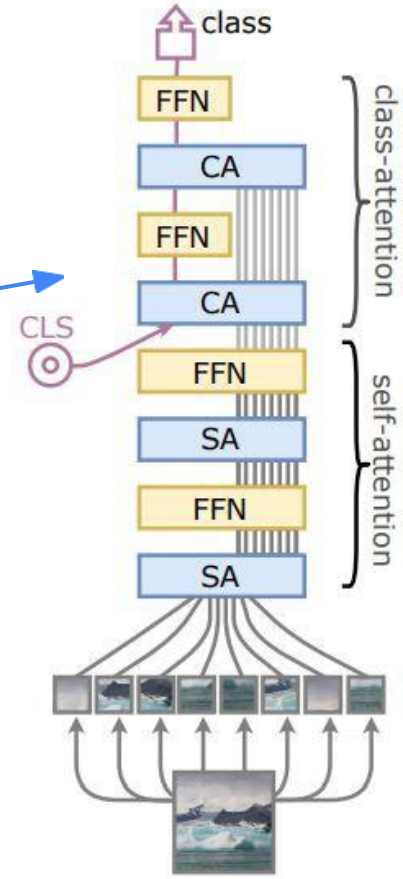
ViT-CNN Spectrum: Head

Various options:

- "Class Token" [Devlin et al. 2018]
- Global Average Pooling (popular for CNNs)
- Multihead Attention Pooling [Zhai et al. 2021] / Class Attention [Touvron et al. 2021]

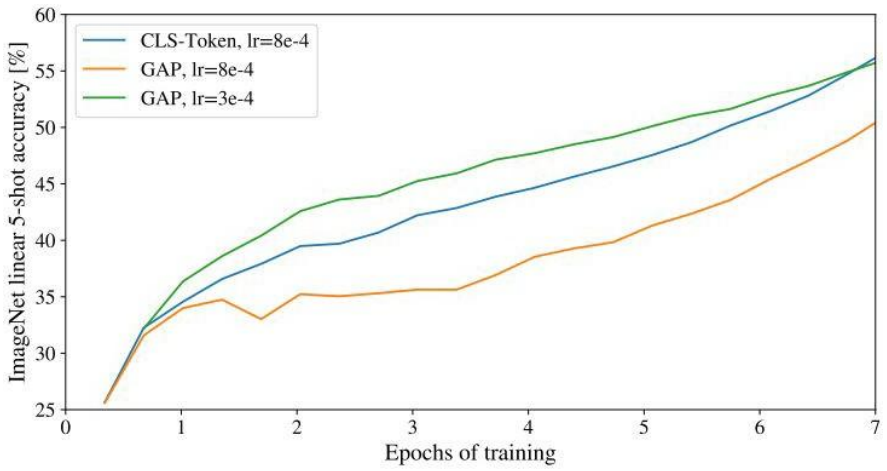
Summary

- Classification accuracy: choice not very important
- "Class Token": Wastes memory on some hardware
- For localization or self-supervised learning *may* make more difference



Touvron et al., 2021

After re-tuning LR, little difference in accuracy



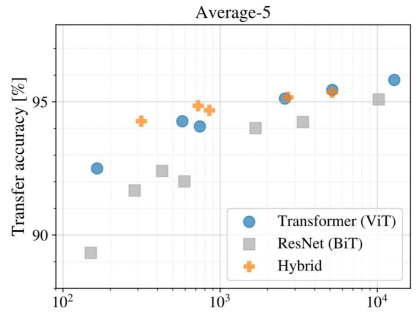
Dosovitskiy et al., 2020

Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ACL 2019
Dosovitskiy et al., An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR 2021
Zhai et al., Scaling Vision Transformers, CVPR 2021
Touvron et al. Going deeper with Image Transformers, ICCV 2021

Replacing self-attention with convolutions (or other local operations)

Option 1: Stacked hybrid

Visformer [Chen et al. 2021]
BotNet [Srinivas et al. 2021]
CoAtNet [Dai et al. 2021]



Pro: Great at medium-scale, works at larger scales

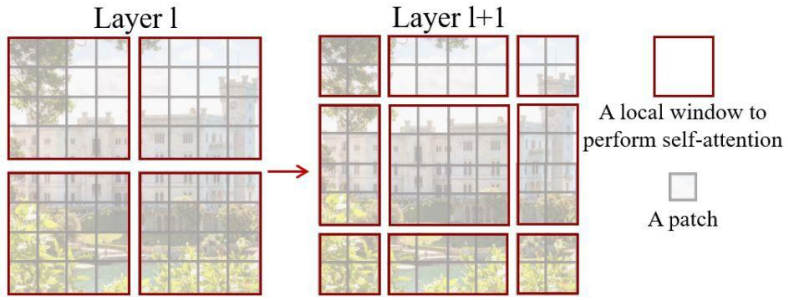
Con: Best balance of conv/attn blocks may be scale & task dependent

BoTNet-50	
7×7, 64, stride 2	
3×3 max pool, stride 2	
1×1, 64 3×3, 64 1×1, 256	×3
1×1, 128 3×3, 128 1×1, 512	×4
1×1, 256 3×3, 256 1×1, 1024	×6
1×1, 512 MHSA, 512 1×1, 2048	×3

Srinivas et al., 2020

Option 2: Add locality prior into attention layer

CvT [Wu et al. 2021]
ConViT [d'Ascoli et al. 2021]
CoAtNet [Dai et al. 2021]
Swin Transformer [Ze et al. 2021]



Ze et al. 2021

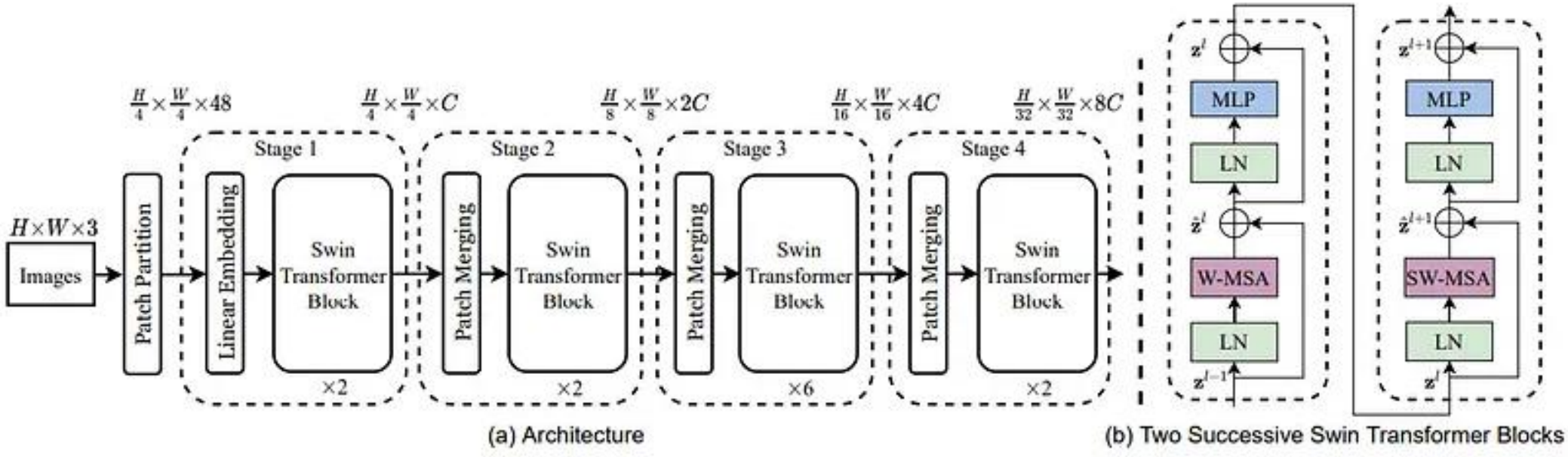
Pro: In principle "best of both worlds"

Con: May be inefficient on some hardware

Wu et al., CvT: Introducing Convolutions to Vision Transformers, ICCV 2021
d'Ascoli et al., Convit: Improving vision transformers with soft convolutional inductive biases, ICML 2021
Srinivas et al., Bottleneck Transformers for Visual Recognition, CVPR 2021
Dai et al., CoAtNet: Marrying Convolution and Attention for All Data Sizes, NeurIPS 2021
Chen et al., Visformer: The Vision-friendly Transformer, ICCV 2021
Ze et al., Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021

Swin Transformer

- Transformer從語言轉換到視覺領域有兩個困難
  - 影像尺度變化大 (大物體/小物體)，不像NLP是標準固定尺寸
  - 影像像素高解析度，計算複雜度是影像size的平方，造成計算量相當大
- Hierarchical Vision Transformer using Shifted Windows

