

# Score-based diffusion models for accelerated MRI

Hyungjin Chung<sup>a</sup>, Jong Chul Ye<sup>a,\*</sup>

<sup>a</sup>Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Republic of Korea

---

## ARTICLE INFO

---

2021 MSC: 92C55, 68U10, 34A55

Keywords:

Score-based models

Diffusion models

Inverse problems

MRI

---

## ABSTRACT

---

Score-based diffusion models provide a powerful way to model images using the gradient of the data distribution. Leveraging the learned score function as a prior, here we introduce a way to sample data from a conditional distribution given the measurements, such that the model can be readily used for solving inverse problems in imaging, especially for accelerated MRI. In short, we train a continuous time-dependent score function with denoising score matching. Then, at the inference stage, we iterate between the numerical SDE solver and data consistency step to achieve reconstruction. Our model requires magnitude images only for training, and yet is able to reconstruct complex-valued data, and even extends to parallel imaging. The proposed method is agnostic to sub-sampling patterns and has excellent generalization capability so that it can be used with any sampling schemes for any body parts that are not used for training data. Also, due to its generative nature, our approach can quantify uncertainty, which is not possible with standard regression settings. On top of all the advantages, our method also has very strong performance, even beating the models trained with full supervision. With extensive experiments, we verify the superiority of our method in terms of quality and practicality. Code available at: <https://github.com/HJ-harry/score-MRI>

© 2022

---

## 1. Introduction

Reconstruction methods from sub-sampled measurements for magnetic resonance imaging (MRI) have seen a lot of progress over the years. Regularized reconstruction methods exploit the sparsity of hand-crafted priors along with data consistency to arrive at a solution (Donoho, 2006), yet the construction of priors is often non-trivial, and *none* of them can properly model the actual complicated data distribution  $p_{\text{data}}$  of MRI scans. Data-driven deep learning methods train the models either directly (Wang et al., 2016) or indirectly (Oh et al., 2020; Yaman et al., 2020), yet the methods rely heavily on the supervision of a well-curated large database of  $k$ -space data, which is hard to obtain.

Recently, score-based models (Hyvärinen and Dayan, 2005; Song and Ermon, 2019), and denoising diffusion probabilistic models (DDPMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020) have gained wide interest as a new class of generative model that achieves surprisingly high sample quality without adversarial training (Song et al., 2021b; Nichol and Dhariwal, 2021; Dhariwal and Nichol, 2021). Among many works, Song et al. (2021b) generalized discrete score-matching procedures

to a continuous stochastic differential equation (SDE), which in fact also subsumes diffusion models into the same framework. We refer to score-based SDEs as score-based diffusion models henceforth to emphasize that our proposed methodology can be flexibly used with any realizations from the two model classes: score-based generative models, and diffusion models.

Score-based diffusion models perturb the data distribution according to the forward SDE by injecting Gaussian noise, arriving at a tractable distribution (e.g. isotropic Gaussian distribution). In order to sample from the data distribution, one can train a neural network to estimate the gradient of the log data distribution (i.e. score function,  $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ ), and use it to solve the reverse SDE numerically. Unconditional generation of samples from  $p(\mathbf{x})$  using these score-based diffusion models have found their applications in image (Song et al., 2021b; Nichol and Dhariwal, 2021; Dhariwal and Nichol, 2021), audio (Kong et al., 2021), and even graph (Niu et al., 2020) synthesis. Conditional generation from  $p(\mathbf{x}|\mathbf{y})$  has also been studied in the context of widely known computer vision problems: in-painting (Song and Ermon, 2019; Song et al., 2021b), super-resolution (Choi et al., 2021; Saharia et al., 2021), and image editing (Meng et al., 2021).

In this work, we propose a framework generally applicable to solving inverse problems in imaging and especially focus on the task of MRI reconstruction. Notably, our method requires

---

\*Corresponding author.

*e-mail:* jong.ye@kaist.ac.kr (Jong Chul Ye)

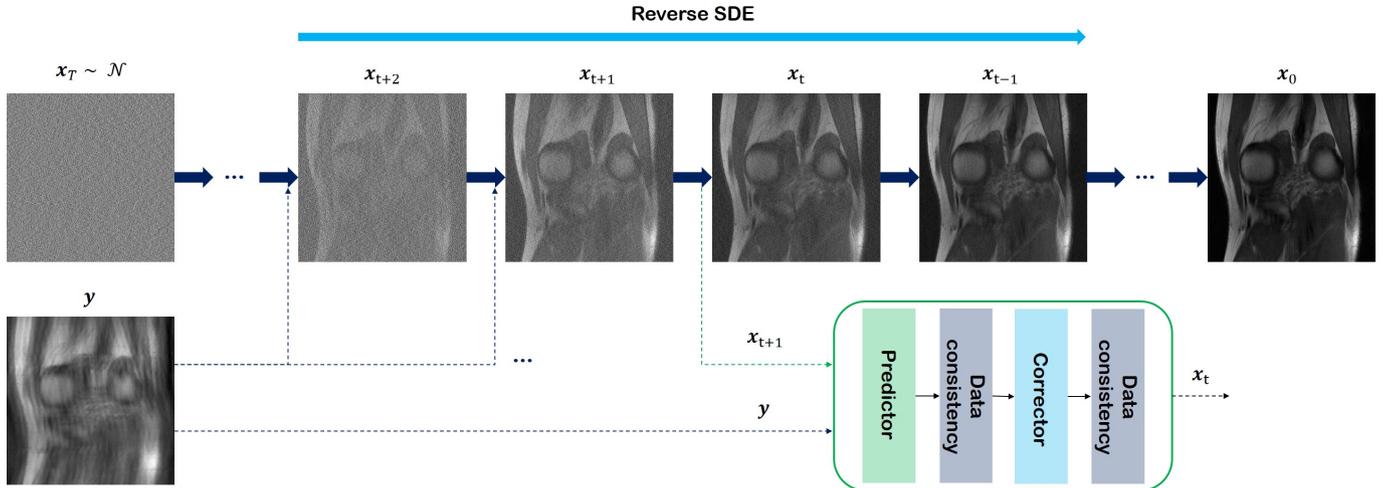


Fig. 1: Overview of the proposed method. Starting from  $x_T$ , sampled from the prior distribution,  $x_0$  is reached by solving the reverse SDE with score-based sampling, alternating between the update step, and the data consistency step.

training a *single* score function with *magnitude* images only. With the trained score model at hand using the de-noising score matching loss, we construct a solver for the reverse SDE from the variance exploding (VE)-SDE (Song et al., 2021b), which enables us to sample from the distribution  $p(\mathbf{x}|\mathbf{y})$ , conditioned on the measurement  $\mathbf{y}$ . This is done by imposing data consistency step at every iteration, after the unconditional update step.

Despite the fact that the score function was not trained to solve the image reconstruction task, our method achieves state-of-the-art performance, even outperforming models that were trained in a supervised fashion specifically for image reconstruction tasks for complex images. Furthermore, our model is agnostic to the sub-sampling pattern used in the acceleration procedure, as opposed to supervised models which require re-training every time a new sampling scheme is designed. In addition, the proposed method can be extended to the reconstruction of complex-valued MR image acquisition using a *single* network that has *never* seen complex-valued data before. Even more, our method can be readily applied to practical multi-coil settings with the same score function, where the update via score function can be applied in parallel to each coil image. It is worth mentioning that the generalization capability of the trained score function is far greater. In fact, when we try the reconstruction of data that is heavily out of training data distribution (e.g. different contrast, and even different anatomy), we are still able to achieve high fidelity reconstruction.

Finally, the proposed method is inherently stochastic, and for that, we can sample multiple reconstruction results from the same measurement vector  $\mathbf{y}$ . This is especially useful since we are able to quantify uncertainty without any specific treatment (e.g. Monte Carlo dropout (Gal and Ghahramani, 2016), estimating variance directly (Kendall and Gal, 2017)) to the neural net. We observe that at low acceleration factors, posterior samples do not deviate much from each other, meaning high confidence of the network. As the acceleration factor is pushed to higher values, the variance gradually increases, which can potentially aid practitioners’ decision-making.

Given the amazing generalization capability and the flexibil-

ity to produce uncertainty maps, we believe that the proposed score-based approach can be an important framework for compressed sensing MRI. Overview of the proposed method is illustrated in Fig. 1, and we detail our proposed method in Section 3.

The manuscript is organized as follows: relevant background on score-based diffusion models is reviewed in Section 2; detailed procedure and algorithm of the proposed method is presented in Section 3; specifications about the implementation and experiments are given in Section 4; experimental results are featured in Section 5; discussion about the concurrent works of conditional generation using diffusion models, limitations, and broader impacts of our work is presented in Section 6; we conclude our work in Section 7.

## 2. Background

### 2.1. Score-based SDE

One can construct a continuous diffusion process  $\{\mathbf{x}(t)\}_{t=0}^T$  with  $\mathbf{x}(t) \in \mathbb{R}^n$ , where  $t \in [0, T]$  is the time index of the progression and  $n$  denotes the image dimension. We choose  $\mathbf{x}(0) \sim p_{data}$  and  $\mathbf{x}(T) \sim p_T$ , where  $p_{data}, p_T$  refers to the data distribution of interest, and the prior distribution which are tractable to sample from (e.g. spherical Gaussian distribution), respectively. Then, the stochastic process can be constructed as the solution to the following SDE

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \quad (1)$$

where  $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$  and  $g : \mathbb{R} \mapsto \mathbb{R}$  correspond to the drift coefficient, and the diffusion coefficient, respectively, and  $\mathbf{w}$  is a standard  $n$ -dimensional Brownian motion.

One can construct different SDEs by choosing different functions for  $\mathbf{f}$  and  $g$ . First, by choosing

$$\mathbf{f} = -\frac{1}{2}\beta(t)\mathbf{x}, \quad g = \sqrt{\beta(t)}, \quad (2)$$

where  $0 < \beta(t) < 1$  is a monotonically increasing function of noise scale, one achieves the variance preserving (VP)-SDE

(Ho et al., 2020). In this case, the magnitude of the signal decays to 0, and the variance is preserved to a fixed constant as  $t \rightarrow \infty$ . In fact, VP-SDE can be seen as the continuous version of DDPM (Song et al., 2021b; Kingma et al., 2021). Therefore, while DDPM was developed in a separate variational framework, it can also be seen as a realization of SDE.

On the other hand, variance exploding (VE) SDEs choose

$$\mathbf{f} = \mathbf{0}, \quad g = \sqrt{\frac{d[\sigma^2(t)]}{dt}}, \quad (3)$$

where  $\sigma(t) > 0$  is again a monotonically increasing function, typically chosen to be a geometric series (Song and Ermon, 2019; Song et al., 2021b). Unlike VP-SDE, VE-SDE diffuses the signal with a very large variance, which explodes as  $t \rightarrow \infty$ , hence its name. Empirically, we found that using VE-SDE typically leads to higher sample qualities, and hence focus on developing our method on top of VE-SDE hereafter. However, we note that the use of VP-SDE (including the family of DDPMs developed under the variational framework) is also straightforward under our framework.

Interestingly, the reverse process of (1) can be constructed with another stochastic process (Song et al., 2021b):

$$\begin{aligned} d\mathbf{x} &= [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \underbrace{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}_{\text{score function}}] dt + g(t) d\bar{\mathbf{w}} \quad (4) \\ &= \frac{d[\sigma^2(t)]}{dt} \underbrace{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}_{\text{score function}} + \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\bar{\mathbf{w}}, \end{aligned}$$

where  $dt$  is the infinitesimal *negative* time step, and  $\bar{\mathbf{w}}$  is again the standard  $n$ -dimensional Brownian motion running backwards. The last equality follows by plugging in Eq. (3) to Eq. (4).

In order to solve (4), one has to know the score function for all  $t$ . One can estimate this score function with a time-conditional neural network  $s_\theta(\mathbf{x}(t), t) \simeq \nabla_{\mathbf{x}} \log p_t(\mathbf{x}(t))$ , and replace the term in (4). Since we do not know the *true* score, we can instead use denoising score matching (Vincent, 2011), where we replace the unknown  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  with  $\nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0))$ , where  $p_{0t}(\mathbf{x}(t)|\mathbf{x}(0))$  is the Gaussian perturbation kernel which perturbs the probability density  $p_0(\mathbf{x})$  to  $p_t(\mathbf{x})$ . Under some regularity conditions,  $s_\theta$  trained with denoising score matching will satisfy  $s_{\theta^*}(\mathbf{x}(t), t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  almost surely (Song et al., 2020). Formally, we optimize the parameters  $\theta$  of the score network with the following cost:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{t \sim U(0,1)} \left[ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \left[ \right. \right. \\ \left. \left. \left\| s_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right], \quad (5) \end{aligned}$$

where  $\lambda(t)$  is an appropriate weighting function, e.g. likelihood weighting of Song et al. (2021a), which puts different emphasis according to the time  $t$ . In the case of Gaussian perturbation kernels, the gradient of the perturbation kernel can be formulated explicitly:  $\nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0)) = (\mathbf{x}(t) - \mathbf{x}(0))/\sigma(t)^2$ . Intuitively, one can also understand (5) as training the neural network to de-noise  $\mathbf{x}(t)$ , which was constructed by adding noise to  $\mathbf{x}(0)$ .

In Song and Ermon (2019), it was shown with an illustrative toy example that when you add Gaussian noise to the random variable, you essentially get a blurred version of the original density, which indeed comes from the property that the addition of two random variables corresponds to the convolution of two densities (Loeve, 2017). Hence, when the score function is trained to denoise the given data across multiple noise scales, one can start the diffusion process with pure noise and gradually decrease the noise following the gradient information of the data density. Consequently, one can arrive at high-density modes of the data distribution.

Once the network is trained with (5), we can plug the approximation  $s_\theta(\mathbf{x}, t) \simeq \nabla_{\mathbf{x}} \log p_t(\mathbf{x}(t))$  to solve the reverse SDE in Eq. (4):

$$d\mathbf{x} = \frac{d[\sigma^2(t)]}{dt} s_\theta(\mathbf{x}(t), t) + \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\bar{\mathbf{w}}. \quad (6)$$

Then, we can solve the SDE numerically, for example, with Euler-Maruyama discretization (Song et al., 2021b). This involves discretizing  $t$  in range  $[0, 1]$  uniformly into  $N$  intervals such that  $0 = t_0 < t_1 < \dots < t_N = 1$ , with  $\Delta t = 1/N$ . Additionally, we can correct the direction of gradient ascent with *corrector* algorithms such as Langevin MC (Parisi, 1981). Iteratively applying predictor and corrector steps yield the predictor-corrector (PC) sampling algorithm (Song et al., 2021b), as presented in Algorithm 1. With the algorithm presented in Algorithm 1, we can sample from the distribution  $p(\mathbf{x})$ . In Section 3, we extend this sampling scheme to a conditional sampling algorithm, which enables us to sample from  $p(\mathbf{x}|\mathbf{y})$ .

---

#### Algorithm 1 Predictor-Corrector (PC) sampling

---

**Require:**  $s_\theta, N, M, \{\epsilon_i\}$      $\triangleright$  step size,  $\{\sigma_i\}$      $\triangleright$  noise schedule

- 1:  $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_N^2 \mathbf{I})$
- 2: **for**  $i = N - 1 : 0$  **do**
- 3:     $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) s_\theta(\mathbf{x}_{i+1}, \sigma_{i+1})$
- 4:     $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 5:     $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2} \mathbf{z}$
- 6:    **for**  $j = 1 : M$  **do**
- 7:      $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 8:      $\mathbf{x}'_i \leftarrow \mathbf{x}_i + \epsilon_j s_\theta(\mathbf{x}_i, \sigma_i)$
- 9:      $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{2\epsilon_j} \mathbf{z}$
- 10:    **end for**
- 11: **end for**
- 12: **return**  $\mathbf{x}_0$

---

### 3. Main Contributions

#### 3.1. Forward Measurement Model

In accelerated MRI, we consider the following measurement model

$$\mathbf{y} = A\mathbf{x} \quad (7)$$

where  $\mathbf{y} \in \mathbb{C}^m$  is the measurement,  $\mathbf{x} \in \mathbb{C}^n$  is the latent image, and a parameterized forward measurement matrix  $A \in \mathbb{C}^{m \times n}$  is defined as

$$A := \mathcal{P}_\Omega \mathcal{F} \mathcal{S}, \quad (8)$$

where  $\mathcal{S} := [\mathcal{S}^{(1)}; \dots; \mathcal{S}^{(c)}]$  is the sensitivity map for  $c$  different coils,  $\mathcal{F}$  denotes Fourier transform, and  $\mathcal{P}_\Omega$  is a diagonal matrix with zeros and ones that represent the sub-sampling operator with the given sampling pattern  $\Omega$ . The sensitivity map  $\mathcal{S}$  are normalized such that we have

$$\mathcal{S}^* \mathcal{S} = I. \quad (9)$$

In the case of single-coil acquisition,  $\mathcal{S}$  reduces to identity matrix such that  $A_{(sc)} = \mathcal{P}_\Omega \mathcal{F}$ .

