# An Image Is Worth 16 × 16 Words : Transformers For Image Recognition at Scale

## - ICLR 21 -

Kookmin Univ.
Se Won Hong

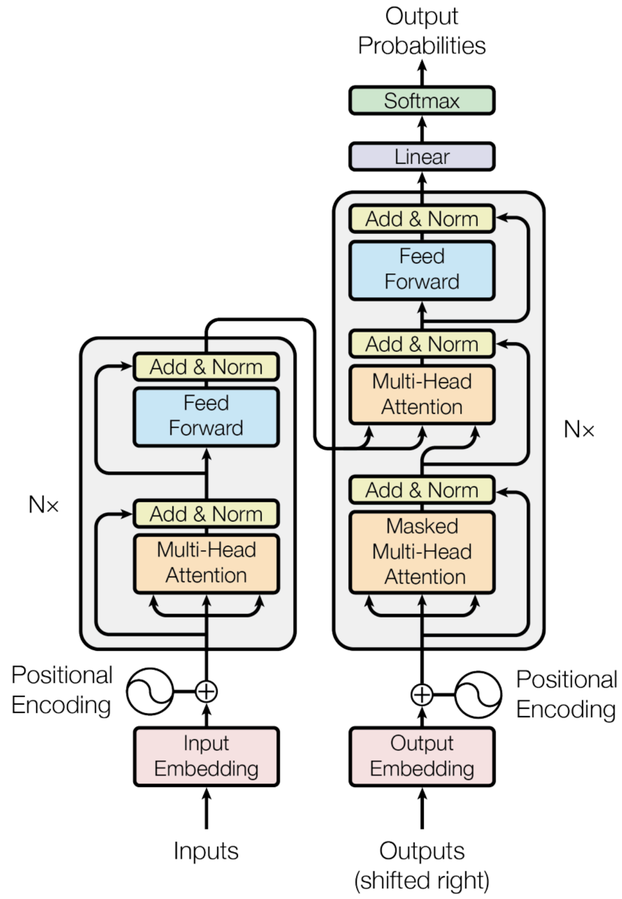
# Index

# About Transformer

# About Transformer

## What is Transformer

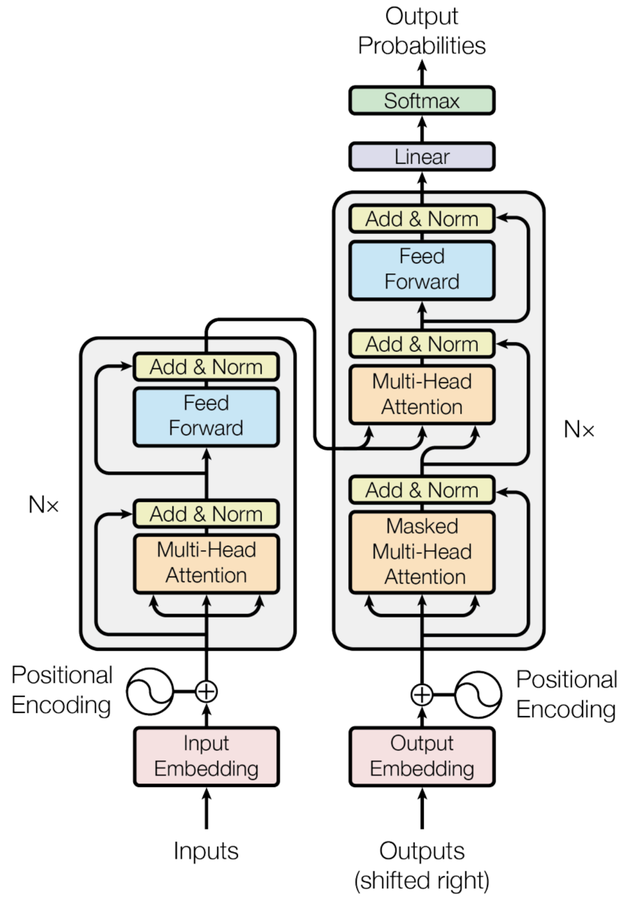
In deep learning, **transformer** is an architecture based on the multi-head attention mechanism, in which text is converted to numerical representations called tokens, and each token is converted into a vector via lookup from a word embedding table.[1] At each layer, each token is then contextualized within the scope of the context window with other (unmasked) tokens via a parallel multi-head attention mechanism, allowing the signal for key tokens to be amplified and less important tokens to be diminished.



# About Transformer

## What is Transformer

In deep learning, **transformer** is an architecture based on the multi-head attention mechanism, in which text is converted to numerical representations called tokens, and each token is converted into a vector via lookup from a word embedding table.[1] At each layer, each token is then contextualized within the scope of the context window with other (unmasked) tokens via a parallel multi-head attention mechanism, allowing the signal for key tokens to be amplified and less important tokens to be diminished.



