“Self-Supervised Deep Active Accelerated MRI”

Authors: Kyong Hwan Jin, Michael Unser & Kwang Moo Yi

Presented By: Jason T. Smith

MR Imaging

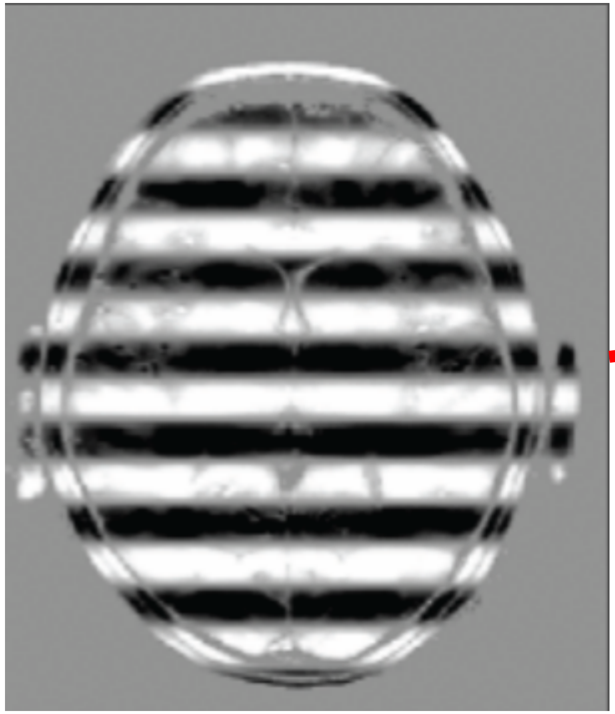
Therefore, minimizing the time required for sensitive MRI requires:

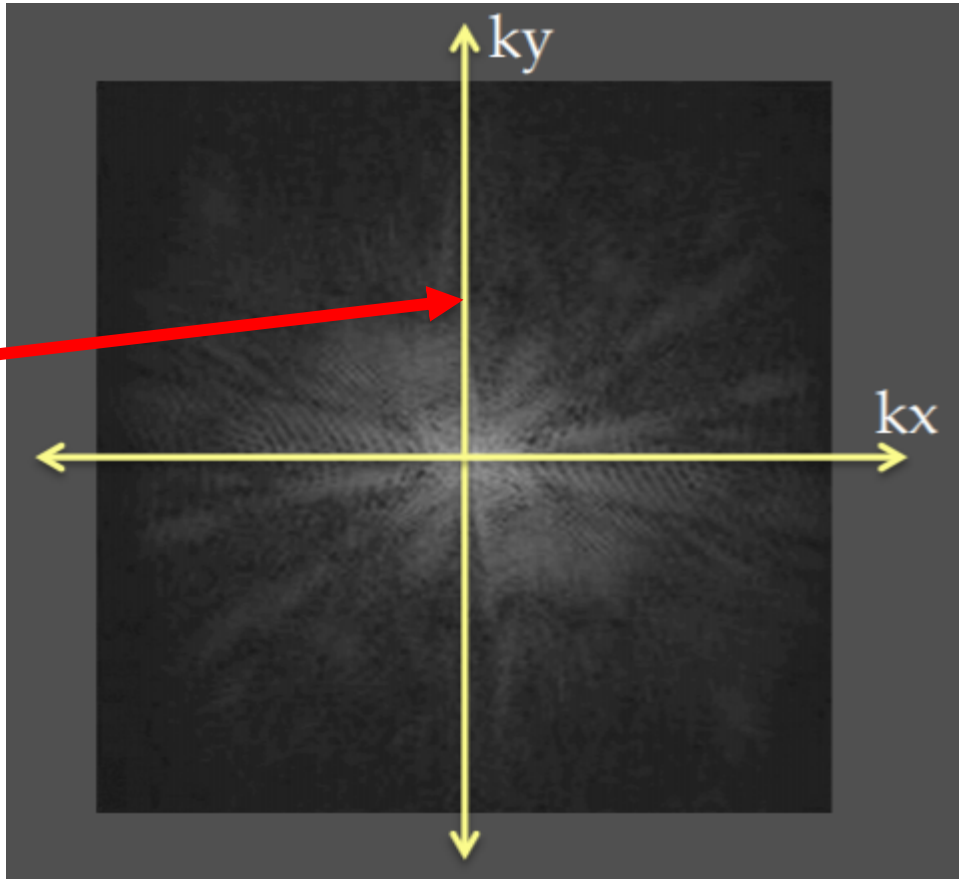
- Taking fewer data scans.
- Optimizing the reconstruction from k space using this reduced data.