### 3.2. Reverse SDE for Accelerated MR Reconstruction

A classic approach to find the solution of Eq. (7) is to solve the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \Psi(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{y} = A\mathbf{x}, \end{aligned} \quad (10)$$

where  $\Psi(\cdot)$  is, for example, a sparsity promoting regularizer derived from compressed sensing (CS) theory (Donoho, 2006), such as  $\ell_1$  wavelet (Lustig et al., 2007) and total variation (TV) (Block et al., 2007). Solving Eq. (10) typically involves proximal algorithms such as variable splitting (Boyd et al., 2011) or projection onto the convex sets (POCS) (Samsonov et al., 2004), which decouples the optimization of the prior term, and the forward consistency term. Then, one can alternate between solving the two sub-problems to arrive at the optimum.

In Bayesian perspective, we immediately see that  $\Psi(\mathbf{x})$  in Eq. (10) is the prior model of the data, i.e.  $p(\mathbf{x})$ . Hence, we can imagine that by more accurately estimating the complex prior data distribution, one would be able to achieve higher quality samples.

That being said, one of the important differences of the proposed method compared to the classical approaches is that rather than modeling the prior distribution  $p(\mathbf{x})$ , we exploit its stochastic samples. Specifically, the samples from the prior distribution can be obtained from the reverse SDE in Eq. (4), which can be discretized as illustrated in Algorithm 1 with

$$\mathbf{x}_i \leftarrow (\sigma_{i+1}^2 - \sigma_i^2) s_\theta(\mathbf{x}_{i+1}, \sigma_{i+1}) + \sqrt{\sigma_{i+1}^2 - \sigma_i^2} \mathbf{z}, \quad (11)$$

Then, the data consistency mapping on the constraint in (10) can be implemented by

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda A^*(\mathbf{y} - A\mathbf{x}_i) = (I - \lambda A^* A)\mathbf{x}_i + A^* \mathbf{y} \quad (12)$$

for  $\lambda \in [0, 1]$ , where  $A^*$  denotes the Hermitian adjoint of  $A$ .

Similar to our companion work (Chung et al., 2021), we impose the constraint on the operator  $A$  such that  $(I - \lambda A^* A)$  is a *non-expansive mapping* (Bauschke et al., 2011):

$$\|(I - \lambda A^* A)\mathbf{x} - (I - \lambda A^* A)\mathbf{x}'\| \leq \|\mathbf{x} - \mathbf{x}'\|, \quad \forall \mathbf{x}, \mathbf{x}' \quad (13)$$

For example, projection onto convex sets (POCS) in (Tang et al., 2011; Fan et al., 2017) or the one-iteration of the standard

---

### Algorithm 2 Score-based sampling (Real)

---

**Require:**  $s_\theta, N, M, \{\epsilon_i\}$   $\triangleright$  step size,  $\{\sigma_i\}$   $\triangleright$  noise schedule

**Define**  $A := \mathcal{P}_\Omega \mathcal{F}$

```

1:  $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_N^2 \mathbf{I})$ 
2: for  $i = N - 1 : 0$  do
3:    $\mathbf{x}_i \leftarrow \text{Predictor}(\mathbf{x}_{i+1}, \sigma_i, \sigma_{i+1})$ 
4:    $\mathbf{x}_i \leftarrow \text{Re}(\mathbf{x}_i + A^*(\mathbf{y} - A\mathbf{x}_i))$ 
5:   for  $j = 1 : M$  do
6:      $\mathbf{x}_i \leftarrow \text{Corrector}(\mathbf{x}_i, \sigma_i, \epsilon_i)$ 
7:      $\mathbf{x}_i \leftarrow \text{Re}(\mathbf{x}_i + A^*(\mathbf{y} - A\mathbf{x}_i))$ 
8:   end for
9: end for
10: return  $\mathbf{x}_0$ 

```

---

gradient descent with controlled step size (Jalal et al., 2021; Ramzi et al., 2020) corresponds to the non-expansive data consistency mapping. In the following, the normalization step in (9) is shown essential to ensure that  $(I - \lambda A^* A)$  is indeed non-expansive:

**Proposition 1.** *With the sensitivity normalization in (9),  $(I - \lambda A^* A)$  is non-expansive for  $\lambda \in [0, 1]$ .*

*Proof.* Using the properties of the spectral norm, we have

$$\begin{aligned} \|A^* A\| &= \|\mathcal{S}^* \mathcal{F}^* \mathcal{P}_\Omega^* \mathcal{P}_\Omega \mathcal{F} \mathcal{S}\| \stackrel{(a)}{=} \|\mathcal{S}^* \mathcal{F}^* \mathcal{P}_\Omega \mathcal{F} \mathcal{S}\| \\ &\stackrel{(b)}{\leq} \|\mathcal{S}^* \mathcal{F}^* \mathcal{F} \mathcal{S}\| \stackrel{(c)}{=} \|\mathcal{S}^* \mathcal{S}\| \stackrel{(d)}{=} 1 \end{aligned}$$

where (a)(b) come that the subsampling operator  $\mathcal{P}_\Omega$  is a diagonal matrix with 0 and 1, (c) is from the orthonormality of the Fourier transform and (d) is from (9). Therefore, we have

$$\|I - \lambda A^* A\| \leq \max\{|1 - \lambda|, 1\} \leq 1$$

for  $\lambda \in [0, 1]$ . Accordingly,

$$\|(I - \lambda A^* A)\mathbf{x} - (I - \lambda A^* A)\mathbf{x}'\| \leq \|I - \lambda A^* A\| \|\mathbf{x} - \mathbf{x}'\| \leq \|\mathbf{x} - \mathbf{x}'\| \quad (14)$$

This concludes the proof.  $\square$

Application of Eqs. (11) and (12) correspond to the predictor step. When using the additional corrector steps as in Algorithm 1, one can also apply the same treatment to the discrete corrector step

$$\begin{aligned} \mathbf{x}_i &\leftarrow \mathbf{x}_{i+1} + \epsilon_i s_\theta(\mathbf{x}_{i+1}, \sigma_{i+1}) + \sqrt{2\epsilon_i} \mathbf{z} \\ \mathbf{x}_i &\leftarrow \mathbf{x}_i + \lambda A^*(\mathbf{y} - A\mathbf{x}_i), \end{aligned} \quad (15)$$

where  $\epsilon_i$  is the step size at the  $i^{\text{th}}$  iteration. Iteratively applying predictor and corrector steps as in PC algorithm gives rise to the inference algorithm, which is described formally in Algorithm 2 when  $\lambda = 1$ .

Unfortunately, this algorithm can only be used when we know a priori that the signal only contains real values, and care must be taken since in most practical situations of MRI reconstruction, the signal that we would like to reconstruct is complex. This introduces a caveat when reconstructing the data

**Algorithm 3** Score-based sampling (SENSE-type)

---

**Require:**  $s_\theta, N, M, \{\epsilon_i\}$   $\triangleright$  step size,  $\{\sigma_i\}$   $\triangleright$  noise schedule

- 1: **if** parallel imaging (PI) **then**
- 2:    $A := \mathcal{P}_\Omega \mathcal{F} \mathcal{S}$
- 3: **else**
- 4:    $A := \mathcal{P}_\Omega \mathcal{F}$
- 5: **end if**
- 6:  $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$
- 7: **for**  $i = N - 1 : 0$  **do**
- 8:    $\text{Re}(\mathbf{x}_i) \leftarrow \text{Predictor}(\text{Re}(\mathbf{x}_{i+1}), \sigma_i, \sigma_{i+1})$
- 9:    $\text{Im}(\mathbf{x}_i) \leftarrow \text{Predictor}(\text{Im}(\mathbf{x}_{i+1}), \sigma_i, \sigma_{i+1})$
- 10:    $\mathbf{x}_i = \text{Re}(\mathbf{x}_i) + \iota \text{Im}(\mathbf{x}_i)$
- 11:    $\mathbf{x}_i \leftarrow \mathbf{x}_i + A^*(y - A\mathbf{x}_i)$
- 12:   **for**  $j = 1 : M$  **do**
- 13:      $\text{Re}(\mathbf{x}_i) \leftarrow \text{Corrector}(\text{Re}(\mathbf{x}_i), \sigma_i, \epsilon_i)$
- 14:      $\text{Im}(\mathbf{x}_i) \leftarrow \text{Corrector}(\text{Im}(\mathbf{x}_i), \sigma_i, \epsilon_i)$
- 15:      $\mathbf{x}_i = \text{Re}(\mathbf{x}_i) + \iota \text{Im}(\mathbf{x}_i)$
- 16:      $\mathbf{x}_i \leftarrow \mathbf{x}_i + A^*(y - A\mathbf{x}_i)$
- 17:   **end for**
- 18: **end for**
- 19: **return**  $\mathbf{x}_0$

---

with the score function because the original theory of score-based SDEs (Song et al., 2021b) did not consider complex signals.

One approach that is feasible is to train a score function so that it handles complex signals, was proposed in (Ramzi et al., 2020). Implementation-wise, this corresponds to considering real and imaginary parts of the signal as separate channels and applying the de-noising score matching objective to handle  $2 \times H \times W$  sized image, where  $H$  and  $W$  are the height, and the width of the image, respectively. However, we empirically found that this treatment reduces the stability of network training, and also hurts the performance of the reconstruction using Algorithm 2. We further note that this treatment limits the practicality since the model now requires raw  $k$ -space data for training.

To overcome these limitations, we propose a simple fix to Algorithm 2, which provides a way to use the score function  $s_\theta$  trained with magnitude images *only*, and use it to reconstruct complex images. The method is presented in Algorithm 3, where we split the image into real and imaginary parts, and apply the predictor-corrector step separately to each part. Accordingly, we can use the same score function that was trained with magnitude images to deal with complex image data in a seamless way. This simple fix works surprisingly well, and we show in Section 5.2 that reconstruction of complex-valued coil data with Algorithm 3 even outperforms the standard feed-forward neural network trained with explicit supervision. Being able to utilize score functions trained with magnitude-only data to reconstruct complex-valued data is a great advantage since we can use only the DICOM data to train the neural network. This is advantageous because the plethora of MR scans exist in the form of DICOM (Zbontar et al., 2018), while the raw  $k$ -space data are usually discarded due to their excessive memory size.

**Algorithm 4** Score-based sampling (SSOS-type)

---

**Require:**  $s_\theta, N, \{\epsilon_i\}$   $\triangleright$  step size,  $\{\sigma_i\}$   $\triangleright$  noise schedule

- 1: **Define**  $A := \mathcal{P}_\Omega \mathcal{F}$
- 2:  $\mathbf{x}_N^{(k)} \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$
- 3: **for**  $i = N - 1 : 0$  **do**
- 4:   **for**  $k = 1 : c$  **do (parallel)**
- 5:      $\text{Re}(\mathbf{x}_i^{(k)}) \leftarrow \text{Predictor}(\text{Re}(\mathbf{x}_{i+1}^{(k)}), \sigma_i, \sigma_{i+1})$
- 6:      $\text{Im}(\mathbf{x}_i^{(k)}) \leftarrow \text{Predictor}(\text{Im}(\mathbf{x}_{i+1}^{(k)}), \sigma_i, \sigma_{i+1})$
- 7:      $\mathbf{x}_i^{(k)} = \text{Re}(\mathbf{x}_i^{(k)}) + \iota \text{Im}(\mathbf{x}_i^{(k)})$
- 8:      $\mathbf{x}_i^{(k)} \leftarrow \mathbf{x}_i^{(k) + A^*(y^{(k)} - A\mathbf{x}_i^{(k)})}$
- 9:      $\text{Re}(\mathbf{x}_i^{(k)}) \leftarrow \text{Corrector}(\text{Re}(\mathbf{x}_i^{(k)}), \sigma_i, \epsilon_i)$
- 10:      $\text{Im}(\mathbf{x}_i^{(k)}) \leftarrow \text{Corrector}(\text{Im}(\mathbf{x}_i^{(k)}), \sigma_i, \epsilon_i)$
- 11:      $\mathbf{x}_i^{(k)} = \text{Re}(\mathbf{x}_i^{(k)}) + \iota \text{Im}(\mathbf{x}_i^{(k)})$
- 12:      $\mathbf{x}_i^{(k)} \leftarrow \mathbf{x}_i^{(k)} + A^*(y^{(k)} - A\mathbf{x}_i^{(k)})$
- 13:   **end for**
- 14: **end for**
- 15:  $\mathbf{x}_0 = \sqrt{\sum_{k=1}^c |\mathbf{x}_0^{(k)}|^2}$   $\triangleright$  SSOS
- 16: **return**  $\mathbf{x}_0$

---

**Algorithm 5** Score-based sampling (Hybrid-type)

---

**Require:**  $s_\theta, N, M, m, \{\epsilon_i\}$   $\triangleright$  step size,  $\{\sigma_i\}$   $\triangleright$  noise schedule

**Define**  $A_{(sc)} := \mathcal{P}_\Omega \mathcal{F}$

- 1: **Define**  $A_{(mc)} := \mathcal{P}_\Omega \mathcal{F} \mathcal{S}$
- 2:  $\mathbf{x}_N^{(k)} \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$
- 3: **for**  $i = N - 1 : 0$  **do**
- 4:   **for**  $k = 1 : c$  **do (parallel)**
- 5:      $\text{Re}(\mathbf{x}_i^{(k)}) \leftarrow \text{Predictor}(\text{Re}(\mathbf{x}_{i+1}^{(k)}), \sigma_i, \sigma_{i+1})$
- 6:      $\text{Im}(\mathbf{x}_i^{(k)}) \leftarrow \text{Predictor}(\text{Im}(\mathbf{x}_{i+1}^{(k)}), \sigma_i, \sigma_{i+1})$
- 7:      $\mathbf{x}_i^{(k)} = \text{Re}(\mathbf{x}_i^{(k)}) + \iota \text{Im}(\mathbf{x}_i^{(k)})$
- 8:      $\mathbf{x}_i^{(k)} \leftarrow \mathbf{x}_i^{(k) + A_{(sc)}^*(y^{(k)} - A_{(sc)}\mathbf{x}_i^{(k)})}$
- 9:      $\text{Re}(\mathbf{x}_i^{(k)}) \leftarrow \text{Corrector}(\text{Re}(\mathbf{x}_i^{(k)}), \sigma_i, \epsilon_i)$
- 10:      $\text{Im}(\mathbf{x}_i^{(k)}) \leftarrow \text{Corrector}(\text{Im}(\mathbf{x}_i^{(k)}), \sigma_i, \epsilon_i)$
- 11:      $\mathbf{x}_i^{(k)} = \text{Re}(\mathbf{x}_i^{(k)}) + \iota \text{Im}(\mathbf{x}_i^{(k)})$
- 12:      $\mathbf{x}_i^{(k)} \leftarrow \mathbf{x}_i^{(k)} + A_{(mc)}^*(y^{(k)} - A_{(mc)}\mathbf{x}_i^{(k)})$
- 13:   **end for**
- 14:   **if**  $\text{mod}(i, m) == 0$  **then**
- 15:      $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \dots, \mathbf{x}_i^{(c)}]$   $\triangleright$  Aggregation
- 16:      $\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda A_{(mc)}^*(y - A_{(mc)}\mathbf{x}_i)$
- 17:   **end if**
- 18: **end for**
- 19:  $\mathbf{x}_0 = \sqrt{\sum_{k=1}^c |\mathbf{x}_0^{(k)}|^2}$   $\triangleright$  SSOS
- 20: **return**  $\mathbf{x}_0$

---

## 3.3. Diffusion model meets Parallel Imaging (PI)

While the proposed score-based framework for the reconstruction of complex-valued data is extremely useful, most modern MRI scanners (Zbontar et al., 2018) have *multiple* receiver coils, which capture the signal with different sensitivities. Since the birth of PI (Deshmane et al., 2012), myriad of techniques to reconstruct the true latent signal have been proposed in the literature, two of the most prominent being SENSE (Pruessmann et al., 1999), and GRAPPA (Griswold

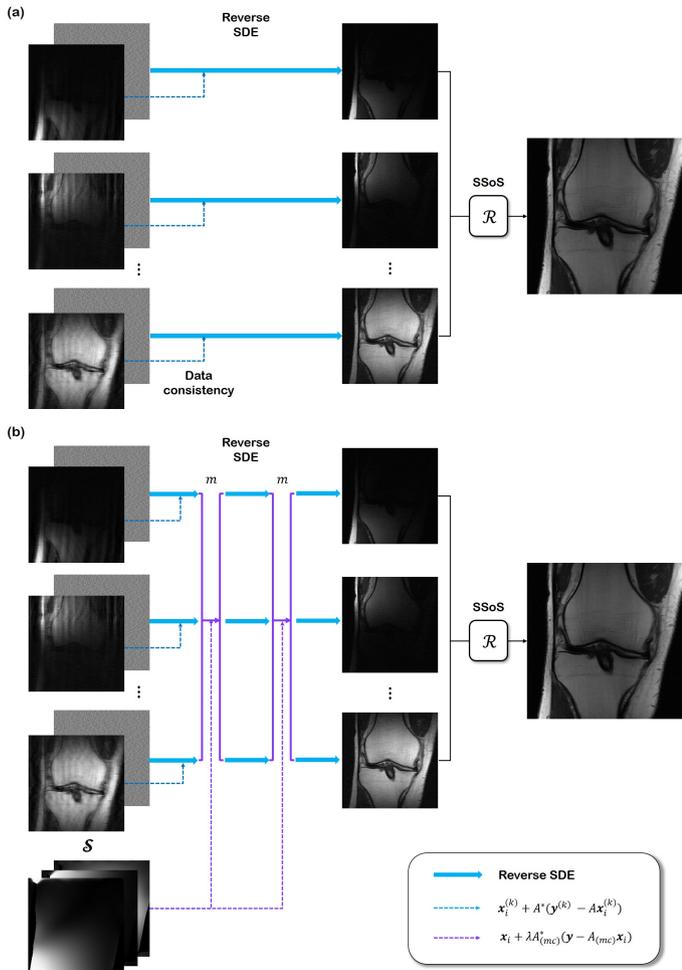


Fig. 2: Illustration of parallel imaging applications. (a) SSOS-type sampling. Coil images are reconstructed separately, then merged with the SSOS operation. (b) Hybrid-type sampling. Dependency between the coil images is injected every  $m$  steps of iteration.

et al., 2002). The former approach requires estimation or calibration of coil sensitivity maps, which are incorporated in the forward model specified in (7). Contrarily, the latter approach drops the need for the sensitivity maps, by simply taking the sum-of-root-sum-of-squares (SSOS) of each reconstructed coil image (Roemer et al., 1990). This approach is one of the most widely used methods in clinical practice due to several advantages including implementation ease. Here, we show that our score-based method can be integrated seamlessly into the SSOS-type approach.