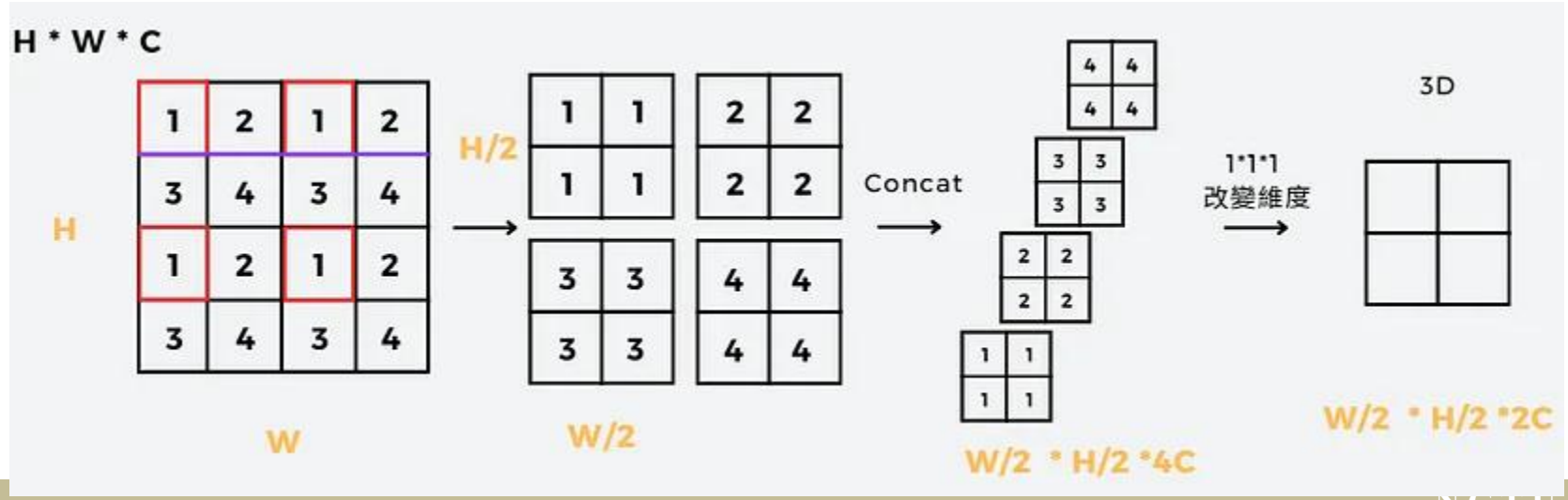


(a) Architecture

(b) Two Successive Swin Transformer Blocks

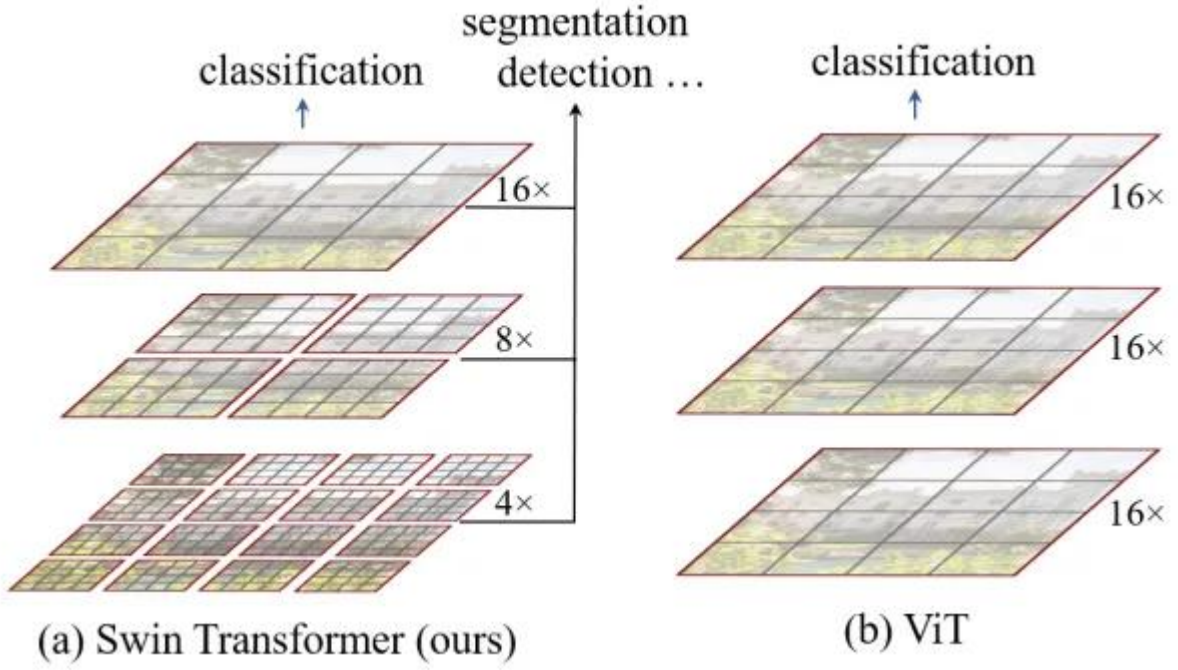
- Shifted Window:
  - 提出了移動窗口(Shifted Window)的自注意力計算方法，在保證高效率計算的同時，也提供了窗口之間的連接，各窗口間的特徵可以互相連接交流，使模型有更豐富的特徵表示。
- Hierarchical Transformer：
  - 使用類似CNN層次化的方式，分層的特徵來適應視覺任務中物件尺度的變化，並只在局部窗口內計算自注意力來實現線性計算複雜度，解決了之前Transformer在視覺任務上的兩個主要問題。
- 在物件偵測、實例分割、影像分類都獲得很好的成績。

- Patch Partition:
  - 可將影像分割為多個tokens，本論文使用4*4 size的patch，產生H/4 * W/4個patches (56*56 patches)。Paches的維度為48，即4* 4 (patch size)*3 RGB channel。
- Linear Embedding:
  - 可以將維度線性轉換為C，swin-T的C為96。
- Patch Merging:
  - 要多尺寸的特徵訊息，需構建一個層級式的Transformer，如CNN中的池化操作。

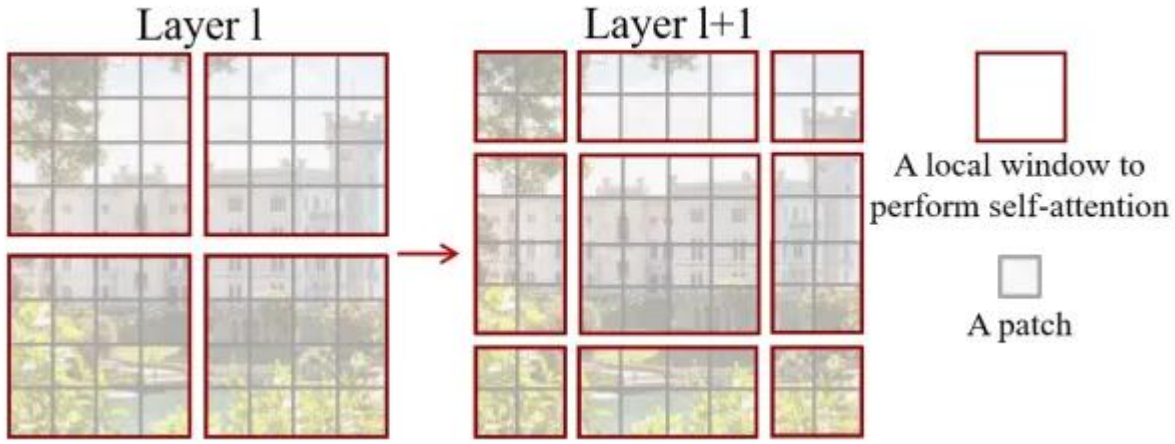


- Vision Transformer其中一個缺點就是全局計算會造成平方倍的時間複雜度，而Swin Transformer則改善了這個情況。
- W-MSA (Window Multi-Head Self Attention):
  - W-MSA將圖像均勻劃分為多個非重疊的窗口，每個窗口包含M×M個patch。在每個窗口內進行自注意力計算，窗口之間互不相關。
  - 如下圖左圖紅框為 Swin-Transformer 單獨對每個窗口(16 個 值)內部進行W-MSA；而Vision Transformer則是產生單一低解析度的特徵圖，並且全局建模計算，計算複雜度較高。

紅框:計算attention
Patch merging: 看更大範圍

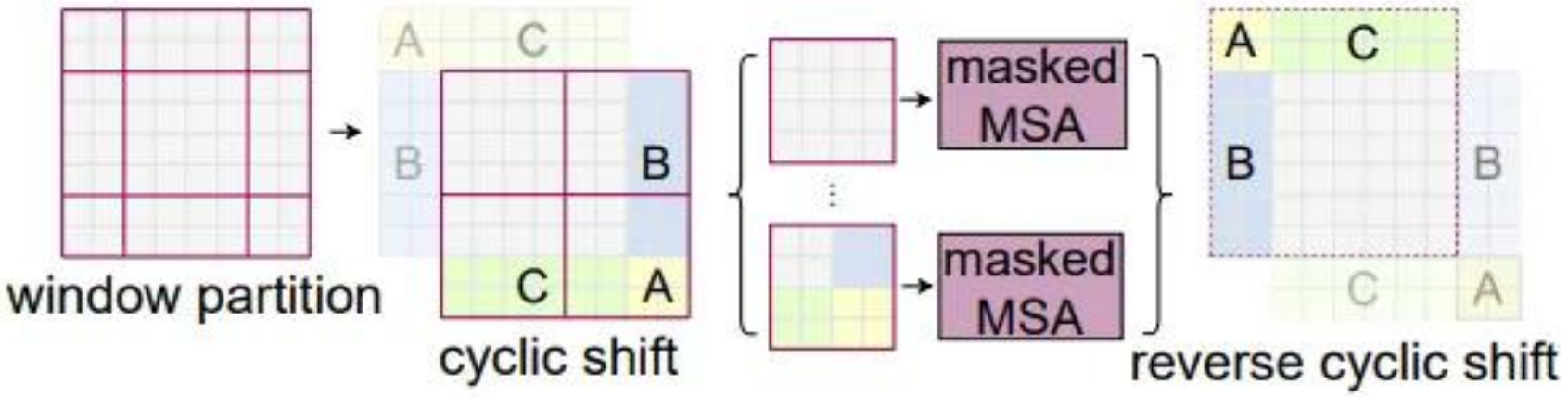


- SW-MSA (Shifted Windows Multi-Head Self-Attention)
  - 由於窗口之間不關聯，W-MSA的建模能力受到限制。因此文中提出了SW-MSA。原本的窗口為Layer 1，將窗口往右下移，變成如Layer 1+ 1所示，窗口移動後，每個patch 可以跟其他窗口patch連結。



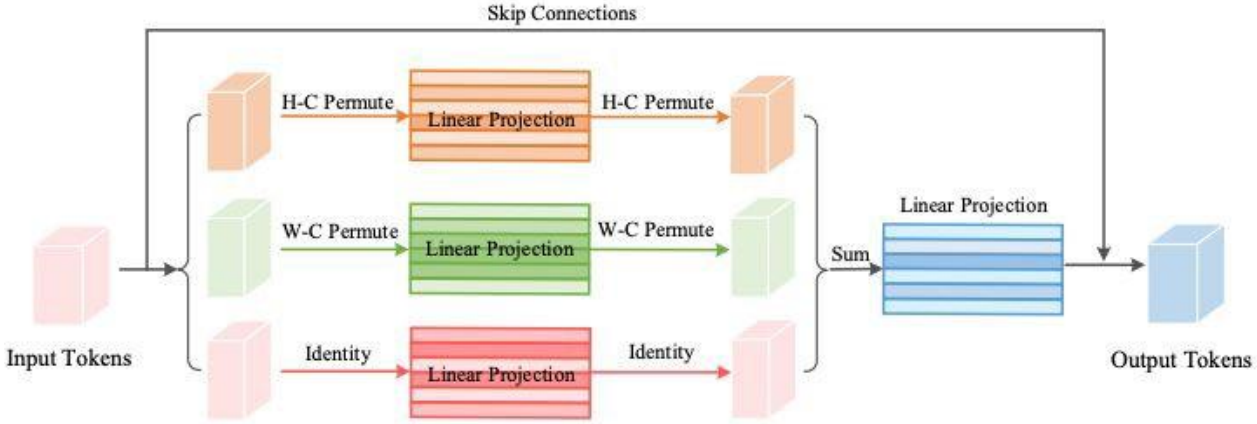
一個完整的Swin Transformer Block會經過W-MSA與SW-MSA。

- Efficient batch computation for shifted configuration
  - 由上述可以發現，原本只要計算四個窗口即可結束，現在卻變成計算九個窗口了，計算複雜度提升，因此為了達成只計算四窗口又可以窗口間互相聯繫，本文提出cyclic shift。如下圖，將A、B、C三塊移到右方，經過循環移位後，可以看到變回四窗口了，然後又有個新問題，這些移動的patch不應該是有聯繫的，不應做自注意力計算，因此本文又提出了masked MSA，讓各窗口可以合理的計算自注意力。Masked MSA 可以看作者於github的解釋。



Other exotic ways of processing global and accommodating spatial priors

- Vision Permutator [Hou et al. 2021]
  - Permutes height, width, and channel axes, and runs separate MLPs over all three in parallel
- CycleMLP [Chen et al. 2021]
  - Generalized channel/token mixing by creating MLPs with receptive fields that span space and channels
- GFNet [Rao et al. 2021]
  - Applies spatial processing in the Fourier domain



Hou et al. 2021

Hou et al., Vision Permutator: A Permutable MLP-Like Architecture for Visual Recognition, TPAMI 2022
Chen et al., CycleMLP: A MLP-like Architecture for Dense Prediction, ICLR 2022
Rao et al., Global Filter Networks for Image Classification, NeurIPS 2021

