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

ICLR 2021

2021. 7. 8
Dajin Han

Index

- Introduction
- Preliminaries
- Motivation
- Methods
- Experiments
- QA

Introduction

Key Concepts

Key Concepts

- New **Architecture** for Image Recognition Task

- **Vision Transformer (VIT)**
  - **Pre-train, fine-tune**
  - **Scalability (no sign of saturating performance)**
  - Computational efficiency
  - Efficient implementations

Preliminaries

Transformer, Related Works

Transformer on NLP

- Attention Is All You Need
- Method
  - **Self-Supervision**
  - **Self-Attention**
- Pros.
  - **Scalability**
  - **Computational Efficiency**
  - Efficient implementations

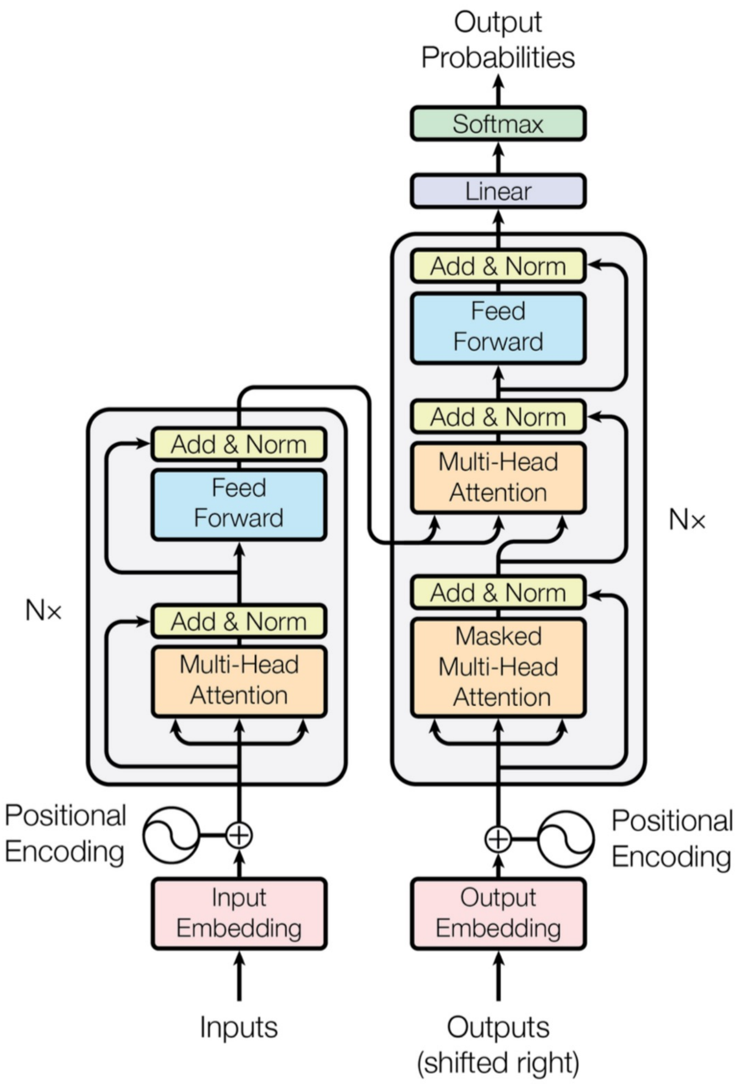


Figure 1: The Transformer - model architecture.

Applications of Transformers

- Self-Attention on CNN
- Patch Extraction from Image
- Combining CNN with Self-Attention
- Image GPT

Applications of Transformers

- **Self-Attention on CNN**
    1. Pixel to every pixel
    2. Pixel to neighborhood pixels
    3. Sparse Transformer
- Patch Extraction from Image
- Combining CNN with Self-Attention
- Image GPT

Applications of Transformers

- Self-Attention on CNN
- **Patch Extraction from Image**
  - Extract patches of size 2x2
  - Full self-attention on top
- Combining CNN with Self-Attention
- Image GPT

Applications of Transformers

- Self-Attention on CNN
- Patch Extraction from Image
- **Combining CNN with Self-Attention**
  - Augmenting feature maps
  - Processing the output of a CNN using self-attention
- Image GPT

Applications of Transformers

- Self-Attention on CNN
- Patch Extraction from Image
- Combining CNN with Self-Attention
- **Image GPT**
  - Reduce image resolution and color space
  - Trained in an unsupervised fashion
  - Fine-tuned

Additional Dataset

- Allows to achieve SOTA results on standard benchmarks

- How CNN performance scales with dataset size
- CNN transfer learning from large scale datasets