Specifically, as introduced in Algorithm 4, our objective is to reconstruct the data coil-by-coil. For PI, we set the number corrector step to one for brevity, i.e.  $M = 1$ . More specifically, unlike GRAPPA, instead of estimating of the GRAPPA kernels, we simply apply Algorithm 3 separately to each coil image, as illustrated in Fig. 2 (a). Notably, although our score function estimator  $s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}_i} \log p(\mathbf{x})$  were not trained with separate coil images, since the distribution of independent coil images does not deviate much from  $p(\mathbf{x})$ , simply applying Algorithm 3 to each coil image induces a very accurate reconstruction.

One downside of Algorithm 4 is that there exists no cross-

talk between the coil reconstructions. This may be suboptimal, because the reconstruction process does not take into account that all the coil images come from the same image. Instead, in order to properly leverage the correlation between different coil images, we additionally implement a hybrid-type method in Algorithm 5 by incorporating the SENSE type constraint. Specifically, for every  $m$  steps of individual coil updates, we coerce dependency between the coils with the following update:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda A_{(mc)}^*(y - A_{(mc)}\mathbf{x}_i), \quad (16)$$

where  $\lambda \in [0, 1]$  decides the extent to which data consistency is imposed, and  $A_{(mc)}$  is the standard multi-coil forward measurement matrix as defined in (8). Once this is done, we arrive at the final result with SSOS operation. See Fig. 2 (b) for an illustration. By applying update steps of (16) in between with linearly decreasing the value of  $\lambda$  as the iteration progresses, we observe improved performance and sharper reconstructions.

Among the different sampling patterns, we see that Algorithm 5 generally performs better with 1D sampling patterns, whereas Algorithm 4 performs better with 2D patterns. Hence, we report on reconstructions with Algorithm 5 for 1D under-sampling, and Algorithm 4 for 2D under-sampling. One caveat for the SSOS-type, and hybrid-type approaches are the slow inference speed. Naively implementing the algorithms will induce  $c$  times longer computation time compared to the single-coil reconstruction. However, this can be much relieved by performing parallel computation with each coil data, since no computation needs to be shared across the threads. Given sufficient GPU resources, we expect that the computation time needed for PI reconstruction will be reduced down to the time needed for single-coil reconstruction.

## 4. Methods

### 4.1. Experimental data

The main experiments, including the training of the score function, were performed with fastMRI knee dataset (Zbontar et al., 2018), which is publicly available<sup>1</sup>. We trained the network with  $320 \times 320$  size target image magnitude, given as the key `reconstruction_esc`. We note that it is possible to train the score function with the same target from `reconstruction_rss` of the multi-coil dataset, but we found no significant difference in the performance.

Among 973 volumes of training data, we dropped the first and last five slices from each volume, to avoid training the model with noise-only data. This results in approximately 25k slices of training data. For testing, we randomly sampled 30 volumes from the validation set, and dropped the first and last 5 slices from each volume. For PI experiments, we resorted to 10 volumes, due to the computational limitations.

<sup>1</sup><https://fastmri.org/>

#### 4.2. Implementation details

We follow similar procedures to train VE-SDE as advised in (Song et al., 2021b). Specifically, we train the network with the objective given in Eq. (5), with setting  $\lambda(t) = \sigma^2(t)$ . Note that this specific choice of  $\lambda(t)$  stabilizes the noise scale across  $t$ , and theoretically corresponds to likelihood weighting, as proven in (Song et al., 2021a). Plugging in the weighting function, we can train the model with the following cost:

$$\min_{\theta} \mathbb{E}_{t \sim U(\epsilon, 1)} \mathbb{E}_{\mathbf{x}(0) \sim p_0} \mathbb{E}_{\mathbf{x}(t) \sim \mathcal{N}(\mathbf{x}(0), \sigma^2(t)I)} \left[ \left\| \sigma(t) s_{\theta}(\mathbf{x}(t), t) - \frac{\mathbf{x}(t) - \mathbf{x}(0)}{\sigma(t)} \right\|_2^2 \right], \quad (17)$$

with setting  $\epsilon = 10^{-5}$  to circumvent numerical issues.

For the step size  $\epsilon_i$  used in the Langevin MC corrector step, we follow what is advised in (Song et al., 2021b), and set

$$\epsilon_i = 2r \frac{\|z\|_2}{\|s_{\theta}(\mathbf{x}_i, \sigma_i)\|_2}, \quad (18)$$

where  $r = 0.16$  is set to a constant value. For noise variance schedule, we fix  $\sigma_{min} = 0.01$ ,  $\sigma_{max} = 378$ , which is similar to what is advised in the technique of (Song and Ermon, 2020), then take the geometric series with the following form:

$$\sigma(t) = \sigma_{min} \left( \frac{\sigma_{max}}{\sigma_{min}} \right)^t. \quad (19)$$

We take the batch size of 1, and optimize the network using the Adam optimizer ( $\beta_1 = 0.9, \beta_2 = 0.999$ ). We use a linear warm-up schedule for the parameters for the first 5000 steps of optimization, reaching  $2e-4$  at the 5000<sup>th</sup> step. We apply gradient clipping with the maximum value of 1.0 (Arjovsky et al., 2017). Exponential moving average with rate 0.999 is applied to the parameters. Optimization was performed for 100 epochs, and it took about 3 weeks of training the score function with a single RTX 3090 GPU. All code was implemented in PyTorch (Paszke et al., 2019).

For all algorithms, we use  $N = 2000, M = 1$  iterations for inference as default, unless specified otherwise. For the hybrid-type Algorithm 5, we start with  $\lambda = 1.0$  in the first iteration, and linearly decrease the value to  $\lambda = 0.2$  at the last iteration. Single forward-pass through  $s_{\theta}$  is required for both predictor and corrector steps, which takes about 150 ms with a commodity GPU. Summing up, this results in about 10 minutes of reconstruction time for real-valued images, and 20 minutes of reconstruction time for complex-valued images. We discuss ways for speeding up inference, and some potential directions for future studies in Section 6.1.

#### 4.3. Model Architecture

We base the implementation of the time-dependent score function model `ncsnpp`<sup>2</sup> as suggested in (Song et al., 2021b). The model architecture stems from U-Net (Ronneberger et al., 2015), and the sub-block which consist U-Net are adopted from

residual blocks of BigGAN (Brock et al., 2019). The skip connections in the residual blocks are scaled by  $1/\sqrt{2}$  as in (Karras et al., 2017, 2019, 2020). For pooling and unpooling, we adopt anti-aliasing pooling of (Zhang, 2019). The resulting U-Net has 4 different levels of scale, with 4 residual networks at each level. Conditioning of network with the time index  $t$  is performed with Fourier features (Tancik et al., 2020), where the conditional features are added to the encoder features. For further details, see Appendix.

#### 4.4. Comparison study

To verify superiority over the current standards, we perform comparison studies with baseline methods used in (Zbontar et al., 2018). We choose total variation (TV) reconstruction (Block et al., 2007) as the representative CS reconstruction method, where we use the implementation in `sigpy.mri.app.TotalVariationRecon`<sup>3</sup>. We perform grid search on the hyper-parameter `lambda`, and report only the best results among them.

For a representative deep learning approach, we use supervised learning-based reconstruction using U-Net (Zbontar et al., 2018). While we could use the open-sourced<sup>4</sup> pre-trained model, we re-implemented the model to achieve better performance.

We additionally compare against the state-of-the-art supervised methods. For the real-valued simulation study and single-coil experiment, we compare against DuDoRNet (Zhou and Zhou, 2020). We use the official implementation<sup>5</sup>, with 4 recurrent blocks and default parameters. We resort to the same proton density (PD)/ proton density fat suppressed (PDFS) image for prior information. For state-of-the-art parallel imaging method, we use end-to-end variational network (E2E-varnet) (Sriram et al., 2020), which simultaneously estimates the sensitivity maps. We use the official fastMRI github with default parameters as advised with the fastMRI knee dataset. For all the deep learning comparison studies, we train the network with Gaussian 1D random sampling masks.

#### 4.5. Measurement of Reconstruction Quality

To quantify the proximity of the reconstructions to the target, we use the standard metrics - peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM). We further test the statistical significance of the differences using repeated measures analysis of variance (RM-ANOVA) with the MedCalc software (Schoonjans et al., 1995). While these are the two most widely used metrics in the community, it is also well known that these metrics hardly line up with the radiologists' scoring on the image quality Mason et al. (2019).

To fully capture the superiority of the proposed method, we focus on the fact that *good* reconstructions are the ones that can be used for *accurate diagnosis*. If the reconstruction quality closely matches the ground truth, there should be no degeneration in the performance of the downstream tasks - in our

<sup>2</sup>[https://github.com/yang-song/score\\_sde\\_pytorch](https://github.com/yang-song/score_sde_pytorch)

<sup>3</sup><https://github.com/mikgroup/sigpy>

<sup>4</sup><https://github.com/facebookresearch/fastMRI>

<sup>5</sup><https://github.com/bbbbbbzhou/DuDoRNet>

case diagnosis. In order to compare against ground truth, the diagnostic ability of each reconstruction, we leverage the recent fastMRI+ (Zhao et al., 2021) dataset. For fastMRI knee data, fastMRI+ annotations provide bounding boxes around the pathologic region. We train a standard object detection model using the ground truth (fully sampled) images and use this network to compare how well the model performs on pathology detection with reconstructions using different methods. By doing so, we can measure the amount of distribution shift that occurs with each reconstruction method. The less the difference in the performance, we conclude that there is less distribution shift from the fully-sampled data. For detailed experimental procedures, see the following section, i.e. Section 4.6. From the experiments, we quantify three standard metrics from the object detection literature - mean average precision (mAP), precision, and recall.

#### 4.6. Pathology Detection

For the object detection model, we use the state-of-the-art YOLO v5<sup>6</sup>. We use the default configuration of YOLOv5m, which is a medium-sized model, often suggested as the baseline model when you do not have a sufficient amount of data. When trying to fit larger versions of the model, namely YOLOv5l, YOLOv5x, etc., we found that overfitting occurs and the performance drops by a small margin.

For training data of the YOLOv5 model, we use all the training data with annotations in fastMRI+, which consists of 8053 images in total. We do not include any images without annotations in the training set. For testing, we select 15 random cases out of the validation set.

Model weights were fine-tuned from the open-source pre-trained model, and were trained for 300 epochs using the batch size of 16. Training took about a day on 2×2080Ti GPU.

#### 4.7. Generalization to different anatomy and contrast

One observation that can be made is that the proposed method, which utilizes the score function as the main workhorse of the algorithm, is robust to distribution shifts. Otherwise, Algorithms 3,4,5 would not have worked in the first place, since the training data distribution and the inference data distribution are different. Subsequently, one might wonder how far we are able to push this discrepancy and still achieve satisfactory results. To further investigate the generalization capability of the proposed method, we ran extensive experiments with data collected from different anatomy and contrast. To achieve maximal diversity, we collected data from various open-source database, including mridata (Flynn et al.), human connectome project (HCP) MRI dataset (<http://db.humanconnectome.org>), <http://github.com/hkaggarwal/modl>, and MASSIVE<sup>7</sup> (<http://massive-data.org/index.html>). For experiments, all data were retrospectively down-sampled from the fully-sampled  $k$ -space.

## 5. Results

### 5.1. Real-valued Simulation study

First, we show the results of the simulation study using retrospectively under-sampled fastMRI real-valued data of size 320×320 in Fig. 3. In the first row, we see reconstructions from 2D ×8 Gaussian random sampling, which is the sub-sampling pattern that induces the least aliasing artifact. Here, we see that the proposed performs nearly perfect reconstruction, where we see virtually no structural difference even in the zoomed-in image. Reconstruction through total variation (TV) induces cartoon-like artifacts, and cannot remove the overall foggy artifacts across the whole image. U-Net and DuDoRNet produce blurry reconstructions that are suboptimal, which indicates that they cannot adapt to different sampling patterns that were not seen from the training process.

In the second row, we compare the reconstructions from acceleration using ×4 uniform random sampling with 4% of the phase encoding lines kept as the autocalibrating signal (ACS) region, producing moderate aliasing artifacts. Even at this level, our method is able to provide accurate reconstruction, capturing most of the high-frequency details of the ground truth. In this acceleration factor, TV is not able to completely remove the aliasing artifacts. The averaging effect of supervised reconstruction now becomes more clear, often omitting the important details of the scan. Reconstruction with DuDoRNet also suffers from leftover aliasing artifacts in this regime, even though this is just a slight difference in the sampling pattern.

The third row shows ×8 Gaussian random sampling (4% ACS region). TV only partially removes the artifacts, while introducing quite an amount of cartoon-like artifacts. Supervised U-Net and DuDoRNet smooths away a lot of details, producing unrealistic texture. Surprisingly, even at this level of acceleration factor, our method provides a fairly accurate reconstruction, matching most of the details from the ground truth.

In the last row, we see the most aggressive under-sampling factor - ×15 variable density poisson disk under-sampling. Thanks to the evenly spread out  $k$ -space samples, this sampling scheme is considered one of the state-of-the-art amongst the various sampling patterns, which produce the least artifacts given the same budget (Dwork et al., 2021). Here, similar to what was seen with 2D Gaussian sampling, only the proposed method is able to produce high-fidelity image reconstruction. Other methods still suffer from severe artifacts, with blurred-out details.

We provide a thorough comparison of quantitative metrics on the test set in Table 1. On all the different sampling patterns, our method significantly outperforms the comparison methods and is on par with the state-of-the-art DuDoRNet (Zhou and Zhou, 2020) in Gaussian 1D sampling pattern. Results of RM-ANOVA also indicates that the superiority of the proposed method is statistically significant against all other methods in most cases.