"Making the most out of each scan."

- Be in discomfort

Sprawls





Each acquisition characterizes the frequency along one line of k-space. **This corresponds to a single datapoint.**

Minimizing Acquisition Time w/o Loss

Question: **How does one obtain the largest amount of information given a set number of k space samples?**

Conventional answers:

1) For a set compression rate, use the lowest frequency portion.
   - Using lowest frequency portion misses high frequency features!

2) **Accelerating MRI** via Aliasing and CS:
   - But… is there a better way than random selection?

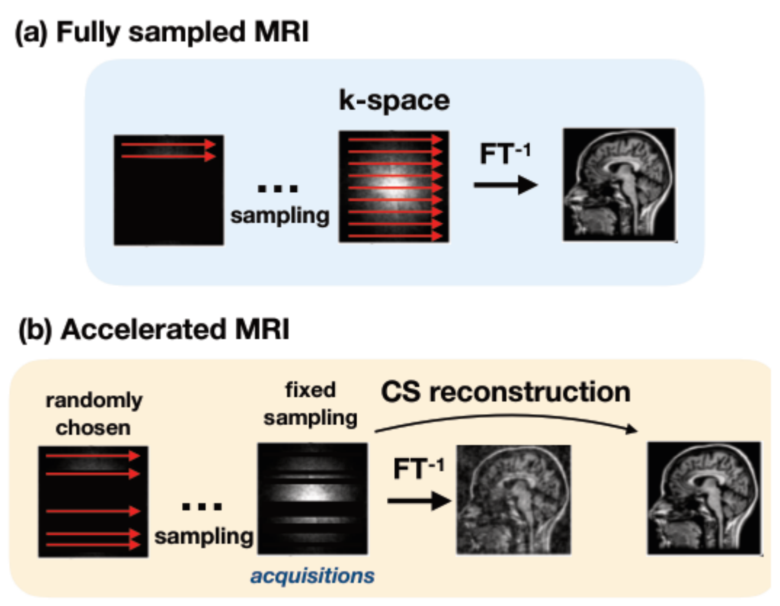


Fig. 2. Accelerated MRI. (a) Fully sampled MRI collects all k-space lines to obtain a target image. (b) Accelerated MRI acquires fewer k-space lines and then reconstructs downsampled acquisitions from an aliased image. Compressed-sensing (CS) reconstructions rely on the sparsity of the image domain.

Reinforcement Learning for Subsample Optimization

"… we draw inspiration from AlphaGo."

The DL framework presented, given a target number of subsamples, ***simultaneously*** learns to reconstruct (**ReconNet portion**) MRI images and determines the next k space subsample to optimize the PSNR (**SampleNet portion**).

