Vision Transformer (ViT)

Arman Afrasiyabi
PHD Student at Université Laval

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

Introduction

First, they project the input image to an embedding, then they apply self-transformer.

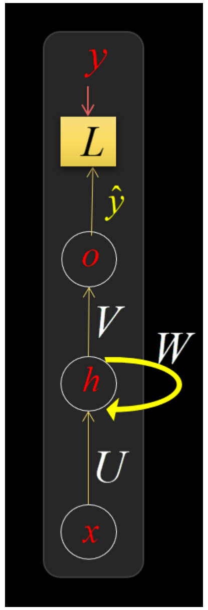
When trained on mid-sized datasets such as ImageNet, such models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.

Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”, ICLR-2020.

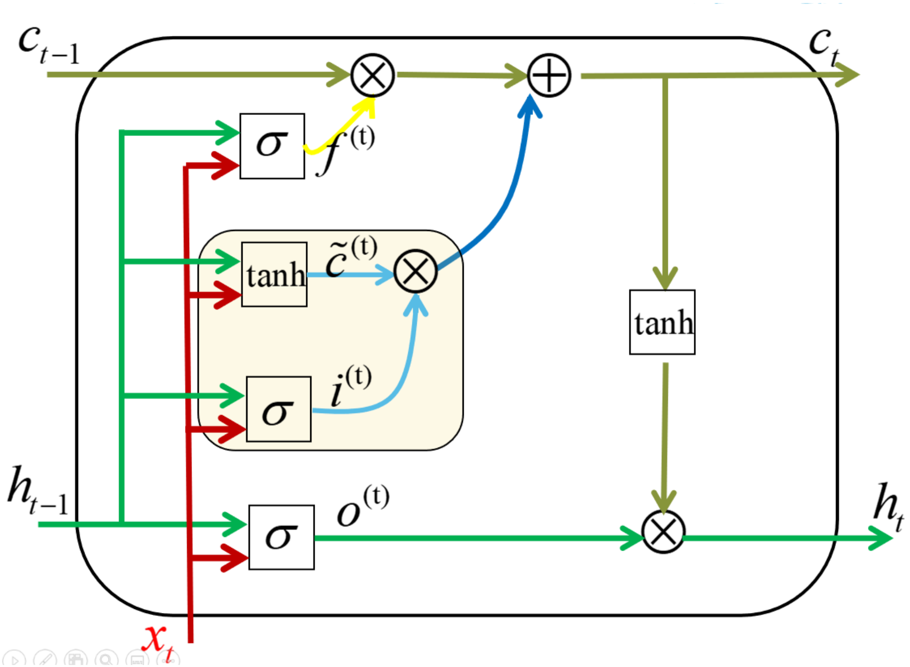
Why?

RNNs and LSTMs

RNN



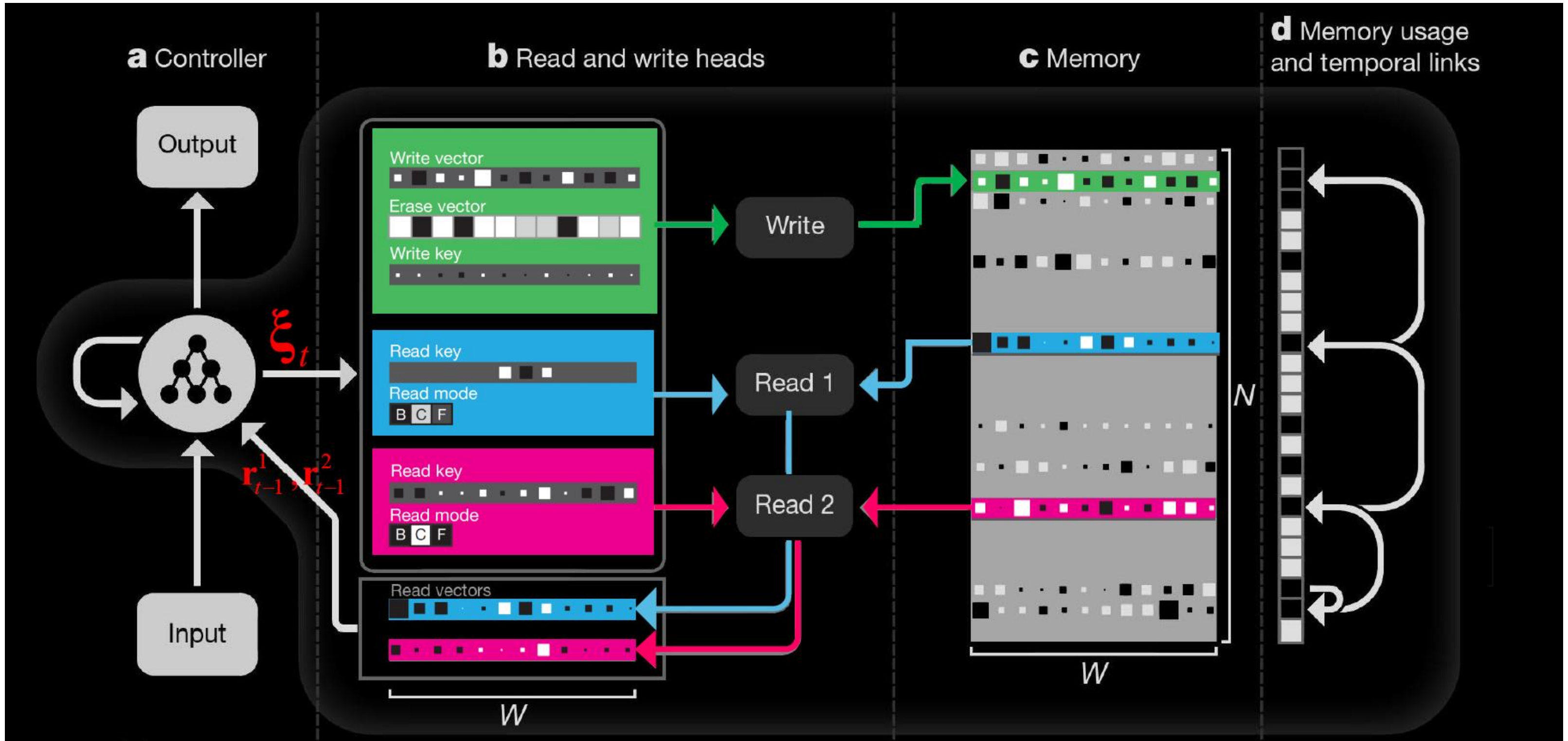
LSTM



See [this] for Arman's slides on RNN and LSTM.

Aim at moving forward with NTM

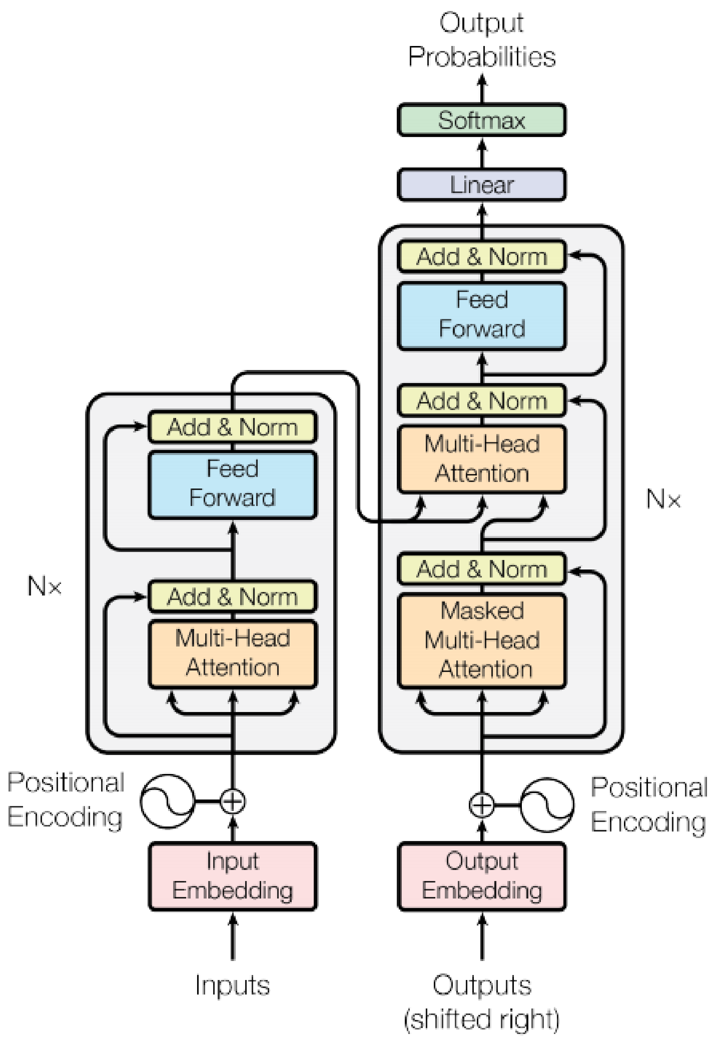
Differentiable Neural Machines (Neural Turing Machines)



$$\xi_t = \left[\mathbf{k}_t^{\mathbf{r},1}; \ldots; \mathbf{k}_t^{\mathbf{r},R}; \hat{\beta}_t^{\mathbf{r},1}; \ldots; \hat{\beta}_t^{\mathbf{r},R}; \mathbf{k}_t^{w}; \hat{\beta}_t^{w}; \hat{\mathbf{e}}_t; \mathbf{v}_t; \hat{\mathrm{f}}_t^1; \ldots; \hat{\mathrm{f}}_t^R; \hat{\mathrm{g}}_t^a; \hat{\mathrm{g}}_t^w; \hat{\pi}_t^1; \ldots; \hat{\pi}_t^R \right]$$

See this for Arman's slides on NTM.

Keeping to move forward with Transformers



Vaswani et al. “Attention is all you need”, NeurIPS-2017.

How?

