JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Solving 3D Inverse Problems using Pre-trained 2D Diffusion Models

Hyungjin Chung

Bio Imaging, Signal Processing and Learning lab (BISPL)

KAIST

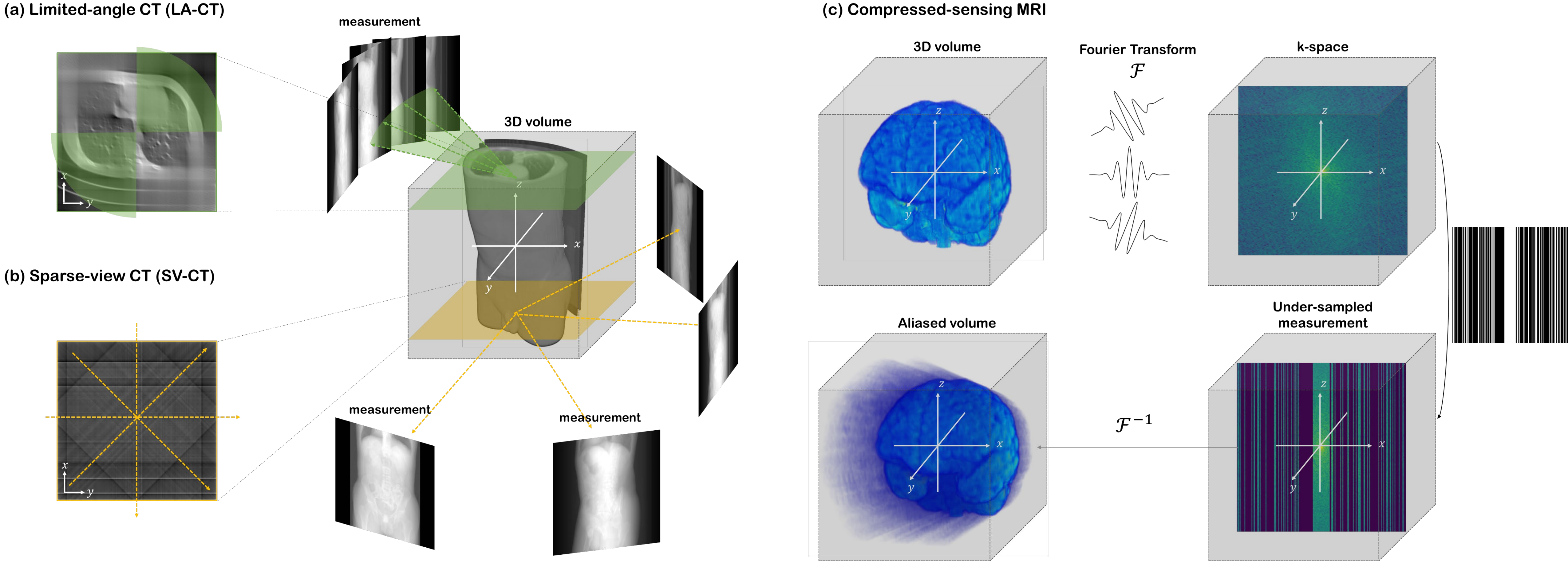
Dohoon Ryu

Michael T. Mccann

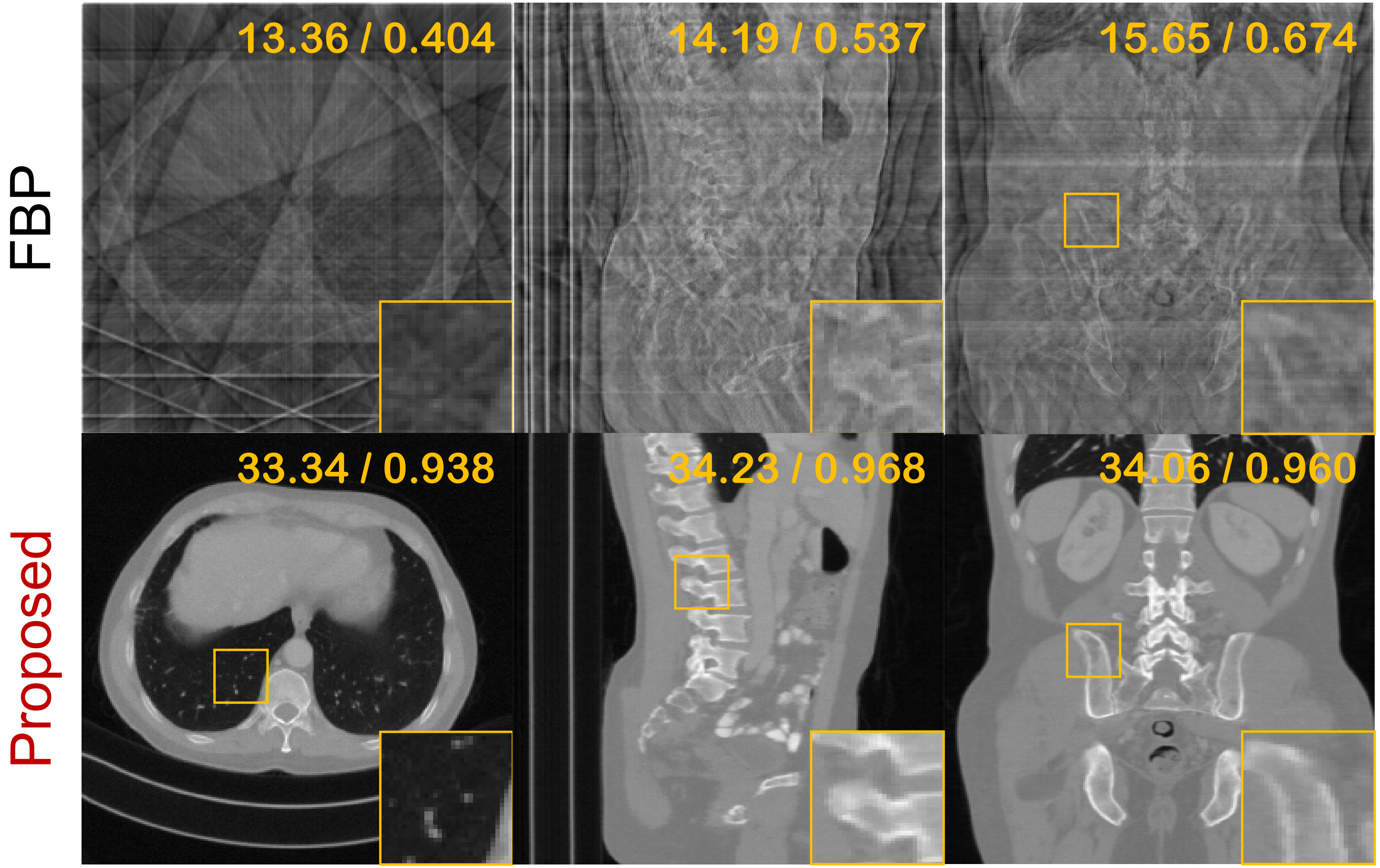
Marc L. Klasky

Jong Chul Ye

3D inverse problems in medical imaging

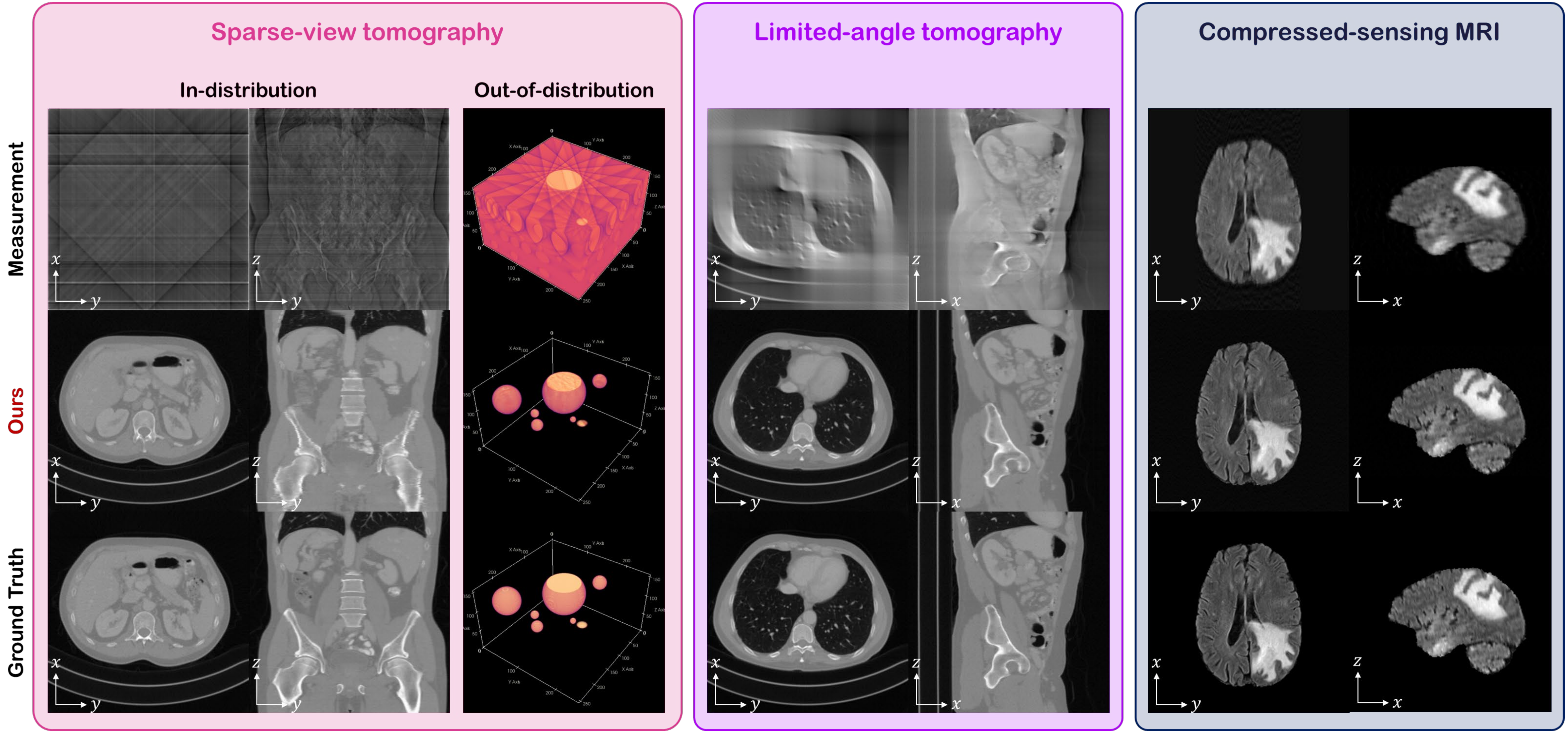


Results (8-view sparse CT)



