# Learning to Distill Global Representation for Sparse-View CT —Supplementary Material—


## Abstract

*This Supplementary Material includes four parts: (1) more ablation study and analysis, (2) efficiency, (3) more visualization results, and (4) network architecture.*


## 1. More Ablation Study and Analysis

### 1.1. Ablation on Framework Design

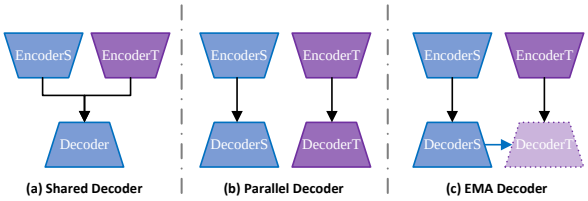


Figure S1. Framework designs of GloReDi: (a) GloReDi-S: shared decoder for student and teacher networks, which is optimized twice per iteration; (b) GloReDi-P: separate parallel decoders for student and teacher; and (c) GloReDi-E: teacher decoder is updated via EMA according to student decoder. For simplicity, we name GloReDi-E trained without distillation loss as GloReDi-N for comparison.

Table S1 presents the experimental results of different framework designs employed in GloReDi, as illustrated in Fig. S1. We found that the parallel decoder design failed to align the representations of different views into a shared latent space, thus, leading to suboptimal results. The result of GloReDi-P is even worse than GloReDi-N, revealing that the domain gap between various views can harm the training. Although the utilization of a shared decoder can enforce the representation to be in the same latent space and outperform GloReDi-P, it could also introduce unstable problems during training, for the parameters of the shared decoder are updated twice in an iteration. By contrast, GloReDi-E achieves the alignment of representations in a shared latent space through an exponential moving average (EMA) update procedure, thereby circumventing interference with student training. The resultant teacher encoder can be considered a stable version of the student encoder, making the teacher GloRe a dependable distillation target. Therefore, we choose GloReDi-E as our final framework design.

| GloReDi | -N | -P | -S | -E |
|---|---|---|---|---|
| PSNR | 37.91 | 37.85 | 38.04 | **38.38** |

Table S1. PSNR evaluation of GloReDi with different framework designs. All networks are trained under $N_{\text{v}} = 18$ for 60 epochs.

### 1.2. Ablation on Configurations of Residual Blocks

| config. | e5d4 | e6d3 | e7d2 | e8d1 |
|---|---|---|---|---|
| $N_{\text{v}} = 18$ | 37.49 | 37.62 | **38.06** | 38.02 |
| $N_{\text{v}} = 72$ | 43.75 | 43.90 | **44.39** | 44.08 |

Table S2. PSNR evaluation of GloReDi with varied numbers of FFC residual blocks in the encoder and decoder. (*e.g.*, e5d4 represents 5 and 4 FFC residual blocks in the encoder and decoder, respectively). All models are trained for 40 epochs considering the computational cost.

Given fixed parameters, a larger encoder can improve the information extraction and recovery process and better bridge the domain gap between the sparse- and denser-view images. In the meantime, a larger decoder can better decode the global representation and improve the reconstruction quality. Table S2 presents the quantitative results of varying numbers of FFC residual blocks in the encoder and decoder. The results suggest that a ratio of $7:2$ for 9 residual blocks in the encoder and decoder is the most favorable for distillation.

### 1.3. Ablation on Distillation Loss

| config. | $\ell_1$ loss | $\ell_2$ loss | (ours) |
|---|---|---|---|
| $N_{\text{v}} = 18$ | 37.44 | 37.29 | **38.06** |
| $N_{\text{v}} = 72$ | 42.88 | 42.56 | **44.39** |

Table S3. PSNR evaluation of GloReDi trained with different distillation loss, including $\ell_1$ loss and $\ell_2$ loss commonly used in knowledge distillation, as well as the proposed one with $\mathcal{L}_{\text{rdd}}$ and $\mathcal{L}_{\text{bcd}}$. All models are trained for 40 epochs considering the computational cost.