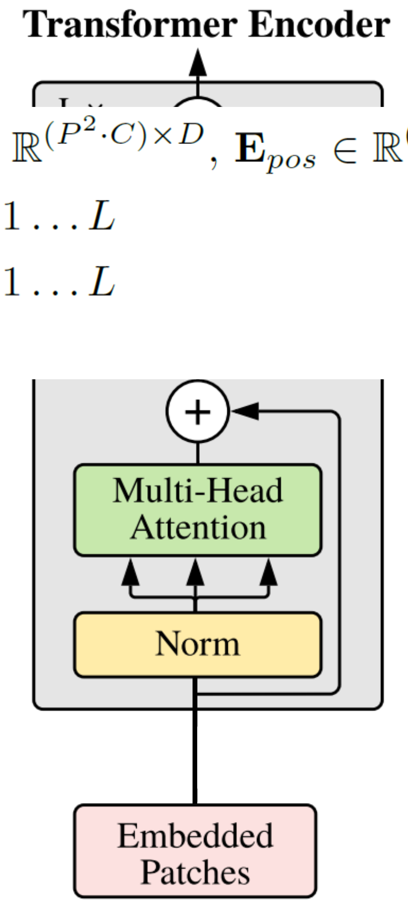
Vision Transformer (ViT)

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

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

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

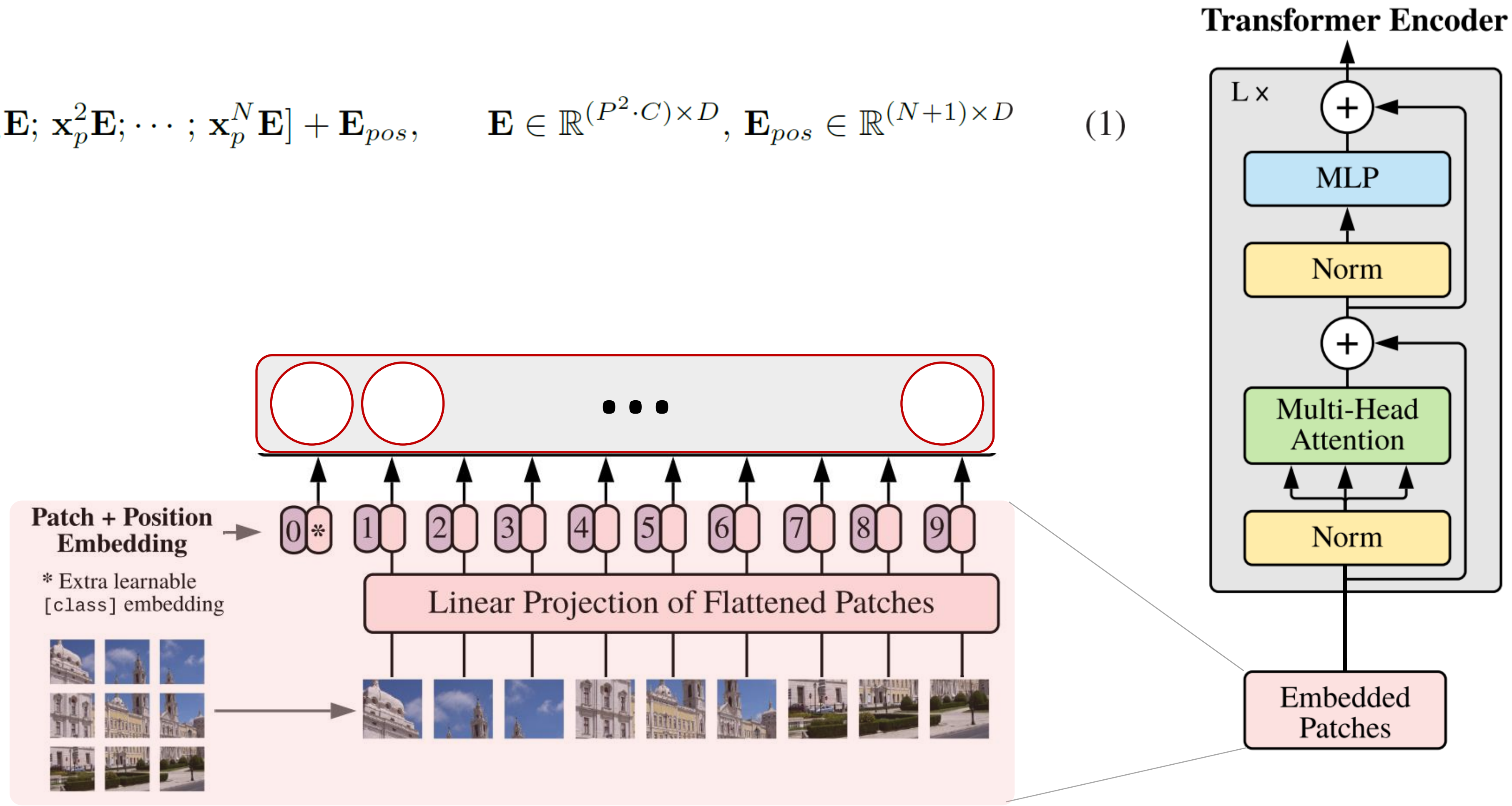
$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \tag{4}$$



Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”, ICLR-2020.

Vision Transformer (ViT)

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

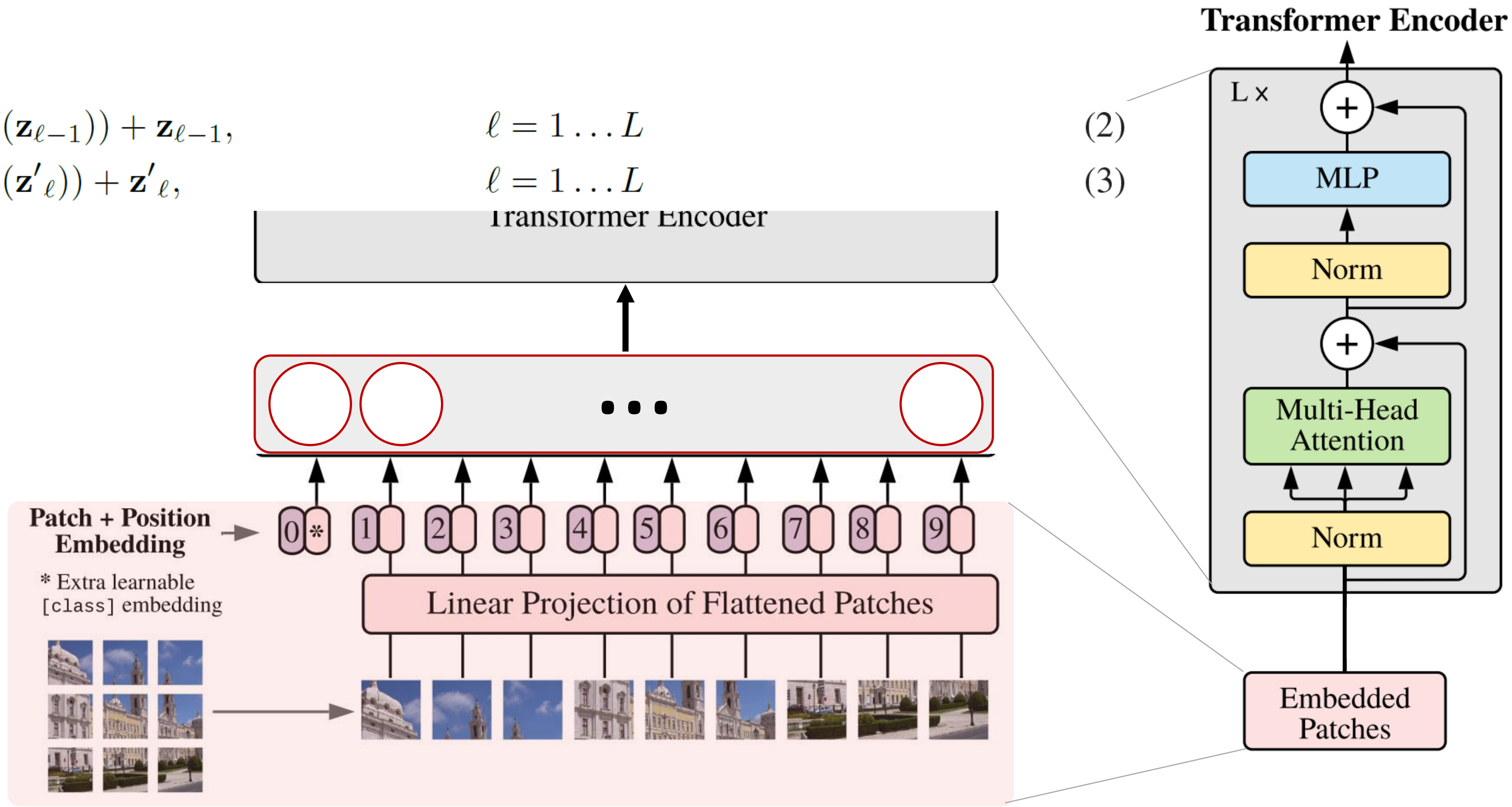


Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR-2020.

Vision Transformer (ViT)

