Deep Residual Learning for Model-Based Iterative CT Reconstruction using Plug-and-Play Framework
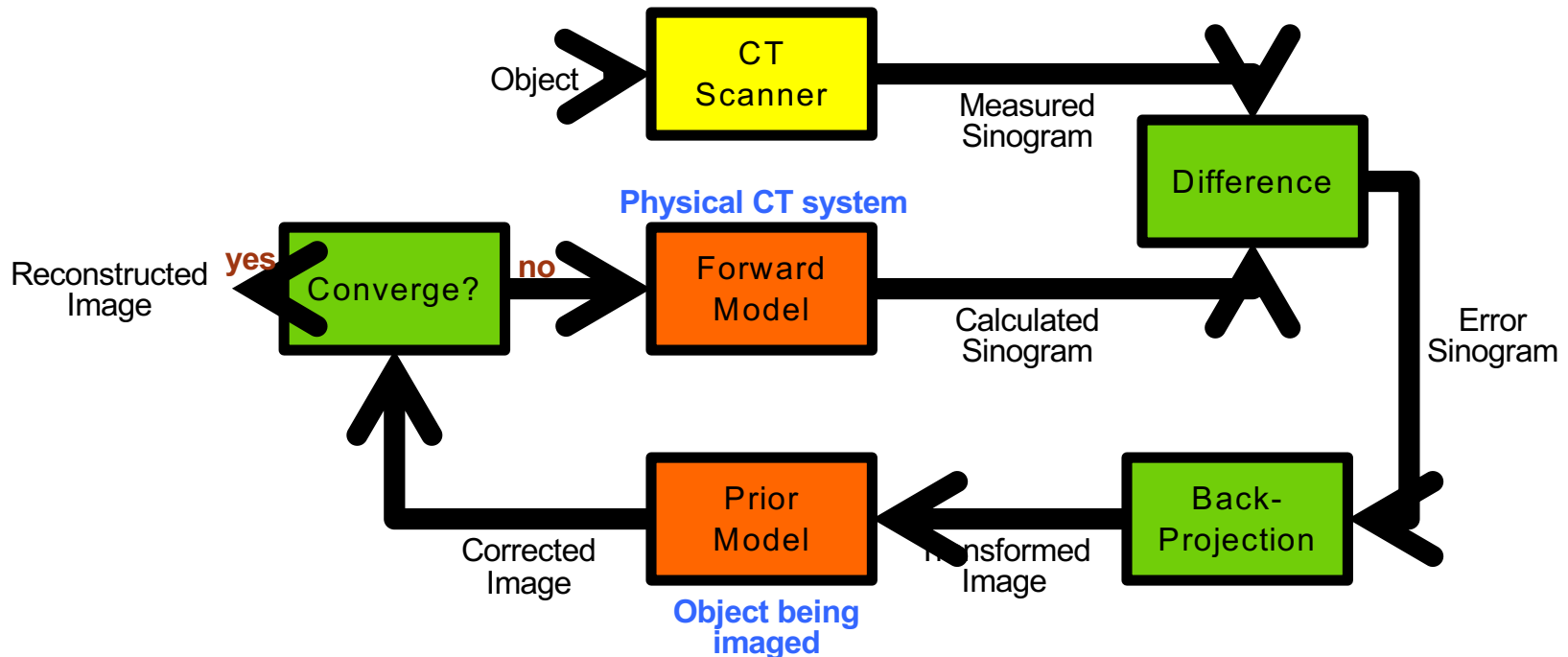
Presenter:
Dong Hye Ye

Dong Hye Ye, Somesh Srivastava, Jean-Baptiste Thibault, Ken Sauer, Charles Bouman
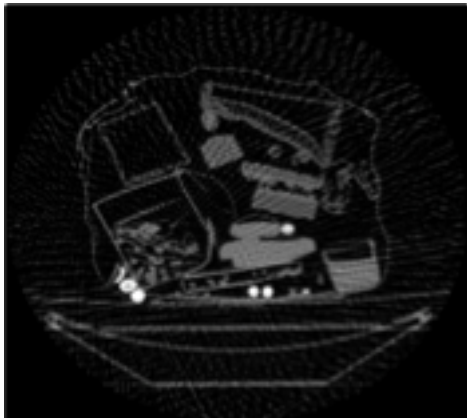
April 18, 2018

Model-Based Iterative Reconstruction (MBIR)

- Computed Tomography (CT) Reconstruction
  - ➤ Diagnostic Radiology
  - ➤ Additive Manufacturing Inspection

