A Dataset

ImageNet-mini [11] is the dataset we used for joint training, which has 100 classes and 60000 images, where 38400 images are selected as the training set. We artificially add noise to the images. After adding noise, if the value of pixels is above 1 or below 0, the value is clipped to 1 or 0 respectively. It should be noted that we generate the noise on the test samples with a unique seed, but the noise for the training samples is totally random, that means the noise of each training sample varies at each epoch.

For test-time adaptation, in addition to ImageNet-mini, we adopt CT [31], fastMRI [44] , SIDD-small [1], DND [28], PolyU [41] and FMDD [46]to test our method on different datasets.

CT dataset involves 2D breast CT simulations, where the simulated breast phantom has random fibro-glandular structure and high-contrast specks. The phantom allows for arbitrarily large training sets to be generated with perfectly known ground truth. The data set consists of ground truth images, 128-view sinogram data, and the corresponding 128-view filtered back-projection (FBP) image. We only use the ground truth data as the clean images, and manually add Gaussian noise (var=0.005) on them.

fastMRI is an MRI image dataset. We used the images obtained through root-sum-of-squares reconstruction of the corresponding multi-coil k-space data as the clean images. We manually add Gaussian noise with standard deviation of 0.005 (pixel values range from 0 and 1) to get the noisy images. It should be noted that fastMRI consists of grayscale images, we expand the number of channels of these images to 3 before inputting them into the denoising model.

SIDD-small is the Smartphone Image Denoising Dataset. The noise on images is naturally generated by optical sensors, and clean images are obtained by averaging multiple shots. Since the images in SIDD are high resolution (4K), we split each image into several 448×448 patches.

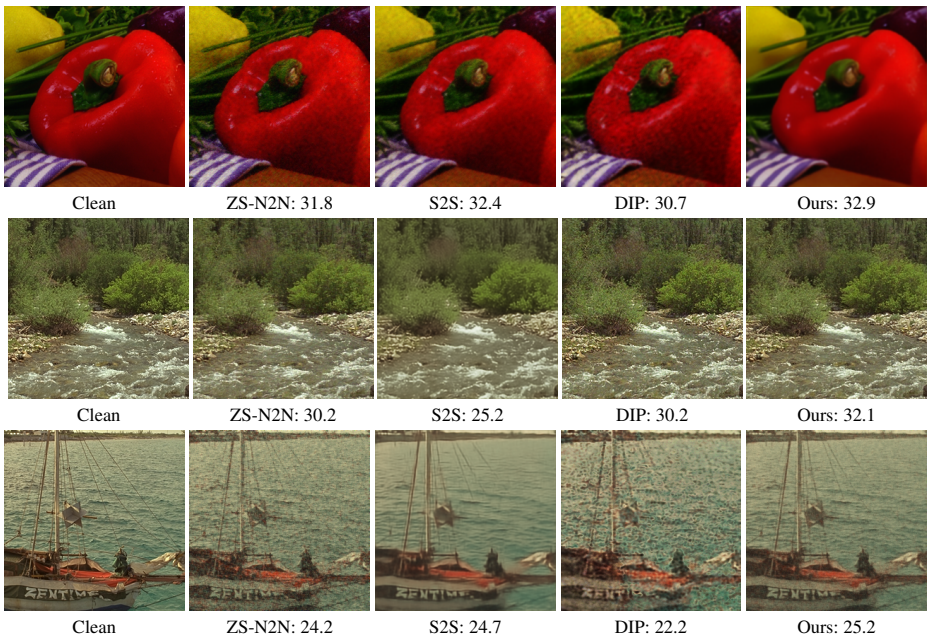
PolyU is PolyU-Real-World-Noisy-Images-Dataset, which is a natural noise dataset, where the images are captured by different cameras, *i.e.*, Canon (Mark 5D, 80D, 600D), Nikon (D800), and Sony (A7 II). We adopt the cropped images in their GitHub repository (https://github.com/csjunxu/PolyU-Real-World-Noisy-Images-Dataset). The size of each image is 512×512.

DND is Darmstadt Noise Dataset, which is a natural noise dataset. The authors capture pairs of images with different ISO values and appropriately adjust exposure times, where the nearly noise-free low-ISO image serves as the reference. To derive the ground truth, they also correct spatial misalignment, cope with inaccuracies in the exposure parameters through a linear intensity transform based on a novel heteroscedastic regression model, and remove residual low-frequency bias that stems, e.g., from minor illumination changes [28]. They split the images into 512×512 patches. The ground truth images are not available. For testing purposes, the denoised images must be sub-

mitted to their online submission system to see the results.

FMDD is Fluorescence Microscopy Denoising Dataset. This dataset is constructed from real noisy fluorescence microscopy images and designed for Poisson-Gaussian denoising purposes [46]. We use the images of noise level 1 (raw) as the noisy images. The images in FMDD are grayscale images, so we also expand the number of channels of these images to 3.

B Samples



C Implementation Details

Noise2Self loss The Noise2Self (N2S) loss [3] is a standard blind-spot self-supervised denoising loss, that does not require the noise distribution in advance. The N2S loss leverages the difference between the value of masked pixels and the average value of their neighboring pixels to indirectly learn the residual between noisy and clean images.

