Appendix A
Implementation Details of U-block

The U-block in our proposed LTwIST handles the mapping problem of nonlinear transformations. It is also formulated as $U_{main}(\cdot)$ function to represent the circulation of data for convenience. Specifically, Fig. 2 describes the details of the U-block architecture, which is designed as a U-shaped convolutional structure (a variant of U-net [50]) combining convolution and pooling operations to provide powerful representation capabilities.

As shown in Fig. 2, different colored cubes depict the state of the feature map after various operations. The blue cubes represent the feature map resulting from a series of convolutions, while the green cubes indicate the feature map following the pooling layer. Furthermore, the light yellow cubes correspond to a sequence of upsampling and deconvolutional layers serving as counterparts to the encoder. These layers play a vital role in progressively restoring resolution and reconstructing the image. Additionally, the purple cubes signify the skip connections, establishing links between the encoder's feature map and the decoder's feature map, thereby retaining more contextual information. Employing distinct colors to denote different types of layers and operations enhances the intuitive visualization of the U-block architecture, facilitating a clearer understanding and more insightful analysis of its working principles.

The U-block takes an image patch with dimension $B \times B \times 1$ as input and applies two layers of convolution operator and max pooling operation repeatedly. Each convolution layer corresponds to its own 3×3 N_f filter (N_f is set to 32 in the experiments), followed by batch normalization (BatchNorm) and a rectified linear unit (ReLU). By default, the two layers of convolution mentioned later include BatchNorm and ReLU. The maximum pooling operation has a stride of 2 for down-sampling, and the number of feature channels is doubled in each down-sampling step. Moreover, the U-block has seven double-layer convolutions and six pooling operations totally. In particular, the U-block U_{main} can be decomposed into three equations:

$$\begin{cases} \mathbf{x}_{down}^{k} = F_{3r}(F_{mp}(F_{2*conv}(\mathbf{x}_{g}^{k}))), \\ \mathbf{x}_{center}^{k} = F_{2*conv}(\mathbf{x}_{down}^{k}), \\ \mathbf{x}_{up}^{k} = F_{3r}(F_{2*conv}(F_{cat}(F_{up}(\mathbf{x}_{center}^{k})))), \end{cases}$$

where $\mathbf{x}_{down}^{k}$, $\mathbf{x}_{center}^{k}$ and $\mathbf{x}_{up}^{k}$ are the intermediate outputs of the k^{th} stage, $F_{2*conv}(\cdot)$ represents two complete convolution operations, $F_{mp}(\cdot)$ denotes the max pooling process, $F_{cat}(\cdot)$ is the concatenating function, F_{up} represents the upward transposed convolution, and $F_{3r}(\cdot)$ stands for repeating the processing in brackets three times in sequence.

The final $\mathbf{x}_{down}^{k}$ is acquired by 3 repetitions of two layers of convolution and pooling operations. Moreover, the lowest $\mathbf{x}_{center}^{k}$ in the U-block is obtained by two convolutions on $\mathbf{x}_{down}^{k}$. Then we perform up-sampling of the feature map, halve the number of feature channels, concatenate together with the corresponding feature map on the left, and then perform two layers of 3×3 transposed convolution. The latest $\mathbf{x}_{up}^{k}$ is obtained after repeating the process three times. Finally, the last layer uses 1×1 convolution to map the dimension to 1 to obtain the output of U-block with dimensions $B \times B \times 1$. The dimension change of the whole U-block is 1→32→64→128→256→128→64→32→1.

The designed U-block architecture can incorporate multi-dimensional scale features, which further improves the abstract feature representation capability of pure convolution, and achieves better reconstruction quality.