DEQ-SOLVER: A POSTERIOR SCORE-BASED DEEP EQUILIBRIUM ODE SOLVER FOR INVERSE PROBLEMS IN MEDICAL IMAGING

Anonymous authors
Paper under double-blind review

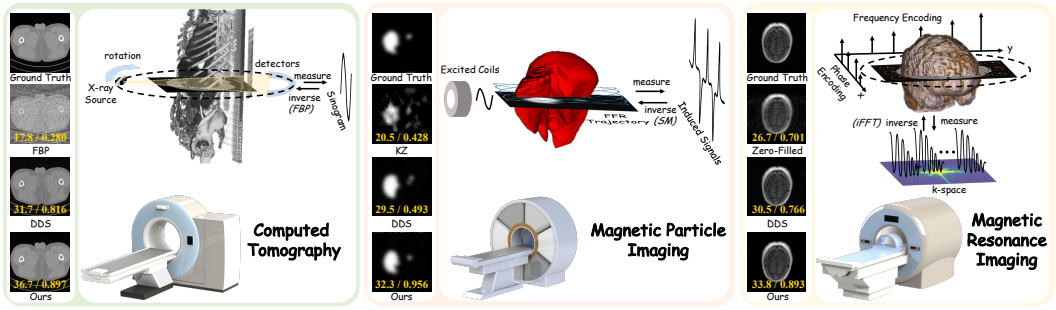

Figure 1: In general, medical image reconstruction can be formulated as an inverse problem. Our DEQ-Solver enables fast, high-quality reconstruction from incomplete or corrupted measurements across different imaging modalities.

ABSTRACT

Score-based Generative Models have emerged as the dominant generative paradigm. Owing to their impressive generative performance and high interpretability, they have also proven to be a compelling generative prior for solving inverse problems in medical imaging. Nevertheless, incorporating the posterior distribution derived from the physical imaging model into the sampling process continues to be a nontrivial issue. Furthermore, their prohibitive sampling cost remains a major challenge. To overcome the above limitations, we propose a posterior score-based Deep Equilibrium Solver (DEQ-Solver). It constructs a measurement-conditioned neural vector field that directly estimates the posterior score. Based on the implicit backward Euler formulation, the resulting reverse-time dynamics can be cast as a global fixed-point problem. By solving for the global equilibrium, we can jointly drive all temporal states along the entire reverse-time trajectory. Therefore, our method improves the efficiency of solving medical imaging inverse problems without compromising reconstruction quality. Across various medical imaging modalities, DEQ-Solver achieves superior reconstruction quality with substantially faster sampling, including a 7.3× speedup over other accelerated samplers in Magnetic Particle Imaging.

1 INTRODUCTION

In medical imaging systems described by linear forward models, solving an inverse problem typically involves reconstructing an unknown structural or functional image from measurements (Wang et al., 2020). The corresponding forward imaging model can be formulated as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the observed measurements; $\boldsymbol{A} : \mathbb{R}^{hw} \to \mathbb{R}^m$ represents the known physical forward operator; $\boldsymbol{x} \in \mathbb{R}^{h\times w}$ denotes the desired image. The measurement noise $\boldsymbol{n} \in \mathbb{R}^m$ is commonly modeled as white Gaussian noise, i.e., $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. The objective is to estimate the unknown medical image $\boldsymbol{x}$ from the observation $\boldsymbol{y}$, such as sinogram data in Computed Tomography (CT), k-space measurements in Magnetic Resonance Imaging (MRI), or induced signals

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

in Magnetic Particle Imaging (MPI). This is achieved using the corresponding discrete transform operator $\boldsymbol{A}$, e.g., the Radon transform for CT (Ramachandran & Lakshminarayanan, 1971), the Fourier transform for MRI (Kumar et al., 1975), and the Magnetization operator for MPI (Gleich & Weizenecker, 2005). Based on these modality-specific physical models, analytical reconstruction methods have been well established. **Nevertheless, when the measurements $\boldsymbol{y}$ are incomplete or corrupted**, such as in sparse-view CT, noisy MPI, or undersampled MRI, as shown in Fig. 1, the ill-posedness of the inverse problem becomes more severe. Consequently, **the performance of analytical reconstruction methods often deteriorates**. Model-based iterative reconstruction methods are commonly employed to solve these inverse problems (Ravishankar et al., 2019). A key prerequisite for the success of these methods is the design of a sophisticated handcrafted data prior, commonly known as a regularization term. This regularization term integrates domain knowledge and helps constrain the solution space by enforcing desirable image properties.

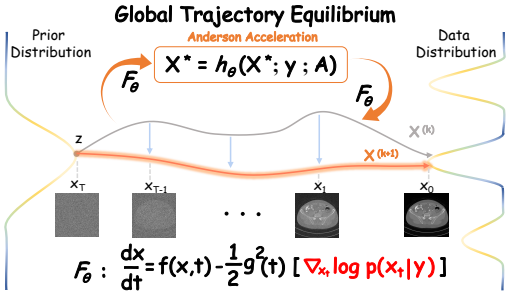

Figure 2: Overall pipeline of our DEQ-Solver. Our method models a measurement-conditioned neural vector field and simultaneously drives the entire reverse-time trajectory toward a global equilibrium.

Recently, Score-based Generative Models (SGMs) (Song et al., 2021b; Lim et al., 2023) have shown remarkable generative capabilities across a wide range of computer vision tasks (Esser et al., 2024). The core idea of SGMs is to progressively transform a simple prior distribution into the target data distribution using the learned gradient of the log-marginal density, i.e., the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ (Esser et al., 2024). In the incomplete or corrupted measurement settings described above, numerous studies (Chung et al., 2023b; Song et al., 2022; Chung et al., 2024) have demonstrated that this diffusion strategy can provide a more expressive data-driven generative prior than conventional handcrafted regularizers, such as total variation and Tikhonov regularization. This can lead to more accurate and reliable reconstructions. The main line of these works (Shi et al., 2026) is to approximate the likelihood score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$, since the posterior distribution $p_t(\boldsymbol{x}_t|\boldsymbol{y})$ is generally intractable. However, the observation $\boldsymbol{y}$ depends directly only on the clean image $\boldsymbol{x}_0$ through the physical models, rather than on the intermediate states $\boldsymbol{x}_t$. Therefore, approximating the likelihood score can introduce errors. Meanwhile, the high sampling cost remains a major bottleneck for the practical application of these methods. These approximation errors can become more pronounced during accelerated sampling. Therefore, **how to solve medical imaging inverse problems faster and better remains an open challenge**.

To this end, we propose DEQ-Solver, a novel score-based deep equilibrium ODE solver for posterior sampling in medical inverse problems, as shown in Fig. 2. **Our DEQ-Solver directly estimates the posterior score through the measurement-conditioned denoising mean.** This posterior formulation gives rise to a measurement-conditioned neural vector field that governs the reverse-time dynamics. We discretize these dynamics using the backward Euler scheme, which naturally yields a fixed-point formulation. To avoid sequentially integrating the reverse-time dynamics defined by this neural vector field, we cast the entire reverse-time trajectory as a global fixed-point problem within the deep equilibrium framework. **All temporal states can then be jointly driven toward the global equilibrium using an accelerated fixed-point solver.** Therefore, our method can achieve efficient and high-fidelity medical image reconstruction. We conduct extensive experiments across a wide range of medical imaging modalities. These results demonstrate the superiority of our DEQ-Solver in terms of sampling speed and high-fidelity reconstruction, as shown in Fig. 3.

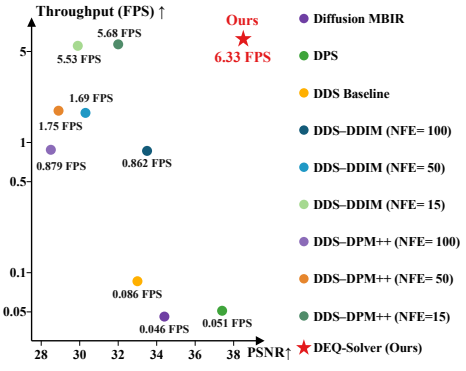

Figure 3: FPS (log scale) vs. PSNR for Magnetic Particle Imaging with SNR=10 dB. Our proposed DEQ-Solver consistently lies in the upper-right region, which indicates a favorable trade-off between reconstruction quality and sampling speed.

2 BACKGROUND

2.1 SCORE-BASED MODELS

According to the Fokker–Planck equation, the evolution of a complex distribution $p_{\text{data}}(\boldsymbol{x})$ towards a tractable prior distribution $p_{\boldsymbol{z}}$ can be modeled as an Itô Stochastic Differential Equation (SDE). As a result, Song & Ermon (2020) unified denoising diffusion probabilistic models (Ho et al., 2020) with Score-based Generative Models (SGMs). Specifically, the SDEs can be formulated as:

$$\mathrm{d}\boldsymbol{x} = f(\boldsymbol{x}, t)\mathrm{d}t + g(t)\mathrm{d}w \tag{2}$$

where w denotes a standard Wiener process and its increment satisfies $dw \sim \mathcal{N}(0, \Delta t)$. $f(\boldsymbol{x}, t)$ and $g(t)$ are the drift and diffusion coefficients, respectively. Given that the prior distribution $p_{\boldsymbol{z}}$ is set as a Gaussian distribution $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, the SDEs' forward process can be implemented by gradually adding noise $\boldsymbol{z}_t$ to the data $\boldsymbol{x}_0$ (time $t \in [0, 1]$). Following Anderson's theorem (Anderson, 1982), the corresponding reverse-time SDEs are given as follows:

$$\mathrm{d}\boldsymbol{x} = [f(\boldsymbol{x}, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)]\mathrm{d}\bar{t} + g(t)\mathrm{d}\bar{w} \tag{3}$$

where $\bar{t}$ decreases from 1 to 0, and $\bar{w} \sim \mathcal{N}(0, \Delta\bar{t})$. $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ is the gradient of the log-marginal distribution, called the score function. It is well known that estimating the score function is intractable due to the high dimensionality of the data distribution. To overcome this challenge, it is common practice to approximate the score function using a trained neural network s_θ (Ho et al., 2020; Song et al., 2021a). During training, the forward SDE is simulated, and at every time step, the model estimates the noise that has been added to the data. Subsequently, the trained score function is applied to the reverse-time SDE, combined with numerical SDE solver techniques, to generate images step-by-step. Furthermore, the reverse-time SDE can also be presented in an Ordinary Differential Equation (ODE) form (Song et al., 2021b):

$$\mathrm{d}\boldsymbol{x} = [f(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)]\mathrm{d}t \tag{4}$$

Thus, various ODE-based sampling methods have been developed (Song et al., 2021a).

