

# Vision Transformer (ViT)

**Shusen Wang**

Stevens Institute of Technology

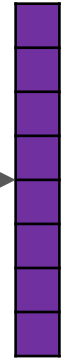
<http://wangshusen.github.io/>



What is in the image?



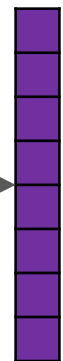
Neural  
Network



$p$

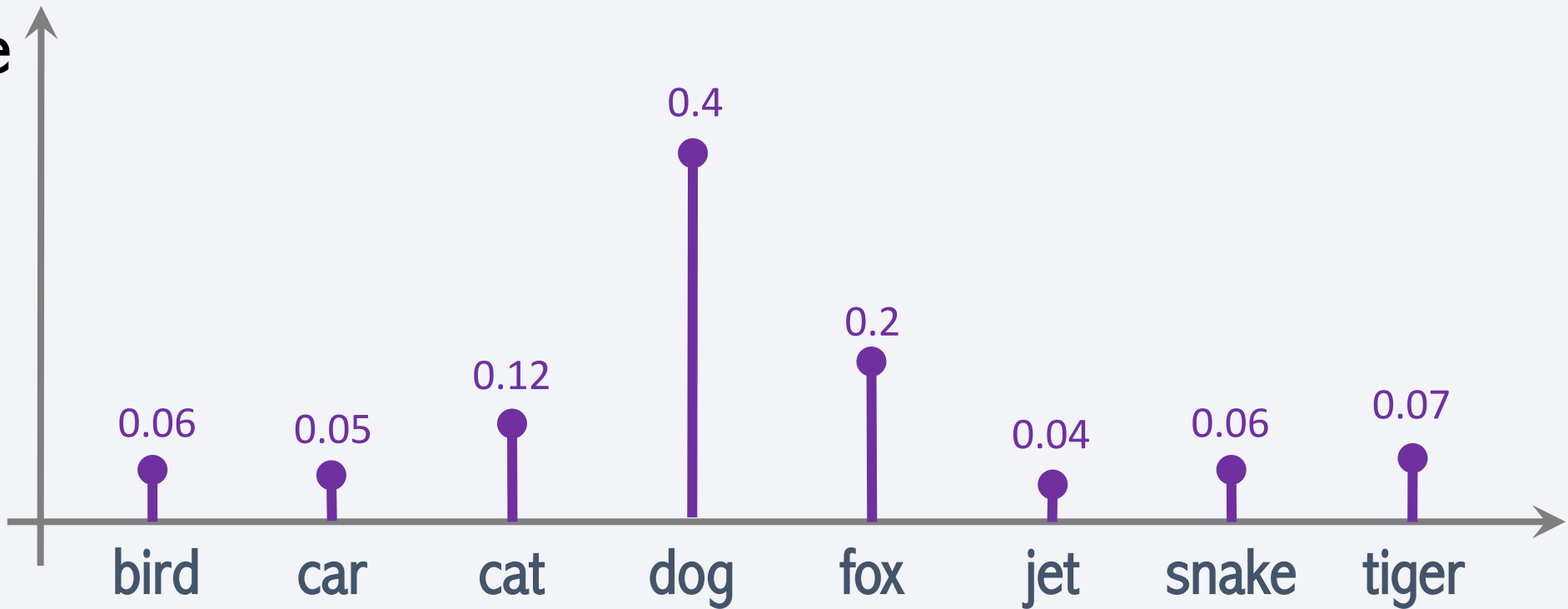


Neural  
Network



$p$

Confidence



Classes

# Image Classification

- CNNs, e.g., ResNet, were the best solutions to image classification.
- Vision Transformer (ViT) [1] beats CNNs (by a small margin), if the dataset for pretraining is sufficiently large (at least 100 million images).
- ViT is based on Transformer (for NLP) [2].

## Reference

1. Dosovitskiy et al. [An image is worth 16×16 words: transformers for image recognition at scale](#). In *ICLR*, 2021.
2. Vaswani et al. [Attention Is All You Need](#). In *NIPS*, 2017.

# Split Image into Patches

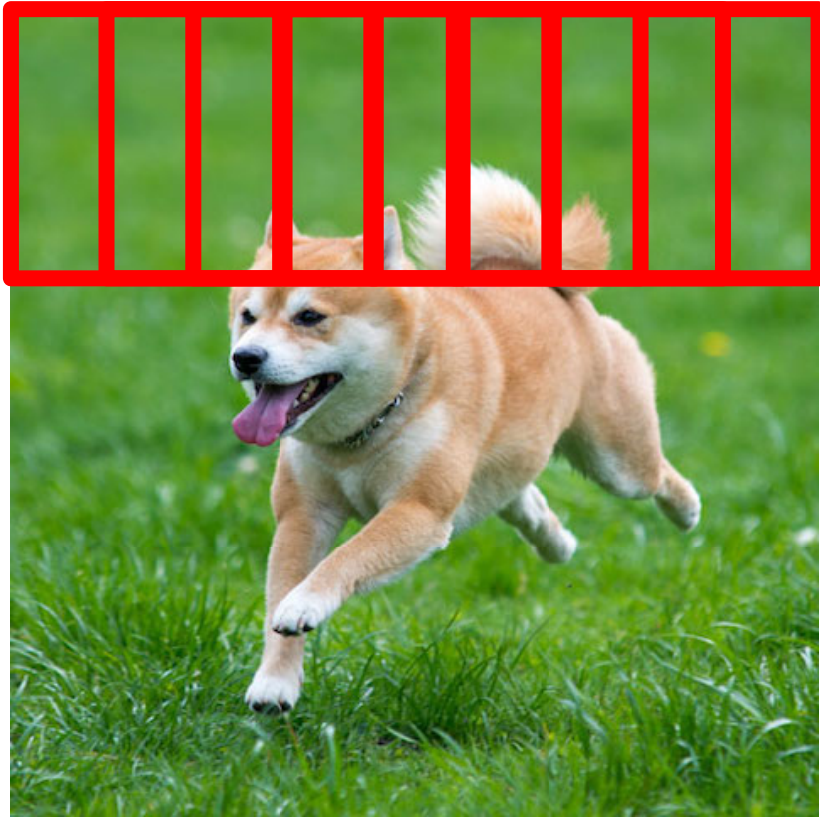


# Split Image into Patches



- Here, the patches do not overlap.

# Split Image into Patches

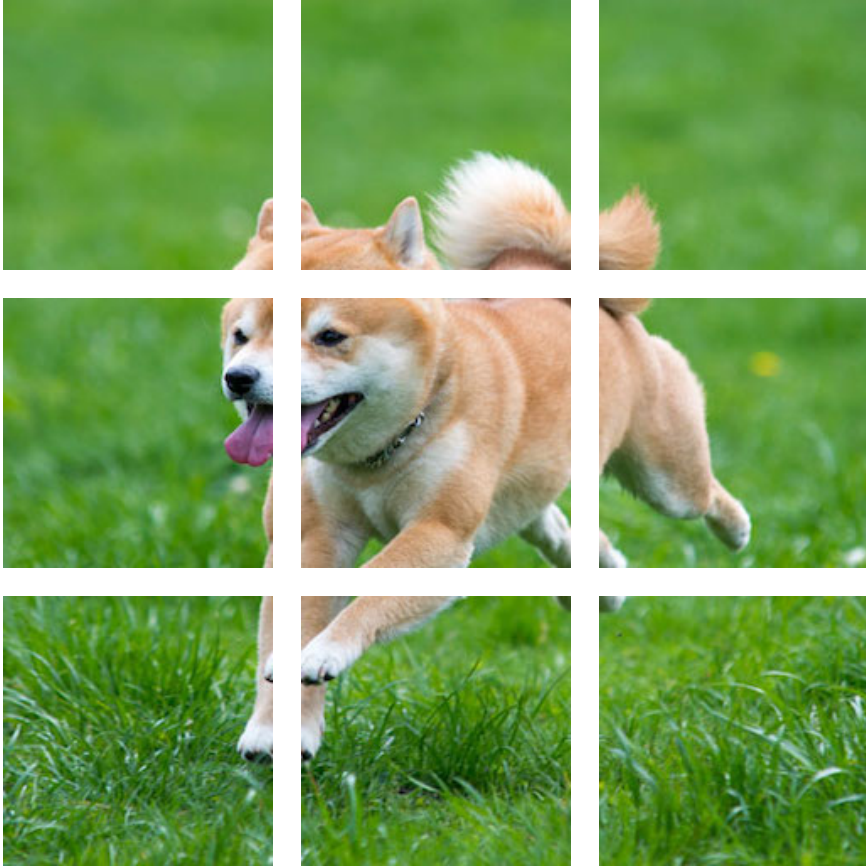


- Here, the patches do not overlap.
- The patches can overlap.
- User specifies:
  - **patch size**, e.g.,  $16 \times 16$ ;
  - **stride**, e.g.,  $16 \times 16$ .

