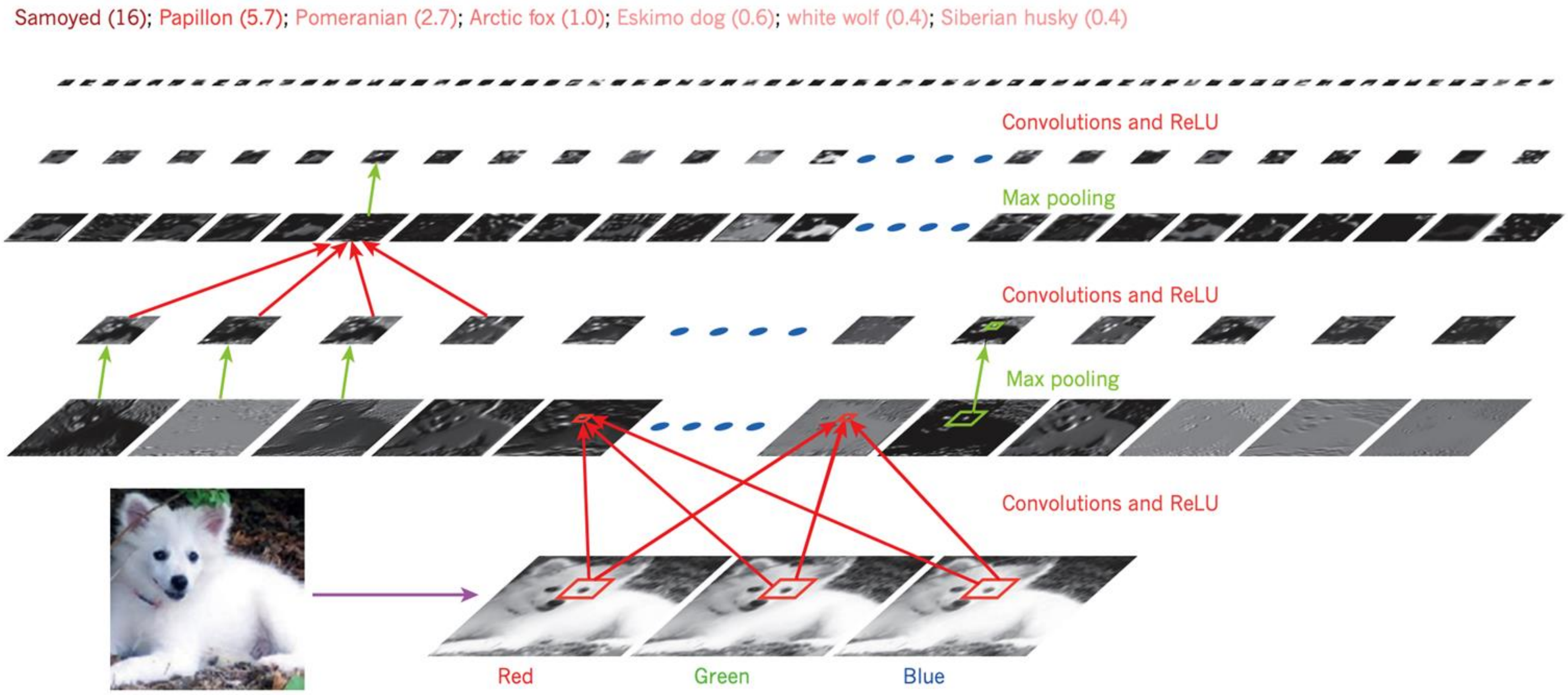
Nested Hierarchical Transformer: Towards Accurate, Data-Efficient and Interpretable Visual Understanding

Zizhao Zhang
Staff Research Software Engineer, Tech Lead
Cloud AI Research, Google

Convolutional Neural Networks for Image Recognition



Convolution with **Pooling** generates **feature maps**

Figure source: https://www.nature.com/articles/nature14539

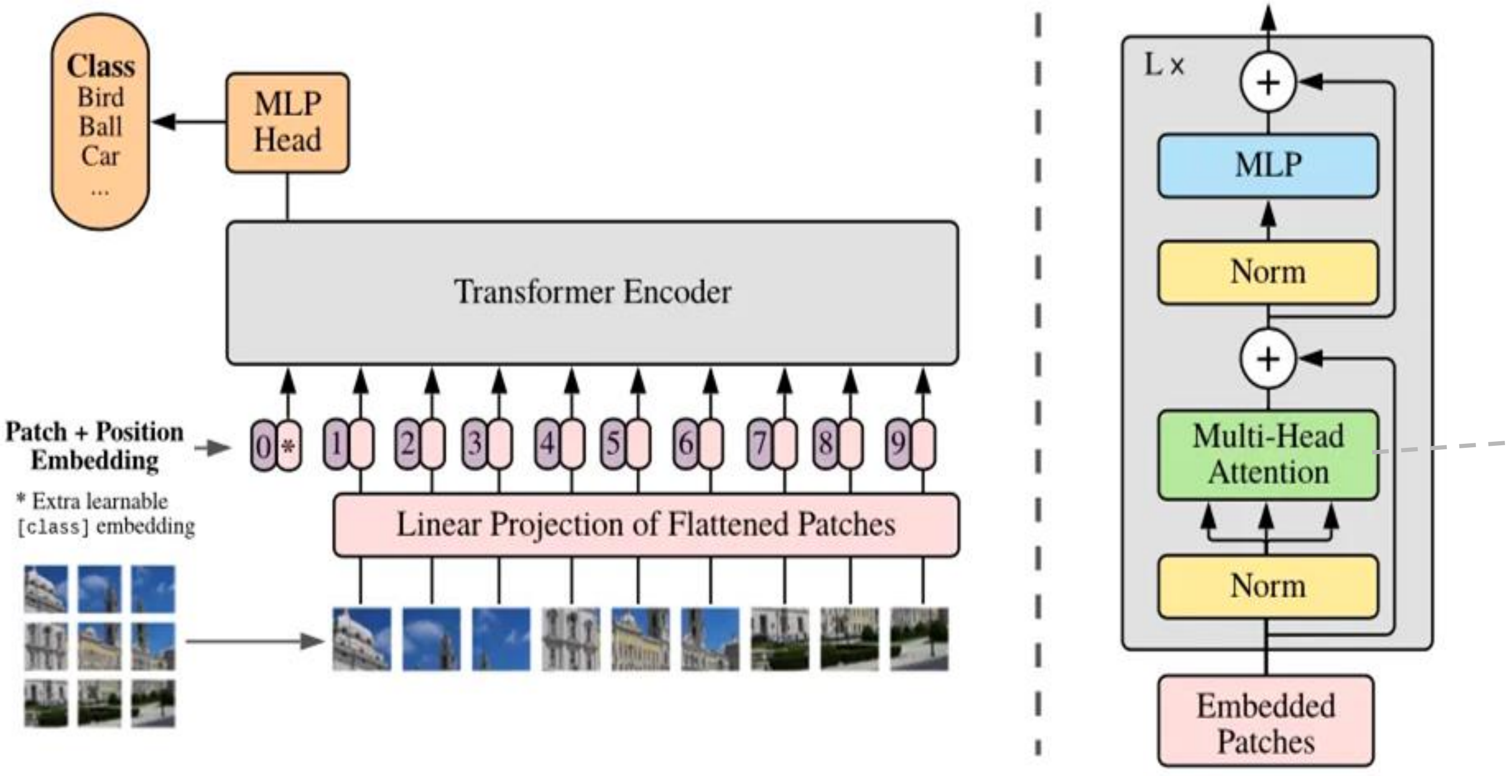
Vision Transformer (ViT)