2.2 DEEP EQUILIBRIUM MODELS

The success of modern deep learning methods largely hinges on the use of sufficiently deep neural networks. However, as network depth increases, the number of parameters and the computational burden grow exponentially. A deep neural network can generally be formulated as a sequence of non-linear transformations, as shown below:

$$\boldsymbol{x}_{i+1} = h_{\boldsymbol{\theta}}^{[i]}(\boldsymbol{x}_i; \boldsymbol{x}_0) \quad i = 0, 1, 2, \cdots, L-1 \tag{5}$$

where i indicates the layer index of the network $f_{\boldsymbol{\theta}}$, and $\boldsymbol{x}_0$ is the input to the first layer. Here, $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ denote the input and output of the i-th layer, respectively. It is worth noting that as the depth increases, the contribution of subsequent layers gradually diminishes. The network output eventually converges to an equilibrium:

$$\lim_{i\to\infty} \boldsymbol{x}_{i+1} = \lim_{i\to\infty} h_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}_i; \boldsymbol{x}_0) \equiv \boldsymbol{x}^* \tag{6}$$

Hence, the deep equilibrium model (DEQ) formulates this equilibrium state as a fixed-point problem (Bai et al., 2019):

$$\boldsymbol{x}^* = h_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}^*; \boldsymbol{x}_0) \tag{7}$$

Furthermore, the equilibrium condition can be reformulated as:

$$g(\boldsymbol{x}^*; \boldsymbol{x}_0; \boldsymbol{\theta}) = \boldsymbol{x}^* - h_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}^*; \boldsymbol{x}_0) = 0 \tag{8}$$

To find the root of the function $g(\boldsymbol{x}; \boldsymbol{x}_0; \boldsymbol{\theta})$, well-established optimization algorithms such as Newton's method, Anderson acceleration, and Broyden's method can be employed. In this paper, the Anderson acceleration method is utilized. Pokle et al. (2022) and Cao et al. (2024) reformulate explicit diffusion sampling chains as joint fixed-point systems for parallel evaluation. Our formulation instead derives the trajectory equations from a backward Euler discretization of the posterior vector field, which introduces an implicit dependence on the state being solved at each time step. Compared with previous methods that also support trajectory-level parallelism, our formulation further exploits the numerical stability of backward Euler to reduce the number of temporal states under coarse discretization.

3 METHOD

3.1 POSTERIOR SAMPLING IN SCORE-BASED MODELS

From a Bayesian perspective, the inverse problem in Eq. (1) can be addressed through posterior sampling from $p(\boldsymbol{x}|\boldsymbol{y})$. Since the measurement $\boldsymbol{y}$ is incomplete, the inverse problem is generally ill-posed, and multiple plausible images may be consistent with the same measurement. Therefore, rather than seeking a single deterministic solution, posterior sampling aims to characterize the distribution of plausible reconstructions conditioned on the measurement $\boldsymbol{y}$. Score-based generative models provide a powerful strategy for such posterior sampling by exploiting the learned score as an implicit image prior. Specifically, posterior sampling can be performed by replacing the unconditional score in the reverse-time SDE with the posterior score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$:

$$\mathrm{d}\boldsymbol{x} = [f(\boldsymbol{x}, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})]\mathrm{d}t + g(t)\mathrm{d}\bar{w} \tag{9}$$

However, the posterior score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$ is nontrivial to estimate. Therefore, several previous studies (Song et al., 2022; 2023; Chung et al., 2023a; 2024) have sought to bypass the direct estimation of the posterior score. According to Bayes' rule, the posterior score can be decomposed into the prior score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ and the likelihood score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$ as:

$$\mathrm{d}\boldsymbol{x} = \{f(\boldsymbol{x}, t) - g^2(t)[\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)]\}\mathrm{d}t + g(t)d\bar{w} \tag{10}$$

Under a linear Gaussian forward imaging model, the likelihood associated with the forward imaging model can be expressed as $p(\boldsymbol{y}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{A}\boldsymbol{x}_0, \sigma^2\boldsymbol{I})$. In other words, the measurement $\boldsymbol{y}$ is directly related to the clean image $\boldsymbol{x}_0$ through the physical forward model, rather than to the intermediate diffusion state $\boldsymbol{x}_t$. Therefore, estimating the likelihood score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$ often requires heuristic hyperparameters to compensate for the approximation error (Jalal et al., 2021; Chung et al., 2022; Song et al., 2023). Meanwhile, a large number of sampling steps is required to mitigate the accumulated approximation error, further increasing the reconstruction time (Shi et al., 2026).

In contrast, we provide a straightforward approach to directly estimating the posterior score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$, rather than approximating the likelihood score. By marginalizing the latent clean image $\boldsymbol{x}_0$ and exploiting the conditional independence $\boldsymbol{x}_t \perp \boldsymbol{y} \mid \boldsymbol{x}_0$ (see Appendix A.1 for detailed derivations), the posterior score can be derived as follows.

Proposition 1. *Specifically, for the case of VP-SDE, the perturbation kernel is given by* $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (\sqrt{1-\bar{\alpha}_t})^2\boldsymbol{I})$. *Then the posterior score can be expressed as:*

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{1-\bar{\alpha}_t} \tag{11}$$

This shows that the posterior score can be determined directly once the measurement-conditioned denoising mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]$ is available. Therefore, we focus on estimating this conditional denoising mean in this paper. In unconditional diffusion sampling, the denoising mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ can be uniquely determined by Tweedie's formula (Kim & Ye, 2021): $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t))$. Similarly, under a linear Gaussian imaging model, the denoising mean conditioned on the measurement $\boldsymbol{y}$ also admits a closed-form expression.

Proposition 2. *Given the denoising distribution* $\boldsymbol{x}_0|\boldsymbol{x}_t \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_t)$ *and the imaging forward model* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{n}$, *with* $\boldsymbol{n} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, *the conditional denoising mean is given by:*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \hat{\boldsymbol{x}}_0 + \boldsymbol{\Sigma}_t\boldsymbol{A}^H(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \tag{12}$$

Notably, the denoising distribution $p(\boldsymbol{x}_0|\boldsymbol{x}_t)$, approximated via Gaussian moment matching (Lemma 3 in Appendix), is characterized by its first and second moments, $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ and $\mathrm{Cov}(\boldsymbol{x}_0|\boldsymbol{x}_t)$. The conditional mean $\hat{\boldsymbol{x}}_0 = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ can be readily obtained as discussed above, while the covariance $\boldsymbol{\Sigma}_t = \mathrm{Cov}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ characterizes the residual uncertainty of the clean image given the diffusion state at time step t. Conventional approaches typically adopt an isotropic approximation $\boldsymbol{\Sigma}_t = \sigma_t\boldsymbol{I}$ with an empirically specified σ_t (Song et al., 2023), which assigns the same uncertainty level to all spatial locations. A more general spatially varying covariance can provide location-dependent weighting when incorporating measurement information in principle.

However, explicitly constructing such a covariance is computationally impractical. For an image of size $h \times w$, a full covariance matrix $\boldsymbol{\Sigma}_t \in \mathbb{R}^{hw \times hw}$ requires quadratic storage and computation with

respect to the number of image pixels. Moreover, the matrix inverse $(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \sigma^2\boldsymbol{I})^{-1}$ is defined in the measurement domain and has size $m \times m$. In medical imaging, m can be large, making direct formation and inversion of this matrix computationally prohibitive. To enable efficient computation, we reformulate Eq. (12) as an equivalent linear system and solve it using the Conjugate Gradient (CG) method.

Remark 1 (Conjugate Gradient Solver)**.** *Using the Woodbury matrix identity (Golub & Van Loan, 2013; Lu & Huang, 2020), the conditional denoising mean in Eq. (12) can be equivalently reformulated as:*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \mathrm{CG}\left(\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A}, \sigma^2\boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + \boldsymbol{A}^H\boldsymbol{y}, \hat{\boldsymbol{x}}_0, k_{\mathrm{CG}}\right), \tag{13}$$

where k_{CG} denotes the number of iterations of CG.

This shows that measurement conditioning admits a precision-weighted form involving the denoising estimate and the physical forward model. Motivated by this structure, we parameterize the spatial weighting directly with a learnable image-sized map, thereby avoiding the explicit construction and storage of the high-dimensional covariance matrix $\boldsymbol{\Sigma}_t$. Specifically, the above formulation leads us to train a network s_θ to predict a measurement-aware denoising estimate $\hat{\boldsymbol{x}}_0$ and the corresponding spatial weighting map $\boldsymbol{W}_t$:

$$s_{\boldsymbol{\theta}}([\boldsymbol{x}_t; \boldsymbol{A}^H\boldsymbol{y}], t) = [\hat{\boldsymbol{x}}_0; \boldsymbol{W}_t]. \tag{14}$$

The predicted weighting map is constrained to be strictly positive as

$$\boldsymbol{W}_t = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\frac{1}{1 + \exp(-\boldsymbol{W}_t)}, \tag{15}$$

where $0 < \lambda_{\min} < \lambda_{\max}$.

Following the precision-weighted formulation in Remark 1, we obtain a measurement-conditioned denoising estimate by substituting the predicted $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{W}_t$ into:

$$\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}} = \mathrm{CG}\left(\mathcal{W}_t, \boldsymbol{W}_t \odot \hat{\boldsymbol{x}}_0 + \boldsymbol{A}^H\boldsymbol{y}, \hat{\boldsymbol{x}}_0, k_{\mathrm{CG}}\right), \tag{16}$$

where

$$\mathcal{W}_t(\boldsymbol{x}) = \boldsymbol{W}_t \odot \boldsymbol{x} + \boldsymbol{A}^H\boldsymbol{A}\boldsymbol{x}. \tag{17}$$

We use the resulting $\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}}$ as an approximation to the measurement-conditioned denoising mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]$. According to Proposition 1, the resulting conditional denoising estimate can be used to estimate the posterior score, thereby avoiding an explicit approximation of the likelihood score. However, its numerical realization still requires a large number of discrete sampling steps along the diffusion trajectory.