Table S3 exhibits the results of GloReDi trained with different distillation loss. Our findings suggest that pixel-wise distillation losses, such as $\ell_1$ and $\ell_2$ loss, are not as

effective as the proposed one. This is attributed to the fact that conventional distillation tasks involve both the student and teacher networks sharing the same input and ground truth. Consequently, the domain gap does not affect them. However, for sparse-view CT reconstruction, it is arduous for the student to recover the missing information entirely. This renders pixel-wise losses too abrupt for distillation purposes.

### 1.4. Ablation on Band-pass-specific Contrastive Distillation

We have demonstrated the effective components by training GloReDi with $\mathcal{L}_{\text{bcd}}$ on specific frequency components. However, various methods exist to split the frequency components [1–4]. Note that in 2D discrete cosine transform, low-frequency components are placed on the upper left. We define the mask $\boldsymbol{M} \in \{0,1\}^{N_\text{w} \times N_\text{h}}$ as follows to select the target components:

$$\boldsymbol{M}_{i,j} = \begin{cases} 1, \text{if } i \in [b_{\text{low}} N_\text{w}, b_{\text{up}} N_\text{w}] \text{ and } j \in [b_{\text{low}} N_\text{h}, b_{\text{up}} N_\text{h}] \\ 0, \text{otherwise,} \end{cases} \tag{1}$$

where $\boldsymbol{M}_{i,j}$ is the element in $\boldsymbol{M}$ at position $(i, j)$; $b_{\text{low}}$ and $b_{\text{up}}$ denote the hand-craft ratios defining the lower and upper bounds, respectively, which range from $[0, 1]$. We then split the DCT spectrum into five groups, demarcated by the intervals $[b_{\text{low}}, b_{\text{up}}]$, as illustrated in Fig. S2. Notably, the model distilled via the vanilla supervised contrastive loss served as the baseline for our comparative analysis and was denoted by the black horizontal line in Fig. S2. Obviously, models trained with frequency components, except for the lowest and highest, perform better than vanilla ones, demonstrating that selecting band-pass components is effective. In addition, middle groups perform relatively better among different groups, demonstrating the effectiveness of the selected band-pass-specific components. Therefore, we select $[b_{\text{low}}, b_{\text{up}}] = [0.2, 0.5]$ to train our final models to balance the performance and memory usage.

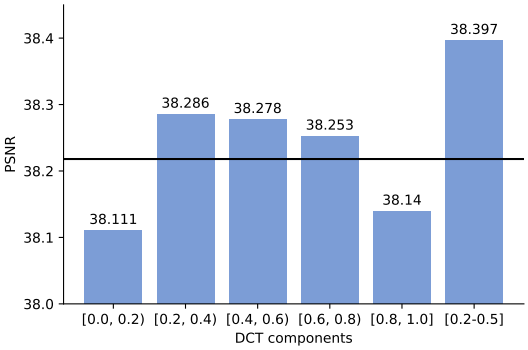


Figure S2. The effect of different frequency components. Note that the black horizontal line represents the contrastive distillation without projecting the representation to the DCT domain. The models are trained with $\mathcal{L}_{\text{bcd}}$ only for 60 epochs.

## 2. Efficiency

| Methods | DDNet | FBPConvNet | DuDoNet | DDPTrans | DuDoTrans | GloReDi |
|---|---|---|---|---|---|---|
| mem. (MB) | 86.4 | 274.9 | 2150.1 | 7220.3 | 3108.5 | 798.8 |
| infer. (ms) | 14.7 | 11.7 | 49.6 | 71.3 | 78.4 | 33.1 |

Table S4. Peak memory usage and mean inference time on a single RTX 3090 GPU using 1000 images, with a batch size of 1, at a resolution of $256 \times 256$.

Table S4 presents the peak memory usage (mem.) and mean inference time (infer.) assessed on a single RTX 3090 GPU with a batch size of 1, averaging over 1000 images at a resolution of $256 \times 256$. Overall, dual-domain methods exhibit lower efficiency compared to image post-processing techniques. Transformer-based methods are suboptimal in both memory usage and inference time to those built with CNN. In contrast, GloReDi demonstrates comparable performance to other post-processing methods while achieving higher efficiency than dual-domain approaches by eliminating the need for the teacher network during inference.