# About Transformer

### What is Attention mechanism

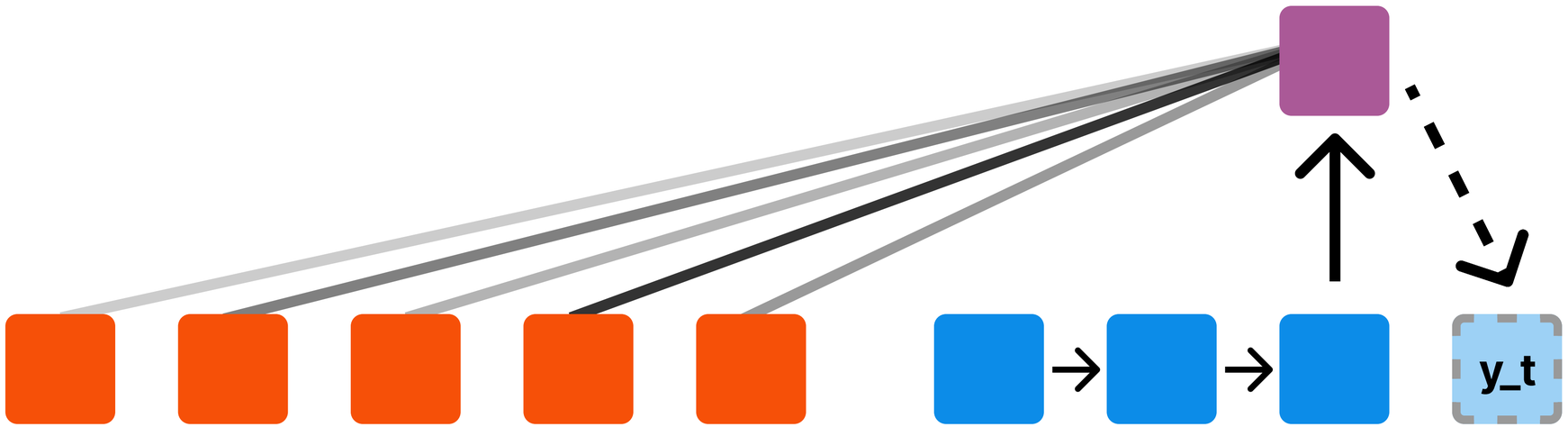
Attention is a mechanism that focuses on
important parts of the input and combines information with weighted sums.

- Cross-Attention
- Self-Attention

# About Transformer

### What is Attention mechanism_Cross Attention

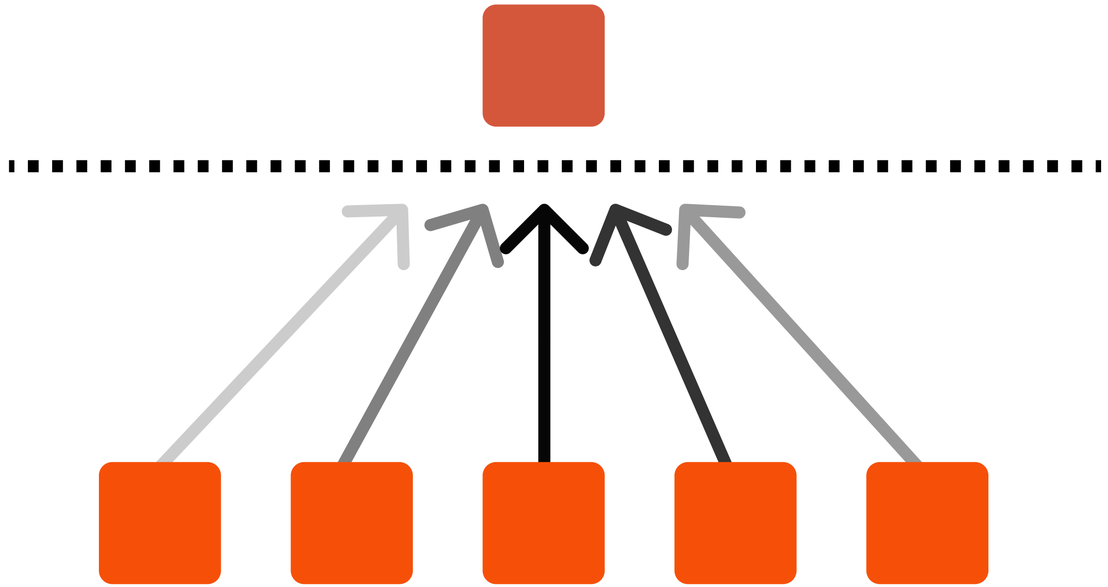
Attention is a mechanism that focuses on
important parts of the input and combines information with weighted sums.



# About Transformer

## What is Attention mechanism_Self Attention

Attention is a mechanism that focuses on
important parts of the input and combines information with weighted sums.