3.2 Deep Equilibrium formulation of Posterior Sampling

To overcome this sequential sampling bottleneck, we depart from the conventional view of reverse diffusion as an explicitly unrolled sequence of time-stepping updates. Our key observation is that all states along the reverse-time trajectory are coupled through a shared posterior vector field:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{t}} &= f(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \\ &= f(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)(-\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}}}{1 - \bar{\alpha}_t})\end{aligned} \tag{18}$$

As usual, this ordinary differential equation can be solved using numerical methods, such as the Euler method (Butcher, 2016) and Runge–Kutta methods (Dormand & Prince, 1980). These solvers rely on sequential time discretization, which results in a trade-off between computational cost and numerical accuracy. Instead of explicitly using a large number of discrete time steps, we characterize the underlying continuous-time dynamics with the following neural vector field (Chen et al., 2018):

$$F_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\ y; \boldsymbol{A}; t) = f(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)(-\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}}}{1 - \bar{\alpha}_t}) \tag{19}$$

The learnable parameters of $F_{\boldsymbol{\theta}}$ are introduced solely through $s_{\boldsymbol{\theta}}([\boldsymbol{x}_t; \boldsymbol{A}^H\boldsymbol{y}], t)$, while the remaining terms are determined by the predefined forward diffusion process. Thus, during the reverse-time

process, for an arbitrary time-step size Δt, the state $\boldsymbol{x}_t$ can be obtained from $\boldsymbol{x}_{t+\Delta t}$ by integrating the ODE over $[t, t+\Delta t]$:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t+\Delta t} - \int_t^{t+\Delta t} F_{\boldsymbol{\theta}}(\boldsymbol{x}_\tau; \boldsymbol{y}; \boldsymbol{A}; \tau)\, \mathrm{d}\tau \tag{20}$$

Although the underlying dynamics evolve continuously over the interval $[t, t+\Delta t]$, numerical integration approximates this continuous trajectory using discrete time steps. Consequently, the choice of Δt introduces an inherent trade-off between numerical accuracy and computational efficiency. To better preserve stability under relatively large time steps Δt, we adopt the backward Euler discretization (Wanner & Hairer, 1996), which yields the implicit update:

$$\boldsymbol{x}_t^* = \boldsymbol{x}_{t+\Delta t} - F_{\boldsymbol{\theta}}(\boldsymbol{x}_t^*; \boldsymbol{y}; \boldsymbol{A}; t) \cdot \Delta t \tag{21}$$

where $\boldsymbol{x}_t^*$ denotes the equilibrium point associated with the hidden state $\boldsymbol{x}_t$. Accordingly, the training objective is to learn $F_{\boldsymbol{\theta}}$ such that the predicted state $\boldsymbol{x}_t^*$ satisfies the backward Euler equilibrium condition. Next, by recursively applying the above implicit update from T to t, we observe that the reverse-time ODE trajectory can be expressed as:

$$\boldsymbol{x}_t^* = \boldsymbol{x}_T^* - \sum_{i=0}^{\frac{T-t}{\Delta t}-1} F_\theta(\boldsymbol{x}_{t+i\cdot\Delta t}^*; \boldsymbol{y}; \boldsymbol{A}; t + i \cdot \Delta t) \cdot \Delta t \tag{22}$$

This defines a fully coupled reverse-time process, where the state $\boldsymbol{x}_t$ depends on all previous states $\boldsymbol{x}_{t+\Delta t:T}$, rather than only on the immediately preceding state $\boldsymbol{x}_{t+\Delta t}$. Let $h_\theta(\cdot)$ denote the mapping that performs the above operation (Eq. (22)) for a single state. The entire reverse trajectory can then be written as:

$$\begin{bmatrix} \boldsymbol{x}_T^* = h_{\boldsymbol{\theta}}(\boldsymbol{x}_T^*; \boldsymbol{y}; \boldsymbol{A}) \\ \boldsymbol{x}_{T-\Delta_t}^* = h_{\boldsymbol{\theta}}(\boldsymbol{x}_{T-\Delta t:T}^*; \boldsymbol{y}; \boldsymbol{A}) \\ \vdots \\ \boldsymbol{x}_0^* = h_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T}^*; \boldsymbol{y}; \boldsymbol{A}) \end{bmatrix} \tag{23}$$

Although the reverse-time states are temporally coupled, this dependency does not require their neural evaluations to be executed sequentially. By stacking all intermediate states into a single hidden variable $\boldsymbol{X} = [\boldsymbol{x}_T; \boldsymbol{x}_{T-\Delta t}; \cdots; \boldsymbol{x}_0]$, we reformulate the lower-triangular temporal dependencies as a global fixed-point system:

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_T \\ \boldsymbol{x}_{T-\Delta_t} \\ \vdots \\ \boldsymbol{x}_0 \end{bmatrix} = h_{\boldsymbol{\theta}} \left(\begin{bmatrix} \boldsymbol{x}_T \\ \boldsymbol{x}_{T-\Delta_t} \\ \vdots \\ \boldsymbol{x}_0 \end{bmatrix}; \boldsymbol{y}; \boldsymbol{A} \right)}_{\boldsymbol{X}^* = h_{\boldsymbol{\theta}}(\boldsymbol{X}^*; \boldsymbol{y}; \boldsymbol{A})} \tag{24}$$

The global fixed-point formulation is algebraically equivalent to the sequential backward Euler updates, and the posterior vector fields at different time points can therefore be evaluated in parallel without additional discretization error. Thus, all trajectory states can be solved simultaneously using an accelerated fixed-point algorithm until the global equilibrium is reached. Moreover, the numerical stability of backward Euler permits relatively large time steps Δt. As a result, fewer temporal states are required, and the memory cost is reduced.

4 Experiments

4.1 Experimental Setup

Dataset We conduct experiments on three imaging modalities: MPI Synthetic Dataset (system matrix-based MPI) (Chen et al., 2026), AAPM (sparse-view CT) (Kang et al., 2017), and fastMRI (compressed-sensing MRI) (Chung & Ye, 2022). Details of the datasets are provided in Appendix A.3. To simulate corrupted measurements $\boldsymbol{y}$ in MPI, we add Gaussian noise at three signal-to-noise ratio (SNR) levels: 10 dB, 20 dB, and 30 dB. For sparse-view CT, we use a 2D fan-beam geometry with 368 detector bins, a source-to-axis distance of 5.95, and an axis-to-detector distance

Table 1: Quantitative comparison of MPI reconstruction (mean ± SD). The best results are highlighted in bold, and the second-best results are underlined. The same notation is used in all subsequent tables.

Method		NFEs	SNR = 10 dB		SNR = 20 dB		SNR = 30 dB	
			PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
KZ		–	22.4±4.32	0.406±0.03	28.4±4.22	0.612±0.10	30.3±4.42	0.705±0.11
CG		–	15.8±4.34	0.157±0.03	25.8±3.88	0.389±0.07	30.2±3.45	0.596±0.08
ADMM		–	16.0±4.32	0.151±0.03	25.7±3.77	0.365±0.07	29.9±3.46	0.578±0.07
DiffusionMBIR		2000	<u>31.0</u>±5.58	0.792±0.12	34.4±6.55	0.815±0.13	<u>42.2</u>±4.41	<u>0.986</u>±0.02
DPS		1000	32.4±5.46	<u>0.956</u>±0.04	<u>37.4</u>±5.57	<u>0.977</u>±0.03	41.2±3.53	0.922±0.12
DDS	Baseline	1000	31.4±4.48	0.940±0.05	33.0±4.53	0.965±0.03	41.5±5.06	0.941±0.05
	DDIM	100	31.5±4.49	0.877±0.07	33.5±4.61	0.939±0.05	40.3±3.89	0.964±0.02
	DDIM	15	27.6±2.92	0.548±0.08	29.9±2.45	0.634±0.09	30.2±2.45	0.649±0.10
	DPM++	100	30.8±4.52	0.896±0.07	28.5±4.77	0.703±0.08	41.1±4.80	0.933±0.05
	DPM++	15	21.6±4.63	0.375±0.07	32.0±4.72	0.697±0.10	38.0±4.34	0.912±0.05
DEQ-MPI		10	31.3±3.80	0.880±0.05	35.6±3.44	0.940±0.04	40.7±3.74	0.978±0.02
DEQ-Solver (ours)		10	**32.9**±5.16	**0.958**±0.03	**38.5**±4.97	**0.980**±0.02	**42.7**±5.22	**0.988**±0.01

of 4.906. Sparse-view measurements are generated using n_v projection views uniformly sampled over $[0, 2\pi)$. Furthermore, Poisson noise with an intensity of 1×10^6 is added to the sinograms to further evaluate the robustness of our method beyond Gaussian noise. For compressed-sensing MRI, we retrospectively undersample the acquired k-space data using an $8\times$1D Gaussian sampling mask, with the central 4% of k-space lines retained as the fully sampled ACS region. Since the measurements are not synthesized from the reference images, they preserve the noise present in real acquisitions.

Training We adopt the VP-SDE with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20$ (Song et al., 2021b), and use $N = 10$ logarithmically spaced time points: $t_i = \exp\left(\log T + \frac{i}{N-1}(\log\epsilon - \log T)\right), i = 0, \ldots, N-1$, where $\epsilon = 10^{-3}$. For the other comparison methods, we use $N = 1000$ time steps during training. For the architecture of the network $s_{\boldsymbol{\theta}}$, we follow the same configuration as in (Chung et al., 2024). We train the model using AdamW with a weight decay of 5×10^{-2} for a total of 200 epochs. The initial learning rate is set to 2×10^{-4} and decayed to 2×10^{-7} using a cosine schedule. Gradient clipping is applied with a maximum norm of 5. We use Anderson acceleration with a tolerance of 10^{-6}, a damping factor of $\tau = 0.4$, and a maximum of 10 iterations (k_{deq}) during both training and inference. The maximum number of CG iterations in Eq. (16) is set to $k_{\text{CG}} = 5$. The hyperparameters $\lambda_{\min}$ and $\lambda_{\max}$ are set to 1×10^{-3} and 1×10^{1}, respectively. The CT and MRI experiments are conducted on 4 NVIDIA A100 80 GB GPUs, with a batch size of 8 per GPU for CT and 4 for MRI. The MPI experiments are conducted on 2 NVIDIA RTX 4090 24 GB GPUs, with a batch size of 16 per GPU.