## 3. More Visualization Results

Fig. S3 presents the visualization results of six groups of sparse-view images. Among all the methods, GloReDi better recovers the clinical details such as the lung trachea in the first row, the round soft tissue in the second row, and the clear boundary highlighted in the fifth row.

Fig. S4 shows another four images in the AAPM dataset. We note that in ultra-sparse scenarios when $N_\text{V} = 18, 36$ as shown in the first and the second rows, only GloReDi precisely reconstructs the structure highlighted by the blue box. When $N_\text{V} = 72$, GloReDi achieves competitive performance compared with DuDoTrans but without using the sinogram data.

## 4. Network Architecture

Tables S5 and S6 show the detailed network architecture of the encoder and decoder, respectively.

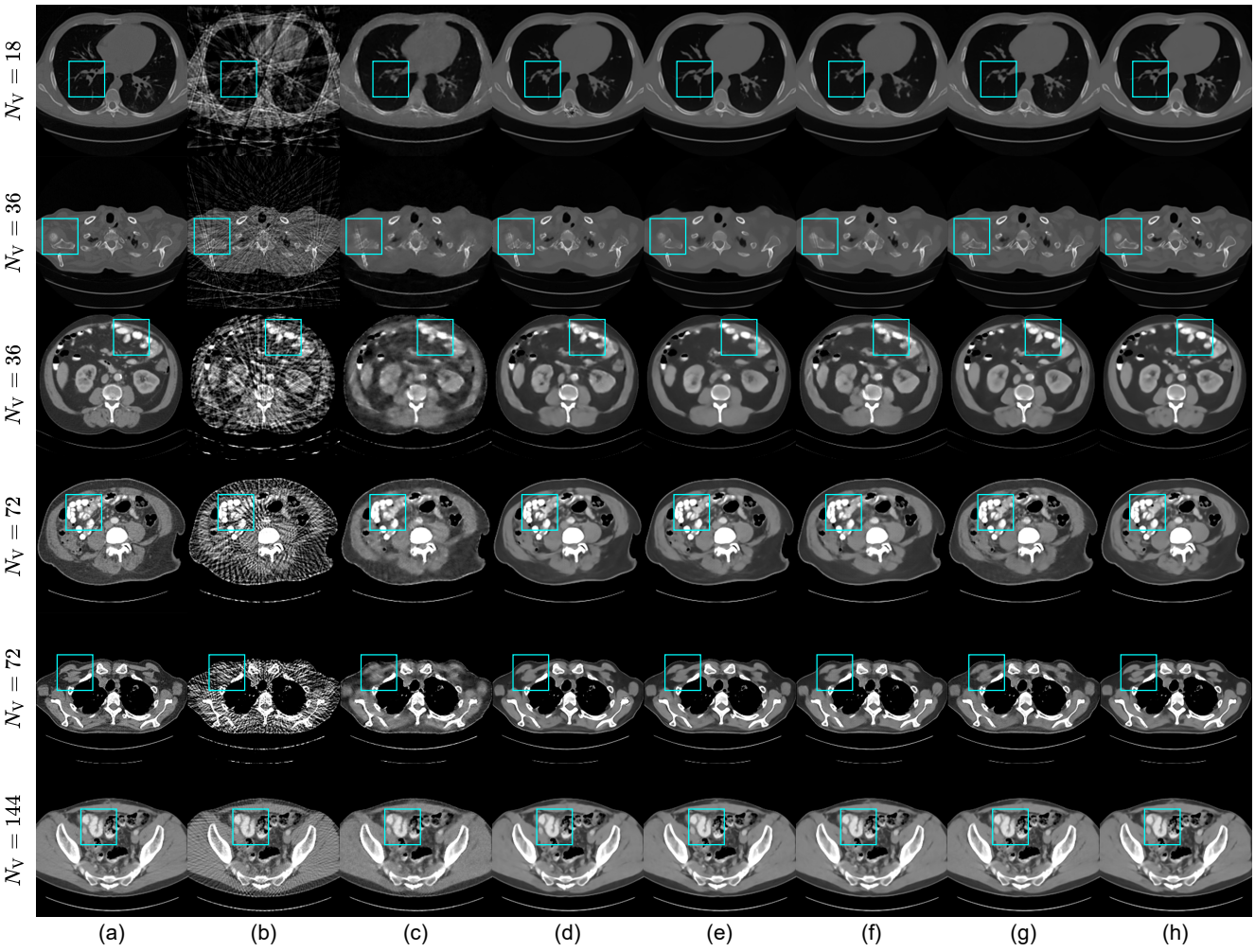


Figure S3. Visual comparison of state-of-the-art methods on DeepLesion dataset: (a) Ground Truth, (b) FBP, (c) DDNet, (d) FBPConvNet, (e) DuDoNet, (f) DDPTrans, (g) DuDoTrans, and (h) GloReDi. From top to bottom: the results under $N_v = 18, 36, 36, 72, 72, 144$; display window is set to [-1000, 2000] HU for the first and the second rows, and [-200, 300] HU for the rests.

