A. Network Architecture

The proposed SAINT consists of AMI and RFN. For AMI, the model architecture for the feature learning stage is described in Section 4 and in [27] with more details. The model architecture for AMI's filter generation stage is presented in Table 3. N_c denotes the number of output channels, C' denotes the channel dimension of generated features F^{LR}. 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	1	Input FDM
CONV0	32	K3-C1-S1-P1
RELU		
CONV1	64	K3-C64-S1-P1
RELU		
CONV2	64	K3-C64-S1-P1
RELU		
CONV3	64	K3-C64-S1-P1
RELU		
CONV4	C'	K3-C64-S1-P1

Table 3: Network architecture of AMI's filter generation stage.

For RFN, we use RDN with five RDBs, four convolutional layers per RDB, and growth rate of sixteen. Additionally, for the first convolutional layer, RFN outputs thirty-two channels instead of sixty-four, which is the default hyperparameter in RDN and is used in AMI's model construction. The upsampling module is a single convolutional layer, since the input and output have the same image height and width. We find that expanding RFN's depth or width does not show improvement to the slice interpolation results quantitatively.

B. Stitching Artifacts

Due to the high memory consumption of 3D volumetric data, CT volumes cannot be directly inferred through deep 3D CNN networks. In Section 4 we infer and compare only the central $256 \times 256 \times Z$ patch for all non-SAINT methods to reduce the memory requirement, with the exception of mDCSRN, which are inferred by the patch-based algorithm discussed in [4].

When an entire 3D volume needs to be super-resolved, all competing 3D CNN models need to use some form of patch-based algorithm that divides CT volumes into individual cubes to be inferred independently. However, such an approach introduces artifacts at the fringe, where the divided cubes are put back together. This is due to SISR models heavily employing padding[4] to keep the same dimensionality throughout convolutions, i.e. for every convolutional layer with a filter size of k, the input tensor needs to be padded by $\lfloor \frac{k}{2} \rfloor$ for the output tensor to retain the same shape. For our implementation of the 3D RDN, there are fifty-two convolutional layers, which means the original input is padded by fifty-two voxels on each side, resulting in an overall padding size of $104 \times 104 \times 104$. Such a large padding size distorts the real data distribution, and adversely affects voxel prediction accuracy, especially at the fringe, of the divided cues. As a result, when the cubes are reassembled together to form the super-resolved volume, the boundaries between them are often inconsistent. We refer to the artifact caused by this inconsistency as the stitching artifact.

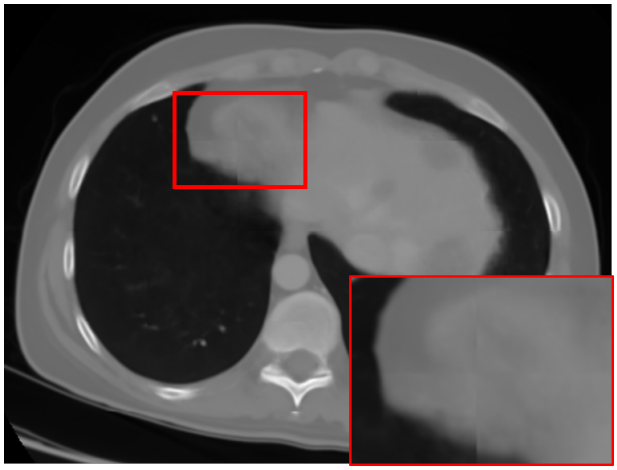


Figure 8: The stitching artifacts, following the procedure described in [4] with three voxel margin.

The patch-based algorithm discussed in [4] attempts to alleviate this problem by introducing overlaps of three voxels between the divided 3D cubes, effectively replacing the padding of three initial convolution layers with real voxel values. As we have shown in Fig. 7 and an enlarged version in Fig. 8, this still leads to noticeable stitching artifacts with a deep network. Theoretically, to completely eliminate such artifact for 3D RDN, the input tensor needs to be padded with at least fifty-two voxels on each side, which leads back to memory bottleneck and inefficiency. In comparison, since SAINT breaks down 3D SISR into separate stages of 2D SISR, it completely eliminates stitching artifacts, thus also allowing for larger network size to be used.

C. Alternative RFN implementations

In this section, we showed the different implementations of RFN that we have experimented with.

