Solving Inverse Problems in Medical Imaging with Score-Based Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Solving inverse problems with a small number of measurements has important applications in medical imaging, including image reconstruction for undersampled MRI and sparse-view CT. With the progress of machine learning, traditional image reconstruction methods have been outperformed by models that learn to directly map measurements to medial images. However, these models require both ground truth medical images and their measurements for training, which complicates data collection and harms their generalization performance to unknown measurement processes. To address these issues, we propose a fully unsupervised technique for inverse problem solving, leveraging the recently introduced score-based generative models. Specifically, we train a score-based generative model to capture the prior distribution of medical images, which is subsequently combined with a given physical measurement process to sample images consistent with measurements at the test time. Our method makes no assumption on the measurement process during training, and can be flexibly adapted to any linear measurement processes. Empirically, we observe comparable or better performance to supervised learning techniques, with better generalization to unknown measurement processes on several MRI/CT datasets.

1 Introduction

Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are commonly used imaging tools for medical diagnosis. Reconstructing MRI/CT images from raw measurements (k-space for MRI and sinograms for CT) is a well-known linear inverse problem. In order to speed up the hour-long process of MRI acquisition and reduce the dose of harmful ionizing radiation needed for CT scans, we aim to reduce the number of measurements required for solving these inverse problems by estimating and leveraging the prior information of MRI/CT images. Specifically, we consider image reconstruction tasks for undersampled MRI and sparse-view CT.

Many machine learning algorithms [1, 2, 3, 4, 5, 6] have been proposed to perform medical image reconstruction given a small number of measurements. However, most of these methods are supervised learning techniques. They learn to map measurements directly to medical images, by training on a large dataset comprising pairs of medical images and MRI/CT measurements. This complicates the data collection procedure and moreover, limits their generalization capability to different measurement processes that may appear at the test time, leading to counterintuitive instabilities such as more measurements causing worse performance [7].

We sidestep this difficulty by proposing unsupervised methods that make no assumption on the measurement process during training. Specifically, we train a score-based generative model [8, 9, 10] on medical images as the data prior, and modify the sampling algorithm as a general way to solve inverse problems with linear measurement processes. As an emerging family of techniques that have

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Algorithm 1 Unconditional sampling

Require: N

$\hat{\mathbf{x}}_1 \sim \pi(\mathbf{x}), \Delta t \leftarrow \frac{1}{N}$

for $i = N-1$ **to** 0 **do**

  $t \leftarrow \frac{i+1}{N}$

  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_t + g(t)^2 \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_t, t)\Delta t$

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

  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)\sqrt{\Delta t}\,\mathbf{z}$

return $\hat{\mathbf{x}}_0$

Algorithm 2 Inverse problem solving

Require: $N, \mathbf{y}, \lambda$

$\hat{\mathbf{x}}_1 \sim \pi(\mathbf{x}), \Delta t \leftarrow \frac{1}{N}$

for $i = N-1$ **to** 0 **do**

  $t \leftarrow \frac{i+1}{N}$

  $\hat{\mathbf{y}}_t \sim p_{0t}(\mathbf{y}_t \mid \mathbf{y})$

  $\hat{\mathbf{x}}_t \leftarrow \lambda \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t + (1-\lambda)\boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t + \boldsymbol{T}^{-1}(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t$

  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_t + g(t)^2 \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_t, t)\Delta t$

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

  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)\sqrt{\Delta t}\,\mathbf{z}$

return $\hat{\mathbf{x}}_0$

achieved top performance in image generation [8, 10, 11, 12], score-based generative models allow us to match or outperform the performance of supervised learning techniques, while being more resilient to the change of measurement processes, such as using a different number of measurements.

2 Background

The latest development of score-based generative models [10] leverages a Markovian diffusion process to progressively perturb data to noise, and then learns to smoothly convert noise to samples by solving its time reversal. Importantly, reversing the diffusion process requires estimating *scores*, *i.e.*, gradients of log probability density functions of the data distribution, from a training dataset.