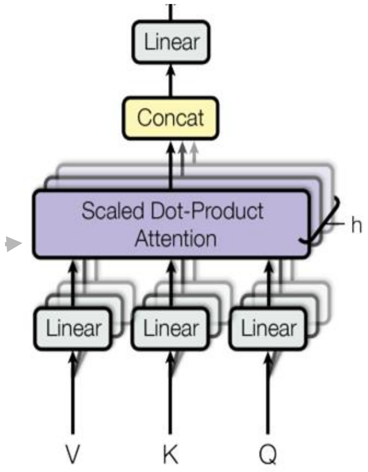
Vision Transformer

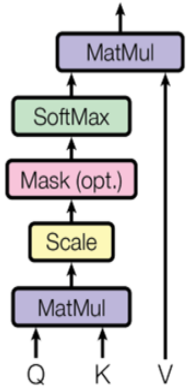
Transformer Encoder

Multi-head attention

Scaled Dot-Product Attention







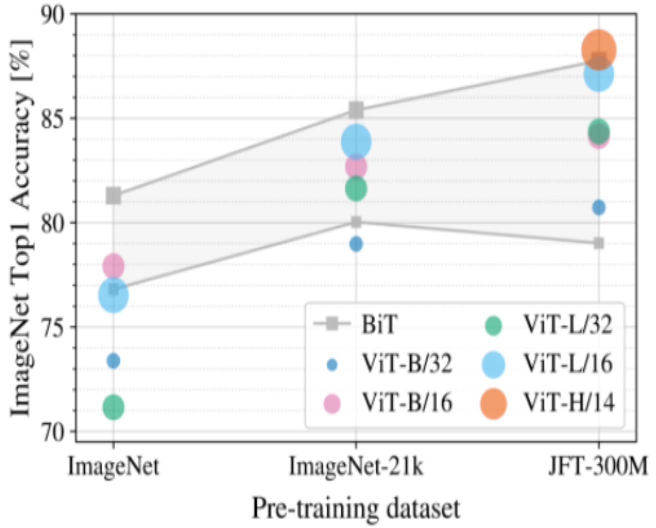
Vision Transformer (ViT)

Self-attention with **tokenized patches** generates sequence features

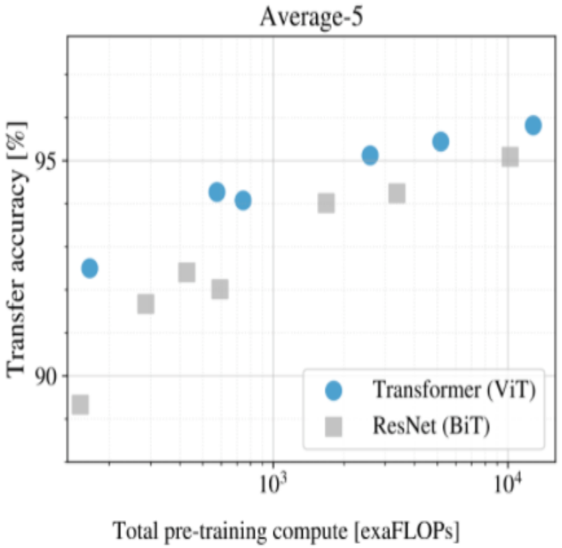
Figure source:
https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html

Advantages

- Easy to scaleup (billions of params).
- Unifies language and vision.



Scale better with larger image pre-training



ViT yields a good performance/compute trade-off.

Figure source:
https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html

Ways to Improve

- **Data efficiency**: ViT usually underperforms ConvNets when given a smaller amount of data.

- **Expensive computation**: Learning self-attention cross all pixels/patches is expensive with long sequences.

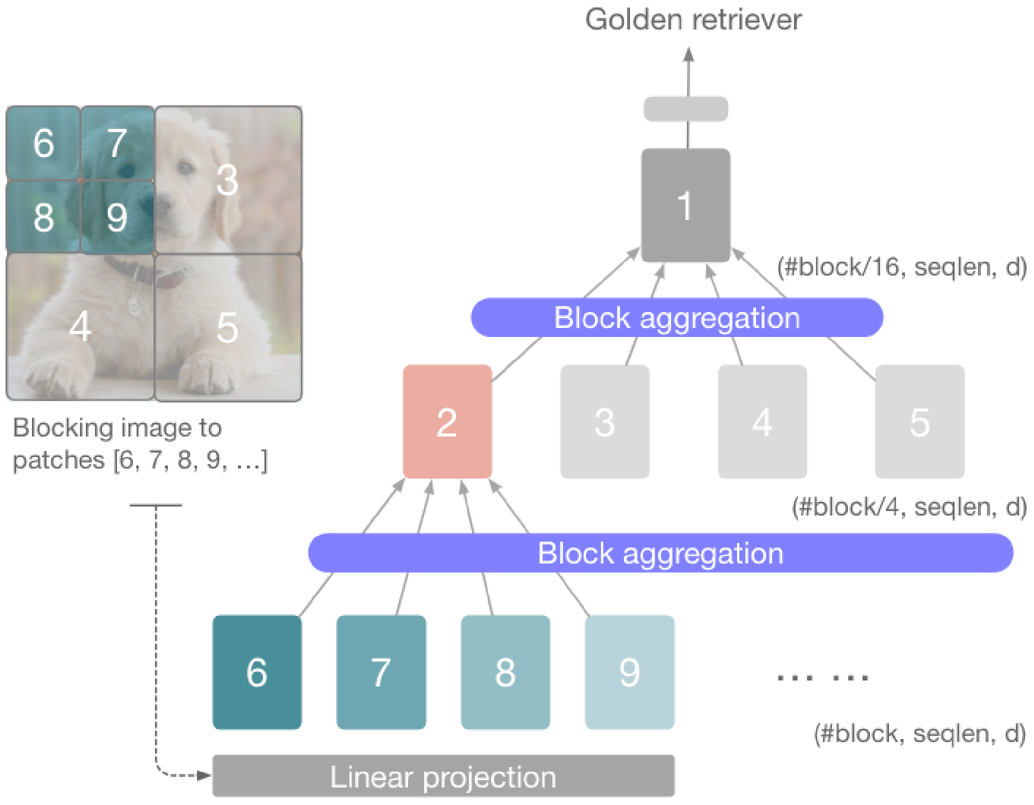
- **Interpretability**: The interpretability of ViT is under-discovered.

Contributions of Our Work

- New concept with simple implementation (**10+** lines of code).
- Improve ViT ImageNet benchmark from 81.8% –> **83.8%** (**20%** reduced params).
- **State-of-the-art** on data efficiency experiments.
- **Interpretable** tree-like structure.
- Speed up training convergence by **3** - **8 times**.

