Vision Transformer [ViT]

AppliedAICourse.com

source

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

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

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

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

https://openreview.net/forum?id=YicbFdNTTy no convolution

ViT → Transformer based [NLP]

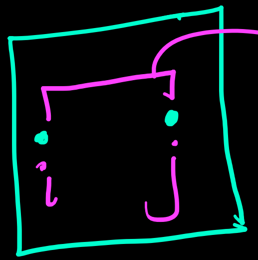
comparable to SOTA CNNS

lower compute-power

Major challenge for non-conv Transformers:

Attention

640x640 pixels



one attention weight for every pair of pixels

$^{640\times640}C_2$ attention

~83.88B

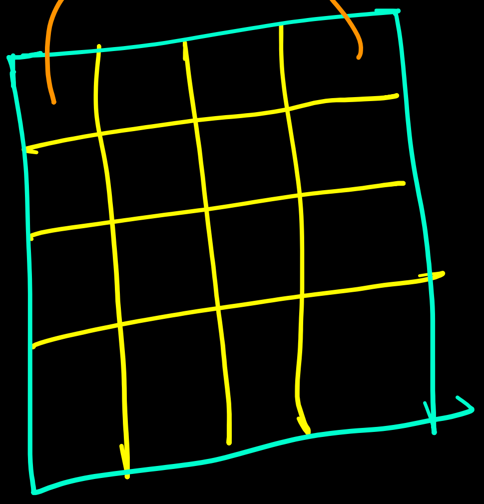
Attention (NLP)

$\longleftarrow$ 512 $\longrightarrow$ $^{512}C_2$

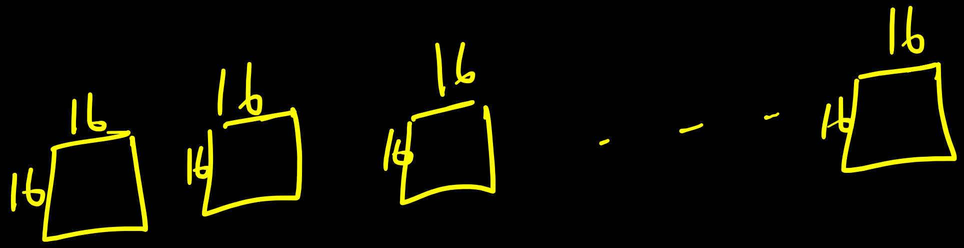
tokens

Key-idea

16x16 patches



$w_1 \quad w_2 \quad w_3 \; ---- \; w_n \rightarrow$ NLP



$\rightarrow$ CV

patch

$640 \times 640 \longrightarrow 40 \times 40$ patches (16×16 size)