Coherent results across the whole volume

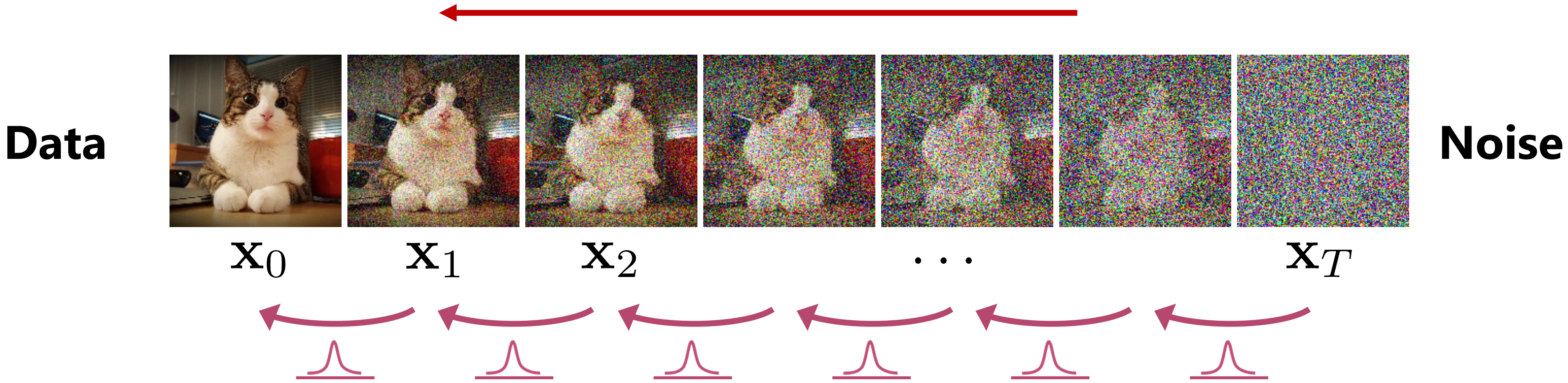
Results: general inverse problem solver



Diffusion models

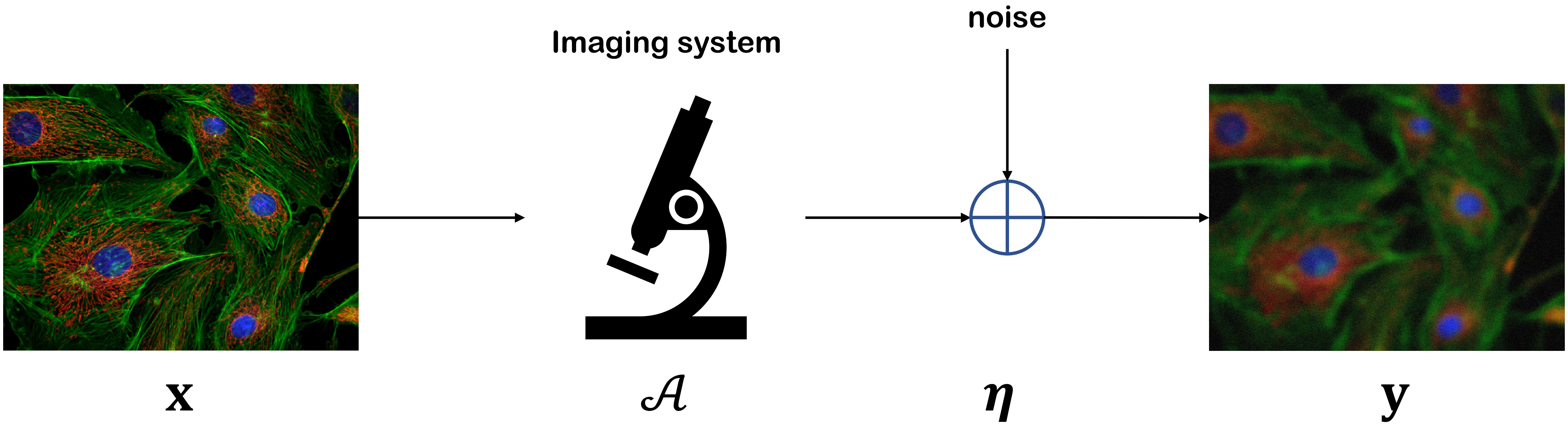
Reverse denoising process

Generate data from noise by denoising, learned



$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), r_t^2 \boldsymbol{I})$$

Inverse problems

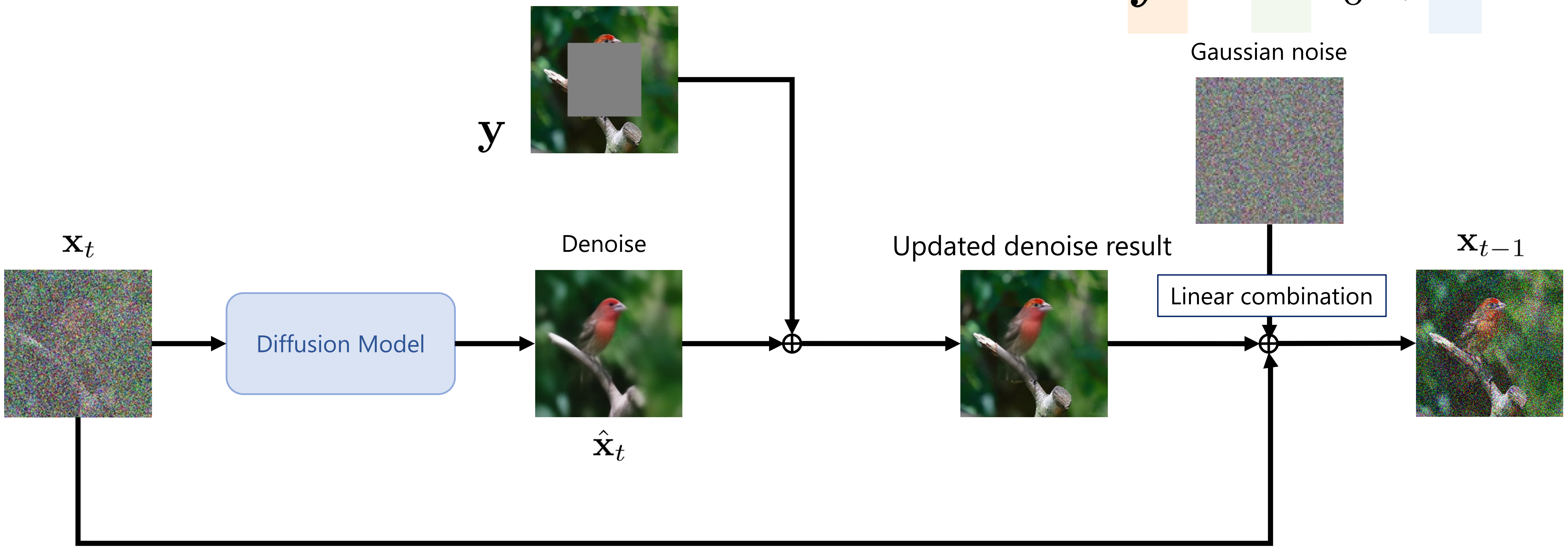


Imaging systems cast as the above **forward model**

- Acquiring the image $\mathbf{x}$: **inverse problem**
- System naturally **ill-posed**: what is the best solution?
- Ex: microscopy, MRI, CT, optics, etc.

Diffusion models for inverse problems (DIS)

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$



Wang *et al.*, "Zero-shot image restoration using denoising diffusion null-space models", ICLR 2023

Diffusion models & Bayesian inference

Reverse SDE

x_T → $dx = [\bar{f}(x,t) - \bar{g}(t)^2 \nabla_x \log p_t(x)]dt + \bar{g}(t)dw$ → x_0