Evaluation Metrics We use the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) to evaluate reconstruction quality quantitatively. For fastMRI, we additionally report the normalized Root-Mean-Square Error (nRMSE) to assess the accuracy of reconstructed image intensities. To evaluate sampling efficiency, we report the Number of Function Evaluations (NFEs) and Frames Per Second (FPS).

4.2 Results

For a comprehensive evaluation, we conduct comparison experiments with several state-of-the-art methods. These include advanced SGM-based approaches (Chung et al., 2023a;b; 2024; Lu et al., 2025) as well as deep unrolling methods (Xia et al., 2024; Zhang et al., 2025; Güngör et al., 2023). Moreover, extensive ablation studies are reported and discussed in Appendix A.5.

System matrix-based MPI The reconstruction results are reported in Tab. 1. In addition to deep learning-based methods, we also evaluate three classical iterative methods: Kaczmarz (KZ), Conjugate Gradient (CG), and the Alternating Direction Method of Multipliers (ADMM). However, their performance degrades when the measurements are corrupted. SGM-based methods generally outperform the deep unrolling method (DEQ-MPI (Güngör et al., 2023)), while our method further improves the reconstruction performance. In particular, the SNR $= 10\,\text{dB}$ setting mimics low-concentration MPI reconstruction, an important practical scenario in MPI. Our method still achieves

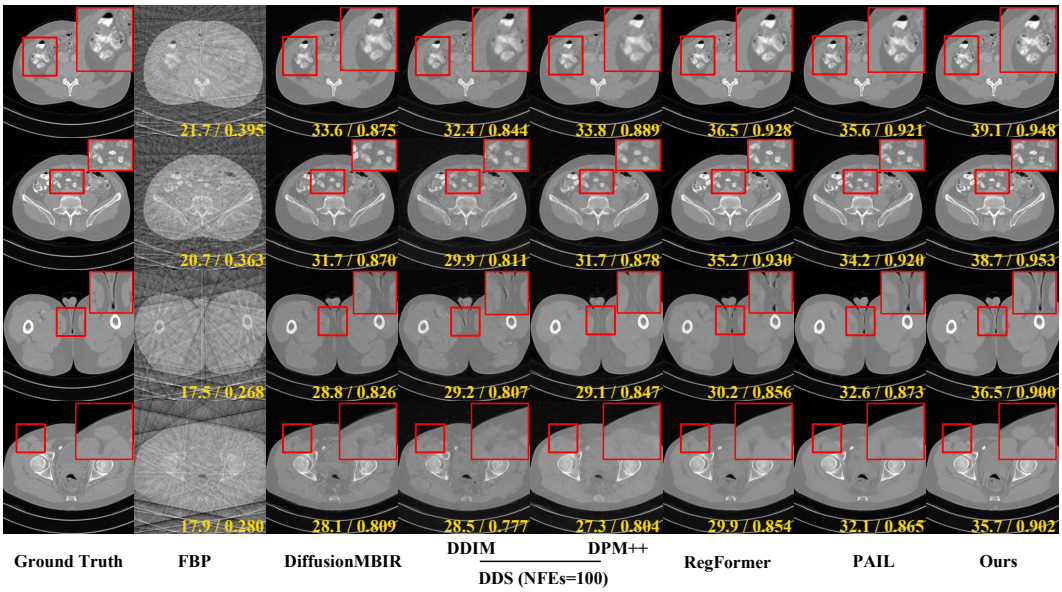

Figure 4: Sparse-View CT reconstruction results. Zoom in for better visualization. The first two rows show the results of 32-view CT reconstruction, while the last two rows correspond to 16-view CT reconstruction. The corresponding PSNR/SSIM values are provided for each reconstruction.

Table 2: Quantitative comparison of sparse-view CT reconstruction at a noise level of 1×10^6.

Method		NFEs	16-view		32-view	
			PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
FBP		–	16.6±0.66	0.252±0.01	20.9±0.67	0.389±0.01
DiffusionMBIR		2000	27.3±1.05	0.790±0.02	33.2±1.36	0.870±0.01
DPS		1000	27.0±1.72	0.768±0.03	29.8±1.59	0.810±0.02
DDS	Baseline	1000	28.5±1.46	0.780±0.02	32.8±1.26	0.854±0.01
	DDIM	100	27.3±1.37	0.747±0.03	31.8±1.42	0.831±0.02
	DDIM	50	26.6±1.25	0.732±0.03	30.9±1.42	0.816±0.02
	DPM++	100	27.0±1.50	0.794±0.03	33.2±1.51	0.883±0.01
	DPM++	50	26.3±1.50	0.775±0.03	32.4±1.48	0.868±0.01
RegFormer		18	31.4±0.93	0.856±0.01	36.0±1.84	0.918±0.01
PAIL		10	29.1±1.07	0.834±0.02	35.2±0.88	0.911±0.01
DEQ-Solver (ours)		10	**34.7**±0.95	**0.895**±0.01	**38.4**±0.47	**0.937**±0.01

the best performance under this condition. This indicates the potential of our method to improve imaging sensitivity. Meanwhile, as shown in Fig. 3, our method is approximately $\mathbf{120}\times$ faster than DPS (Chung et al., 2023a) (**6.33** FPS vs. **0.051** FPS). This shows a substantial improvement in sampling speed. Furthermore, an ablation study on the number of DDS (Chung et al., 2024) sampling steps shows that reducing the number of sampling steps can achieve a throughput comparable to ours, but with a considerable degradation in reconstruction performance. More qualitative results are displayed in Appendix A.6.

Sparse-view CT We conduct reconstruction experiments on two sparse-view CT settings ($n_v = 16$ and $n_v = 32$). Meanwhile, we add Poisson noise to the sinograms, since Poisson noise is non-negligible in practical CT acquisitions in addition to Gaussian noise. This setting further evaluates the robustness of our method under more realistic noise conditions. The experimental results are exhibited in Tab. 2. While deep unrolling methods (RegFormer (Xia et al., 2024), PAIL (Zhang et al., 2025)) can achieve competitive performance, their deterministic end-to-end predictions tend to regress toward the conditional mean, as shown in Fig. 4. This often produces over-smoothed reconstructions and suppresses the native noise texture of CT images. Moreover, SGM-based methods do not perform as well as reported in previous studies. One possible reason is that many prior evaluations retrospectively synthesize sinograms under idealized settings and neglect measurement noise, which is also consistent with the observations reported in the CT reconstruction benchmark DM4CT (Shi et al., 2026). In contrast, our method shows stronger robustness to noisy measurements. This robustness may stem from the spatially

adaptive weighting learned jointly with the denoising estimate, allowing the data-consistency update to modulate the relative contribution of the learned prior and the physical measurement at each spatial location. In addition, the FPS vs. PSNR results are plotted in Fig. 7 (see Appendix A.5 for details). Meanwhile, additional qualitative results are provided in Appendix A.6.

Compressed Sensing MRI In this experiment, we move further toward a real-world setting by directly using the acquired multi-coil k-space measurements and their corresponding MVUE images. The quantitative comparisons with other methods are provided in Tab. 3. The DDS method (Chung et al., 2024), accelerated by DDIM, achieves better performance than DiffusionMBIR (Chung et al., 2023b) and DPS (Chung et al., 2023a) with fewer NFEs. We further implement DPM++ (Lu et al., 2025) within DDS, since both methods operate on the predicted $\hat{\boldsymbol{x}}_0$. This further improves the reconstruction performance over the DDIM-based variant when using 100 NFEs.

Table 3: Quantitative comparison of compressed-sensing knee MRI reconstruction at $8\times$ undersampling.

Method		NFEs	$8\times$ undersampling	
			PSNR ↑	SSIM ↑
Zero-Filled		–	25.8±2.77	0.674±0.07
DiffusionMBIR		2000	30.3±3.09	0.741±0.13
DPS		1000	30.4±2.80	0.649±0.13
DDS	Baseline	1000	29.9±2.61	0.725±0.12
	DDIM	100	30.5±2.65	0.758±0.11
	DDIM	50	30.4±2.48	0.752±0.09
	DPM++	100	<u>30.7</u>±2.50	<u>0.766</u>±0.09
	DPM++	50	29.7±2.26	0.677±0.07
DEQ-Solver (ours)		10	**32.9**±2.82	**0.790**±0.10

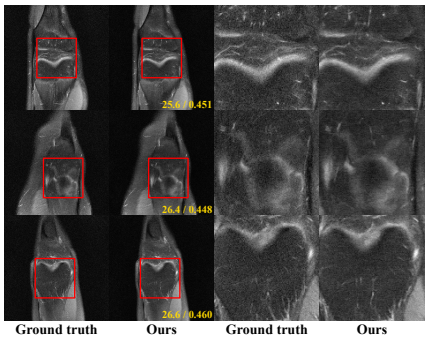

Figure 5: Failure cases of our method on fastMRI Knee under $8\times$ undersampling. The two rightmost columns show zoomed-in views of the regions indicated by the red boxes.

Our DEQ-Solver achieves better reconstruction quality while requiring only 10 NFEs and providing nearly a $1.3\times$ speedup, as illustrated in Fig. 7. This translates into meaningful computational savings in large-scale medical imaging applications. For example, reconstructing the entire test set saves approximately 180 minutes compared with the accelerated baseline (DDS with NFEs $= 100$). The reconstruction results are visualized in Fig. 13 and Fig. 14 (Appendix A.6). In addition to presenting high-quality reconstruction results, we also analyze representative failure cases, as shown in Fig. 5, to discuss the limitations of our method and provide a more comprehensive understanding of its performance. We select the three cases with the lowest quantitative metrics as representative failure cases. Although their overall structures are well restored, fine textures and weak edges remain degraded under high-noise conditions. This may partly result from the final denoising step from $\boldsymbol{x}_\epsilon$ to $\boldsymbol{x}_0$, where some high-frequency textures and weak structural details may be treated as noise and consequently smoothed out. Moreover, their relatively low scores may also partly arise from inaccurate intensity rescaling, since the scaling factor is estimated from the 99th percentile of the Zero-Filled reconstruction and may deviate from the true intensity range under high-noise conditions. Furthermore, we conduct out-of-distribution generalization experiments on fastMRI Brain. Please refer to Appendix A.5 for details.