Attention $\longrightarrow$ ${}^{40 \times 40}C_2 = 1.279$ Million

$\downarrow$

more manageable

APPLIED ROOTS

Big-picture

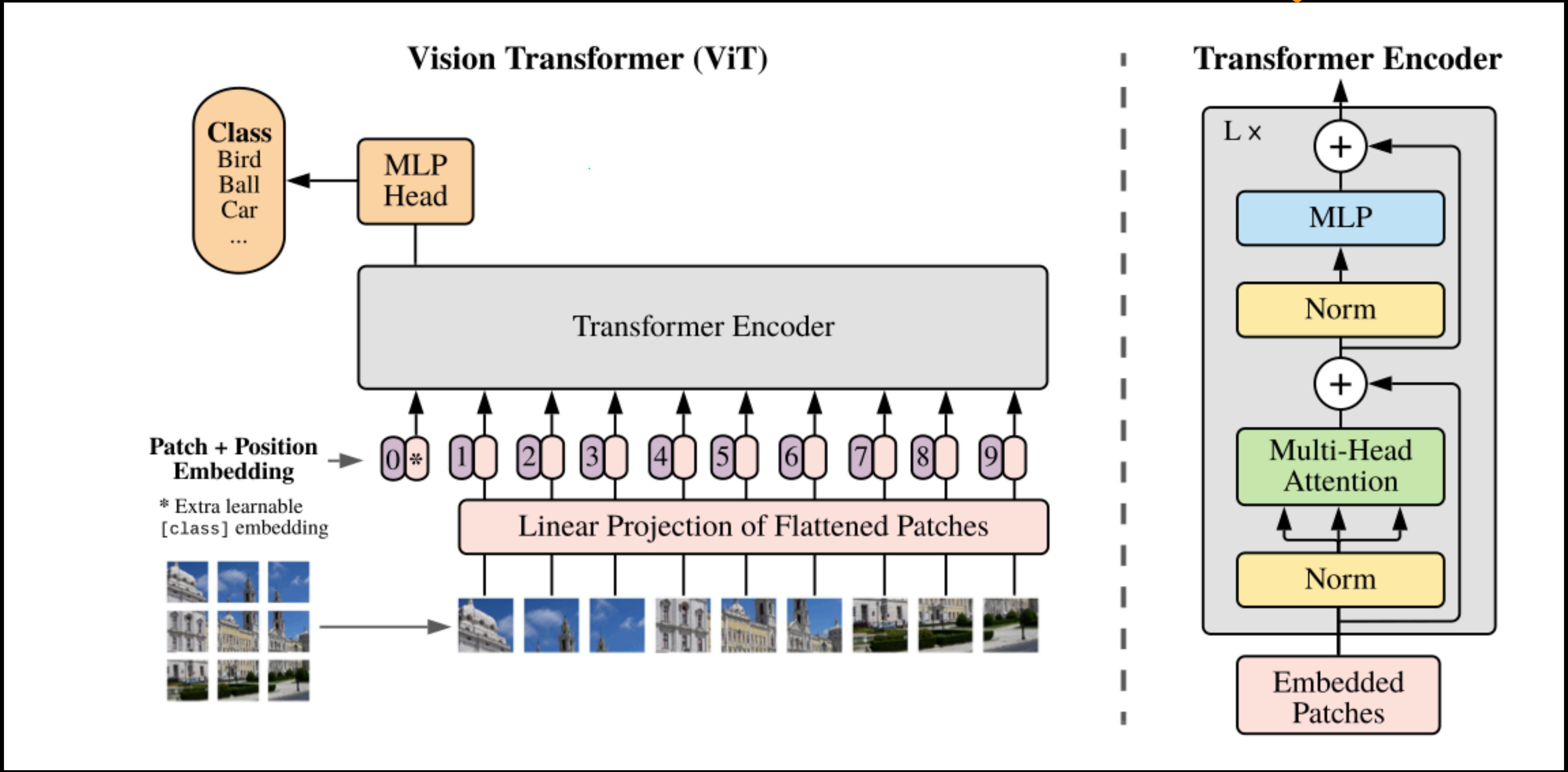
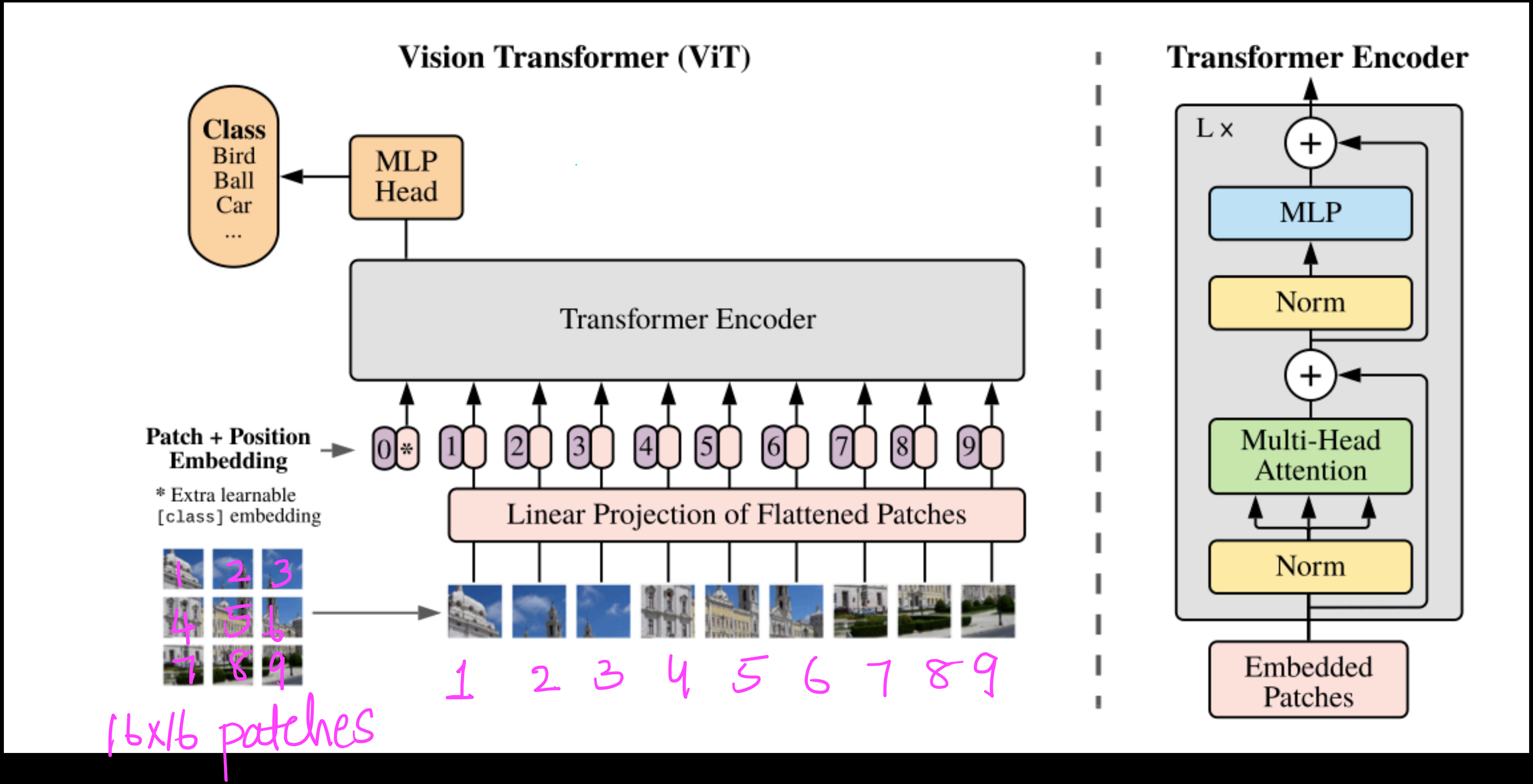
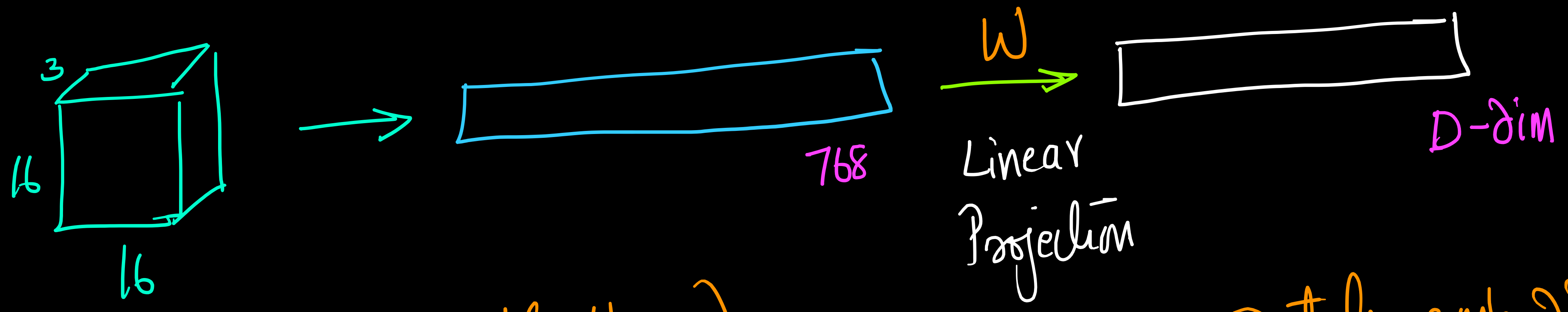