[4] zero-padding is used for all models in this paper

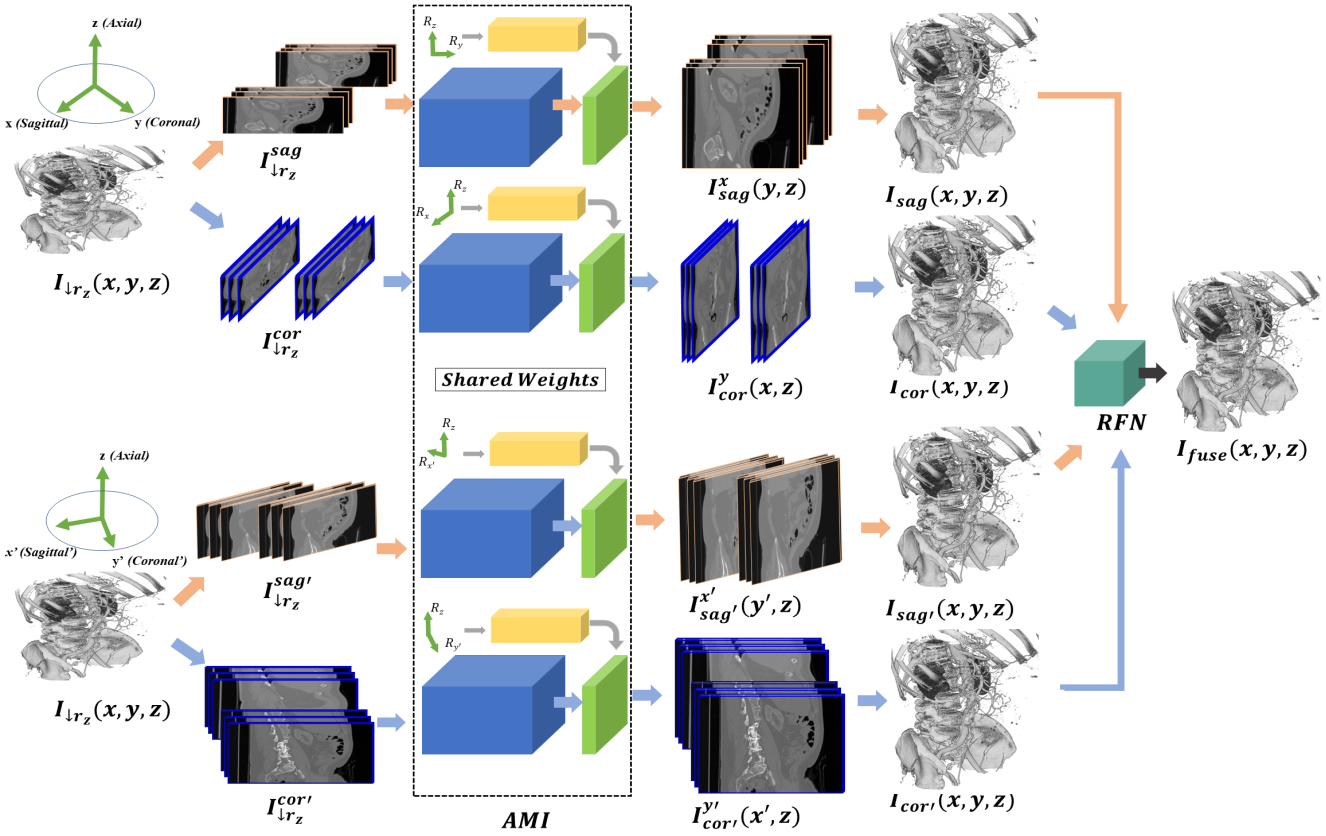


Figure 9: The augmented version of Spatially Aware Interpolation NeTwork (SAINT). Instead of using the sagittal and coronal views, the augmented SAINT also attempts to incorporate alternative views. For visualization purpose, the volumes are rendered in 3D based on their bone structures.

3D RFN Due to RFN's lightweight and shallow network structure, it is memory-wise feasible to employ the patch-based algorithm for inference with enough margin on each side to eliminate the stitching artifacts. We implement a 3D version of RFN, where it uses 3D convolutional filters instead of 2D, to observe if that allows better modelling of the 3D context. As shown in Table 3, we do not see any observable difference quantitatively between 2D and 3D RFN's results.

Four Views To axially interpolate a 3D volume, AMI first upsamples it from the coronal view and sagittal view, i.e. $I^{y}_{\downarrow r_z}(x,z)$ and $I^{x}_{\downarrow r_z}(y,z)$, and leaves RFN to improve consistency from the axial view $I^{z}(x,y)$. However, $I^{y}_{\downarrow r_z}(x,z)$ and $I^{x}_{\downarrow r_z}(y,z)$ are not the only two views in a 3D volume that can be used to super-resolve the z axis. Technically, there are infinite number of views that include the z axis in 3D. To this end, we perform an experiment to see if axially upsampling volumes from alternative views can improve performance.

