Supplementary Material

Wei-An Lin*[1] Haofu Liao*[2] Cheng Peng[1] Xiaohang Sun[3] Jingdan Zhang[4]
Jiebo Luo[2] Rama Chellappa[1] Shaohua Kevin Zhou[5,6]
[1]University of Maryland, College Park [2]University of Rochester [3]Princeton University
[4]Z2W Corporation [5]Chinese Academy of Sciences [6]Peng Cheng Laboratory, Shenzhen

1. Fanbeam CT Geometry

Figure 1 illustrates the general fanbeam CT geometry. The X-ray source and the arc detector rotate with respect to the origin. The distance between the X-ray source and the origin is D. For each projection angle β, the arc detector receives the X-rays transmitted from the object. The intensity values received by the detector is represented as a 1D signal with independent variable γ. As shown in the top of Figure 1, the sinogram data $Y_{fan}(\beta, \gamma)$ consists of the 1D signals received in different projection angles β.

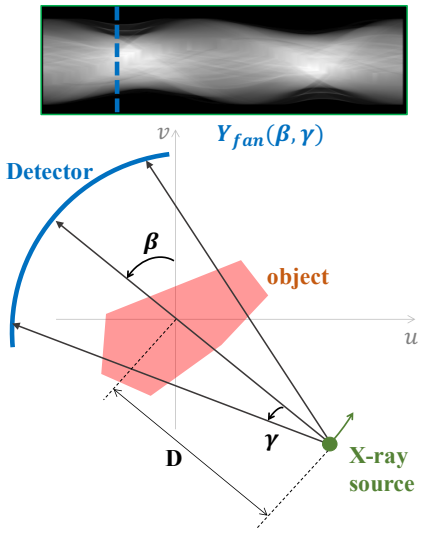


Figure 1: Fanbeam CT geometry.

2. Implementation Details

2.1. Network Architecture

The proposed DuDoNet consists of SE-Net and IE-Net. The architecture of SE-Net is presented in Table 1. The architecture of IE-Net is identical to U-Net [4]. N_c denotes the number of output channels. $\mathcal{M}_t \downarrow k$ represents the sinogram mask down-sized to $1/k$. All downsampling convolution layers except for the first layer use leaky ReLU activation function with $\alpha = 0.2$. All upsampling convolution layers except for the last layer use ReLU activation function. We use 'K#-C#-S#-P#' to denote the configuration of the convolution layers, where 'K', 'C', 'S' and 'P' stand for the kernel, channel, stride and padding size, respectively.

Name	N_c	Description
INPUT	2	Input sinogram and $\mathcal{M}_t$
DOWN_CONV0	64	K4-C64-S2-P1
CONCAT0	65	Concatenate $\mathcal{M}_t \downarrow 2$
DOWN_CONV1	128	K4-C128-S2-P1
CONCAT1	129	Concatenate $\mathcal{M}_t \downarrow 4$
DOWN_CONV2	256	K4-C256-S2-P1
CONCAT2	257	Concatenate $\mathcal{M}_t \downarrow 8$
DOWN_CONV3	512	K4-C512-S2-P1
CONCAT3	513	Concatenate $\mathcal{M}_t \downarrow 16$
DOWN_CONV4	512	K4-C512-S2-P1
CONCAT4	513	Concatenate $\mathcal{M}_t \downarrow 32$
UPSAMPLE5	513	
UP_CONV5	512	K3-C512-S1-P1
CONCAT5	$(512 + 513)$	Concatenate CONCAT3
UPSAMPLE6	$(512 + 513)$	
UP_CONV6	256	K3-C256-S1-P1
CONCAT6	$(256 + 257)$	Concatenate CONCAT2
UPSAMPLE7	$(256 + 257)$	
UP_CONV7	128	K3-C128-S1-P1
CONCAT7	$(128 + 129)$	Concatenate CONCAT1
UPSAMPLE8	$(128 + 129)$	
UP_CONV8	64	K3-C64-S1-P1
CONCAT8	$(64 + 65)$	Concatenate CONCAT0
UPSAMPLE9	$(64 + 65)$	
UP_CONV9	1	K3-C1-S1-P1

Table 1: Network architecture of SE-Net.

2.2. Radon Inversion Layer

We implement our Radon Inversion Layer (RIL) in PyTorch [3] and CUDA. In the following, we detail our imple-

mentation.