Perturbation process Suppose the dataset is sampled from an unknown data distribution $p(\mathbf{x})$. For any datapoint $\mathbf{x}^* \in \mathbb{R}^n \sim p(\mathbf{x})$, we perturb it with a stochastic process over a time horizon $[0, 1]$, governed by a stochastic differential equation (SDE) of the following form

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \tag{1}$$

where $\boldsymbol{f}(\cdot, t) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(t) \in \mathbb{R}$ are called the *drift* and *diffusion* coefficients, $\{\mathbf{w}_t \in \mathbb{R}^n\}_{t\in[0,1]}$ denotes a standard Wiener process (a.k.a., Brownian motion), and $\{\mathbf{x}_t\}_{t\in[0,1]}$ symbolizes the trajectory of random variables in the stochastic process. We further denote the marginal probability distribution of $\mathbf{x}_t$ as $p_t(\mathbf{x})$, and the transition distribution from $\mathbf{x}_0$ to $\mathbf{x}_t$ as $p_{0t}(\mathbf{x}_t \mid \mathbf{x}_0)$. By definition, we clearly have $\mathbf{x}_0 = \mathbf{x}^*$ and $p_0(\mathbf{x}) \equiv p(\mathbf{x})$. The drift and diffusion coefficients are typically hand-crafted such that the distribution at the end of the perturbation process, $p_1(\mathbf{x})$, is close to a pre-defined prior distribution $\pi(\mathbf{x})$. Examples of such SDEs include Variance Exploding (VE), Variance Preserving (VP), and subVP SDEs [10].

Sampling process By reversing the perturbation process in Eq. (1), we can start from a sample $\mathbf{x}_1 \sim p_1(\mathbf{x})$ and gradually remove the noise therein to obtain a data sample $\mathbf{x}_0 \sim p_0(\mathbf{x}) \equiv p(\mathbf{x})$. Crucially, the time reversal of Eq. (1) is given by the following reverse-time SDE [13, 14]

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t, \tag{2}$$

where $\{\bar{\mathbf{w}}_t\}_{t\in[0,1]}$ is now a standard Wiener process in the reverse-time direction, and $\mathrm{d}t$ represents an infinitesimal negative time step, since the above SDE must be solved backwards from $t = 1$ to $t = 0$. When $\mathbf{x}_1 \sim p_1(\mathbf{x})$, the trajectory of the reverse stochastic process given by Eq. (2) is again $\{\mathbf{x}_t\}_{t\in[0,1]}$, same as the one obtained by the forward SDE in Eq. (1). Once given an initial sample from $p_1(\mathbf{x})$, as well as scores at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, we can use any numerical SDE solver to solve the reverse-time SDE in Eq. (2). The initial sample can be approximately drawn from $\pi(\mathbf{x})$ since $p_1(\mathbf{x}) \approx \pi(\mathbf{x})$, while the scores can be estimated by training a neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$ (named the time-dependent score model) with denoising score matching [15, 10], such that $\boldsymbol{s}_{\boldsymbol{\theta}*}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. Here $\boldsymbol{\theta}^*$ denotes the optimal model parameters. In Algorithm 1, we provide one such sampling method based on the classic Euler-Maruyama solver.

3 Method

By incorporating data consistency constraints into the sampling process, we can use score-based generative models to solve linear inverse problems. We consider a compressed sensing [16, 17, 18] setting, while replacing sparsity with a data prior learned by our score models.

Suppose $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a linear operator with full rank, *i.e.*, $\operatorname{rank}(\boldsymbol{A}) = \min(n, m)$. Let $\mathbf{x} \in \mathbb{R}^n$ be a signal, and $\mathbf{y} \in \mathbb{R}^m$ be a corresponding observation obtained by m measurements. In an inverse problem, we are given an observation $\mathbf{y}$ and aim to find $\mathbf{x}$ such that $\boldsymbol{A}\mathbf{x} = \mathbf{y}$. In order for the inverse problem to have at least one solution, we clearly require $m \leqslant n$. Since there might be multiple solutions, we further assume a prior distribution over $\mathbf{x}$, denoted as $p(\mathbf{x})$. With this assumption, solving the inverse problem becomes equivalent to finding the posterior distribution $p(\mathbf{x} \mid \mathbf{y})$.