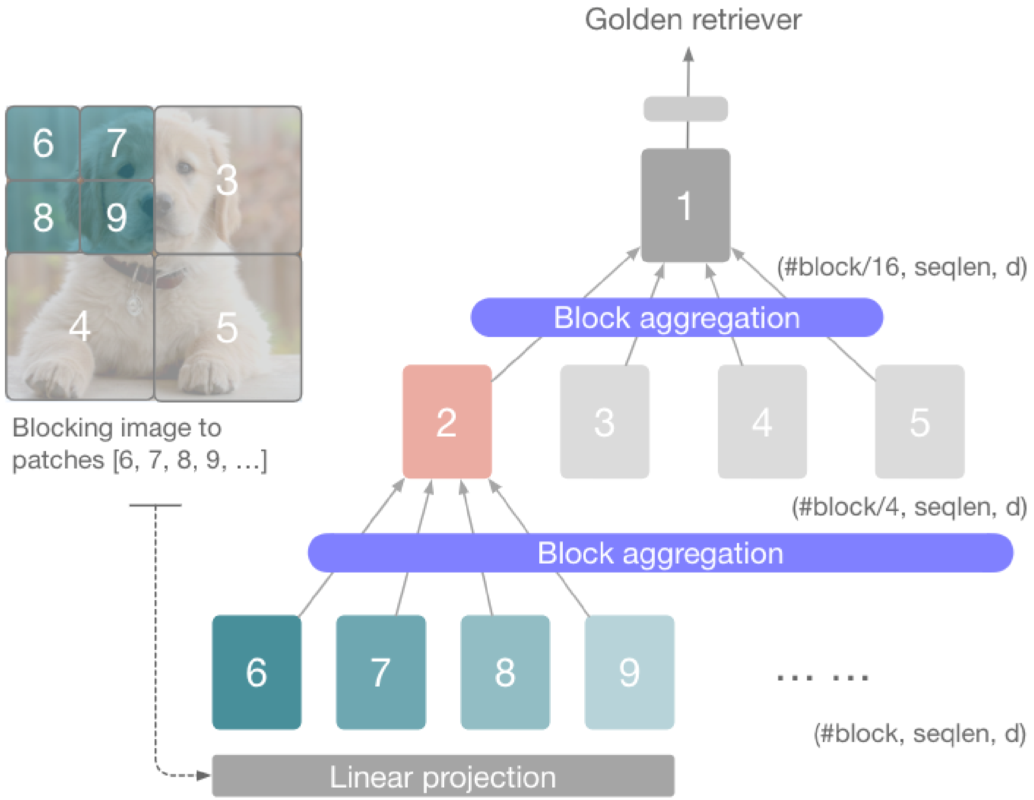
Aggregating Nested Transformer (NesT)

Aggregating Nested Transformer (NesT)



- Achieving non-local communication via the proposed **aggregation function**.
- Decouple **local feature learning** and **global feature communication** processes.
- It resembles **decision tree-like structure** that offers interpretation benefits.
- Easy to implement.

NesT Pseudo-code



Pseudo code: NesT

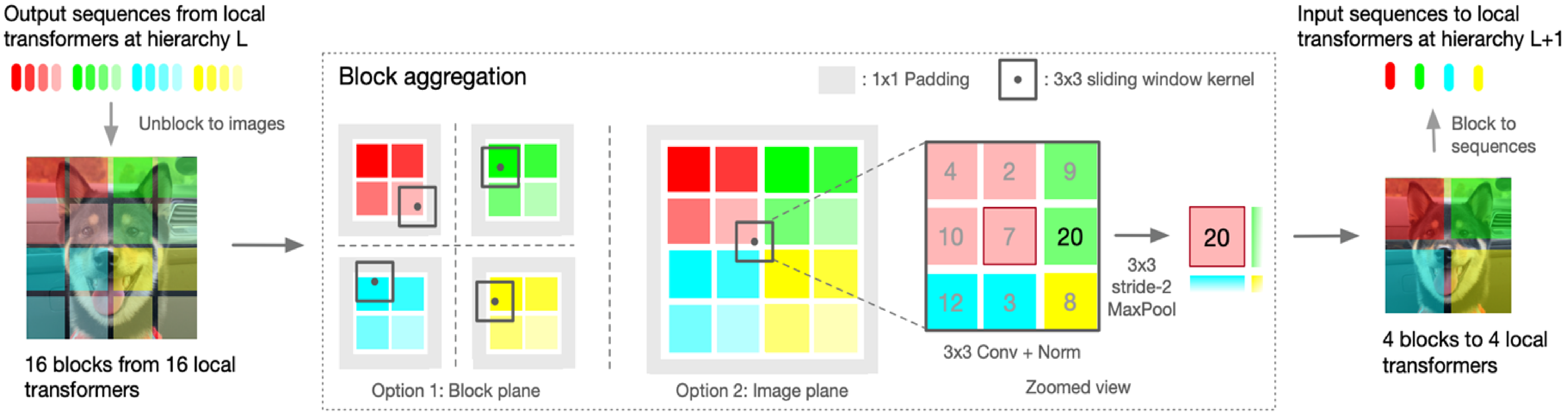
```
# embed and block image to (#block,seqlen,d)
x = Block(PatchEmbed(input_image))

for i in range(num_hierarchy):
  # apply transformer layers T_i within each block
  # with positional encodings (PE)
  y = Stack([T_i(x[0] + PE_i[0]), ...])
  if i < num_hierarchy - 1:
     # aggregate blocks and reduce #block by 4
    x = Aggregate(y, i)

h = GlobalAvgPool(x) # (1,seqlen,d) to (1,1,d)
logits = Linear(h[0,0]) # (num_classes,)

def Aggregate(x, i):
  z = UnBlock(x) # unblock seqs to (h,w,d)
  z = ConvNormMaxPool_i(x) # (h/2,w/2,d)
  return Block(z) # block to seqs
```

Aggregation Functions: Design Matters!



- Block aggregation reduces the spatial size by 2x2.
- **Small kernels** on **image plane** is important.

ImageNet Results

ImageNet benchmark

Arch. base	Method	#Params	Top-1 acc. (%)
Convolutional	ResNet-50	25M	76.2
	RegNetY-4G	21M	80.0
	RegNetY-16G	84M	82.9
Transformer full-attention	ViT-B/16	86M	77.9
	DeiT-S	22M	79.8
	DeiT-B	86M	81.8
Transformer local-attention	Swin-T	29M	81.3
	Swin-S	50M	83.0
	Swin-B	88M	83.3
	NesT-T	17M	81.5
	NesT-S	38M	83.3
	NesT-B	68M	**83.8**