# Vectorization

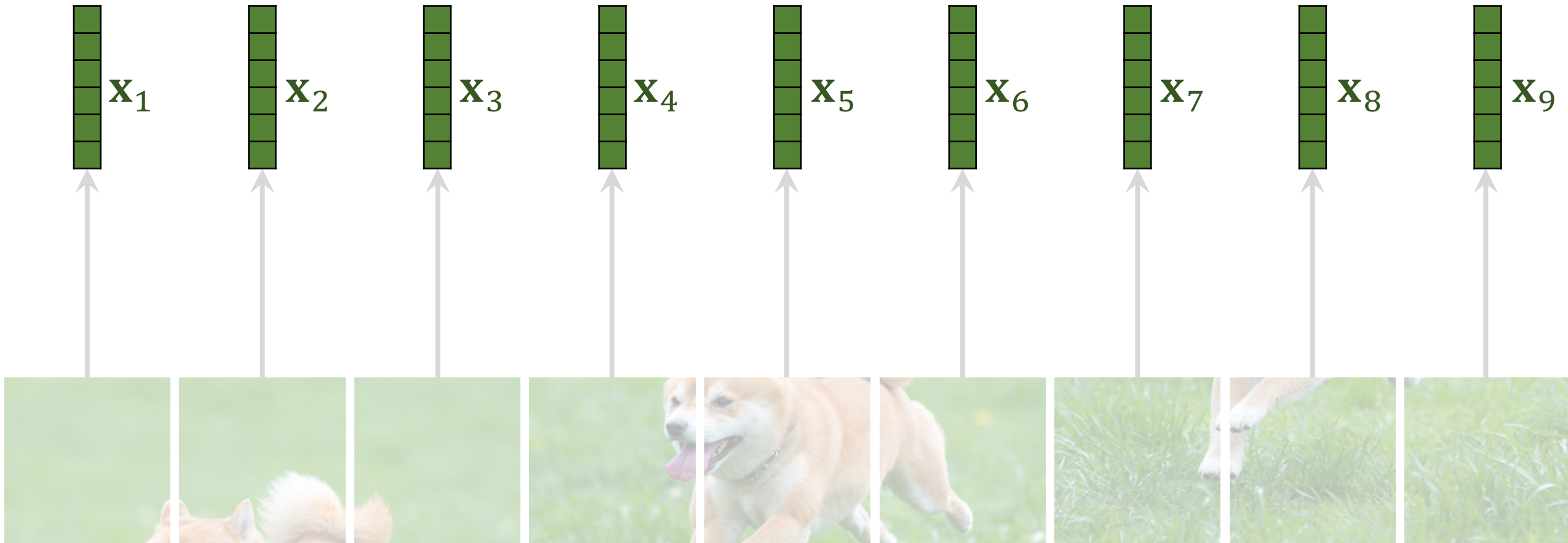


# Vectorization



# Vectorization

If the patches are  $d_1 \times d_2 \times d_3$  tensors, then the vectors are  $d_1 d_2 d_3 \times 1$ .