5 Conclusion

In this paper, we investigate the application of SGM-based methods to inverse problems in medical imaging. We observe that existing methods remain subject to a trade-off between reconstruction quality and sampling efficiency. To this end, we propose a novel posterior score-based deep equilibrium ODE solver. It directly estimates the posterior score and formulates the entire reverse-time trajectory as a global fixed-point problem. Extensive quantitative and qualitative experiments across various medical imaging modalities demonstrate the superior sampling efficiency and reconstruction quality of our method. In particular, we further extend our evaluation to Magnetic Particle Imaging, an emerging medical imaging modality. We believe that this work can provide new insights into efficient posterior sampling for the medical imaging community.

LLM USAGE

We affirm that this work was conducted in accordance with the ICLR guidelines on the use of large language models (LLMs). In this work, we used generative AI tools in three ways. First, they were used to assist with writing and language polishing, including the correction of spelling, grammar, and formatting errors. Second, they were used for literature retrieval and discovery, including identifying potentially relevant prior work; all cited references were independently verified by the authors against the original sources. Third, they were used to assist with research ideation and implementation. All AI-assisted suggestions were carefully reviewed by the authors, and only those that were independently verified were retained. The authors take full responsibility for the final content of this work, including the text, claims, and any artifacts produced with the assistance of generative AI.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Riccardo Barbano, Alexander Denker, Hyungjin Chung, Tae Hoon Roh, Simon Arridge, Peter Maass, Bangti Jin, and Jong Chul Ye. Steerable conditional diffusion for out-of-distribution adaptation in medical image reconstruction. *IEEE Transactions on Medical Imaging*, 44(5):2093–2104, 2025.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Jiezhang Cao, Yue Shi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. Deep equilibrium diffusion restoration with parallel sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2824–2834, June 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

Ziwei Chen, Wenxuan Zou, Gen Shi, Jian'an Ye, Ning Sun, Zhongwei Bian, Yu An, and Jie Tian. Deliver-mpi: Diffusion embedded-physics model for magnetic particle imaging reconstruction under strong liver background. In *2026 IEEE 23rd International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2026. doi: 10.1109/ISBI61048.2026.11515511.

Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated mri. *Medical image analysis*, 80:102479, 2022.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=nJJjv0JDJju.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Hyungjin Chung, Dohoon Ryu, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Solving 3d inverse problems using pre-trained 2d diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22542–22551, 2023b. doi: 10.1109/CVPR52729.2023.02159.

Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DsEhqQtfAG.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

Bernhard Gleich and Jürgen Weizenecker. Tomographic imaging using the nonlinear response of magnetic particles. *Nature*, 435(7046):1214–1217, 2005.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

Alper Güngör, Baris Askin, Damla Alptekin Soydan, Can Bariş Top, Emine Ulku Saritas, and Tolga Cukur. Deq-mpi: A deep equilibrium reconstruction with learned consistency for magnetic particle imaging. *IEEE Transactions on Medical Imaging*, 43(1):321–334, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in neural information processing systems*, 34:14938–14954, 2021.

Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ZqEUs3sTRU0.

Anil Kumar, Dieter Welti, and Richard R Ernst. Nmr fourier zeugmatography. *Journal of Magnetic Resonance (1969)*, 18(1):69–83, 1975.

Wenxuan Li, Xinze Zhou, Qi Chen, Tianyu Lin, Pedro R. A. S. Bassi, Xiaoxi Chen, Chen Ye, Zheren Zhu, Kai Ding, Heng Li, Kang Wang, Yang Yang, Yucheng Tang, Daguang Xu, Alan Yuille, and Zongwei Zhou. PanTS: The pancreatic tumor segmentation dataset. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=0BCUXg40r7.

Sungbin Lim, Eunbi Yoon, Taehyun Byun, Taewon Kang, Seungwoo Kim, Kyungjae Lee, and Sungjoon Choi. Score-based generative modeling through stochastic evolution equations in hilbert spaces. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GrElRvXnEj.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

You Lu and Bert Huang. Woodbury transformations for deep generative flows. *Advances in Neural Information Processing Systems*, 33:5801–5811, 2020.

Jun Ma and Bo Wang. *Fast, Low-resource, and Accurate Organ and Pan-cancer Segmentation in Abdomen CT: MICCAI Challenge, FLARE 2023, Held in Conjunction with MICCAI 2023, Vancouver, BC, Canada, October 8, 2023, Proceedings*, volume 14544. Springer Nature, 2024.

Ashwini Pokle, Zhengyang Geng, and J Zico Kolter. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems*, 35:37975–37990, 2022.

GN Ramachandran and AV Lakshminarayanan. Three-dimensional reconstruction from radiographs and electron micrographs: application of convolutions instead of fourier transforms. *Proceedings of the National Academy of Sciences*, 68(9):2236–2240, 1971.

Saiprasad Ravishankar, Jong Chul Ye, and Jeffrey A Fessler. Image reconstruction: From sparsity to data-adaptive methods and machine learning. *Proceedings of the IEEE*, 108(1):86–109, 2019.

Jiayang Shi, Daniel Pelt, and Joost Batenburg. DM4CT: Benchmarking diffusion models for computed tomography reconstruction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YE5scJekg5.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vaRCHVj0uGI.

Ge Wang, Jong Chul Ye, and Bruno De Man. Deep learning for tomographic image reconstruction. *Nature machine intelligence*, 2(12):737–748, 2020.

Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*, volume 375. Springer Berlin Heidelberg New York, 1996.

Wenjun Xia, Ziyuan Yang, Zexin Lu, Zhongxian Wang, and Yi Zhang. Regformer: A local–nonlocal regularization-based model for sparse-view ct reconstruction. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 8(2):184–194, 2024. doi: 10.1109/TRPMS.2023.3281148.

Jianjia Zhang, Zirong Li, Jiayi Pan, Shaoyu Wang, and Weiwen Wu. Trustworthy limited data ct reconstruction using progressive artifact image learning. *IEEE Transactions on Image Processing*, 34:1163–1178, 2025. doi: 10.1109/TIP.2025.3534559.

Wenxuan Zou, Gen Shi, Siao Lei, Guanghui Li, Guangxing Zhou, Yang Jing, Jie He, Zhenchao Tang, Yu An, and Jie Tian. U-n2c: A dual memory-guided disentanglement framework for unsupervised system matrix denoising in magnetic particle imaging. *IEEE Transactions on Image Processing*, 34:2867–2882, 2025. doi: 10.1109/TIP.2025.3564845.

A Appendix

A.1 Proofs

Lemma 1. *The gradient of the posterior $p_t(\boldsymbol{x}_t|\boldsymbol{y})$ can be computed by marginalizing over the latent variable $\boldsymbol{x}_0$:*

$$\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) p_t(\boldsymbol{x}_0|\boldsymbol{y}) \mathrm{d}\boldsymbol{x}_0 \tag{25}$$

Proof. By marginalizing the latent variable $\boldsymbol{x}_0$, the conditional distribution of $\boldsymbol{x}_t$ given $\boldsymbol{y}$ can be written as

$$p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})\mathrm{d}\boldsymbol{x}_0. \tag{26}$$

Taking the gradient with respect to x_t on both sides gives:

$$\begin{aligned}\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{y}) &= \nabla_{\boldsymbol{x}_t} \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})\mathrm{d}\boldsymbol{x}_0 \\ &= \int \nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})\mathrm{d}\boldsymbol{x}_0. \end{aligned} \tag{27}$$

Using the identity

$$\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0), \tag{28}$$

we obtain

$$\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})\mathrm{d}\boldsymbol{x}_0. \tag{29}$$

This completes the proof. □

Lemma 2. *Because $\boldsymbol{x}_t$ and $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{x}_0$, i.e., $\boldsymbol{x}_t \perp \boldsymbol{y}|\boldsymbol{x}_0$, the posterior distribution of $\boldsymbol{x}_0$ given the diffusion state $\boldsymbol{x}_t$ and measurement $\boldsymbol{y}$ can be factorized as:*

$$p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}) = \frac{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})} \tag{30}$$

Proof. According to Bayes' rule, the conditional posterior distribution can be written as

$$\begin{aligned} p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}) &= \frac{p_t(\boldsymbol{x}_0, \boldsymbol{x}_t|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})} \\ &= \frac{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{y})p_t(\boldsymbol{x}_0|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})}. \end{aligned} \tag{31}$$

Since $\boldsymbol{x}_t$ and $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{x}_0$, we have $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{y}) = p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$. Therefore, the posterior distribution can be simplified as

$$p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}) = \frac{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})}. \tag{32}$$

□

Proposition 1. *In the case of VP-SDE, the perturbation kernel satisfies $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (\sqrt{1-\bar{\alpha}_t})^2\boldsymbol{I})$. Then the posterior score can be expressed as:*

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{1-\bar{\alpha}_t} \tag{33}$$

Proof. The posterior score can be derived as:

$$\begin{aligned} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) &= \frac{\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})} \\ &= \frac{\int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})\mathrm{d}\boldsymbol{x}_0}{p_t(\boldsymbol{x}_t|\boldsymbol{y})} \\ &= \int \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\frac{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})}\mathrm{d}\boldsymbol{x}_0 \\ &= \int \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y})\mathrm{d}\boldsymbol{x}_0 \\ &= \mathbb{E}_{p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y})}[\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)]. \end{aligned} \tag{34}$$

Since

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (\sqrt{1-\bar{\alpha}_t})^2 I), \tag{35}$$

we have

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t}. \tag{36}$$

Therefore,

$$\begin{aligned}\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) &= \mathbb{E}_{\boldsymbol{x}_0 \sim p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y})}\left[-\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t}\right] \\ &= -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{1-\bar{\alpha}_t}.\end{aligned} \tag{37}$$