Three different sizes: **T**: Tiny, **S**: Small, **B**: Base

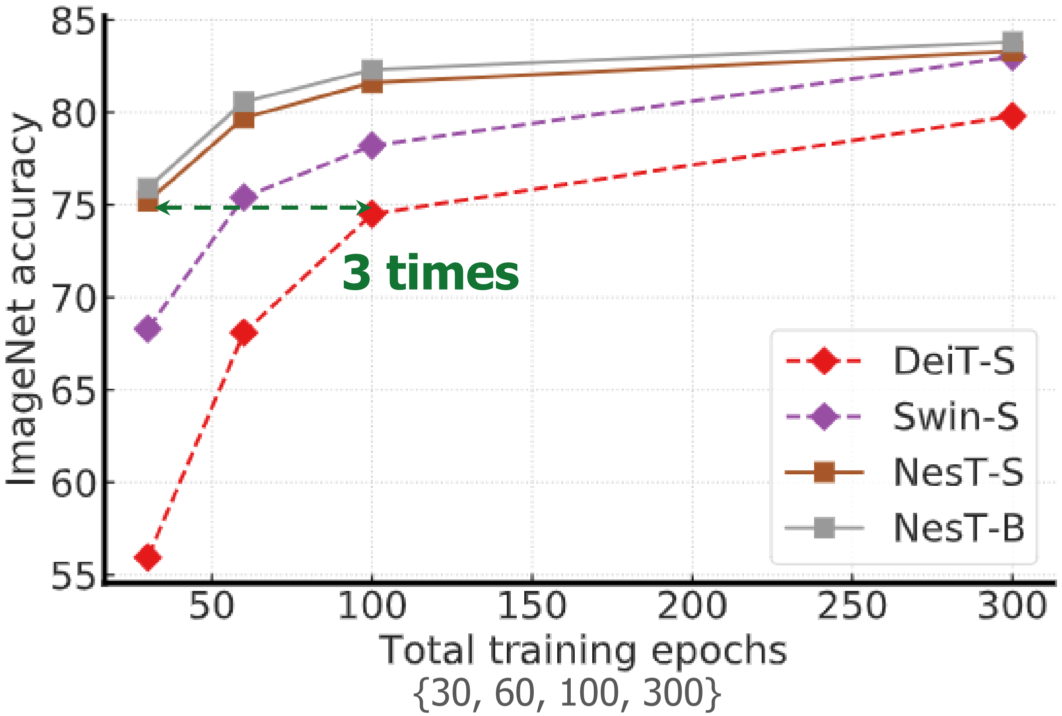
ImageNet benchmark with ImageNet-22K pre-training

	ViT-B/16	Swin-B	Nest-B
ImageNet Acc. (%)	84.0	86.0	**86.2**

Note: **DeiT** is **ViT** trained with strong data augmentations (which are used by most following papers). In rest of presentation, we mostly compare with DeiT.

DeiT: Training data-efficient image transformers & distillation through attention, ICML2021

Convergence and Effects of Data Augmentation

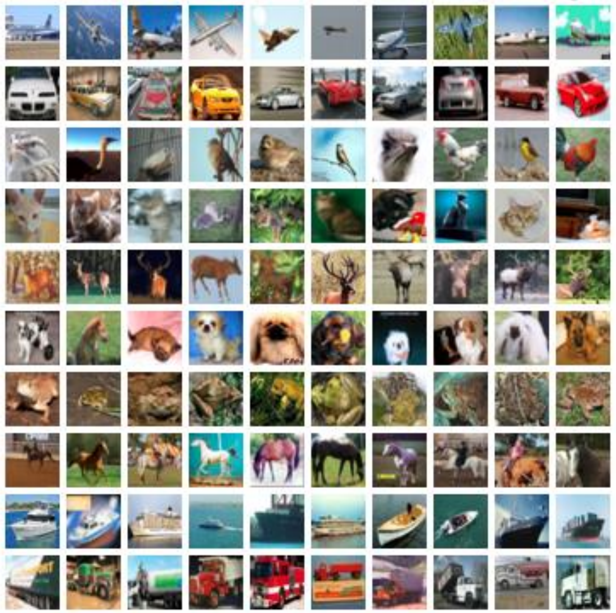


Augmentation Removed	ImageNet Accuracy (%)	
	DeiT-B	NesT-T
None	81.8	81.5
RandomErasing	4.3	81.4
RandAugment	79.6	81.2
CutMix&MixUp	75.8	79.8

NesT uses less training time to reach similar performance.

NesT is much more robust to data augmentation ablations.

Data Efficiency Experiments: CIFAR Results



CIFAR10/100 datasets have 60k images with 32x32 resolution.

Arch. base	Method	C10 (%)	C100 (%)
Convolutional	Pyramid-164-48	95.97	80.70
	WRN28-10	95.83	80.75
Transformer full-attention	DeiT-T	88.39	67.52
	DeiT-S	92.44	69.78
	DeiT-B	92.41	70.49
	PVT-T	90.51	69.62
	PVT-S	92.34	69.79
	PVT-B	85.05*	43.78*
	CCT-7/3×1	94.72	76.67
Transformer local-attention	Swin-T	94.46	78.07
	Swin-S	94.17	77.01
	Swin-B	94.55	78.45
	NesT-T	96.04	78.69
	NesT-S	96.97	81.70
	NesT-B	**97.20**	**82.56**

PVT: Pyramid vision transformer, Wang et al., ICCV2021
CCT: Escaping the Big Data Paradigm with Compact Transformers, Hassani el al.,Arxiv, 2021

Google Research

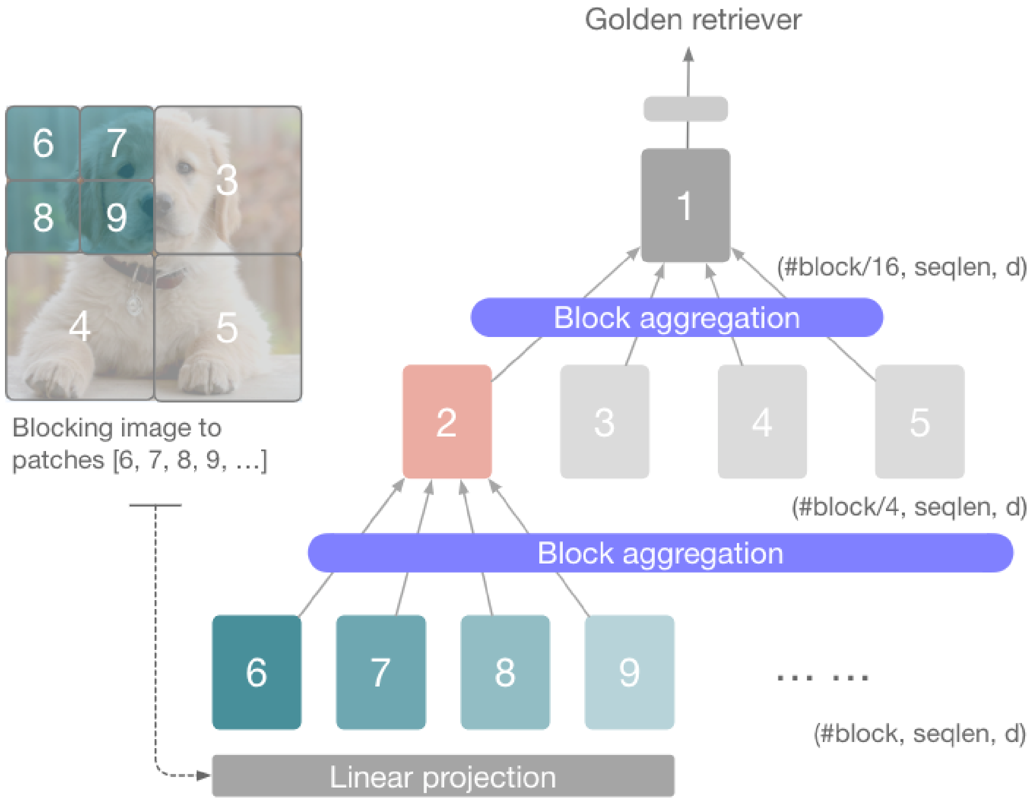
Interpretability

"Deep neural networks usually do not explain their predictions, which is a barrier to their adoption in the real world."

Interpretability of NesT

- **Tree Traversal** to locate the class-aware decision path.
- **Class Activation Map** (CAM) to locate objects.