To facilitate our theoretical discussion, we introduce an alternative formulation of $\boldsymbol{A}$ below:

Lemma 1. *If $\boldsymbol{A}$ has full rank, we can always write it as $\boldsymbol{A} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}$, where $\boldsymbol{T} \in \mathbb{R}^{n \times n}$ is an invertible matrix, $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix taking values in $\{0, 1\}$ that satisfies $\operatorname{tr}(\boldsymbol{\Lambda}) = m$, and $\mathfrak{C}_{\boldsymbol{\Lambda}} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator that takes any vector $\boldsymbol{a} \in \mathbb{R}^n$ and reduces its dimensionality to m by removing each i-th element of $\boldsymbol{a}$ if $\boldsymbol{\Lambda}_{ii} = 0$.*

For example, $\boldsymbol{T}$ corresponds to the Fourier transform for undersampled MRI reconstruction, and the Radon transform for sparse-view CT reconstruction.

Given a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$, the approach outlined in Algorithm 1 allows us to train a time-dependent score model $\boldsymbol{s}_{\boldsymbol{\theta}*}(\mathbf{x}, t)$ to generate samples from $p(\mathbf{x})$. To sample from $p(\mathbf{x} \mid \mathbf{y})$, however, we will need to incorporate the additional information of $\mathbf{y}$ into the sampling process. In inverse problems, $\mathbf{y}$ and $\mathbf{x}$ are connected through $\mathbf{y} = \boldsymbol{A}\mathbf{x}$. By introducing $\mathbf{y}_t = \boldsymbol{A}\mathbf{x}_t$, we generalize this connection to stochastic processes $\{\mathbf{y}_t\}_{t \in [0,T]}$ and $\{\mathbf{x}_t\}_{t \in [0,T]}$. When the drift coefficient $\boldsymbol{f}(\cdot, t)$ in Eq. (1) is linear, the transition density for $\{\mathbf{x}_t\}_{t \in [0,T]}$ is always a Gaussian distribution [10], taking the form $p_{0t}(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\alpha}(t)\mathbf{x}_0, \beta(t)\boldsymbol{I})$ where $\boldsymbol{\alpha}(t)$ and $\beta(t)$ can be computed from $\boldsymbol{f}(\cdot, t)$ and $g(t)$. The transition density of $\{\mathbf{y}_t\}_{t \in [0,T]}$ is therefore given by $p_{0t}(\mathbf{y}_t \mid \mathbf{y}_0) = \mathcal{N}(\mathbf{y}_t \mid \boldsymbol{\alpha}(t)\mathbf{y}_0, \beta(t)\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{I})$. When $\mathbf{y}_0$ is fixed to $\mathbf{y}^*$, the marginal distributions of $\mathbf{y}_t$ equals $p_{0t}(\mathbf{y}_t \mid \mathbf{y}_0 = \mathbf{y}^*)$ and can be computed in closed form.

Suppose $\hat{\mathbf{x}}_t$ is an intermediate sample obtained by running one step of the sampler in Algorithm 1, and let $\hat{\mathbf{y}}_t \sim p_{0t}(\mathbf{y}_t \mid \mathbf{y})$. Our goal is to steer $\hat{\mathbf{x}}_t$ towards $\hat{\mathbf{y}}_t$ to promote data consistency. To this end, we propose to find a steering function $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^n$, such that $\hat{\mathbf{x}}'_t = \boldsymbol{h}(\hat{\mathbf{x}}_t)$ satisfies both $\hat{\mathbf{x}}'_t \approx \hat{\mathbf{x}}_t$ and $\boldsymbol{A}\hat{\mathbf{x}}'_t \approx \hat{\mathbf{y}}_t$. Specifically, this steering function needs to minimize the distance between $\hat{\mathbf{x}}'_t$ and $\hat{\mathbf{x}}_t$, as well as the distance between $\hat{\mathbf{x}}'_t$ and the hyperplane $\boldsymbol{A}\boldsymbol{x} = \hat{\mathbf{y}}_t$. We propose to balance these two objectives by solving the following optimization problem

$$\begin{aligned} \hat{\mathbf{x}}'_t = \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} \{(1-\lambda)\left\|\boldsymbol{z} - \hat{\mathbf{x}}_t\right\|_{\boldsymbol{T}}^2 + \min_{\boldsymbol{u} \in \mathbb{R}^n} \lambda \left\|\boldsymbol{z} - \boldsymbol{u}\right\|_{\boldsymbol{T}}^2\} \\ s.t. \quad \boldsymbol{A}\boldsymbol{u} = \hat{\mathbf{y}}_t, \end{aligned} \tag{3}$$

where $0 < \lambda < 1$ controls the strength of the data-consistency loss, and we choose the norm $\|\boldsymbol{a}\|_{\boldsymbol{T}}^2 := \|\boldsymbol{T}\boldsymbol{a}\|_2^2$ to simplify our theoretical analysis. The following result gives a closed-form solution to the optimization problem in Eq. (3).