$\mathbf{x}_1$



$\mathbf{x}_2$

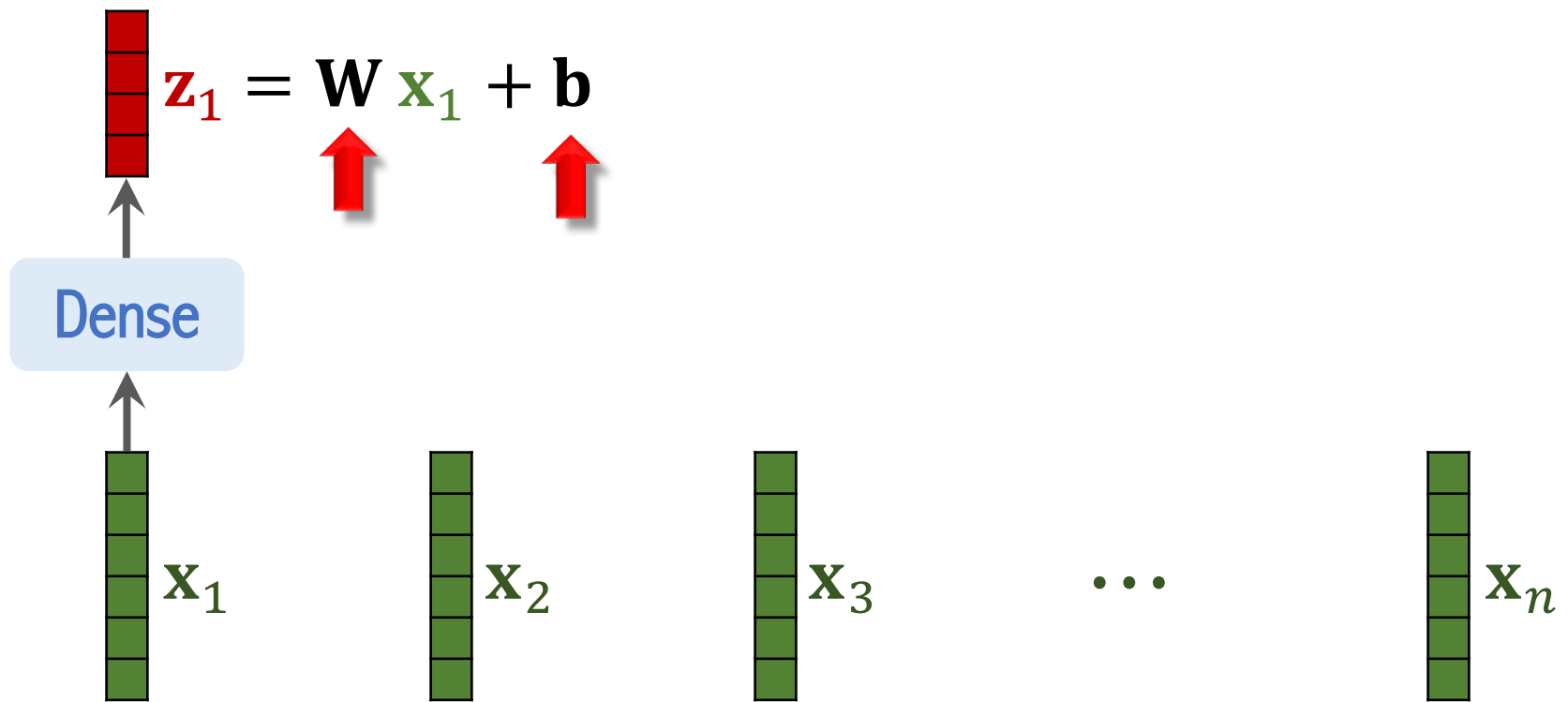


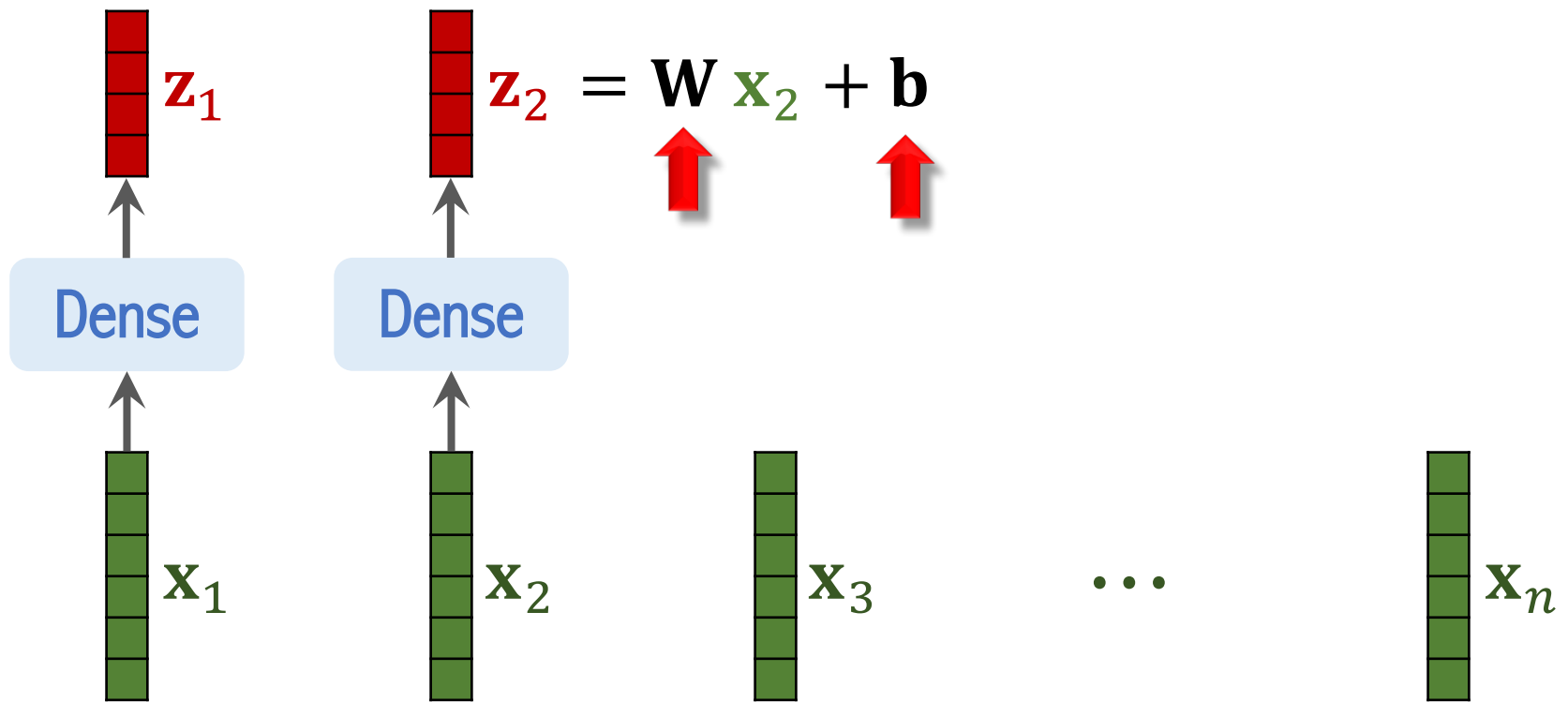
$\mathbf{x}_3$

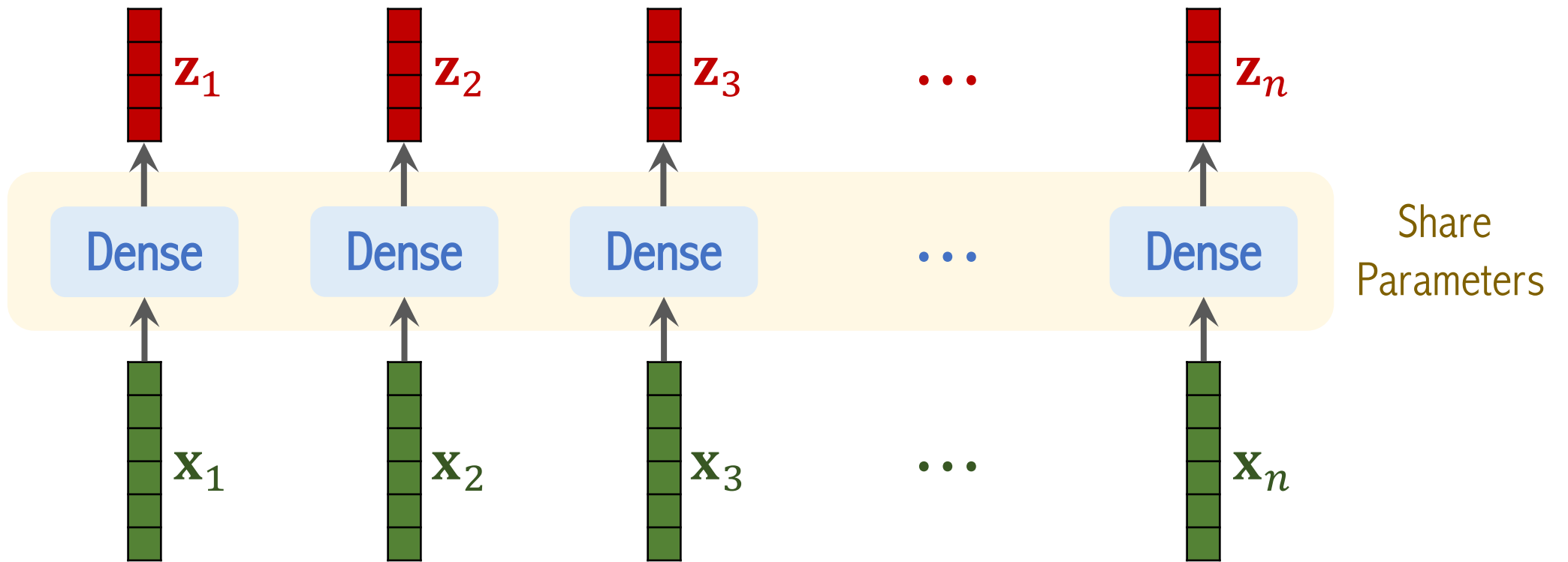
...