# About Transformer

## How to get Attention Score

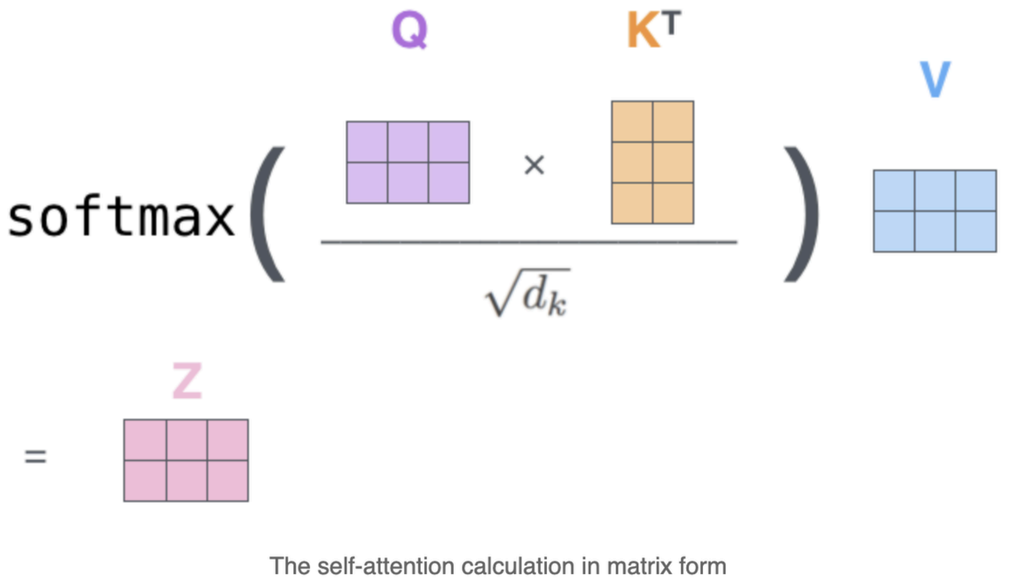
**X : n x d**

**W : d x d_k**

**Q = X x W_q**

**K = X x W_k**

**V = X x W_v**

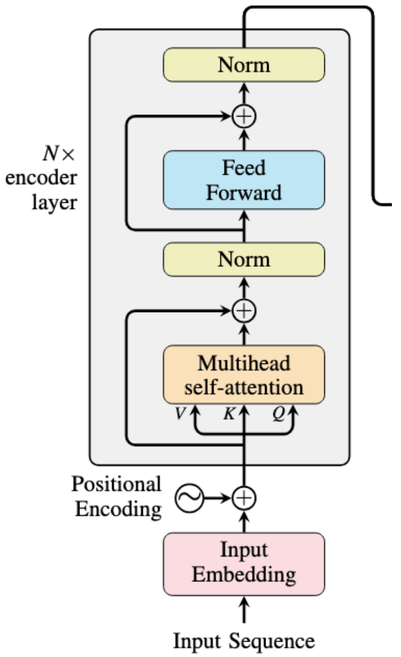



The self-attention calculation in matrix form

# About Transformer

## What is Transformer_Encoder

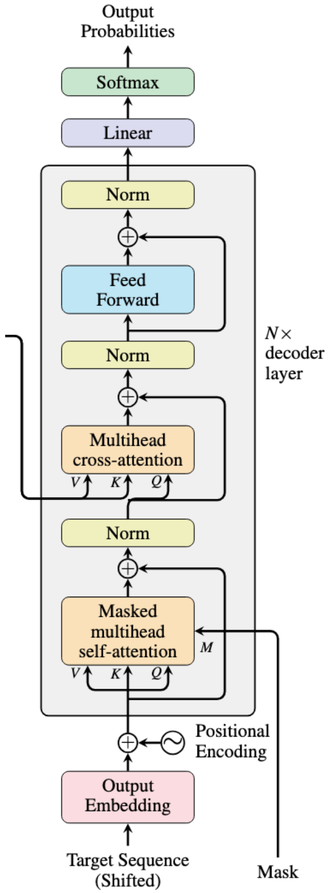
- Input tokens are embedded with positional encoding to capture order
- Multi-head self-attention and feed-forward layers with residual connections enable stable learning



# About Transformer

## What is Transformer_Decoder

- Applies masked self-attention on previous output tokens to block future information
- Uses encoder outputs as Keys and Values; decoder states as Queries for cross-attention to input context



# Introduction

# Introduction

## How to apply the Transformer to Images

## Related Work Idea

- Applied the Self Attention only in Local Neighborhoods
- Replace Convolution with Self Attention
- Reduced computation using sparsely connected self-attention
- Scaled attention using blocks of varying sizes
- Applied attention only along individual axes (1D) for efficiency

# Introduction

## How to apply the Transformer to Images

**Related Work Idea**

- Applied the Self-Attention only in Local Neighborhoods
- Replace Convolution
- Reduced-cost computation using
- Scaled attention using blocks of varying sizes
- Applied attention only along individual axes [3] for efficiency