Although the above result is derived for VP-SDE, an analogous posterior score formulation can be readily derived for other diffusion processes with Gaussian perturbation kernels, such as VE-SDE, by adapting the corresponding mean and variance of the perturbation kernel. □

Lemma 3 (Gaussian Moment Approximation)**.** *Given only the first and second moments of the denoising posterior $p_t(\boldsymbol{x}_0|\boldsymbol{x}_t)$, the moment-matched Gaussian distribution is the maximum-entropy approximation:*

$$p_t(\boldsymbol{x}_0|\boldsymbol{x}_t) \approx \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_t) \tag{38}$$

where

$$\hat{\boldsymbol{x}}_0 = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t], \qquad \boldsymbol{\Sigma}_t = \mathrm{Cov}(\boldsymbol{x}_0|\boldsymbol{x}_t) \tag{39}$$

Proof. Consider a family of distributions $q(x_0|x_t)$ that share the same first and second moments with the posterior distribution $p_t(\boldsymbol{x}_0|\boldsymbol{x}_t)$, i.e.

$$\mathbb{E}_q[\boldsymbol{x}_0] = \hat{\boldsymbol{x}}_0, \qquad \mathrm{Cov}_q(\boldsymbol{x}_0) = \boldsymbol{\Sigma}_t \tag{40}$$

Among all distributions satisfying these moment constraints, the maximum-entropy distribution q can be obtained by solving

$$q^* = \arg\max_q H(q), \tag{41}$$

where

$$H(q) = -\int q(\boldsymbol{x}_0|\boldsymbol{x}_t) \log q(\boldsymbol{x}_0|\boldsymbol{x}_t) \mathrm{d}\boldsymbol{x}_0 \tag{42}$$

is the differential entropy.

Introducing Lagrange multipliers for the normalization, mean, and covariance constraints gives the following functional:

$$\begin{aligned}\mathcal{L}(q) = &-\int q(\boldsymbol{x}_0|\boldsymbol{x}_t) \log q(\boldsymbol{x}_0|\boldsymbol{x}_t) \mathrm{d}\boldsymbol{x}_0 + \lambda_0 \left(\int q(\boldsymbol{x}_0|\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_0 - 1\right) + \boldsymbol{\lambda}_1^T \left(\int \boldsymbol{x}_0 q(\boldsymbol{x}_0|\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\right) \\ &+ \mathrm{Tr}\left[\boldsymbol{\Lambda}^T \left(\int (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T q(\boldsymbol{x}_0|\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_0 - \boldsymbol{\Sigma}_t\right)\right]\end{aligned} \tag{43}$$

where $\lambda_0 \in \mathbb{R}$, $\boldsymbol{\lambda}_1 \in \mathbb{R}^d$, and $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$ are Lagrange multipliers. Since the covariance constraint is symmetric, only the symmetric part of $\boldsymbol{\Lambda}$ contributes to the trace term. Therefore, without loss of generality, we assume $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T$.

Taking the functional derivative $\frac{\partial \mathcal{L}}{\partial q}$ with respect to $q(\boldsymbol{x}_0|\boldsymbol{x}_t)$ and setting it to zero yields:

$$\begin{aligned}\log q(\boldsymbol{x}_0|\boldsymbol{x}_t) &= \lambda_1^T \boldsymbol{x}_0 + (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T \boldsymbol{\Lambda}^T (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0) + \lambda_0 - 1 \\ &= \boldsymbol{x}_0^T \boldsymbol{\Lambda} x_0 + (\lambda_1 - 2\boldsymbol{\Lambda}\hat{\boldsymbol{x}}_0)^T \boldsymbol{x}_0 + C\end{aligned} \tag{44}$$

where $C = \lambda_0 - 1 + \hat{\boldsymbol{x}}_0^T \boldsymbol{\Lambda} \hat{\boldsymbol{x}}_0$ denotes a constant term independent of $\boldsymbol{x}_0$.

Therefore, the distribution $q(\boldsymbol{x}_0|\boldsymbol{x}_t)$ corresponds to an exponential quadratic form:

$$q^*(\boldsymbol{x}_0|\boldsymbol{x}_t) \propto \exp\left(\boldsymbol{x}_0^T \boldsymbol{\Lambda} \boldsymbol{x}_0 + (\boldsymbol{\lambda}_1 - 2\boldsymbol{\Lambda}\hat{\boldsymbol{x}}_0)^T \boldsymbol{x}_0\right) \tag{45}$$

By completing the square and defining $\boldsymbol{\Sigma}_t^{-1} = -2\boldsymbol{\Lambda}$, $\boldsymbol{\lambda}_1 = \boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + 2\boldsymbol{\Lambda}\hat{\boldsymbol{x}}_0$, the exponential quadratic form can be rewritten as:

$$q^*(\boldsymbol{x}_0|\boldsymbol{x}_t) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\boldsymbol{\Sigma}_t^{-1}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)\right) \tag{46}$$

Therefore, the optimal distribution belongs to the Gaussian family, i.e., the maximum entropy approximation of the posterior distribution $p_t(\boldsymbol{x}_0|\boldsymbol{x}_t)$ is given by its moment-matched Gaussian distribution:

$$p_t(\boldsymbol{x}_0|\boldsymbol{x}_t) \approx \mathcal{N}(\boldsymbol{x}_0; \hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_t) \tag{47}$$

□

Proposition 2. *Given the distribution $\boldsymbol{x}_0|\boldsymbol{x}_t \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_t)$ and the imaging forward model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{n}$, with $\boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{y}})$, the conditional denoising mean is given by:*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \hat{\boldsymbol{x}}_0 + \boldsymbol{\Sigma}_t\boldsymbol{A}^H(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \tag{48}$$

Proof. According to the Gaussian Moment Approximation, the denoising posterior distribution is approximated as:

$$\boldsymbol{x}_0|\boldsymbol{x}_t \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_t) \tag{49}$$

The linear imaging model is given by:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{n}, \qquad \boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{y}}). \tag{50}$$

Defining the centered variable and measurement residual as:

$$\boldsymbol{\delta} = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0, \qquad \boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0, \tag{51}$$

we have:

$$\boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t), \tag{52}$$

and

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{n}. \tag{53}$$

The cross-covariance between $\boldsymbol{\delta}$ and $\boldsymbol{r}$ is:

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{\delta}, \boldsymbol{r}) &= \mathbb{E}[(\boldsymbol{\delta} - \mathbb{E}[\boldsymbol{\delta}])(\boldsymbol{r} - \mathbb{E}[\boldsymbol{r}])^H] \\
&= \mathbb{E}[\boldsymbol{\delta}(\boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{n})^H] \\
&= \mathbb{E}[\boldsymbol{\delta}(\boldsymbol{\delta}^H\boldsymbol{A}^H)] + \mathbb{E}[\boldsymbol{\delta}]\mathbb{E}[\boldsymbol{n}^H] \\
&= \mathrm{Cov}(\boldsymbol{\delta})\boldsymbol{A}^H \\
&= \boldsymbol{\Sigma}_t\boldsymbol{A}^H,
\end{aligned} \tag{54}$$

where the independence between $\boldsymbol{\delta}$ and $\boldsymbol{n}$ is used. Similarly, $\mathrm{Cov}(\boldsymbol{r})$ is:

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{r}) &= \mathbb{E}[(\boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{n})(\boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{n})^H] \\
&= \boldsymbol{A}\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^H]A^H + E[\boldsymbol{n}\boldsymbol{n}^H] \\
&= \boldsymbol{A}\,\mathrm{Cov}(\boldsymbol{\delta})\boldsymbol{A}^H + \mathrm{Cov}(\boldsymbol{n}) \\
&= \boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \boldsymbol{\Sigma}_{\boldsymbol{y}}.
\end{aligned} \tag{55}$$

Since $\boldsymbol{\delta}$ and $\boldsymbol{n}$ are independent Gaussian random variables and $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{n}$ is a linear transformation of $(\boldsymbol{\delta}, \boldsymbol{n})$, the pair $(\boldsymbol{\delta}, \boldsymbol{r})$ is jointly Gaussian conditioned on $\boldsymbol{x}_t$. Therefore, the conditional expectation is:

$$\mathbb{E}[\boldsymbol{\delta}|\boldsymbol{r}] = \mathrm{Cov}(\boldsymbol{\delta}, \boldsymbol{r})\,\mathrm{Cov}(\boldsymbol{r})^{-1}\boldsymbol{r}. \tag{56}$$

Substituting the above covariance matrices gives

$$\mathbb{E}[\boldsymbol{\delta}|\boldsymbol{r}] = \boldsymbol{\Sigma}_t\boldsymbol{A}^H(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0). \tag{57}$$

Algorithm 1 Training

Require: physical forward operator $\boldsymbol{A}$ and joint dataset distribution $p_{\text{data}}(\boldsymbol{x}_0, \boldsymbol{y})$

repeat
  Sample $i \sim \text{Uniform}(\{0, 1, \cdots, N-1\})$
  $t_i = \exp(\log T + \frac{i}{N-1}(\log \epsilon - \log T))$
  Initialize $\boldsymbol{x}_{t_i}^{(0)} \sim p_z$
  repeat ▷ *Apply Anderson Method*
    $\boldsymbol{x}_{t_i}^{(k)} \leftarrow h_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_i:t_{i+1}}^{(k-1)}; \boldsymbol{y}; \boldsymbol{A})$
  until converged to $\boldsymbol{x}_{t_i}^*$
  $\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}} \leftarrow s_{\boldsymbol{\theta}}([\boldsymbol{x}_{t_i}^*; \boldsymbol{A}^H\boldsymbol{y}], t_i)$
  $\mathcal{L} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_{t_i}, y, t_i}[\|\boldsymbol{x}_{t_i} - \boldsymbol{x}_{t_i}^*\|_2^2 + \|\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}} - \boldsymbol{x}_0\|_2^2]$
  $\boldsymbol{\theta} \leftarrow \text{AdamW}(\nabla_{\boldsymbol{\theta}}(\mathcal{L}))$ ▷ *Phantom Gradient*
until converged
return trained network $s_{\boldsymbol{\theta}}$