x_T ← $dx = \bar{f}(x,t)dt + \bar{g}(t)dw$ ← x_0

Forward SDE

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

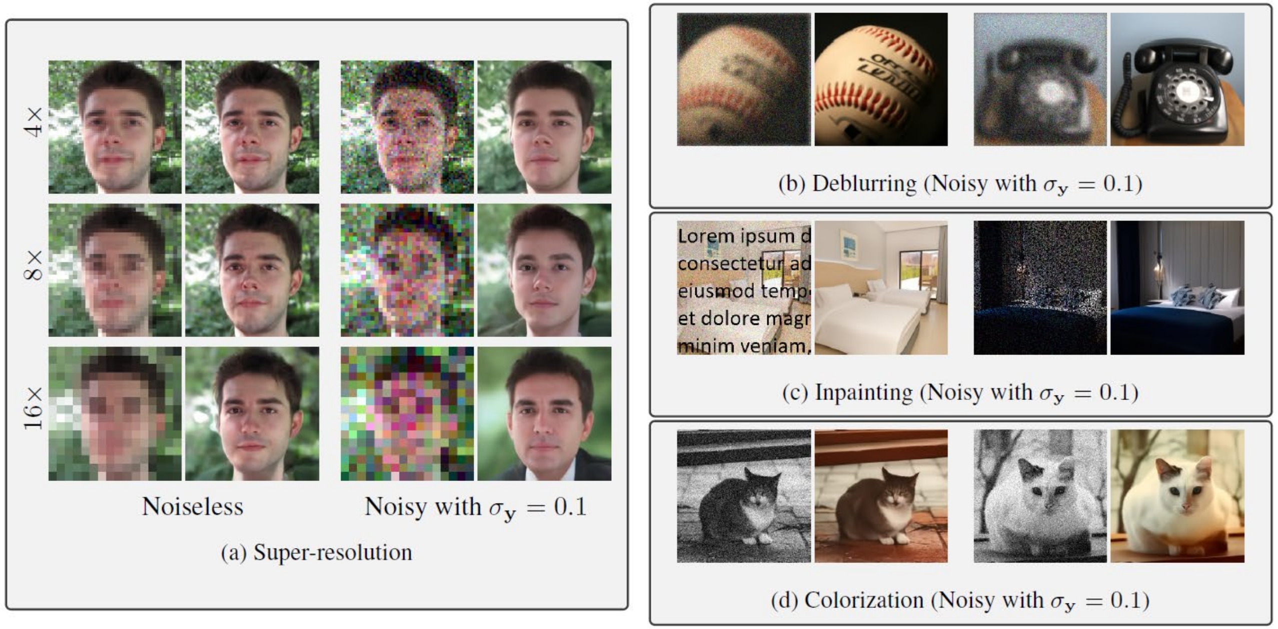
Measurement process

Denoiser

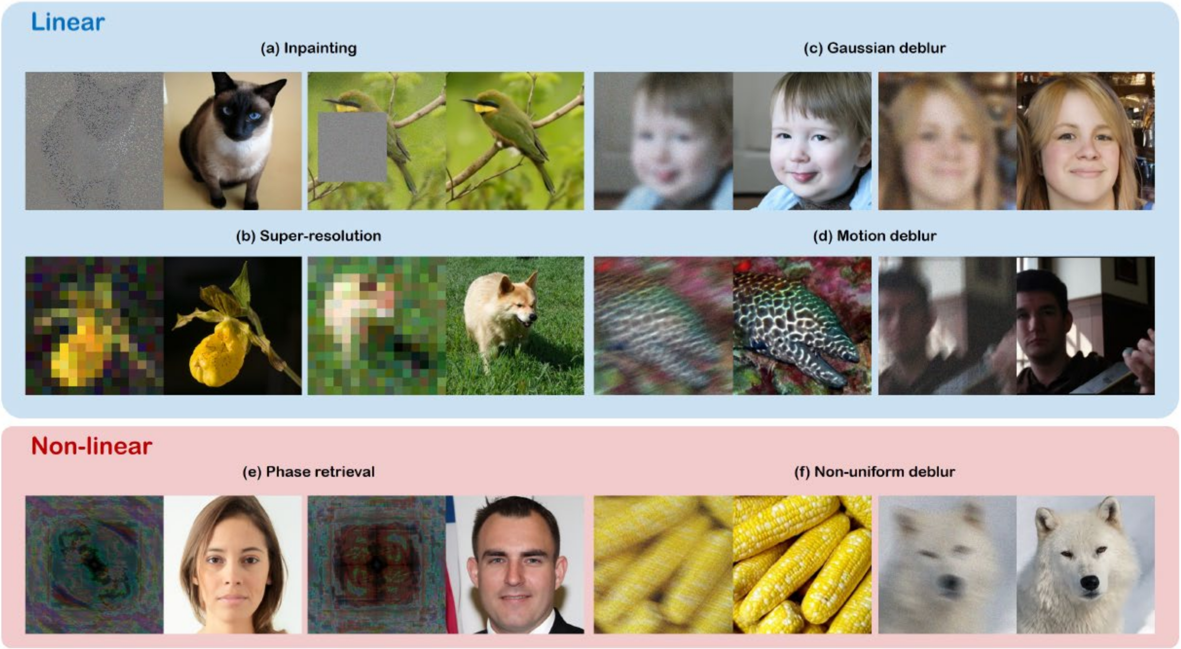
Song *et al.*, "Score-based generative modeling through stochastic differential equations", ICLR 2021