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

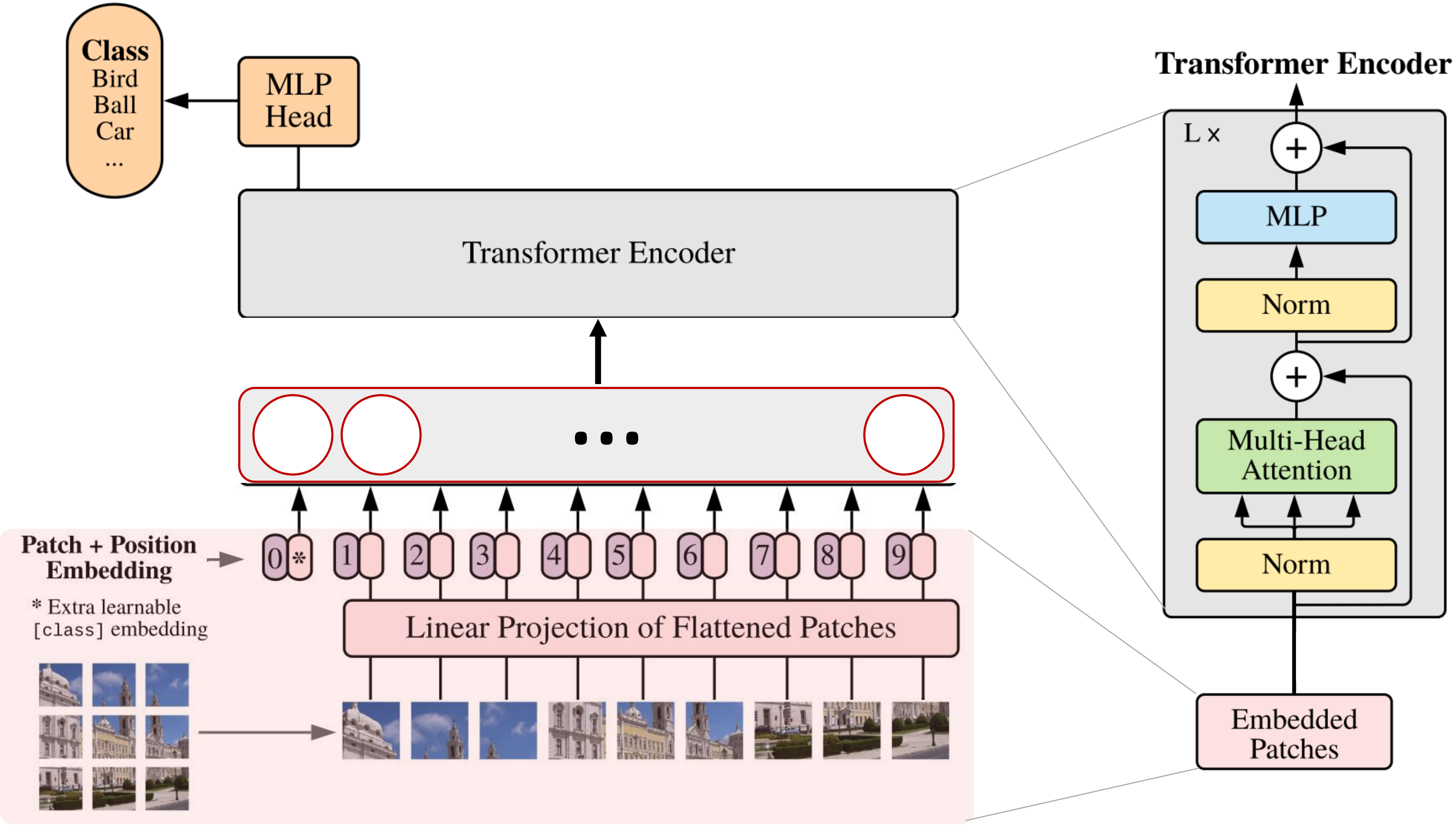
$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L \qquad (3)$$



Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”, ICLR-2020.

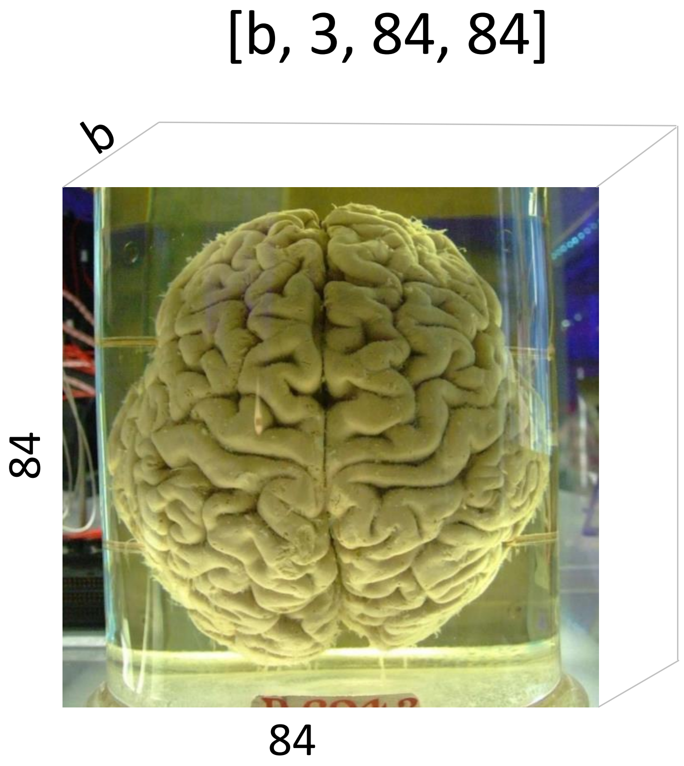
Vision Transformer (ViT)

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



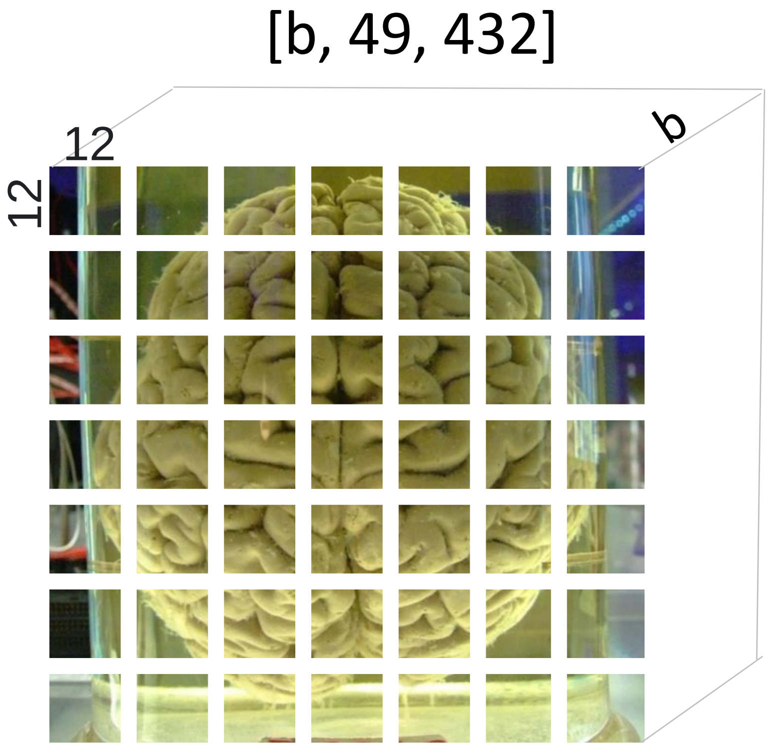
Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR-2020.

Input image format



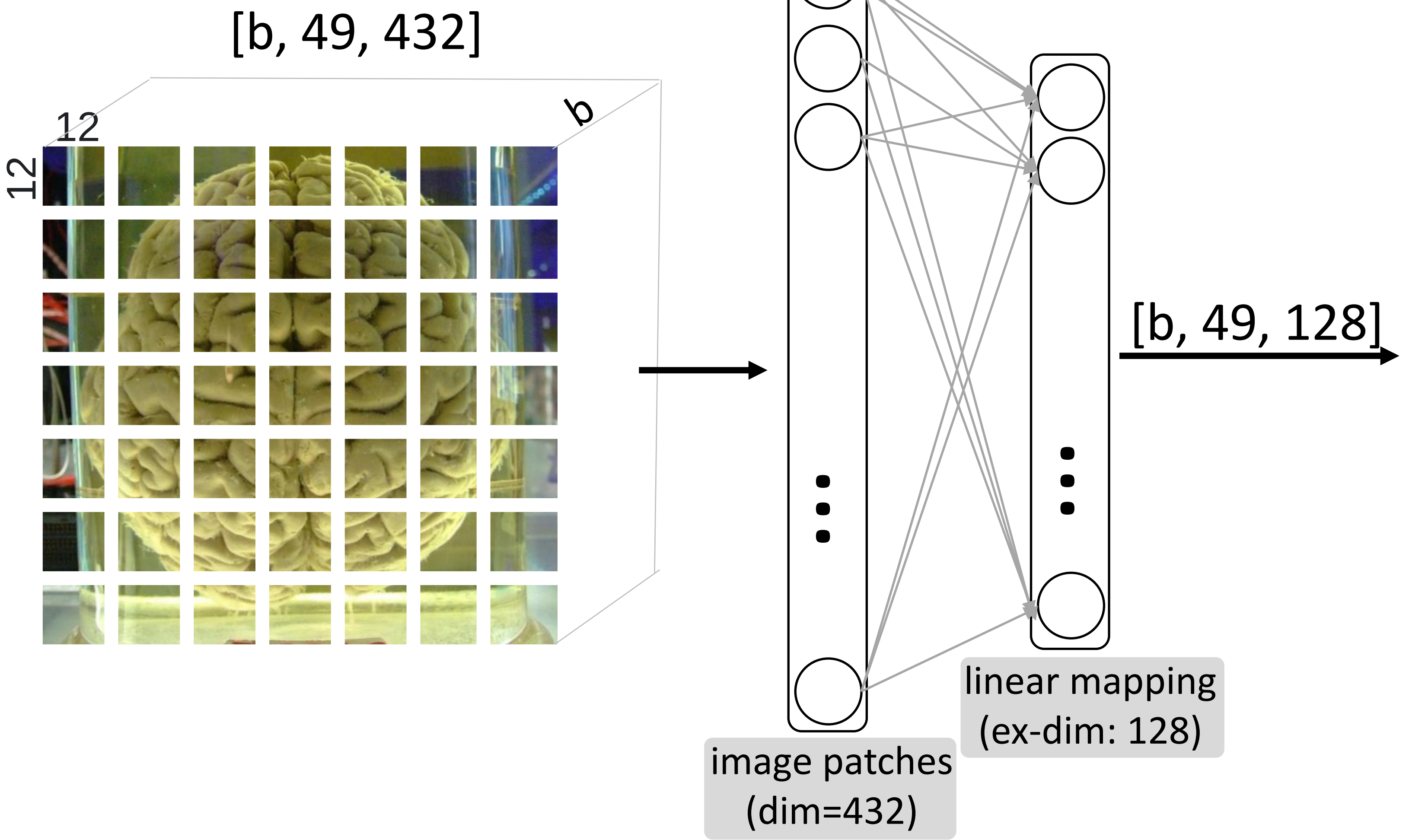
$$\mathbf{x} \in \mathbb{R}^{H \times W \times C}$$