Summary so far

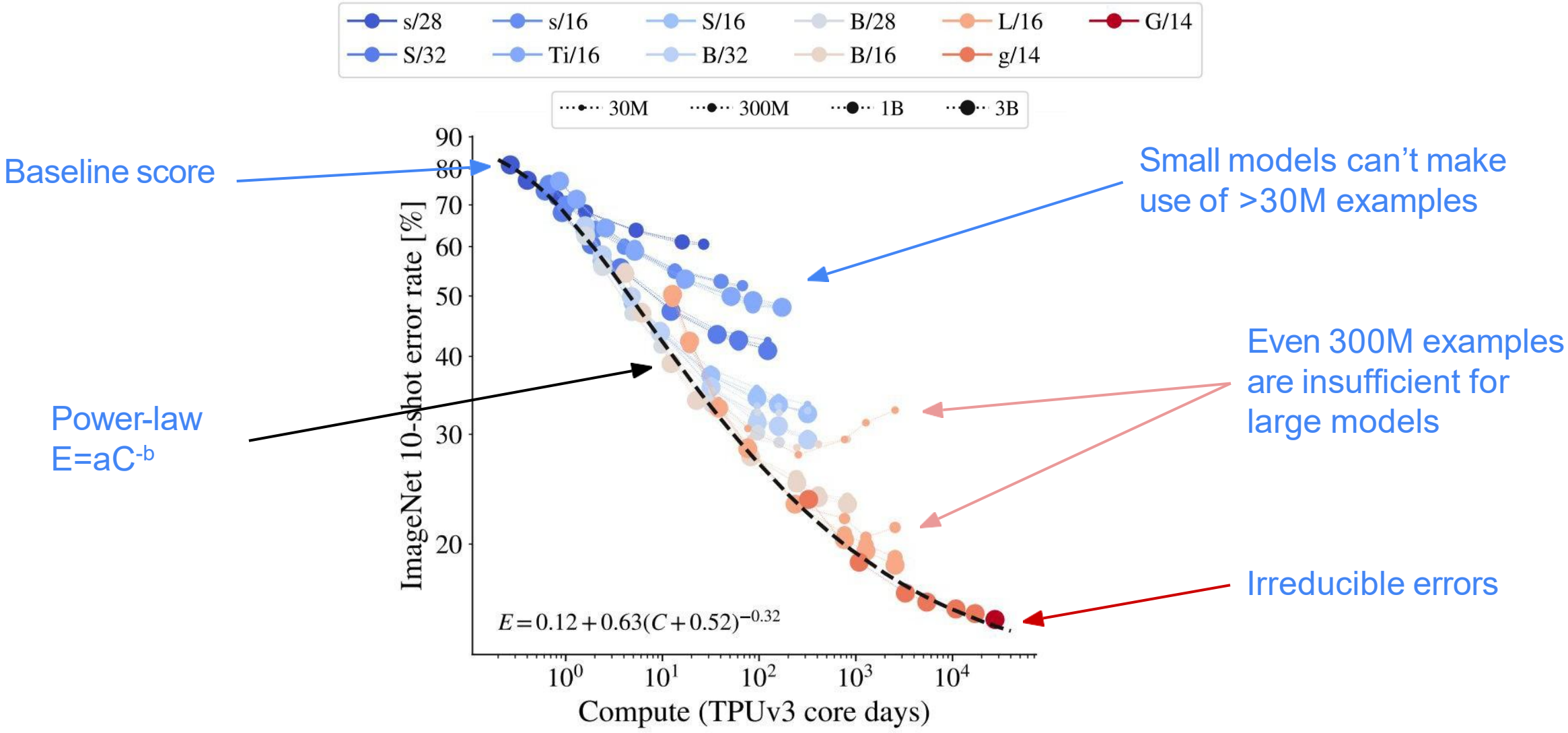
- Transformers/MLPs can learn great features for vision with ample data or the right training recipe
- ViTs/CNNs differ in a number of small ways
- ViT-CNN hybrid models can work well, particularly at small/medium scales

Scaling up

- Even with many advances in data-efficient training, transfer learning is still effective: ImageNet SOTA 91.0% vs. 88.3% without extra data (PeCo)
- Transfer learning benefits from scale
- Architectures introduced so far still relatively small

Dong et al., PeCo: Perceptual Codebook for BERT Pre-training of Vision Transformers, arXiv 2022

Scaling Laws for Vision



Baseline score

Power-law
$E=aC^{-b}$

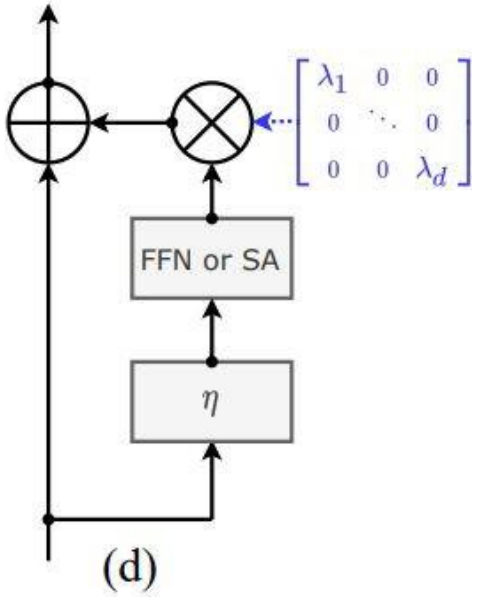
Small models can't make use of >30M examples

Even 300M examples are insufficient for large models

Irreducible errors

Zhai et al., 2021

Scaling Techniques: Deeper ImageNet models



CaiT [Touvorn et al. 2021]: Deeper models fail to converge on ImageNet from scratch. Address with LayerScale, CA Head, tuning of training recipe

- 48 layers, 350M params, 86.5% ImageNet from scratch (matching best CNNs)

Improvement	top-1 acc.	#params	FLOPs
DeiT-S [*d=384,300 epochs*]	79.9	22M	4.6B
+ More heads [*8*]	80.0	22M	4.6B
+ Talking-heads	80.5	22M	4.6B
+ Depth [*36 blocks*]	69.9†	64M	13.8B
+ **Layer-scale** [*init* $\varepsilon = 10^{-6}$]	80.5	64M	13.8B
+ Stch depth. adaptation [d_r=*0.2*]	83.0	64M	13.8B
+ **CaiT architecture** [*specialized class-attention layers*]	83.2	68M	13.9B
+ Longer training [*400 epochs*]	83.4	68M	13.9B
+ Inference at higher resolution [*256*]	83.8	68M	18.6B
+ Fine-tuning at higher resolution [*384*]	84.8	68M	48.0B
+ Hard distillation [*teacher: RegNetY-16GF*]	85.2	68M	48.0B
+ Adjust crop ratio [*0.875 → 1.0*]	85.4	68M	48.0B

Touvron et al., 2021

Touvron et al., Going deeper with Image Transformers, ICCV 2021

Summary so far

- Transformers/MLP can learn great features for vision with ample data or the right training settings
- ViT-CNN hybrid models can work well, particularly at small/medium scales
- Deep ViTs can be trained from scratch on ImageNet, but with extra effort to stabilize training
- At large scale, generic looking models/training setups can work well
- Sparsity in the medium/large scale regimes is less explored

Concluding thoughts (i)

1. Transformer-based vision models have lead to improved performance due to (1) capacity to learn efficiently, (2) driving innovation & tuning of training recipes.
2. Regularization & hybrid architectures are relatively well studied, interaction between optimization/architecture perhaps less so.
3. Work tends to focus on either ImageNet training or transfer from large datasets. Systematic reporting across many orders of magnitude (of data and compute) might help the field better understand interactions between scale/methods.

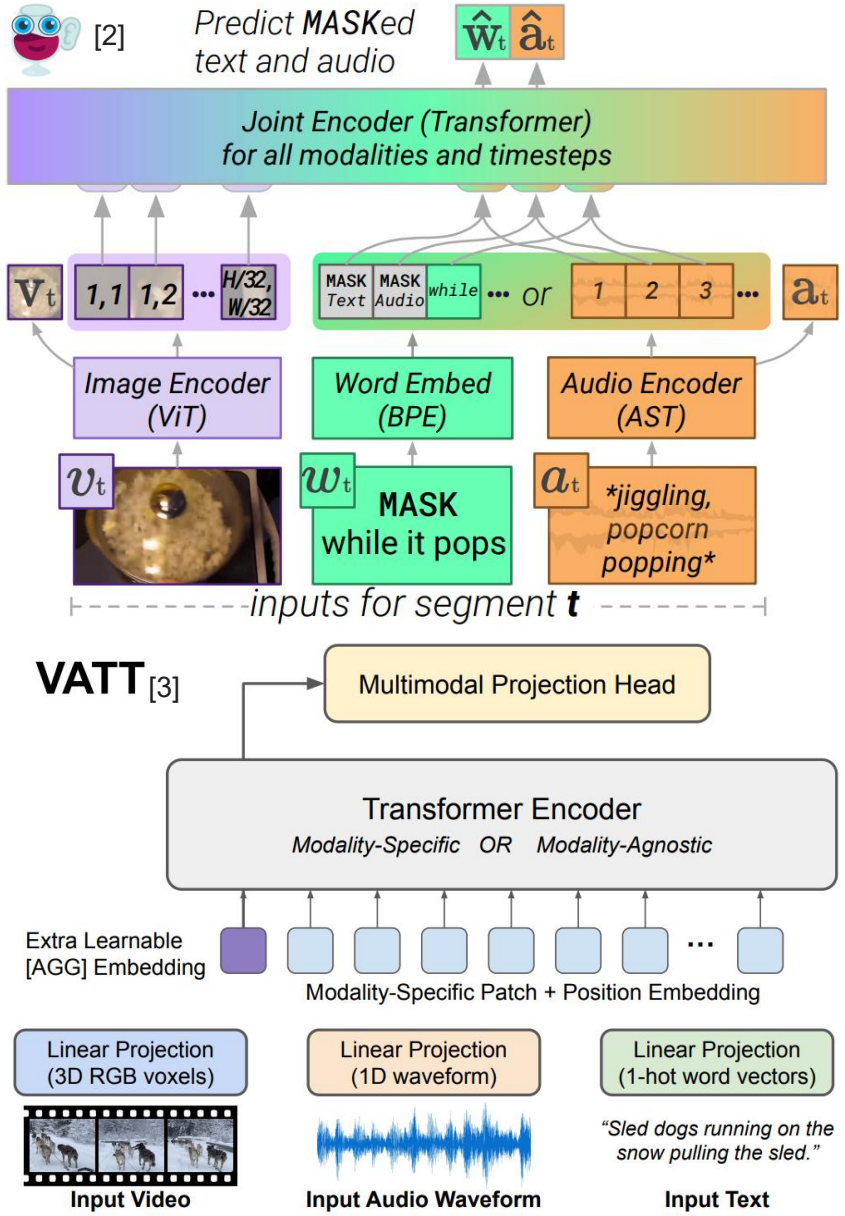
Anything you can tokenize, you can feed to Transformer

ca 2021 and onwards

Tokenize different modalities each in their own way (some kind of "patching"), and send them all jointly into a Transformer...

Seems to just work...
Currently an explosion of works doing this!