Motivation

Better application of Transformer to Recognition Task

Prior Studies

- PROS of Transformer architecture
  - Self-Attention
  - Self-Supervised

- LIMIT : No good application to computer vision tasks

Vision Transformer

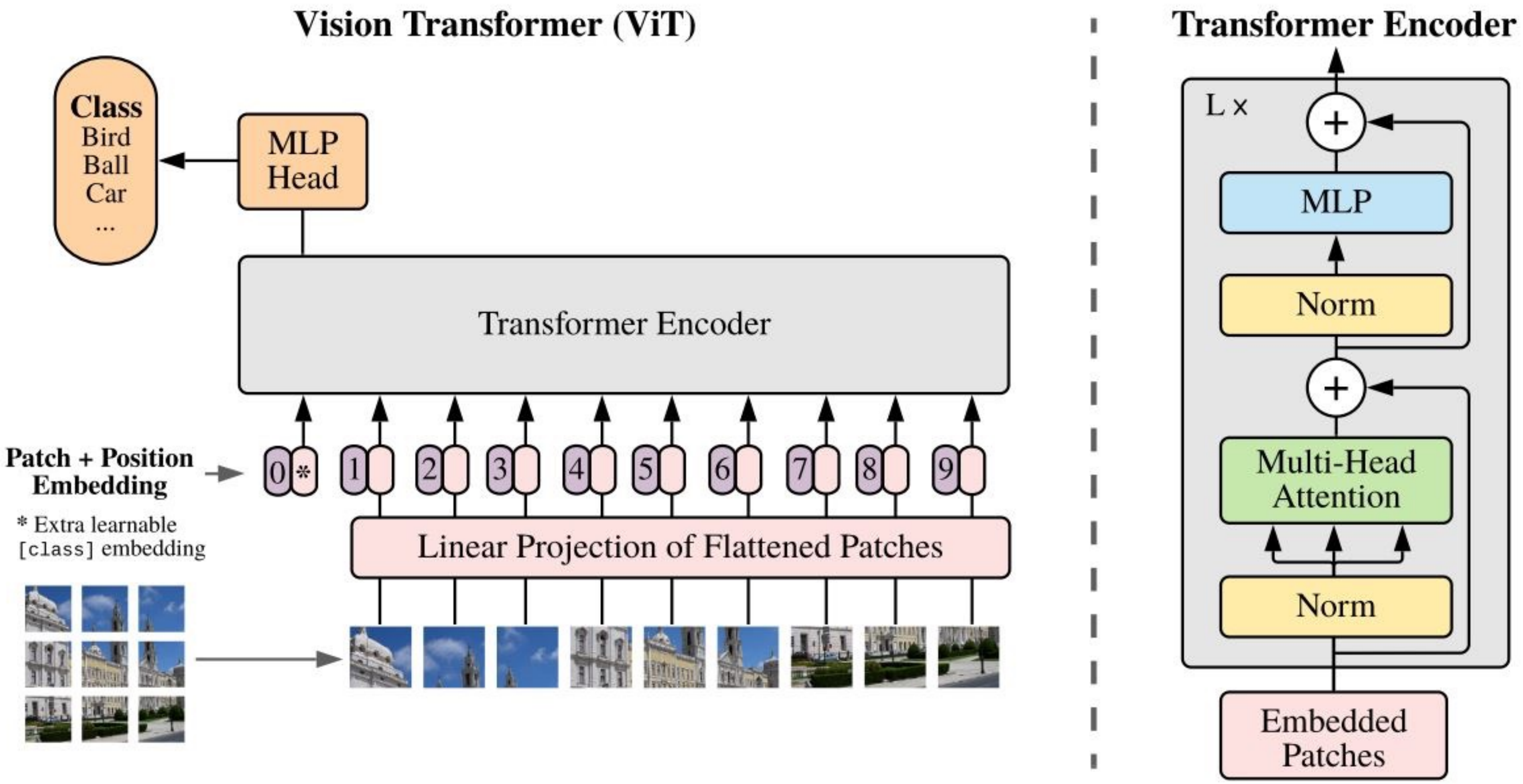
- Apply **transformer** on **Recognition** Task

- With Transformer Network
- With Transformer Pre-training method
- With Transformer Fine-tuning method
- With Transformer Attention module