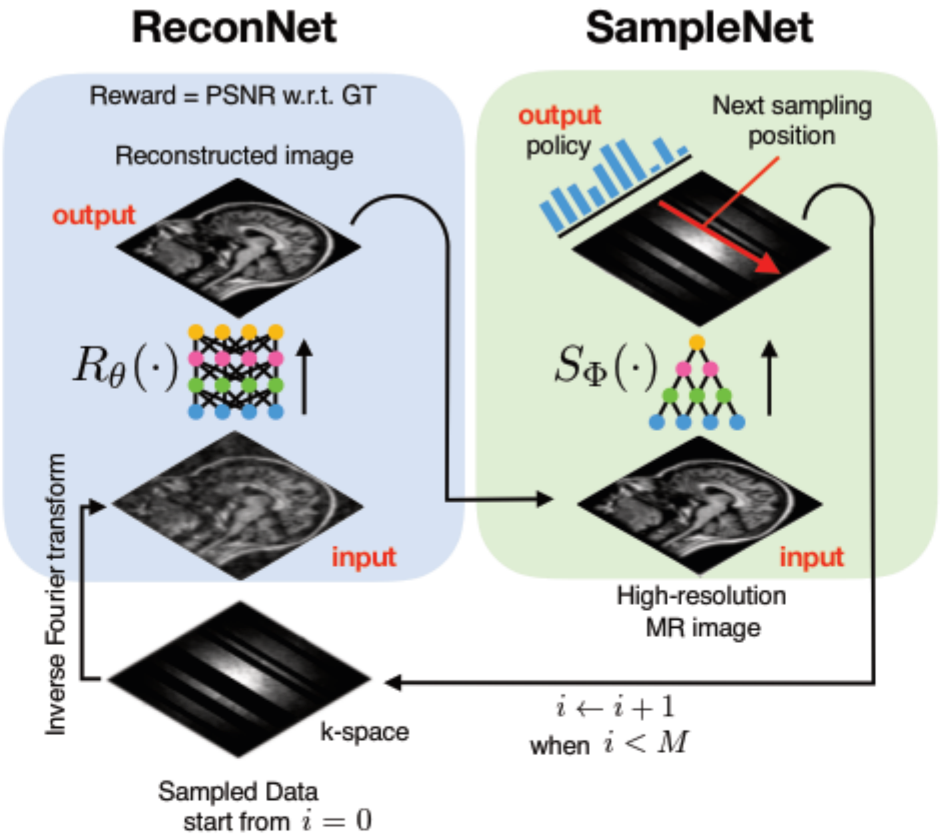


Fig. 3. Overall framework of our method. We train two deep neural networks. One learns to reconstruct a high-resolution MRI, and the other learns to estimate the policy for determining the position of the next sample. We progressively sample, with SampleNet, based on the reconstructed outcome of ReconNet using collected data.

Problem Statement

x = full-resolution signal (image domain)