Image - Patches



Linear Projection of flattened patches



patch

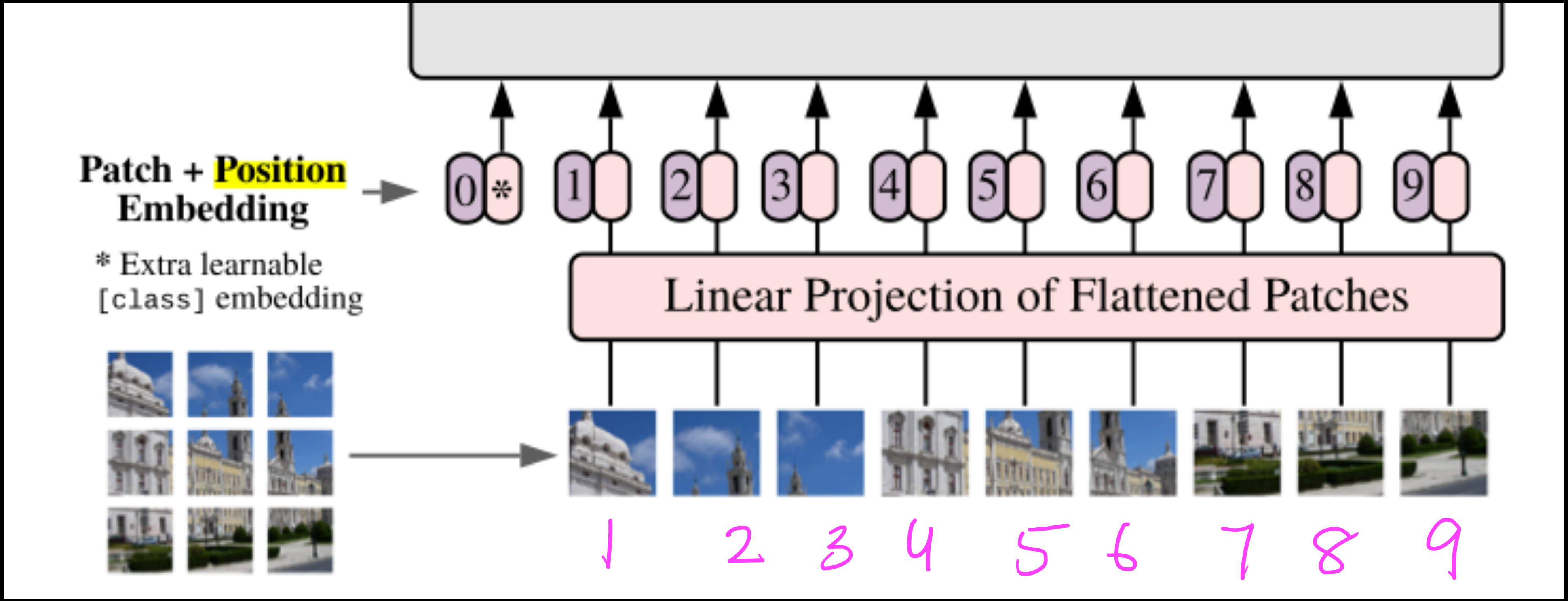
flattened patch

patch-embedding

(similar to word embedding)