It is worth mentioning the overall difference between the reconstructions through supervised methods (i.e., U-Net, DuDoRNet), and the proposed method. We observe that the reconstructions using supervised U-Net become blurrier as we push the acceleration factor to higher values. This is a widely known

<sup>6</sup><https://github.com/ultralytics/yolov5>

<sup>7</sup>Multiple Acquisition for Standardization of Structural Imaging Validation and Evaluation

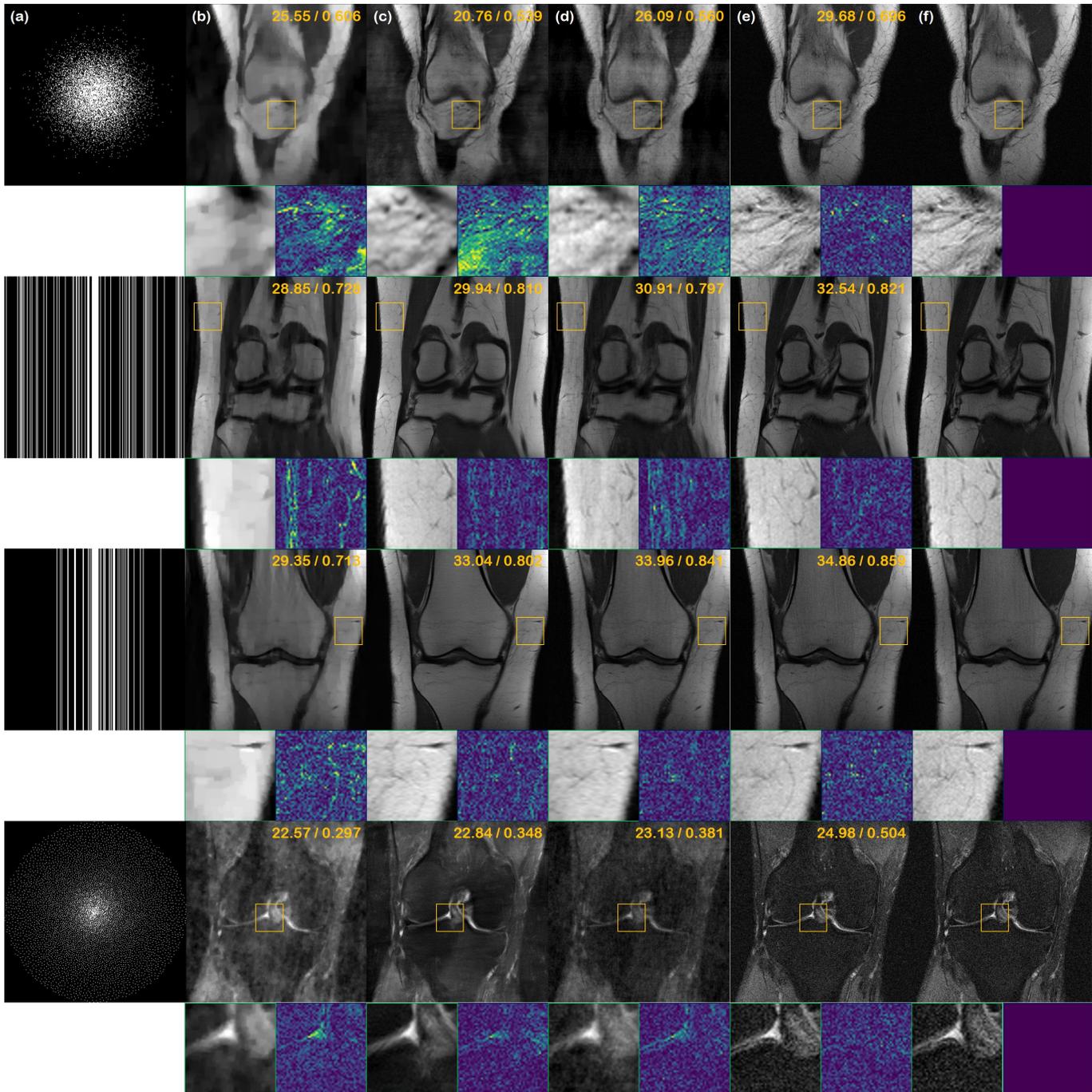


Fig. 3: Reconstructions of the real-valued simulation study. (a) Sub-sampling mask used to generate under-sampled image, (b) TV, (c) supervised learning (U-Net) (d) DuDoRNet (Zhou and Zhou, 2020), (e) proposed method, and (f) ground truth. 1<sup>st</sup> row: 2D  $\times 8$  Gaussian random sampling, 2<sup>nd</sup> row: 1D  $\times 4$  uniform random sampling, 3<sup>rd</sup> row: 1D  $\times 8$  Gaussian random sampling, 4<sup>th</sup> row:  $\times 15$  variable density poisson disk sampling. Green box: Zoom in version of the indicated yellow box, Blue box: Difference magnitude of the inset (in *Viridis* colormap). Yellow numbers in the upper right corner indicate PSNR [db], and SSIM, respectively.

effect for models that are trained with supervision using L1 loss, L2 loss, etc (Ledig et al., 2017). Typically, this behavior stems from the models collapsing to a single mode of the distribution, whereas the actual distribution is highly multi-modal. On the contrary, our method does not show such an effect, visiting the modes of high probability in each sample. For that matter, our method is able to 1) reconstruct high-frequency details even at high acceleration factors, and 2) quantify uncertainty, as we discuss in-depth in Section 5.5

## 5.2. Complex-valued Single-coil Reconstruction

In Fig. 4, we illustrate the reconstruction results of the proposed method using Algorithm 3, along with the baseline comparisons. In the first row, we have  $\times 8$  2D Gaussian random sampling acceleration. The proposed method performs a very accurate reconstruction, without inducing any additional blurriness from the reconstruction process. TV, supervised U-Net, and DuDoRNet fall largely behind the proposed method both in terms of perceptual quality and quantitative metrics.

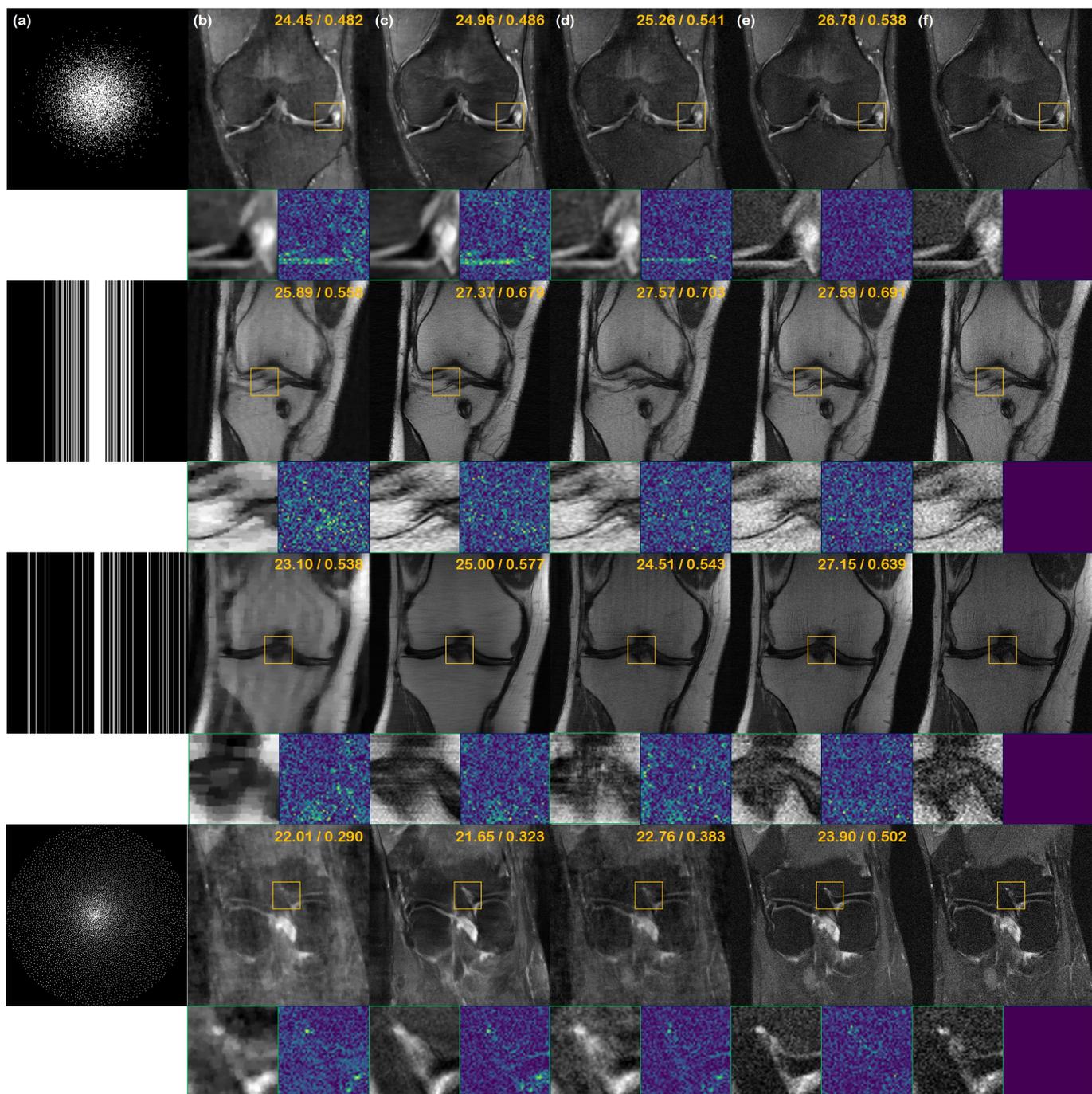


Fig. 4: Single-coil complex-valued image reconstruction results. (a) Sub-sampling mask used to generate under-sampled image, (b) TV, (c) supervised learning (U-Net) (d) DuDoRNet (Zhou and Zhou, 2020), (e) proposed method, and (f) ground truth. 1<sup>st</sup> row: 2D  $\times 8$  Gaussian random sampling, 2<sup>nd</sup> row: 1D  $\times 4$  Gaussian random sampling, 3<sup>rd</sup> row: 1D  $\times 8$  uniform random sampling, 4<sup>th</sup> row:  $\times 15$  variable density poisson disk sampling. Green box: Zoom in version of the indicated yellow box, Blue box: Difference magnitude of the inset (in *Viridis* colormap). Yellow numbers in the upper right corner indicate PSNR [db], and SSIM, respectively.

In the second row, we have  $\times 4$  1D Gaussian random sampling (8% ACS region) acceleration. Again, the proposed method reconstructs the aliased image with high accuracy and has higher quality than the comparison methods.

The third row shows  $\times 8$  1D uniform random sampling (4% ACS region). The proposed method is still able to reconstruct the aliased image with virtually no degradation in the high-frequency details. On the other hand, supervised U-Net recon-

struction produces horizontal strip artifacts, hampering the visual quality. DuDoRNet does not remove the aliasing artifacts completely. TV hardly produces satisfactory results. The final row shows reconstructions with  $\times 15$  VD poisson sampling. Due to the bias toward 1D sampling patterns, both U-Net and DuDoRNet produce unsatisfactory results with washed-out details. Our method is able to preserve sharp edges and texture, clearly depicting the anatomical structure. Quantitative metrics

|                   |             |           |               | TV            | Supervised U-Net     | DuDoRNet             | Proposed             | $p^* < 0.0001$      |
|-------------------|-------------|-----------|---------------|---------------|----------------------|----------------------|----------------------|---------------------|
| Simulation (real) | Uniform 1D  | × 4       | PSNR [db]     | 27.55 ± 1.60  | 29.21 ± 2.09         | 30.79 ± 0.91         | <b>31.10</b> ± 1.60  | TV, U-Net           |
|                   |             |           | SSIM          | 0.667 ± 0.045 | 0.788 ± 0.030        | <b>0.796</b> ± 0.022 | 0.795 ± 0.036        |                     |
|                   | Uniform 1D  | × 8       | PSNR [db]     | 26.35 ± 1.74  | 27.40 ± 2.39         | 24.08 ± 1.80         | <b>28.37</b> ± 1.97  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.631 ± 0.052 | 0.722 ± 0.042        | 0.603 ± 0.037        | <b>0.771</b> ± 0.051 |                     |
|                   | Gaussian 1D | × 4       | PSNR [db]     | 30.77 ± 1.24  | 32.85 ± 1.25         | 33.01 ± 1.05         | <b>33.32</b> ± 1.22  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.752 ± 0.029 | 0.829 ± 0.023        | <b>0.858</b> ± 0.016 | 0.825 ± 0.028        |                     |
|                   | Gaussian 1D | × 8       | PSNR [db]     | 28.87 ± 1.56  | 30.81 ± 1.37         | 30.46 ± 1.28         | <b>30.94</b> ± 1.17  | TV, DuDoRNet        |
|                   |             |           | SSIM          | 0.70 ± 0.041  | <b>0.779</b> ± 0.029 | 0.776 ± 0.025        | 0.761 ± 0.025        |                     |
|                   | Gaussian 2D | × 8       | PSNR [db]     | 23.19 ± 2.30  | 21.92 ± 4.51         | 25.29 ± 4.16         | <b>29.95</b> ± 2.04  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.599 ± 0.082 | 0.573 ± 0.173        | 0.643 ± 0.077        | <b>0.701</b> ± 0.071 |                     |
| Gaussian 2D       | × 15        | PSNR [db] | 18.20 ± 2.80  | 17.43 ± 0.45  | 21.24 ± 2.09         | <b>29.58</b> ± 1.47  | TV, U-Net, DuDoRNet  |                     |
|                   |             | SSIM      | 0.456 ± 0.096 | 0.451 ± 0.158 | 0.518 ± 0.046        | <b>0.678</b> ± 0.052 |                      |                     |
| VD Poisson disk   | × 8         | PSNR [db] | 21.26 ± 0.96  | 23.47 ± 1.07  | 23.22 ± 0.93         | <b>31.83</b> ± 1.15  | TV, U-Net, DuDoRNet  |                     |
|                   |             | SSIM      | 0.516 ± 0.038 | 0.606 ± 0.083 | 0.516 ± 0.063        | <b>0.769</b> ± 0.030 |                      |                     |
| VD Poisson disk   | × 15        | PSNR [db] | 19.56 ± 0.51  | 20.97 ± 1.73  | 22.84 ± 2.34         | <b>30.46</b> ± 1.21  | TV, U-Net, DuDoRNet  |                     |
|                   |             | SSIM      | 0.517 ± 0.042 | 0.620 ± 0.054 | 0.561 ± 0.063        | <b>0.709</b> ± 0.035 |                      |                     |
| Single-coil       | Uniform 1D  | × 4       | PSNR [db]     | 27.13 ± 1.91  | 30.90 ± 1.78         | 30.43 ± 0.79         | <b>31.95</b> ± 1.45  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.636 ± 0.041 | 0.801 ± 0.027        | 0.793 ± 0.024        | <b>0.812</b> ± 0.036 |                     |
|                   | Uniform 1D  | × 8       | PSNR [db]     | 24.38 ± 2.01  | 27.48 ± 1.66         | 24.72 ± 1.89         | <b>27.97</b> ± 2.03  | TV, DuDoRNet        |
|                   |             |           | SSIM          | 0.601 ± 0.090 | 0.720 ± 0.039        | 0.641 ± 0.055        | <b>0.738</b> ± 0.053 |                     |
|                   | Gaussian 1D | × 4       | PSNR [db]     | 28.39 ± 1.09  | 32.86 ± 1.23         | 33.46 ± 1.35         | <b>33.96</b> ± 1.27  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.679 ± 0.077 | 0.828 ± 0.024        | <b>0.856</b> ± 0.022 | 0.858 ± 0.028        |                     |
|                   | Gaussian 1D | × 8       | PSNR [db]     | 25.91 ± 3.21  | 30.80 ± 1.34         | 29.65 ± 1.76         | <b>30.82</b> ± 1.37  | TV, DuDoRNet        |
|                   |             |           | SSIM          | 0.622 ± 0.050 | <b>0.777</b> ± 0.028 | 0.777 ± 0.028        | 0.762 ± 0.034        |                     |
|                   | Gaussian 2D | × 8       | PSNR [db]     | 20.09 ± 6.13  | 19.99 ± 5.12         | 21.53 ± 10.86        | <b>29.45</b> ± 2.97  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.592 ± 0.140 | 0.520 ± 0.187        | 0.541 ± 0.152        | <b>0.676</b> ± 0.118 |                     |
|                   | Gaussian 2D | × 15      | PSNR [db]     | 17.99 ± 3.15  | 17.24 ± 4.01         | 18.86 ± 5.51         | <b>26.15</b> ± 4.44  | TV, U-Net, DuDoRNet |
|                   |             |           | SSIM          | 0.460 ± 0.250 | 0.436 ± 0.156        | 0.490 ± 0.192        | <b>0.587</b> ± 0.148 |                     |
| VD Poisson disk   | × 8         | PSNR [db] | 20.90 ± 3.92  | 19.67 ± 1.72  | 22.91 ± 2.54         | <b>31.50</b> ± 1.24  | TV, U-Net, DuDoRNet  |                     |
|                   |             | SSIM      | 0.626 ± 0.101 | 0.604 ± 0.067 | 0.643 ± 0.041        | <b>0.762</b> ± 0.030 |                      |                     |
| VD Poisson disk   | × 15        | PSNR [db] | 22.13 ± 1.92  | 21.67 ± 1.06  | 23.95 ± 1.13         | <b>30.66</b> ± 1.30  | TV, U-Net, DuDoRNet  |                     |
|                   |             | SSIM      | 0.616 ± 0.089 | 0.573 ± 0.098 | 0.593 ± 0.032        | <b>0.706</b> ± 0.035 |                      |                     |