DIS heavily focused on 2D problems

DDRM



DPS

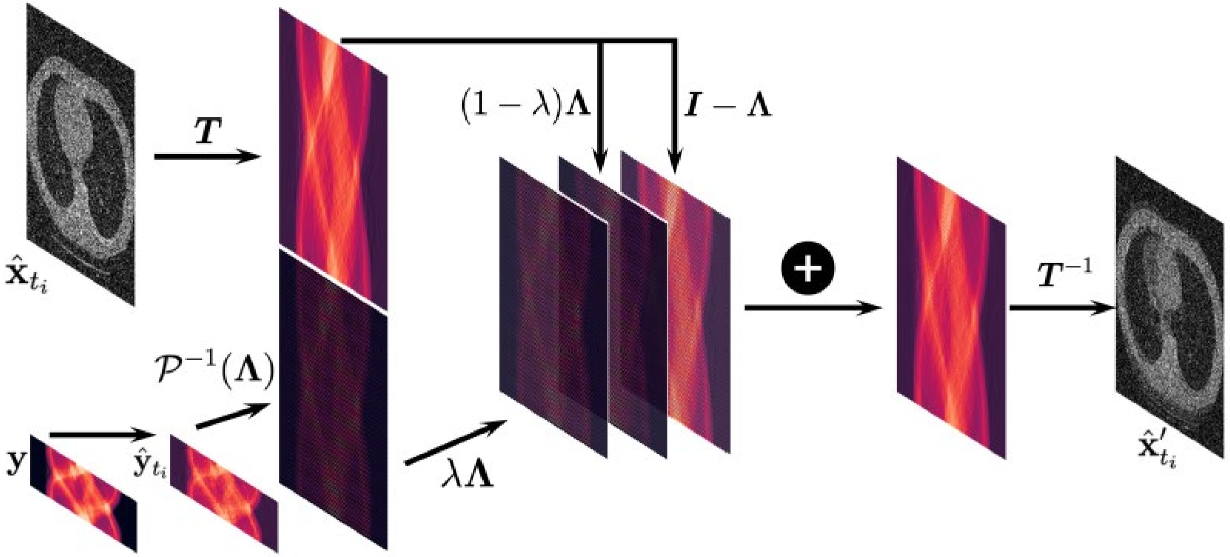


Kawar *et al*., "Denoising Diffusion Restoration Models", NeurIPS 2022

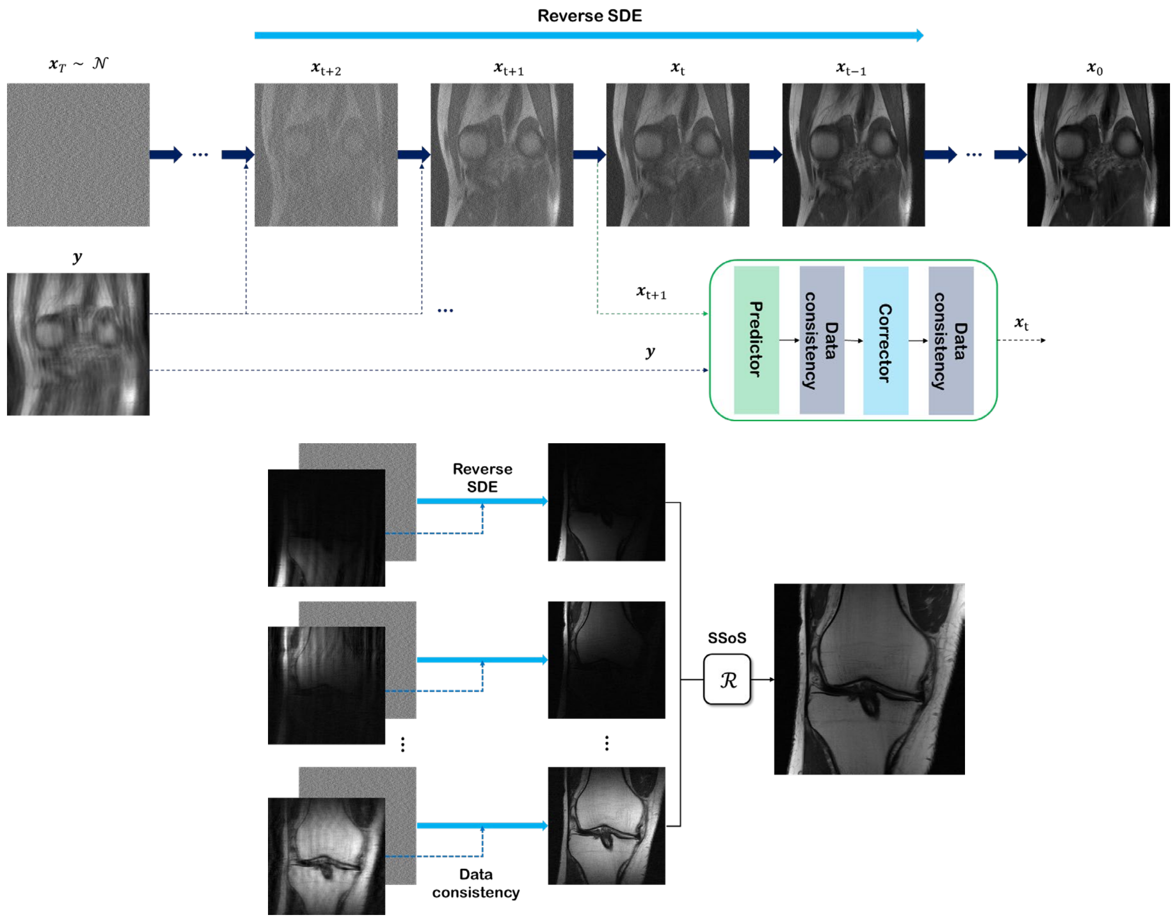
Chung *et al*., "Diffusion Posterior Sampling for General Noisy Inverse Problems", ICLR 2023