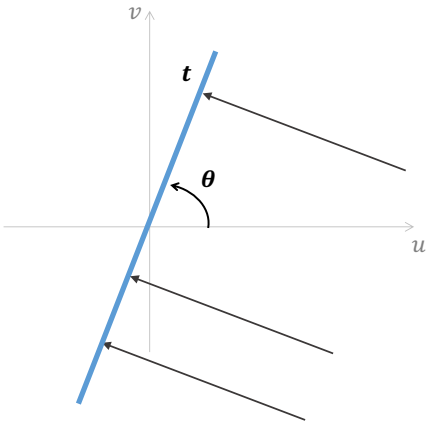


Figure 2: Parallel-beam CT geometry.

RIL consists of three modules: (1) parallel-beam conversion module, (2) Ram-Lak filtering module and (3) backprojection module. Given a fanbeam sinogram $Y_{fan}(\beta, \gamma)$, we first convert it to a parallel beam sinogram $Y_{para}(t, \theta)$ using the fan-to-parallel beam conversion. Then, we can reconstruct the CT image $X(u, v)$ using the Ram-Lak filtering and backprojection. The parallel-beam conversion is implemented according to the following relation

$$\theta = \gamma + \beta, \tag{1}$$
$$t = D \sin \gamma, \tag{2}$$

where t is the projection location on the parallel beam detector and θ is the projection angle in the parallel-beam geometry as shown in Figure 2. For efficiency, we implement the change of variable using two 1D interpolations, one for θ in (1) and the other for t in (2).

The Ram-Lak filtering for $Y_{para}(t, \theta)$ is implemented by

$$Q(t, \theta) = \mathcal{F}_t^{-1} \left\{ |\omega| \cdot \mathcal{F}_t \left\{ Y_{para}(t, \theta) \right\} \right\}, \tag{3}$$

where $\mathcal{F}_t$ and $\mathcal{F}_t^{-1}$ are the Discrete Fourier Transform (DFT) and inverse Discrete Fourier Transform (iDFT) with respect to the detector dimension. The filtering module is implemented using the operations torch.fft and torch.ifft in PyTorch.

The backprojection module takes the filtered projection $Q(t, \theta)$ as input, and reconstructs $X(u, v)$ via

$$\begin{aligned} X(u, v) &= \int_0^{\pi} Q(u \cos \theta + v \sin \theta, \theta) d\theta \\ &\approx \Delta\theta \sum_i Q(u \cos \theta_i + v \sin \theta_i, \theta_i) \\ &\approx \Delta\theta \sum_i (\lceil t_i \rceil - t_i) Q(\lfloor t_i \rfloor, \theta_i) \\ &\quad + (t_i - \lfloor t_i \rfloor) Q(\lceil t_i \rceil, \theta_i), \end{aligned} \tag{4}$$

where $t_i = u \cos \theta_i + v \sin \theta_i$ is a function of u, v, and i. The forward-pass of (4) is parallelizable in θ_i. During backpropagation, the gradients of the CT image with respect to the sinogram are given by

$$\frac{\partial X(u, v)}{\partial Q(t, \theta)} = \begin{cases} \Delta\theta(\lceil t_i \rceil - t_i), & \text{if } t = \lfloor t_i \rfloor, \\ \Delta\theta(t_i - \lfloor t_i \rfloor), & \text{if } t = \lceil t_i \rceil, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The backprojection module is implemented as a CUDA extension of PyTorch.

3. Evaluation on CT Images with Real Metal Artifact

Evaluating MAR methods on CT images of patients carrying metal implants is challenging for two reasons: 1) Modern clinical CT machines have certain build-in MAR algorithms. Evaluations on CT images after MAR would not be meaningful; 2) Sinogram data with metal artifacts are difficult to access, except perhaps from machine manufacturers. To the best of our knowledge, there is no existing sinogram database which targets MAR.

In order to compare different MAR methods, we manually collect CT images with metal artifact from DeepLesion [6] and apply the following steps to obtain the metal trace $\mathcal{M}_t$ and the LI sinogram Y_{LI}. DuDoNet can be applied by taking $\mathcal{M}_t$ and Y_{LI} as inputs. Conceptually, the steps can be understood as projecting the input CT image with unknown imaging geometry to the source domain[1] with known geometry.

(i) $\mathcal{M}_t$: We first segment out the metal mask by applying a threshold of 2,000 HU to the metal-corrupted CT image. $\mathcal{M}_t$ can be obtained by forward projection with the imaging geometry presented in Section 4 in the manuscript.