P = Boolean matrix

F = DFT

y = raw measurements

Θ corresponds to sampling

***g*(…)** corresponds to reconstruction quality metric (SSIM, PSNR, etc.)

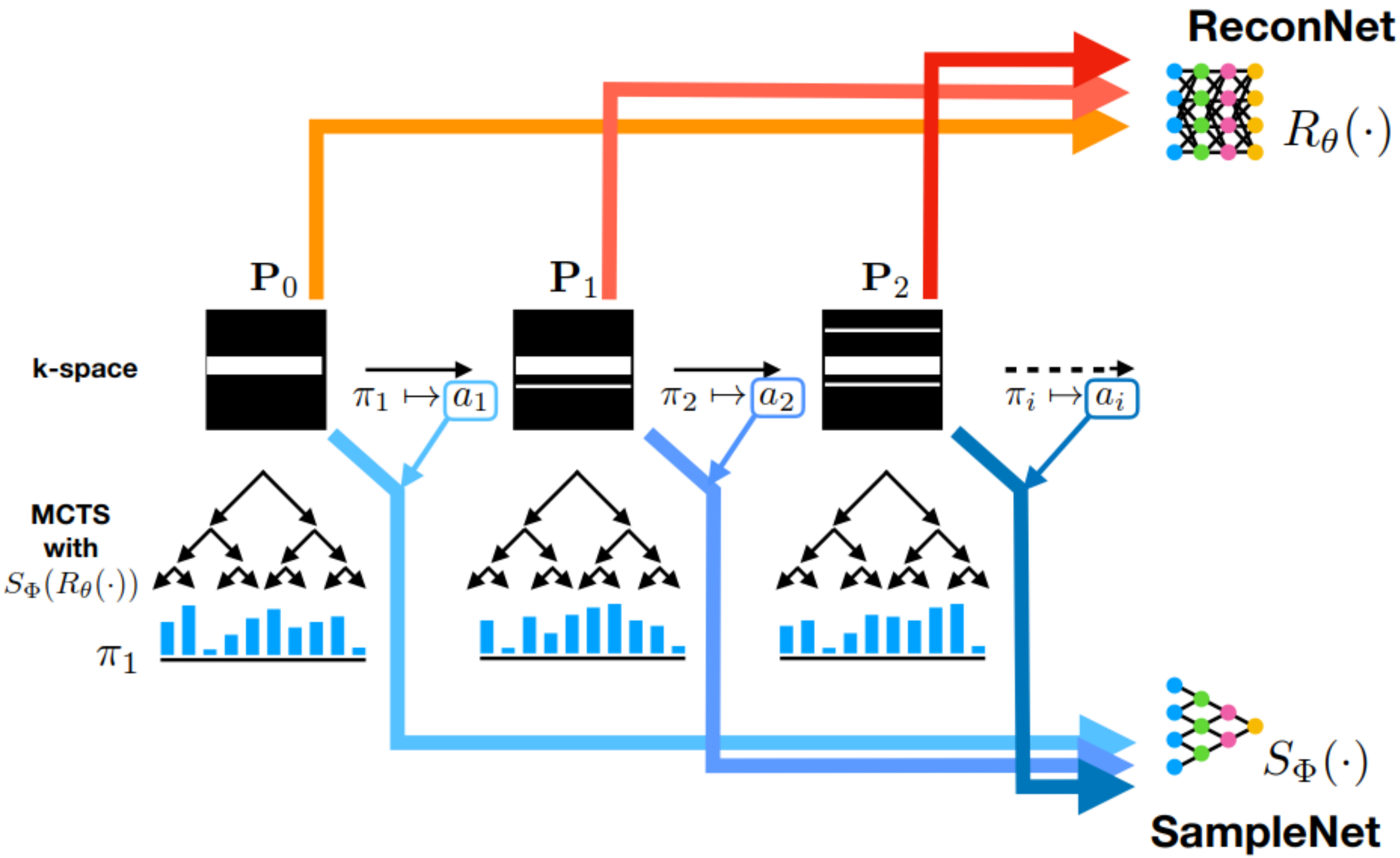
$$\mathbf{y} = \mathbf{PFx}.$$

$$\hat{\boldsymbol{\theta}}, \hat{\mathbf{P}} = \arg\max_{\boldsymbol{\theta},\mathbf{P}} \mathbb{E}_{\mathbf{x}} \left[g\left(f_{\boldsymbol{\theta}}\left(\mathbf{PFx}\right), \mathbf{x}\right)\right]$$

Wavelet…

$$f\left(\mathbf{y}\right) = \arg\min_{\mathbf{x}} \|\boldsymbol{\Psi}\mathbf{x}\|_{\ell_1} \quad \text{subject to } \mathbf{y} = \mathbf{PFx}$$

✗ Optimizes only Θ



Monte Carlo tree search (MCTS) performed until sampling budge = N_{total}.

"…during training, the sampling results with MCTS should always be better than what the two networks could provide without, thus providing guidance."