Image to Patches



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

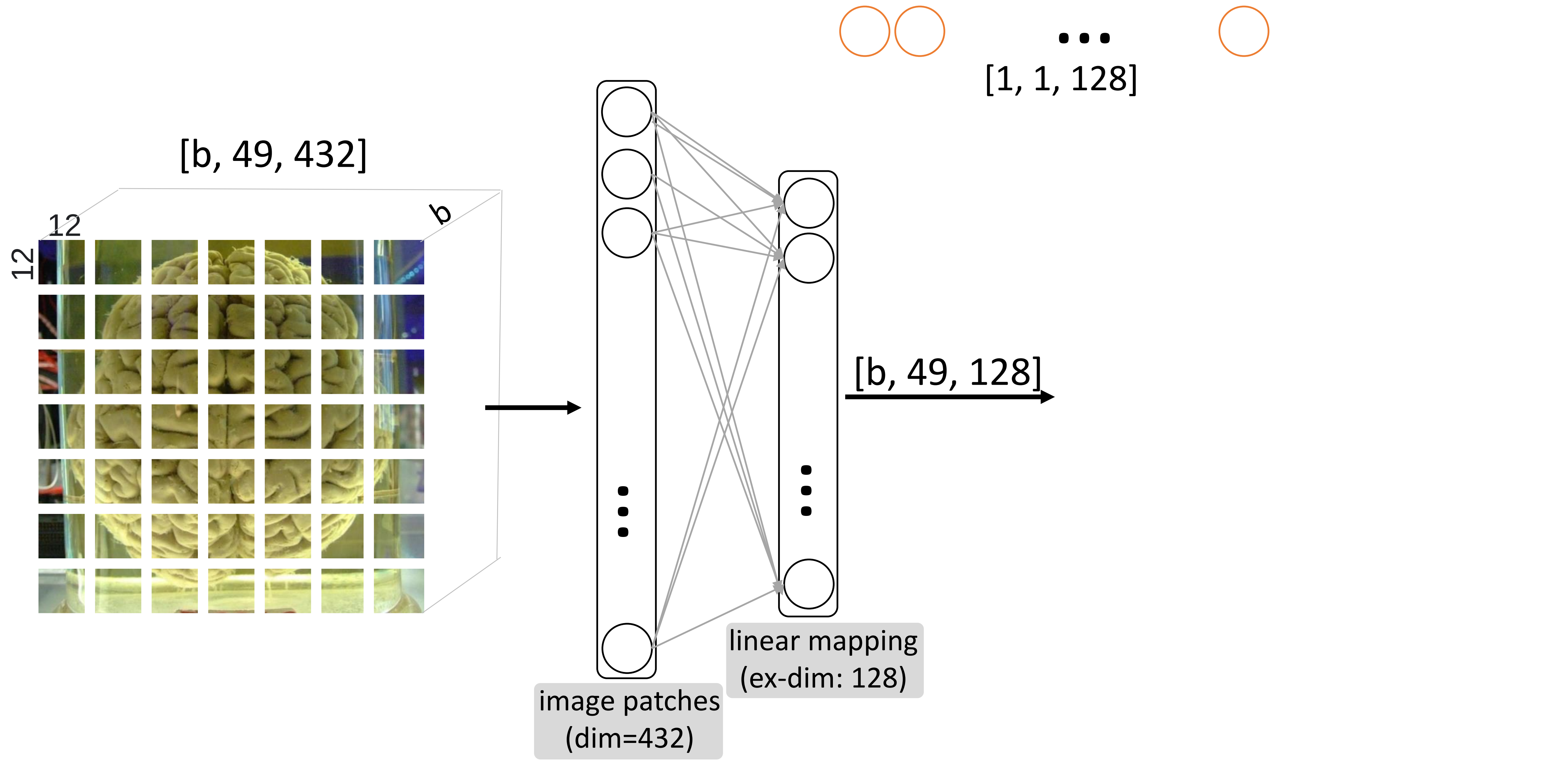
Patch to Token



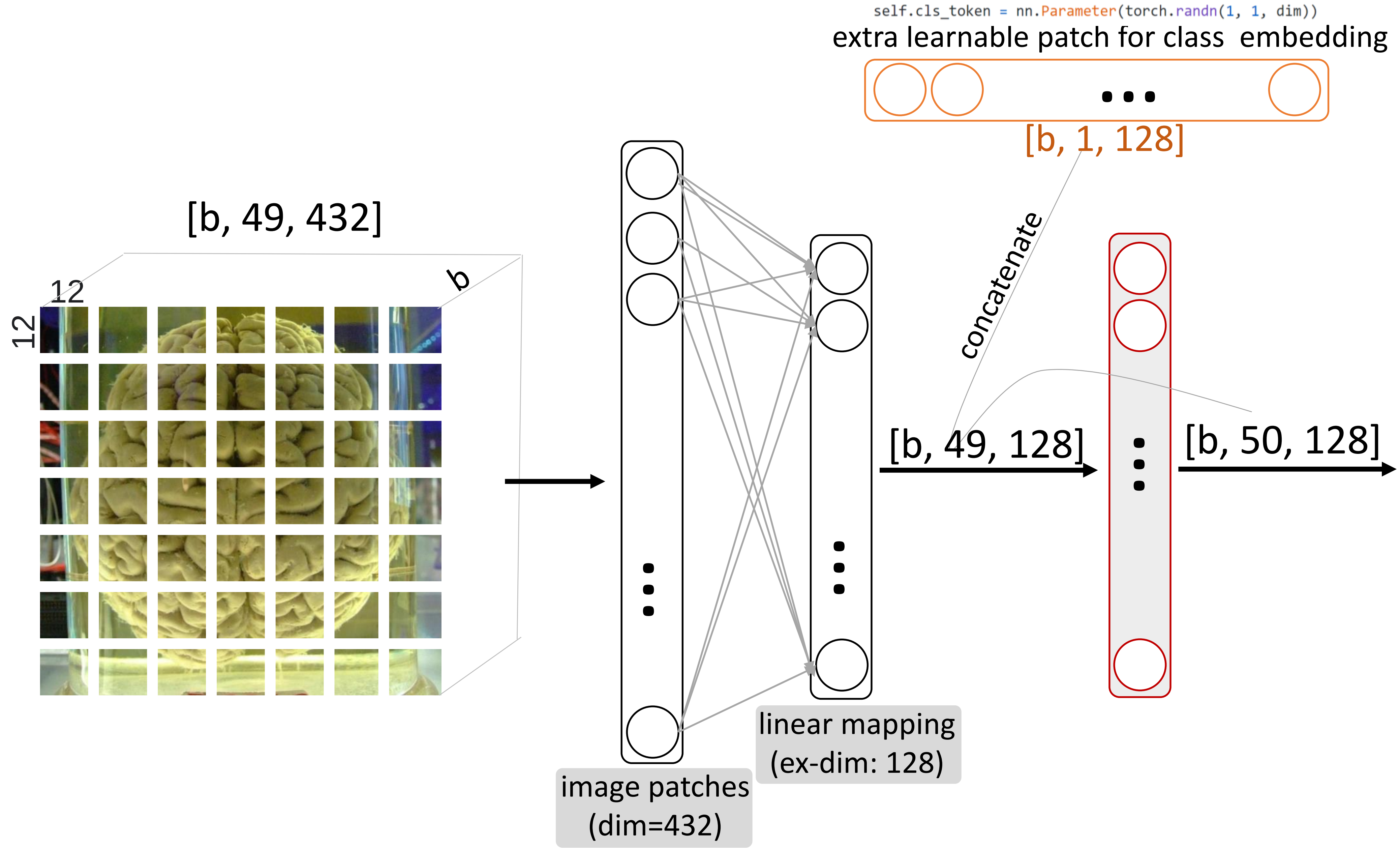
Concat. of the Learnable Class Token

```
self.cls_token = nn.Parameter(torch.randn(1, 1, dim))
```