Algorithm 2 Reconstruction

Require: Physical forward operator $\boldsymbol{A}$; measurement $\boldsymbol{y}$

Initialize $\boldsymbol{X}^{(0)} = [\boldsymbol{x}_T^{(0)}, \ldots, \boldsymbol{x}_\epsilon^{(0)}]$
repeat ▷ Apply Anderson Method
  $\boldsymbol{F}^{(k-1)} = F_{\boldsymbol{\theta}}(\boldsymbol{X}^{(k-1)}; \boldsymbol{y}; \boldsymbol{A}; \boldsymbol{t})$
  $\boldsymbol{X}^{(k-1)} = \boldsymbol{x}_T - \Delta t \ \text{CumSum}(\boldsymbol{F}^{(k-1)})$
  $\boldsymbol{r}^{(k-1)} = \boldsymbol{X}^{(k-1)} - \boldsymbol{X}^{(k-1)}$
  $\boldsymbol{R}_k = [\boldsymbol{r}^{(k-m_k)}, \ldots, \boldsymbol{r}^{(k-1)}]$
  $\boldsymbol{\alpha}^{(k-1)} = \arg\min_{\mathbf{1}^\top \boldsymbol{\alpha}=1} \|\boldsymbol{R}_k \boldsymbol{\alpha}\|_2^2$
  $T(\boldsymbol{X}^{(k-1)}) = \sum_{i=0}^{m_k-1} \alpha_i^{(k-1)} \boldsymbol{X}^{(k-1-i)}$
  $\boldsymbol{X}^{(k)} = (1-\tau)\boldsymbol{X}^{(k-1)} + \tau T(\boldsymbol{X}^{(k-1)})$
until converged to $\boldsymbol{X}^*$
$\boldsymbol{x}_0 \leftarrow s_{\boldsymbol{\theta}}([\boldsymbol{x}_\epsilon^*; \boldsymbol{A}^H\boldsymbol{y}], \epsilon)$
return reconstructed image $\boldsymbol{x}_0$

Finally, since $\boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0 + \boldsymbol{\delta}$,

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] &= \mathbb{E}[\hat{\boldsymbol{x}}_0 + \boldsymbol{\delta}|\boldsymbol{x}_t, \boldsymbol{y}] \\ &= \hat{\boldsymbol{x}}_0 + \mathbb{E}[\boldsymbol{\delta}|\boldsymbol{x}_t, \boldsymbol{y}] \\ &= \hat{\boldsymbol{x}}_0 + \boldsymbol{\Sigma}_t \boldsymbol{A}^H (\boldsymbol{A}\boldsymbol{\Sigma}_t \boldsymbol{A}^H + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1} (\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0). \end{aligned} \tag{58}$$

□

Remark 1 (Conjugate Gradient Solver)**.** *Using the Woodbury matrix identity (Golub & Van Loan, 2013; Lu & Huang, 2020), the conditional denoising mean in Eq. (12) can be equivalently reformulated as:*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \text{CG}\left(\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A}, \sigma^2\boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + \boldsymbol{A}^H\boldsymbol{y}, \hat{\boldsymbol{x}}_0, k_{\text{CG}}\right), \tag{59}$$

Here, the first argument of CG denotes the linear operator $\boldsymbol{x} \mapsto (\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A})\boldsymbol{x}$, and k_{CG} denotes the number of CG iterations.

Proof. Starting from Proposition 2, we have:

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \hat{\boldsymbol{x}}_0 + \boldsymbol{\Sigma}_t \boldsymbol{A}^H \left(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \sigma^2\boldsymbol{I}\right)^{-1} (\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0). \tag{60}$$

According to the Woodbury matrix identity, we have

$$\boldsymbol{\Sigma}_t \boldsymbol{A}^H \left(\boldsymbol{A}\boldsymbol{\Sigma}_t\boldsymbol{A}^H + \sigma^2\boldsymbol{I}\right)^{-1} = \left(\boldsymbol{\Sigma}_t^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^H\boldsymbol{A}\right)^{-1} \frac{1}{\sigma^2}\boldsymbol{A}^H. \tag{61}$$

Therefore, Eq. (61) can be reformulated as follows:

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \left(\boldsymbol{\Sigma}_t^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^H\boldsymbol{A}\right)^{-1} \left(\boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + \frac{1}{\sigma^2}\boldsymbol{A}^H\boldsymbol{y}\right). \tag{62}$$

This formulation naturally leads to an equivalent linear system, which can be efficiently solved using the conjugate gradient (CG) method. We denote the conditional denoising mean by $\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}} := \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]$. Then, the corresponding linear system is given by:

$$\left(\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A}\right)\hat{\boldsymbol{x}}_{0|t,\boldsymbol{y}} = \sigma^2\boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + \boldsymbol{A}^H\boldsymbol{y}. \tag{63}$$

Since $\boldsymbol{\Sigma}_t \succ 0$ and $\sigma^2 > 0$, the coefficient matrix $\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A}$ is Hermitian positive definite. Hence, the conditional denoising mean can be efficiently solved using conjugate gradient with k_{CG} iterations:

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \text{CG}\left(\sigma^2\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{A}^H\boldsymbol{A}, \sigma^2\boldsymbol{\Sigma}_t^{-1}\hat{\boldsymbol{x}}_0 + \boldsymbol{A}^H\boldsymbol{y}, \hat{\boldsymbol{x}}_0, k_{\text{CG}}\right). \tag{64}$$

A.2 Training and Reconstruction Strategy

In this section, we detail the training and reconstruction strategies to provide a clearer description of our method.

Forward Training Process We formulate the dynamics governed by $F_{\boldsymbol{\theta}}$ in Eq. (19) as a neural ODE and train the system as a whole. Our backward Euler discretization yields an implicit equilibrium equation at each time step. This formulation enables training within a deep equilibrium model (Bai et al., 2019). The overall training strategy is illustrated in Algorithm 1. Specifically, we construct a logarithmically spaced time grid over $[\epsilon, T]$ and uniformly sample a time point t_i from this grid. Then, following the specified perturbation kernel, we construct a pair of adjacent trajectory states $[\boldsymbol{x}_{t_{i+1}}, \boldsymbol{x}_{t_i}]$ and initialize $\boldsymbol{x}_{t_i}^{(0)}$ from $p_{\boldsymbol{z}}$, where $p_{\boldsymbol{z}}$ denotes the prior distribution associated with the corresponding SDE. Here, $\boldsymbol{x}_{t_{i+1}}$ is treated as the fixed boundary condition that anchors the equilibrium solution at the preceding noisy state. Starting from $\boldsymbol{x}_{t_i}^{(0)}$, Anderson acceleration is applied to solve for the equilibrium state $\boldsymbol{x}_{t_i}^*$, which is trained to match the corresponding target $\boldsymbol{x}_{t_i}$. Meanwhile, the network $s_{\boldsymbol{\theta}}$ is further trained to map $\boldsymbol{x}_{t_i}^*$ to the conditional denoising mean $\tilde{\boldsymbol{x}}_0$. This additional supervision is necessary because the final transition from $\boldsymbol{x}_\epsilon$ to $\boldsymbol{x}_0$ is performed by a single denoising step, requiring the network to directly recover the clean image from the terminal diffusion state. For the loss function $\mathcal{L}$, the implicit function theorem gives:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^*}\left(\boldsymbol{I} - \frac{\partial h_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}^*}\right)^{-1} \frac{\partial h_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \tag{65}$$

However, the implicit differentiation requires evaluating the inverse Jacobian-related term $\left(\boldsymbol{I} - \frac{\partial h_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}^*}\right)^{-1}$, which could be computationally expensive. To alleviate this computational burden, we adopt the one-step phantom gradient approximation (Geng et al., 2021). It avoids explicitly computing the inverse Jacobian while providing an efficient approximation to the implicit gradient.

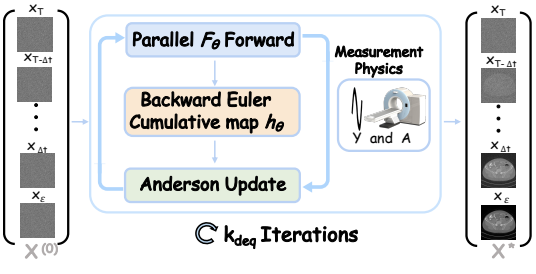

Figure 6: Illustration of our DEQ-Solver for solving medical inverse problems. Our method reformulates the reverse-time trajectory as a global fixed-point problem. The convergence of this problem to the equilibrium can be accelerated via the Anderson method.

Solving Inverse Problems Since our method can drive all temporal states to the global equilibrium in parallel, we initialize the entire reverse-time trajectory from $p_{\boldsymbol{z}}$. For the entire reverse-time trajectory, the terminal state $\boldsymbol{x}_T \sim p_{\boldsymbol{z}}$ serves as the boundary condition of the global fixed-point problem. Algorithm 2 and Fig. 6 detail this process, where the colored blocks correspond to each other across the algorithm and the figure. The entire trajectory is batch-evaluated through the neural vector field $F_{\boldsymbol{\theta}}$, after which each temporal state is updated using the cumulative backward Euler formulation. The Anderson acceleration is then employed to facilitate convergence to the equilibrium state. Therefore, in our method, the Number of Function Evaluations (NFEs) is determined by the number of Anderson iterations (k_{deq}), rather than scaling with the number of temporal states.

A.3 Dataset Details

We evaluate the efficiency of our DEQ-Solver on three distinct medical imaging modalities, ranging from functional to structural imaging.

MPI Synthetic Dataset To exhibit the strengths of MPI for high-sensitivity imaging of perihepatic tumors, we construct a synthetic MPI dataset. This dataset includes 16592 training, 3500 validation, and 4312 test images, drawn from three public datasets: FLARE23 (Ma & Wang, 2024), PanTS (Li et al., 2026), and MSD Pancreas Tumour (Antonelli et al., 2022). For each case, we selected axial slices containing both liver and tumor regions. To mimic the tracer-based functional imaging characteristics of MPI, only the liver and tumor regions were retained using the segmentation masks, while the background was removed. We resized the resulting images to 32×32, smoothed them with a Gaussian filter ($\sigma = 2$), and normalized them to $[0, 1]$ for MPI simulation. The system matrix ($\boldsymbol{A}$)

is derived from calibration 5 of the OpenMPI dataset, retaining components with a signal-to-noise ratio (SNR) ≥ 5 (Zou et al., 2025).