Method

Architecture, Embedding, Encoder, Inductive Bias

Architecture



Methods

1. Split an image into **patches**
2. **Embedding** each patch
3. Make **sequence** of embedded patches
4. Put sequence into the **Encoder**
5. **MLP header**

$$\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$$

$$N = HW/P^2$$

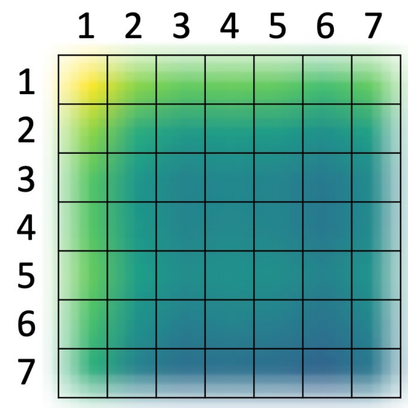
C : number of channels

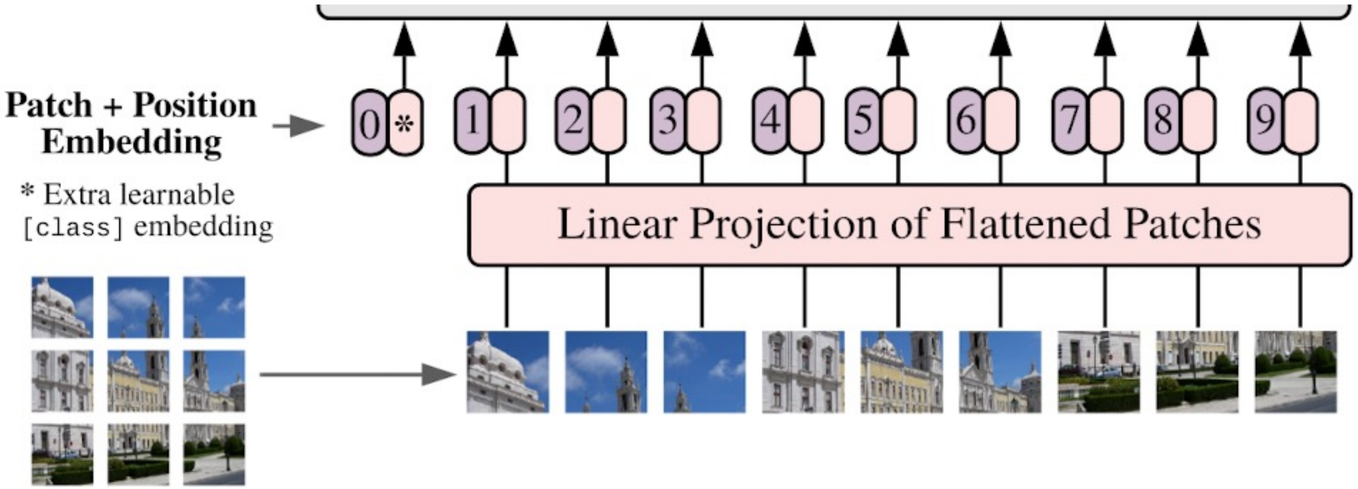
(P ,P) : resolution of image patch

N : number of patches

Methods

1. Split an image into **patches**
2. **Embedding each patch**
3. Make **sequence** of embedded patches
4. Put sequence into the **Encoder**
5. **MLP header**





Methods

1. Split an image into **patches**
2. **Embedding** each patch
3. Make **sequence** of embedded patches
4. Put sequence into the **Encoder**
5. **MLP header**

Methods

1. Split an image into **patches**
2. **Embedding** each patch
3. Make **sequence** of embedded patches
4. Put sequence into the **Encoder**
5. **MLP header**

MSA	: Multihead Self-Attention
MLP	: Multi Layer Perceptron
LN	: Layernorm
D	: constant latent vector size
L	: total layer numbers
E	: Embeddings

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E}\in\mathbb{R}^{(P^2\cdot C)\times D},\, \mathbf{E}_{pos}\in\mathbb{R}^{(N+1)\times D} \tag{1}$$