Theorem 1. *The optimal solution $\hat{\mathbf{x}}'_t$ is given by*

$$\hat{\mathbf{x}}'_t = \lambda \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t + (1-\lambda)\boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t + \boldsymbol{T}^{-1}(\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t, \tag{4}$$

where $\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1} : \mathbb{R}^m \to \mathbb{R}^n$ denotes any right inverse of $\mathfrak{C}_{\boldsymbol{\Lambda}}$.

By transforming $\hat{\mathbf{x}}_t$ to $\hat{\mathbf{x}}'_t$ after each sampling step, we can incorporate $\mathbf{y}$ to generate approximate samples from $p(\mathbf{x} \mid \mathbf{y})$. Our method requires minimal modification to the original unconditional sampling method of score-based generative models. For example, we can convert the sampler in Algorithm 1 to an inverse problem solver in Algorithm 2 by adding just two lines of pseudo-code.

A concurrent work in [19] proposes a different method to solve inverse problems with score-based generative models, leveraging posterior sampling and Langevin dynamics. In comparison, our method is applicable to a larger family of sampling methods in score-based generative models, such as numerical SDE solvers [10], and achieves better empirical performance in our experiments.

Table 1: Performance on image reconstruction for undersampled MRI and sparse-view CT.

Method	Measurements	PSNR	SSIM
Undersampled MRI on BraTS 240 × 240			
Cascade DenseNet	30	28.35±2.30	0.845±0.038
DuDoRNet	30	**37.88±3.03**	**0.985±0.007**
Langevin	30	36.44±2.28	0.952±0.016
Ours	30	37.63±2.70	0.958±0.015
Sparse-view CT on LIDC 320 × 320			
FISTA-TV	23	21.72±7.87	0.656±0.252
cGAN	23	18.27±2.22	0.473±0.111
Neumann	23	19.34±3.14	0.525±0.103
SIN-4c-PRN	23	30.48±3.99	0.895±0.047
Ours	23	**35.41±2.74**	**0.907±0.046**

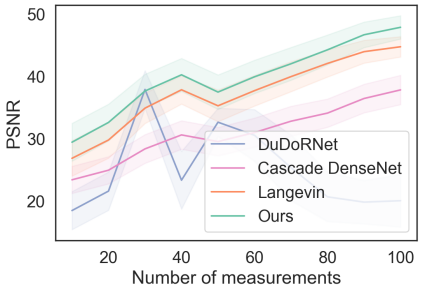

Figure 1: PSNR vs. number of measurements for undersampled MRI on BraTS. Shaded areas represent standard deviation.

4 Experiments

We consider two important examples of inverse problem solving in medical imaging: image reconstruction for undersampled MRI and sparse-view CT. We compare the performance of our method against traditional reconstruction algorithms as well as other deep learning based methods.

Datasets For undersampled MRI, we use the Brain Tumor Segmentation (BraTS) 2021 dataset [20, 21]. We slice 2D images of resolution 240×240 from original 3D MRI volumes. We simulate measurements by computing the k-space with Fast Fourier Transform, and follow [22, 23] to undersample the k-space with an equispaced Cartesian mask. For sparse-view CT, we use the Lung Image Database Consortium image collection (LIDC) [24, 25] dataset. We obtain 2D images of resolution 320×320 by slicing the 3D CT volumes. We simulate measurements (sinograms) with a parallel-beam geometry using projection angles equally distributed across 180 degrees.

Evaluation Our baselines include the traditional CT reconstruction method FISTA-TV [26], supervised learning methods Cascade DenseNet [27], DuDoRNet [28], cGAN [5], Neumann [29], SIN-4c-PRN [6], and a concurrent unsupervised learning method based on Langevin dynamics sampling from score-based generative models [30], which we denote as "Langevin". We evaluate the performance of different methods using the peak signal-to-noise ratio (PSNR) and the structural similarity metric (SSIM [31]). For deep learning based methods, we train and tune the corresponding models with 30 measurements for MRI and 23 projections for CT. We use the same score models in both Langevin and our method.