extra learnable patch for class embedding



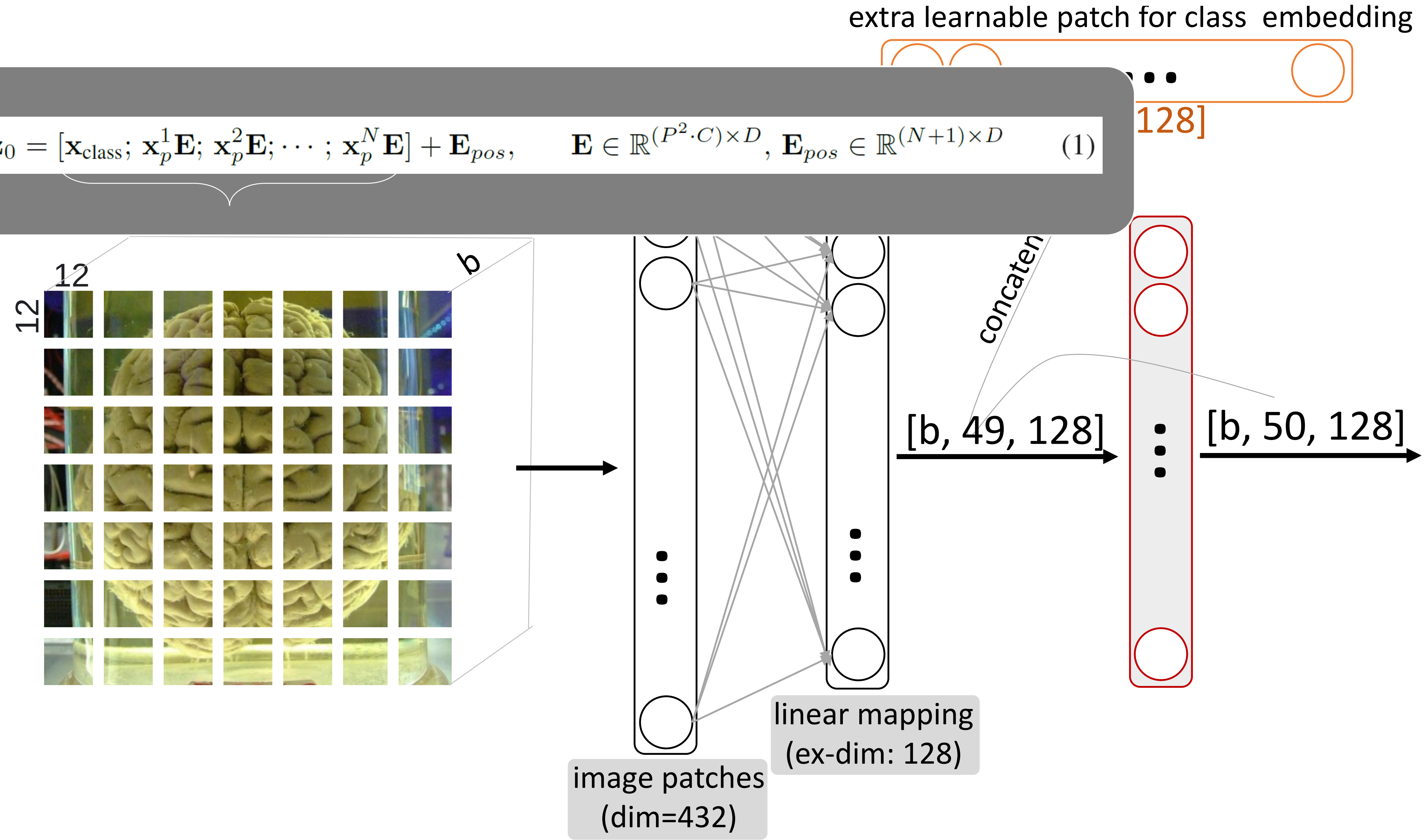
Concat. learnable Class Token



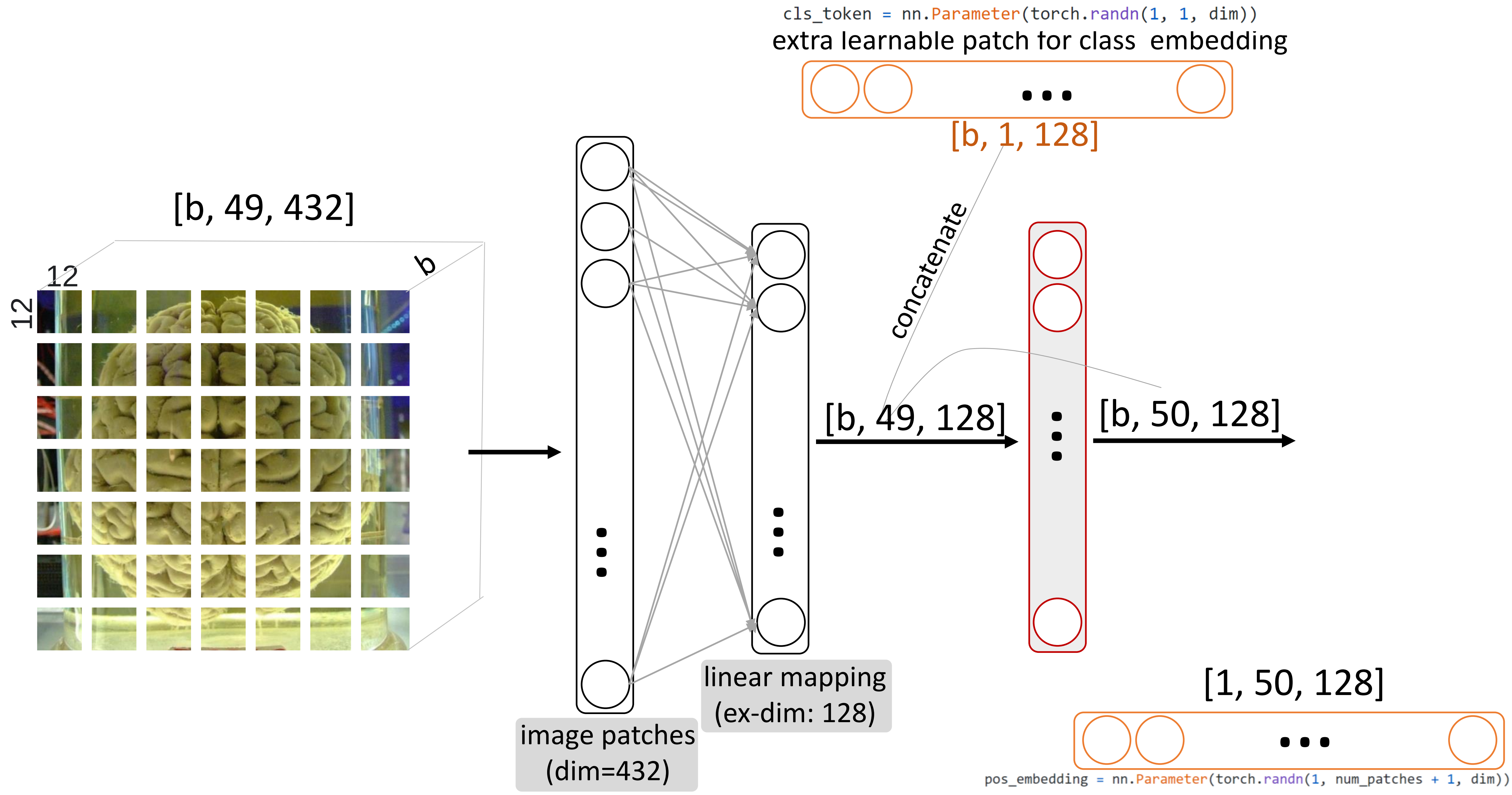
Concat. learnable Class Token

```
self.cls_token = nn.Parameter(torch.randn(1, 1, dim))
```

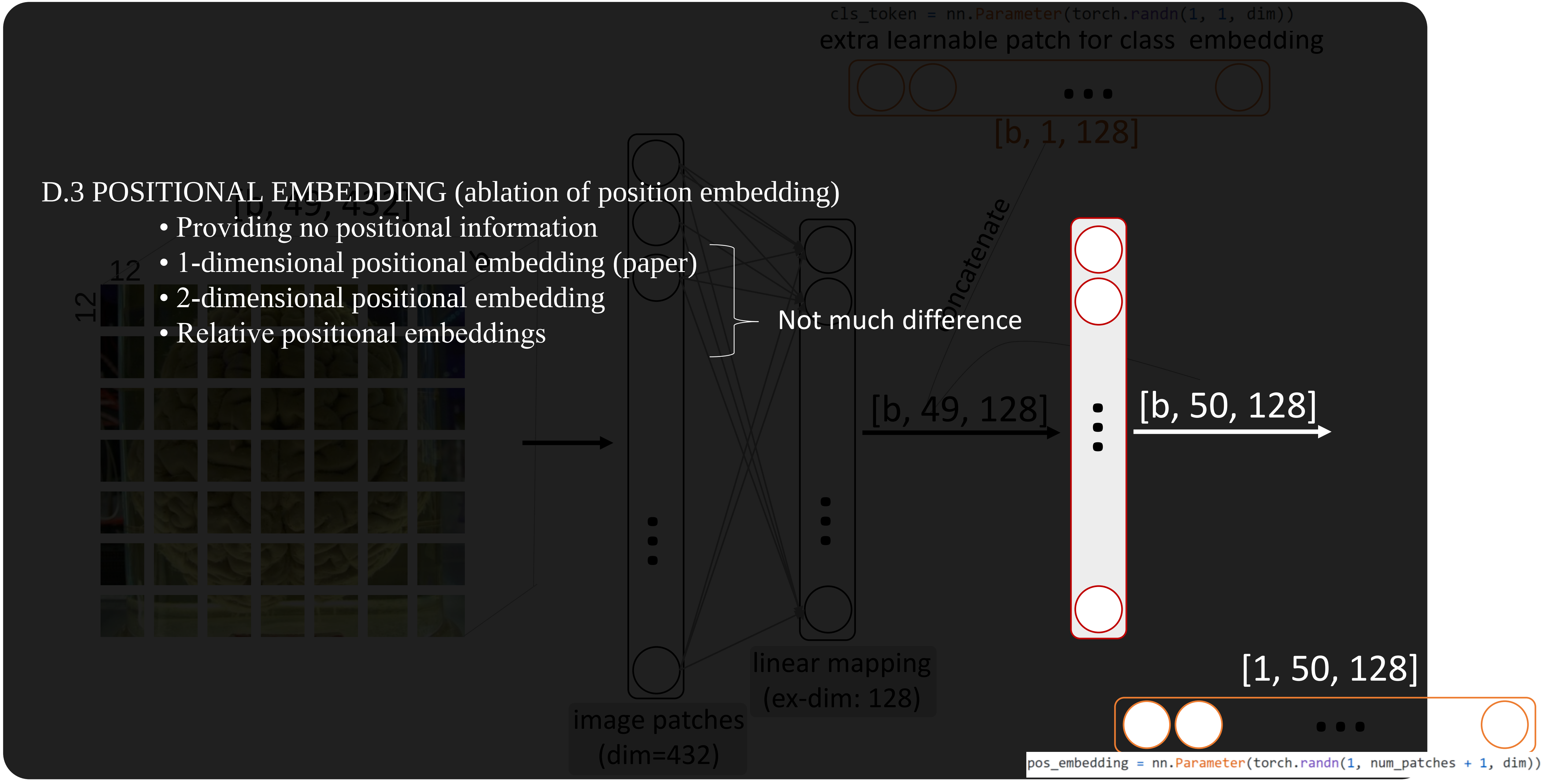
$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C)\times D},\, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D} \qquad (1)$$



Adding position embeddings



Adding position embeddings



D.3 POSITIONAL EMBEDDING (ablation of position embedding)

- Providing no positional information
- 1-dimensional positional embedding (paper)
- 2-dimensional positional embedding
- Relative positional embeddings