[1]



Images from:
[1] LIMoE by B Mustafa, C Riquelme, J Puigcerver, R Jenatton, N Houlsby
[2] MERLOT Reserve by R Zellers, J Lu, X Lu, Y Yu, Y Zhao, M Salehi, A Kusupati, J Hessel, A Farhadi, Y Choi
[3] VATT by H Akbari, L Yuan, R Qian, W-H Chuang, S-F Chang, Y Cui, B Gong

A note on Efficient Transformers

The self-attention operation complexity is $O(N^2)$ for sequence length N.

We'd like to use large N:

- Whole articles or books
- Full video movies
- High resolution images

Many O(N) approximations to the full self-attention have been proposed in the past two years.

Unfortunately, none provides a clear improvement. They always trade-off between speed and quality.

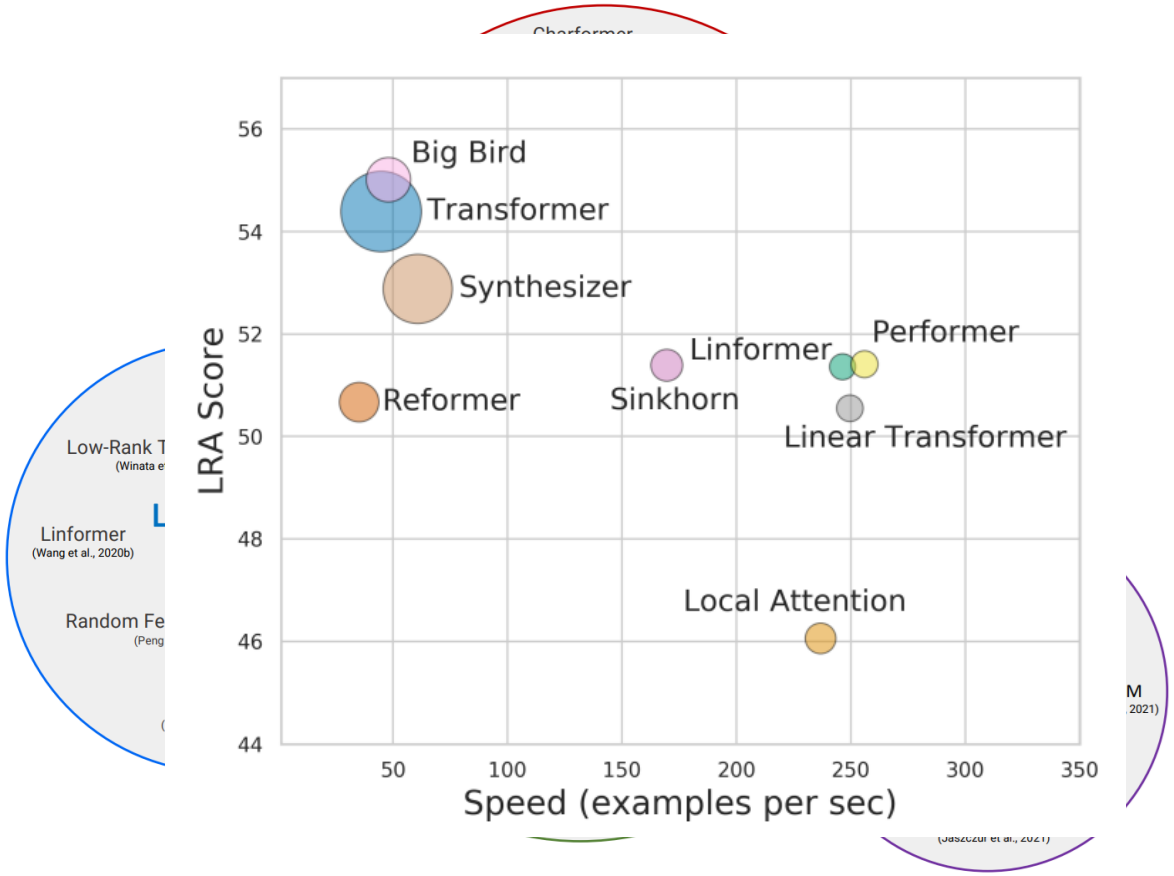


Figure 2: Taxonomy of Efficient Transformer Architectures.

Based on "Efficient Transformers: A Survey" by Yi Tay, Mostafa Dehghani, Dara Bahri, Donald Metzler and "Long Range Arena: A Benchmark for Efficient Transformers" by Y Tay, M Dehghani, S Abnar, Y Shen, D Bahri, P Pham, J Rao, L Yang, S Ruder, D Metzler

IMPROVEMENT OF VIT (OPTIONAL)

https://github.com/lucidrains/vit-pytorch

Survey of Transformers

- https://github.com/DirtyHarryLYL/Transformer-in-Vision

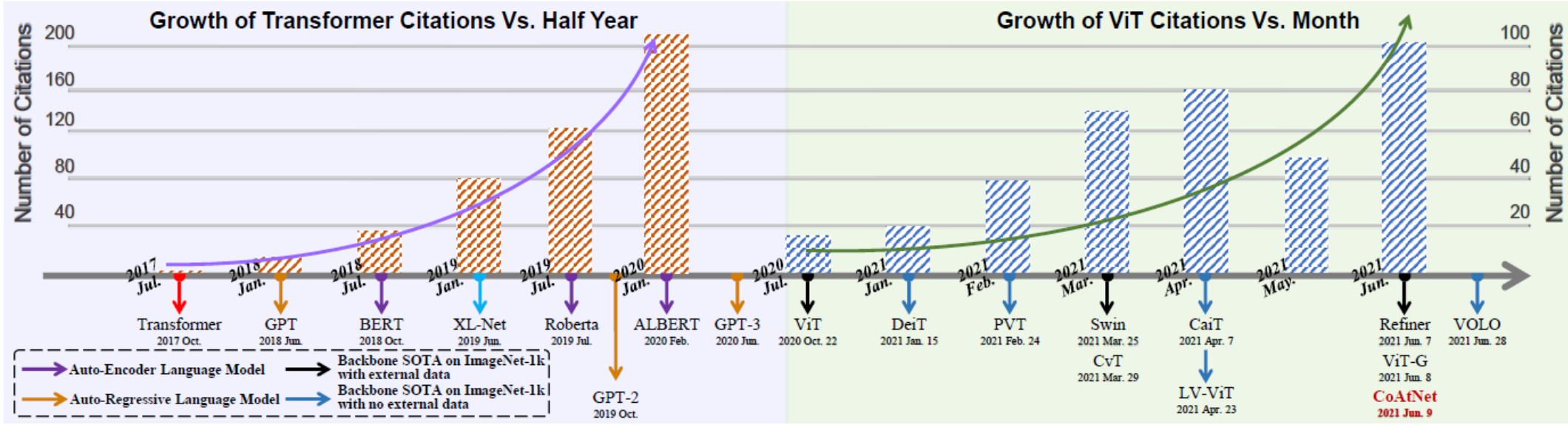


Fig. 1. Odyssey of Transformer application & Growth of both Transformer [1] and ViT [27] citations according to Google Scholar. (Upper Left) Growth of Transformer citations in multiple conference publication including: NIPS, ACL, ICML, IJCAI, ICLR, and ICASSP. (Upper Right) Growth of ViT citations in Arxiv publications. (Bottom Left) Odyssey of language model [1]–[8]. (Bottom Right) Odyssey of visual Transformer backbone where the black [27], [33]–[37] is the SOTA with external data and the blue [38]–[42] refers to the SOTA without external data (best viewed in color).

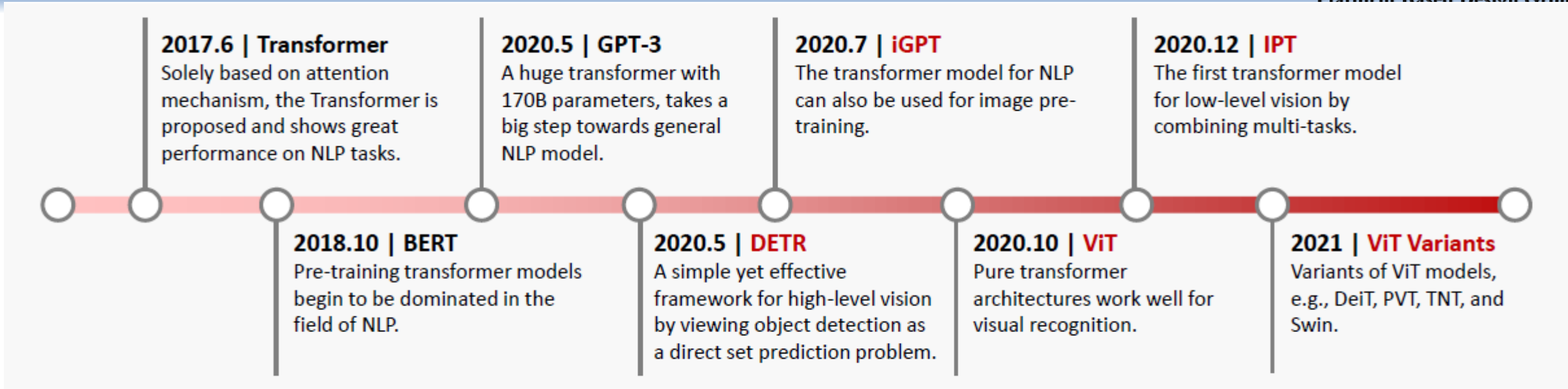


Fig. 1: Key milestones in the development of transformer. The vision transformer models are marked in red.

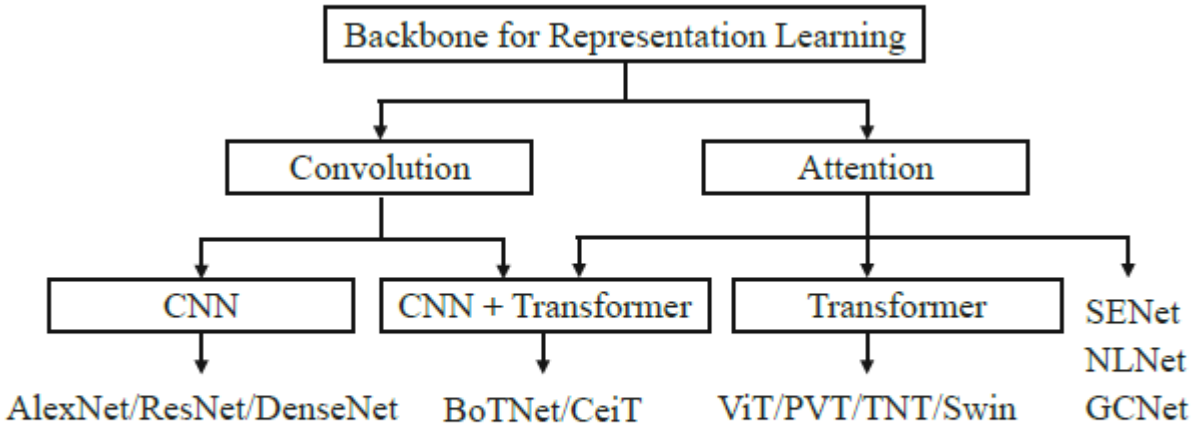


Fig. 5: A taxonomy of backbone using convolution and attention for representation learning.