Table 1: Quantitative metrics of real-valued simulation study and single-coil complex-valued image experiment. Numbers are presented as mean value ± unit standard deviation. Numbers in bold face indicate the best metric out of all the methods.  $p$ -values indicate results from RM-ANOVA. Proposed method statistically relevant against methods listed in the last column in terms of PSNR. \*: Bonferroni corrected.

|                 |             |           |               | TV            | Supervised U-Net | E2E-Varnet           | Proposed              | $p^* < 0.0001$        |
|-----------------|-------------|-----------|---------------|---------------|------------------|----------------------|-----------------------|-----------------------|
| Multi-coil      | Uniform 1D  | × 4       | PSNR [db]     | 27.32 ± 1.30  | 31.77 ± 2.30     | 32.96 ± 1.46         | <b>33.25</b> ± 1.42   | TV, U-Net             |
|                 |             |           | SSIM          | 0.662 ± 0.112 | 0.846 ± 0.031    | 0.856 ± 0.072        | <b>0.857</b> ± 0.201  |                       |
|                 | Uniform 1D  | × 8       | PSNR [db]     | 25.02 ± 2.90  | 29.51 ± 3.99     | 31.98 ± 1.59         | <b>32.01</b> ± 1.55   | TV, U-Net             |
|                 |             |           | SSIM          | 0.532 ± 0.158 | 0.780 ± 0.058    | <b>0.828</b> ± 0.051 | 0.821 ± 0.071         |                       |
|                 | Gaussian 1D | × 4       | PSNR [db]     | 30.55 ± 1.75  | 32.66 ± 1.18     | 33.15 ± 2.92         | <b>34.25</b> ± 1.74   | TV, U-Net, E2E-Varnet |
|                 |             |           | SSIM          | 0.789 ± 0.128 | 0.866 ± 0.020    | 0.878 ± 0.062        | <b>0.885</b> ± 0.024  |                       |
|                 | Gaussian 1D | × 8       | PSNR [db]     | 27.98 ± 2.12  | 31.64 ± 1.27     | <b>33.15</b> ± 2.64  | 32.43 ± 1.69          | TV, U-Net             |
|                 |             |           | SSIM          | 0.747 ± 0.114 | 0.841 ± 0.021    | <b>0.868</b> ± 0.069 | 0.855 ± 0.021         |                       |
|                 | Gaussian 2D | × 8       | PSNR [db]     | 29.20 ± 1.85  | 24.51 ± 3.16     | 20.97 ± 2.98         | <b>31.43</b> ± 1.70   | TV, U-Net, E2E-Varnet |
|                 |             |           | SSIM          | 0.781 ± 0.012 | 0.724 ± 0.103    | 0.642 ± 0.085        | <b>0.831</b> ± 0.036  |                       |
| Gaussian 2D     | × 15        | PSNR [db] | 26.28 ± 1.02  | 14.93 ± 0.95  | 16.66 ± 3.01     | <b>29.17</b> ± 2.19  | TV, U-Net, E2E-Varnet |                       |
|                 |             | SSIM      | 0.547 ± 0.072 | 0.372 ± 0.070 | 0.435 ± 0.106    | <b>0.704</b> ± 0.042 |                       |                       |
| VD Poisson disk | × 8         | PSNR [db] | 29.52 ± 1.39  | 20.89 ± 1.10  | 20.70 ± 1.20     | <b>31.98</b> ± 2.07  | TV, U-Net, E2E-Varnet |                       |
|                 |             | SSIM      | 0.562 ± 0.078 | 0.576 ± 0.063 | 0.592 ± 0.045    | <b>0.816</b> ± 0.022 |                       |                       |
| VD Poisson disk | × 15        | PSNR [db] | 26.19 ± 1.19  | 16.01 ± 1.95  | 18.82 ± 1.85     | <b>29.59</b> ± 2.22  | TV, U-Net, E2E-Varnet |                       |
|                 |             | SSIM      | 0.510 ± 0.093 | 0.537 ± 0.077 | 0.548 ± 0.050    | <b>0.702</b> ± 0.028 |                       |                       |

Table 2: Quantitative metrics of multi-coil parallel imaging. Numbers are presented as mean value ± unit standard deviation. Numbers in bold face indicate the best metric out of all the methods.  $p$ -values indicate results from RM-ANOVA. Proposed method statistically relevant against methods listed in the last column in terms of PSNR. \*: Bonferroni corrected.

in Table 1 also confirm the superiority.

As mentioned earlier, all the results depicted in Fig. 4 were generated using the single score function, trained with DICOM images only. This means that the model has never seen complex-valued data before, and yet the quality of reconstruction is surprisingly high. This is an important advantage of our

model over conventional DL methods. Most DL methods require raw  $k$ -space data to train the network, and this is hard to achieve since most raw data are discarded after the scan (Zbontar et al., 2018). In contrast, DICOM images are relatively much easier to collect, and this enables practitioners to collect a large database for training.

### 5.3. Complex-valued Multi-coil Reconstruction

We compare the results of PI reconstruction in Fig. 5. Consistent with the results from the prior sections, our method significantly outperforms other methods and is the only method that produces high fidelity reconstructions regardless of the sampling pattern. This can also be seen in Table 2, where we see that our method is on par with the SOTA E2E-varnet on 1D sampling patterns, while significantly outperforming all the methods in 2D sampling patterns.

The bias towards 1D sampling pattern for the deep-learning based method is much larger in the multi-coil case. This can be seen in the severe artifacts that can be seen in reconstructions via E2E-varnet (Fig. 5 (d)). We conjecture that this is because the sensitivity map estimation module of E2E-varnet fails dramatically with 2D sampling patterns, leading to horizontal streaking artifacts. Furthermore, although U-Net does not explicitly deal with sensitivity maps, the bias towards certain sampling patterns seems to be greater than in the single-coil experiments, showing reconstructions with heavy artifacts. On the other hand, our score-based method is agnostic to sampling patterns, clearly outperforming the comparison methods in all cases.

However, in the case of PI, we do observe some potential limitations of our method. Namely, since we consider stochastic samples from the conditional distribution for each coil, and then take the SSOS of the coil images at the last step to form the reconstruction, we recognize a small amount of averaging effect compared to single-coil reconstruction cases. Nevertheless, we emphasize that our method was not trained with multi-coil data, and is yet able to reconstruct PI data with state-of-the-art performance. Our method works with any arbitrary number of coils, without the need to calibrate sensitivity maps, thereby enabling a practical application to any type of scan.

### 5.4. Pathology detection

Overall results on the pathology detection task is illustrated in Fig. 6, and the quantitative metric is shown in Fig. 7. The images in Fig. 6 (a-d) were reconstructed from  $\times 4$  1D Gaussian random under-sampling (8% ACS region), (e) is the detection result using fully-sampled data, and (f) is the ground-truth bounding boxes along with the corresponding labels.

The patient scan in the first row of Fig. 6(f) shows grade 1 sprain on the posterior cruciate ligament (PCL). Moreover, both on the medial and the lateral side of the cartilage, partial-thickness loss/defect is observed. This is accurately captured in Fig. 6 (e), where all three bounding boxes are correctly predicted. Reconstruction with the proposed method come as close, with virtually no difference with the fully-sampled reference (see Fig. 6(c)). Reconstruction with supervised U-Net performs much worse on this downstream task, missing the cartilage partial thickness loss/defect (Fig. 6(b)). DuDoRNet reconstruction gets only one of the cartilage partial thickness loss/defect, while falsely detecting meniscus tears. As expected, diagnosis with TV reconstruction performs even worse, falsely predicting meniscus tear on both medial/lateral sides.

The patient scan in the second row of Fig. 6(f) show a grade 1 sprain on the anterior cruciate ligament (ACL). On

both fully-sampled data, and reconstruction using the proposed method in Fig. 6(e),(d), respectively, the model correctly predicts ACL low-grade sprain, with small differences in the size of the bounding box. Both models also falsely predict meniscus tear, which seems reasonable given that the model is not perfect. Contrarily, reconstruction using supervised U-Net in Fig. 6(b) does not capture ACL low-grade sprain, and only falsely estimates meniscus tear. DuDoRNet comes similar to fully-sampled data, but is confused at the exact location of the meniscus tear. TV reconstruction in Fig. 6(a) only falsely predicts cartilage loss/defect, missing the true lesion.

As can be seen from standard metrics in Fig. 7, the mAP metric using the fully-sampled data reaches 0.754, which serves as the upper bound for the detection model. Detection using the proposed reconstructions performs on par, or sometimes even better than the fully-sampled reference. Since the model is imperfect and the test dataset size is relatively small, this does not mean that the proposed method is better than the fully-sampled data. However, it does mean that reconstruction via the proposed method does not hamper the diagnostic capability, which is important in clinical practice. On the other hand, the comparison methods do limit the diagnostic capability, and thus should be questioned before usage in clinical practice. For further statistical analysis, see Appendix.

### 5.5. Quantifying uncertainty of the prediction

Our method is a generative algorithm, with two sources of stochasticity. First, the sample starts from a randomly sampled vector  $x_N$ . Second, both predictor and corrector steps involve sampling random noise vectors and adding them to the estimate. Therefore, the iterative procedure of the proposed algorithms typically converges to different outcomes.

Due to this generative nature, we can run multiple reconstructions in parallel, and quantify the uncertainty of the prediction, as depicted in Fig. 8. Here, the ground truth, and the aliased images are shown in Fig. 8 (a),(b), respectively. For the experiment, we take a batch size of 8, and run the reconstruction in parallel. The mean value of the reconstruction is shown in Fig. 8 (c), and the pixel-wise standard deviation values are shown in Fig. 8 (d). At low acceleration factors ( $\times 2$ ), we see very little variation between the different reconstructions. This indicates high confidence of the model, and hence we can conclude that the reconstruction is exact in all parts of the image. As the acceleration factor is increased, and the degree of aliasing artifacts become more severe, we see that the uncertainty increase in specific regions. Potentially, this measure of uncertainty can inform the practitioners on how much they should rely on the reconstruction, thereby deciding whether to use a different diagnostic tool. We additionally provide specific realizations, rather than the mean of the reconstruction, at each acceleration factor in supplementary video 1.

### 5.6. Reconstruction out-of-distribution (OOD) data

In Fig. 9, we present reconstruction results, which are heavily out of distribution. Note that our score function has only learnt the distribution of proton density (PD) / proton density fat suppression (PDFS) coronal knee scans. Nevertheless, we observe

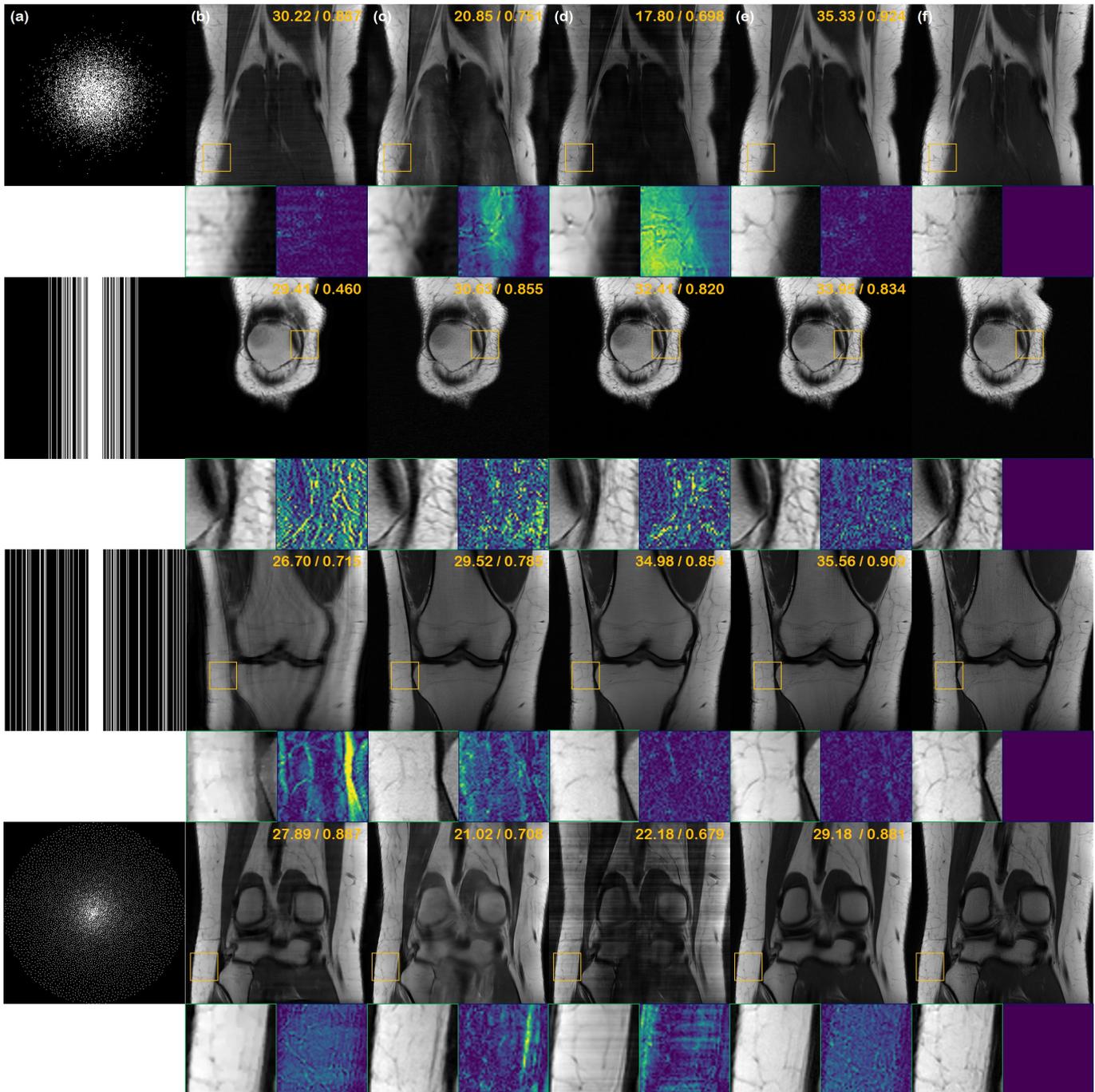


Fig. 5: Multi-coil reconstruction results. (a) Sub-sampling mask used to generate under-sampled image, (b) TV, (c) supervised learning (U-Net), (d) E2E-varnet (Sriram et al., 2020), (e) the proposed method, and (f) the ground truth. 1<sup>st</sup> row: 2D  $\times$ 8 Gaussian random sampling, 2<sup>nd</sup> row: 1D  $\times$ 4 Gaussian random sampling, 3<sup>rd</sup> row: 1D  $\times$ 4 uniform random sampling, 4<sup>th</sup> row:  $\times$ 8 variable density poisson disk sampling. Green box: Zoom in version of the indicated yellow box, Blue box: Difference magnitude of the inset (in Viridis colormap). Yellow numbers in the upper right corner indicate PSNR [db], and SSIM, respectively.

that with the proposed method, we are able to achieve high fidelity reconstructions regardless of the anatomy and contrast. While other methods such as U-Net and DuDoRNet generalize to a certain extent, we can clearly observe leftover aliasing artifacts. On the other hand, the proposed method clearly outperforms all the other methods with minimal residuals and sharp contrast. We recently found that this observation was also made independently in (Jalal et al., 2021), where the authors partially proved that posterior sampling is indeed highly robust

to distribution shifts. This property is indeed very advantageous in real-world settings, since one may be able to use a single neural network regardless of the specific anatomy and contrast. For further experimental results of different anatomy, please see Fig. 14 and Fig. 15 in the Appendix.