APPLIED ROOTS

Positional Embedding (PE)



1-dim PE

Sin

Learnable

APPLIED ROOTS

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

1-D pos-Embed @ input is good enough

Learned PE captures patch positions

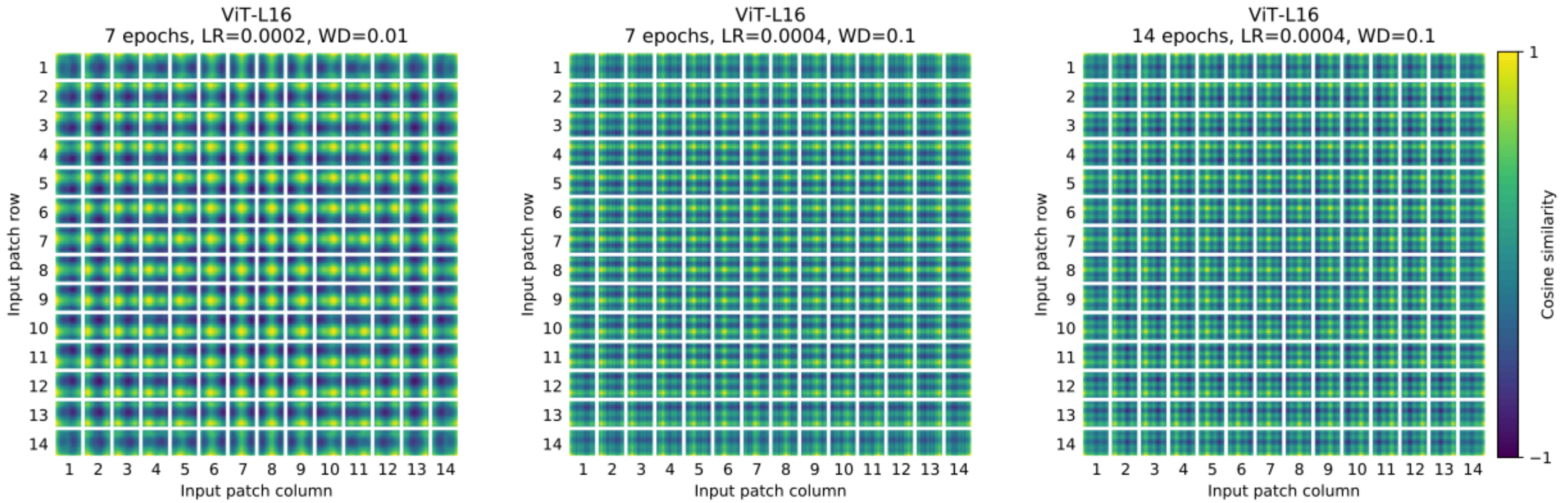
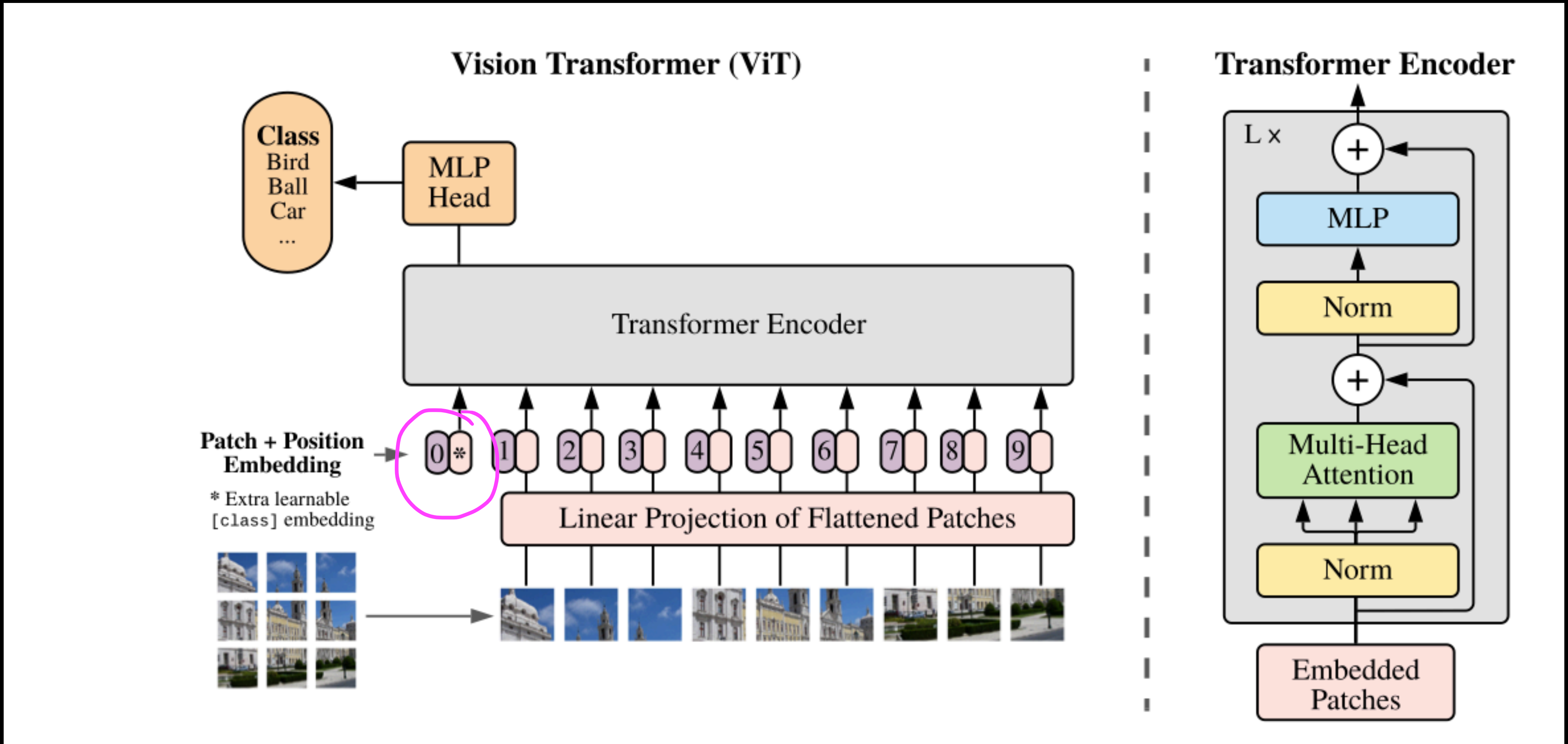