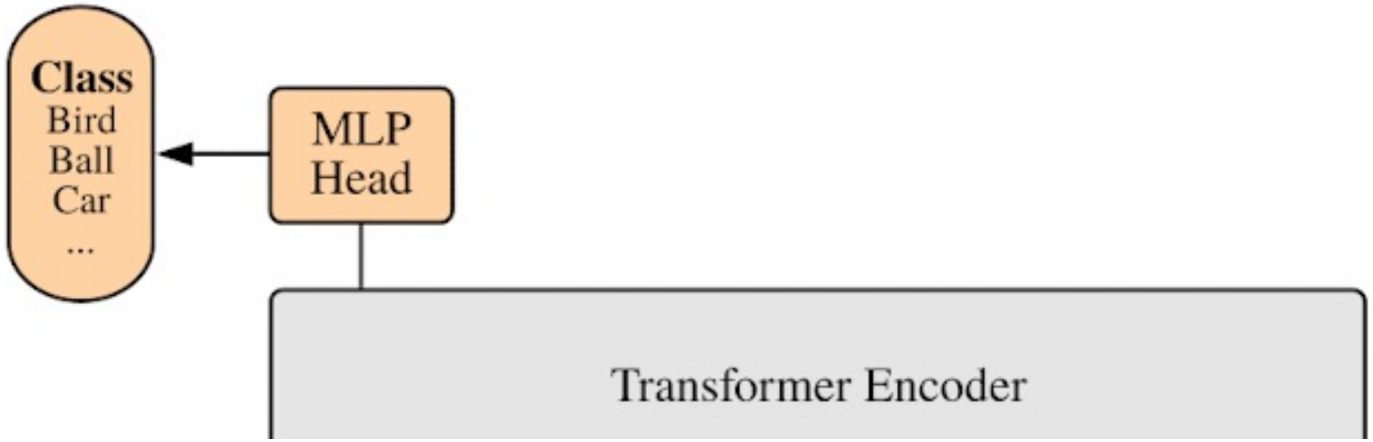
$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1\ldots L \tag{2}$$

$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1\ldots L \tag{3}$$

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \tag{4}$$

Methods

1. Split an image into **patches**
2. **Embedding** each patch
3. Make **sequence** of embedded patches
4. Put sequence into the **Encoder**
5. **MLP header**



Inductive Bias

- Less image-specific inductive bias than CNNs
- CNN
  - Two-dimensional neighborhood structure
  - Translation equivariance
- ViT
  - MLP : local and translationally equivariant
  - Self-attention : global

- Solved by larger dataset

Hybrid Architecture

- **Feature map** instead of raw image

1. Split feature map into patches (spatial size 1x1)
2. Apply patch embedding to each patch

Fine-Tuning

1. Split an image into **patches**
2. **Embedding** each patch
3. Make s**equence** of embedded patches
4. Put sequence into the **encoder**
5. **MLP header**

----

1. **Fine tuning with new header**

Fine-Tuning and Higher Resolution

- Fine-tuning
  - Remove pre-trained prediction head
  - Attach a **zero-initialized** feedforward layer
  - Beneficial to fine-tune at **higher resolution than pre-training**

- Higher Resolution
  - Keep the **patch size** the same when feeding images of higher resolution
  - **Larger effective sequence length**
  - Not meaningful pretrained position embeddings
  - **2D interpolation on pre-trained position embeddings**

Experiments

Datasets, Model variants, Comparison to SOTA

Datasets

- Pre-train
  - ILSVRC-2012 ImageNet-k 1.3M
  - ImageNet-21k 14M
  - JFT-18k 303M
- Fine-tuning
  - ImageNet
  - Real labels
  - CIFAR 10/10
  - Oxford-IIIT Pets, Oxford Flowers-102, 19-task VTAB classification suite

Model Variants

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.

Comparison to SOTA

	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** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