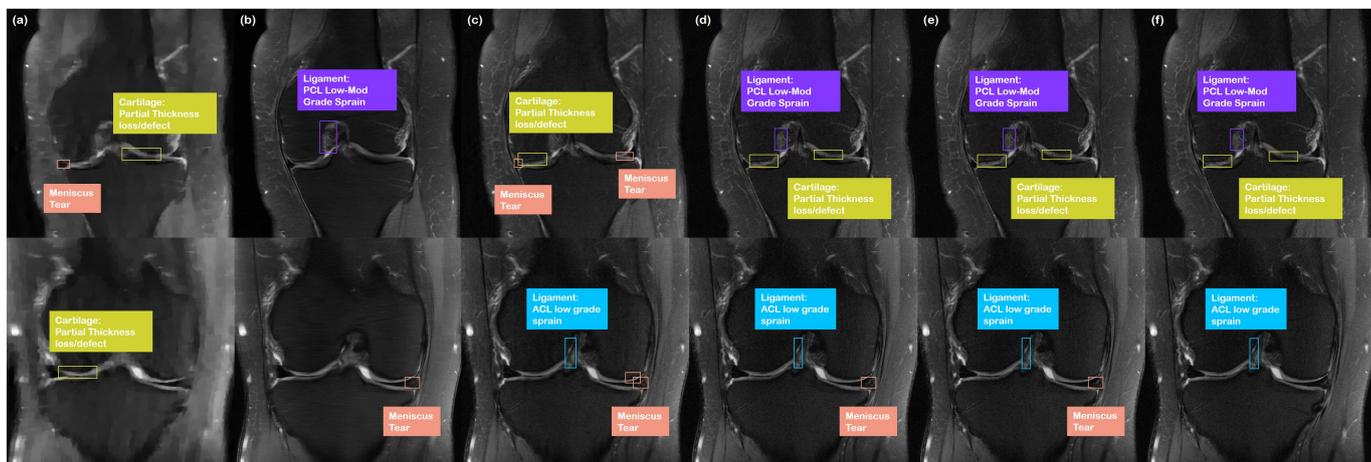


Fig. 6: Results of pathology detection. Detection using (a) TV reconstruction, (b) supervised U-Net, (c) DuDoRNet, (d) proposed method, (e) fully-sampled images. Ground-truth label for the pathologies are shown in (f). (Yellow green box): Cartilage partial thickness loss/defect, (Pink box): Meniscus tear, (Purple box): Ligament PCL Low-Mod grade sprain, (Skyblue box): Ligament ACL low grade sprain.

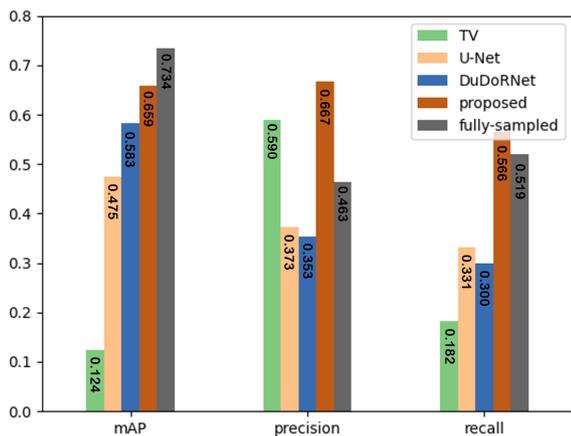


Fig. 7: Quantitative metrics of pathology detection.

## 6. Discussions

### 6.1. Speeding up inference

One obvious limitation of using score-based diffusion models for image reconstruction is the time required for inference. As stated in Section 4.2, it requires about 10 minutes of inference time for using  $N = 2000$  discretization steps. A naïve way for faster inference is to reduce the number of discretization steps, and we provide the trade-off between image quality vs. steps in Fig. 10. Here, we observe that naïvely interleaving the discretization steps work quite well, with minimal compromise in image quality. This is especially the case of low acceleration factors (e.g.  $\times 4$ ), where we achieve highly accurate reconstruction only with 50 iterations. As the acceleration factor gets aggressive, our method typically requires more iterations for maximal performance. However, from the figure, we see that the performance caps at about  $N = 500$ . One could possibly adjust this as a hyperparameter on-the-fly, according to the degree of acceleration at hand.

One can also employ the state-of-the-art acceleration strategy of diffusion models for inverse problems, called come-

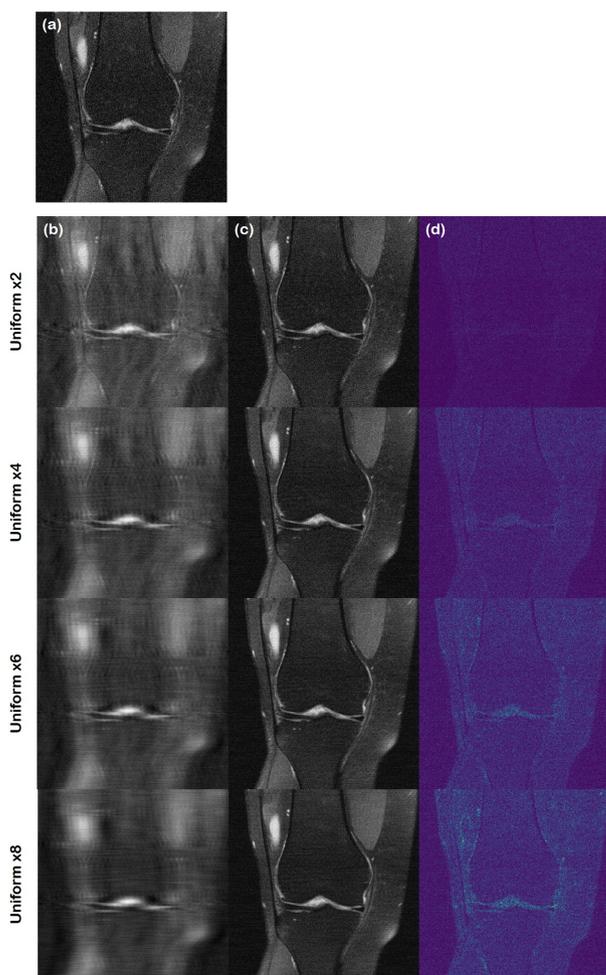


Fig. 8: Quantifying the uncertainty of reconstruction. (a) Ground truth, (b) aliased image from sub-sampling, (c) mean of the reconstruction, (d) standard deviation of the samples: range is set to  $[0, 0.02]$  (on Viridis colormap). From the 1<sup>st</sup> row to the 4<sup>th</sup> row, the acceleration factor grows from  $\times 2$  to  $\times 8$ .

closer-diffuse-faster (CCDF), which was recently proposed in our companion work (Chung et al., 2021). Specifically, CCDF

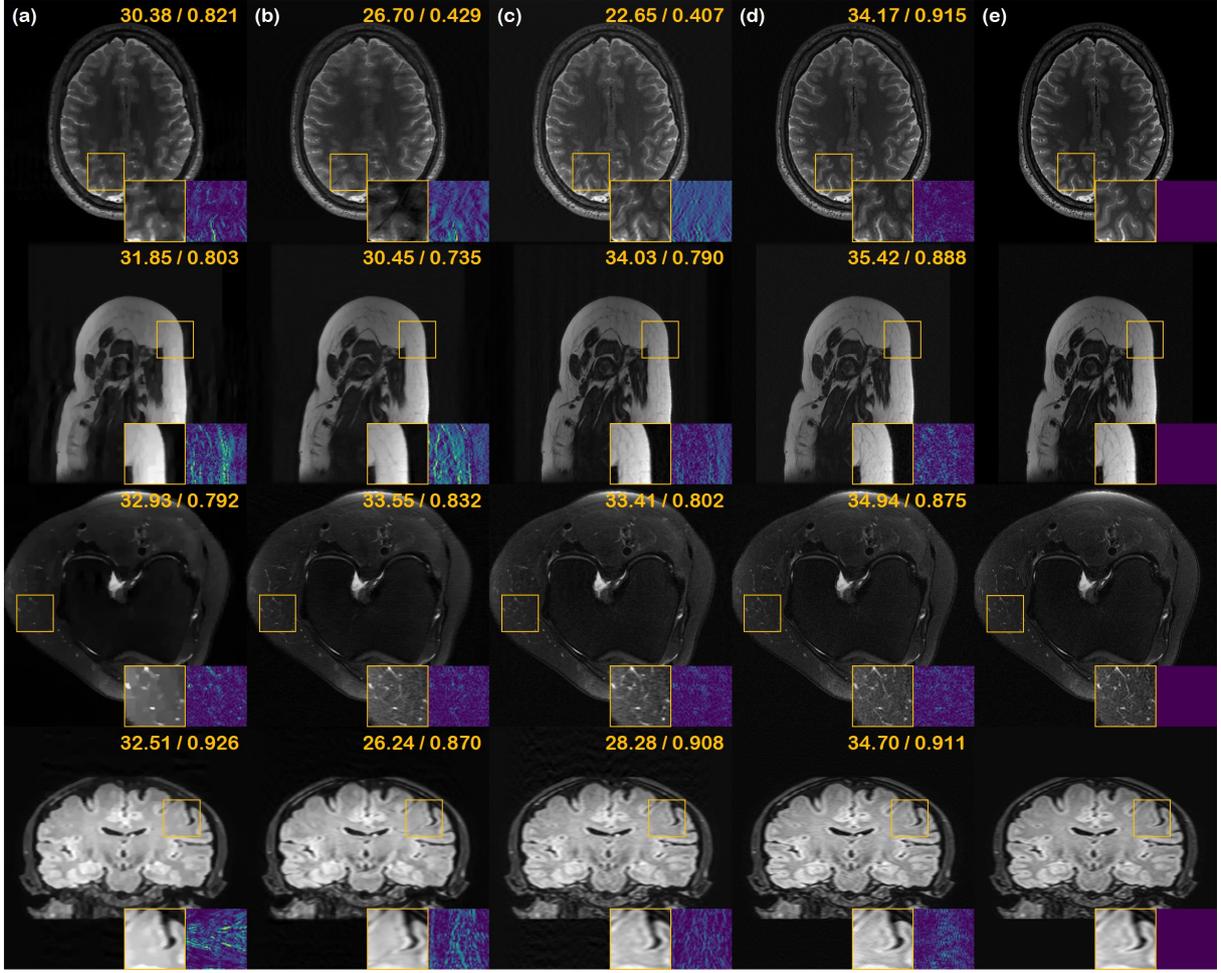


Fig. 9: Reconstruction results of various anatomic structure/contrast. (a) TV, (b) U-Net, (c) DuDoRNet, (d) proposed method, (e) ground truth. 1<sup>st</sup> row: HCP axial brain scan, 2<sup>nd</sup> row: MRI scan of left leg collected from (Flynn et al.), 3<sup>rd</sup> row: axial knee scan from (Flynn et al.), 4<sup>th</sup> row: MASSIVE coronal brain scan. 1D Gaussian  $\times 4$  under-sampling was performed.

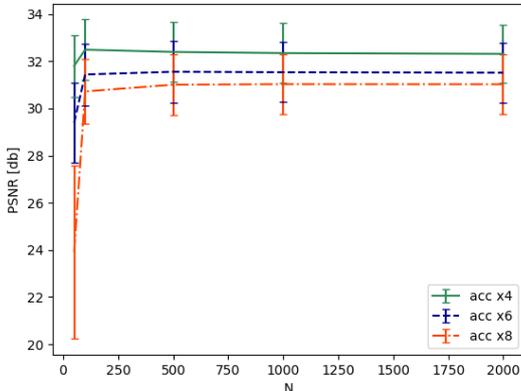


Fig. 10: Tradeoff between number of iterations vs. PSNR [db]. Errorbars indicate unit standard deviation. Sampling scheme used for this ablation study is 1D Gaussian random under-sampling.

states that there is no need to use the full reverse diffusion procedure. Rather, one can start to apply reverse diffusion from a forward-diffused image from better an initialization to achieve reconstruction performance that is one par or *better*. The re-

---

#### Algorithm 6 CCDF sampling (Chung et al., 2021)

---

**Require:**  $s_\theta, N', M, \{\epsilon_i\}_{i=1}^{N'}, \{\sigma_i\}_{i=1}^{N'}, \mathbf{x}_0$

- 1: **Define**  $A := \mathcal{P}_\Omega \mathcal{F}$
- 2:  $\mathbf{x}_{N'} \leftarrow \mathbf{x}_0 + \sigma_{N'} \mathbf{z}$
- 3: **for**  $i = N' - 1 : 0$  **do**
- 4:    $\mathbf{x}_i \leftarrow \text{Predictor}(\mathbf{x}_{i+1}, \sigma_i, \sigma_{i+1})$
- 5:    $\mathbf{x}_i \leftarrow \text{Re}(\mathbf{x}_i + A^*(\mathbf{y} - A\mathbf{x}_i))$
- 6:   **for**  $j = 1 : M$  **do**
- 7:      $\mathbf{x}_i \leftarrow \text{Corrector}(\mathbf{x}_i, \sigma_i, \epsilon_i)$
- 8:      $\mathbf{x}_i \leftarrow \text{Re}(\mathbf{x}_i + A^*(\mathbf{y} - A\mathbf{x}_i))$
- 9:   **end for**
- 10: **end for**
- 11: **return**  $\mathbf{x}_0$

---

sulting short partial time horizon significantly accelerates the reconstruction time. The strategy is backed by rigorous proofs using the theory of stochastic contraction. Interested readers are referred to Chung et al. (2021).

More specifically, one specifies a significantly short timestep  $t_0 < T$  (in the discretized setting, this corresponds to  $N' := t_0 N$ ). Then, the initial reconstruction  $\mathbf{x}_0$  is forward-diffused

with the prescribed forward SDE in a single step. In the case of VE-SDE, this corresponds to

$$\mathbf{x}_{N'} = \mathbf{x}_0 + \sigma_{N'} \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

One can then follow the reverse SDE for  $t \in [0, t_0]$  running backwards in time as in Algorithms 2,3,4,5.

For simplicity, here we present CCDF adopted to Algorithm 2, which is shown in Algorithm 6. For the initialization of  $\mathbf{x}_0$ , we resort to U-Net. With this simple combination of the pre-trained feed forward neural network, we observe that it is possible to use as little as *40 iterations* (corresponding to  $0.02 \times \text{NFEs}^8$ ) with similar or better performance, as reported in Table 3.

| method     | TV    | U-Net | DuDoRNet | proposed<br>(2000) | CCDF<br>(40) |
|------------|-------|-------|----------|--------------------|--------------|
| $\times 4$ | 30.77 | 32.85 | 33.01    | 33.32              | <b>34.11</b> |
| $\times 8$ | 28.87 | 30.81 | 30.46    | 30.94              | <b>32.08</b> |

Table 3: Effect of acceleration using the CCDF Chung et al. (2021) strategy to the proposed method. PSNR( $\uparrow$ ) on Gaussian 1D under-sampling masks with different acceleration factors are reported. Number in parenthesis indicate the number of iterations used. Numbers in boldface indicate the best results among the rows.

## 6.2. Conditional generation with diffusion models

Score-based diffusion models are now one of the most popular methods in the context of image synthesis, matching the image fidelity of state-of-the-art GANs (Dhariwal and Nichol, 2021), and achieving the state-of-the-art log-likelihood on various datasets (Kingma et al., 2021; Kim et al., 2021). The interest in using these models for conditional image generation is also rising. Song and Ermon (2019) first proposed to use score models trained with discrete de-noising score matching for image inpainting. This was further developed in (Song et al., 2021b) for image colorization, and class-conditional image synthesis, using continuous-time score models. The same group published a work for image editing (Meng et al., 2021) using VE-SDEs, which uses a similar algorithm to image inpainting used in (Song and Ermon, 2019; Song et al., 2021b). ILVR (Choi et al., 2021) adopts diffusion models (Ho et al., 2020) for image super-resolution and image translation. All these works require training of a score model irrelevant to the actual objective task, and are thus flexible. Nevertheless, all the prior works have focused on applications where the condition also stays in the image domain, which makes the problem easier to solve. Our method adds further flexibility by showing that conditions can be applied in measurement domains that are not necessarily in the same image domain.

We are aware of one prior work which used denoising score matching for MRI reconstruction (Ramzi et al., 2020). The authors of (Ramzi et al., 2020) use amortized residual de-noising autoencoder (AR-DAE) score matching loss (Lim et al., 2020) to train the score function, then uses annealed Hamiltonian

MC (Neal et al., 2011) to perform reconstruction from measurement. However, (Ramzi et al., 2020) reported that their method falls behind supervised learning approaches by a large margin, especially when considering single samples. Moreover, the training methodology in (Ramzi et al., 2020) targets separate channel complex-valued data, which limits their application. Our method, on the other hand, beats neural networks trained with supervision, and requires only the magnitude images for training. It is also notable that the proposed method is applicable to PI.

After the submission of this paper, we found two independent works that are closely related to the proposed work. Jalal et al. (2021) proposes to use score-based generative models to train a score function that is similar to ours, and sample by taking annealed Langevin dynamics (ALD) (Song and Ermon, 2019) together with the gradient information with respect to the data fidelity term by assuming Gaussian measurement noise. As was also shown in our work, Jalal et al. (2021) illustrates the robustness of using a score-based generative model for reconstruction over different sub-sampling patterns and diverse anatomy. The largest difference between us and Jalal et al. (2021) comes from the fact that our method only requires DICOM images for training the score function. This is in stark contrast with Jalal et al. (2021), as they require fully-sampled  $k$ -space data for training the score function. Furthermore, our work is based on the continuous version of score matching (Song et al., 2021b), whereas the work of Jalal et al. (2021) is based on a discrete version (Song and Ermon, 2020). It is also worth mentioning that we use an advanced sampler (PC), and a more efficient network architecture, which was shown to improve the performance of generative modeling by a large margin (Song et al., 2021b). Finally, Jalal et al. (2021) introduces *annealing* the data fidelity gradient terms which require specifying the variance schedule per each noise scale. Our method does not have additional hyper-parameters, and is hence easier to implement.