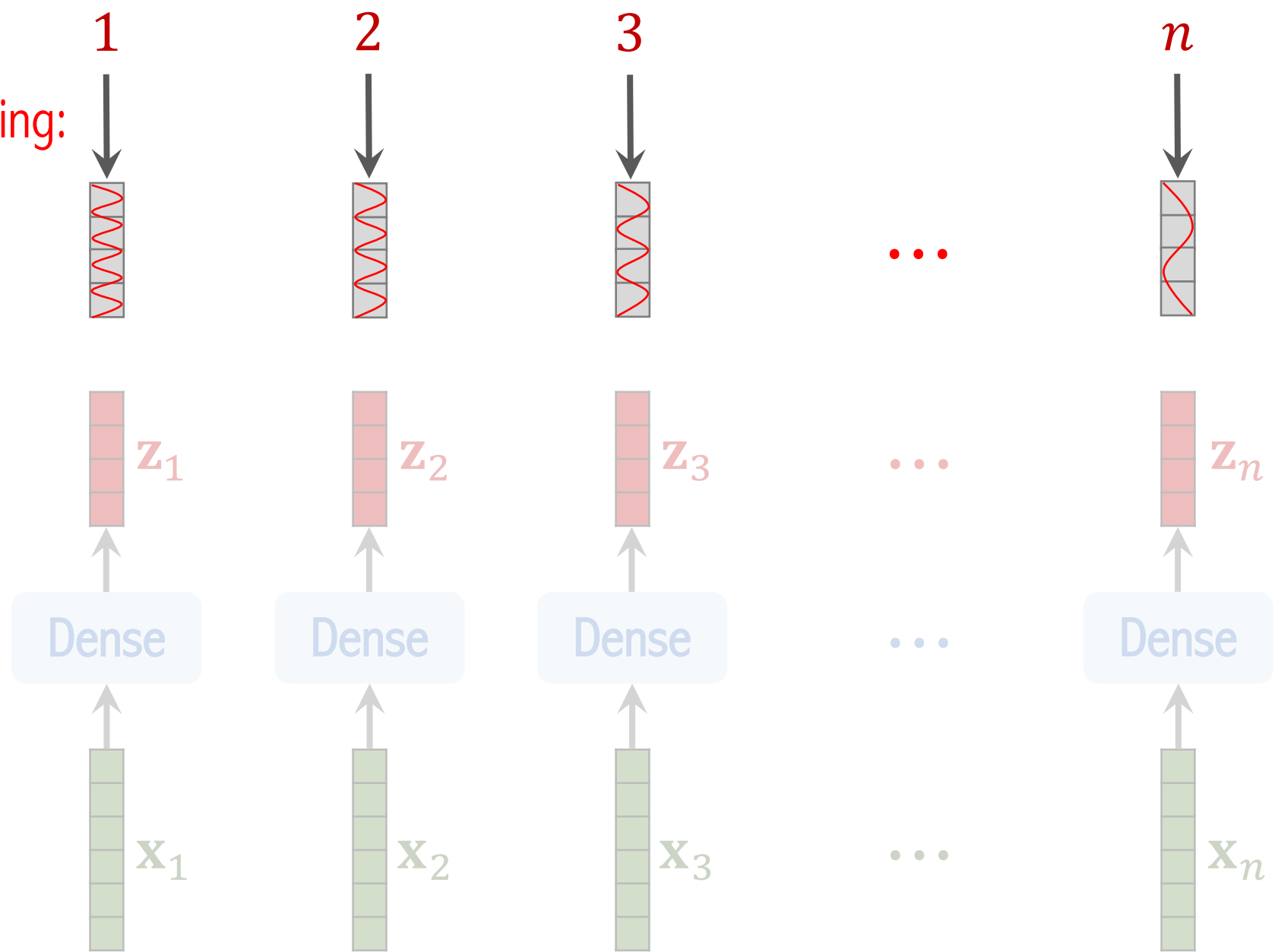
$\mathbf{x}_n$



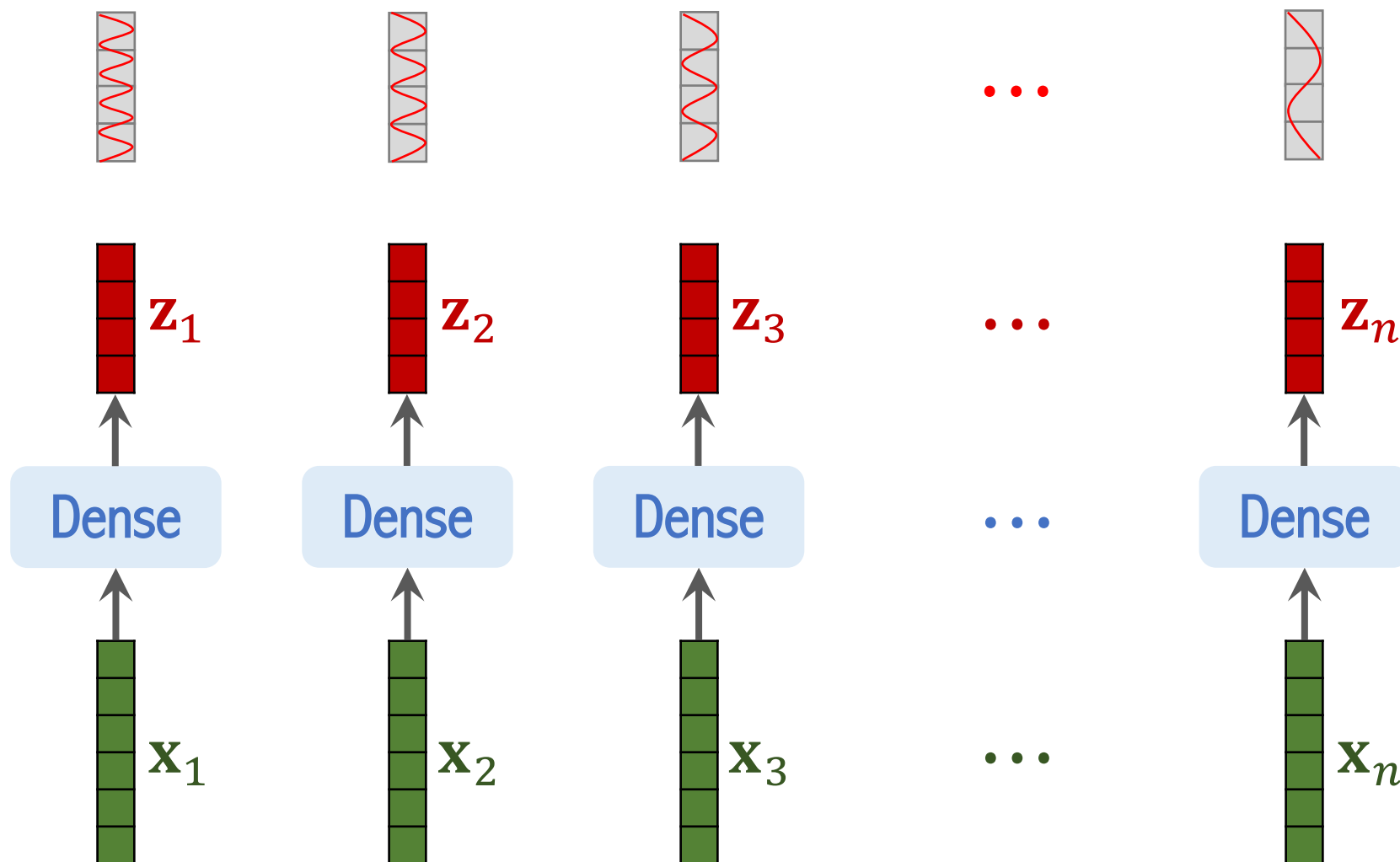




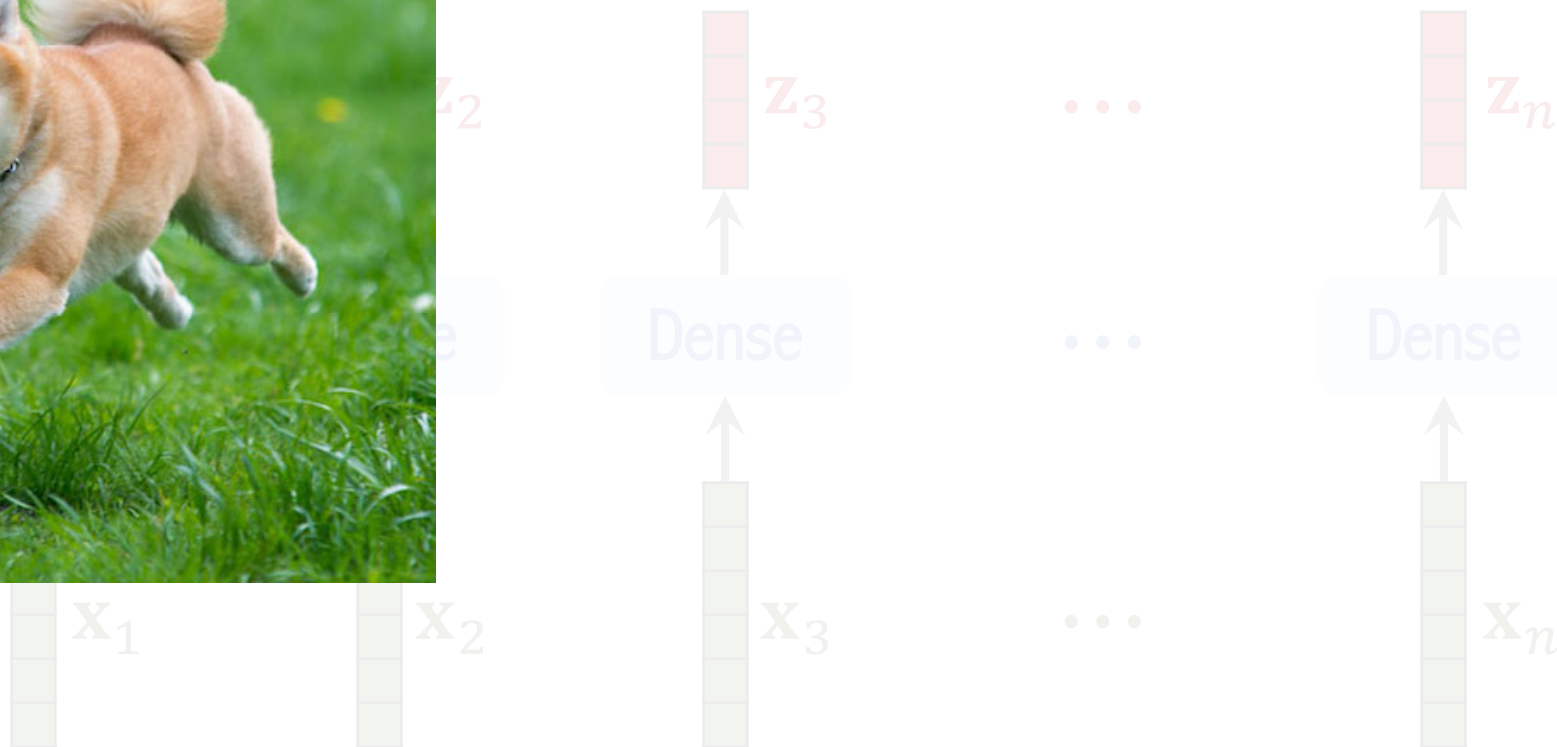
Positional Encoding:



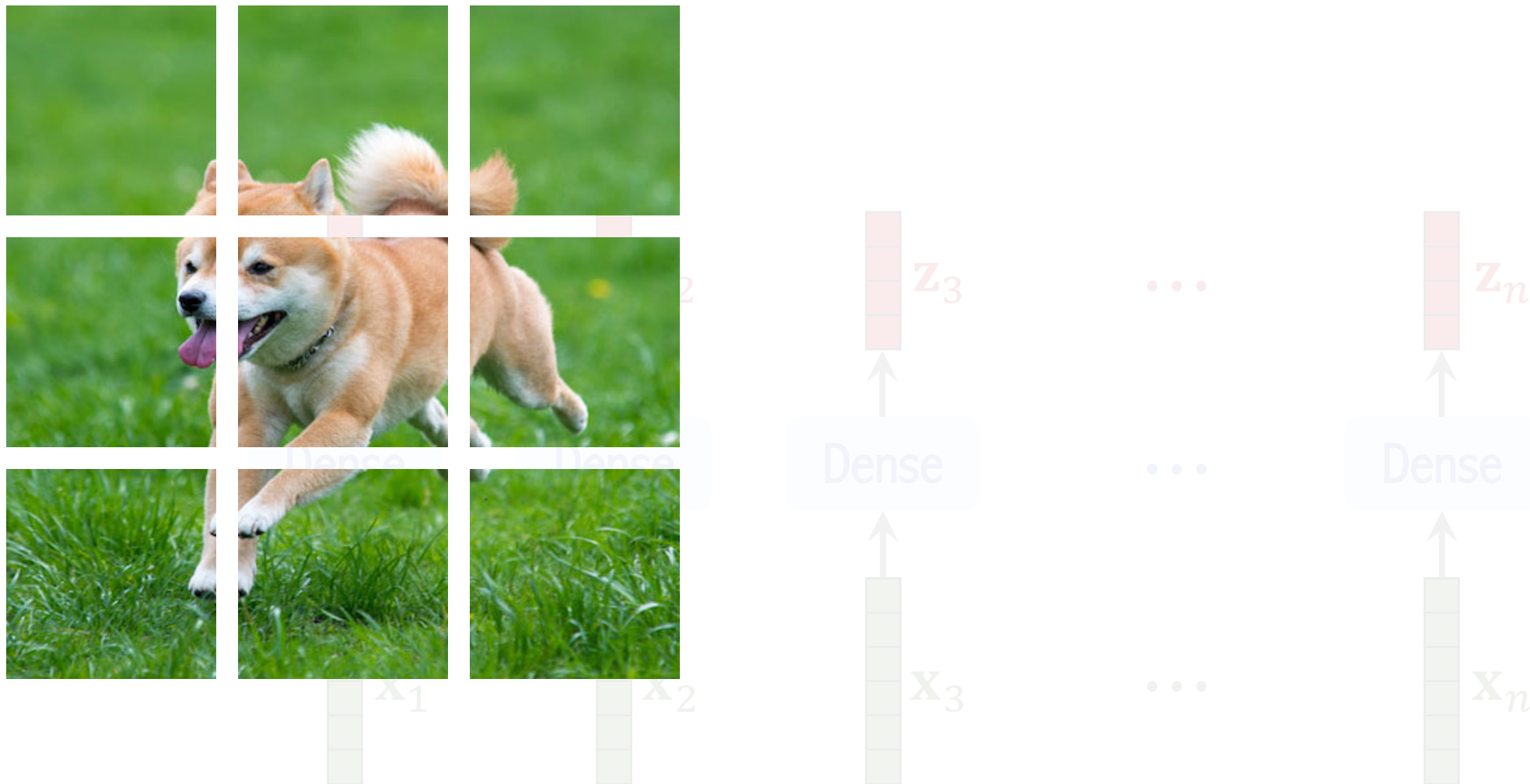
Add positional encoding vectors to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ .



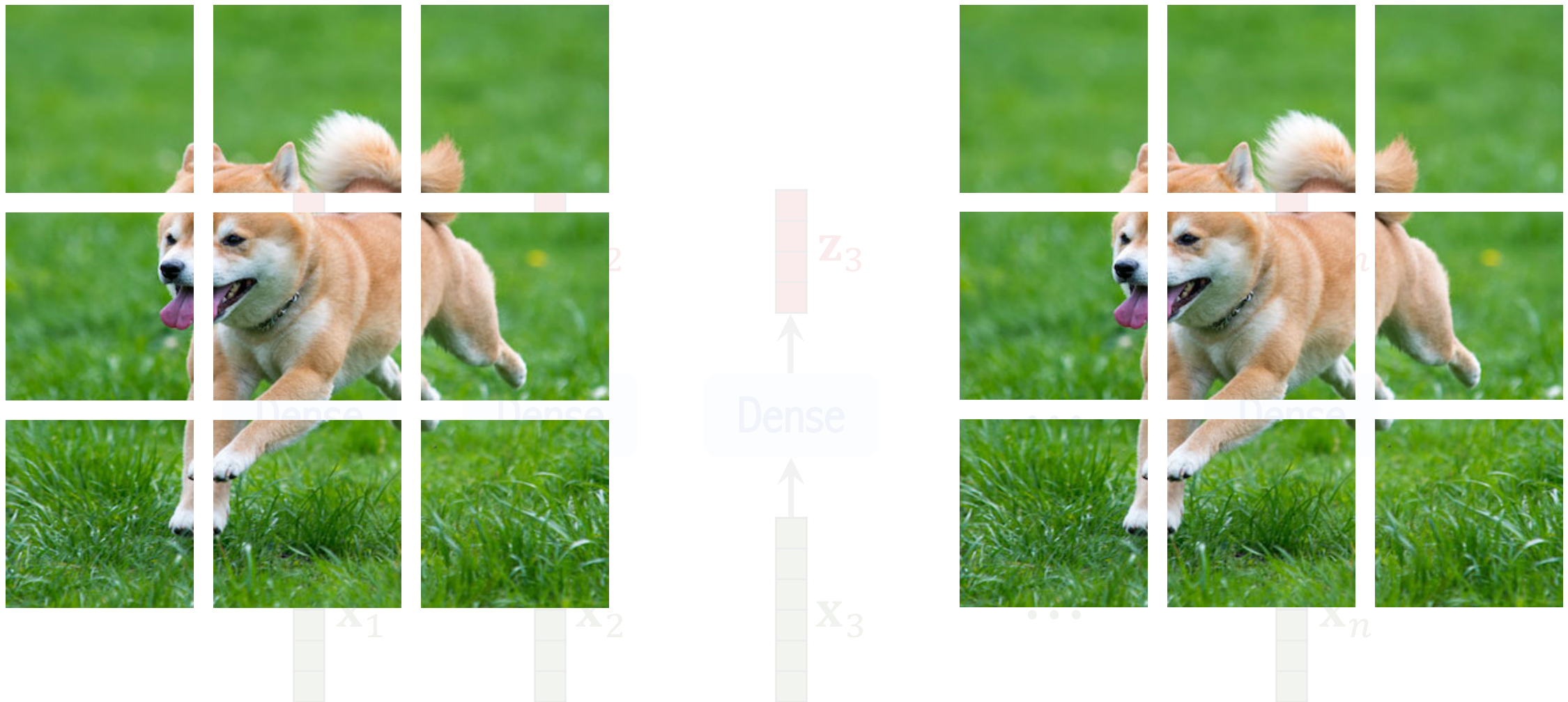
Add positional encoding vectors to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ . (Why?)