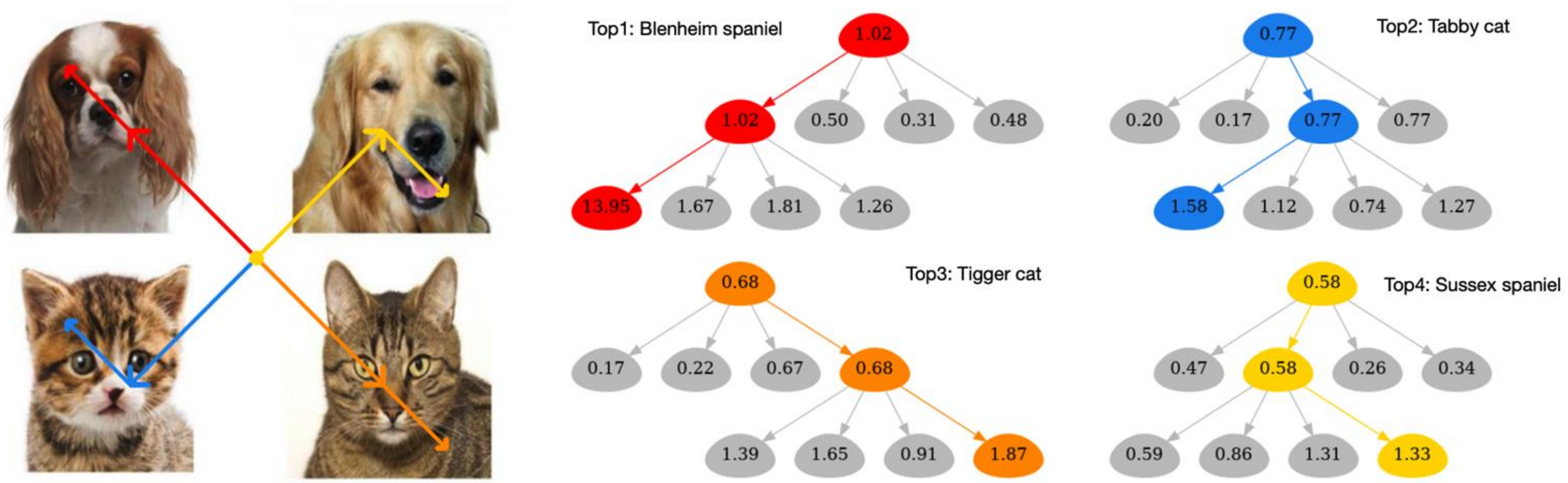
GradCAT: Interpretability via Tree Traversal



- Each node only processes information over corresponding regions.
- Block aggregation combines information of adjacent nodes.
- It resumes a decision tree-like structure that naturally has interpretability benefits.

GradCAT: Interpretability Visualization

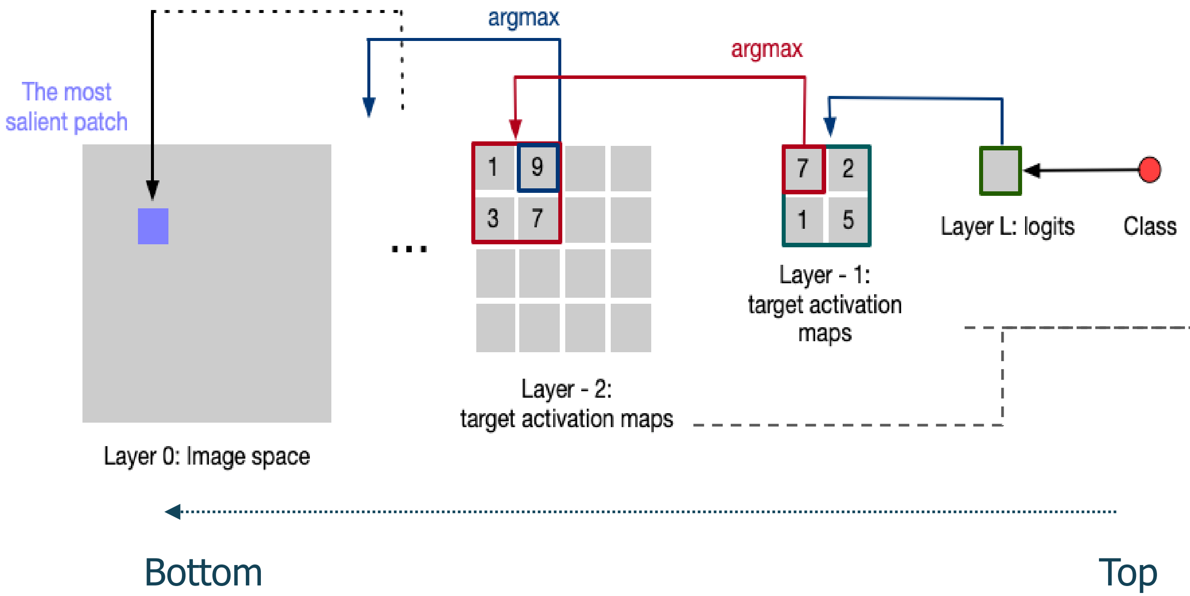
GradCAT: Interpretability Visualization



Given the left input image (containing four animals), the figure visualizes the top-4 class traversal results (4 colors) using an ImageNet-trained NesT (with three tree hierarchies).

Each tree node denotes the averaged activation value.

Gradient-based Class-aware Tree-traversal (GradCAT)



Algorithm 1: GradGAT

Define: A_l denotes the feature maps at hierarchy l. Y_c is the logit of predicted class c. $[\cdot]_{2\times2}$ indexes one of 2×2 partitions of input maps.

Input: $\{A_l | l = 2, ..., T_d\}, \alpha_{T_d} = A_{T_d}, P = []$

Output: The traversal path P from top to bottom

for $l = [T_d, ..., 2]$ **do**

$h_l = \alpha_l \cdot (-\frac{\partial Y_c}{\alpha_l})$ *# obtain target activation maps*

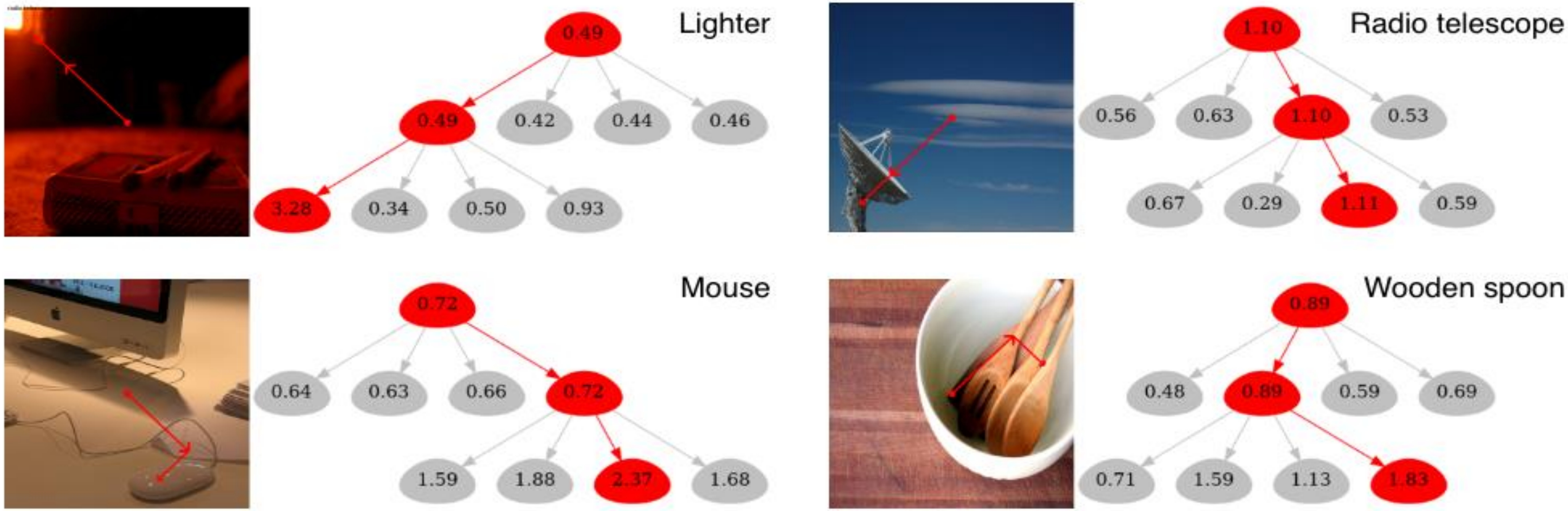
$\hat{h}_l = \mathrm{AvgPool}_{2\times2}(h_l) \in \mathbb{R}^{2\times2}$