Song et al. (2022) is perhaps the most related to our work, in that the authors also propose to use VE-SDE of (Song et al., 2021b), and they use the same network architecture together with the PC sampler, as in our work. The crucial difference of our work from Song et al. (2022) is that we derive solvers that are capable of reconstructing complex-valued data, and also multi-coil data. Song et al. (2022) only focused on solving simulated reconstructions from real-valued images, which limits the practicality. Moreover, the data fidelity imposing step differs slightly from our work.

## 6.3. Energy based models

Energy-based models (EBMs) are non-normalized probabilistic models, which have the advantage of circumventing the need to compute the normalizing constant (i.e. the partition function) (Song and Kingma, 2021). Our work relies on denoising score matching to estimate the score, which also belongs to the category of EBMs. There are, of course, other ways to train EBMs, and one of the most widely known methods other than score matching, is contrastive divergence (CD) (Carreira-Perpinan and Hinton, 2005).

Concurrent to our work, an MR acceleration algorithm utilizing CD was proposed (Guan et al., 2021). This work grounds

<sup>8</sup>short for Neural Function Evaluations

their method on an EBM, trained with persistent contrastive divergence (PCD), which is a variant of CD using sample buffers (Du and Mordatch, 2019). Specifically, a parameterized energy function  $E_\theta(\cdot)$  is trained such that it takes low values when the input to the function is likely to be in the data distribution, and high values when this is not the case. Once the energy function is trained, one can generate unconditional samples via MCMC, or apply data consistency projection in between the MCMC update steps to sample from a conditional distribution.

This procedure is in fact similar to our method, with the trained functions forming the relationship of  $\nabla_x \log p_\theta(\mathbf{x}) = -\nabla_x E_\theta(\mathbf{x})$ . However, we note two key differences of the proposed method from (Guan et al., 2021). First, in order to train  $E_\theta$  with CD, one has to produce negative samples with MCMC at every iteration of training, which is non-trivial and costly. Several heuristics need to be applied to make the algorithm work. In contrast, the training of our score function is much more straightforward and robust, breaking down to an explicit form of the loss function, as formulated in Eq. (17). Second, our method only requires magnitude (DICOM) images for training, whereas (Guan et al., 2021) requires raw data.

#### 6.4. Generative models for inverse problems

Before the recent surge of diffusion models, there have been several approaches to use generative models, especially generative adversarial networks (GAN) (Goodfellow et al., 2014) as priors for solving inverse problems (Marinescu et al., 2021; Asim et al., 2020). These methods offer improved flexibility over supervised models which are trained for specific problems. Rather, one can leverage a well-trained generator  $G_\varphi$ , which was trained without the knowledge of the forward physics. In essence, in order to solve inverse problems with GAN prior, one would typically optimize for the following:

$$\min_z \|AG_\varphi(z) - \mathbf{y}\|, \quad (20)$$

where  $A$  and  $\mathbf{y}$  are as defined in eq. (7), and  $z$  denotes the latent (noise) vector. This corresponds to finding the correct latent vector  $z$  which minimizes the data fidelity. One can also try

$$\min_\varphi \|AG_\varphi(z) - \mathbf{y}\|, \quad (21)$$

which corresponds to tweaking the model parameters  $\varphi$  so that the generator *adapts* to the forward physics of the problem. Once the optimization via (20) or (21) is complete, a single forward pass through the generator  $G_\varphi$  is sufficient for reconstruction. Unfortunately, there are several problems with these methods.

First, both the problems (20) and (21) are nontrivial to solve, and require several heuristics such as using a sophisticated loss function (Marinescu et al., 2021; Asim et al., 2020). Considering that GANs themselves are also notoriously hard to train, methods that rely on GAN priors are relatively hard to reproduce. Second, the final reconstruction step involves a *single* forward pass through  $G_\varphi$ . For highly ill-posed problems, it could be particularly hard to generate high-quality samples from this single pass, which could be a reason why it is hard to achieve a reconstruction with both high quality and data fidelity.

The proposed method, which proposes to use diffusion models instead of GANs, solves both problems. Diffusion models have a relatively well-defined loss and are thus easier to train. Moreover, the sampling procedure can easily be done with the most basic method of solving inverse problems. Further, with diffusion models, one can achieve fine-grained control since we iteratively refine our reconstruction.

#### 6.5. Broader Impact

The proposed method can be readily applied to other problems in computational imaging, with well trained score function and the right modifications to the inference procedure. A single score function has already shown broad applicability: SR (Choi et al., 2021; Saharia et al., 2021), image reconstruction, and others. This could potentially shift the current paradigm of deep learning in biomedical imaging. For example, one can train a single score function for the imaging modality, and use it as a universal problem solver, given enough capacity.

#### 6.6. Limitations

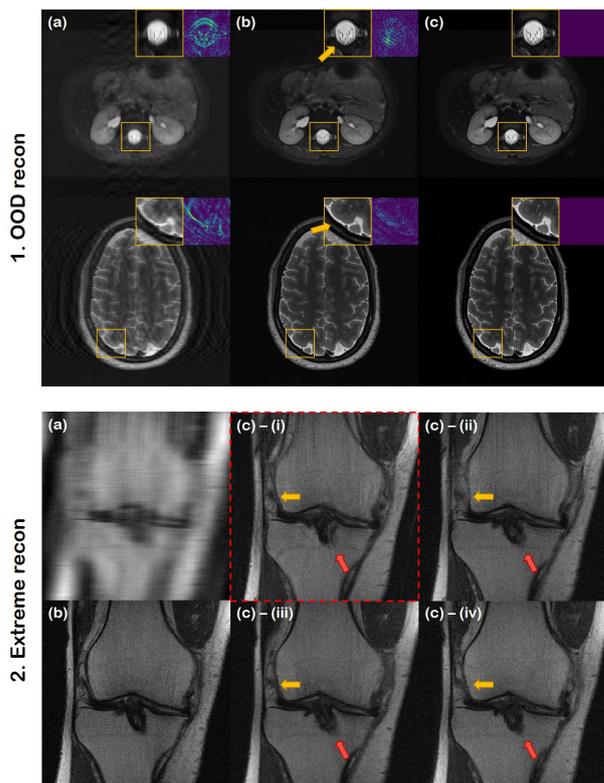


Fig. 11: Limitations of the proposed method. 1. OOD reconstruction with 1D under-sampling pattern. (a) zero-filled, (b) proposed reconstruction, and (c) ground truth. Yellow arrows indicate the artifacts. 2. Extreme reconstruction (Uniform 1D  $\times 15$  acc.) (a) zero-filled, (b) ground truth, and (c) - (i vi) posterior samples. Red dotted box indicates the posterior sample of the worst quality. Yellow and red arrows mark regions with high alteration.

For completeness, here we list two limitations of the current work. First, when we attempt to reconstruct OOD data with 1D under-sampling pattern, we sometimes observe mild aliasing-like artifacts in local edges. As illustrated in Fig. 11 (1. OOD recon), the artifact is not significant. However, care should be

taken when extending the proposed framework to OOD data, as the robustness will be compromised. We note that such an artifact is not observed in 2D sampling patterns, as was shown in Fig. 9.

Second, when pursuing extreme-condition reconstruction, as in Fig. 11 (2. Extreme recon), we occasionally acquire results that are unsatisfactory (e.g. sample marked with the red dotted line). Moreover, we observe that the detailed structure has high variance within the posterior samples, due to the high ill-posedness. Hence, care should be taken when pushing the accelerating factor to very high values, by, for example, sampling multiple reconstructions and considering the uncertainty as was discussed in section 5.5.

## 7. Conclusion

In conclusion, we propose a novel score-based reconstruction method for accelerated MRI. We train the gradient of the log data distribution with continuous-time denoising score matching using the magnitude data. Using the learned score as the prior, one can sample from the conditional distribution given the measurement by simply applying data consistency projection at every step. Our method produces reconstructions of high accuracy, whether it be single-coil, or multi-coil cases. Compared to prior arts, we show the superiority of our method both in terms of quality, and practicality.

We believe that our method opens up a new generation of methods for inverse problems in imaging. Direct application of our method to other venues to test the generality is an interesting direction of future research. Other than that, there still remain unanswered questions, for example, reducing the reconstruction speed gap between our method and feedforward neural network approaches. We expect that many interesting questions and answers will be actively discussed in the near future.

## Acknowledgments

This work was supported by the National Research Foundation (NRF) of Korea grant NRF-2020R1A2B5B03001980.

## Appendix

### Details of network architecture

In this section, we elaborate on the `ncsnpp` network architecture used for constructing the score function  $s_\theta$ , especially focusing on the way the network incorporates time conditions. Overall illustration of the architecture is depicted in Fig. 12. The network takes in as input,  $x_t$ , and  $t$ . Note that since we are solving for a continuous SDE, the additional time conditioning variable is important. For the time condition  $t$ , it passes through the time conditioning module, as shown in Fig. 12 (c), utilizing the Fourier features introduced in Tancik et al. (2020). In the figure, this is given as the Gaussian Fourier Projection (GFP) module, which can be mathematically written with:

$$\text{GFP}(a) := [\sin(2\pi w_1 a), \sin(2\pi w_2 a), \dots, \sin(2\pi w_d a), \\ \cos(2\pi w_1 a), \cos(2\pi w_2 a), \dots, \cos(2\pi w_d a)],$$

where the weights  $w_1, w_2, \dots, w_d$  are non-trainable random initialized network parameters that are initialized with `torch.randn(d) * scale`, and we set `scale` to be 16. Subsequently, the embedding vector passes through a few layers of MLP to match the size of the channel dimension of input image features. Finally, the temporal feature vectors are broadcasted across the  $H, W$  dimensions to be added to the image features via residual blocks at each level.

## Statistical analysis of pathological detection

Bland-Altman plot of the pathologic detection task is shown in Fig. 13. For individual data points, we resort to the mAP50 score for each class. With the plots, we can check for the agreement of each reconstruction method with the fully sampled data. Consistent with the observations that were made in the main text, we see that the most agreeing method is the proposed method, which shows the least variance in the difference. DuDoRNet and UNet both have higher variance, and TV is radically different from the fully sampled data.

## Additional results

Additional reconstruction results of different anatomies are presented in Fig. 14 and Fig. 15.

## References

- Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks, in: International conference on machine learning, PMLR. pp. 214–223.
- Asim, M., Daniels, M., Leong, O., Ahmed, A., Hand, P., 2020. Invertible generative models for inverse problems: mitigating representation error and dataset bias, in: International Conference on Machine Learning, PMLR. pp. 399–409.
- Bauschke, H.H., Combettes, P.L., et al., 2011. Convex analysis and monotone operator theory in Hilbert spaces. volume 408. Springer.
- Block, K.T., Uecker, M., Frahm, J., 2007. Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 57, 1086–1098.
- Boyd, S., Parikh, N., Chu, E., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc.
- Brock, A., Donahue, J., Simonyan, K., 2019. Large scale GAN training for high fidelity natural image synthesis, in: International Conference on Learning Representations (ICLR).
- Carreira-Perpinan, M.A., Hinton, G., 2005. On contrastive divergence learning, in: International workshop on artificial intelligence and statistics, PMLR. pp. 33–40.
- Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S., 2021. ILVR: Conditioning method for denoising diffusion probabilistic models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).
- Chung, H., Sim, B., Ye, J.C., 2021. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. arXiv preprint arXiv:2112.05146.
- Deshmane, A., Gulani, V., Griswold, M.A., Seiberlich, N., 2012. Parallel MR imaging. *Journal of Magnetic Resonance Imaging* 36, 55–72.
- Dhariwal, P., Nichol, A., 2021. Diffusion models beat GANs on image synthesis. arXiv preprint arXiv:2105.05233.
- Donoho, D.L., 2006. Compressed sensing. *IEEE Transactions on information theory* 52, 1289–1306.

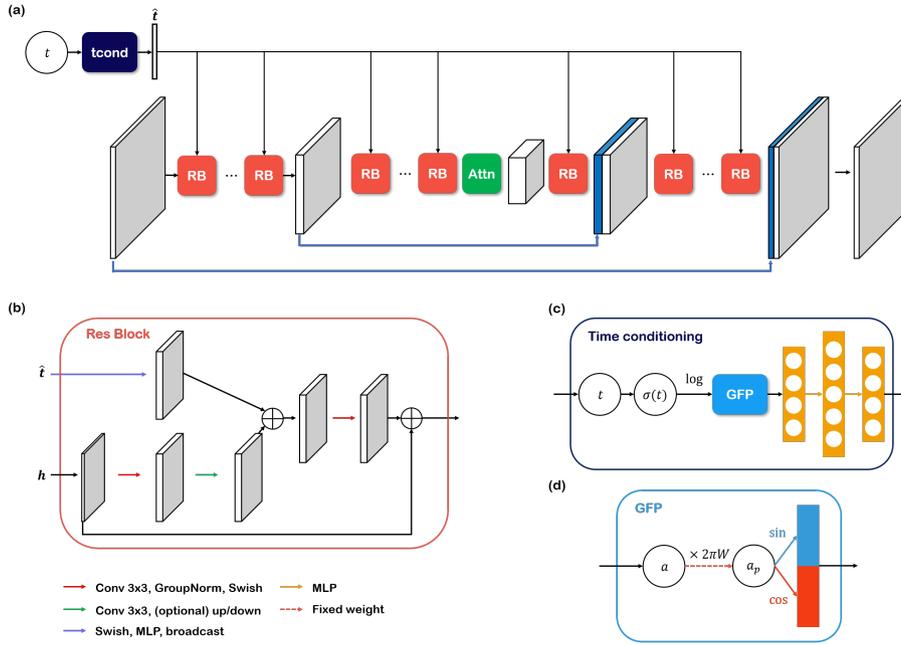


Fig. 12: Illustration of the  $ncsnpp$  architecture used to model  $s_\theta$ . (a) Overall U-Net like structure, consisting of BigGAN residual blocks (RB), and attention blocks. Additional  $t$  is incorporated with time conditioning block, with the output  $\hat{t}$  affecting all the residual blocks. (b) Basic building block of  $ncsnpp$ . (c) Time conditioning module, which utilizes Gaussian Fourier Projection (GFP) module. (d) Detailed illustration of the GFP module.

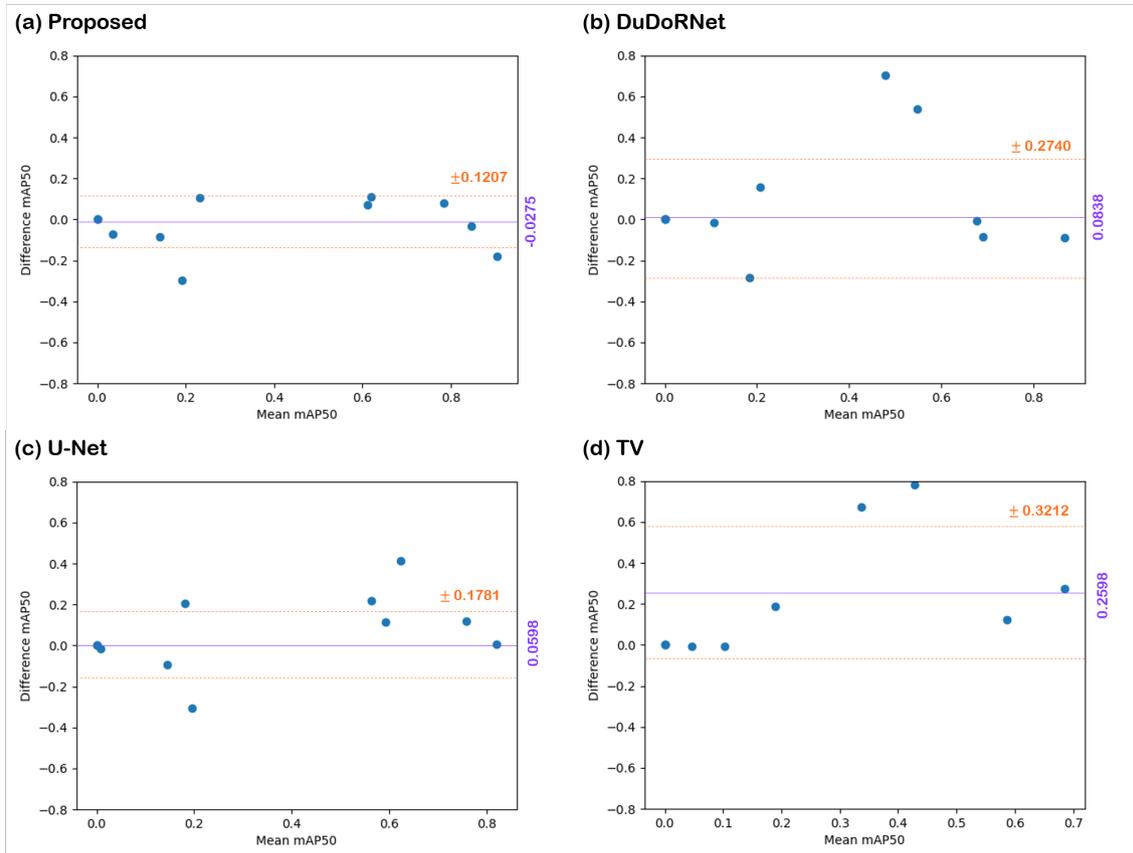


Fig. 13: Bland-Altman plot of mAP score of pathologic detection with different methods. (a) Proposed method, (b) DuDoRNet (Zhou and Zhou, 2020), (c) U-Net, (d) TV.  $x$ -axis represent the mean value, and the  $y$ -axis denote the difference from the result using fully-sampled data.