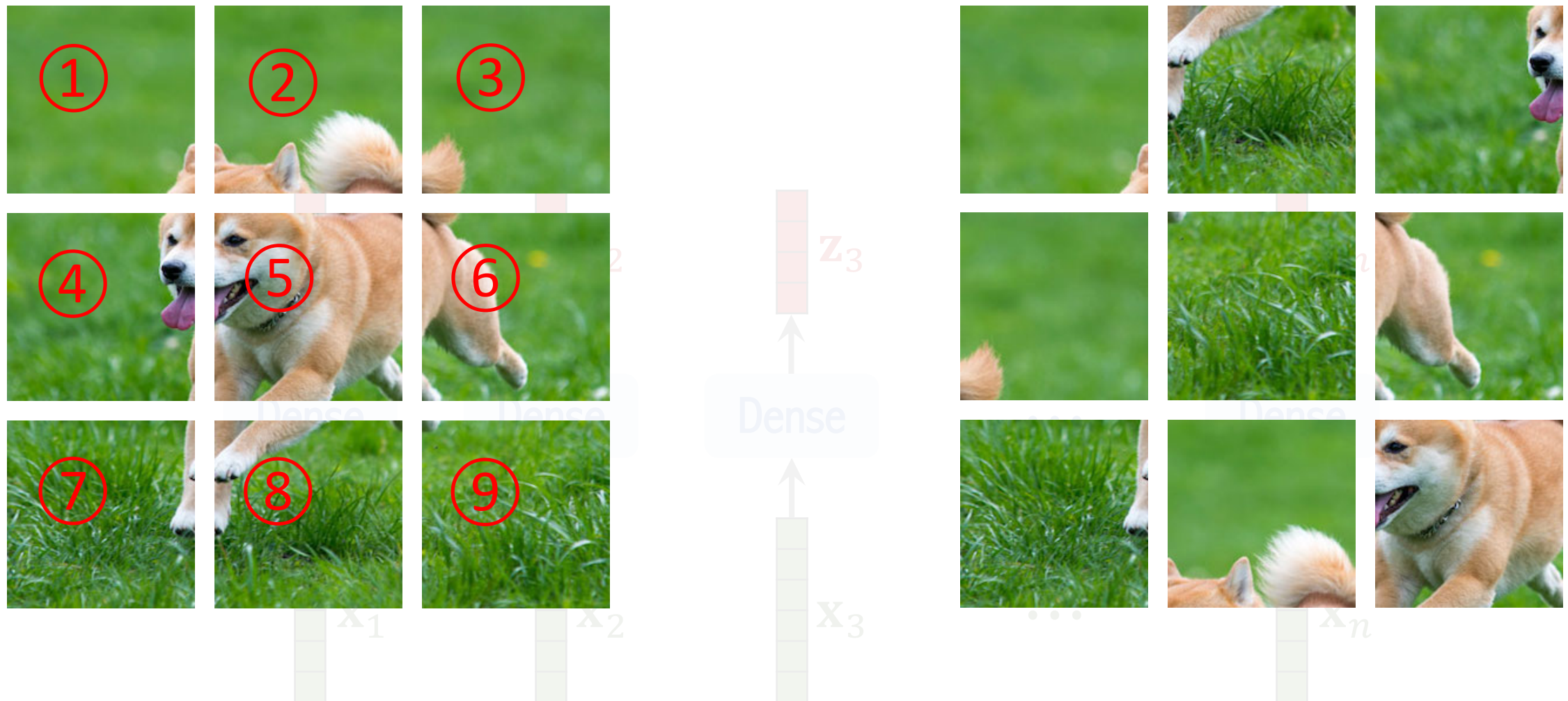
Add positional encoding vectors to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ . (Why?)

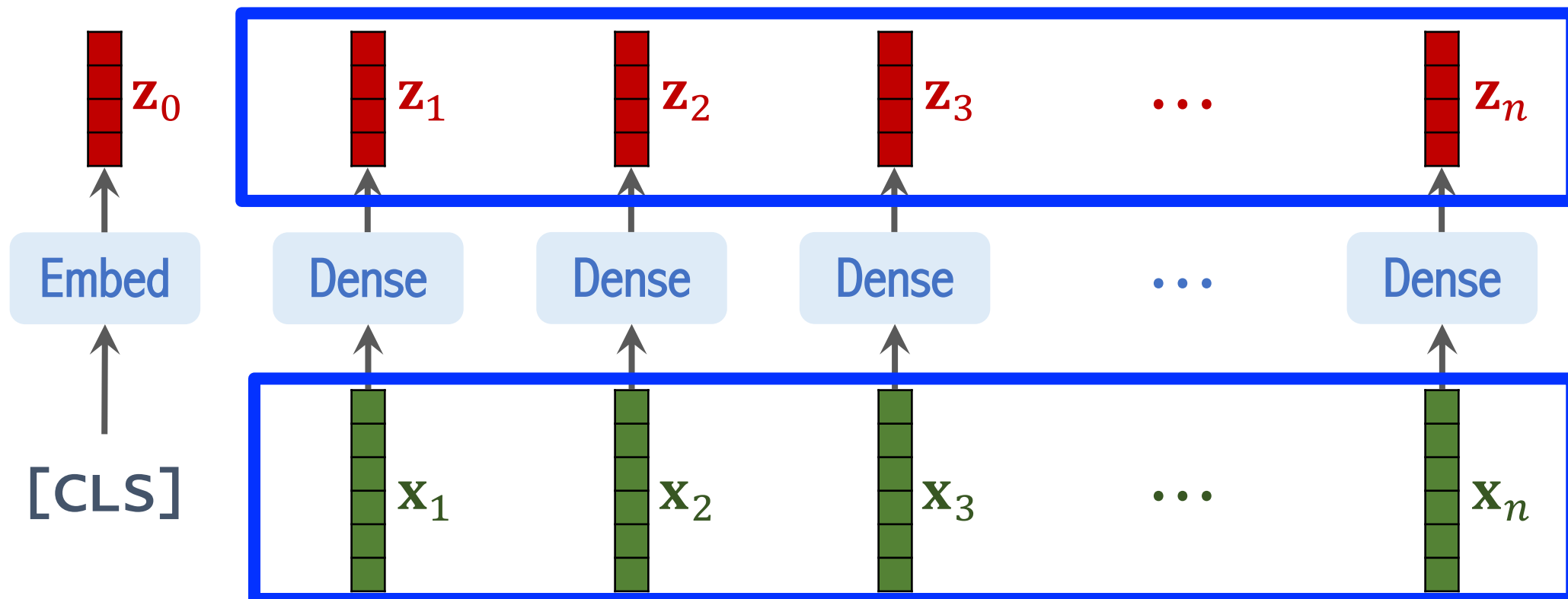


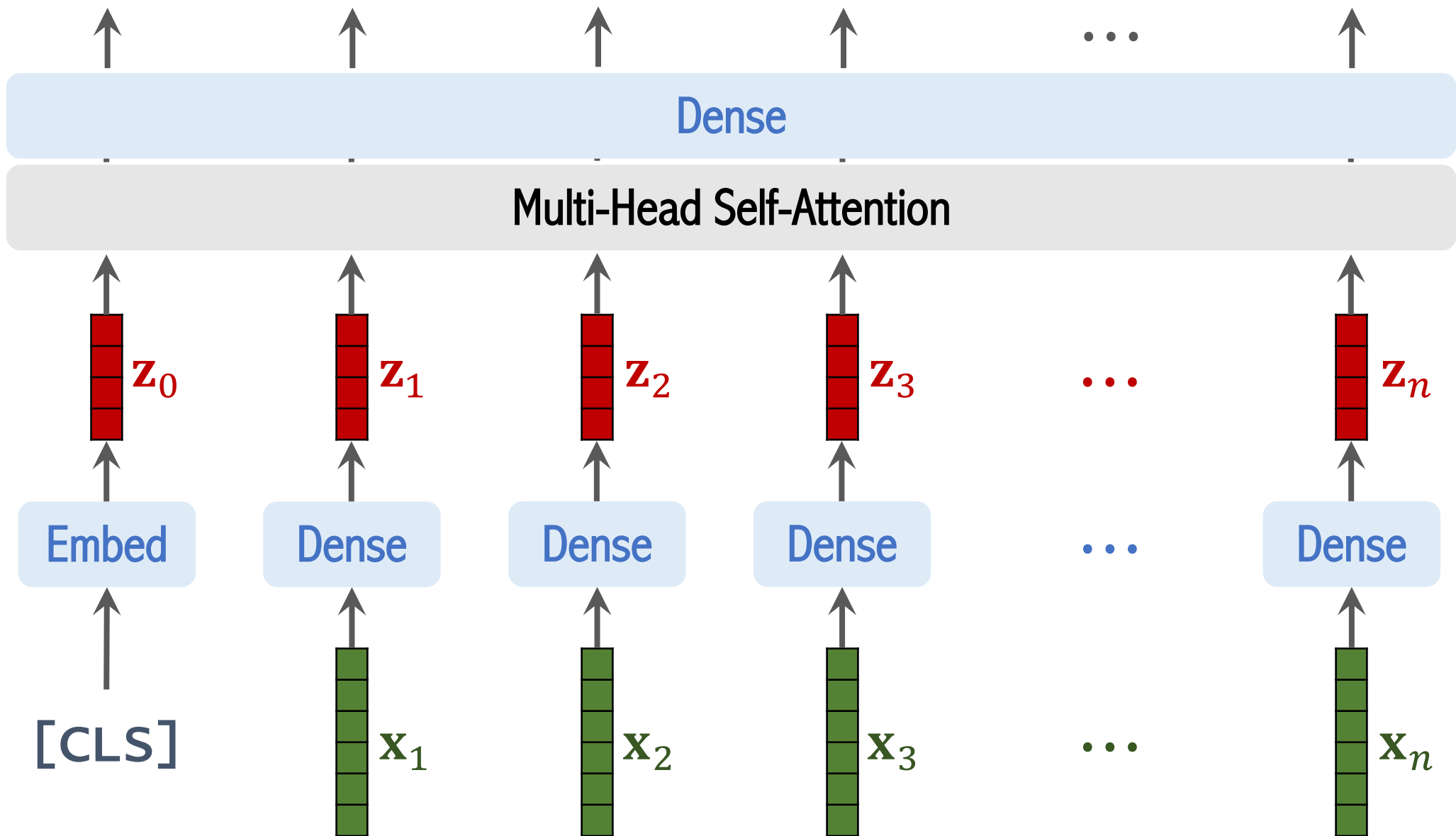
Add positional encoding vectors to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ . (Why?)

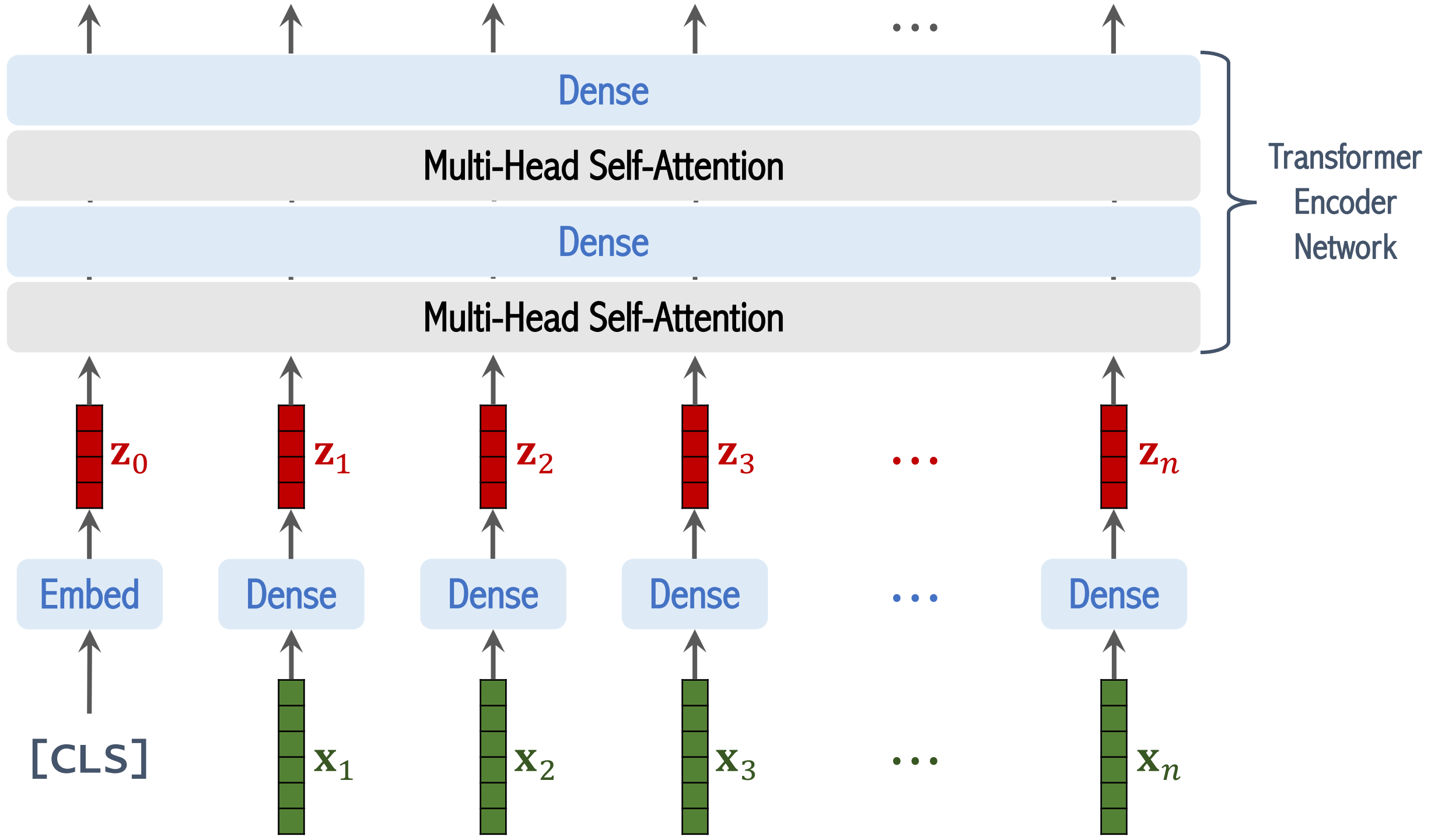


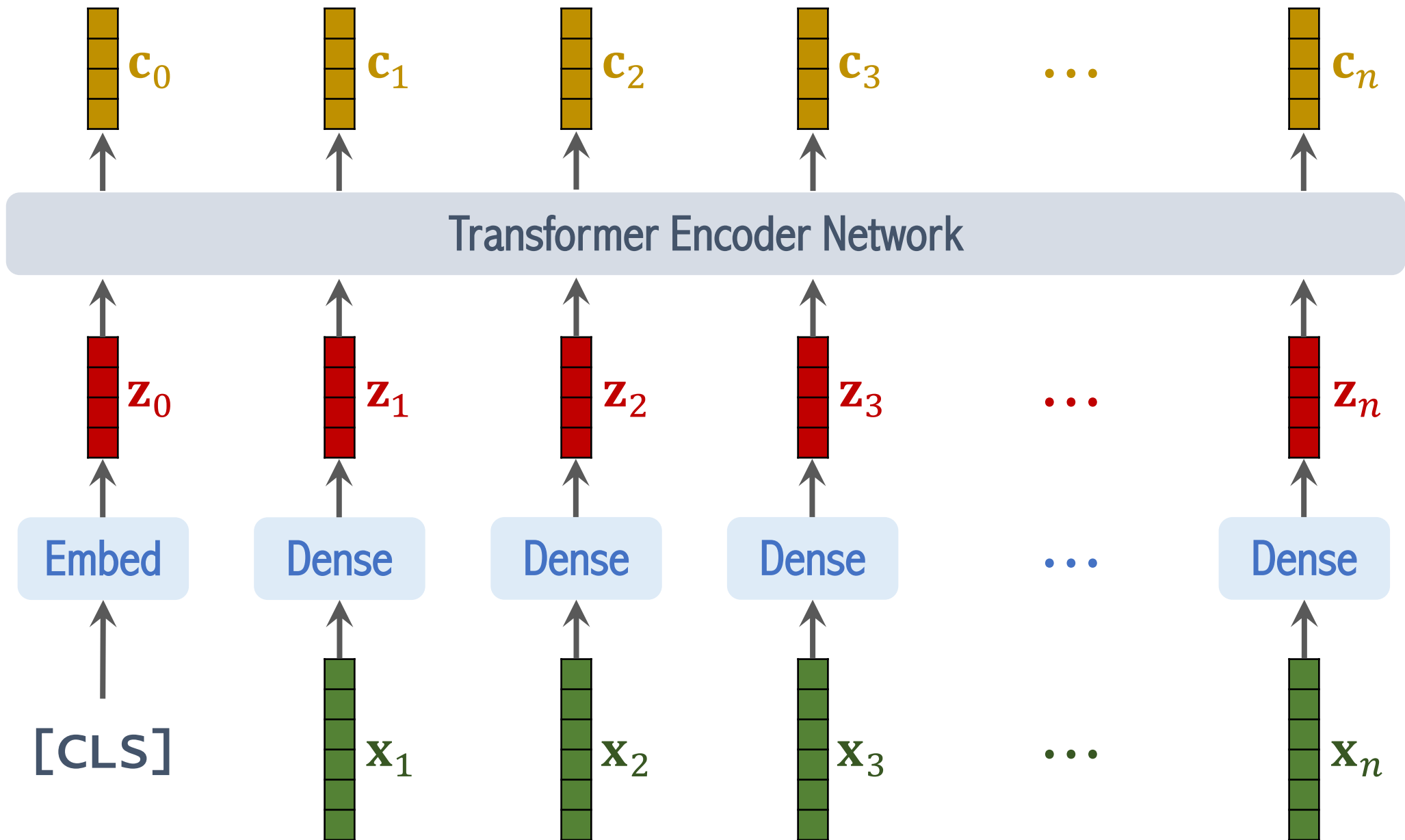
Add positional encoding vectors to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ . (Why?)

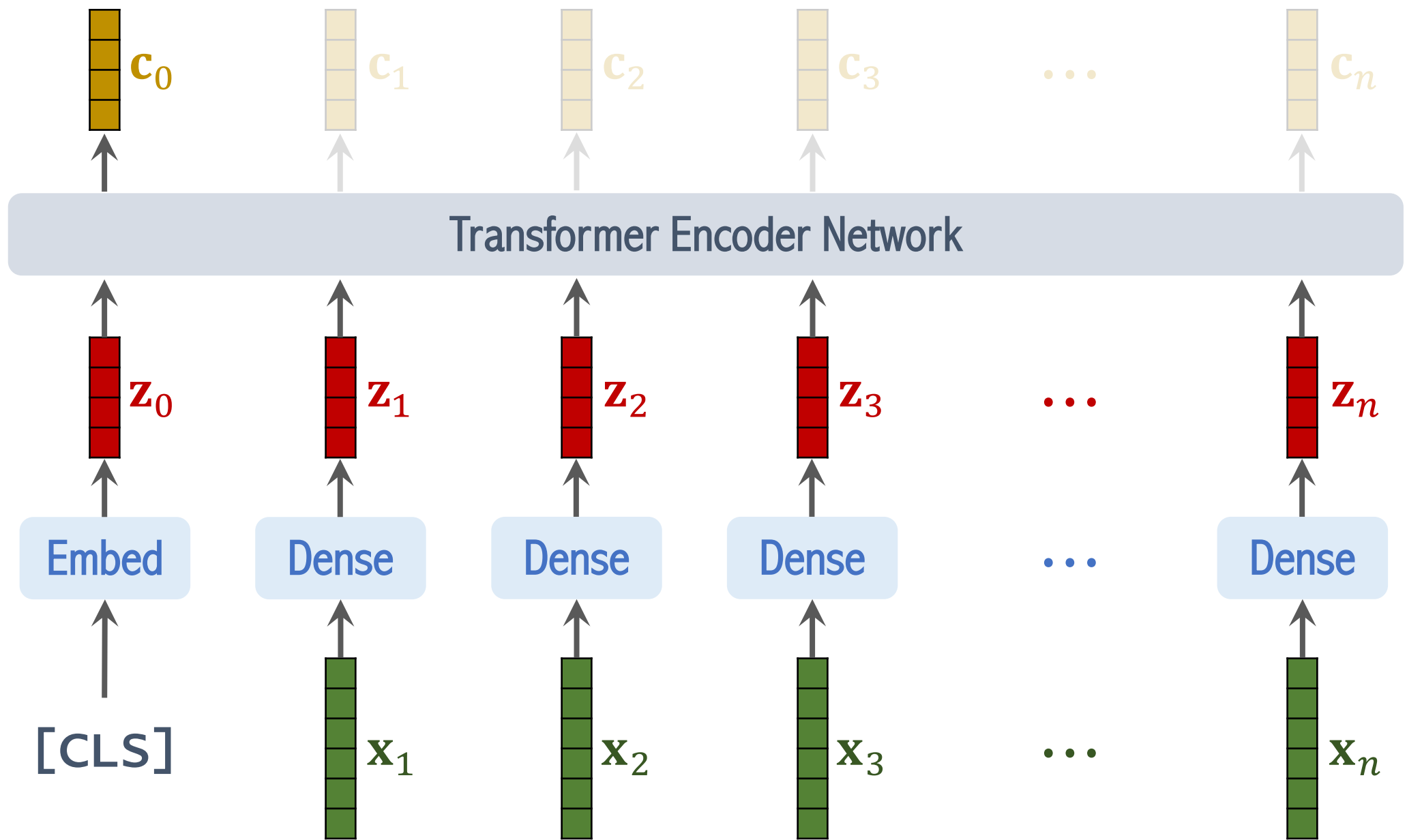


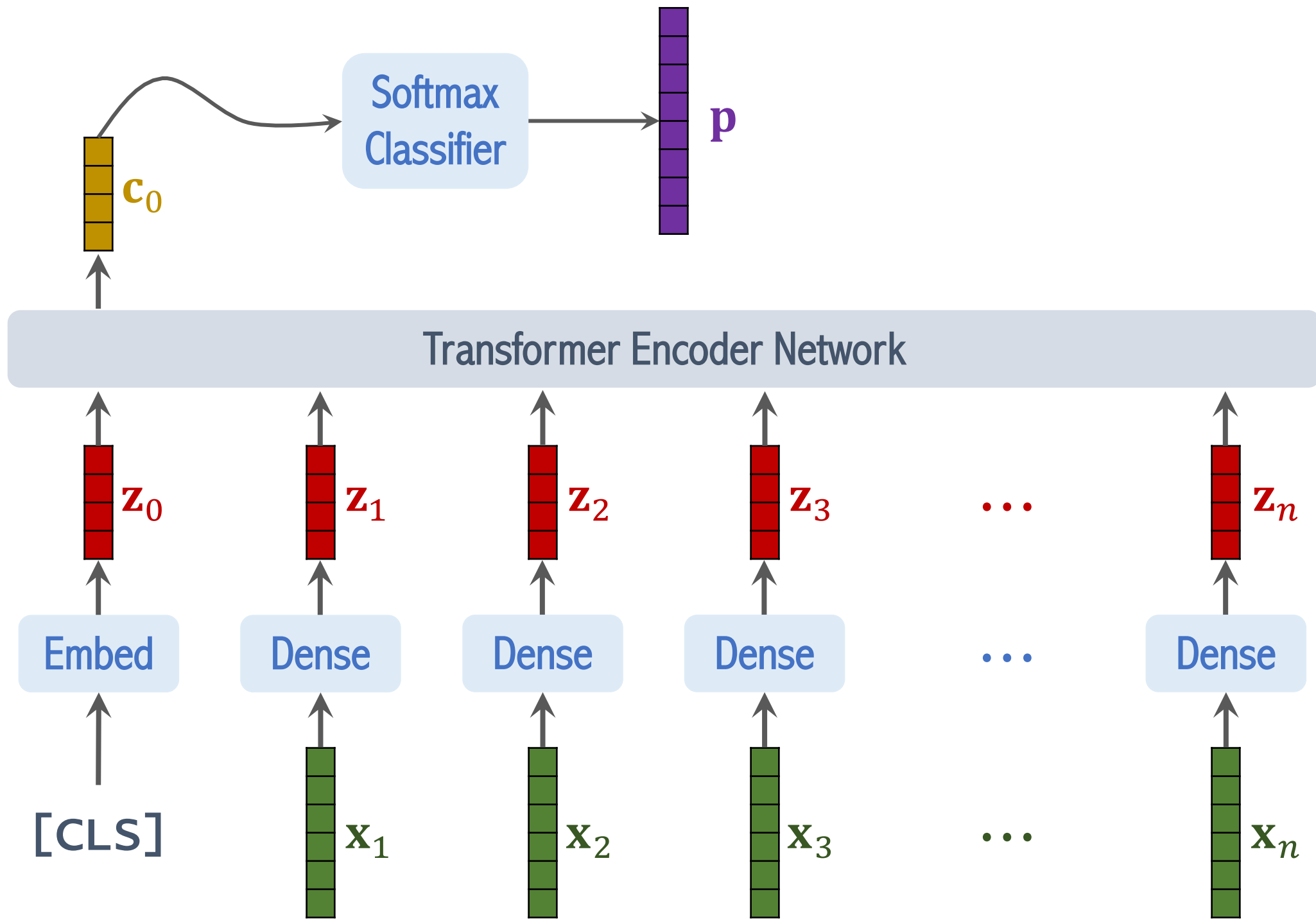


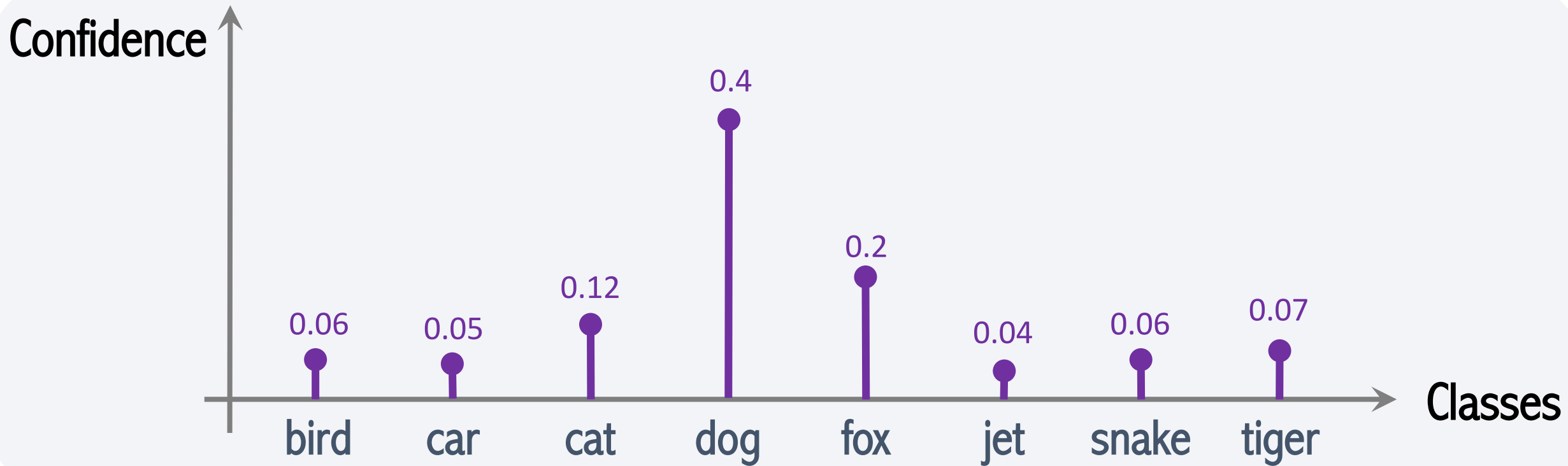
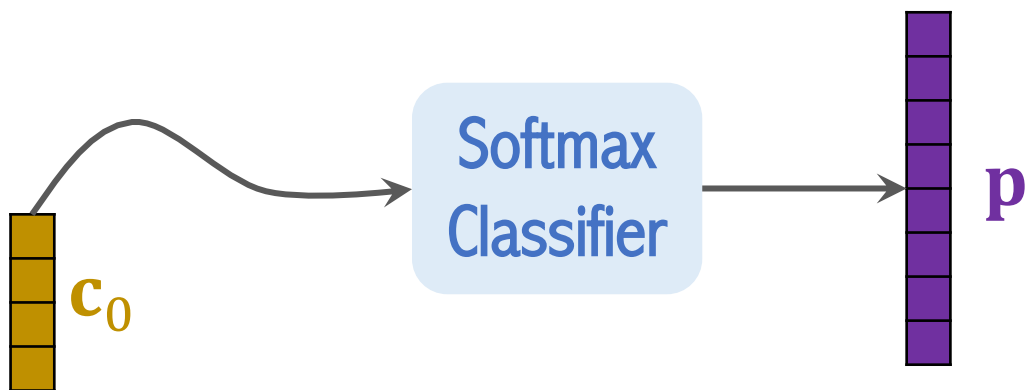