Not much difference

Position embeddings depends on hyperparameters

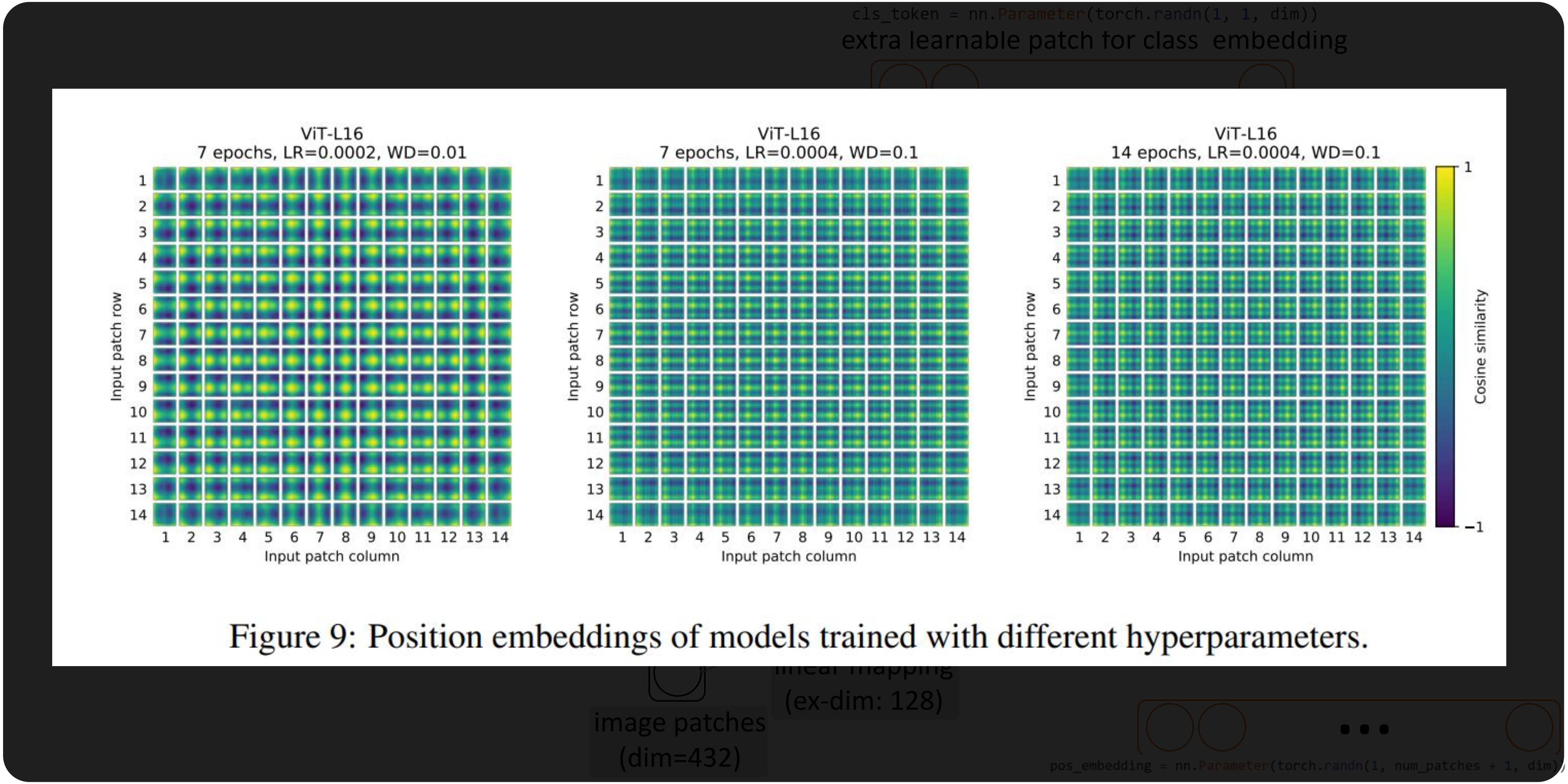
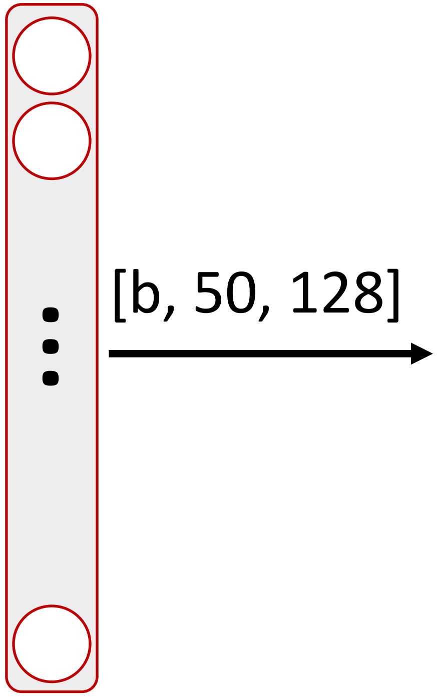
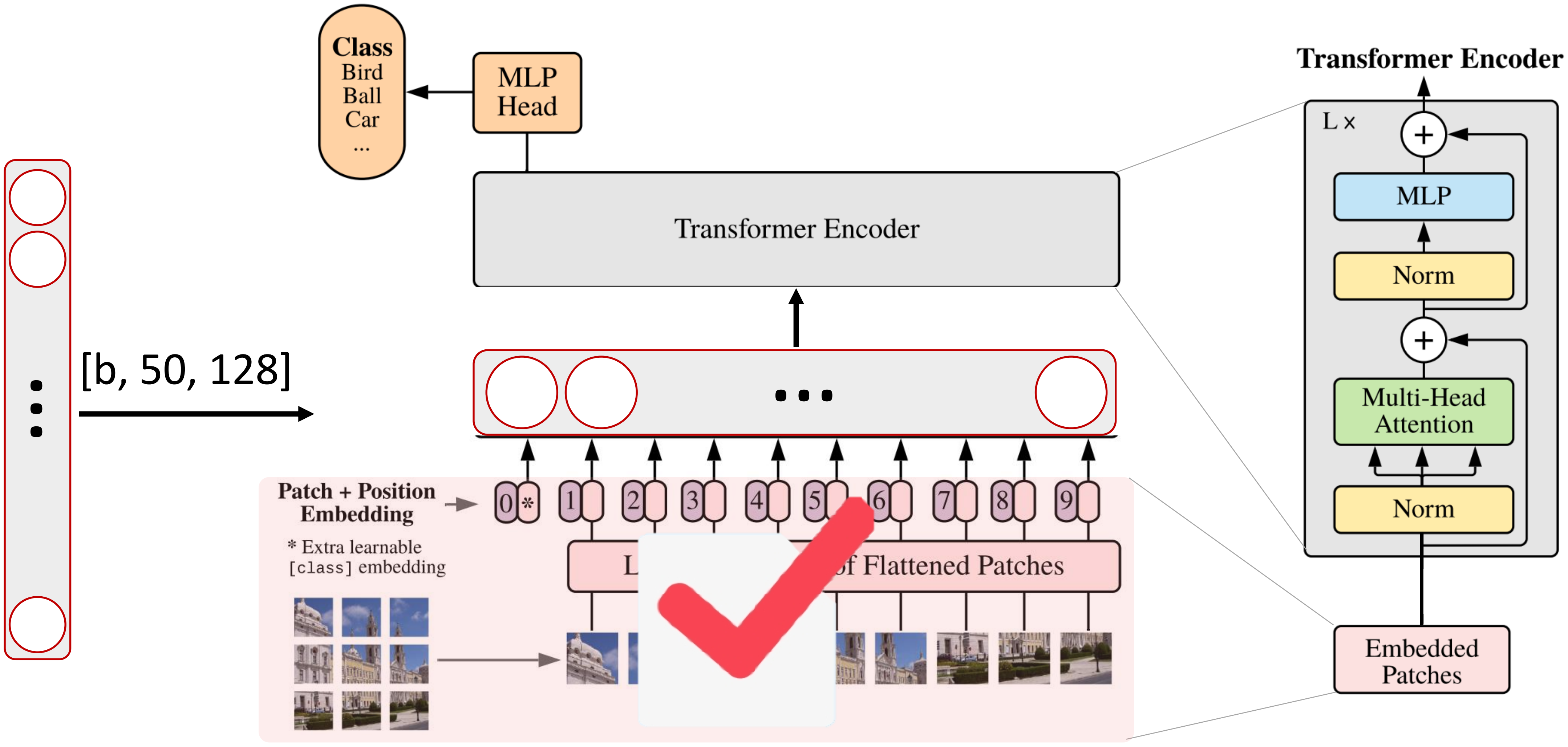


Figure 9: Position embeddings of models trained with different hyperparameters.

Patch embedding



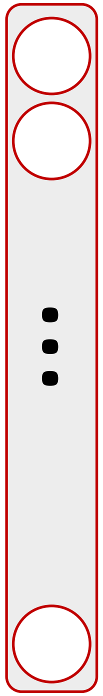
Big picture



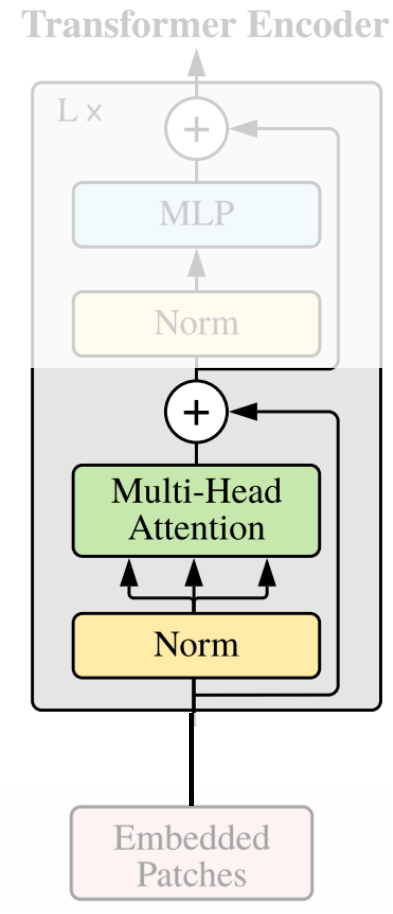
[b, 50, 128]
Class
Bird
Ball
Car
...
MLP
Head
Transformer Encoder
...
Patch + Position
Embedding
* Extra learnable
[class] embedding
0
*
1
2
3
4
5
6
7
8
9
L
of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head
Attention
Norm
Embedded
Patches

Transformer: MH-Self Attention (MSA)