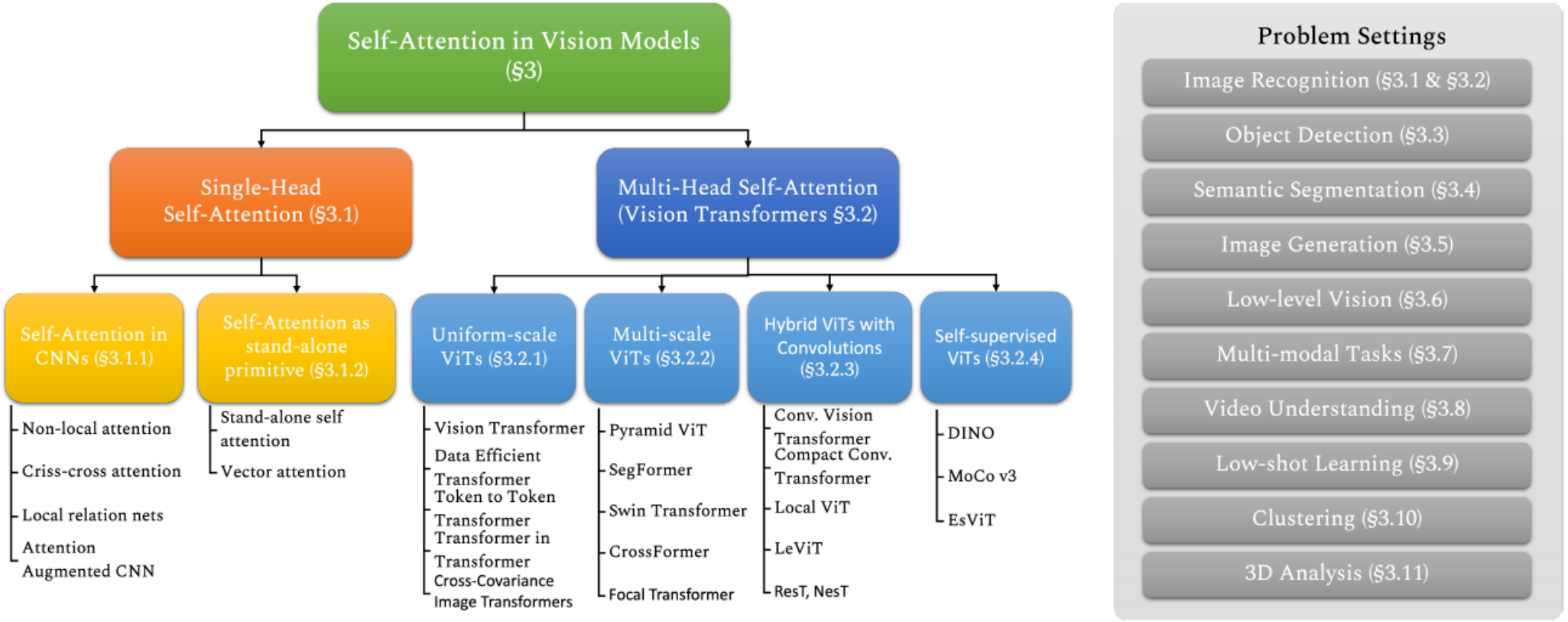


Fig. 4: *A taxonomy of self-attention design space*. Existing approaches based on self-attention explore single-head or multi-head (transformer) designs for vision tasks. We note that interesting efforts have been made to utilize knowledge from convolution based architectures to improve ViTs (e.g., multi-scale and hybrid designs). We categorize the upcoming sections of this survey according to the types of self-attention block (*left tree diagram*) as well as the prominent tasks in computer vision (*right*).

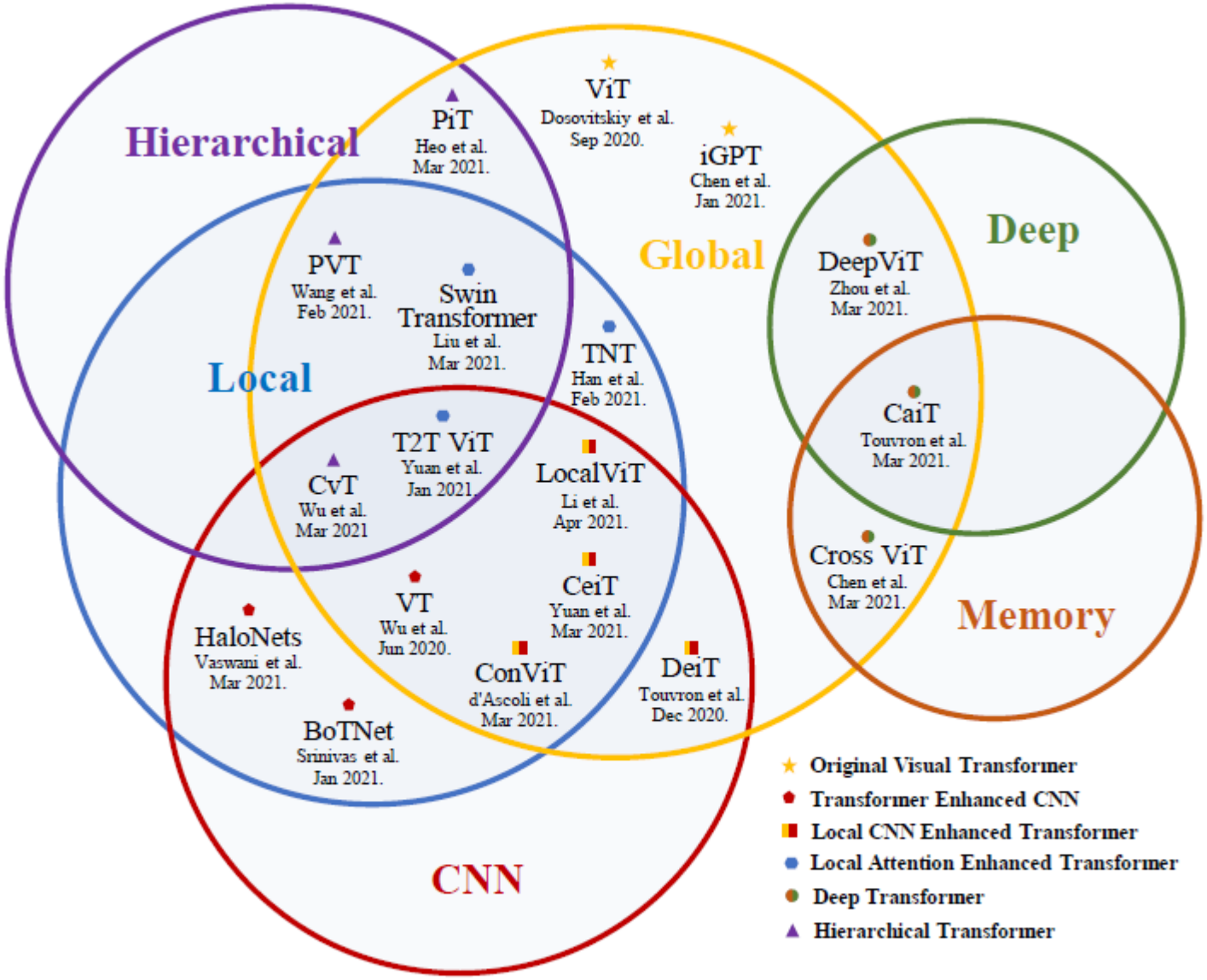
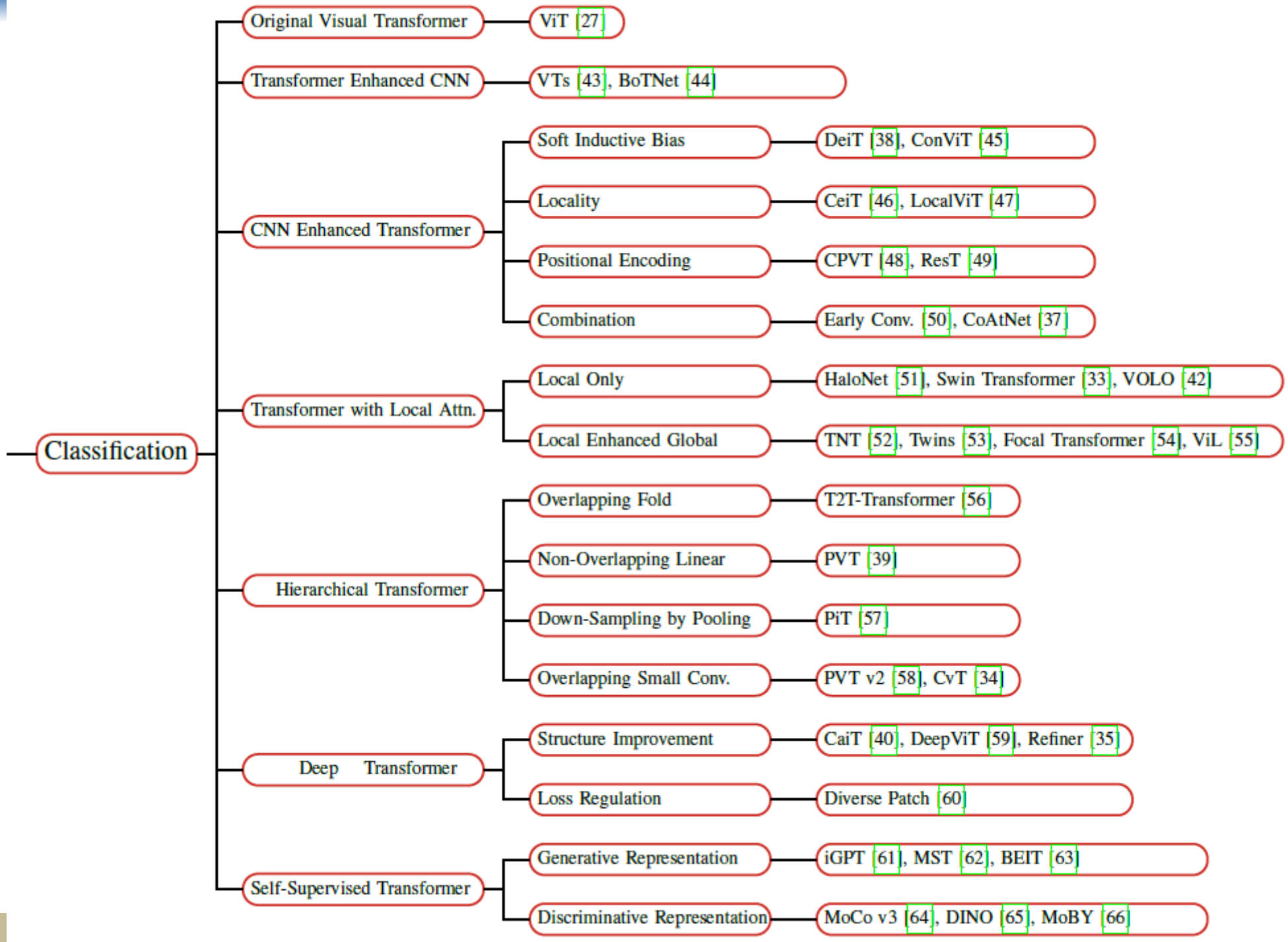
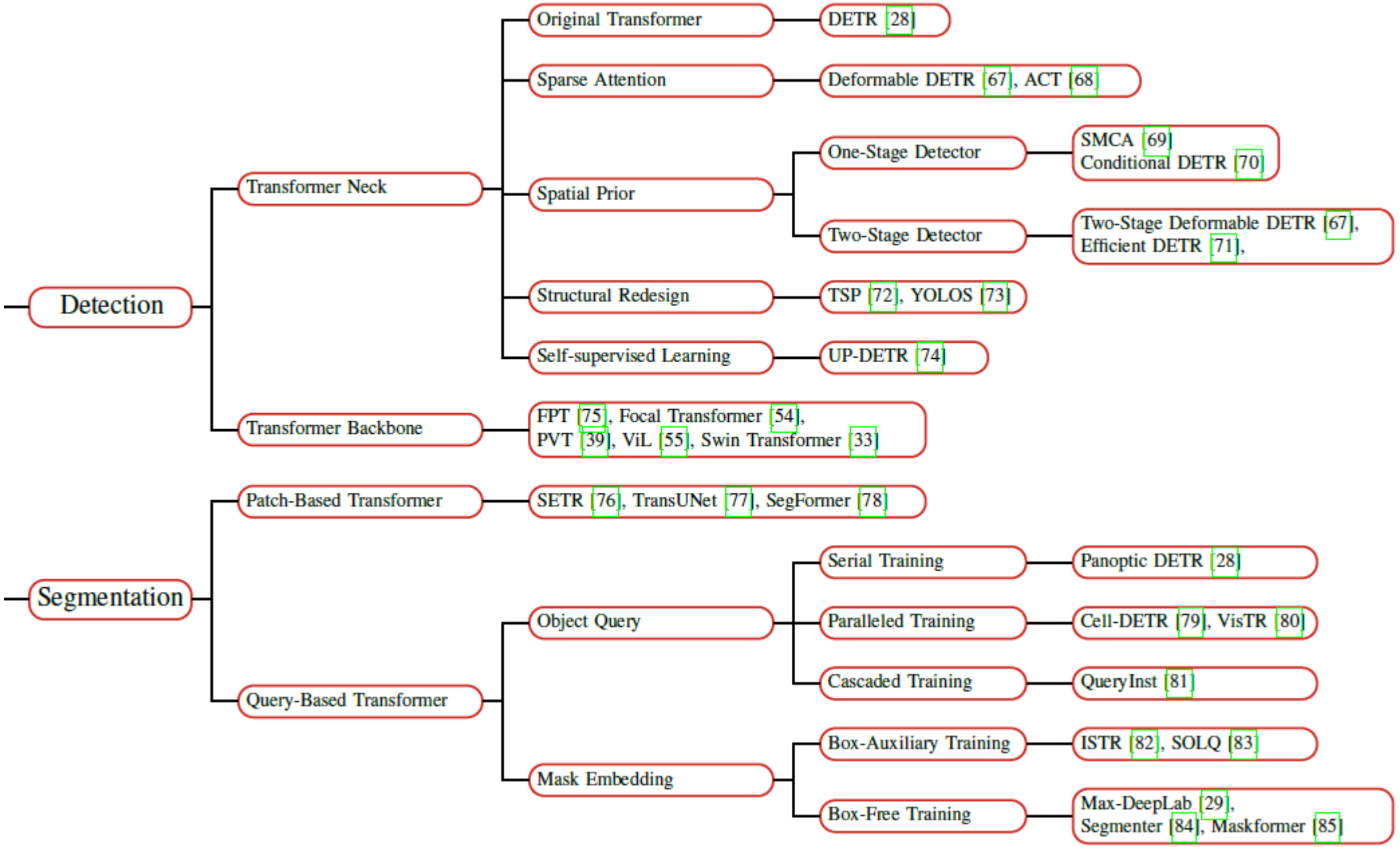