- MBIR Flowchart

Object → **CT Scanner** → Measured Sinogram → **Difference**

Physical CT system

Reconstructed Image ← **yes** — **Converge?** — **no** → **Forward Model** → Calculated Sinogram → **Difference**

Error Sinogram

Corrected Image ← **Prior Model** ← Transformed Image ← **Back-Projection**

Object being imaged

Advantage of MBIR over Filtered-Back Projection (FBP)

- Superior Image Quality: Low Noise and High Resolution

	FBP	MBIR
Low-dose (quarter dose)		
Sparse-view (16 views)		

Prior Model in MBIR

- Accurate prior modeling is critical to the image quality of MBIR.

- Typical Prior Model: MRF
  - Penalize intensity fluctuation in the neighborhood
  - Challenge: Noise-induced fluctuation vs. underlying object

- Solution: Prior Model from an Image Database
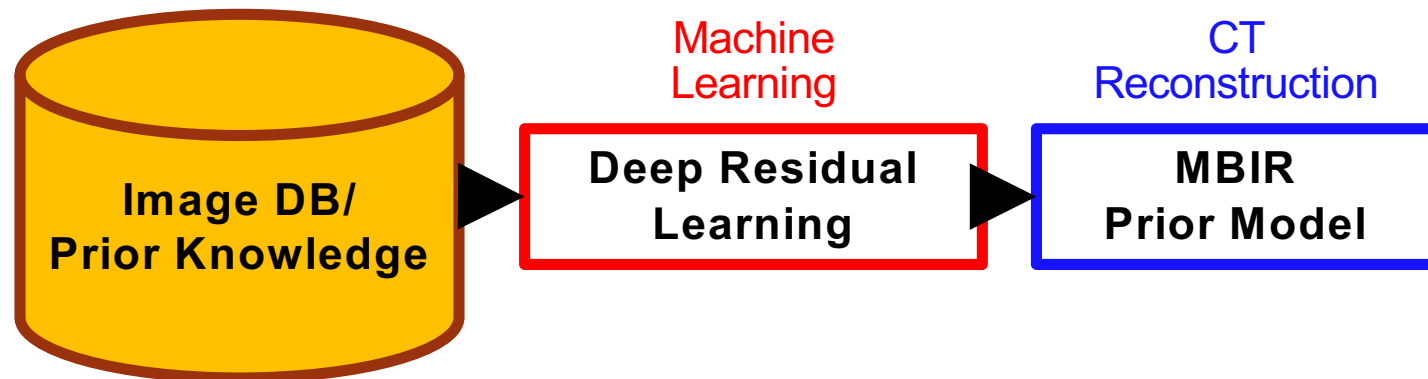
MBIR Optimization

- MAP Estimation

$$\hat{x} = \arg\min_x \left[\boxed{\frac{1}{2}\|\mathbf{A}x - y\|_{\mathbf{W}}^2} + \frac{1}{2\sigma^2}\boxed{\Phi(x)} \right]$$

Forward Model **Prior Model**

$x \in \mathbb{\sim}^N$: reconstructed image, $y \in \mathbb{\sim}^M$: measured CT scan

$\mathbf{A} \in \mathbb{\sim}^{M \times N}$: system matrix for CT scan, $\mathbf{W} \in \mathbb{\sim}^{M \times M}$: measurement noise variance

$\sigma \in \mathbb{\sim}$: regularization parameter, Φ: $\mathbb{\sim}^N \to \mathbb{\sim}$: prior model

- First-order iterative optimization
  - ➤ Iterative Coordinate Descent (ICD) / Ordered Subset (OS)
  - ➤ Prior model should be first-order differentiable.

Not flexible for Data-driven Prior

Plug-and-Play (PnP) Framework

- Variable Splitting

Separate variable

$$(\hat{x}, \hat{v}) = \arg\min_{x,v} \left[\frac{1}{2}\|\mathbf{A}x - y\|_{\mathbf{W}}^2 + \frac{1}{2\sigma^2}\boxed{\Phi(v)} \right] \quad \text{s.t. } x = v$$

- Alternating Direction Method of Multipliers (ADMM)

$$L_\lambda(x, v, u) = \frac{1}{2}\|\mathbf{A}x - y\|_{\mathbf{W}}^2 + \frac{1}{2\sigma^2}\Phi(v) + \frac{1}{2\lambda^2}\|x - v + u\|_2^2$$

