where i represents the index of channel, u is the index of spatial position (height together with width for simplicity). *Anomaly localization* aims to localize anomalous regions, producing an anomaly score map, $\boldsymbol{s}(u)$, which assigns an anomaly score for each pixel, u. $\boldsymbol{s}(u)$ is calculated as the $L2$ norm of the feature difference vector, $\boldsymbol{d}(:,u)$.

$$\boldsymbol{s}(u) = ||\boldsymbol{d}(:,u)||_2. \tag{3}$$

Anomaly detection aims to detect whether an image contains anomalous regions. We intuitively take the maximum value of the averagely pooled $\boldsymbol{s}(u)$ as the anomaly score of the whole image.

3.2 Preventing "Identical Mapping" with Transformer

We suspect that, compared with CNN, the query embedding in attention layer makes transformer difficult to learn an "identical mapping". We denote the features in a normal image as $\boldsymbol{x}^+ \in \mathbb{R}^{K \times C}$, where K is the feature number, C is the channel dimension. The features in an anomalous image are denoted as $\boldsymbol{x}^- \in \mathbb{R}^{K \times C}$. We take a 1-layer network as the reconstruction net, which is trained on $\boldsymbol{x}^+$ with the MSE loss and tested to detect anomalous regions in $\boldsymbol{x}^-$.

Convolutional layer in CNN. We first visit a fully-connected layer, whose weights and bias are denoted as $\boldsymbol{w} \in \mathbb{R}^{C \times C}, \boldsymbol{b} \in \mathbb{R}^{C}$, respectively. When using this layer as the reconstruction model of normal samples, it can be written as,

$$\hat{\boldsymbol{x}} = \boldsymbol{x}^+ \boldsymbol{w} + \boldsymbol{b} \in \mathbb{R}^{K \times C}. \tag{4}$$

With the MSE loss pushing $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}^+$, the model may take shortcut to regress $\boldsymbol{w} \to \boldsymbol{I}$ (identity matrix), $\boldsymbol{b} \to \boldsymbol{0}$. Ultimately, this model could also reconstruct $\boldsymbol{x}^-$ well, failing in anomaly detection. A convolutional layer with 1×1 kernel is equivalent to a fully-connected layer. Besides, An $n \times n$ ($n > 1$) kernel has more parameters and larger capacity, and can complete whatever 1×1 kernel can. Thus, the convolutional layer also has the chance to learn a shortcut.

Transformer with query embedding contains an attention layer with a learnable query embedding, $\boldsymbol{q} \in \mathbb{R}^{K \times C}$. This attention layer can be denoted as,

$$\hat{\boldsymbol{x}} = \texttt{softmax}(\boldsymbol{q}(\boldsymbol{x}^+)^T/\sqrt{C})\boldsymbol{x}^+ \in \mathbb{R}^{K \times C}. \tag{5}$$

To push $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}^+$, the attention map, $\texttt{softmax}(\boldsymbol{q}(\boldsymbol{x}^+)^T/\sqrt{C})$, should approximate $\boldsymbol{I}$ (identity matrix), so $\boldsymbol{q}$ must be highly related to $\boldsymbol{x}^+$. Considering that $\boldsymbol{q}$ in the trained model is relevant to normal samples, the model could not reconstruct $\boldsymbol{x}^-$ well. The ablation study in Sec. 4.4 shows that without the attention layer or the query embedding, the performance of transformer respectively drops by 2.4% or 3%, which is almost the same as CNN. This reflects that the query embedding in attention layer helps prevent transformer from learning an "identical shortcut".

3.3 Adaptation with Anomaly-available Case

In practice, anomalies gradually increase with the runs of production lines, which brings the demands of compatibility with these increasing anomalies. Thus we adapt ADTR to ADTR+ for compatibility with the anomaly-available case.