Fig. 5. Taxonomy of Visual Transformer Backbone (best viewed in color).


Classification
Original Visual Transformer
ViT [27]
Transformer Enhanced CNN
VTs [43], BoTNet [44]
CNN Enhanced Transformer
Soft Inductive Bias
DeiT [38], ConViT [45]
Locality
CeiT [46], LocalViT [47]
Positional Encoding
CPVT [48], ResT [49]
Combination
Early Conv. [50], CoAtNet [37]
Transformer with Local Attn.
Local Only
HaloNet [51], Swin Transformer [33], VOLO [42]
Local Enhanced Global
TNT [52], Twins [53], Focal Transformer [54], ViL [55]
Hierarchical Transformer
Overlapping Fold
T2T-Transformer [56]
Non-Overlapping Linear
PVT [39]
Down-Sampling by Pooling
PiT [57]
Overlapping Small Conv.
PVT v2 [58], CvT [34]
Deep Transformer
Structure Improvement
CaiT [40], DeepViT [59], Refiner [35]
Loss Regulation
Diverse Patch [60]
Self-Supervised Transformer
Generative Representation
iGPT [61], MST [62], BEIT [63]
Discriminative Representation
MoCo v3 [64], DINO [65], MoBY [66]

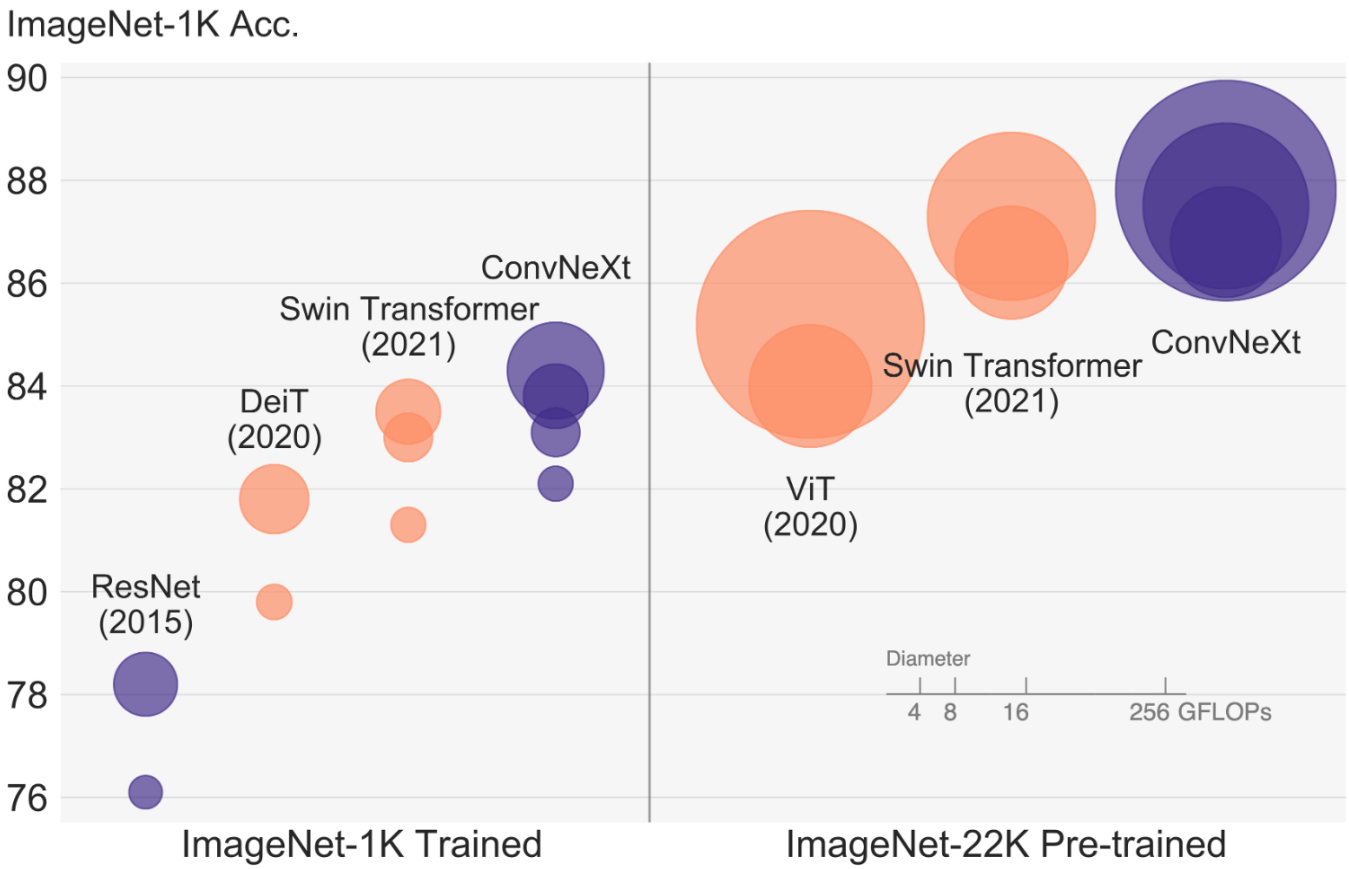

Detection
Transformer Neck
Original Transformer
DETR [28]
Sparse Attention
Deformable DETR [67], ACT [68]
Spatial Prior
One-Stage Detector
SMCA [69]
Conditional DETR [70]
Two-Stage Detector
Two-Stage Deformable DETR [67],
Efficient DETR [71],
Structural Redesign
TSP [72], YOLOS [73]
Self-supervised Learning
UP-DETR [74]
Transformer Backbone
FPT [75], Focal Transformer [54],
PVT [39], ViL [55], Swin Transformer [33]
Segmentation
Patch-Based Transformer
SETR [76], TransUNet [77], SegFormer [78]
Query-Based Transformer
Object Query
Serial Training
Panoptic DETR [28]
Paralleled Training
Cell-DETR [79], VisTR [80]
Cascaded Training
QueryInst [81]
Mask Embedding
Box-Auxiliary Training
ISTR [82], SOLQ [83]
Box-Free Training
Max-DeepLab [29],
Segmenter [84], Maskformer [85]