Figure 9: Position embeddings of models trained with different hyperparameters.

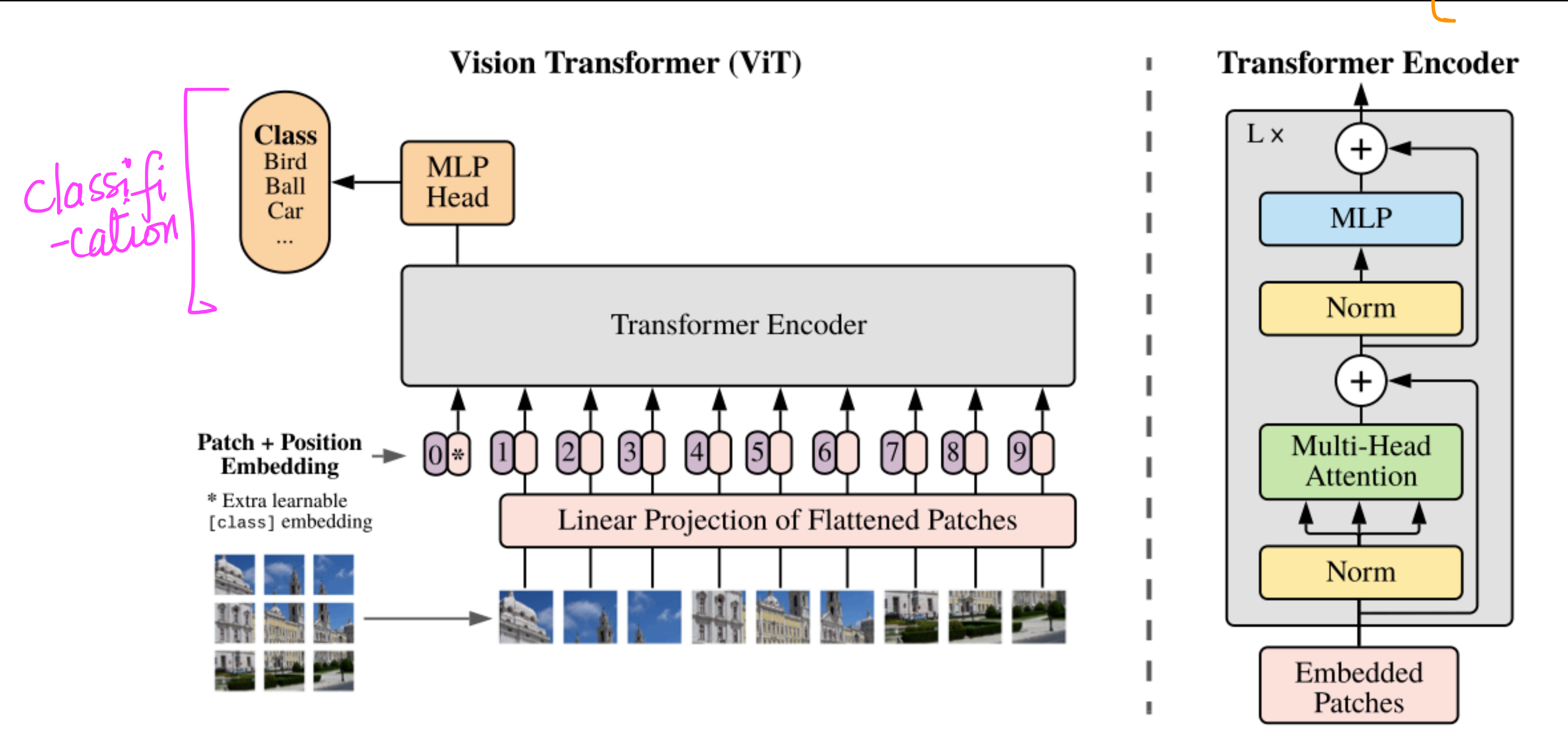
224x224 sized images as input

BERT [CLS] like

0[*]



Encoder-only Transformer
[BERT]



Variations

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.