Results We provide experimental results in Table 1 and Fig. 1. For undersampled MRI, our method achieves comparable performance to the best supervised learning technique DuDoRNet. However, DuDoRNet cannot generalize well to different number of measurements since, as a supervised learning technique, it is specifically trained using 30 measurements. In contrast, our method is unsupervised and can generalize to an arbitrary number of measurements. As shown in Fig. 1, our method uniformly outperforms competing methods when using measurements other than 30. For sparse-view CT, we are able to outperform all baselines, even including supervised learning methods tested with the same number of measurements (23 projections) for training.

5 Conclusion

This paper describes a new method to solve linear inverse problems with score-based generative models. Our method is fully unsupervised, requires no paired or labeled data for training, and can flexibly adapt to different measurement processes at the test time. Preliminary results demonstrate that our method can match or outperform existing supervised learning counterparts for undersampled MRI and sparse-view CT reconstruction, while being more robust to the change of the number of measurements.

References

[1] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

[2] Morteza Mardani, Enhao Gong, Joseph Y Cheng, Shreyas Vasanawala, Greg Zaharchuk, Marcus Alley, Neil Thakur, Song Han, William Dally, John M Pauly, et al. Deep generative adversarial networks for compressed sensing automates mri. *arXiv preprint arXiv:1706.00051*, 2017.

[3] Liyue Shen, Wei Zhao, and Lei Xing. Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning. *Nature biomedical engineering*, 3(11):880–888, 2019.

[4] Tobias Würfl, Mathis Hoffmann, Vincent Christlein, Katharina Breininger, Yixin Huang, Mathias Unberath, and Andreas K Maier. Deep learning computed tomography: Learning projection-domain weights from image domain in limited angle problems. *IEEE transactions on medical imaging*, 37(6):1454–1463, 2018.

[5] Muhammad Usman Ghani and W Clem Karl. Deep learning-based sinogram completion for low-dose ct. In *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 2018.

[6] Haoyu Wei, Florian Schiffers, Tobias Würfl, Daming Shen, Daniel Kim, Aggelos K Katsaggelos, and Oliver Cossairt. 2-step sparse-view ct reconstruction with a domain-specific perceptual network. *arXiv preprint arXiv:2012.04743*, 2020.

[7] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

[8] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11918–11930, 2019.

[9] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33, 2020.

[12] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[13] Brian D.O. Anderson. Reverse-Time Diffusion Equation Models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[14] Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205, 1986.

[15] Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[16] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[17] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[18] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[19] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jonathan I Tamir. Robust compressed sensing mri with deep generative priors. *arXiv preprint arXiv:2108.01368*, 2021.

[20] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[21] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4(1):1–13, 2017.

[22] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An open dataset and benchmarks for accelerated MRI. 2018.

[23] Florian Knoll, Jure Zbontar, Anuroop Sriram, Matthew J Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J Geras, Joe Katsnelson, Hersh Chandarana, et al. fastmri: A publicly available raw k-space and dicom dataset of knee images for accelerated mr image reconstruction using machine learning. *Radiology: Artificial Intelligence*, 2(1):e190007, 2020.

[24] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[25] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013.

[26] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[27] Hao Zheng, Faming Fang, and Guixu Zhang. Cascaded dilated dense network with two-step data consistency for mri reconstruction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[28] Bo Zhou and S Kevin Zhou. Dudornet: Learning a dual-domain recurrent network for fast mri reconstruction with deep t1 prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4282, 2020.

[29] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019.

[30] Ajil Jalal, Liu Liu, Alexandros G Dimakis, and Constantine Caramanis. Robust compressed sensing using generative models. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[33] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

[34] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 2014.

[35] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[36] Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

A Related Work

The idea of solving inverse problems with generative models has been proposed in the Compressed Sensing with Generative Models (CSGM) framework [18]. However, the original CSGM method is centered around latent variable models such as Generative Adversarial Networks (GANs [32]) and Variational Auto-Encoders (VAEs [33, 34]). It is unclear how to apply CSGM to other families of generative models like Energy-Based Models (EBMs [35]) and score-based generative models.

Solving linear inverse problems with unsupervised learning and denoising score matching [15] has also been explored in [36]. In contrast with our approach, their method requires projecting an noisy sample onto the kernel space of a linear operator, which can be expensive to compute for high dimensional data. They did not consider applications in medical imaging either.