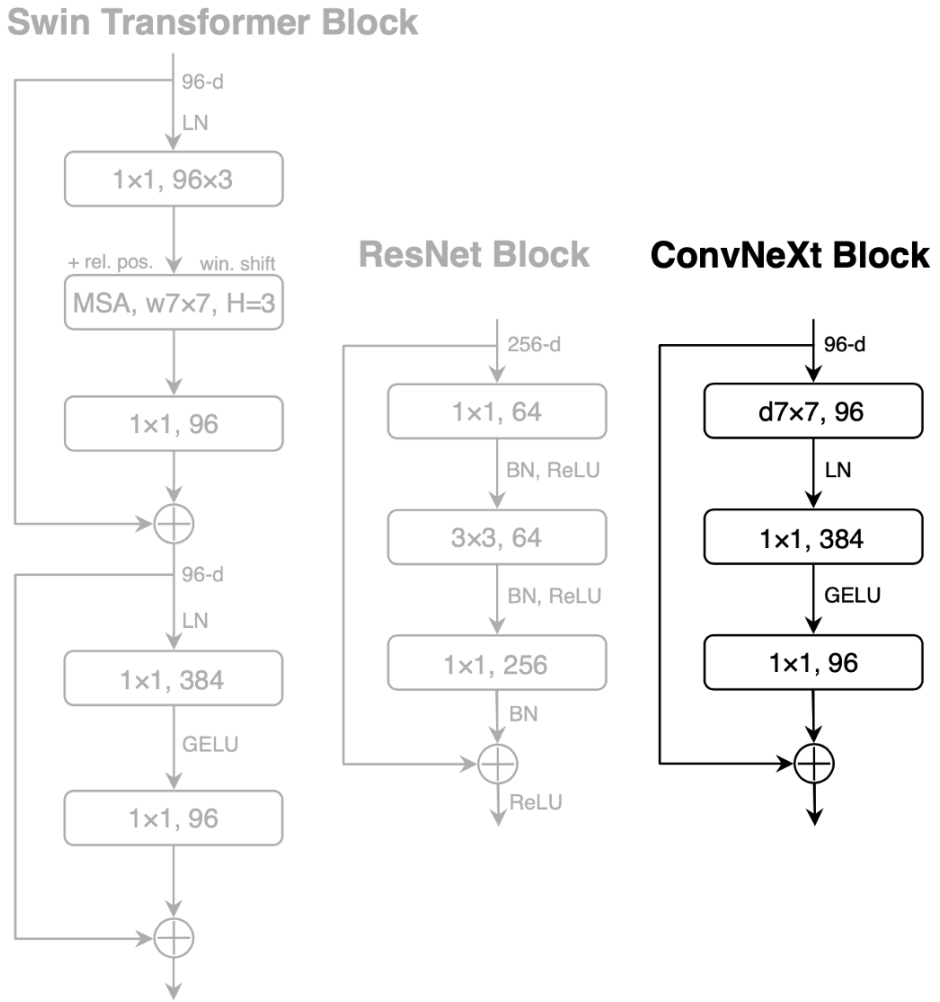
CONVOLUTION STRIKE BACK

A ConvNet for the 2020s

- ConvNeXt: the debate heats up between ConvNets and Transformers for vision!
  - FAIR+BAIR colleagues showing that with the right combination of methods, ConvNets are better than Transformers for vision.
- tricks make complete sense
  - larger kernels, layer norm, fat layer inside residual blocks, one stage of non-linearity per residual block, separate downsampling layers
- "Conv is all you need"?
  - ConvNet (or ConvNeXt) for the first few layers, then something more memory-based and permutation invariant on top, like transformer blocks, for object-based reasoning
  - Like DETR

ConvNeXt





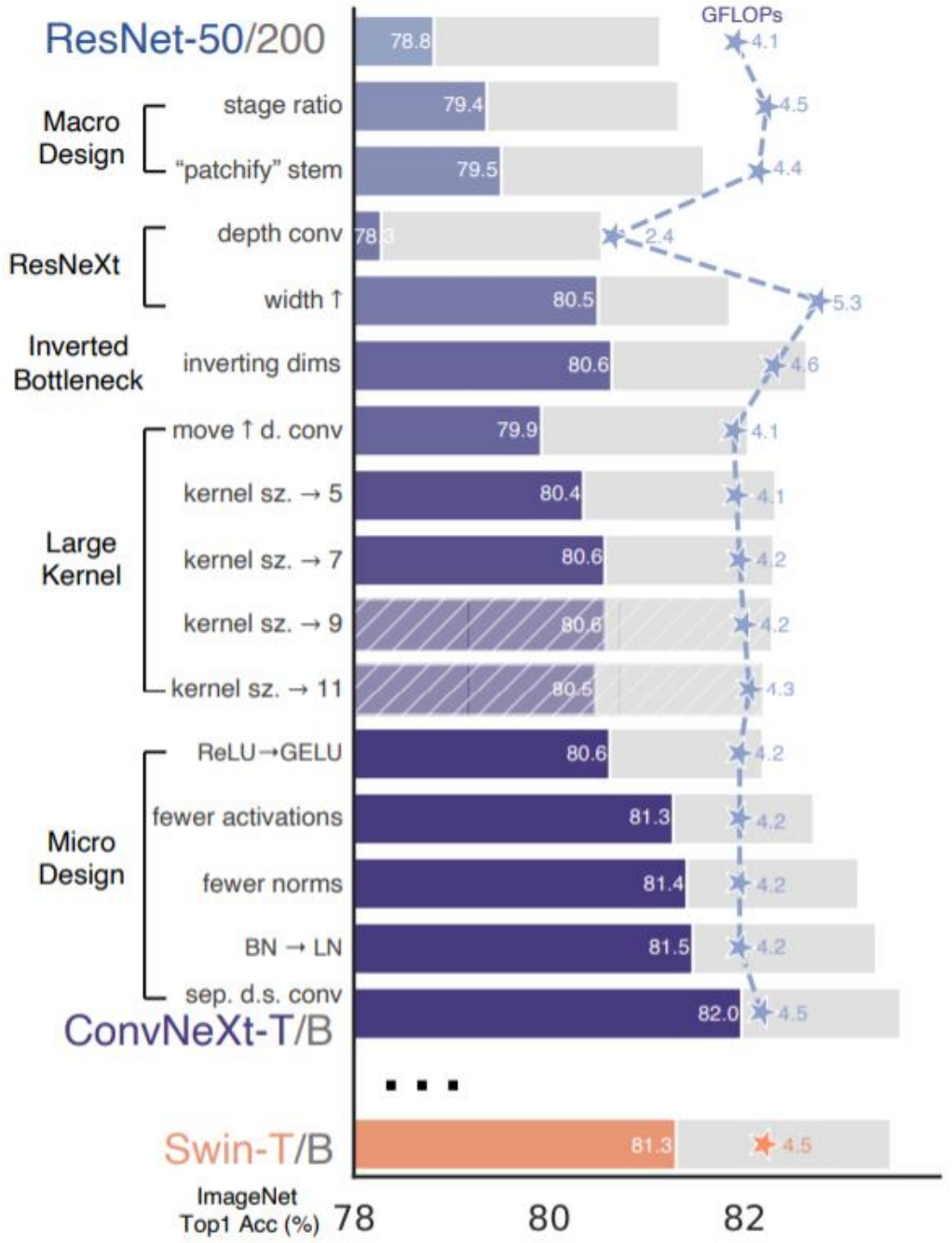
https://github.com/facebookresearch/ConvNeXt

- reexamine the design spaces and test the limits of what a pure ConvNet can achieve
- vanilla ViT model
  - faces many challenges in being adopted as a generic vision backbone
  - The biggest challenge is ViT's global attention design,
    - which has a quadratic complexity with respect to the input size. This might be acceptable for ImageNet classification, but quickly becomes intractable with higher-resolution inputs
  - Hierarchical Transformers like Swin Transformer solve this
    - the essence of convolution is not becoming irrelevant; rather, it remains much desired and has never faded.
    - Cost: a naive implementation of sliding window self attention can be expensive
      - ironic that a ConvNet already satisfies many of those desired properties
  - performance difference is usually attributed to the superior scaling behavior of Transformers

“Modernize” ResNet

- How do design decisions in Transformers impact ConvNets’ performance?

- We modernize a standard ConvNet (ResNet) towards the design of a hierarchical vision Transformer (Swin), without introducing any attention-based modules.
  - The foreground bars are model accuracies in the ResNet-50/Swin-T FLOP regime; results for the ResNet-200/Swin-B regime are shown with the gray bars.
  - A hatched bar means the modification is not adopted. Detailed results for both regimes are in the appendix.
  - Many Transformer architectural choices can be incorporated in a ConvNet, and they lead to increasingly better performance. In the end, our pure ConvNet model, named ConvNeXt, can outperform the Swin Transformer



- Training recipe (close to Deit and Swin Transformer)
  - 90 -> 300 epochs
  - AdamW
  - data augmentation techniques such as Mixup [85], Cutmix [84], RandAugment [12], Random Erasing [86], and regularization schemes including Stochastic Depth [33] and Label Smoothing [65]

(pre-)training config	ConvNeXt-T/S/B/L ImageNet-1K 224^2	ConvNeXt-B/L/XL ImageNet-22K 224^2
optimizer	AdamW	AdamW
base learning rate	4e-3	4e-3
weight decay	0.05	0.05
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
batch size	4096	4096
training epochs	300	90
learning rate schedule	cosine decay	cosine decay
warmup epochs	20	5
warmup schedule	linear	linear
layer-wise lr decay [6, 10]	None	None
randaugment [12]	(9, 0.5)	(9, 0.5)
label smoothing [65]	0.1	0.1
mixup [85]	0.8	0.8
cutmix [84]	1.0	1.0
stochastic depth [34]	0.1/0.4/0.5/0.5	0.1/0.1/0.2
layer scale [69]	1e-6	1e-6
gradient clip	None	None
exp. mov. avg. (EMA) [48]	0.9999	None

Table 5. **ImageNet-1K/22K (pre-)training settings**. Multiple stochastic depth rates (e.g., 0.1/0.4/0.5/0.5) are for each model (e.g., ConvNeXt-T/S/B/L) respectively.

pre-training config	ConvNeXt-B/L ImageNet-1K 224^2	ConvNeXt-B/L/XL ImageNet-22K 224^2
fine-tuning config	ImageNet-1K 384^2	ImageNet-1K 224^2 and 384^2
optimizer	AdamW	AdamW
base learning rate	5e-5	5e-5
weight decay	1e-8	1e-8
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
batch size	512	512
training epochs	30	30
learning rate schedule	cosine decay	cosine decay
layer-wise lr decay	0.7	0.8
warmup epochs	None	None
warmup schedule	N/A	N/A
randaugment	(9, 0.5)	(9, 0.5)
label smoothing	0.1	0.1
mixup	None	None
cutmix	None	None
stochastic depth	0.8/0.95	0.2/0.3/0.4
layer scale	pre-trained	pre-trained
gradient clip	None	None
exp. mov. avg. (EMA)	None	None/None/0.9999

Table 6. **ImageNet-1K fine-tuning settings**. Multiple values (e.g., 0.8/0.95) are for each model (e.g., ConvNeXt-B/L) respectively.

	output size	• ResNet-50	• ConvNeXt-T	○ Swin-T
stem	56×56	7×7, 64, stride 2 3×3 max pool, stride 2	4×4, 96, stride 4	4×4, 96, stride 4
res2	56×56	[1×1, 64 3×3, 64 1×1, 256] × 3	[d7×7, 96 1×1, 384 1×1, 96] × 3	[1×1, 96×3 MSA, w7×7, H=3, rel. pos. 1×1, 96 [1×1, 384 1×1, 96]] × 2
res3	28×28	[1×1, 128 3×3, 128 1×1, 512] × 4	[d7×7, 192 1×1, 768 1×1, 192] × 3	[1×1, 192×3 MSA, w7×7, H=6, rel. pos. 1×1, 192 [1×1, 768 1×1, 192]] × 2
res4	14×14	[1×1, 256 3×3, 256 1×1, 1024] × 6	[d7×7, 384 1×1, 1536 1×1, 384] × 9	[1×1, 384×3 MSA, w7×7, H=12, rel. pos. 1×1, 384 [1×1, 1536 1×1, 384]] × 6
res5	7×7	[1×1, 512 3×3, 512 1×1, 2048] × 3	[d7×7, 768 1×1, 3072 1×1, 768] × 3	[1×1, 768×3 MSA, w7×7, H=24, rel. pos. 1×1, 768 [1×1, 3072 1×1, 768]] × 2
FLOPs		4.1×10^9	4.5×10^9	4.5×10^9
# params.		25.6×10^6	28.6×10^6	28.3×10^6

Table 9. **Detailed architecture specifications** for ResNet-50, ConvNeXt-T and Swin-T.