(ii) Y_{LI}: We adopt the same simulation procedures and imaging geometry as in the manuscript to synthesize metal-corrupted sinogram Y. Y_{LI} can be generated from Y and $\mathcal{M}_t$ by linear interpolation.

Figure 3 presents visual comparisons of different MAR algorithms. Metal masks obtained by step (i) are colored in yellow. We would like to emphasize that the true sinogram of a given CT image cannot be inferred without information about the actual imaging geometry (e.g. source to detector distance, and number of projection views). Therefore, in Figure 3, due to inconsistent imaging geometry, sinogram-based MAR approaches (e.g. LI) may lead to an even worse visual quality than raw CT. In contrast, DuDoNet effectively reduces metal artifacts in real CT images.

[1] The domain of CT images with simulated metal artifacts.

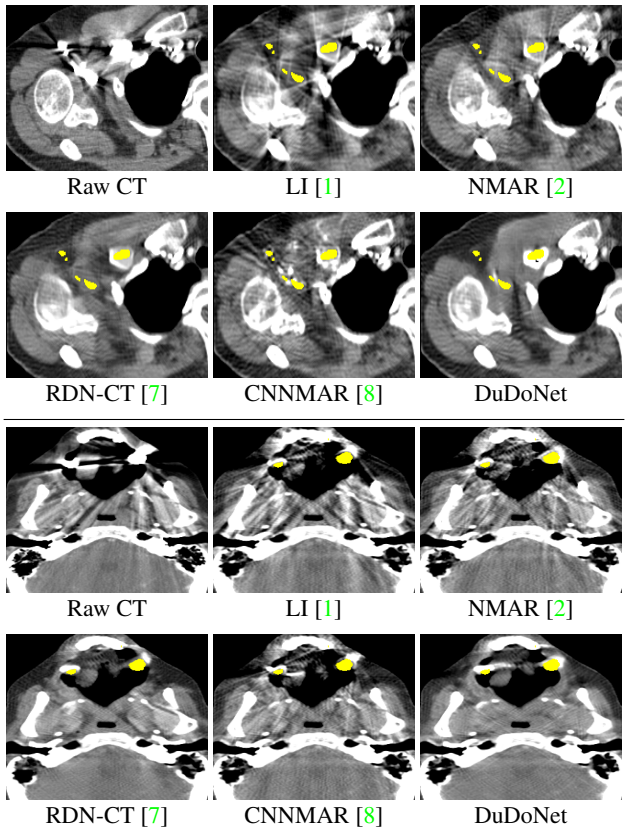


Figure 3: Evaluations on real data. All models are exactly the same as in the main paper (no re-training).

4. Additional Visual Comparisons on CT images with Synthesized Metal Artifact

See pages 4 and 5.

5. Practical Issues

In this section, we discuss practical issues when applying deep learning for MAR. Suppose we have access to the sinograms and CT images taken from a CT machine, methods such as CNNMAR [8], and cGAN-CT [5] require paired data, i.e., CT images with and without metallic implants from the same patient. In our approach, data within the metal trace is viewed as missing and replaced using LI [1]. Therefore, to train a DuDoNet, only implant-free sinograms, CT images and masks of metallic implants are required. In the experimental evaluations, we synthesize metal artifacts mainly for comparing with existing MAR approaches. In real applications, simulated pairs are not needed by the proposed DuDoNet.

References

[1] W. A. Kalender, R. Hebel, and J. Ebersberger. Reduction of ct artifacts caused by metallic implants. *Radiology*, 164(2):576–577, 1987. 3, 4, 5

[2] E. Meyer, R. Raupach, M. Lell, B. Schmidt, and M. Kachelrieß. Normalized metal artifact reduction (nmar) in computed tomography. *Medical physics*, 37(10):5482–5493, 2010. 3, 4, 5