**Still involves high engineering cost for hardware implementation**

# Introduction

## How to apply the Transformer to Images

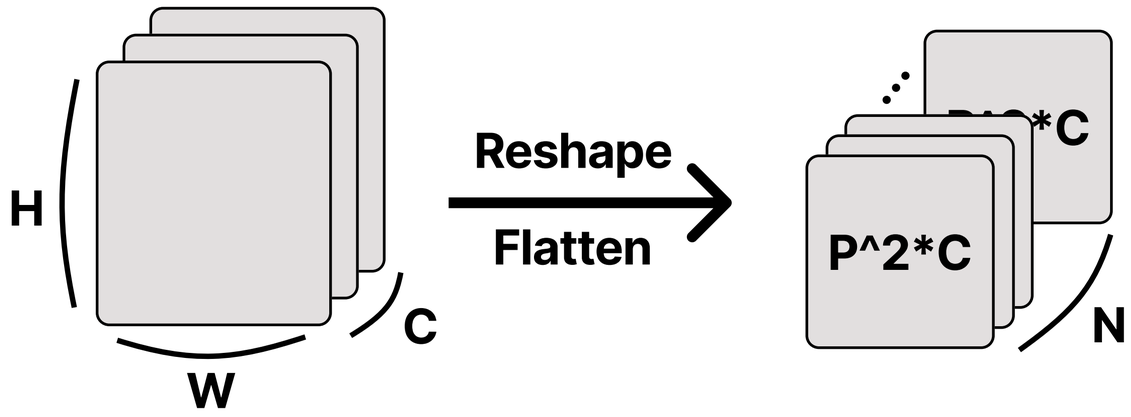
## ViT

- Follow the original Transformer architecture (Vaswani et al., 2017) as closely as possible
- Intentionally keep the design simple and clean
- Allows using scalable NLP Transformer architectures and efficient implementations