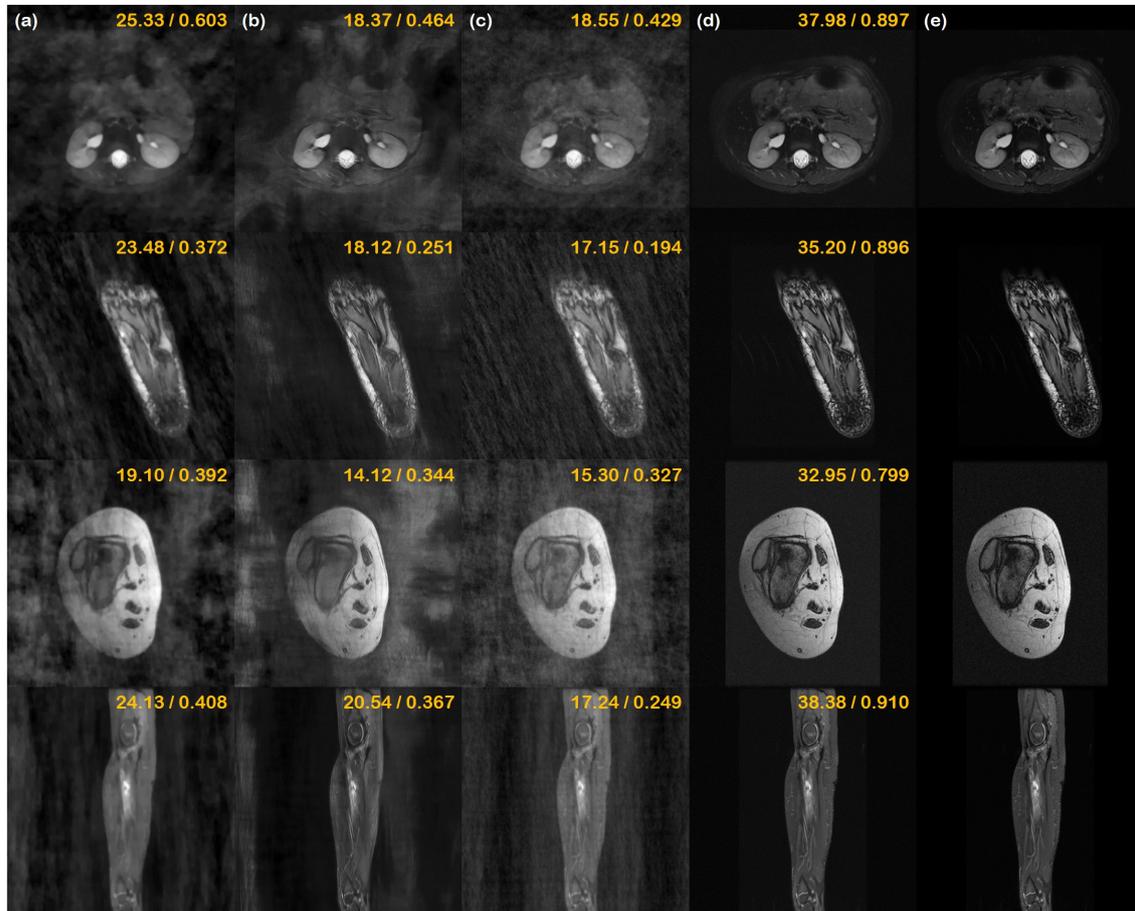


Fig. 14: Reconstruction results of various anatomic structure/contrast. (a) TV, (b) U-Net, (c) DuDoRNet, (d) proposed method, (e) ground truth. All images are open-source data collected from sources indicated in Section 4.7. VD Poisson disk  $\times 8$  under-sampling was performed. All data were collected from Flynn et al.. 1<sup>st</sup> row: cardiac MRI, 2<sup>nd</sup> row: foot axial scan, 3<sup>rd</sup> row: lower extremity axial scan, 4<sup>th</sup> row: LT bone.

- tional variation for compressed sensing in mri. *Magnetic Resonance Imaging* 77, 186–193.
- Fan, C., Wu, C., Li, G., Ma, J., 2017. Projections onto convex sets super-resolution reconstruction based on point spread function estimation of low-resolution remote sensing images. *Sensors* 17, 362.
- Flynn, A., Ong, F., Nahum, G., Cheng, J., Lustig, M., Toh, J., Virtue, P., Amin, S., Vasawala, S., Tariq, U., . mridata.org. <http://mridata.org>. Accessed: 2021-02-18.
- Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR. pp. 1050–1059.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems* 27.
- Griswold, M.A., Jakob, P.M., Heidemann, R.M., Nittka, M., Jellus, V., Wang, J., Kiefer, B., Haase, A., 2002. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 47, 1202–1210.
- Guan, Y., Tu, Z., Wang, S., Liu, Q., Wang, Y., Liang, D., 2021. MRI Reconstruction Using Deep Energy-Based Model. *arXiv preprint arXiv:2109.03237*.
- Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models, in: *Advances in Neural Information Processing Systems*, pp. 6840–6851.
- Hyvärinen, A., Dayan, P., 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research* 6.
- Jalal, A., Arvinte, M., Daras, G., Price, E., Dimakis, A.G., Tamir, J., 2021. Robust compressed sensing MRI with deep generative priors. *Advances in Neural Information Processing Systems* 34.
- Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
- Karras, T., Laine, S., Aila, T., 2019. A style-based generator architecture for generative adversarial networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410.
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T., 2020. Analyzing and improving the image quality of styleGAN, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119.
- Kendall, A., Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision?, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc.
- Kim, D., Shin, S., Song, K., Kang, W., Moon, I.C., 2021. Score matching model for unbounded data score. *arXiv preprint arXiv:2106.05527*.
- Kingma, D.P., Salimans, T., Poole, B., Ho, J., 2021. Variational diffusion models, in: *Advances in Neural Information Processing Systems*.
- Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B., 2021. DiffWave: A versatile diffusion model for audio synthesis, in: *9th International Conference on Learning Representations, ICLR*.
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690.
- Lim, J.H., Courville, A., Pal, C., Huang, C.W., 2020. AR-DAE: Towards Unbiased Neural Entropy Gradient Estimation, in: *International Conference on Machine Learning, PMLR*. pp. 6061–6071.
- Loeve, M., 2017. *Probability theory*. Courier Dover Publications.
- Lustig, M., Donoho, D., Pauly, J.M., 2007. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in*

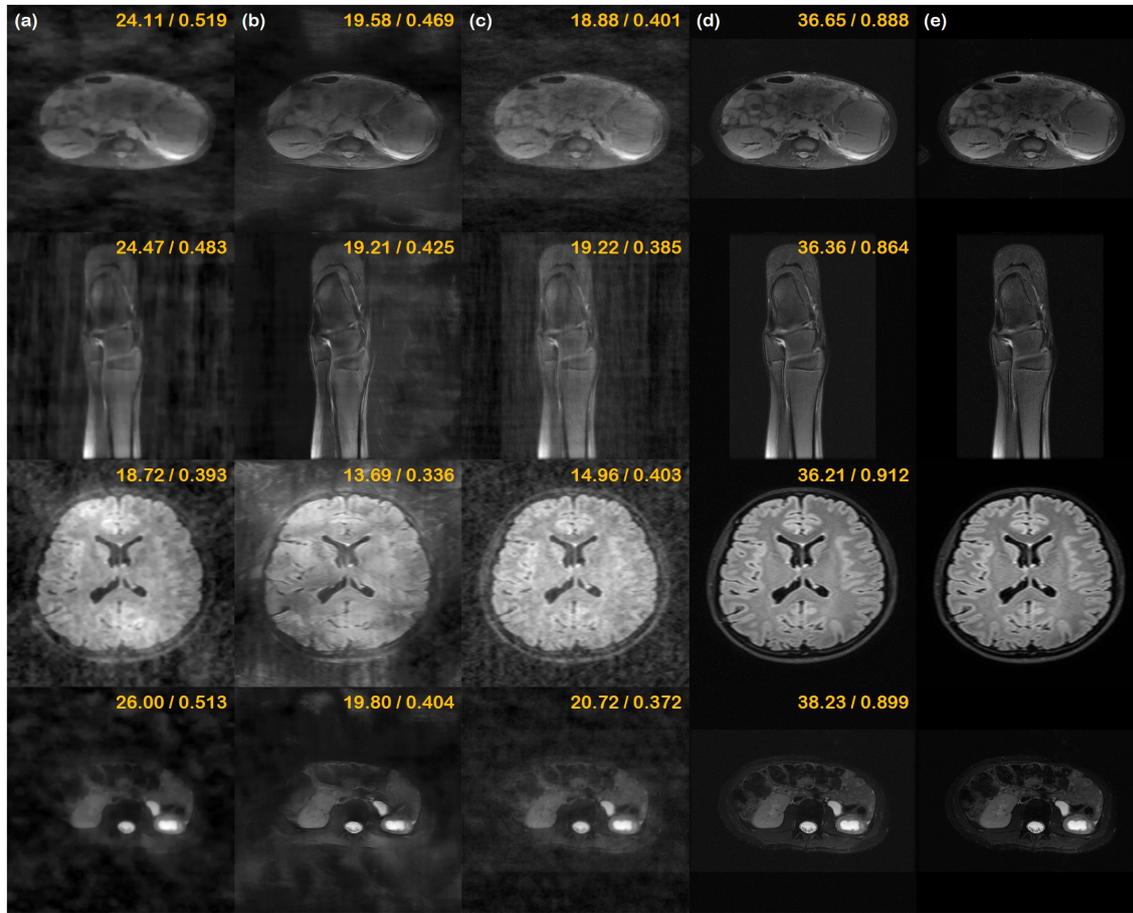


Fig. 15: Reconstruction results of various anatomic structure/contrast. (a) TV, (b) U-Net, (c) DuDoRNet, (d) proposed method, (e) ground truth. All images are open-source data collected from sources indicated in Section 4.7. VD Poisson disk  $\times 8$  under-sampling was performed. The data in 1,2,4<sup>th</sup> row were collected from Flynn et al., and the data in 3<sup>rd</sup> row was collected from MASSIVE. 1<sup>st</sup> row: abdomen, 2<sup>nd</sup> row: ankle (foot), 3<sup>rd</sup> row: axial brain, 4<sup>th</sup> row: upper torso abdomen.

- Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine 58, 1182–1195.
- Marinescu, R.V., Moyer, D., Golland, P., 2021. Bayesian image reconstruction using deep generative models, in: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.
- Mason, A., Rioux, J., Clarke, S.E., Costa, A., Schmidt, M., Keough, V., Huynh, T., Beyea, S., 2019. Comparison of objective image quality metrics to expert radiologists' scoring of diagnostic quality of MR images. *IEEE transactions on medical imaging* 39, 1064–1072.
- Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S., 2021. SDEdit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.
- Neal, R.M., et al., 2011. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo* 2, 2.
- Nichol, A.Q., Dhariwal, P., 2021. Improved denoising diffusion probabilistic models, in: *International Conference on Machine Learning*, PMLR. pp. 8162–8171.
- Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., Ermon, S., 2020. Permutation invariant graph generation via score-based generative modeling, in: *International Conference on Artificial Intelligence and Statistics*, PMLR. pp. 4474–4484.
- Oh, G., Sim, B., Chung, H., Sunwoo, L., Ye, J.C., 2020. Unpaired deep learning for accelerated MRI using optimal transport driven cycleGAN. *IEEE Transactions on Computational Imaging* 6, 1285–1296.
- Parisi, G., 1981. Correlation functions and computer simulations. *Nuclear Physics B* 180, 378–384.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32, 8026–8037.
- Pruessmann, K.P., Weiger, M., Scheidegger, M.B., Boesiger, P., 1999. SENSE: sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 42, 952–962.
- Ramzi, Z., Remy, B., Lanusse, F., Starck, J.L., Ciuciu, P., 2020. Denoising score-matching for uncertainty quantification in inverse problems, in: *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*.
- Roemer, P.B., Edelstein, W.A., Hayes, C.E., Souza, S.P., Mueller, O.M., 1990. The NMR phased array. *Magnetic resonance in medicine* 16, 192–225.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: *International Conference on Medical image computing and computer-assisted intervention*, Springer. pp. 234–241.
- Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M., 2021. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*.
- Samsonov, A.A., Kholmovski, E.G., Parker, D.L., Johnson, C.R., 2004. POC-SENSE: POCS-based reconstruction for sensitivity encoded magnetic resonance imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 52, 1397–1406.
- Schoonjans, F., Zalata, A., Depuydt, C., Comhaire, F., 1995. Medcalc: a new computer program for medical statistics. *Computer methods and programs in biomedicine* 48, 257–262.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics, in: *International Conference on Machine Learning*, PMLR. pp. 2256–2265.
- Song, Y., Durkan, C., Murray, I., Ermon, S., 2021a. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing*

- cessing Systems 34.
- Song, Y., Ermon, S., 2019. Generative modeling by estimating gradients of the data distribution, in: *Advances in Neural Information Processing Systems*.
- Song, Y., Ermon, S., 2020. Improved techniques for training score-based generative models, in: *Advances in Neural Information Processing Systems*, pp. 12438–12448.
- Song, Y., Garg, S., Shi, J., Ermon, S., 2020. Sliced score matching: A scalable approach to density and score estimation, in: *Uncertainty in Artificial Intelligence*, PMLR. pp. 574–584.
- Song, Y., Kingma, D.P., 2021. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*.
- Song, Y., Shen, L., Xing, L., Ermon, S., 2022. Solving inverse problems in medical imaging with score-based generative models, in: *International Conference on Learning Representations*.
- Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B., 2021b. Score-based generative modeling through stochastic differential equations, in: *9th International Conference on Learning Representations*, ICLR.
- Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C.L., Yakubova, N., Knoll, F., Johnson, P., 2020. End-to-end variational networks for accelerated MRI reconstruction, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer. pp. 64–73.
- Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* 33, 7537–7547.
- Tang, Z., Deng, M., Xiao, C., Yu, J., 2011. Projection onto convex sets super-resolution image reconstruction based on wavelet bi-cubic interpolation, in: *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, IEEE. pp. 351–354.
- Vincent, P., 2011. A connection between score matching and denoising autoencoders. *Neural computation* 23, 1661–1674.
- Wang, S., Su, Z., Ying, L., Peng, X., Zhu, S., Liang, F., Feng, D., Liang, D., 2016. Accelerating magnetic resonance imaging via deep learning, in: *2016 IEEE 13th international symposium on biomedical imaging (ISBI)*, IEEE. pp. 514–517.
- Yaman, B., Hosseini, S.A.H., Moeller, S., Ellermann, J., Uğurbil, K., Akçakaya, M., 2020. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magnetic resonance in medicine* 84, 3172–3191.
- Zbontar, J., Knoll, F., Sriram, A., Murrell, T., Huang, Z., Muckley, M.J., Defazio, A., Stern, R., Johnson, P., Bruno, M., et al., 2018. *fastMRI: An open dataset and benchmarks for accelerated MRI*. *arXiv preprint arXiv:1811.08839*.
- Zhang, R., 2019. Making convolutional networks shift-invariant again, in: *International conference on machine learning*, PMLR. pp. 7324–7334.
- Zhao, R., Yaman, B., Zhang, Y., Stewart, R., Dixon, A., Knoll, F., Huang, Z., Lui, Y.W., Hansen, M.S., Lungren, M.P., 2021. *fastMRI+: Clinical Pathology Annotations for Knee and Brain Fully Sampled Multi-Coil MRI Data*. *arXiv preprint arXiv:2109.03812*.
- Zhou, B., Zhou, S.K., 2020. 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*, pp. 4273–4282.