[3] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 1

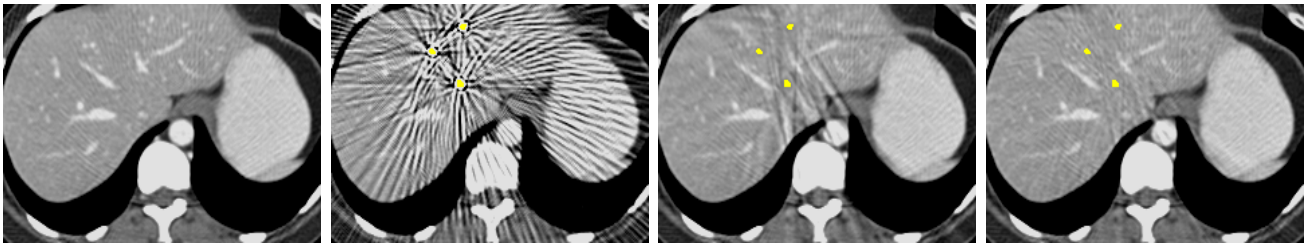
[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 1

[5] J. Wang, Y. Zhao, J. H. Noble, and B. M. Dawant. Conditional generative adversarial networks for metal artifact reduction in ct images of the ear. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2018. 3, 4, 5

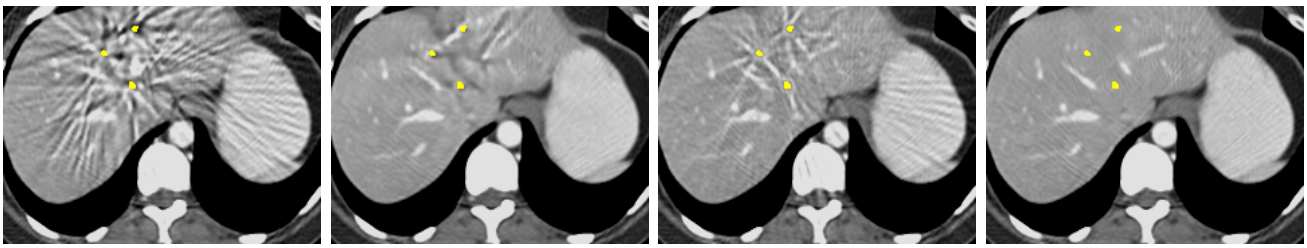
[6] K. Yan, X. Wang, L. Lu, L. Zhang, A. P. Harrison, M. Bagheri, and R. M. Summers. Deep lesion graphs in the wild: Relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[7] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 4, 5

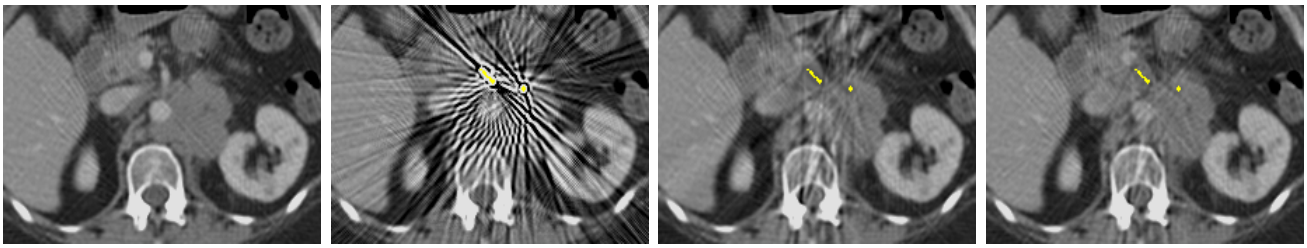
[8] Y. Zhang and H. Yu. Convolutional neural network based metal artifact reduction in x-ray computed tomography. *IEEE Transactions on Medical Imaging*, 2018. 3, 4, 5



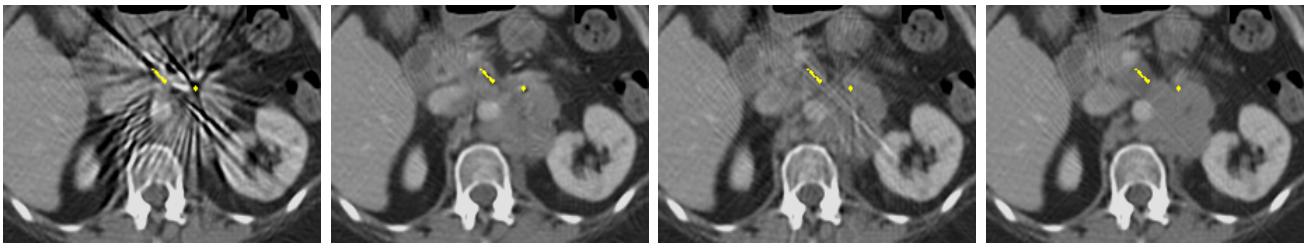
Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



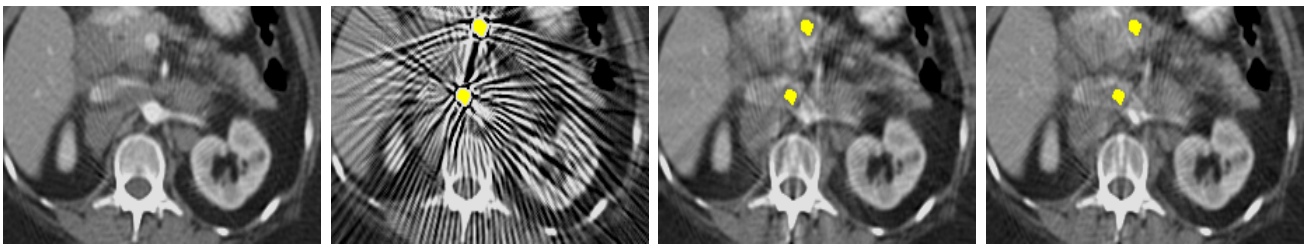
cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet



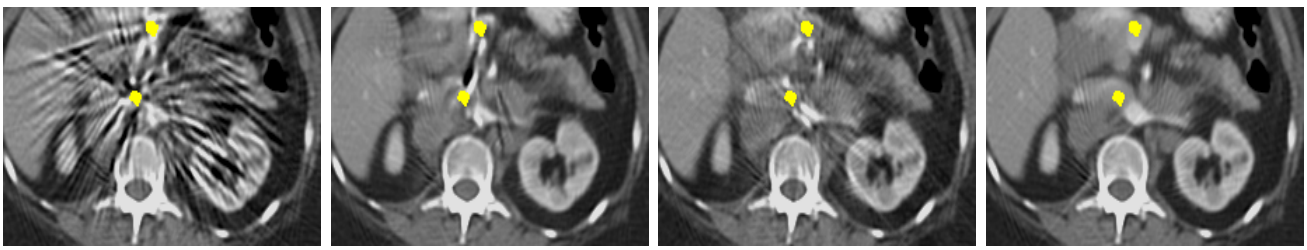
Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



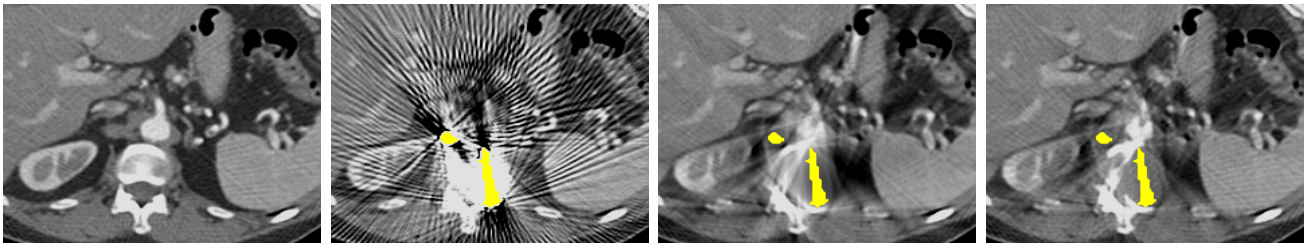
cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet



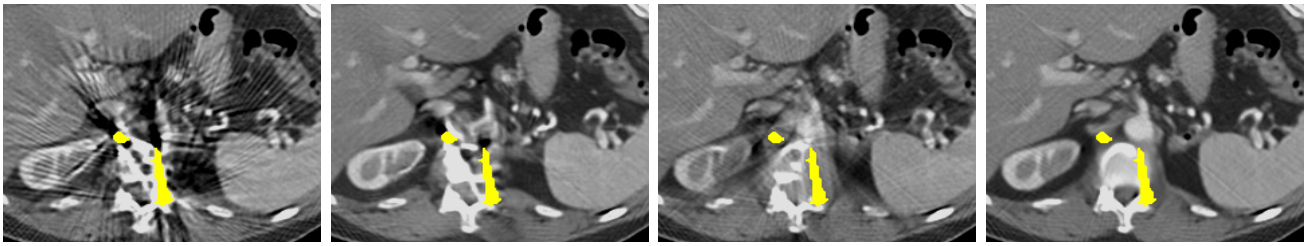
Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



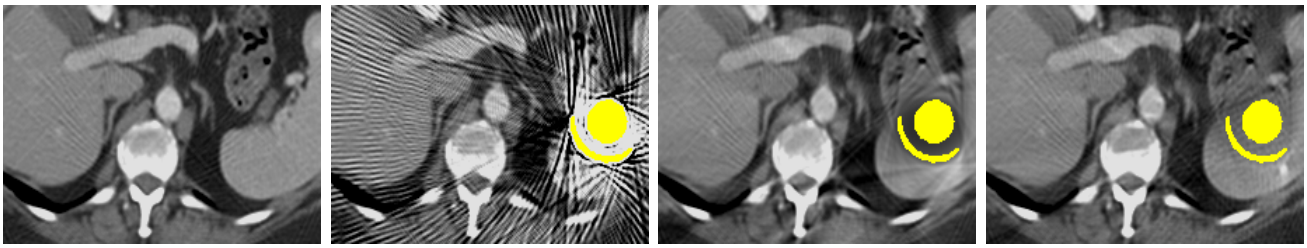
cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet



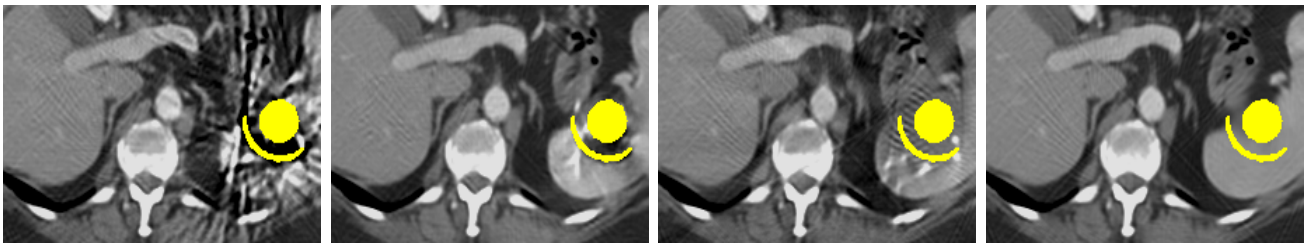
Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



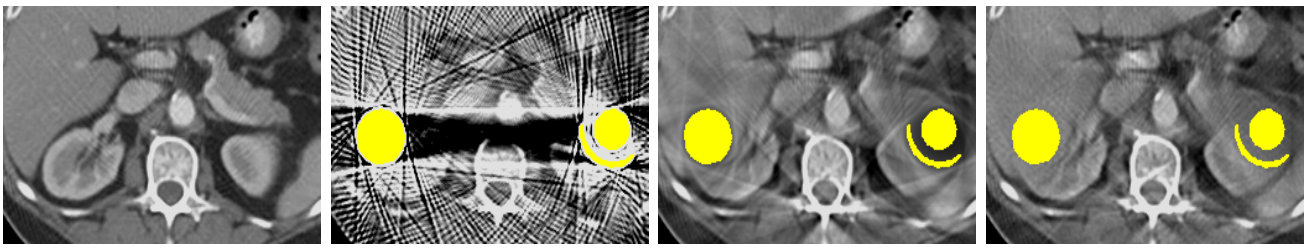
cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet



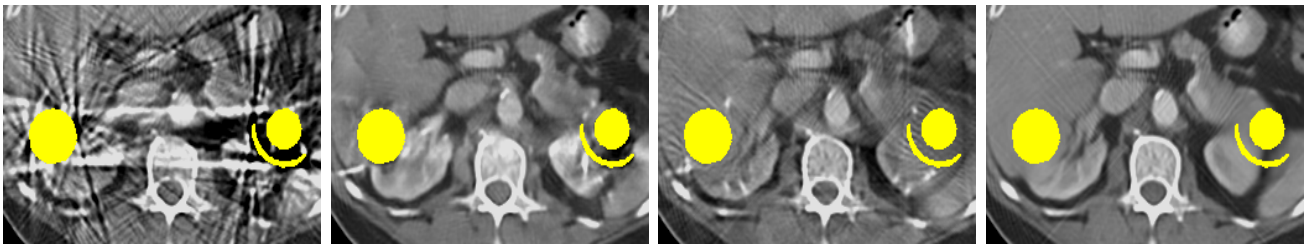
Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet



Ground Truth | With Metal Artifact | LI [1] | NMAR [2]



cGAN-CT [5] | RDN-CT [7] | CNNMAR [8] | DuDoNet