Similar to MIM, it also uses a mask matrix to corrupt the image, but it replaces the masked pixels with the average of their neighbors instead of learnable tokens. Different

from MIM, the mask is not random and patch-wise but uniform and grid-wise. Moreover N2S performs the masking operation before the projection as opposed to after the projection as in MIM. The calculation of the average of the neighbors for pixel $i_{x,y}$ is defined as:

$$i'_{x,y} = \frac{1}{6}(0.5i_{x-1,y-1} + i_{x,y-1} + 0.5i_{x+1,y-1} + i_{x-1,y} + i_{x+1,y} + 0.5i_{x-1,y+1} + i_{x,y+1} + 0.5i_{x+1,y+1}), \tag{7}$$

where (x, y) denotes the position of the pixel on the image. In practice, it can be easily implemented by a convolution operation. Given a fixed kernel

$$\mathbf{K} = \frac{1}{6}[[0.5, 1, 0.5], [1, 0, 1], [0.5, 1, 0.5]], \tag{8}$$

and a binary mask matrix $\mathbf{M}' \in \mathbb{R}^{H \times W \times C}$, the masked image $\mathbf{I}'$ is:

$$\mathbf{I}' = (\mathbb{1} - \mathbf{M}') \odot \mathbf{I} + \mathbf{M}' \odot (\mathbf{I} * \mathbf{K}), \tag{9}$$

where $\odot$ denotes the element-wise multiplication, and $*$ denotes the convolution operation. Noting that, $\mathbf{I}'$ is then fed into the UNet as the input, the output $\mathbf{O}$ is the residual between the masked image and original image, and the loss calculation is defined as:

$$\mathcal{L}_{N2S} = ||\mathbf{M}' \odot [\mathbf{O} - (\mathbf{I}' - \mathbf{I})]||_1. \tag{10}$$

Joint Training We pretrain the UNet for 100 epochs. The batch size is 80. The optimizer is AdamW [19], [23], where beta1 and beta2 are 0.9 and 0.99, respectively. The initial learning rate is 1e-3, the weight decay rate is 1e-4. A cosine decay learning rate scheduling with linear warmup is used, where the last learning rate is 1e-6, and the number of warmup epochs is 5. The mask ratio is 0.6 and the mask patch size is 7. For augmentation, we randomly resize and crop the images into 224×224 and randomly horizontally flip them.

Model The architecture of the UNet implemented in this work is illustrated in Figure C.1. In contrast to the original UNet, we add a projection layer before the encoder to make the model compatible to SimMIM, and add an auxiliary head to reconstruct the masked patches. We replace the default batch normalization [18] with group normalization [38] that requires no batch statistics to avoid dependency on the batch size.

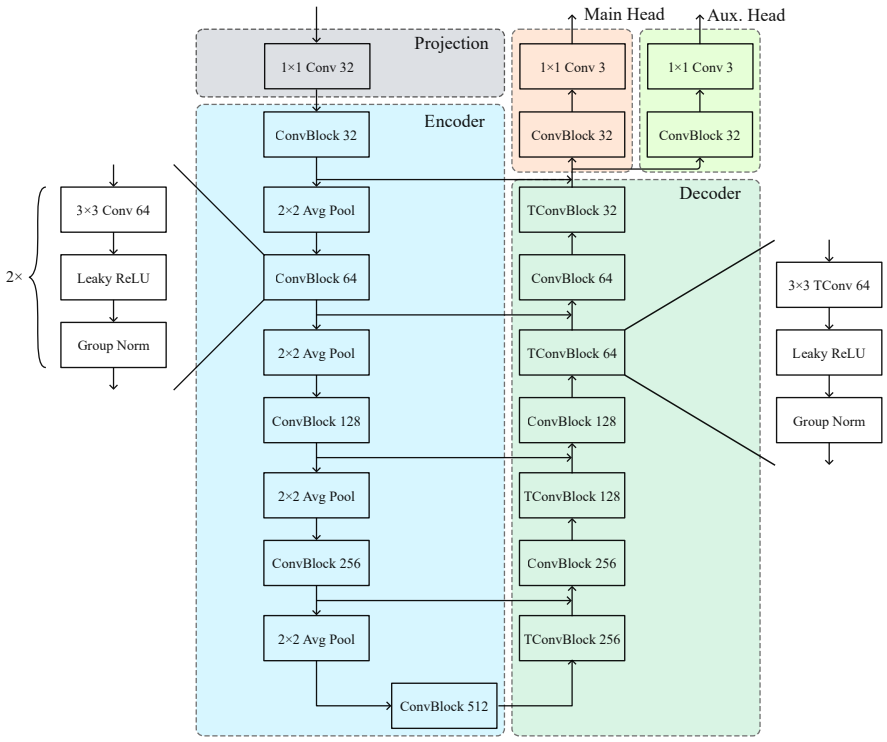


Fig. C.1: Architecture of the UNet implemented in this work. "3×3 Conv 64" denotes that the kernel size of the conv layer is 3×3 and the number of output channels is 64. TConv denotes transpose conv layer. The negative slope of Leaky ReLU is 0.2, and the number of groups of Group Norm is #channels/16.