As shown in Fig. 9, we experiment with an augmented version of SAINT, where AMI upsamples 2D images from four views, instead of just the sagittal and coronal views. In addition to (x,z) and (y,z), we define two additional axes x' and y', which are rotated from the x and y axes by 45°on the (x,y) plane. Following similar procedures described in Section 3.1, we sample from volume $I_{\downarrow r_z}$ to obtain $I^{x'}_{\downarrow r_z}(y',z)$ and $I^{y'}_{\downarrow r_z}(x',z)$, of which we super-resolve with AMI. The super-resolved slices are reformatted into 3D volumes $I_{cor'}(x',y',z)$ and $I_{sag'}(x',y',z)$[5], and are passed to RFN with I_{cor} and I_{sag}.

For RFN, I_{avg} is the average of four volumes I_{sag}, I_{cor}, $I_{sag'}$, $I_{cor'}$, and I^{z}_{fuse} becomes:

$$I^{z}_{fuse}(x,y) = I^{z}_{avg}(x,y) + \mathcal{F}_{\phi}(I^{z}_{sag}(x,y), I^{z}_{cor}(x,y), I^{z}_{sag'}(x,y), I^{z}_{cor'}(x,y)). \quad (12)$$

All loss functions and network structures remain the same.

[5] $I_{cor'}(x',y',z)$ and $I_{sag'}(x',y',z)$ can be converted to $I_{cor'}(x,y,z)$ and $I_{sag'}(x,y,z)$ through simple rotation of axes.

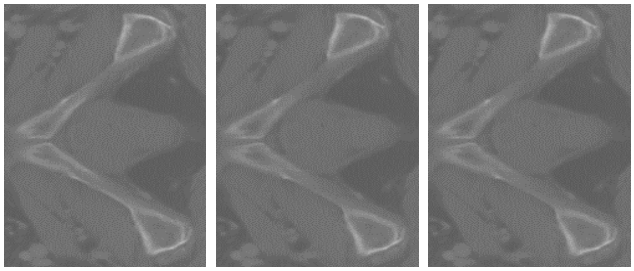
Scale	PSNR/SSIM	Parameters	Liver	Colon	Hepatic Vessels	Kidney
x4	AMI+RFN$^{2D}_{2View}$	2.92M	34.91/0.9603	34.19/0.9579	34.48/0.9630	**35.79/0.9597**
	AMI+RFN$^{3D}_{2View}$	2.92M	34.84/0.9602	34.21/0.9583	34.50/0.9631	35.44/0.9566
	AMI+RFN$^{2D}_{4View}$	2.92M	**34.94/0.9611**	**34.29/0.9590**	**34.60/0.9639**	35.56/0.9575
x6	AMI+RFN$^{2D}_{2View}$	2.92M	**32.49/0.9395**	31.48/0.9321	31.87/0.9404	**33.22/0.9393**
	AMI+RFN$^{3D}_{2View}$	2.92M	32.36/0.9390	31.51/0.9324	31.87/0.9404	32.92/0.9352
	AMI+RFN$^{2D}_{4View}$	2.92M	32.37/0.93890	**31.52/0.9324**	**31.89/0.9404**	32.92/0.9352

Table 4: Quantitative Comparison of different RFN implementations. The superscript on RFN describes whether RFN is implemented with 2D or 3D filters; the subscript describes whether RFN fuses volumes super-resolved from two views (sagittal and coronal) or four views (as described in C). The best results are in **bold**, and the second best results are underlined.

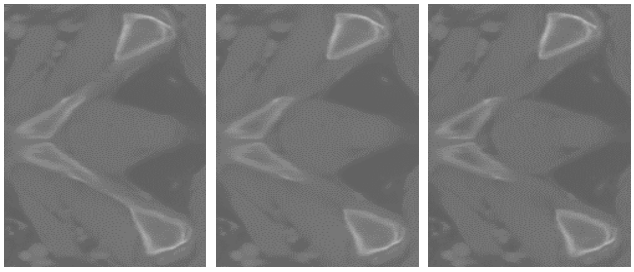
We found that the two additional planes only improve SAINT performance marginally.

D. Effects of FDM on interpolation results

SAINT generates interpolated slices based on the input of FDM, which is dependent on the voxel spacing of specific slices (as shown in Algorithm 1). We believe that the incorporation of voxel spacing, especially the spacing between slices R_z, is important, as it is an indication of how much the details should shift between consecutive slices.