MCTS Possible Moves

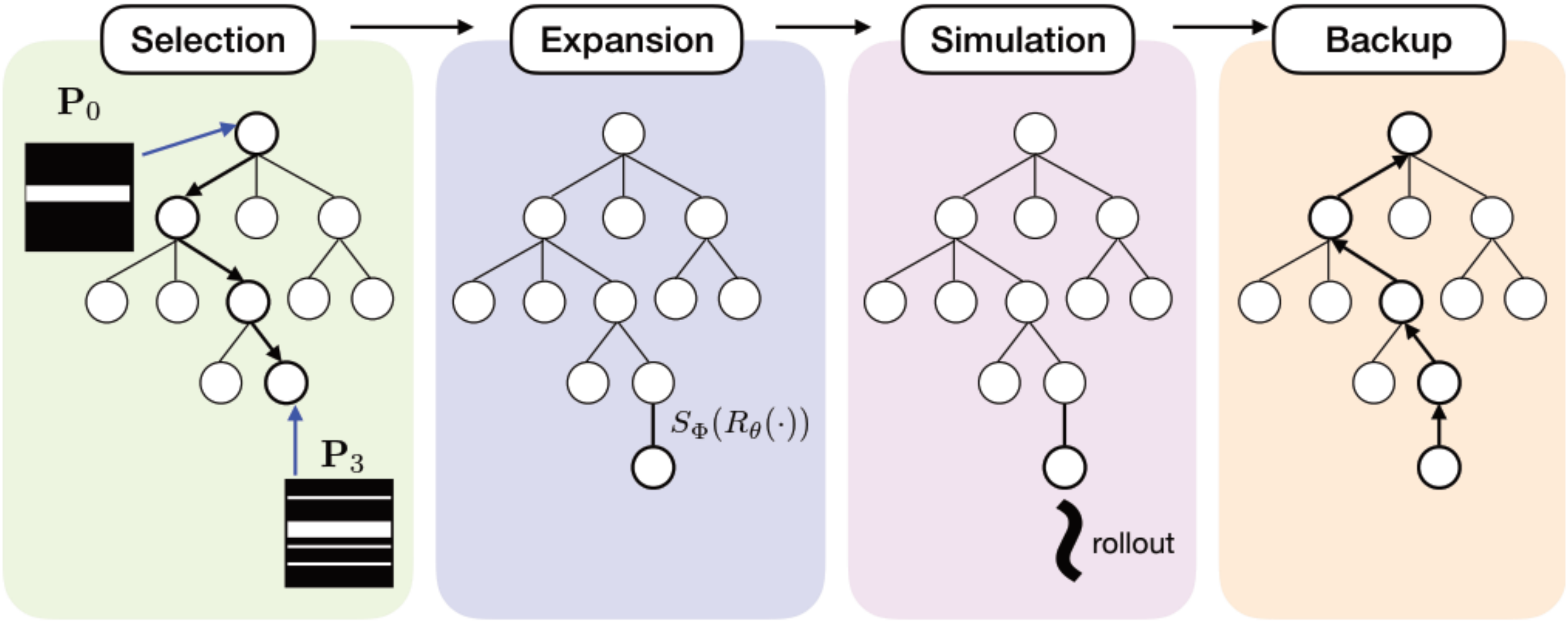


Fig. 5. Our Monte Carlo tree search. Each node in the tree denotes a sample pattern, while a move down the tree involves sampling a new position. We traverse the tree until a leaf node has been reached, where we simulate sampling with the sampling network (SampleNet) until some sampling budget has been reached. We then use the actual reconstruction performance as the reward, which is then backed up to all parent nodes. When performing the backup, we save the average and the maximum to consider the best reconstruction within all child nodes as well as the average reconstruction quality.

Guiding the Tree of Subsamples

$$U(\mathbf{P}, a) = (1-\alpha) Q\left(\begin{bmatrix} \mathbf{P} \\ \mathbb{1}_a \end{bmatrix}\right) + \alpha V\left(\begin{bmatrix} \mathbf{P} \\ \mathbb{1}_a \end{bmatrix}\right) + C_{puct}\left((1-\epsilon)\,\boldsymbol{\pi}_a + \epsilon\boldsymbol{\delta}\right)\sqrt{\frac{N(\mathbf{P})}{N\left(\begin{bmatrix} \mathbf{P} \\ \mathbb{1}_a \end{bmatrix}\right)}}$$

a = movement

P = state

$\boldsymbol{\pi}_a$ = policy at *a*

$\boldsymbol{\alpha}$, $\mathbf{C}_{puct}$ = Hyperparameters

$\boldsymbol{\delta}$ = Dirichlet noise (encourage exploration ($\boldsymbol{\epsilon}$, scalar))

U(P) = average of all rewards

V(P) = maximum reward

N(P) = # of visits to a particular node

Network Architecture(s)