DIS heavily focused on 2D problems

Song *et al.*



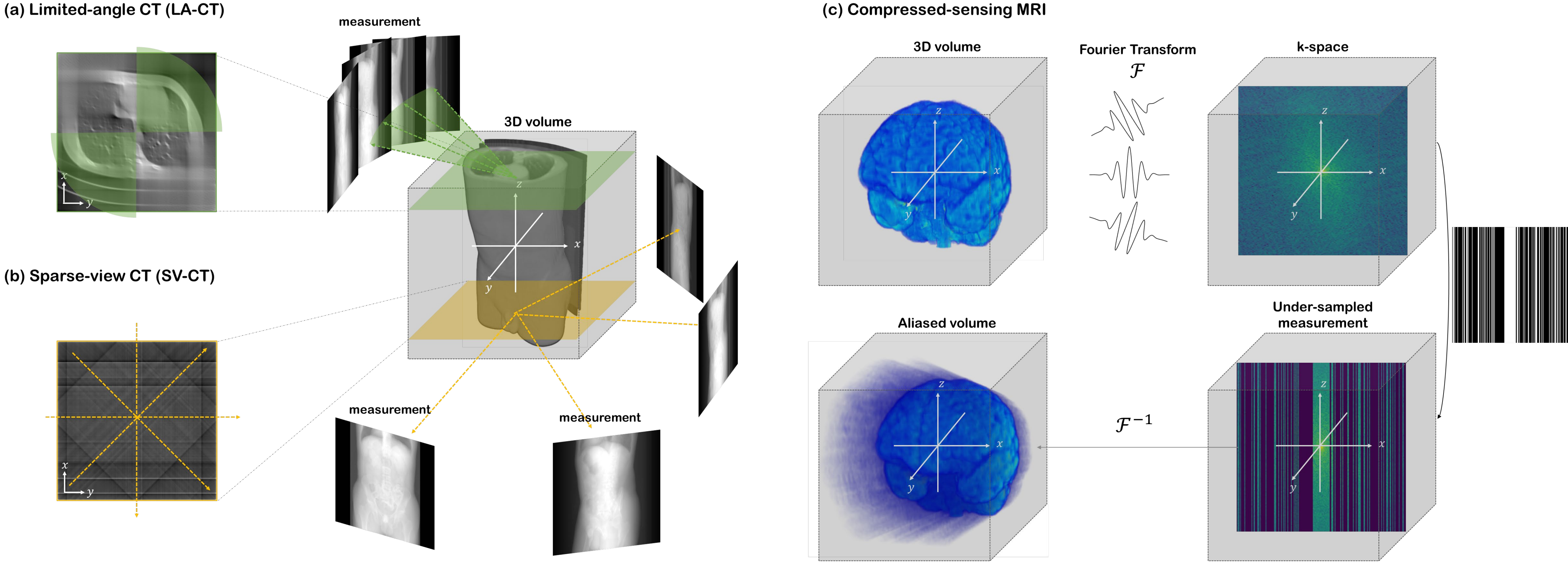
Score-MRI



Song *et al.*, "Solving inverse problems in medical imaging with score-based generative models", ICLR 2022

Chung *et al.*, "Score-based diffusion models for accelerated MRI", MeDIA 2022

Practical 3D Inverse Problems



3D problems?

- 3D representations are **memory heavy**
  - **Voxels**: Hard to deal with > 128^3 data
  - **Point clouds / Mesh / NeRF**: Compact representation, but not suitable for medical imaging inverse problems
- **3D voxel diffusion**?
  - The whole diffusion process **stays in the data dimension**
  - **Computationally too heavy**

DiffusionMBIR: factored prior

Posterior   Likelihood (given)

$$\overbrace{p_\theta(\mathbf{x}|\mathbf{y})}^{\text{Posterior}} \propto \underbrace{p_\theta(\mathbf{x})}_{\text{Prior}} \overbrace{p(\mathbf{y}|\mathbf{x})}^{\text{Likelihood}}$$

Prior (our interest)

DiffusionMBIR: factored prior

$$\overbrace{p_\theta(\mathbf{x}|\mathbf{y})}^{\text{Posterior}} \propto \underbrace{p_\theta(\mathbf{x})}_{\textbf{Prior}\text{ (our interest)}} \overbrace{p(\mathbf{y}|\mathbf{x})}^{\text{Likelihood (given)}}$$

$$p_\theta(\mathbf{x}) := p_\theta^{xy}(\mathbf{x}) p^z(\mathbf{x})$$

Proposal: factored prior

Diffusion + model-based prior

$$p_\theta(\mathbf{x}) := p_\theta^{xy}(\mathbf{x})p^z(\mathbf{x})$$

Proposal: factored prior

$$\nabla_\mathbf{x} \log p_\theta^{xy}(\mathbf{x}) \simeq s_{\theta^*}(\mathbf{x})$$

Generative diffusion prior (xy):
Pre-trained 2D diffusion model

$$-\log p^z(\mathbf{x}) \simeq TV^z(\mathbf{x})$$

Model-based prior (z):
Total-variation to impose smoothness

Reverse diffusion + ADMM optimization

1. Denoising with score function (**parallel**)

$$\mathbf{x}'_i \leftarrow \text{Denoise}(\mathbf{x}'_{i+1}, \boldsymbol{s}_{\theta^*})$$

2. Data consistency + TV prior augmenting (**joint**)

$$\mathbf{x}_\mathrm{i} \leftarrow \underset{\mathbf{x}'_i}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \boldsymbol{A}\mathbf{x}'_i\|_2^2 + \|\boldsymbol{D}_z \mathbf{x}'_i\|_1$$

Reverse diffusion + ADMM optimization

1. **Denoising** with score function (**parallel**)

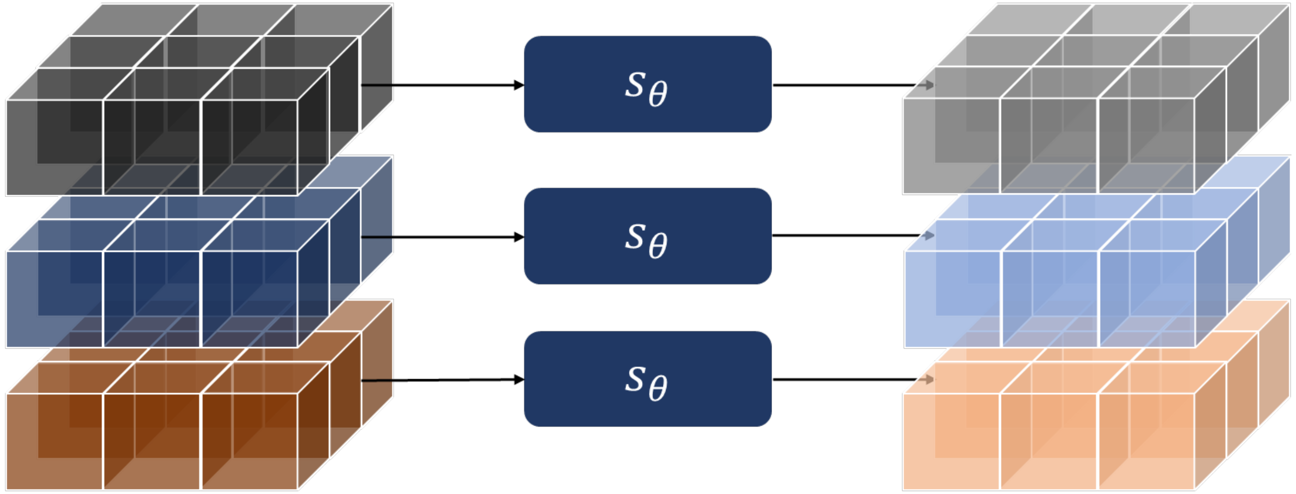
$$\mathbf{x}'_i \leftarrow \text{Denoise}(\mathbf{x}'_{i+1}, \overbrace{\boldsymbol{s}_{\theta^*}}^{\nabla_{\mathbf{x}} \log p_\theta^{xy}(\mathbf{x})})$$

