Supplemental Materials

0.1 Kernel Sparsity ($\mathbf{s_c}$)

During the ConvNet backpropagation, the update of the 2D kernels, $W_{n,c}$ (where n and c are, respectively, the indices of the output and input feature maps), is given by:

$$W_{n,c}^{(t+1)} = W_{n,c}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial Y_n^{(t)}} X_c^{(t)} - \frac{\partial \mathcal{R}(W_{n,c}^{(t)})}{\partial W_{n,c}^{(t)}}, \tag{1}$$

where X_c and Y_n represent, respectively, the input and output feature maps of the convolutional layer, η denotes the learning rate, and $\mathcal{L}$ and $\mathcal{R}$ denote the loss function and weight regularization, respectively. A sparse input feature map, X_c^t, may result in a sparse weight, $W_{n,c}^{(t+1)}$, during training. This is because the sparse feature map yields a small weight update The kernel sparsity for the c^{th} input feature map is defined as:

$$s_c = \sum_{n=1}^{N} |W_{n,c}|, \tag{2}$$

where N denotes the total number output feature maps.

0.2 Kernel Entropy ($\mathbf{e_c}$)

Kernel entropy is built on the fact that the diversity of the input feature maps is directly related to that of the corresponding convolution kernels. To determine the diversity of the kernels, a nearest neighbor distance matrix, A_c, is first computed for the c^{th} convolution kernel. The value in the i^{th} row and j^{th} column of A_c is assigned to be:

$$A_{c_{i,j}} = \begin{cases} \|W_{i,c} - W_{j,c}\|, & \text{if } W_{j,c} \in \{W_{i,c}\}_k \\ 0, & \text{otherwise} \end{cases}$$

where $\{W_{i,c}\}_k$ represents the k-nearest-neighbor of $W_{i,c}$. Then, a density metric is calculated for $W_{i,c}$, which is defined as:

$$dm(W_{i,c}) = \sum_{j=1}^{N} A_{c_{i,j}}. \tag{3}$$

If $dm(W_{i,c})$ is large, then the convolution kernel is more different from the its neighbors, and vice versa. The kernel entropy is calculated as the entropy of the density metrics:

$$e_c = -\sum_{i=1}^{N} \frac{dm(W_{i,c})}{\sum_{i=1}^{N} dm(W_{i,c})} \log_2 \frac{dm(W_{i,c})}{\sum_{i=1}^{N} dm(W_{i,c})}. \tag{4}$$

A small e_c indicates diverse convolution kernels. Thus, the corresponding input feature map provides more information to the ConvNet.

0.3 Kernel Sparsity & Entropy (KSE)

KSE is then defined as:

$$KSE = \sqrt{\frac{s_c}{1 + \alpha e_c}}, \tag{5}$$

where KSE, s_c, and e_c are normalized into [0, 1], and α is a parameter for controlling weight between s_c and e_c, which is set to 1.

0.4 Evaluation Metrics

For quantitative evaluation of the denoised v2 whole-body scans, the ensemble bias in the mean standard uptake value (SUV) of the simulated tumor that was inserted in a real patient background, and the liver coefficient of variation (CoV) were calculated from 10 noise realizations. The ensemble bias is formulated as:

$$\text{BIAS}(\%) = \frac{\frac{1}{R}\sum_r^R \mu_r^L - T^L}{T^L} \times 100, \tag{6}$$

where μ_r^R denotes the average counts within the lesion L of the r^{th} noise realization, and T^L represents the "true" (from high quality PET scan) intensity value within the lesion.

The liver CoV was computed as:

$$\text{CoV}(\%) = \frac{\frac{1}{N}\sum_{i \in B} \sigma_j^R}{\bar{\mu}_B} \times 100, \tag{7}$$

where σ_j^R denotes the ensemble standard deviation of j^{th} voxel across R ($R = 10$) realizations, N is the total number of voxels in the background volume-of-interest (VOI) B. The liver CoV is computed within a hand-drawn 3D VOI within the liver.

0.5 Comparisons of The Training Time

Method	Img. Recon.	Network Training	Total Time	Percent Time Saved
v1/v2-net	1 $\frac{\text{wk.}}{\text{Pt.}}$ $\times$ 20 Pts.	5.5 days	20.8 wks.	-
FT/TGD-net	1 $\frac{\text{wk.}}{\text{Pt.}}$ $\times$ 7 Pts.	2.5 days	7.4 wks.	64%

Table 1: Comparisons of data preparation and network training time used between the proposed method and training-from-scratch. "FT" denotes "fine-tuning", and "Wk." and "Pt." stand for, respectively, "week" and "patient". To form a complete dataset, it required approximately a week to reconstruct training pairs of noisy inputs and target for each patient (1 target + 6 different count levels), and a total of 20 patients were used for training v1-net and v2-net.

0.6 Additional Results

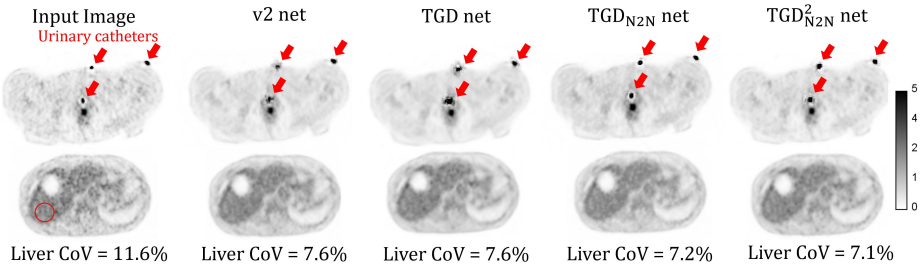

Fig. 1: Comparisons of proposed TGD-N2N and other methods on a FDG-PET patient study which had urinary catheters attached during the scan (denoted by the arrows). The first and second rows show the trans-axial slices of the urinary catheters and liver of the same study, respectively. All the images are displayed in the same inverted grey scale. This study was acquired for 120-sec, which was rebinned into 2 noise samples with equal count levels (60-sec) for the TGD-N2N training. Both TGD-N2N networks were retrained for 150 epoch. All the networks were then applied to the 120-sec scan (input image) to generate the denoised results. The out-of-distribution objects (catheters) led to artifacts in both v2-net and TGD-net results. The online learning approaches using TGD$_{N2N}$-net$^{\phi=0.36}$ and TGD$^2_{N2N}$-net$^{\phi=0.3,0.4}$ alleviated the artifacts while retaining similar denoising performance in terms of liver Coefficient-of-Variations (CoV) in the ROI denoted by the red circle. The KSE threshold for both TGD-N2N results were adjusted to achieve similar liver CoV for a fair comparison.