- depthwise convolution
  - is similar to the weighted sum operation in self-attention, which operates on a per-channel basis, i.e., only mixing information in the spatial dimension.
  - The use of depthwise convolution effectively reduces the network FLOPs and, as expected, the accuracy.
  - Following the strategy proposed in ResNeXt, we increase the network width to the same number of channels as Swin-T's (from 64 to 96).
  - This brings the network performance to 80.5% with increased FLOPs (5.3G).

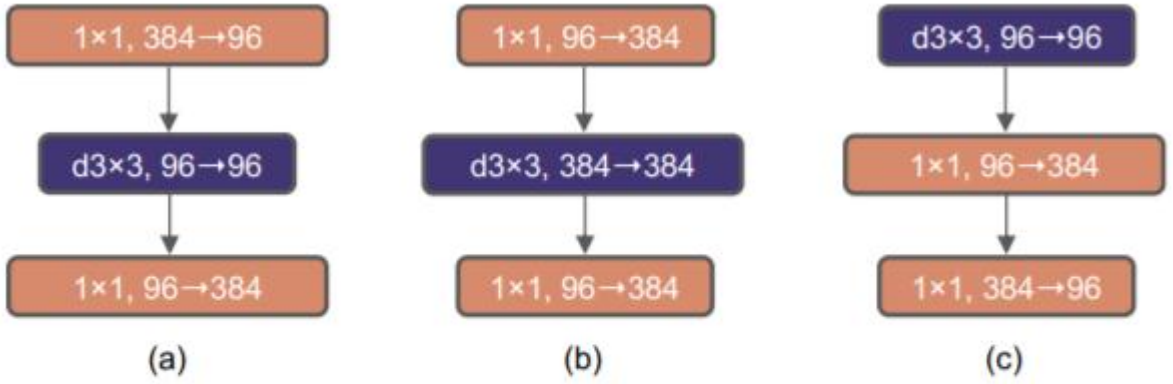


Figure 3. **Block modifications and resulted specifications. (a)** is a ResNeXt block; in **(b)** we create an inverted bottleneck block and in **(c)** the position of the spatial depthwise conv layer is moved up.

every Transformer block is that it creates an inverted bottleneck, i.e., the hidden dimension of the MLP block is four times wider than the input dimension

Moving up depthwise conv layer: The complex/inefficient modules (MSA, large-kernel conv) will have fewer channels, while the efficient, dense 1×1 layers will do the heavy lifting

WHAT DO VISION TRANSFORMERS LEARN? A VISUAL EXPLORATION

- Like CNN
  - features progress from abstract patterns in early layers to concrete objects in late layers

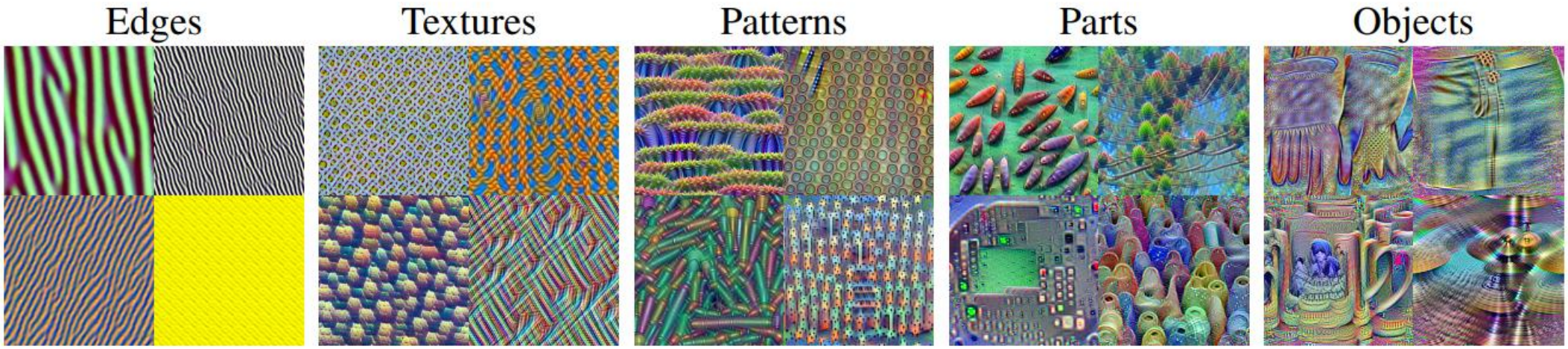


Figure 1: **The progression for visualized features of ViT B-32.** Features from early layers capture general edges and textures. Moving into deeper layers, features evolve to capture more specialized image components and finally concrete objects.

https://github.com/hamidkazemi22/vit-visualization

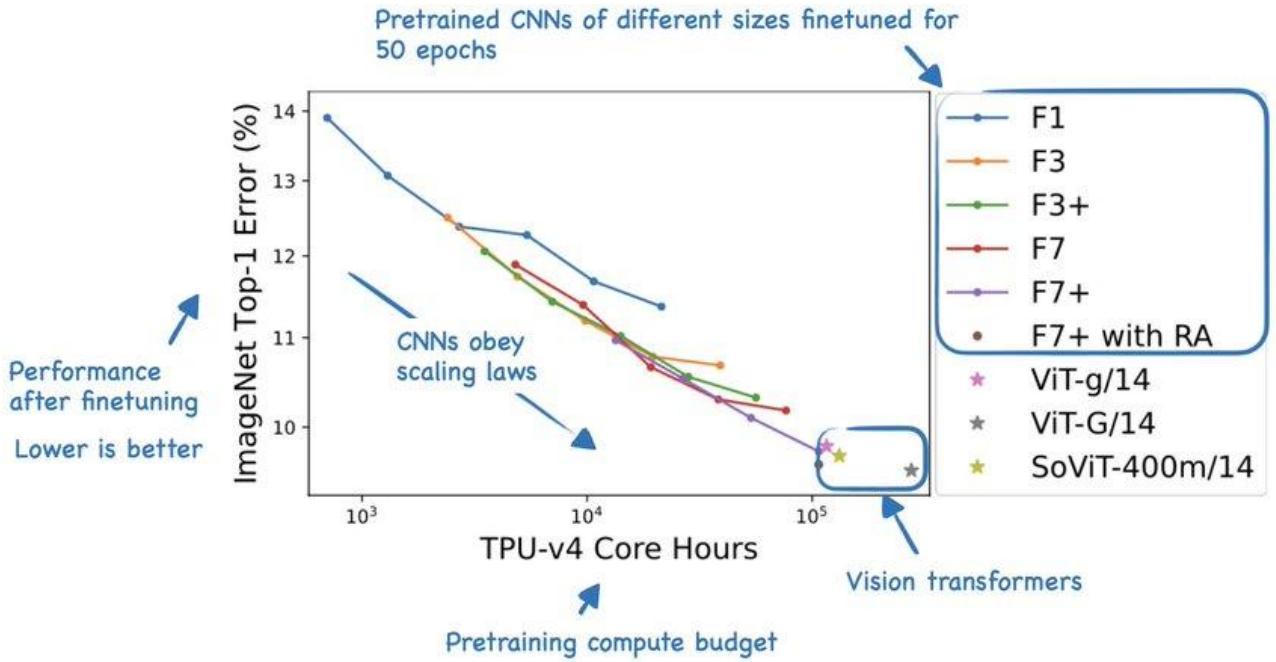
- patch-wise image activation patterns for ViT features essentially behave like saliency maps, highlighting the regions of the image a given feature attends to
- ViTs make better use of background information and rely less on high-frequency, textural attributes
- patch tokens preserve spatial information throughout all layers except the last attention block

(a) (b)

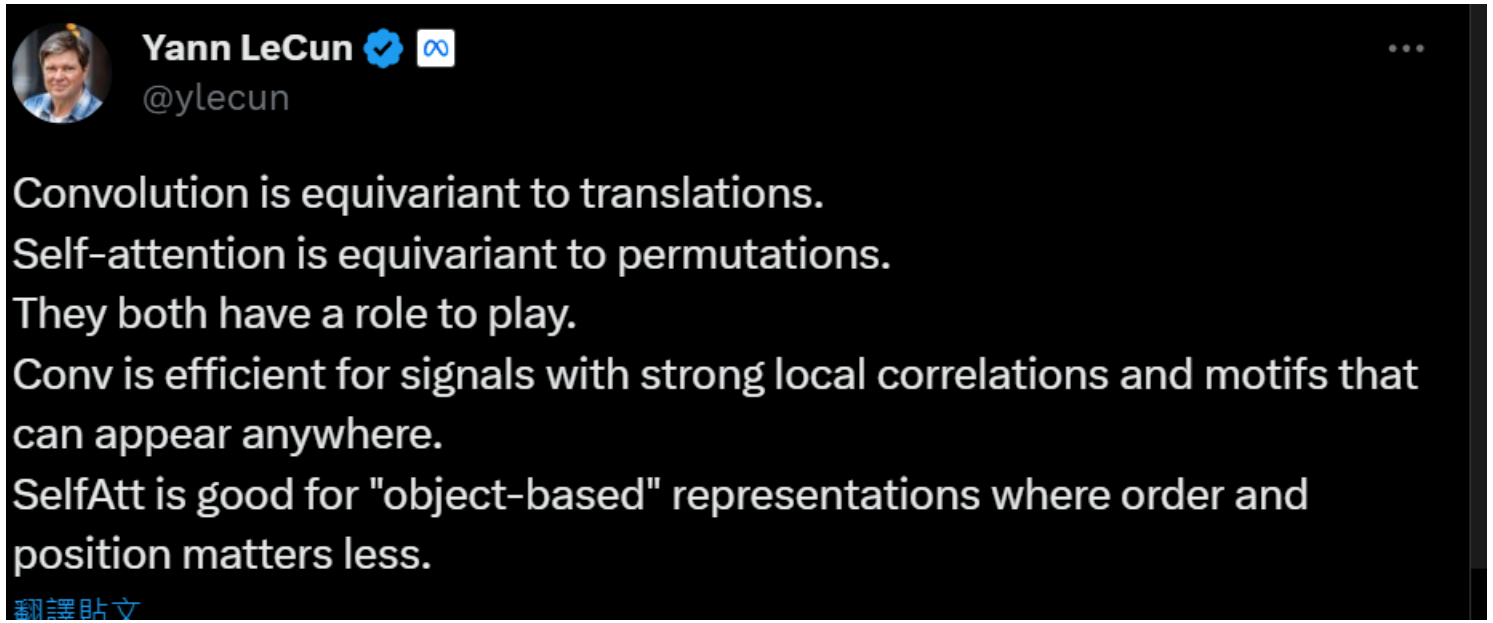
Figure 9: *(a)*: **ViT-B16 detects background features**. *Left:* Image optimized to maximally activate a feature from layer 6. *Center:* Corresponding maximally activating example from ImageNet. *Right:* The image's patch-wise activation map. *(b)*: **An example of an original image and masked-out foreground and background.**

Transformers v.s. Convolution

- ConvNets Match Vision Transformers at Scale
  - when CNNs are pretrained with a compute budget similar to what is typically used for ViTs, they can match the performance of ViTs.



ConvNets Match Vision Transformers at Scale, https://arxiv.org/abs/2310.16764


Yann LeCun
@ylecun
Convolution is equivariant to translations.
Self-attention is equivariant to permutations.
They both have a role to play.
Conv is efficient for signals with strong local correlations and motifs that can appear anywhere.
SelfAtt is good for "object-based" representations where order and position matters less.
翻譯貼文