2. Data consistency + TV prior augmenting (**joint**)

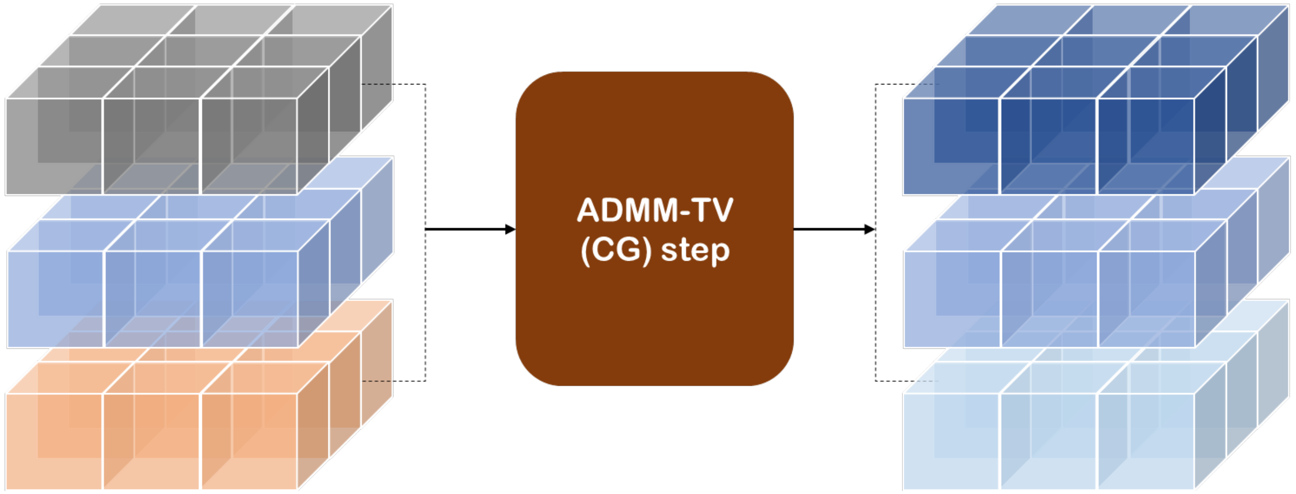
$$\mathbf{x}_i \leftarrow \underset{\mathbf{x}'_i}{\text{argmin}} \frac{1}{2} \underbrace{\|\mathbf{y} - \boldsymbol{A}\mathbf{x}'_i\|_2^2}_{-\log p(\mathbf{y}|\mathbf{x})} + \underbrace{\|\boldsymbol{D}_z \mathbf{x}'_i\|_1}_{-\log p^z(\mathbf{x})}$$

Reverse diffusion + ADMM optimization

1. Score function denoising (parallel)



2. ADMM-TV (joint)



Fast DiffusionMBIR

Algorithm 1 DiffusionMBIR concept

Require: s_{θ^*}, N

1: $\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \sigma_T^2 \boldsymbol{I})$
2: **for** $i = N-1 : 0$ **do** ▷ Reverse diffusion
3: $\quad \boldsymbol{x}'_i \leftarrow \text{Denoise}(\boldsymbol{x}_{i+1}, s_{\theta^*})$
4: $\quad \boldsymbol{x}_i \leftarrow \text{argmin}_{\boldsymbol{x}'_i} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}'_i\|_2^2 + \|\boldsymbol{D}_z \boldsymbol{x}'_i\|_1$
5: **end for**
6: **return** $\boldsymbol{x}_0$

Running optimization per denoising step would be too costly

Fast DiffusionMBIR

Algorithm 2 Diffusion-MBIR (fast; variable sharing)

Require: $s_\theta, N, \lambda, \rho, \{\sigma_i\}$

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$

$\boldsymbol{z}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$

$\boldsymbol{w}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$

for $i = N-1 : 0$ **do** ▷ SDE iteration

  $\bar{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) \boldsymbol{s}_\theta(\boldsymbol{x}_{i+1}, \sigma_{i+1})$

  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  $\bar{\boldsymbol{x}}_i \leftarrow \bar{\boldsymbol{x}}_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\,\epsilon$

  $\boldsymbol{A}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{A} + \rho \boldsymbol{D}_z^T \boldsymbol{D}_z$

  $\boldsymbol{b}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{y} + \rho \boldsymbol{D}_z^T (\boldsymbol{z}_{i+1} - \boldsymbol{w}_{i+1})$

  $\boldsymbol{x}_i \leftarrow \text{CG}(\boldsymbol{A}_{\text{CG}}, \boldsymbol{b}_{\text{CG}}, 1)$

  $\boldsymbol{z}_i \leftarrow \mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z \boldsymbol{x}_i + \boldsymbol{w}_{i+1})$

  $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i+1} + \boldsymbol{D}_z \boldsymbol{x}_i - \boldsymbol{z}_i$

end for

return x_0

Sharing primal/dual variables

- Warm start
- Much faster convergence

Fast DiffusionMBIR

Algorithm 2 Diffusion-MBIR (fast; variable sharing)

Require: $s_\theta, N, \lambda, \rho, \{\sigma_i\}$

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$

$\boldsymbol{z}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$

$\boldsymbol{w}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$

for $i = N - 1 : 0$ **do** ▷ SDE iteration

$\quad \bar{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) \boldsymbol{s}_\theta(\boldsymbol{x}_{i+1}, \sigma_{i+1})$

$\quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\quad \bar{\boldsymbol{x}}_i \leftarrow \bar{\boldsymbol{x}}_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\, \epsilon$

$\quad \boldsymbol{A}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{A} + \rho \boldsymbol{D}_z^T \boldsymbol{D}_z$

$\quad \boldsymbol{b}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{y} + \rho \boldsymbol{D}_z^T (\boldsymbol{z}_{i+1} - \boldsymbol{w}_{i+1})$

$\quad \boldsymbol{x}_i \leftarrow \text{CG}(\boldsymbol{A}_{\text{CG}}, \boldsymbol{b}_{\text{CG}}, 1)$

$\quad \boldsymbol{z}_i \leftarrow \mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z \boldsymbol{x}_i + \boldsymbol{w}_{i+1})$

$\quad \boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i+1} + \boldsymbol{D}_z \boldsymbol{x}_i - \boldsymbol{z}_i$

end for

return $\boldsymbol{x}_0$

Sharing primal/dual variables

- Warm start
- Much faster convergence

Score update (denoising)

- Prior in xy dimension

Fast DiffusionMBIR

Algorithm 2 Diffusion-MBIR (fast; variable sharing)