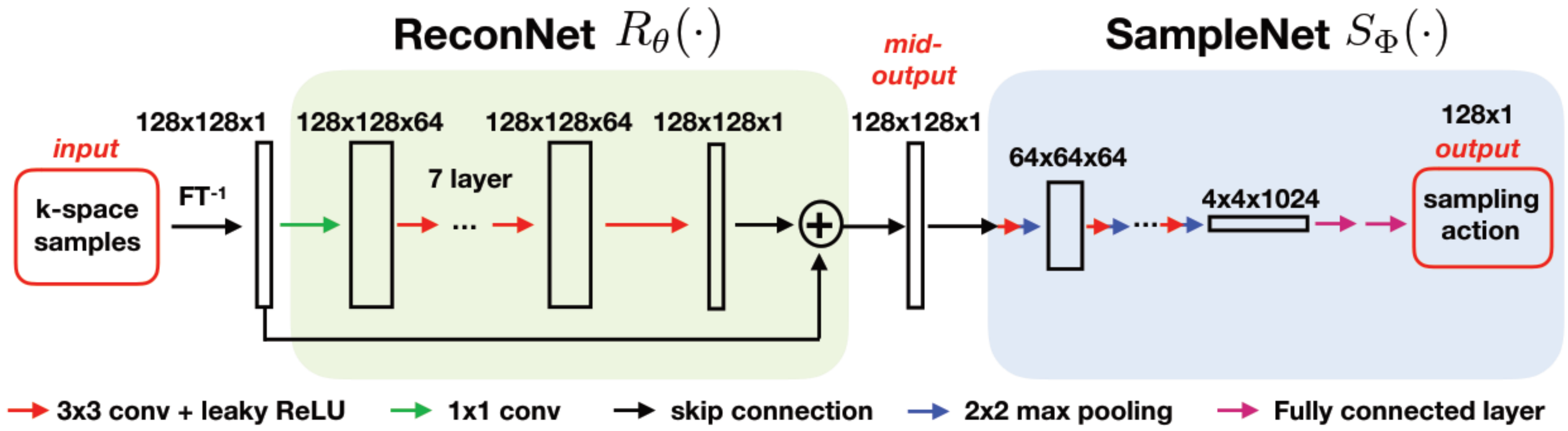
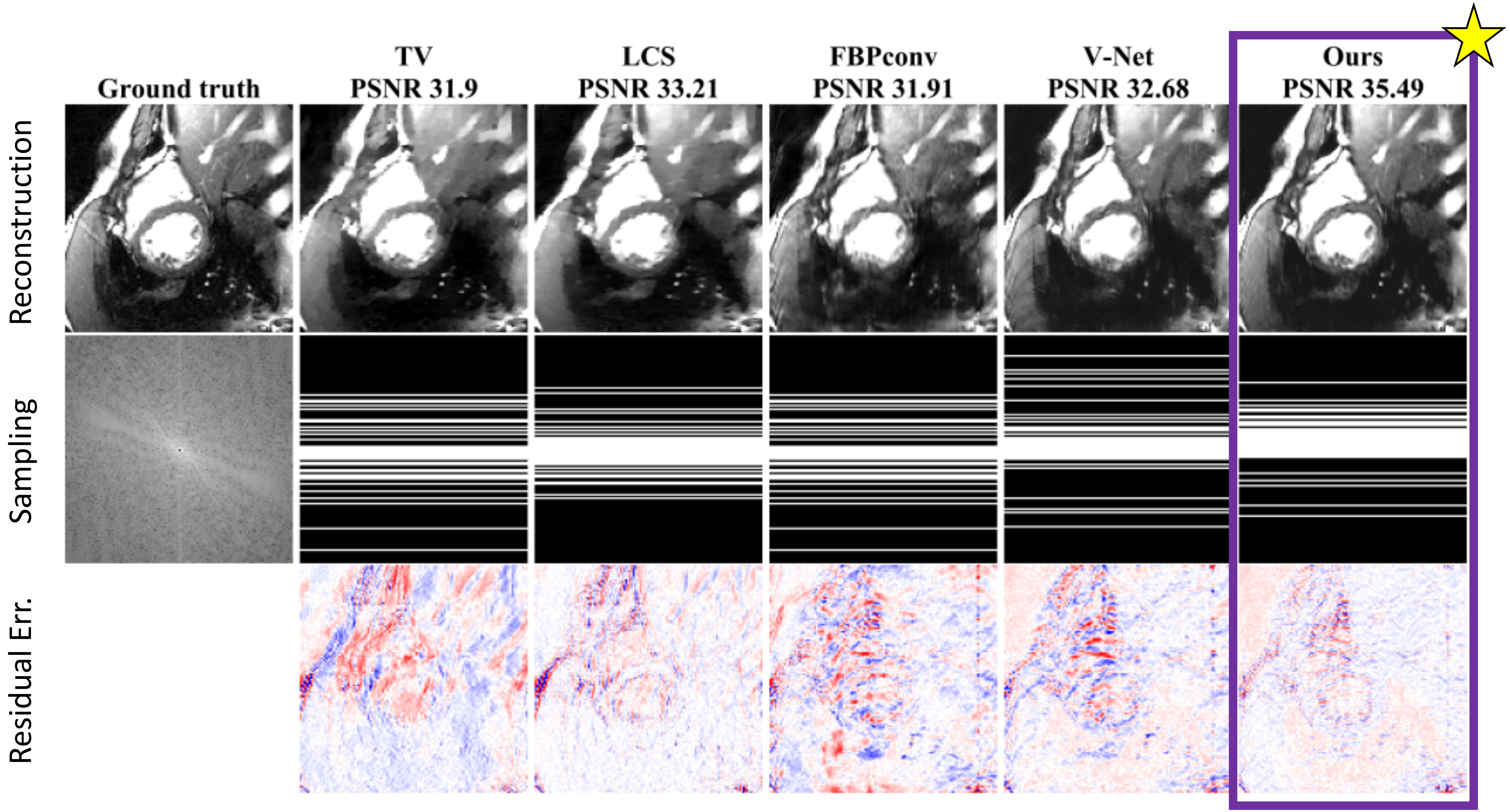