$n_l^* = \arg\max \hat{h}_l, \quad P = P + [n_l^*]$ *# pick the maximum index*

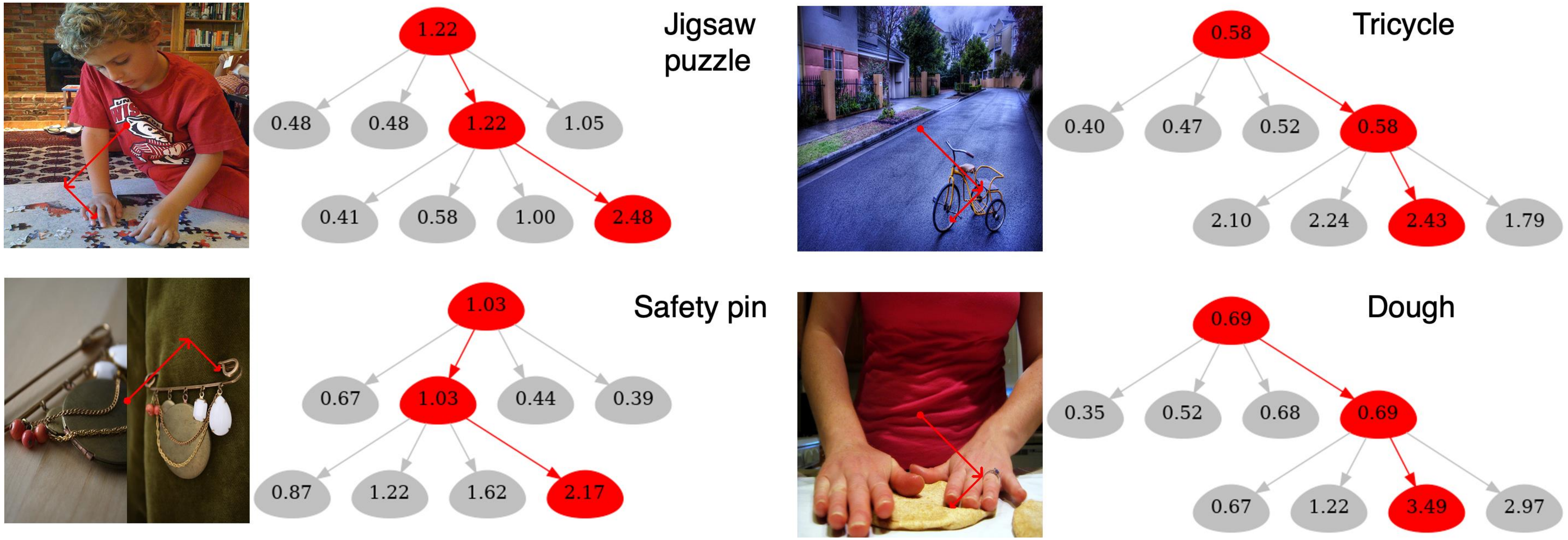
$\alpha_l = A_l[n_l^*]_{2\times2}$ *# obtain the partition for the index*