APPLIED ROOTS

MLP's in Transformer Encoder

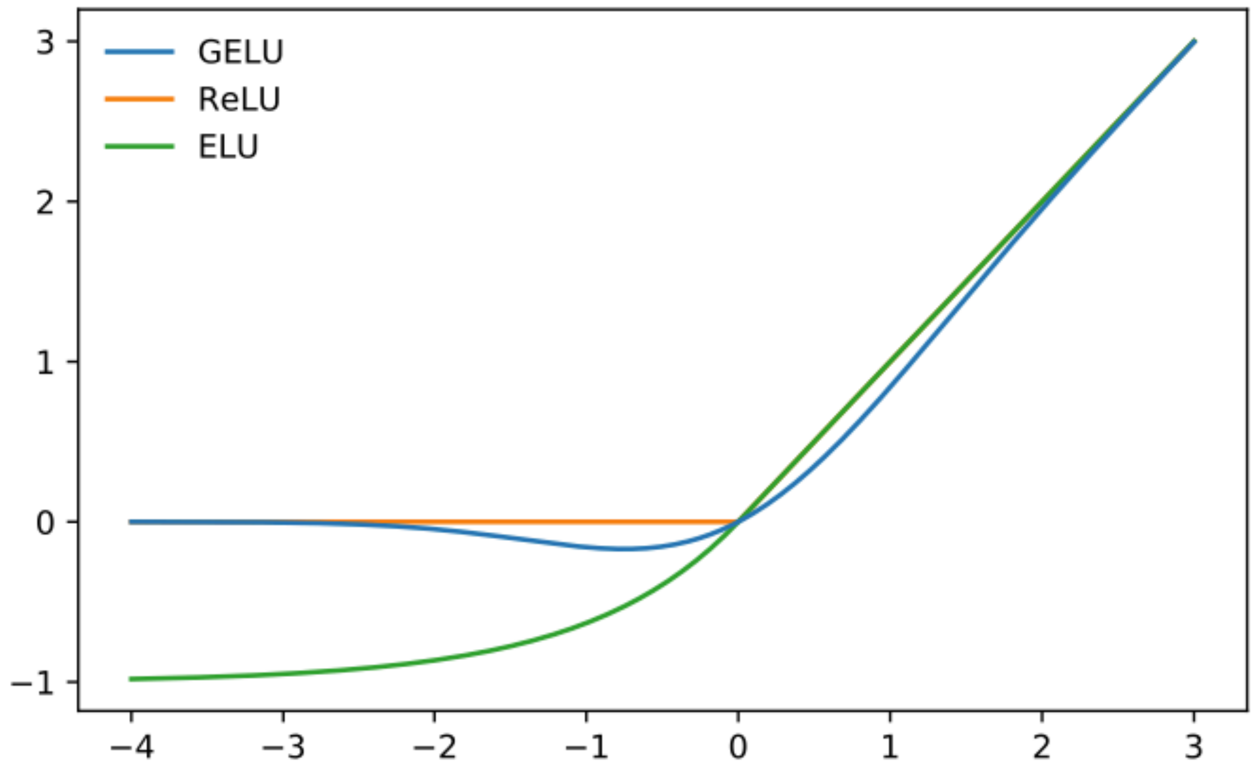


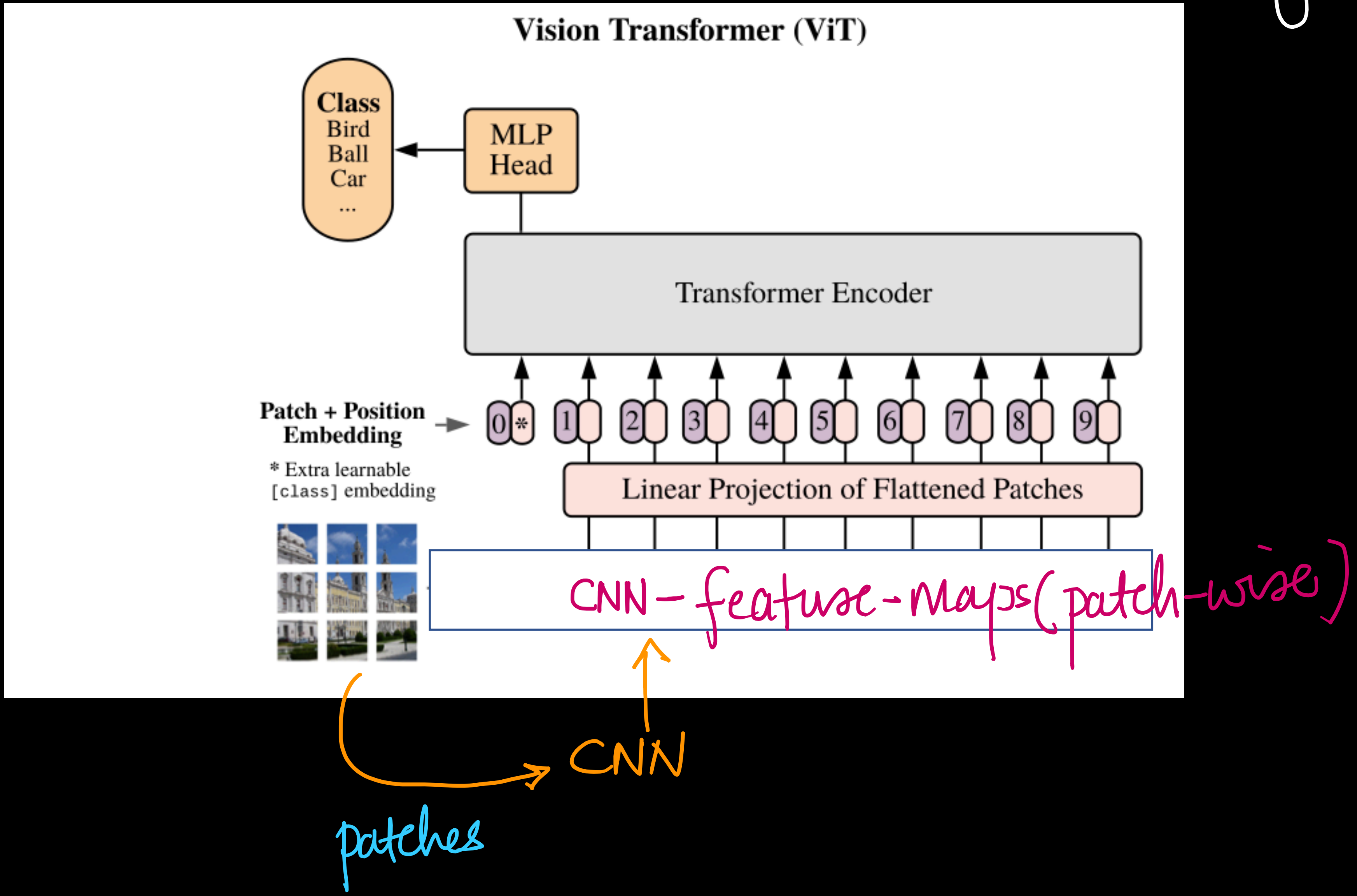
Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

https://arxiv.org/pdf/1606.08415.pdf

Smoother RELU

~ 0.5x(1 + tanh[SQRT(2/π)(x + 0.044715x^3)])