Concurrently, ref. [30] extends the NCSNv2 model [9], a score-based generative model sampled by annealed Langevin dynamics [8], to solve inverse problems in downsampled MRI reconstruction. Our method is applicable to more general sampling methods [10] for score-based generative models, such as numerical SDE solvers, Predictor-Corrector samplers, and probability flow ODE samplers. We also obtained uniformly better empirical performance in our experiments.

B Proofs

Lemma 1. *If $\boldsymbol{A}$ has full rank, we can always write it as $\boldsymbol{A} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}$, where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is an invertible matrix, $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is a diagonal matrix taking values in $\{0,1\}$ that satisfies $\mathrm{tr}(\boldsymbol{\Lambda}) = m$, and $\mathfrak{C}_{\boldsymbol{\Lambda}} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator that takes any vector $\boldsymbol{a} \in \mathbb{R}^n$ and reduces its dimensionality to m by removing each i-th element of $\boldsymbol{a}$ if $\boldsymbol{\Lambda}_{ii} = 0$.*

Proof. Let $\boldsymbol{A} = (\boldsymbol{a}_1^\mathsf{T}, \boldsymbol{a}_2^\mathsf{T}, \cdots, \boldsymbol{a}_m^\mathsf{T}) \in \mathbb{R}^{m\times n}$. Since $\boldsymbol{A}$ has full rank, the row vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m\}$ are linearly independent. We can therefore extend them to a total of n linearly independent vectors, *i.e.*, $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_{n-m}\}$. Due to the linear independence, we know $\boldsymbol{T} = (\boldsymbol{a}_1^\mathsf{T}, \boldsymbol{a}_2^\mathsf{T}, \cdots, \boldsymbol{a}_m^\mathsf{T}, \boldsymbol{b}_1^\mathsf{T}, \cdots, \boldsymbol{b}_{n-m}^\mathsf{T}) \in \mathbb{R}^{n\times n}$ has full rank and is invertible. Next, we define

$$\boldsymbol{\Lambda} = \mathrm{diag}(\underbrace{1, 1, \cdots, 1}_{m}, \underbrace{0, 0, \cdots, 0}_{n-m}),$$

where diag converts a vector to a diagonal matrix. Clearly $\mathrm{tr}(\boldsymbol{\Lambda}) = m$ and $\boldsymbol{A} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}$, which completes the proof. □

Lemma 2. *Let $\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1} : \mathbb{R}^m \to \mathbb{R}^n$ be any right inverse of $\mathfrak{C}_{\boldsymbol{\Lambda}} : \mathbb{R}^n \to \mathbb{R}^m$. For any $\boldsymbol{u} \in \mathbb{R}^n$ and $\hat{\mathbf{y}}_t \in \mathbb{R}^m$, we have*

$$\mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \hat{\mathbf{y}}_t \iff \boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{u} = \boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t$$

Proof. By the definition of $\mathfrak{C}_{\boldsymbol{\Lambda}}$, we have $\mathfrak{C}_{\boldsymbol{\Lambda}} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}$, and

$$\forall \boldsymbol{a} \in \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}^n : \quad \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{a} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{b} \iff \boldsymbol{\Lambda}\boldsymbol{a} = \boldsymbol{\Lambda}\boldsymbol{b}. \tag{5}$$

To prove the "if" direction, we note that

$$\begin{aligned}
\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{u} = \boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t &\implies \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{u} = \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t \\
&\implies \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \mathfrak{C}_{\boldsymbol{\Lambda}}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t \\
&\implies \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \hat{\mathbf{y}}_t.
\end{aligned}$$

To prove the "only if" direction, we have

$$\begin{aligned}
\mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \hat{\mathbf{y}}_t &\implies \mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \mathfrak{C}_{\boldsymbol{\Lambda}}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t \\
&\overset{(i)}{\implies} \boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{u} = \boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t,
\end{aligned}$$

where (i) is due to the property in Eq. (5). This completes the proof for both directions.

□

Theorem 1. *The optimal solution $\hat{\mathbf{x}}_t'$ is given by*

$$\hat{\mathbf{x}}_t' = \lambda \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t + (1-\lambda)\boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t + \boldsymbol{T}^{-1}(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t, \tag{4}$$

where $\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1} : \mathbb{R}^m \to \mathbb{R}^n$ denotes any right inverse of $\mathfrak{C}_{\boldsymbol{\Lambda}}$.