Comparison to SOTA

	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

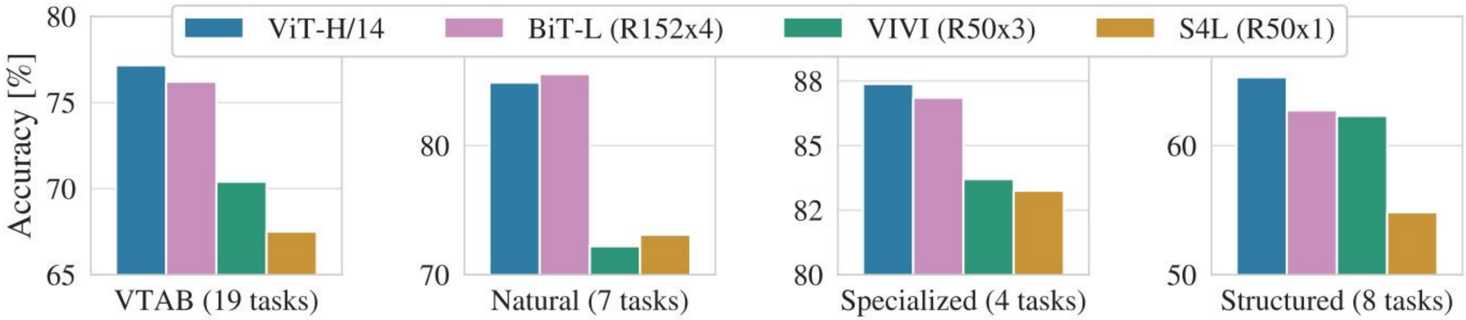
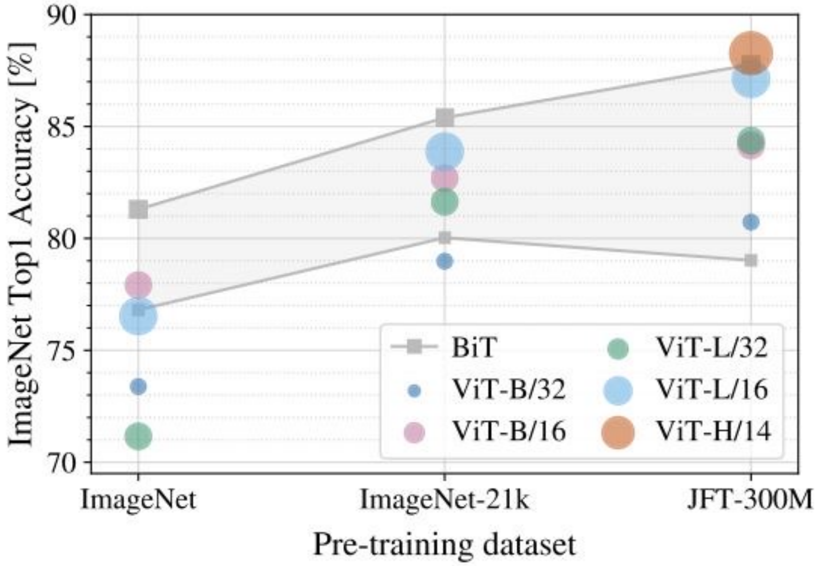


Figure 2: Breakdown of VTAB performance in *Natural*, *Specialized*, and *Structured* task groups.