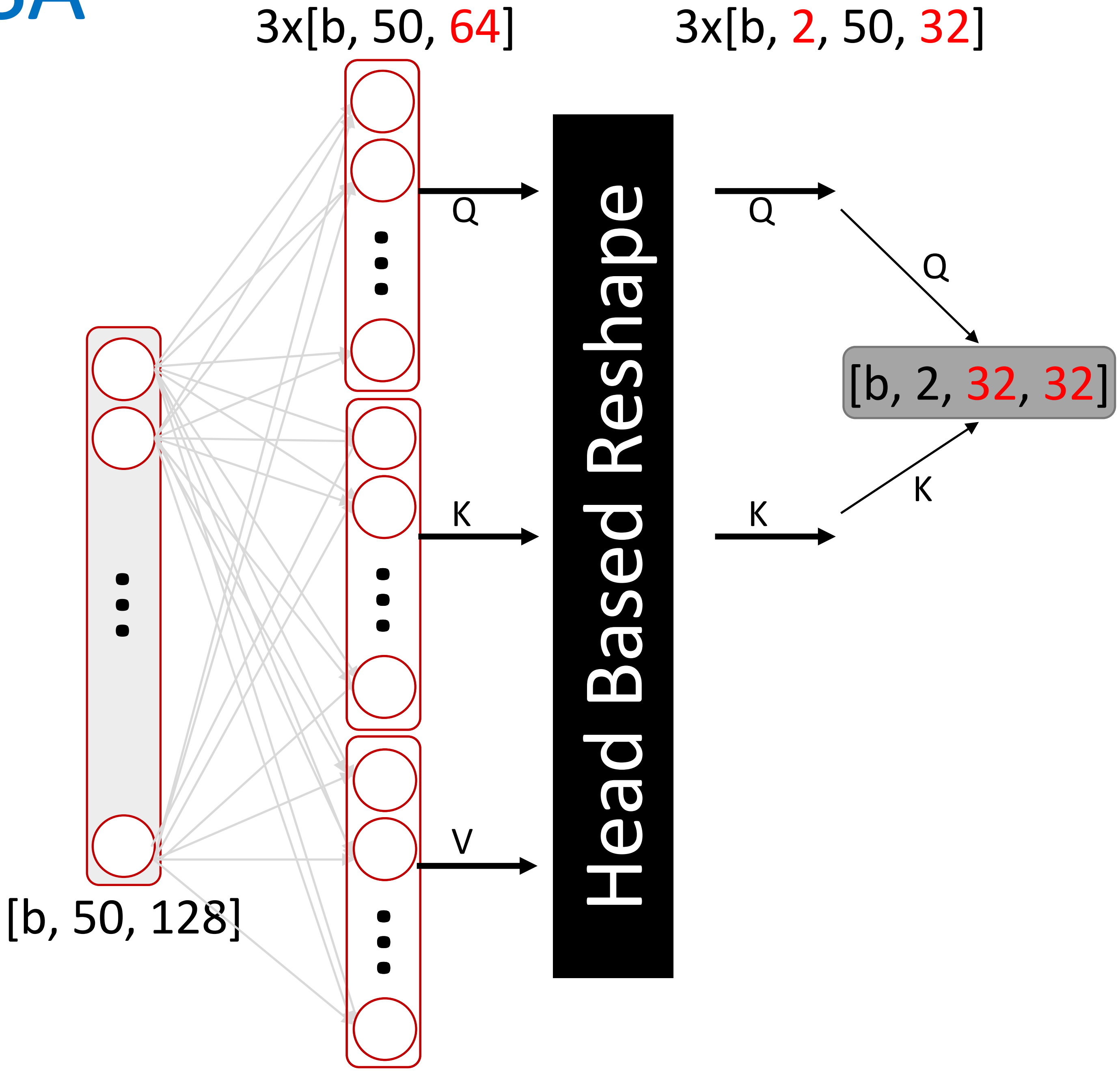
$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L$$



[b, 50, 128]