Augmented Lagrangian

Step 1: Reconstruction Module

$$\hat{x} = \arg\min_x L_\lambda(x, \hat{v}, u) = \arg\min_x \left[\frac{1}{2}\|\mathbf{A}x - y\|_{\mathbf{W}}^2 + \frac{1}{2\lambda^2}\|x - \hat{v} + u\|_2^2 \right]$$

Step 2: De-noising Module

$$\hat{v} = \arg\min_v L_\lambda(\hat{x}, v, u) = \arg\min_v \left[\frac{1}{2\sigma^2}\Phi(v) + \frac{1}{2\lambda^2}\boxed{\|\hat{x} + u} - v\|_2^2 \right]$$

**Step 3: Update Dual Variable**

$$u \leftarrow u + (\hat{x} - \hat{v})$$

Input noisy image

Independent Module for De-noising

Deep Residual Learning for De-noising

- Deep Neural Network
  - Powerful performance for vision tasks such as de-noising
  - Weights of a neural network learned on large training dataset
  - Challenge: Long training time

- Deep Residual Learning for Efficient Training
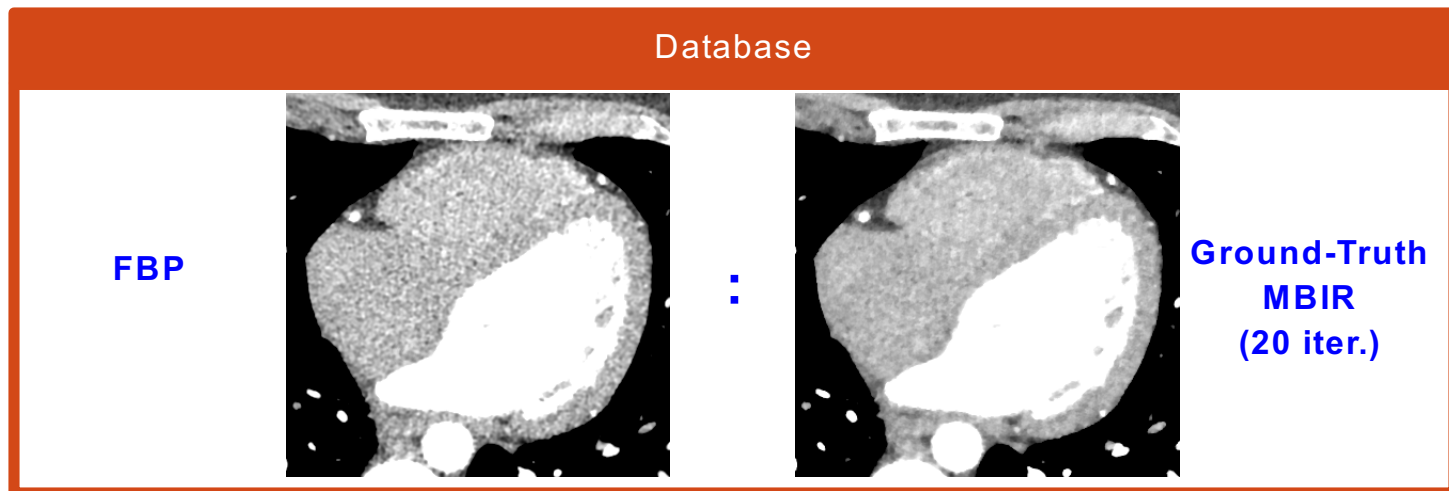 - Bypassing low-freq. image
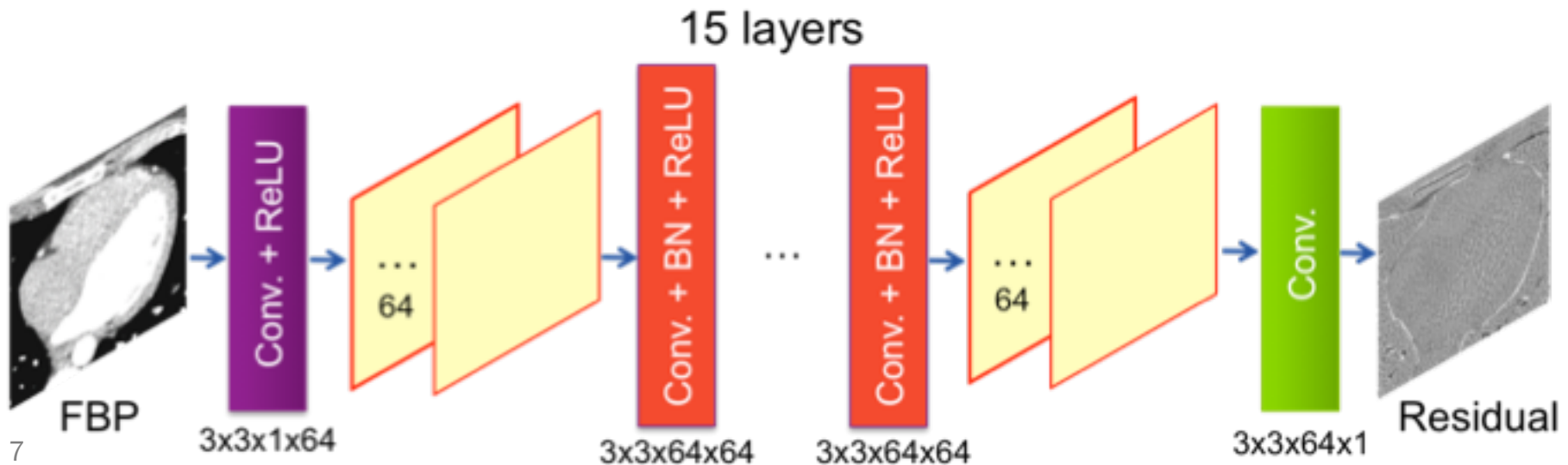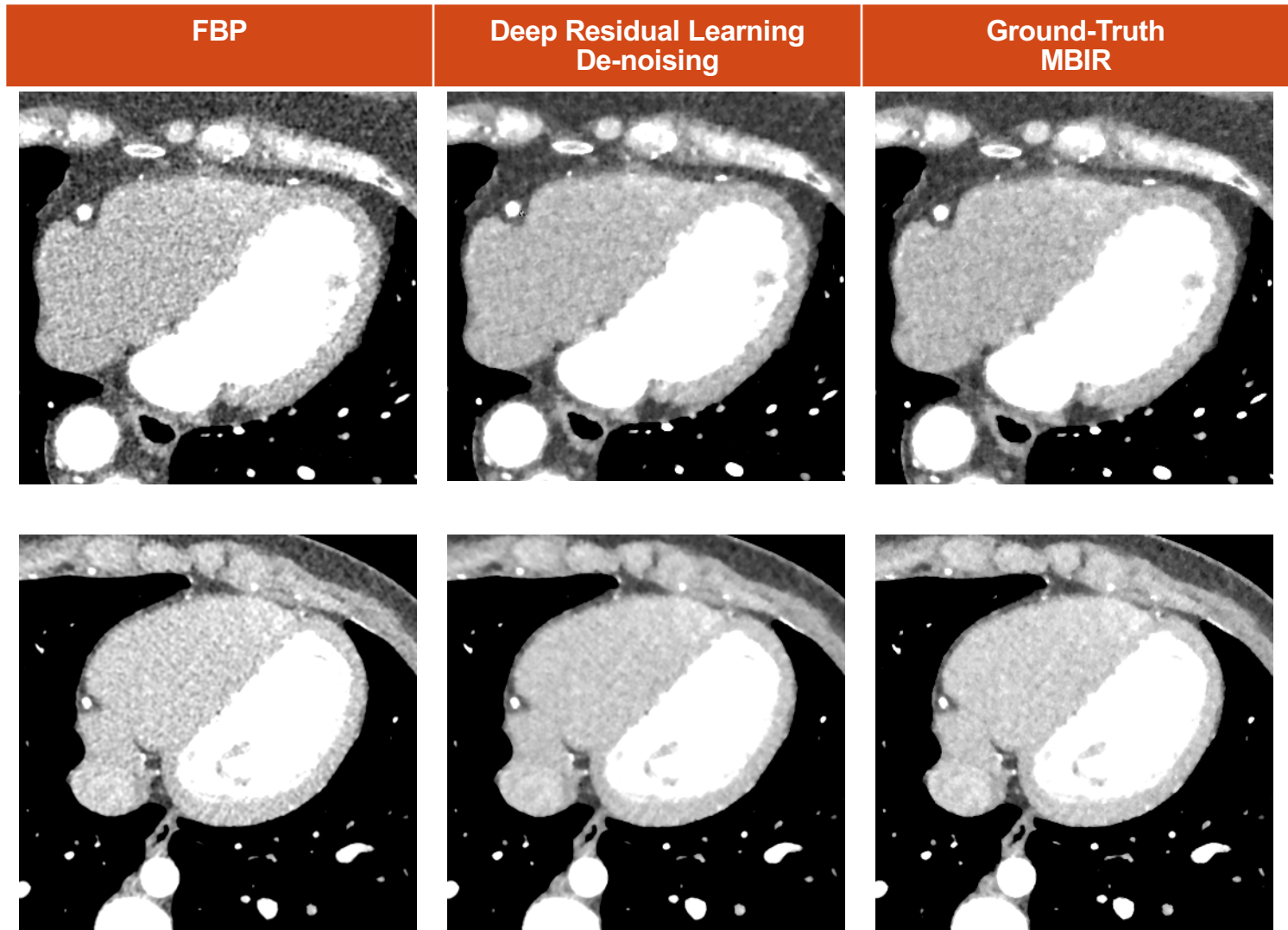
Deep Residual Learning: Training