Require: $s_\theta, N, \lambda, \rho, \{\sigma_i\}$

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$
$\boldsymbol{z}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$
$\boldsymbol{w}_N \leftarrow$ `torch.zeros_like`$(\bar{\boldsymbol{x}}_i)$
for $i = N-1 : 0$ **do** ▷ SDE iteration
  $\bar{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) \boldsymbol{s}_\theta(\boldsymbol{x}_{i+1}, \sigma_{i+1})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\bar{\boldsymbol{x}}_i \leftarrow \bar{\boldsymbol{x}}_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\,\boldsymbol{\epsilon}$
  $\boldsymbol{A}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{A} + \rho \boldsymbol{D}_z^T \boldsymbol{D}_z$
  $\boldsymbol{b}_{\text{CG}} \leftarrow \boldsymbol{A}^T \boldsymbol{y} + \rho \boldsymbol{D}_z^T (\boldsymbol{z}_{i+1} - \boldsymbol{w}_{i+1})$
  $\boldsymbol{x}_i \leftarrow \text{CG}(\boldsymbol{A}_{\text{CG}}, \boldsymbol{b}_{\text{CG}}, 1)$
  $\boldsymbol{z}_i \leftarrow \mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z \boldsymbol{x}_i + \boldsymbol{w}_{i+1})$
  $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i+1} + \boldsymbol{D}_z \boldsymbol{x}_i - \boldsymbol{z}_i$
end for
return x_0

Sharing primal/dual variables

- Warm start
- Much faster convergence

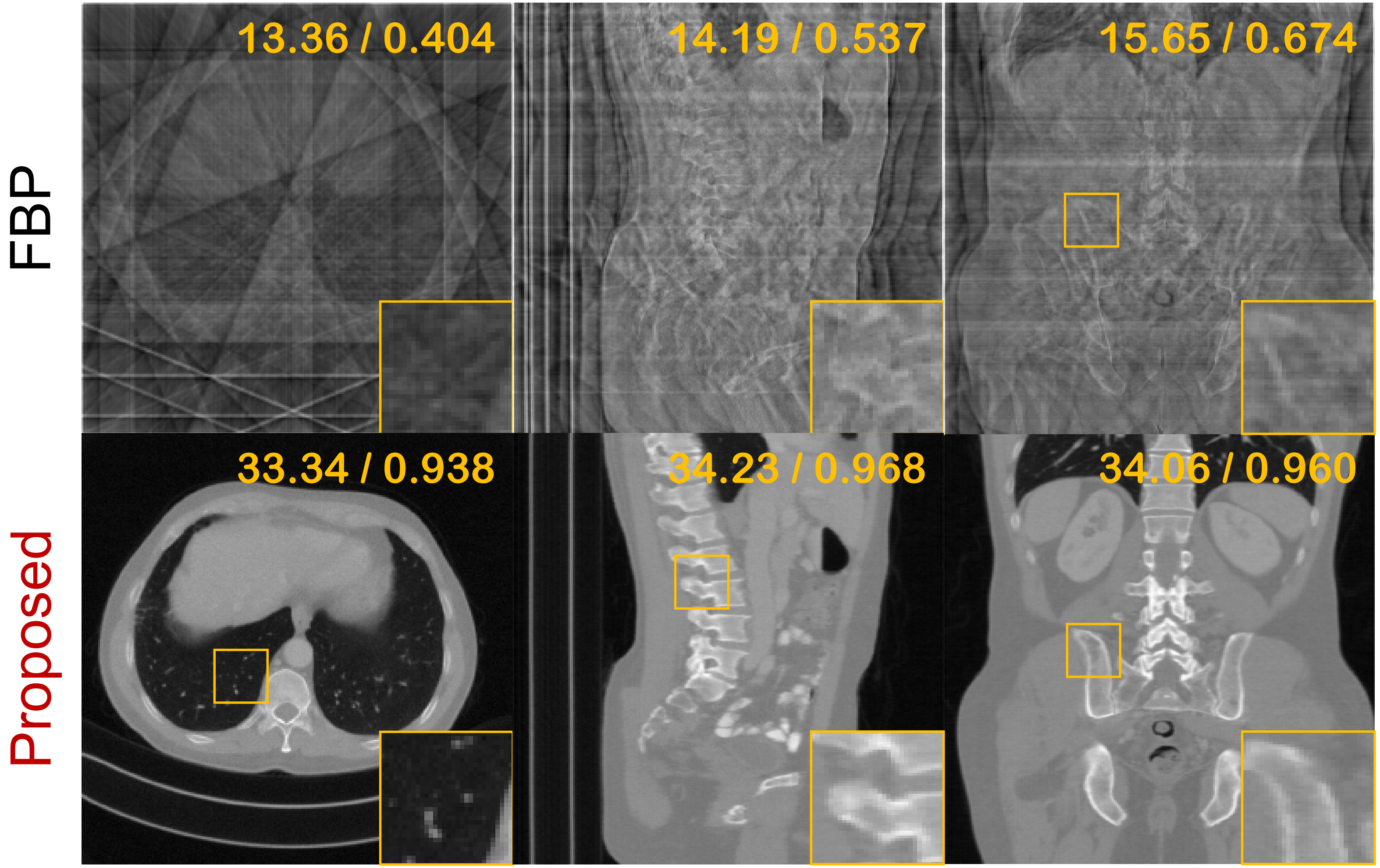
Score update (denoising)

- Prior in xy dimension

ADMM-TV iteration

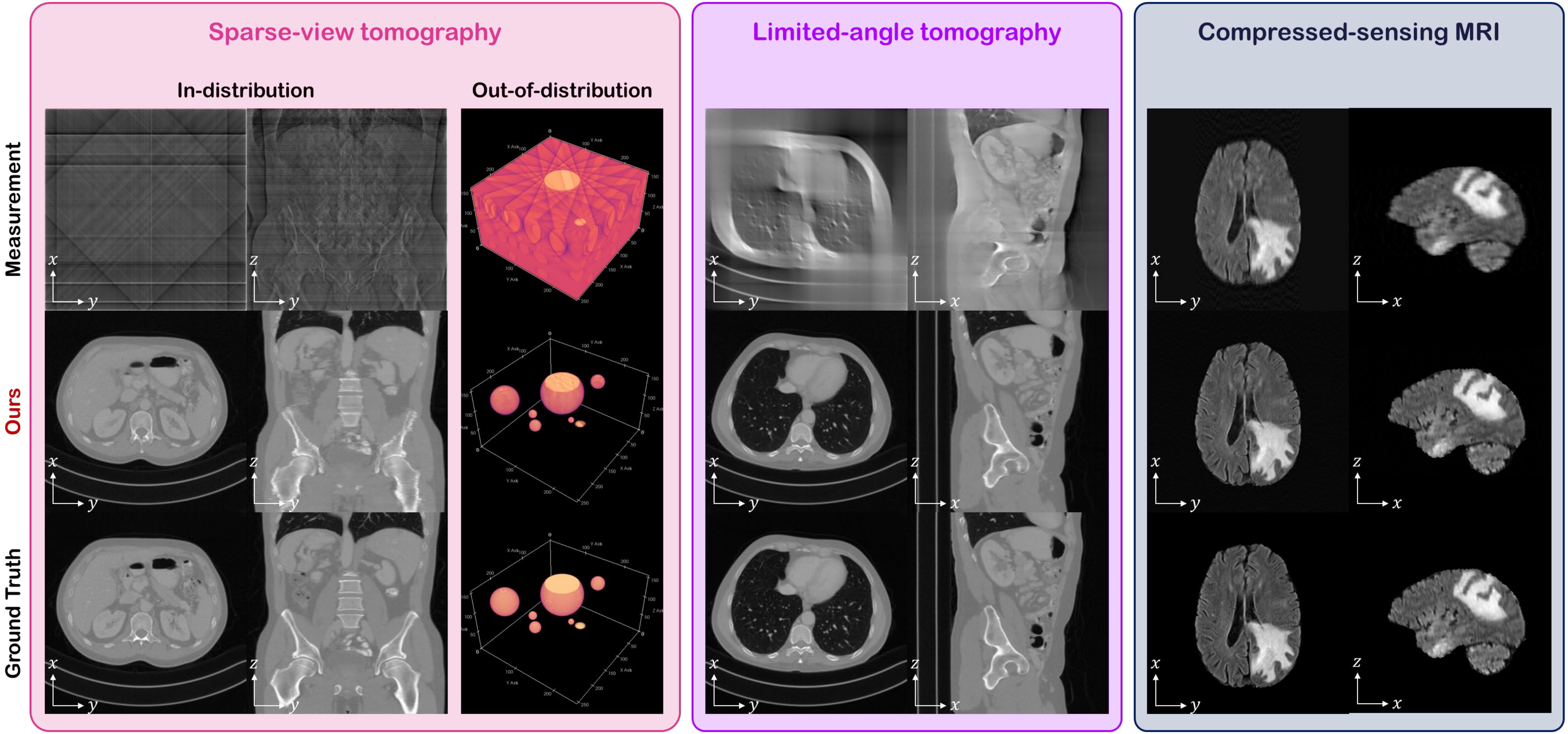
- Data consistency
- Prior in the z dimension