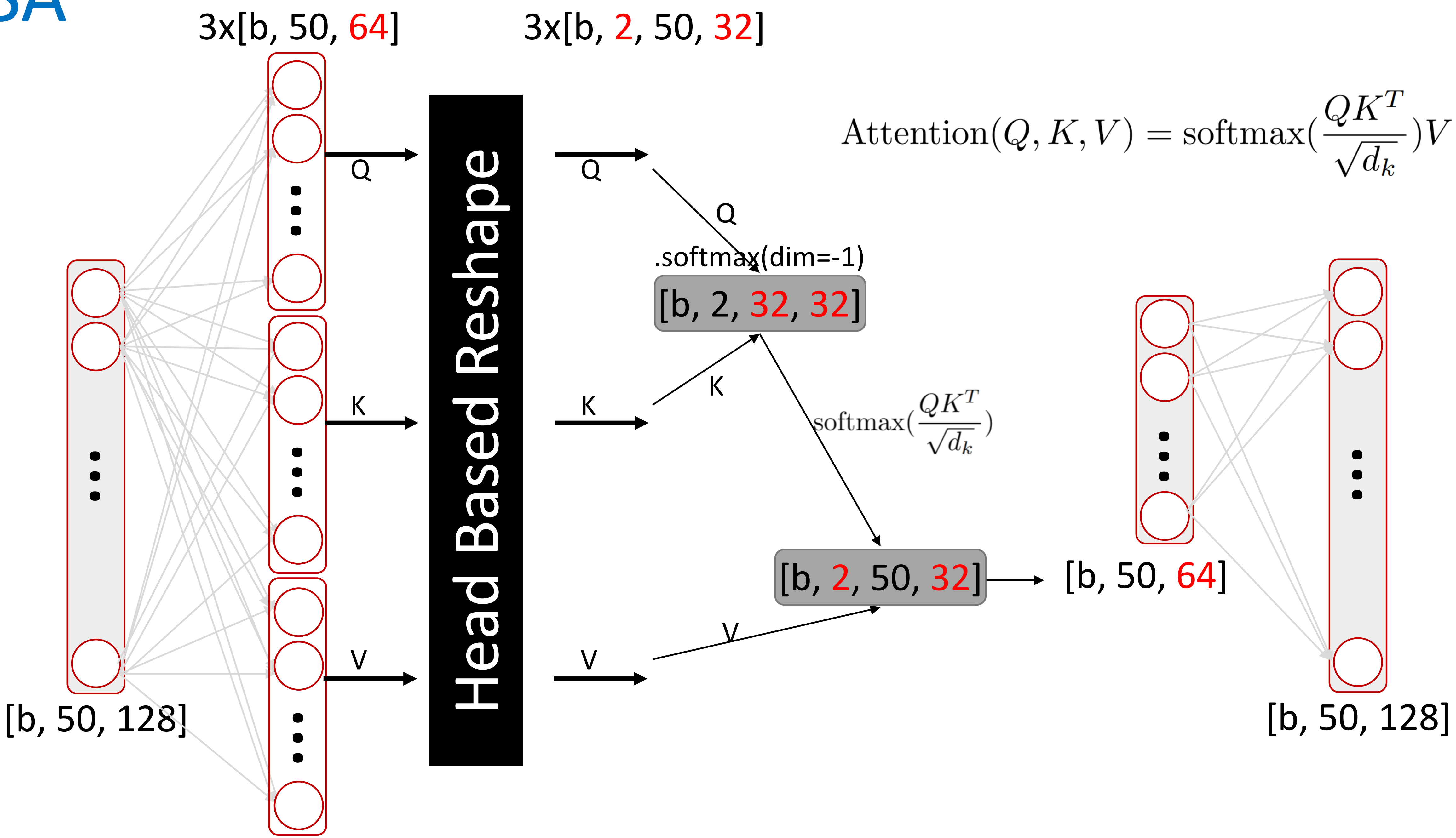


MSA

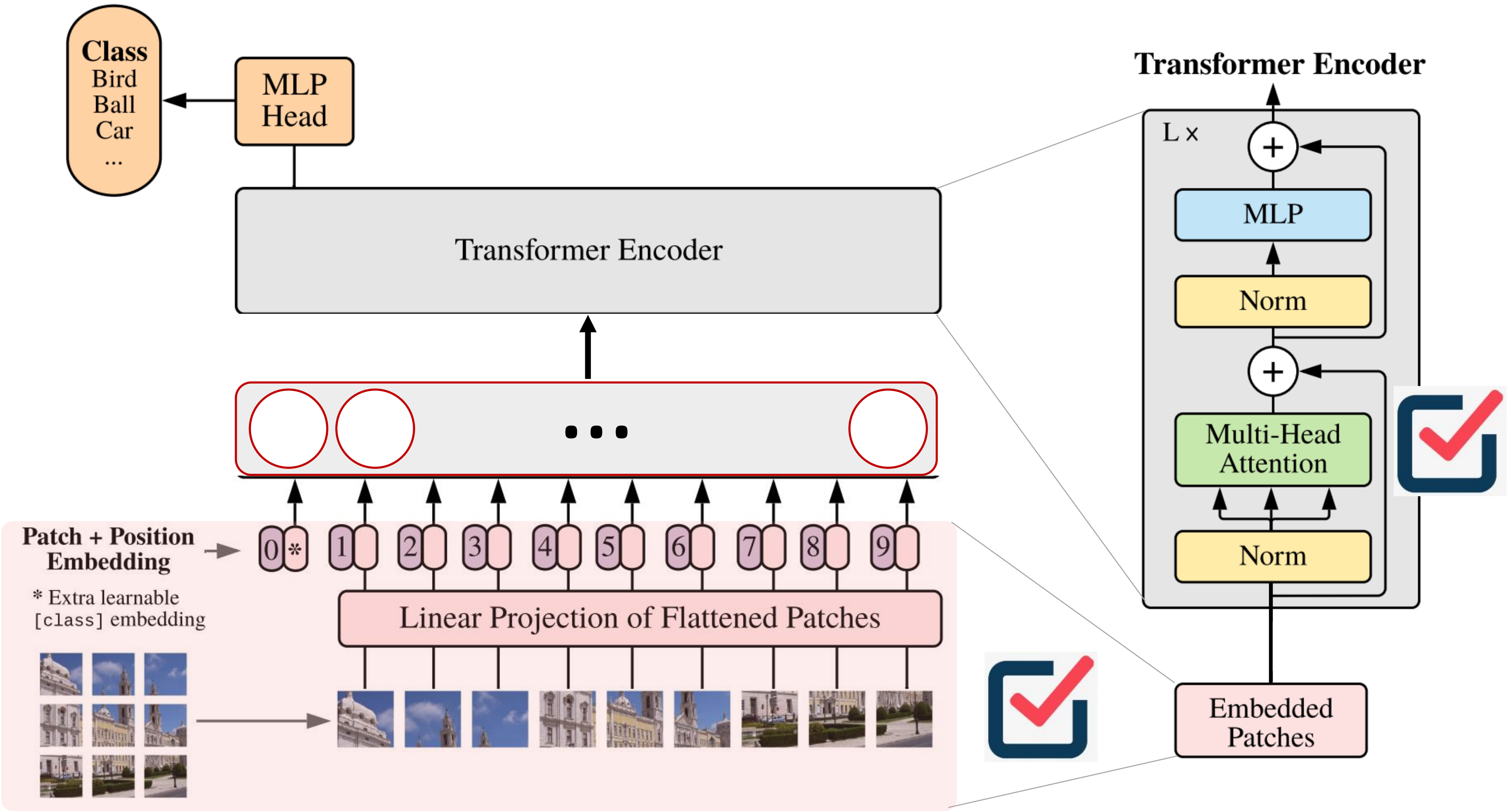


$$\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})$$

MSA

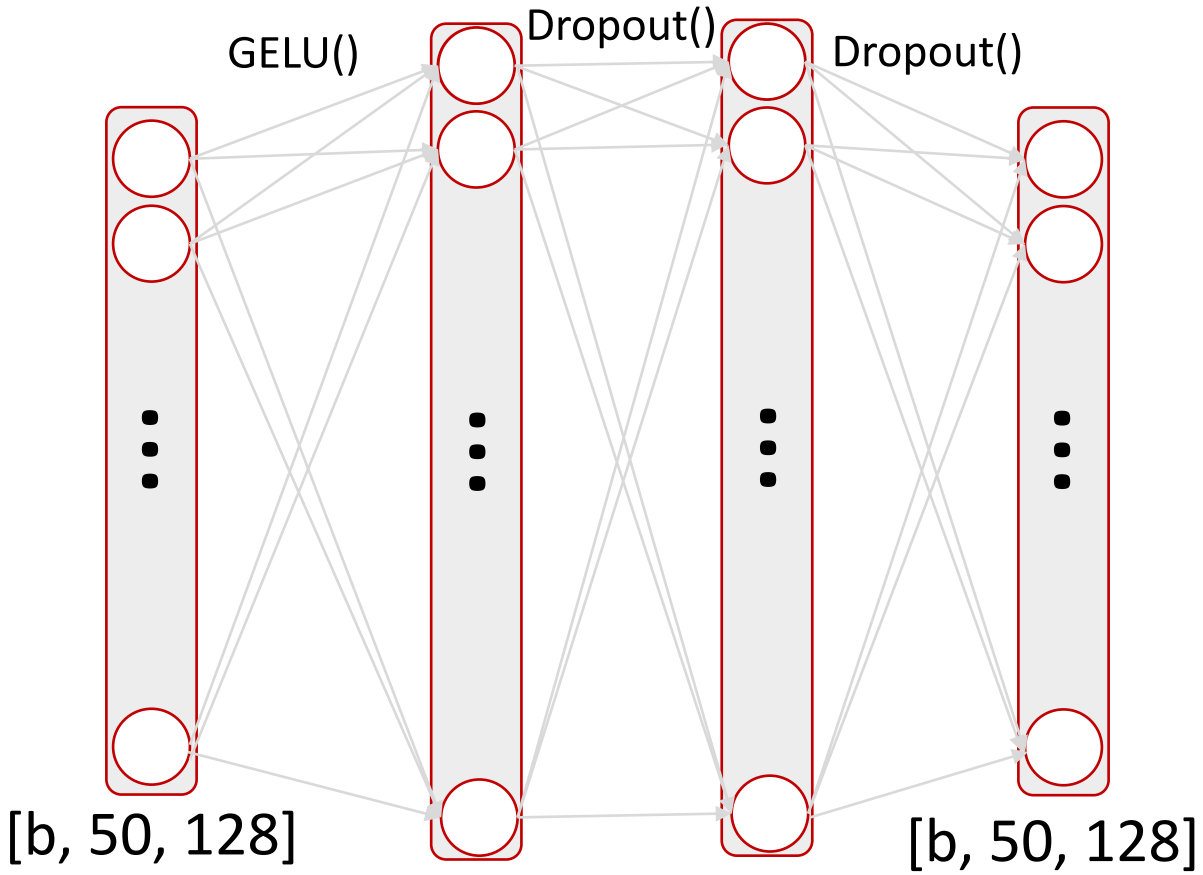

3x[b, 50, 64]
3x[b, 2, 50, 32]
Head Based Reshape
Q
K
V
$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
.softmax(dim=-1)
[b, 2, 32, 32]
$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})$
[b, 2, 50, 32]
[b, 50, 64]
[b, 50, 128]
[b, 50, 128]

Big picture

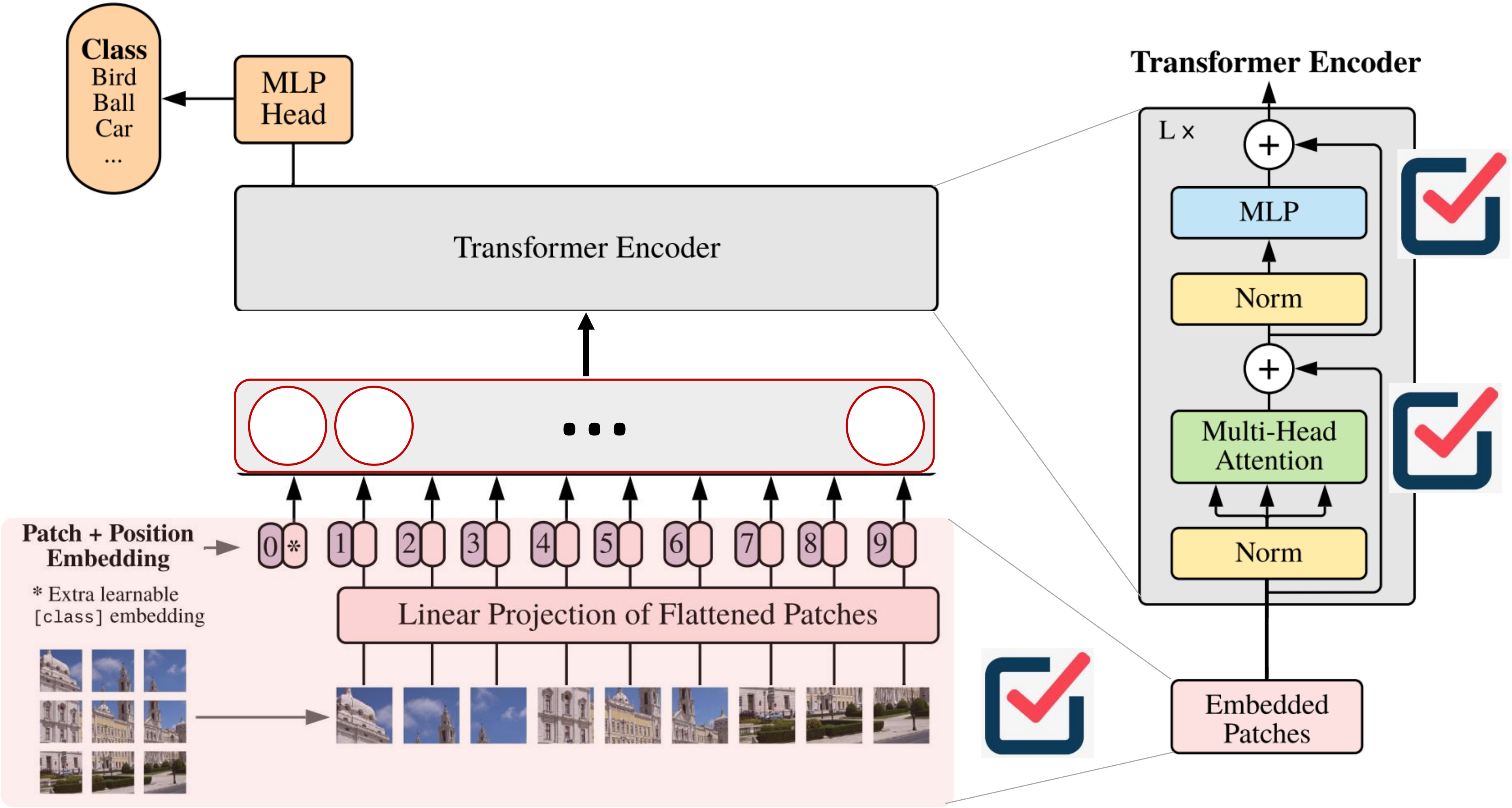


Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Transformer: MLP

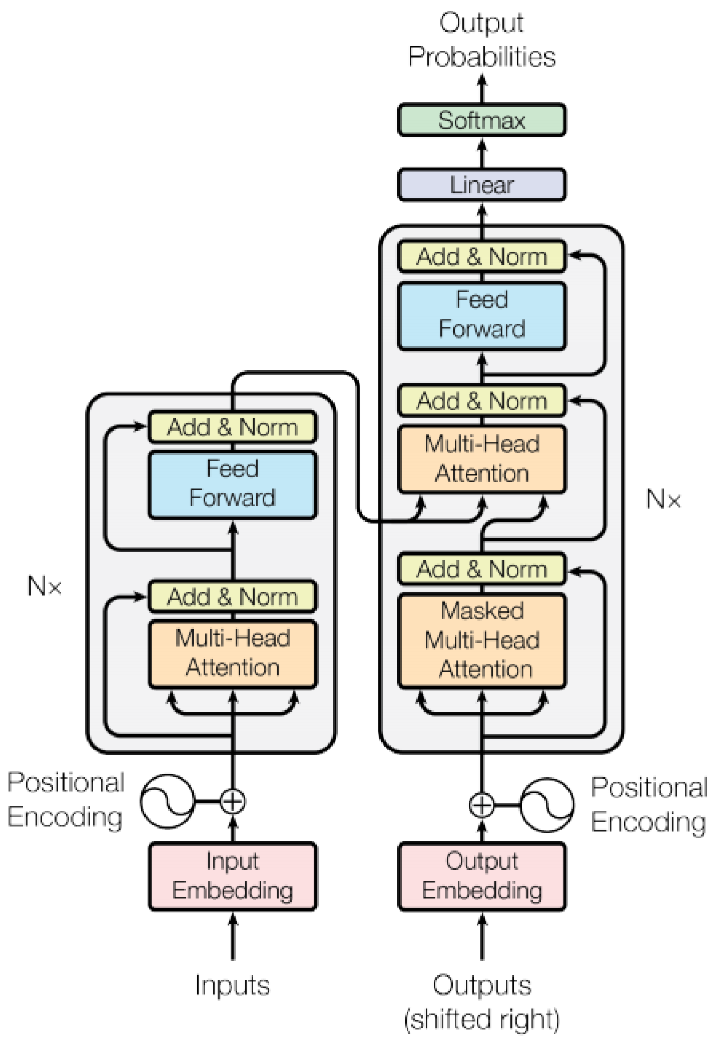


Big picture



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

Transformers



Vaswani et al. “Attention is all you need”, NeurIPS-2017.