Hybrid Architecture



APPLIED ROOTS

Experimental Results

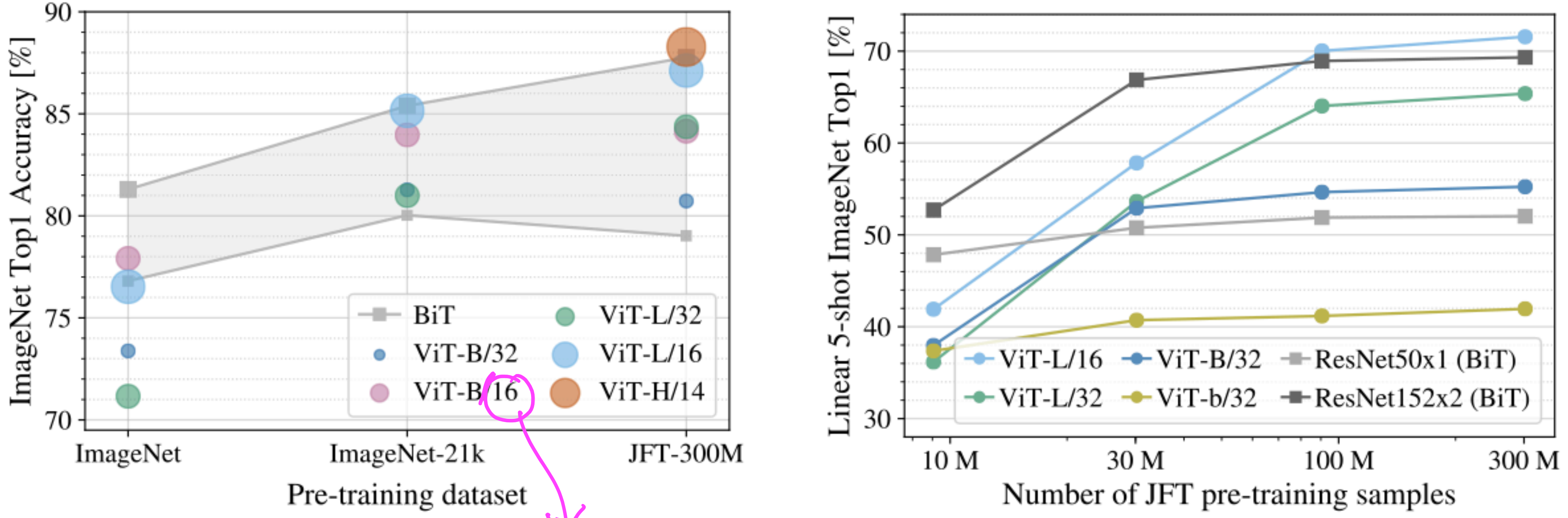
ViT on ImageNet → 77.9% top-1 accuracy

SOTA [ResNet-based CNN] → 85.8%

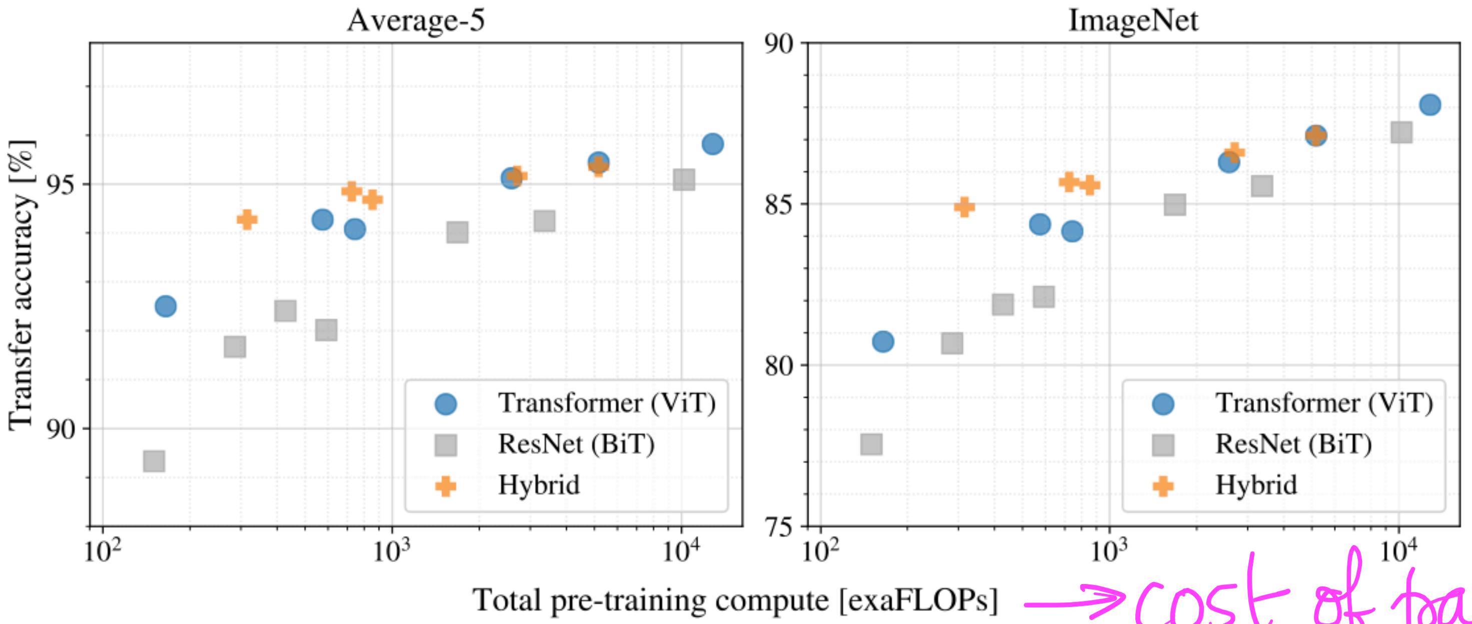
overfitting

	ImageNet	ImageNet-21K [14M, 21K classes]	JFT [300MM, 18K classes]
BiT [CNN-based]	✓	~	–
ViT	–	~	✓

APPLIED ROOTS



16x16 patches



→ cost of training

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

APPLIED ROOTS

Code

https://github.com/google-research/vision_transformer

Model : https://github.com/google-research/vision_transformer/blob/master/vit_jax/models.py

jax = autograd
+ XLA (Speedup ML)

using a pretrained Model:

https://colab.research.google.com/github/google-research/vision_transformer/blob/master/vit_jax.ipynb

APPLIED ROOTS

Conclusion

→ first steps towards non-conv models in CV

→ ~18-24 months for more models/variations

Q & A