end for

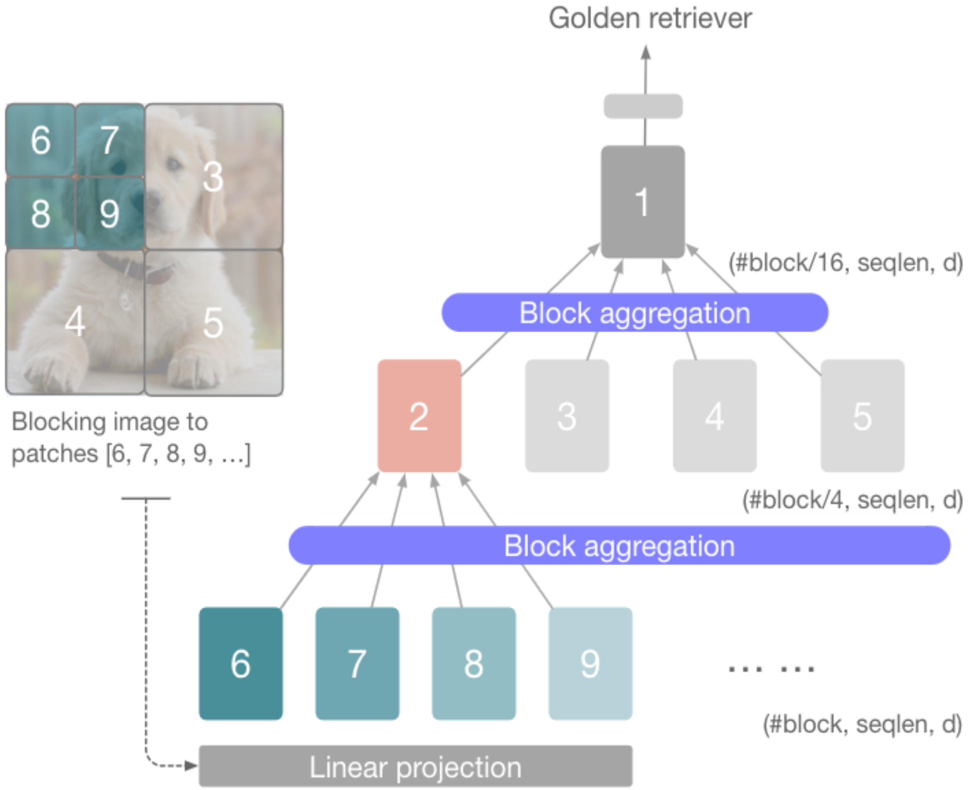
GradCAT Results on ImageNet



GradCAT Results on ImageNet



Class Activation Map (CAM) Visual Object Attention



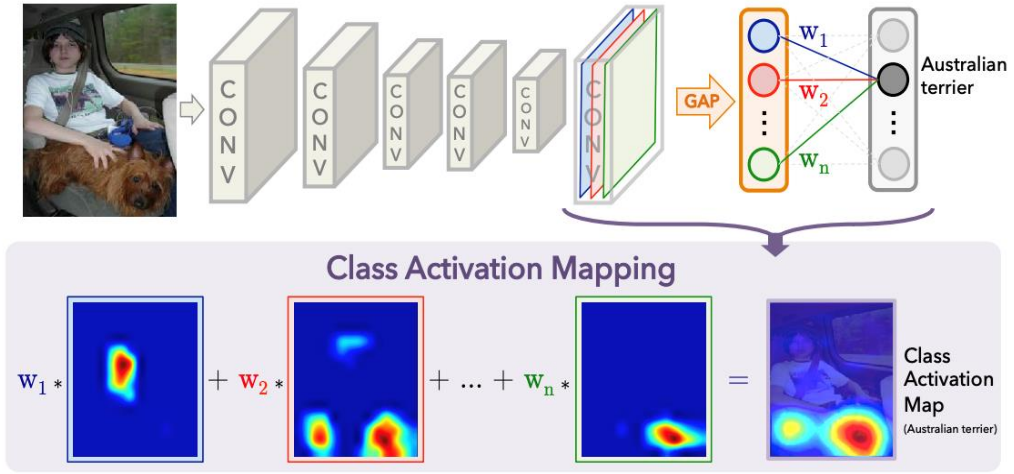
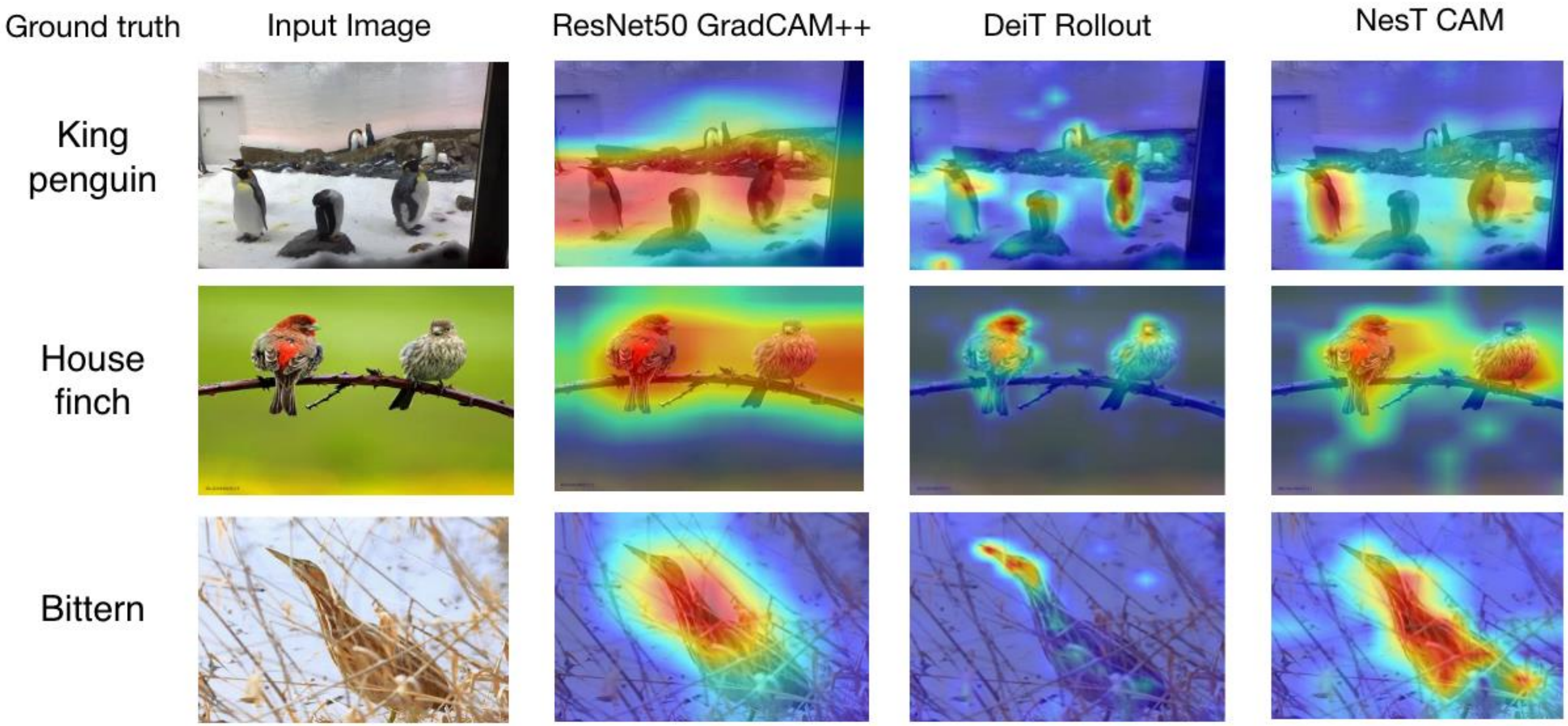


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Learning Deep Features for Discriminative Localization, Zhou et al. **CVPR2016**

Qualitative Comparison Results

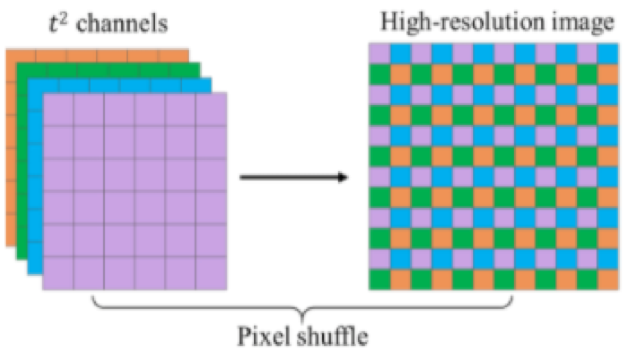


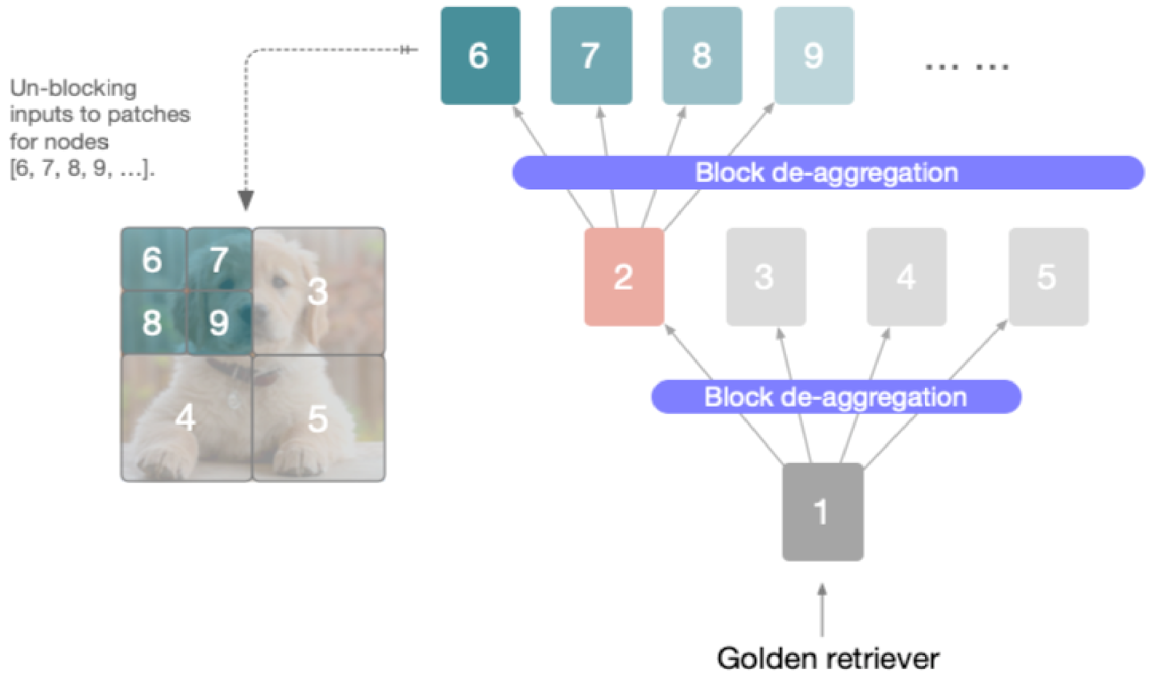
Google Research

Apply NesT to Image Generation

- Replace **Block Aggregation** with **Block De-aggregation**.