Pre-training Data Requirements



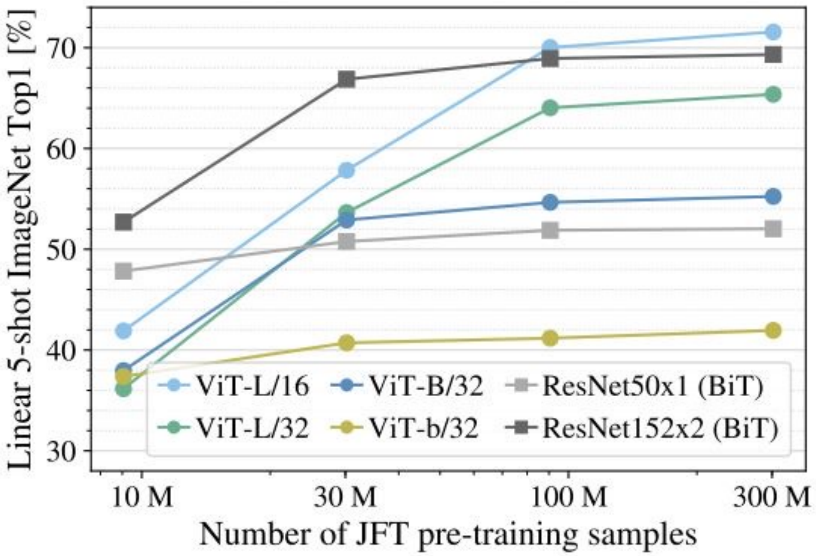


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

Cost Efficiency

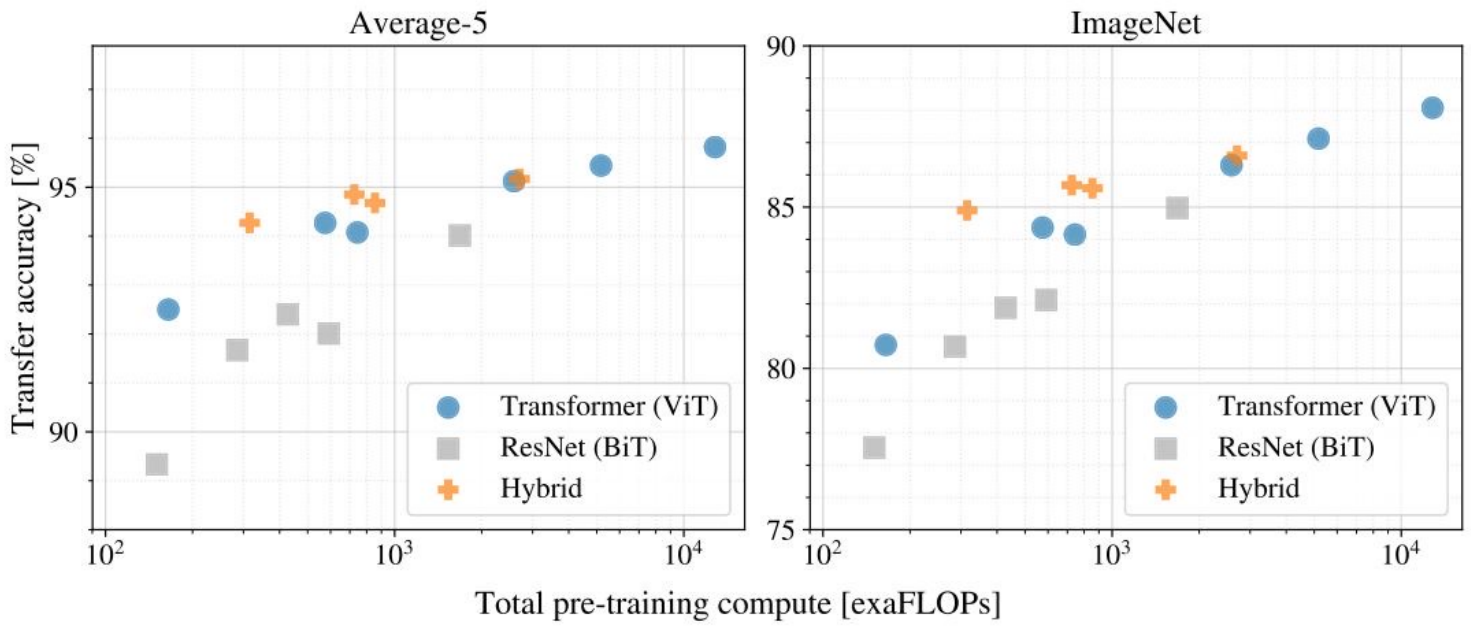


Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanished for larger models.