- Cost

$$l(\Theta) = \frac{1}{2K} \sum_{k=1}^{K} \left\| R(\tilde{v}_k^{tr}; \Theta) - (\hat{v}_k^{tr} - \tilde{v}_k^{tr}) \right\|_2^2$$

$\{\tilde{v}_k^{tr}, \hat{v}_k^{tr}\}$: noisy and clean image pairs, $R(\bullet; \Theta)$: Residual neural network
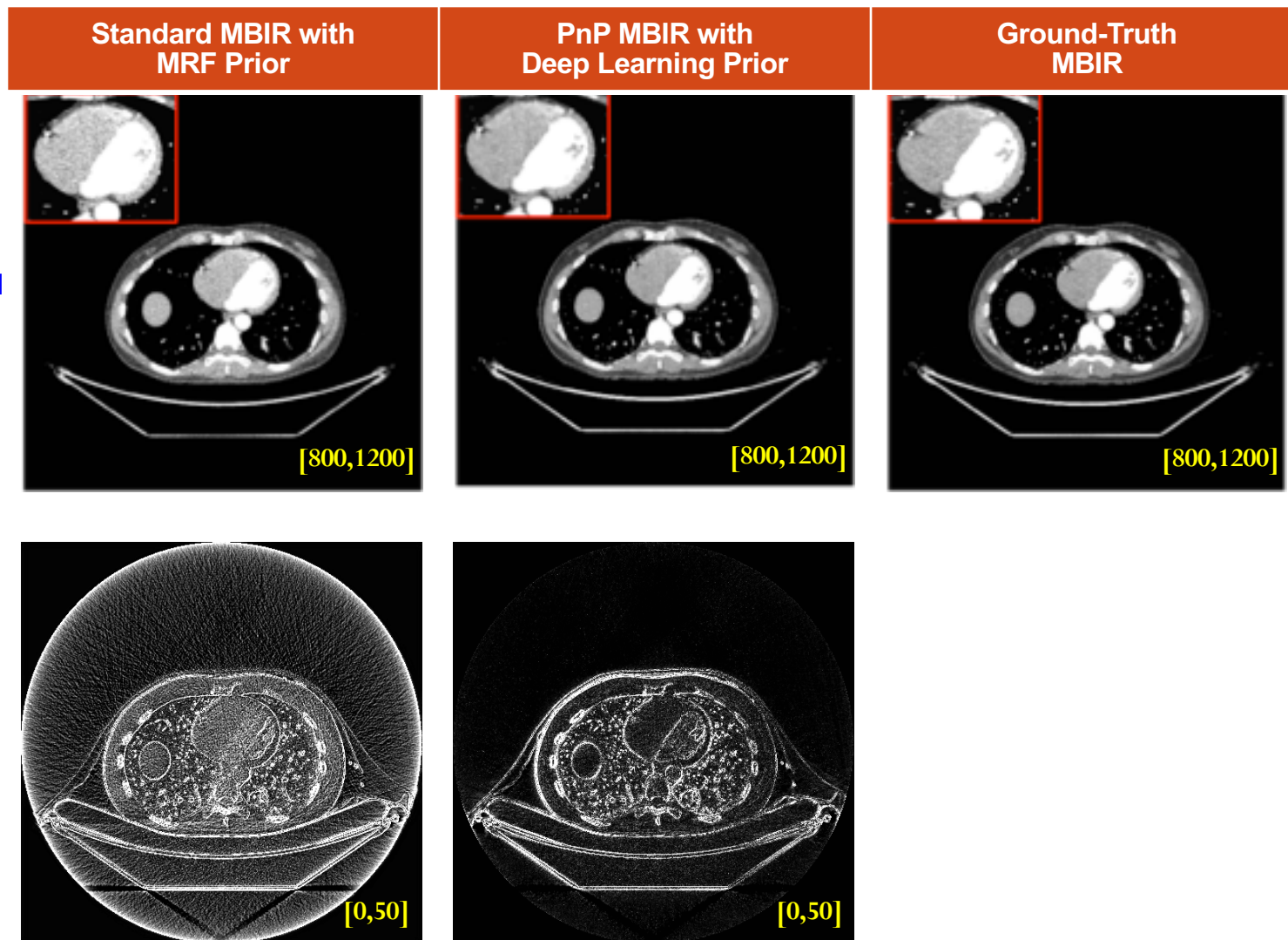
- Training Database
  - ➤ 40x40 patches for all slices, Data augmentation (flip, rotation)
  - ➤ Randomly selected 256000 patches, mini-batch size: 128
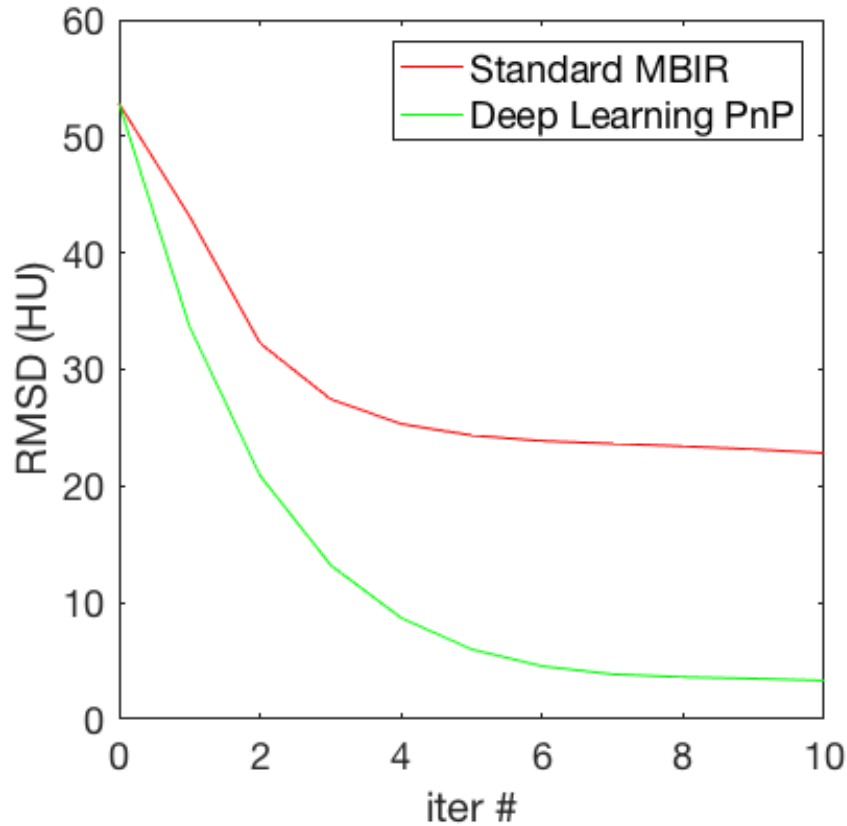