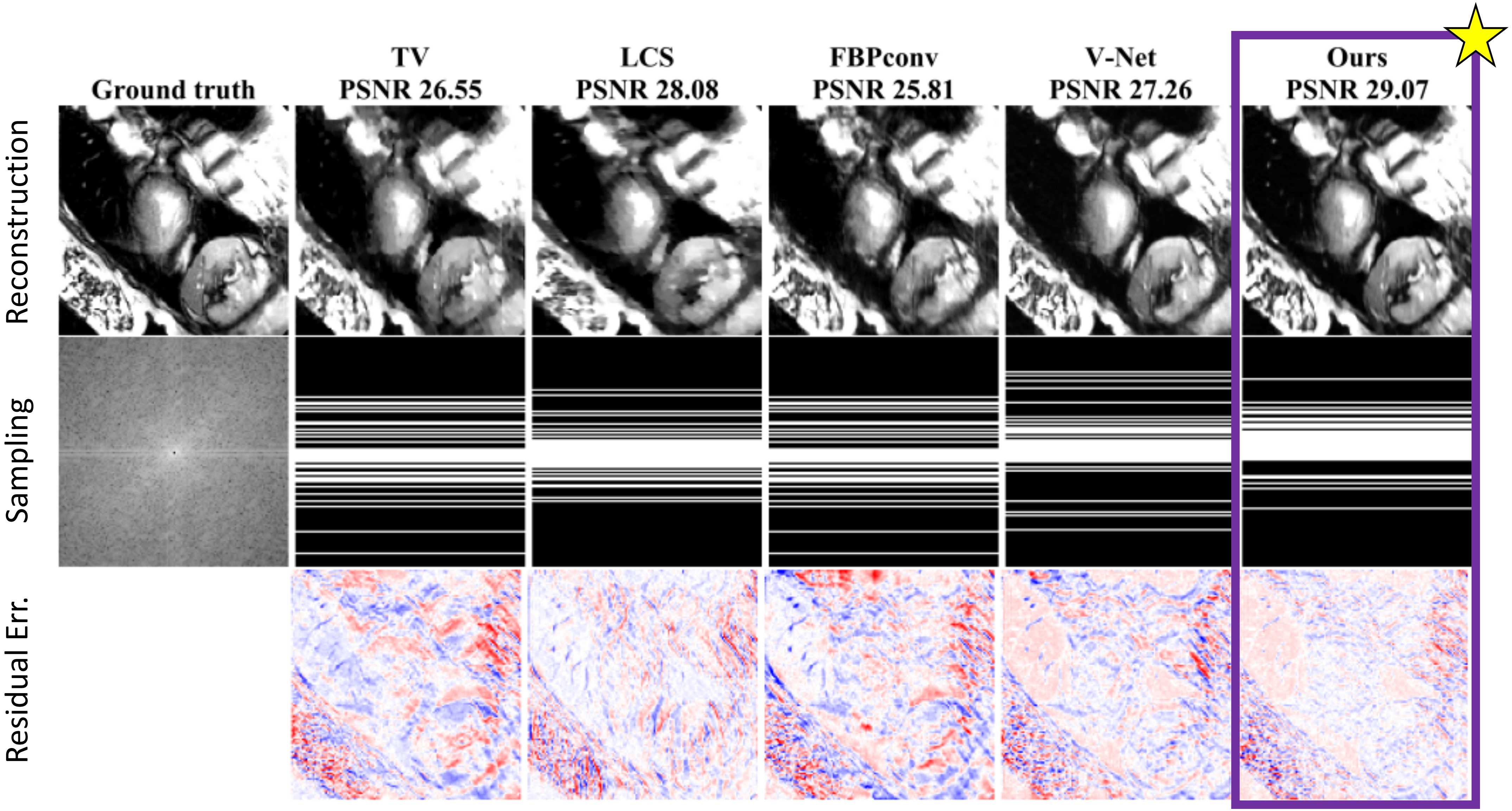