Cost Efficiency

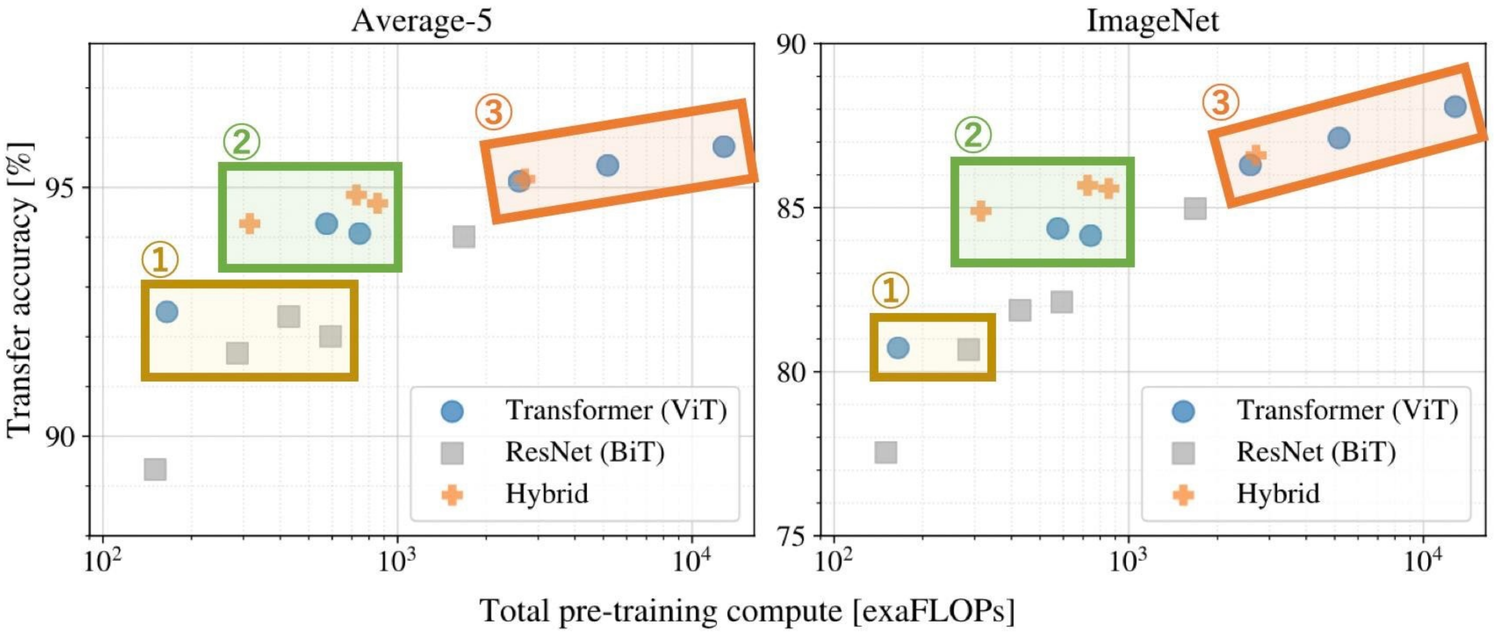
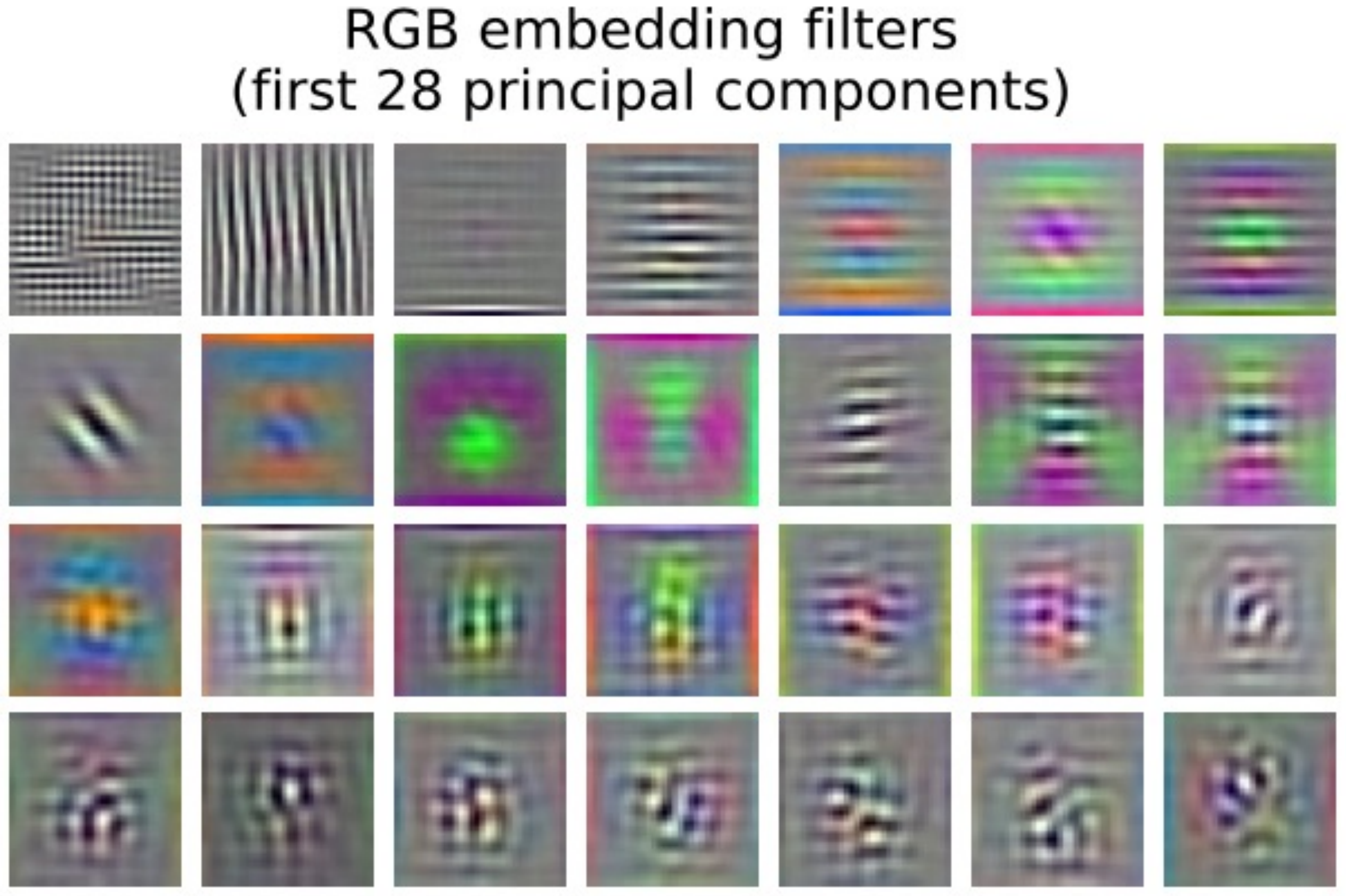