Database

FBP : **Ground-Truth MBIR (20 iter.)**

Deep Residual Learning: Testing

FBP	Deep Residual Learning De-noising	Ground-Truth MBIR

Testing Scan #1

Testing Scan #2

MBIR Result: Qualitative @ 1 iter.

Standard MBIR with MRF Prior	PnP MBIR with Deep Learning Prior	Ground-Truth MBIR

Reconstructed Image

[800,1200] — [800,1200] — [800,1200]

Difference Image

[0,50] — [0,50]

MBIR Result: RMSD

Significant Speed-up with Faster Convergence

Computational Time

- Deep Learning Training Time
  - 4 NVIDIA Titan X GPU (12GB memory)
  - Google Tensor Flow
  - 65 minutes / 50 epochs

- Deep Learning Testing Time
  - ~10ms/slice

- Standard MBIR and PnP MBIR require similar amount of recon. time per iteration.

Conclusion

- Summary
  1. Image prior modeling from FBP/MBIR database
     - Deep Residual Learning for Image De-noising
  2. Incorporating the prior model from a database into MBIR
     - Plug-and-Play Optimization Framework

- Deep Residual Learning is effective in reducing the noise and enhancing the resolution in FBP.

- PnP MBIR with deep learning prior significantly improves the image quality compared with standard MBIR.