Proof. The optimization objective function in Eq. (3) can be written as

$$\begin{aligned}
&(1-\lambda)\left\|\boldsymbol{z}-\hat{\mathbf{x}}_t\right\|_{\boldsymbol{T}}^2 + \lambda\left\|\boldsymbol{z}-\boldsymbol{u}\right\|_{\boldsymbol{T}}^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\boldsymbol{u}\right\|_2^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u}) + (\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u})\right\|_2^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u})\right\|_2^2 + \lambda\left\|(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u})\right\|_2^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t\right\|_2^2 + \lambda\left\|(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u})\right\|_2^2
\end{aligned}$$

Since $\boldsymbol{A}\boldsymbol{u} = \hat{\mathbf{y}}_t$, we have $\mathfrak{C}_{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{u} = \hat{\mathbf{y}}_t$ and equivalently $\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{u} = \boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t$ due to Lemma 2. This constraint does not restrict the value of $(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\boldsymbol{u}$. Therefore, when $\boldsymbol{A}\boldsymbol{u} = \hat{\mathbf{y}}_t$, we have

$$\begin{aligned}
&\left\|\boldsymbol{z}-\hat{\mathbf{x}}_t\right\|_{\boldsymbol{T}}^2 + \min_{\boldsymbol{u}}(1-\lambda)\lambda\left\|\boldsymbol{z}-\boldsymbol{u}\right\|_{\boldsymbol{T}}^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \min_{\boldsymbol{u}}\lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t\right\|_2^2 + \lambda\left\|(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}(\boldsymbol{z}-\boldsymbol{u})\right\|_2^2 \\
=&(1-\lambda)\left\|\boldsymbol{T}\boldsymbol{z}-\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t\right\|_2^2 \\
=&(1-\lambda)\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t\right\|_2^2 + (1-\lambda)\left\|(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\boldsymbol{z}-(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2.
\end{aligned}$$

This simplifies the optimization problem in Eq. (3) to

$$\min_{\boldsymbol{z}}(1-\lambda)\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2 + \lambda\left\|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{z}-\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t\right\|_2^2 + (1-\lambda)\left\|(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\boldsymbol{z}-(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t\right\|_2^2,$$

which is minimizing a quadratic function of $\boldsymbol{z}$. The optimal solution $\boldsymbol{z}^*$ is thus in closed form:

$$\boldsymbol{z}^* = \boldsymbol{T}^{-1}[(\boldsymbol{I}-\boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t + (1-\lambda)\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t + \lambda\boldsymbol{\Lambda}\mathfrak{C}_{\boldsymbol{\Lambda}}^{-1}\hat{\mathbf{y}}_t].$$

According to the definition, $\hat{\mathbf{x}}_t' = \boldsymbol{z}^*$, whereby the proof is completed. □

C Additional details

Training and sampling We use the NCSN++ model architecture in [10], and perturb the data with the Variance Exploding (VE) SDE. Instead of generating samples according to the numerical SDE solver in Algorithm 1, we use the Predictor-Corrector (PC) sampler as described in [10] since it generally has better performance for VE SDEs. In PC samplers, the predictor refers to a numerical solver for the reverse-time SDE while the corrector can be any Markov chain Monte Carlo (MCMC) method that only depends on the scores. One such MCMC method considered in this work is Langevin dynamics, whereby we transform any initial sample $\mathbf{x}^{(0)}$ to an approximate sample from $p_t(\mathbf{x})$ via the following procedure:

$$\mathbf{x}^{(i+1)} \leftarrow \mathbf{x}^{(i)} + \epsilon\nabla_{\mathbf{x}}\log p_t(\mathbf{x}^{(i)}) + \sqrt{2\epsilon}\,\mathbf{z}^{(i)}, \quad i = 0, 1, \cdots, N-1. \tag{6}$$

Here $N \in \mathbb{N}_{>0}$, $\epsilon > 0$, and $\mathbf{z}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The theory of Langevin dynamics guarantees that in the limit of $N \to \infty$ and $\epsilon \to 0$, $\mathbf{x}^{(N)}$ is a sample from $p_t(\mathbf{x})$ under some regularity conditions. Note that Langevin dynamics only requires the knowledge of $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$, which can be approximated using the time-dependent score model $\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x}, t)$. In PC samplers, each predictor step immediately follows multiple consecutive corrector steps, all using the same $\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x}, t)$ evaluated at the same t. This jointly ensures that our intermediate sample at t is approximately distributed according to $p_t(\mathbf{x})$. As shown in [10], PC sampling often outperforms numerical solvers for the reverse-time SDE, especially when the forward SDE in Eq. (1) is a VE SDE. In order to use PC samplers for inverse problem solving, our modification is similar to the change made in Algorithm 2 for Algorithm 1. Specifically, we run line 4 & 5 in Algorithm 2 before every corrector or predictor step. The step size ϵ in the Langevin corrector, as well as λ in Eq. (3) are all tuned by running Bayesian optimization for 100 steps.