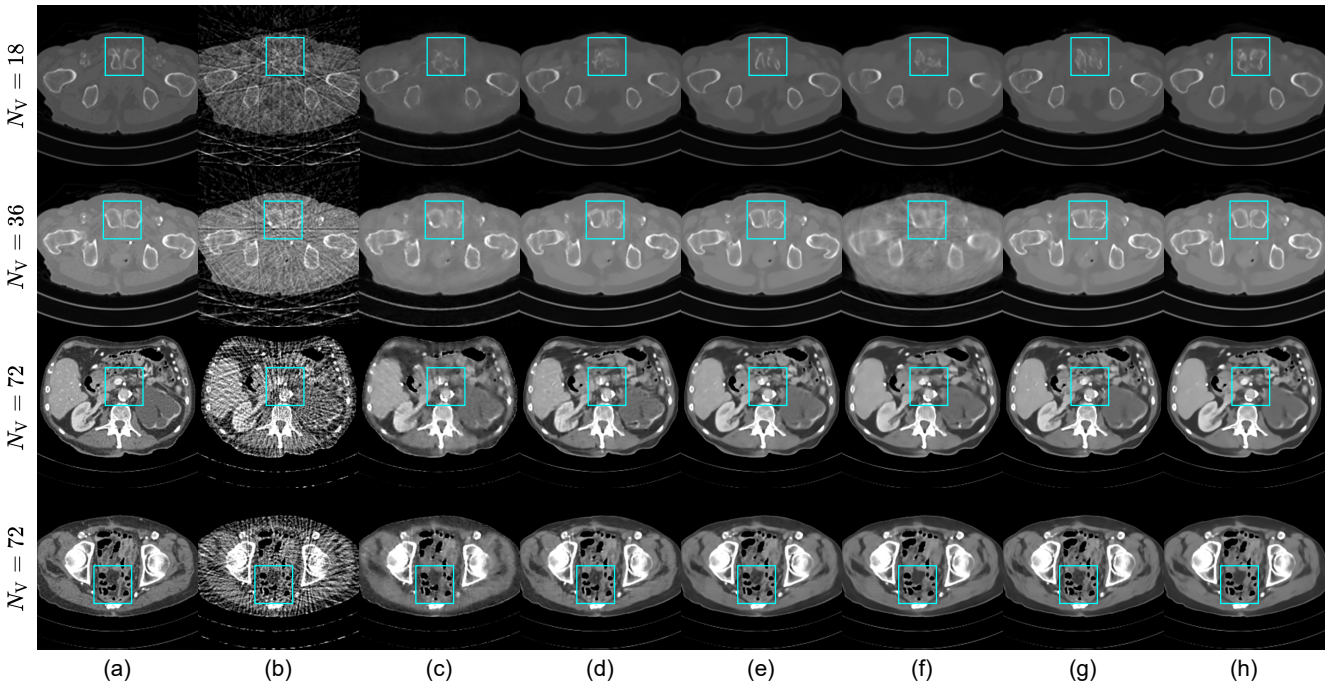


Figure S4. Visual comparison of state-of-the-art methods on AAPM dataset: (a) Ground Truth, (b) FBP, (c) DDNet, (d) FBPConvNet, (e) DuDoNet, (f) DDPTrans, (g) DuDoTrans, and (h) GloReDi. From top to bottom: the results under $N_{\mathrm{v}} = 18, 36, 72, 72$; display window is set to [-1000, 2000] HU for the first row, [-1000, 1000]HU for the second row and [-200, 300] HU for the third and fourth row.

| Name | Channels | Description |
|---|---|---|
| Input | 2 | sparse-view images $\boldsymbol{I}_{\mathrm{S}}$ or $\boldsymbol{I}_{\mathrm{T}}$ |
| Rpad0 | 2 | reflectionpad2d((3,3,3,3)) |
| Down1 | 64 | K7C64S1P1-BN-ReLU |
| Down2 | 128 | K3C128S2P1-BN-ReLU |
| **FFC-Split** $\times m$ | 256 | local branch: K3C64S2P1-BN-ReLU<br>global branch: K3C192S2P1-BN-ReLU |
| **FFC-1** $\times m$ | 256 | convl2l: K3C64S1P1<br>convl2g: K3C192S1P1<br>convg2l: K3C64S1P1<br>convg2g: K1C96S1-bn-relu-FFT-K1C192S1-iFFT-K1C192S1<br>local branch: BN-ReLU<br>global branch: BN-ReLU |
| **FFC-2** $\times m$ | 256 | convl2l: K3C64S1P1<br>convl2g: K3C192S1P1<br>convg2l: K3C64S1P1<br>convg2g: K1C96S1-bn-relu-FFT-K1C192S1-iFFT-K1C192S1<br>local branch: BN-ReLU<br>global branch: BN-ReLU |
| **FFC-Cat** $\times m$ | 256 | concat(local branch, global branch) w/ residual learning |

Table S5. Network architecture of student and teacher encoder. We use '`K-C-S-P`' to denote the kernel, channel, stride, and padding configuration of convolution layers.