AAPM For sparse-view CT reconstruction, we adopt the training subset from the AAPM 2016 Low-Dose CT Grand Challenge (Kang et al., 2017). This comprises 10 full-dose CT volumes, with 7 volumes (4532 2D slices) for training, 2 volumes (844 slices) for validation, and 1 volume (600 slices) for testing. All images are resized to 256×256, clipped to the HU window$[-1000, 800]$, and then normalized to $[0, 1]$.

fastMRI We perform compressed-sensing MRI reconstruction experiments using the multi-coil fastMRI knee dataset. The training set contains 973 volumes (25012 2D slices). We split the 199 volumes from the validation set into 99 volumes (2549 slices) for validation and the remaining 100 volumes (1596 slices) for testing. Additionally, we use 100 volumes (568 slices) from the 0-th validation batch of fastMRI Brain to evaluate out-of-distribution generalization. All images are resized to 320×320. We discard the first and last five slices of each volume for training and validation, and the first and last ten slices for testing. For multi-coil reconstruction, we utilize MVUE complex-valued data as our reference (Jalal et al., 2021). Since the MVUE images have very small amplitudes and direct normalization to $[0, 1]$ would alter their relative scale with respect to the measured data $\boldsymbol{y}$, we rescale each image by the 99th percentile of its magnitude for stable network training.

A.4 Comparison Methods

In this section, we introduce the comparison methods and provide the corresponding parameter settings and implementation details used in our experiments.

KZ For Magnetic Particle Imaging reconstruction, we employ a Tikhonov regularization algorithm to solve:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2 \tag{66}$$

The Kaczmarz algorithm iteratively projects the current estimate onto the solution hyperplane defined by each row of the linear system. By introducing an auxiliary variable $\boldsymbol{r}$, the problem can be expressed as the augmented linear system:

$$[\boldsymbol{A}, \sqrt{\lambda}\boldsymbol{I}] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{r} \end{bmatrix} = \boldsymbol{y}. \tag{67}$$

For the k-th row, the regularized Kaczmarz update is given by

$$\beta_k = \frac{y_k - \boldsymbol{a}_k^H \boldsymbol{x} - \sqrt{\lambda} r_k}{\|\boldsymbol{a}_k\|_2^2 + \lambda}, \tag{68}$$

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta_k \boldsymbol{a}_k, \qquad r_k \leftarrow r_k + \sqrt{\lambda}\beta_k. \tag{69}$$

The updates are sequentially performed over all rows of the system matrix for a predefined number of iterations. We empirically set the number of Kaczmarz iterations to 5. The Tikhonov regularization parameter is set to $\lambda = 10^{-3}\|\boldsymbol{A}\|_F$, where $\|\boldsymbol{A}\|_F$ denotes the Frobenius norm of the system matrix.

CG For Eq. (66), the conjugate gradient (CG) method is also commonly used to solve the corresponding linear system. By taking the gradient of the objective function with respect to $\boldsymbol{x}$ and setting it to zero, we obtain the following normal equation:

$$(\boldsymbol{A}^H\boldsymbol{A} + \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{A}^H\boldsymbol{y} \tag{70}$$

The CG method is particularly suitable for systems with Hermitian positive-definite coefficient matrices. Starting from an initial estimate $\boldsymbol{x}^{(0)}$, we solve the regularized normal equation using:

$$\hat{\boldsymbol{x}} = \mathrm{CG}(\boldsymbol{A}^H\boldsymbol{A} + \lambda\boldsymbol{I}, \boldsymbol{A}^H\boldsymbol{y}, \boldsymbol{x}^{(0)}, k_{\mathrm{CG}}) \tag{71}$$

Specifically, CG initializes the residual and search direction as

$$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{B}\boldsymbol{x}^{(0)}, \qquad \boldsymbol{p}^{(0)} = \boldsymbol{r}^{(0)}, \tag{72}$$

where $\boldsymbol{B} = \boldsymbol{A}^H\boldsymbol{A} + \lambda\boldsymbol{I}$ and $\boldsymbol{b} = \boldsymbol{A}^H\boldsymbol{y}$. At the k-th iteration, the solution is updated by:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k\boldsymbol{p}^{(k)}, \quad \alpha_k = \frac{(\boldsymbol{r}^{(k)})^H\boldsymbol{r}^{(k)}}{(\boldsymbol{p}^{(k)})^H\boldsymbol{B}\boldsymbol{p}^{(k)}}. \tag{73}$$

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{B}\boldsymbol{p}^{(k)}, \tag{74}$$

$$\boldsymbol{p}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{p}^{(k)}, \quad \beta_k = \frac{(\boldsymbol{r}^{(k+1)})^H \boldsymbol{r}^{(k+1)}}{(\boldsymbol{r}^{(k)})^H \boldsymbol{r}^{(k)}}. \tag{75}$$

In experiments, we use $\lambda = 1 \times 10^{-3}$ with 20 iterations.

ADMM While ℓ_2 regularization provides a stable solution, it generally favors smooth estimates and does not explicitly promote sparsity. To better exploit the sparse spatial distribution that often arises in MPI, we further consider a LASSO-regularized formulation:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{76}$$

Since the ℓ_1 penalty in LASSO is non-smooth, we employ the Alternating Direction Method of Multipliers (ADMM) to decouple the quadratic data-fidelity term from the sparsity regularization, enabling efficient optimization through CG and soft-thresholding. Specifically, the ADMM updates are given by:

$$\boldsymbol{x}^{(k+1)} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{v}^{(k)} + \boldsymbol{w}^{(k)}\|_2^2, \tag{77}$$

$$\boldsymbol{v}^{(k+1)} = \arg\min_{\boldsymbol{v}} \lambda\|\boldsymbol{v}\|_1 + \frac{\rho}{2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{v} + \boldsymbol{w}^{(k)}\|_2^2, \tag{78}$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)}. \tag{79}$$

The $\boldsymbol{x}$-subproblem is solved using CG, while the $\boldsymbol{v}$-subproblem admits a closed-form solution through the LASSO soft-thresholding operator:

$$\boldsymbol{v}^{(k+1)} = \mathcal{S}_{\lambda/\rho}\left(\boldsymbol{x}^{(k+1)} + \boldsymbol{w}^{(k)}\right), \tag{80}$$

where $\mathcal{S}_{\lambda/\rho}(\cdot)$ denotes the soft-thresholding operator. For ADMM, we set $\lambda = 1$, $\rho = 1 \times 10^{-3}$, and the maximum number of ADMM iterations to 20. The CG solver used in the $\boldsymbol{x}$-subproblem is also run for a maximum of 20 iterations.

FBP For sparse-view CT, we employ fan-beam Filtered Back Projection (FBP) as the analytical baseline. Let $\boldsymbol{y}(\beta, u)$ denote the fan-beam projection data at projection angle β and detector coordinate u. FBP first applies geometry-dependent weighting followed by ramp filtering:

$$\tilde{\boldsymbol{y}}(\beta, u) = \left[\frac{D}{\sqrt{D^2 + u^2}}\boldsymbol{y}(\beta, u)\right] * h_{\text{ramp}}(u), \tag{81}$$

and subsequently reconstructs the image through distance-weighted backprojection

$$\hat{\boldsymbol{x}}(\xi, \eta) = \int \frac{D^2}{U^2(\xi, \eta, \beta)} \tilde{\boldsymbol{y}}(\beta, u(\xi, \eta, \beta))\, d\beta, \tag{82}$$

where $\hat{x}(\xi, \eta)$ denotes the reconstructed image intensity at spatial location (ξ, η).

Zero-Filled For compressed-sensing MRI, we use Zero-Filled reconstruction as a conventional baseline. The unmeasured k-space coefficients are filled with zeros, followed by an inverse Fourier transform to obtain the reconstructed image. For multi-coil MRI, the measurement of the c-th coil is modeled as

$$\boldsymbol{y}_c = \boldsymbol{A}_c\boldsymbol{x} + n_c \tag{83}$$

$$= \boldsymbol{M}\boldsymbol{F}\boldsymbol{S}_c\boldsymbol{x} + \boldsymbol{n}_c, \tag{84}$$

where $\boldsymbol{S}_c$ denotes the sensitivity map of the c-th coil, $\boldsymbol{F}$ is the Fourier transform, and $\boldsymbol{M}$ denotes the undersampling mask. The unmeasured k-space coefficients are first set to zero according to $\boldsymbol{M}$, followed by an inverse Fourier transform for each coil and sensitivity-weighted coil combination:

$$\hat{\boldsymbol{x}}_{\text{ZF}} = \frac{\sum_{c=1}^{C} \boldsymbol{S}_c^H \boldsymbol{F}^H (\boldsymbol{M}\boldsymbol{y}_c)}{\sqrt{\sum_{c=1}^{C} |\boldsymbol{S}_c|^2}}. \tag{85}$$

Here, C denotes the number of receiver coils.

For a fair comparison, the remaining SGM-based methods are evaluated using the same pre-trained score-model checkpoint.

DiffusionMBIR (Chung et al., 2023b) This method alternates between diffusion-based denoising and an explicit model-based optimization step, where measurement consistency is enforced through a likelihood-derived data-fidelity term together with a handcrafted regularizer. Since our experiments are conducted on 2D slices, the original through-plane TV regularization is not directly applicable. We therefore adapt the regularization strategy for different imaging modalities. At each sampling step, an intermediate reconstruction is first obtained as

$$\boldsymbol{x}'_{t-1} \leftarrow \mathrm{Solve}(\boldsymbol{x}_t, s^*_{\boldsymbol{\theta}}). \tag{86}$$

For CT and MPI, we replace the original TV regularization with an ℓ_2 regularization term:

$$\boldsymbol{x}_{t-1} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \boldsymbol{x}'_{t-1}\|_2^2. \tag{87}$$

The resulting quadratic problem is solved using CG with 5 iterations. We set $\lambda = 1$ for CT and $\lambda = 10$ for MPI.

For MRI, we instead employ an ