

## Appendix

### A More Details

**Backbone.** We use the ImageNet pre-trained EfficientNet-B4 [32]<sup>1</sup> as the backbone. The features from layer1 to layer5 of EfficientNet-B4 [32] have the channel of 24, 32, 56, 160, 448, respectively. Here we define “layer” as the combination of stages that have the same size of features. The 5 features are resized to the same size and concatenated together to form a 720-channel feature map. For MVTEC-AD [4], the image size and the feature size are set as  $512 \times 512$  and  $32 \times 32$ , respectively. Therefore, a feature map with the shape of  $32 \times 32 \times 720$  is obtained. For CIFAR-10 [18], the image size is  $32 \times 32$ , which is quite small. Thus the size of the feature map is set relatively large (with the output stride of 4), so an  $8 \times 8 \times 720$  feature map is obtained.

**Transformer.** A  $1 \times 1$  convolution is applied firstly to the feature map to reduce the channel from 720 to 256. Then the feature map is split to separate feature tokens. For MVTEC-AD [4] and CIFAR-10 [18], there are 1024 and 64 feature tokens with the channel of 256, respectively. The position embedding is a learned embedding with the same size as the input feature tokens.

The transformer encoder follows the standard architecture in [33] with 4 layers. Each layer consists of a multi-head self-attention layer, a feed forward layer, and a shortcut connection with layer normalization. The head number in attention is 8. The architecture of the feed forward layer is shown as follows.

Layer	Input	FC1	Relu	FC2
Output Size	256	1024	1024	256

Besides, the position embedding is added in each self-attention layer rather than only in the first layer to keep more position information.

The transformer decoder also has 4 decoder layers. Each layer is composed of 2 parts, the self-attention part and the cross-attention part. The self-attention part includes a multi-head self-attention layer and a shortcut connection with layer normalization. The cross-attention part consists of a multi-head cross-attention layer, a feed forward layer, and a shortcut connection with layer normalization. The head number in both attention layers is set as 8. The architecture of the feed forward layer is the same as that in the transformer encoder. Also, the position embedding is added in all attention layers. The query embedding is a learned embedding with the same size as the input feature tokens.

The outputs of the transformer have the same size as the inputs ( $1024 \times 256$  for MVTEC-AD,  $64 \times 256$  for CIFAR-10). Then a  $1 \times 1$  convolution is applied to increase the channel from 256 to 720. After reshape, we obtain the reconstructed feature map ( $32 \times 32 \times 720$  for MVTEC-AD,  $8 \times 8 \times 720$  for CIFAR-10).

<sup>1</sup> We use the EfficientNet-B4 checkpoint in <https://github.com/lukemelas/EfficientNet-PyTorch/releases/download/1.0/efficientnet-b4-6ed6700e.pth>