Datasets The Lung Image Database Consortium image collection (LIDC) [24, 25] consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans for lung cancer detection and diagnosis, which contains 1018 cases. We convert the Hounsfield units loaded from dicom files to the attenuation coefficients and set the background as zero. Then, 2D CT images are sliced from 3D CT cases. The sinogram are simulated from 2D CT images based on parallel-beam geometry with different number of projection angles that are equally distributed across 180 degrees. We conduct experiments of 2D MRI image reconstruction on a multi-modality MR image dataset. The Brain Tumor Segmentation (BraTS) 2021 dataset [20, 21] collected for a image segmentation challenge contains 2000 cases (8000 MRI scans), where each case has four different MR contrasts: native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR). For each 3D MR volume, we extract 2D slices from 3D volumes and simulate k-space data by Fourier Transform. To reconstruct MR images, k-space data is under-sampled by a equispaced Cartesian mask, where the center k-space is fully sampled while the left k-space is under-sampled by equispaced columns.

Baselines To evaluate the proposed method, we compare with baseline models including the traditional CT reconstruction method FISTA-TV [26], supervised learning methods for CT reconstruction: cGAN [5], Neumann [29], SIN-4c-PRN [6], and for MRI reconstruction: Cascade DenseNet [27], and DuDoRNet [28].

CT Reconstruction: FISTA-TV [26] is a fast iterative shrinkage-thresholding algorithm (FISTA) for solving linear inverse problems in image processing. It uses a total variation term as the regularization in the optimization procedure. Each optimization iteration involves a matrix-vector multiplication followed by a shrinkage/soft-threshold step. Conventional iterative CT reconstruction algorithms like FISTA-TV are typically slow due to their iterative nature. In ref. [5], authors propose to cast sparse-view CT reconstruction as a sinogram inpainting problem. Specifically, they propose to use conditional GANs to complete the projection data (sinogram) prior to reconstructing CT images, thereby avoiding the costly iterative tomographic inversion. However, the imperfect sinogram inpainting may cause other image artifacts. To reduce such artifacts, SIN-4c-PRN [6] proposed a 2-step sparse-view CT reconstruction model, which contains a sinogram inpainting network (SIN) to generate super-resolved sinograms and then a post-processing refining network (PRN) to further remove image artifacts. Both networks are connected through a filtered back-projection operation (FBP) operation. Meanwhile, in another parallel direction, researchers proposed to learn the regularizer used in optimization from training data, outperforming traditional regularizers. Specifically, ref. [29] presented an end-to-end, data-driven method for solving inverse problems inspired by the Neumann series, called a Neumann network, which directly solves the linear inverse problem with a data-driven nonlinear regularizer. Note that except for iterative reconstruction algorithm FISTA-TV, all the other deep-learning-based algorithms are supervised learning methods that require paired measurements (sinogram) and ground truth medical images for training networks.

MRI Reconstruction: To reconstruct de-aliased MR images from under-sampled k-space data, ref. [27] proposed a cascaded dilated dense network (CDDN) for MRI reconstruction, based on stacked dense blocks with residual connections while using the zero-filled MR image as inputs. Specifically, they use a two-step data consistency layer for k-space correction, and replace corresponding phase-coding lines of generated image with the original sampled k-space data after each block. Based on this model, ref. [28] proposed a dual domain recurrent network (DuDoRNet) to to simultaneously recover k-space and images for MRI reconstruction, in order to address aliasing artifacts in both frequency and image domains. The original model in [28] also embedded a deep T1 prior to make use of fully-sampled short protocol (T1) as complementary information. For a fair comparison with other supervised learning approaches, in our experiments, we do not include this additional information but train the DuDoRNet model without T1 prior. In this setting, we also observe that cascaded densenet generalizes better to more measurements than DuDoRNet.