# Method

# Method

## Image Preprogressing

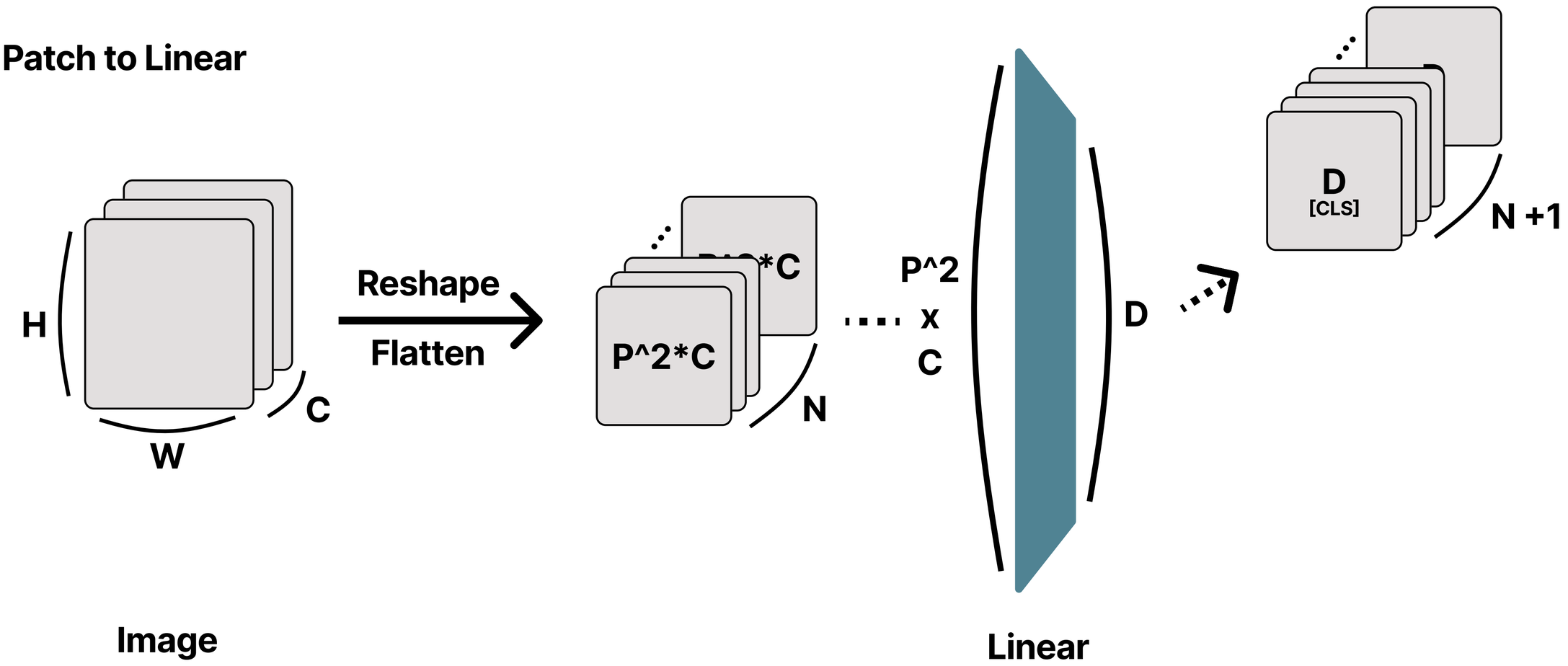



$N = H*W/P^2$

Image

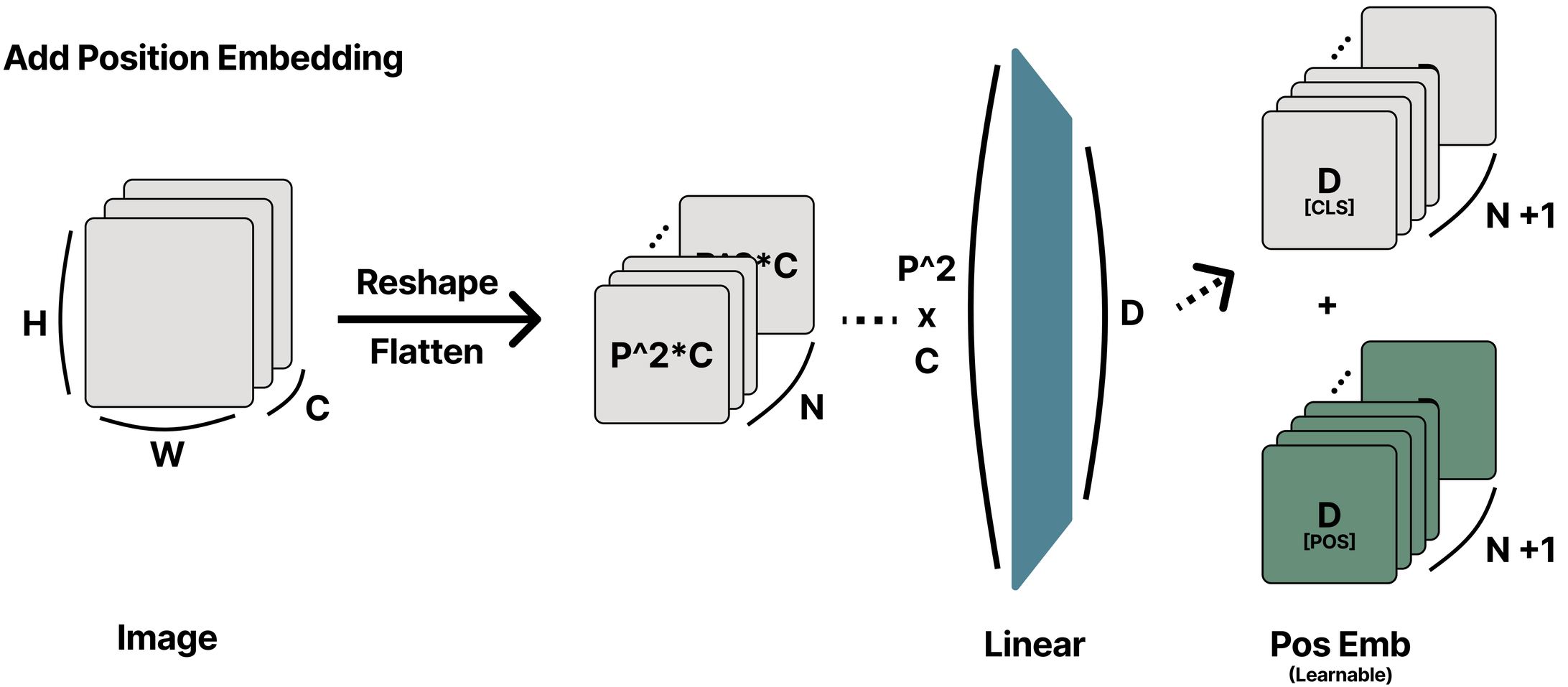
# Method

## Patch to Linear



# Method

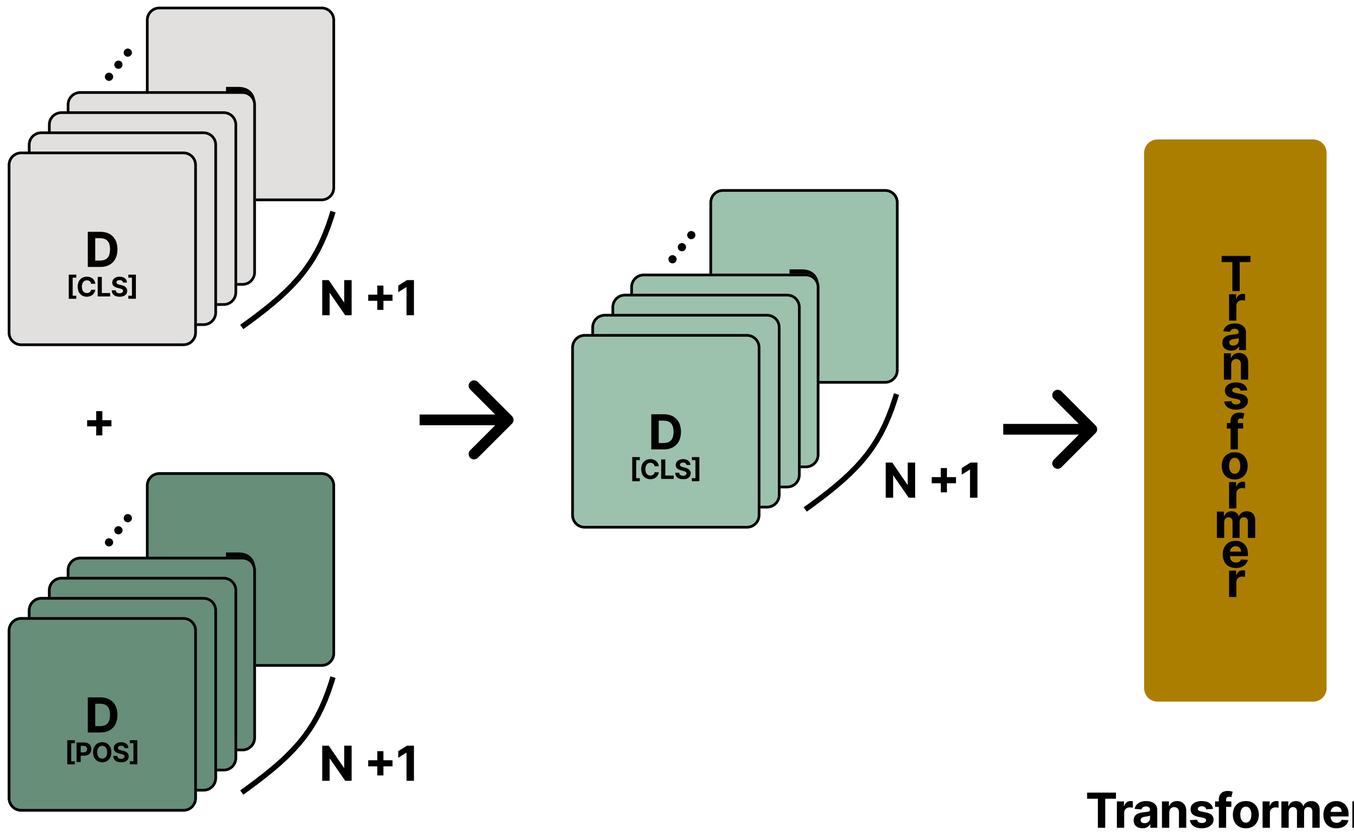
## Add Position Embedding




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

# Method

## Transformer

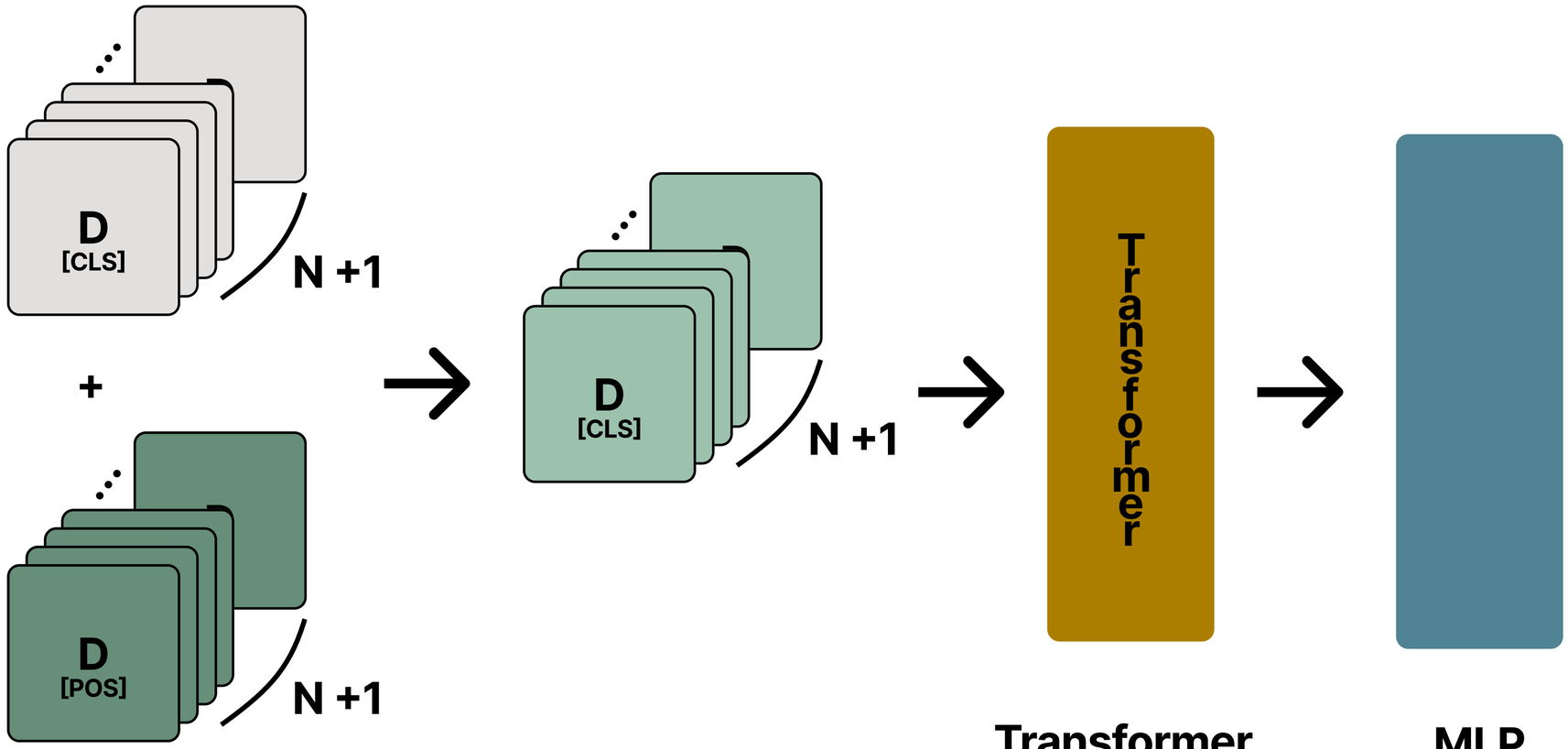