Results: sparse-view CT



Coherent results across the whole volume

Results


Sparse-view tomography
In-distribution
Out-of-distribution
Limited-angle tomography
Compressed-sensing MRI
Measurement
Ours
Ground Truth

Results: comparison study

SV-CT

Method	8-view Axial* PSNR ↑	SSIM ↑	8-view Coronal PSNR ↑	SSIM ↑	8-view Sagittal PSNR ↑	SSIM ↑	4-view Axial* PSNR ↑	SSIM ↑	4-view Coronal PSNR ↑	SSIM ↑	4-view Sagittal PSNR ↑	SSIM ↑	2-view Axial* PSNR ↑	SSIM ↑	2-view Coronal PSNR ↑	SSIM ↑	2-view Sagittal PSNR ↑	SSIM ↑
DiffusionMBIR (ours)	**33.49**	**0.942**	**35.18**	**0.967**	**32.18**	**0.910**	**30.52**	**0.914**	**30.09**	**0.938**	**27.89**	**0.871**	24.11	0.810	23.15	**0.841**	**21.72**	**0.766**
Chung *et al.* [1]	28.61	0.873	28.05	0.884	24.45	0.765	27.33	0.855	26.52	0.863	23.04	0.745	**24.69**	**0.821**	**23.52**	0.806	20.71	0.685
Lahiri *et al.* [2]	21.38	0.711	23.89	0.769	20.81	0.716	20.37	0.652	21.41	0.721	18.40	0.665	19.74	0.631	19.92	0.720	17.34	0.650
FBPConvNet	16.57	0.553	19.12	0.774	18.11	0.714	16.45	0.529	19.47	0.713	15.48	0.610	16.31	0.521	17.05	0.521	11.07	0.483
ADMM-TV	16.79	0.645	18.95	0.772	17.27	0.716	13.59	0.618	15.23	0.682	14.60	0.638	10.28	0.409	13.77	0.616	11.49	0.553

LA-CT

Method	Axial* PSNR ↑	SSIM ↑	Coronal PSNR ↑	SSIM ↑	Sagittal PSNR ↑	SSIM ↑
DiffusionMBIR (ours)	**34.92**	**0.956**	**32.48**	**0.947**	**28.82**	0.832
Chung *et al.* [1]	26.01	0.838	24.55	0.823	21.59	0.706
Lahiri *et al.* [2]	28.08	0.931	26.02	0.856	23.24	0.812
Zhang *et al.*	26.76	0.879	25.77	0.874	22.92	**0.841**
ADMM TV	23.19	0.793	22.96	0.758	19.95	0.782

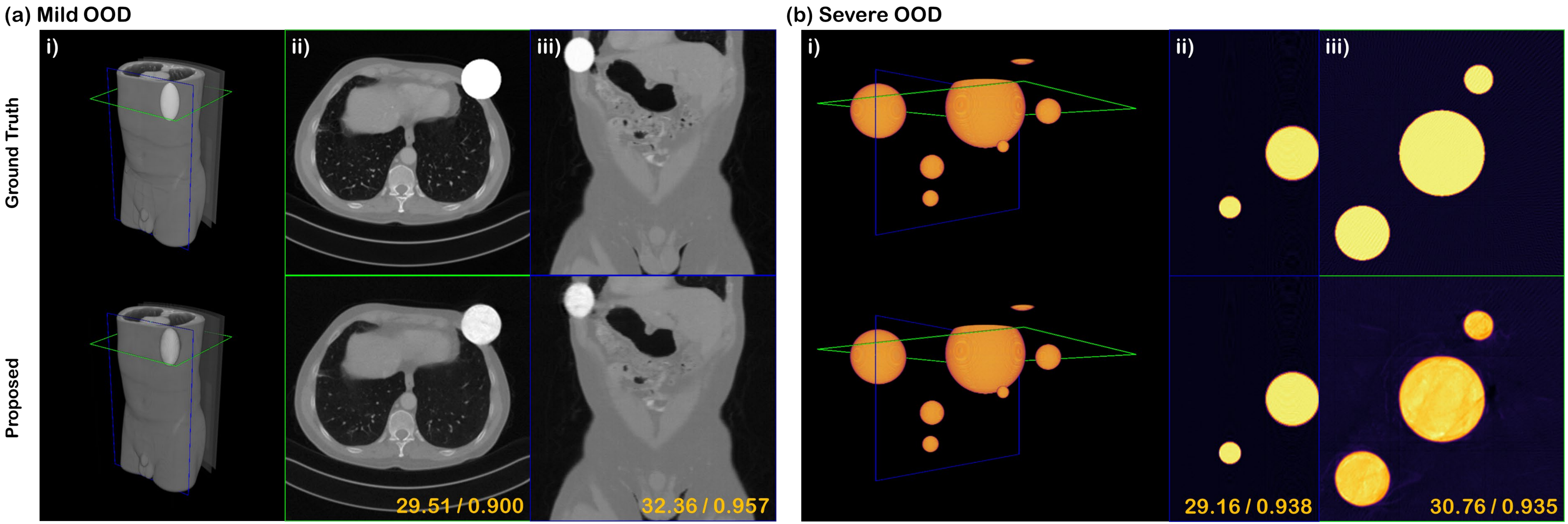
CS-MRI

Method	Axial* PSNR ↑	SSIM ↑	Coronal PSNR ↑	SSIM ↑	Sagittal PSNR ↑	SSIM ↑
DiffusionMBIR (ours)	**41.49**	**0.974**	**37.36**	**0.942**	**37.18**	**0.953**
Score-MRI	40.38	0.968	33.97	0.925	34.02	0.928
DuDoRNet	39.78	**0.974**	33.56	0.927	33.48	0.927
Unet	37.15	0.929	31.56	0.899	30.90	0.816
Zero-filled	34.18	0.923	29.53	0.897	27.82	0.903

[1] Chung *et al.*, "Improving diffusion models for inverse problems using manifold constraints", NeurIPS 2022

[2] Lahiri *et al.*, "Sparse-View Cone Beam CT Reconstruction Using Data-Consistent Supervised and Adversarial Learning From Scarce Training Data", IEEE TCI 2023

Results: out-of-distribution



Thank you!

Paper: https://arxiv.org/abs/2211.10655

Code: https://github.com/HJ-harry/DiffusionMBIR