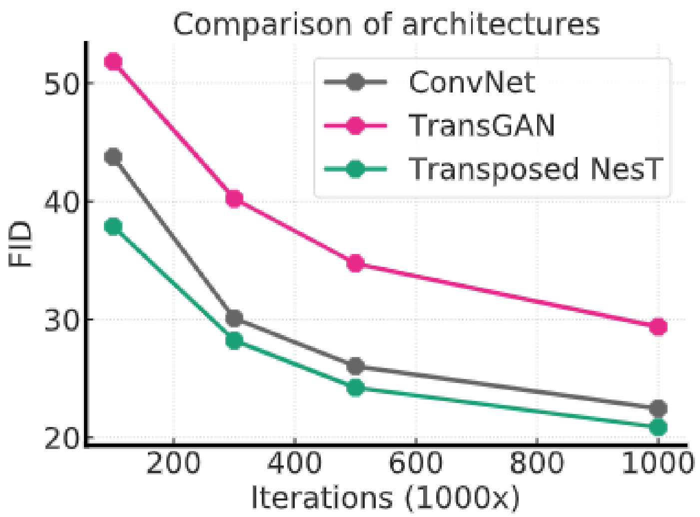
- Use **Pixel Shuffle** to achieve de-aggregation (i.e., upsampling).

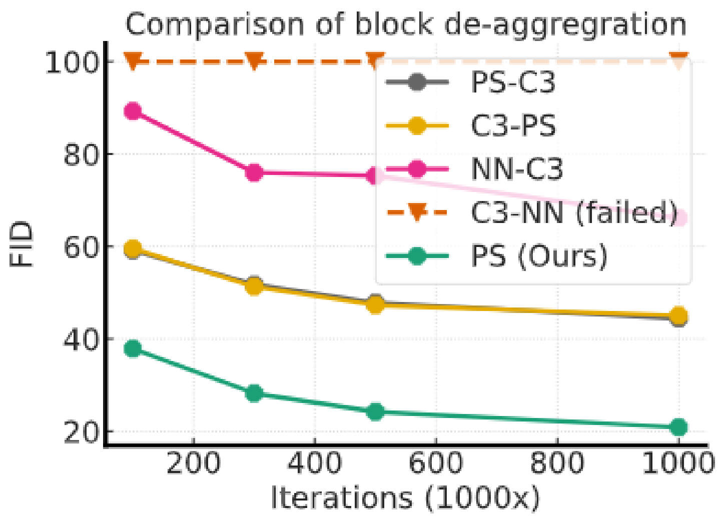




Pixel Shuffle: Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, Shi et al., **CVPR2016**

Image Generation on ImageNet





Method	#Params (millions)	Throughput * (images/s)
Convnet [63]	77.8M	709.1
TransGAN [28]	82.6M	67.7
Transposed NesT	74.4M	523.7

*Measure on single V100 GPU

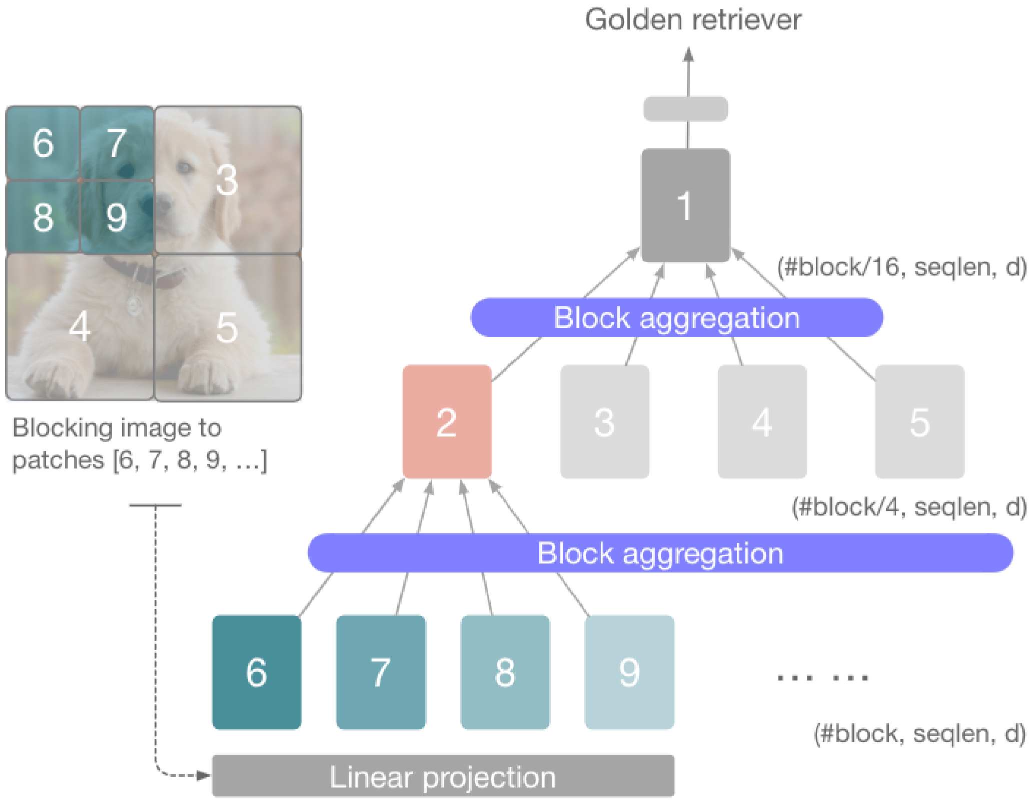
FID: Fréchet inception distance

Different de-aggregation designs:
PS: Pixel Shuffle
C3: 3x3 transpose convolution
NN: Nearest neighbor

- Transposed NesT firstly demonstrates ViT-based architecture can achieve faster convergence than ConvNet-based architecture for image generation.
- See Improved Transformer for High-Resolution GANs, **NeurIPS2021,** for extended work on this task.

Google Research

Conclusion



- A novel architecture that simplifies previous designs via the proposed aggregation function.
- A new interpretability method that make NesT interpretable by tree traversal.
- Competitive ImageNet results and SoTA data-efficiency results.
- Faster convergence and low sensitivity to data augmentations.
- Easy to generalize to other applications.

Resources

Main paper, AAAI'22 Oral

PDF

https://arxiv.org/pdf/2105.12723.pdf

Github (code+pretrained models)

https://github.com/google-research/nested-transformer

Blog post

https://ai.googleblog.com/2022/02/nested-hierarchical-transformer-towards.html

Reference

Vision Transformer

https://arxiv.org/pdf/2010.11929.pdf

Training data-efficient image transformers & distillation through attention

https://arxiv.org/pdf/2012.12877.pdf

Improved Transformer for High-Resolution GANs

https://arxiv.org/pdf/2106.07631.pdf