### Transformer

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

# Method

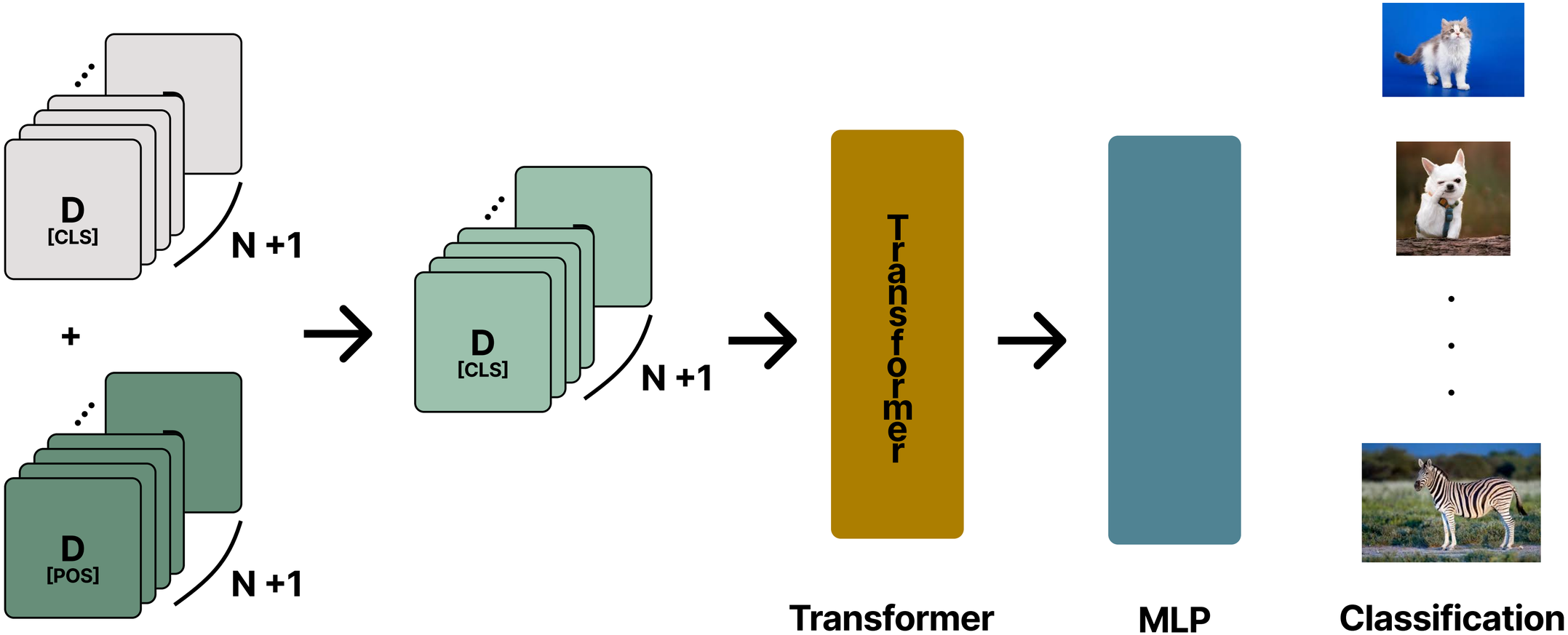
## MLP Head




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

# Method

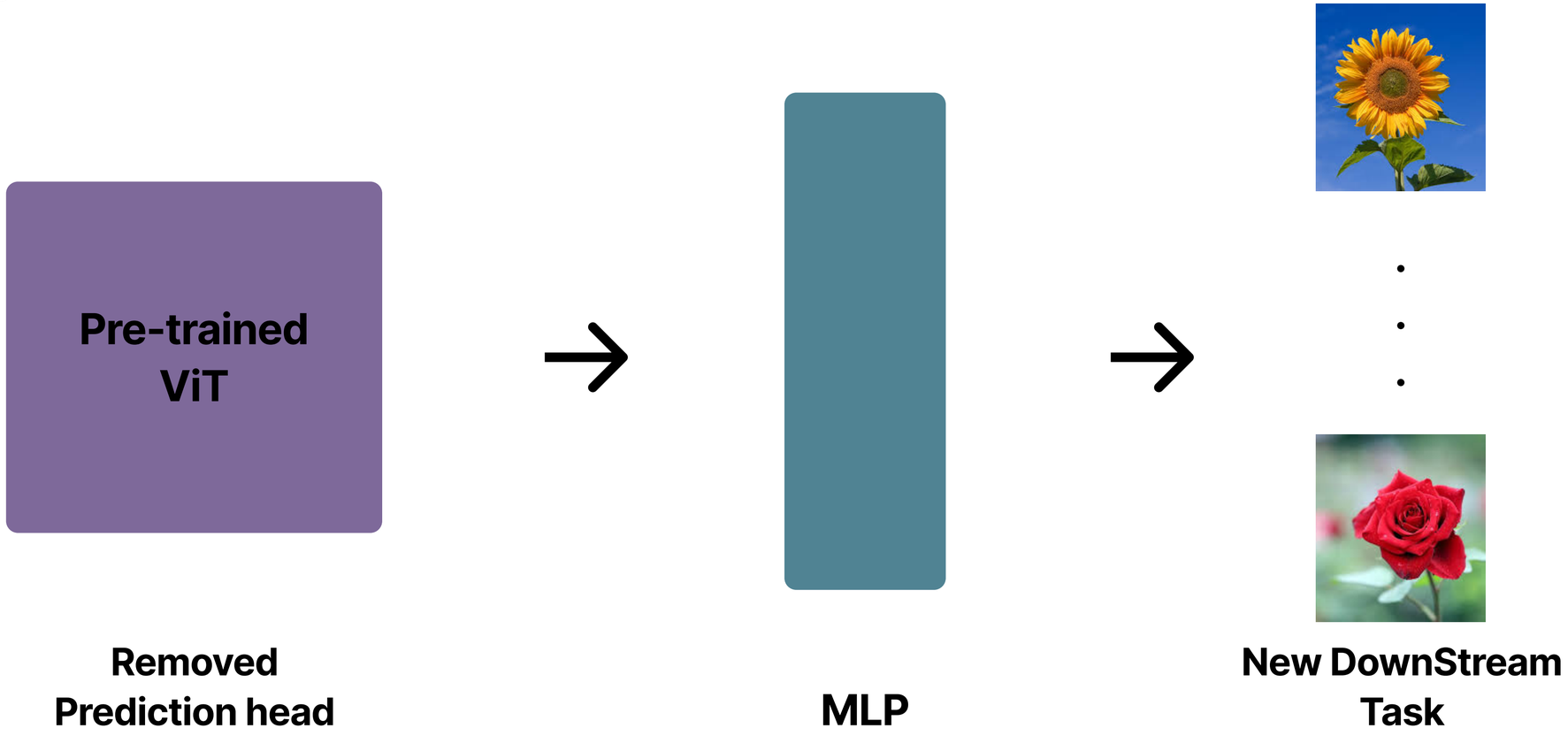
## Classification




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

# Method

## Finetuning ViT



# Experiments

# Experiments

## Pre-trained Datasets

| Dataset Name | Class Num | Image Num |
| --- | --- | --- |
| **ImageNet-21k** | 21,000 | 14M |
| **JFT** | 18,000 | 303M |

- Without strong inductive biases, ViT needs large data to learn image priors.

# Experiments

## Fine-tuning and Evaluation Datasets

| Dataset Name | # Classes | # Images |
|---|---|---|
| **ImageNet (ILSVRC-2012)** | 1,000 | 1.3M |
| **CIFAR-10/100** | 10 / 100 | 60K |
| **Oxford-IIIT Pets** | 37 | ~7,000 |
| **Oxford Flowers-102** | 102 | ~8,000 |
| **VTAB** | 19 tasks | 1,000 samples each |

# Experiments

| | 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 |

Superior Performance Compared to Previous State-of-the-Art ResNet