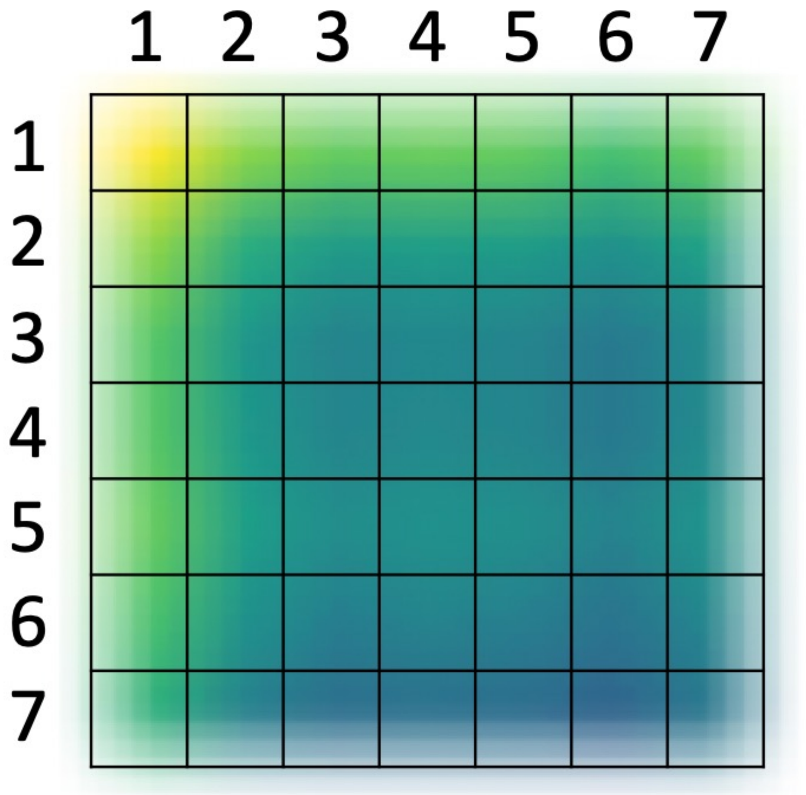
Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanished for larger models.

QA

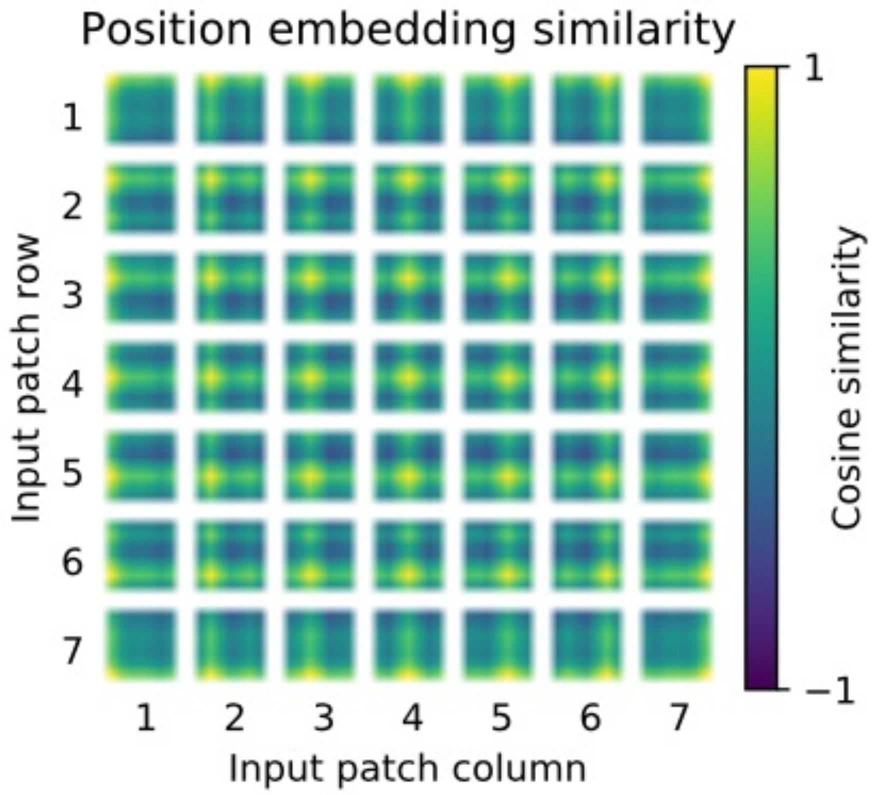
Appendix



Appendix



1 2 3 4 5 6 7
1
2
3
4
5
6
7



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