Fig. 6. Architecture of the two networks. Note that the sampling network (SampleNet) takes as input the output of the reconstruction network (ReconNet). Both of our networks are based on residual blocks [2].

RESULTS


Ground truth
TV
PSNR 31.9
LCS
PSNR 33.21
FBPconv
PSNR 31.91
V-Net
PSNR 32.68
Ours
PSNR 35.49
Reconstruction
Sampling
Residual Err.


Ground truth
TV
PSNR 26.55
LCS
PSNR 28.08
FBPconv
PSNR 25.81
V-Net
PSNR 27.26
Ours
PSNR 29.07
Reconstruction
Sampling
Residual Err.

“Note that our method does not require per-dataset parameter tuning and is able to outperform the state of the art with a single hyper-parameter setup. By contrast, methods relying on TV require per-dataset tuning.”

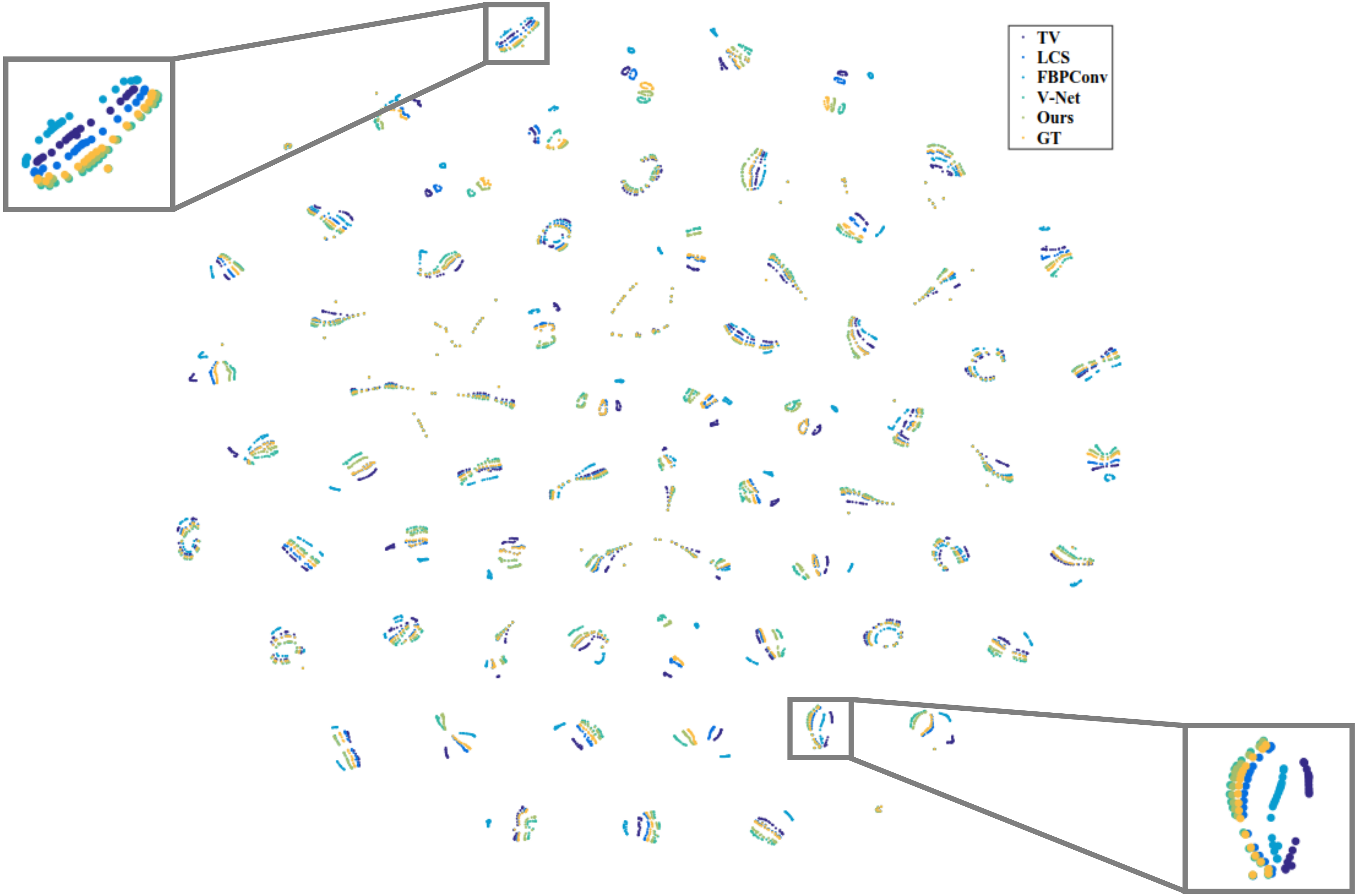


Fig. 9. t-SNE [36] embedding of the reconstructions for the testset of the *cardiac* dataset.

TABLE 3
Quantitative results in terms of PSNR, with different MCTS α values. Best results are shown in bold.

minimax backup α	*cardiac*
0	32.99
0.5	**34.22**
1	28.96

TABLE 4
Quantitative results in terms of PSNR, with ReconNet (our method) and without ReconNet. Best results are shown in bold.

Dataset	Without ReconNet	Our Method
cardiac	33.53	**34.22**

Coupled network results in highest reconstructive performance.

Notables & Discussion

- The simultaneously trained network (***ReconNet*** & ***SampleNet***) is observed to outperform ***ReconNet*** trained on the k space samples alone.

- The network is **not image specific** (like wavelet) and still allows for higher PSNR.

- The approach has the potential to be adopted into ***many*** imaging modalities.

QUESTIONS