(a) Interpolated Results, $R_z = 1mm$



(b) Interpolated Results, $R_z = 5mm$

Figure 10: Visual comparison of slice interpolation ($r_z = 4$) with different voxel spacing input. Notice how the bone structures change faster for (a) as compared to (b), as the slices are supposed to be further apart according to the respective R_z.

To visually understand how changing voxel spacing values impact interpolation results from SAINT, we use AMI to super-resolve the same CT volume with different values of R_z, as shown in Fig. 10. We found that through the formulation of FDM, the interpolated slices produce details that change more rapidly if R_z is high, and more slowly if R_z is low.

E. Additional Visual Comparisons on CT images

Please refer to Fig. 11 and Fig. 12 for more visual comparisons of synthesized slices from different methods. To better demonstrate the results in 3D, sagittal slices are also included for reference.

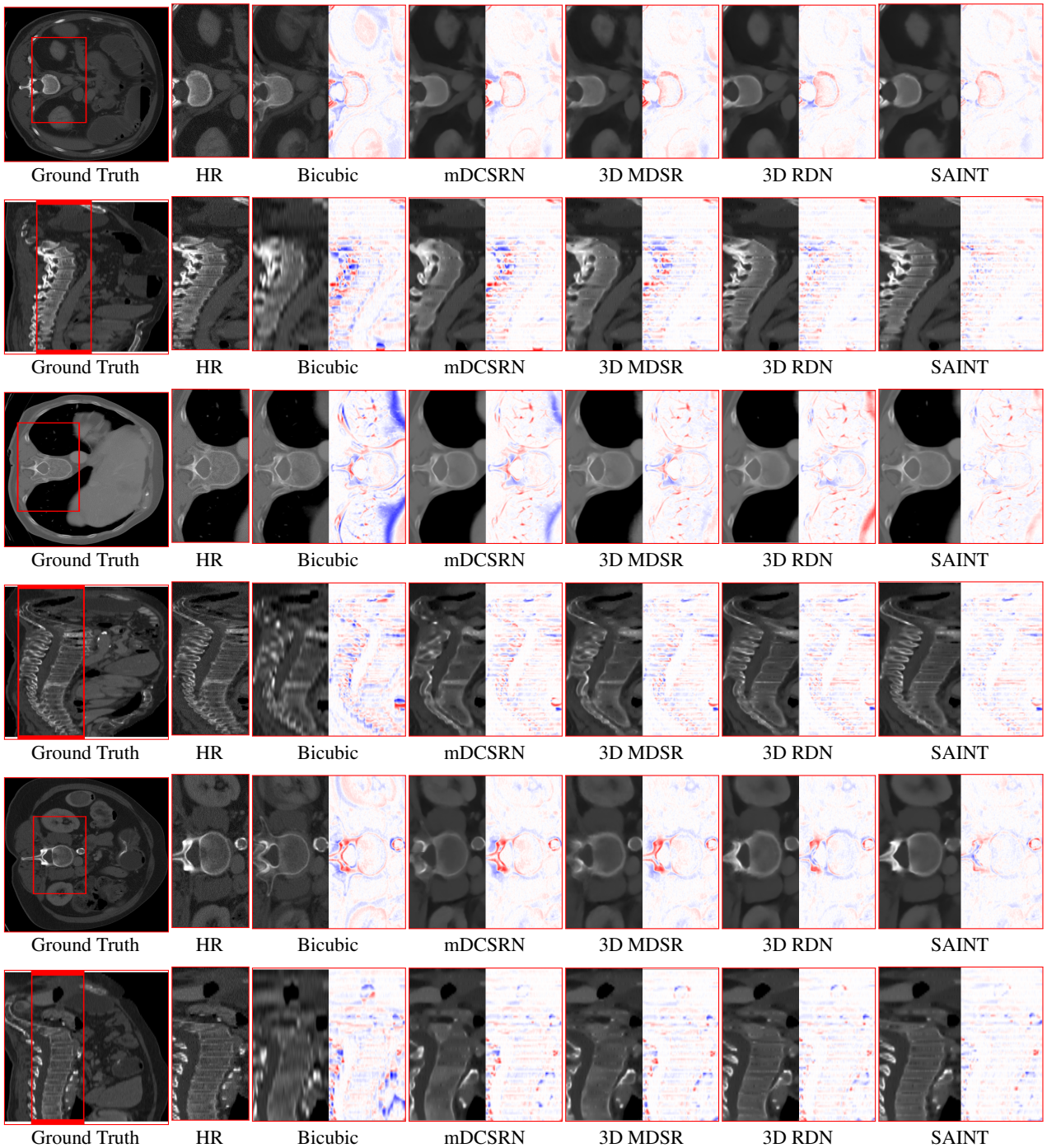


Figure 11: Visual comparisons of different methods against SAINT for $r_z = 4$. The difference maps are provided to the right of the results for better visualization. Images are best viewed when magnified.

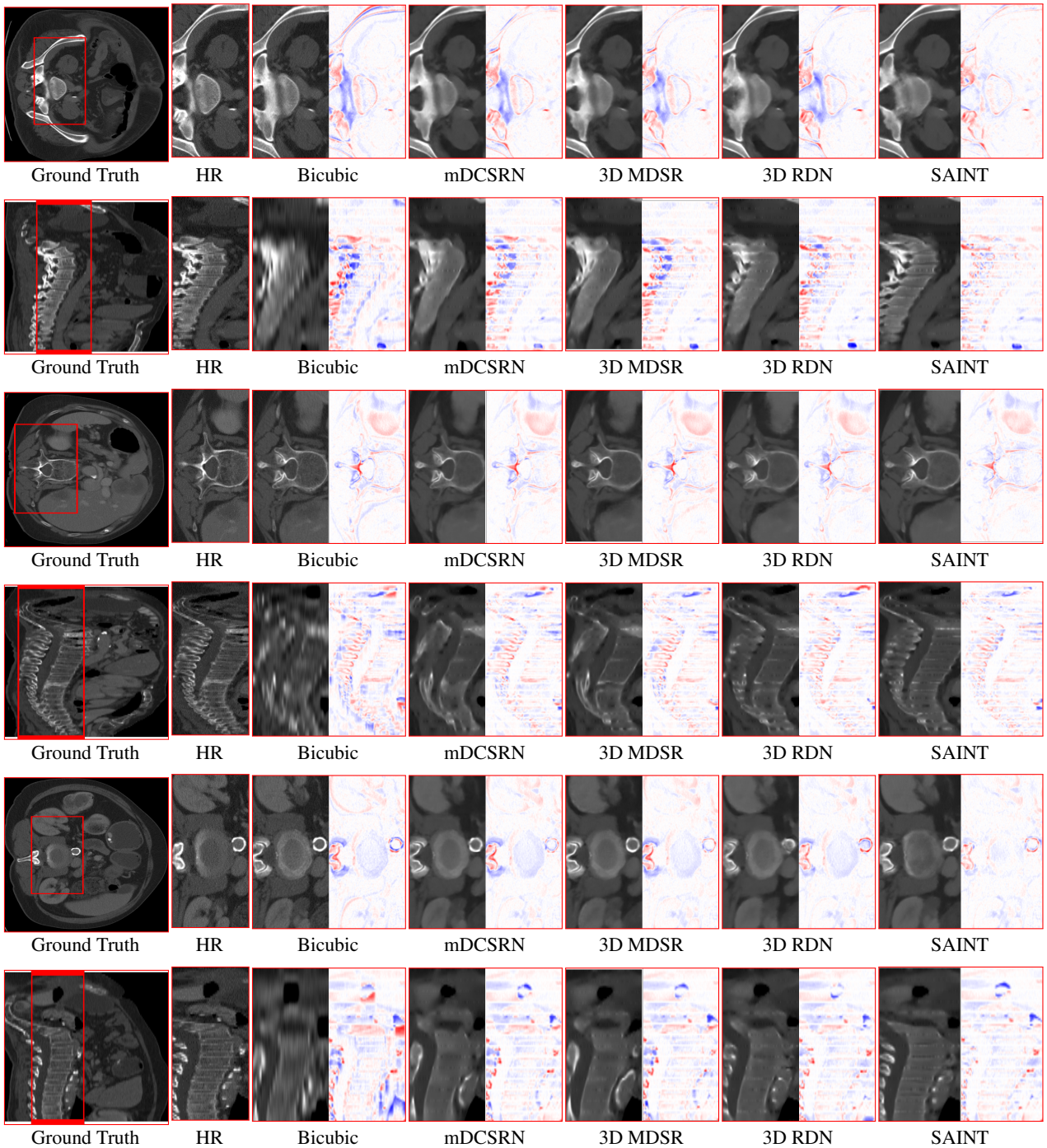


Figure 12: Visual comparisons of different methods against SAINT for $r_z = 6$. The difference maps are provided to the right of the results for better visualization. Images are best viewed when magnified.