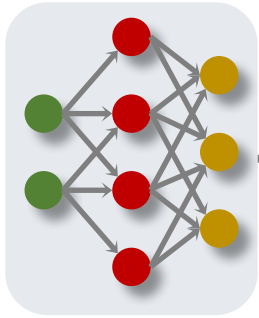




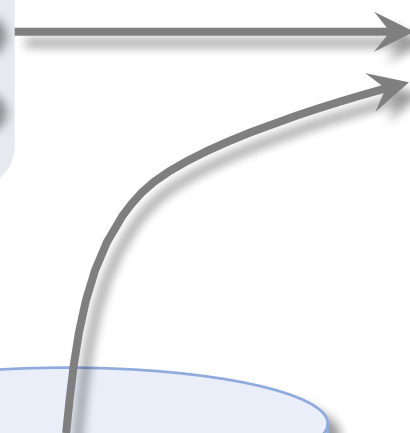
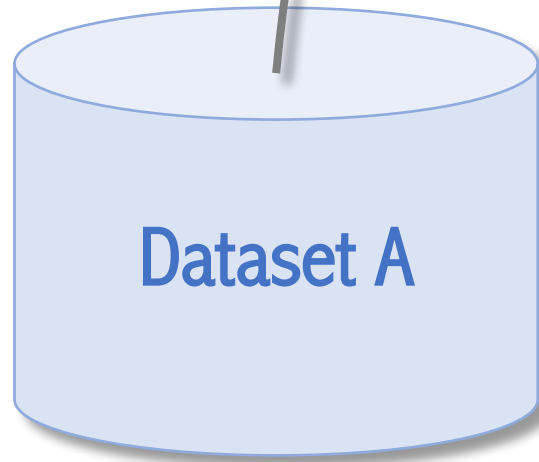
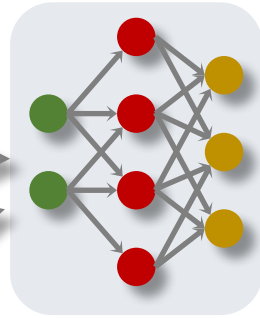




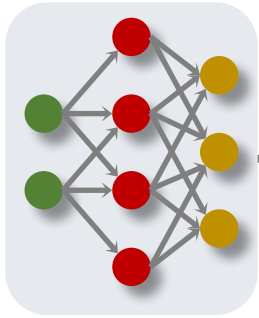
Randomly  
Initialized



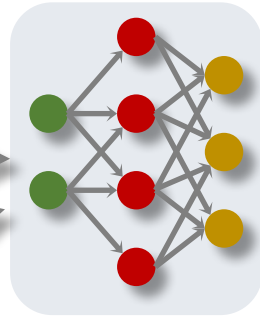
Pretrained



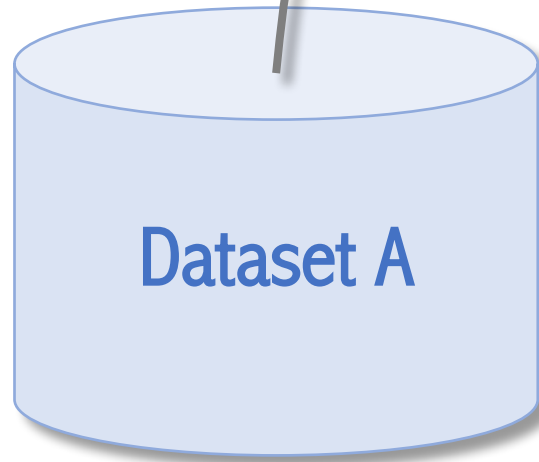
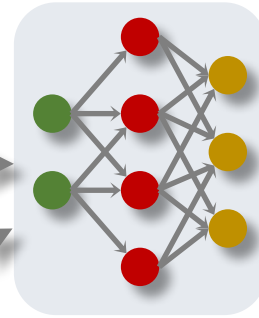
Randomly  
Initialized



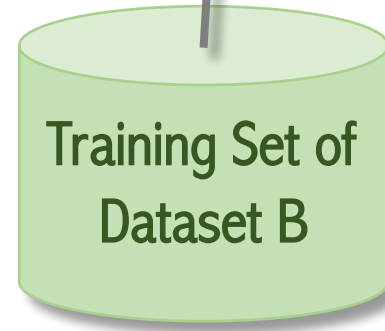
Pretrained



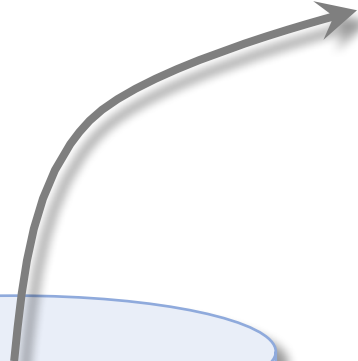
Fine-tuned



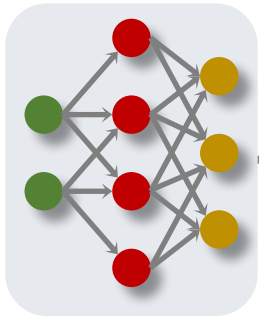
Dataset A



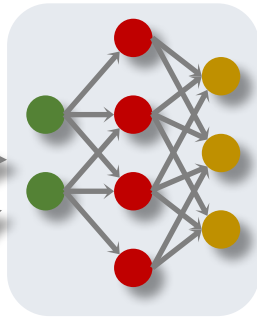
Training Set of  
Dataset B



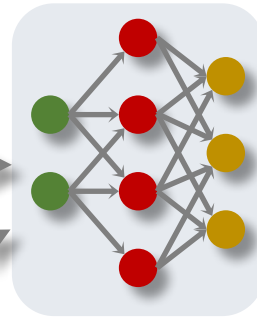
Randomly  
Initialized



Pretrained



Fine-tuned

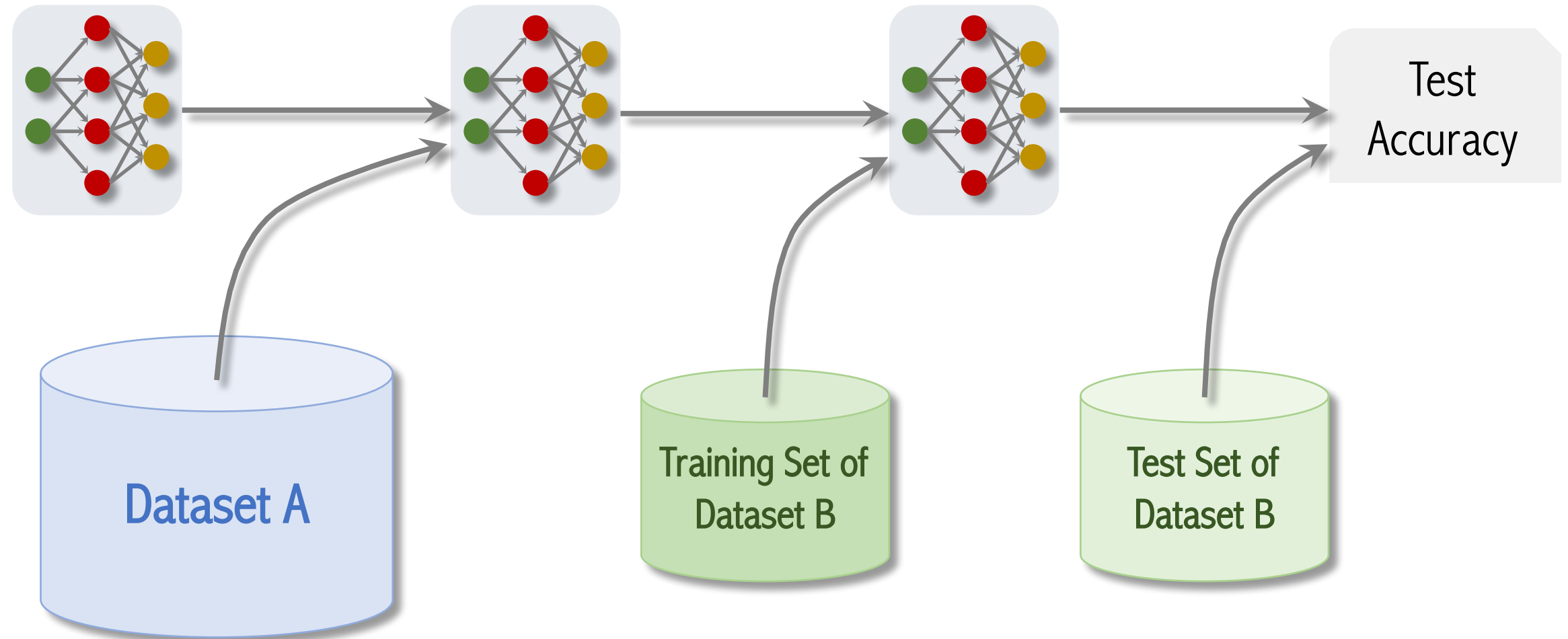


Test  
Accuracy

Dataset A

Training Set of  
Dataset B

Test Set of  
Dataset B



# Datasets

	# of Images	# of Classes
ImageNet (Small)	1.3 Million	1 Thousand
ImageNet-21K (Medium)	14 Million	21 Thousand
JFT (Big)	300 Million	18 Thousand

# Image Classification Accuracies

- Pretrain the model on **Dataset A**, fine-tune the model on **Dataset B**, and evaluate the model on **Dataset B**.
- Pretrained on **ImageNet (small)**, ViT is slightly **worse** than ResNet.
- Pretrained on **ImageNet-21K (medium)**, ViT is **comparable** to ResNet.
- Pretrained on **JFT (large)**, ViT is slightly **better** than ResNet.

# Image Classification Accuracies

ResNet is better

ViT is better

# of Images  
for  
pretraining

100M Images

300M Images

**Thank You!**

<http://wangshusen.github.io/>