| Name | Channels | Description |
|---|---|---|
| **FFC-Split** $\times n$ | 256 | local branch: K3C64S2P1-BN-ReLU<br>global branch: K3C192S2P1-BN-ReLU |
| **FFC-1** $\times n$ | 256 | convl2l: K3C64S1P1<br>convl2g: K3C192S1P1<br>convg2l: K3C64S1P1<br>convg2g: K1C96S1-bn-relu-FFT-K1C192S1-iFFT-K1C192S1<br>local branch: BN-ReLU<br>global branch: BN-ReLU |
| **FFC-2** $\times n$ | 256 | convl2l: K3C64S1P1<br>convl2g: K3C192S1P1<br>convg2l: K3C64S1P1<br>convg2g: K1C96S1-bn-relu-FFT-K1C192S1-iFFT-K1C192S1<br>local branch: BN-ReLU<br>global branch: BN-ReLU |
| **FFC-Cat** $\times n$ | 256 | concat(local branch, global branch) w/ residual learning |
| Up1 | 128 | ConvTranspose2d: K3C128S2P1-BN-ReLU |
| Up2 | 64 | ConvTranspose2d: K3C64S2P1-BN-ReLU |
| Rpad1 | 64 | reflectionpad2d((3,3,3,3)) |
| Out | 1 | K7C1S1 |

Table S6. Network architecture of the shared decoder. We use '`K-C-S-P`' to denote the kernel, channel, stride, and padding configuration of convolution layers.

## References


[1] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. FSDR: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6891–6902, 2021. 2

[2] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 783–792, 2021. 2

[3] Wenbin Xie, Dehua Song, Chang Xu, Chunjing Xu, Hui Zhang, and Yunhe Wang. Learning frequency-aware dynamic network for efficient super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4308–4317, 2021. 2

[4] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2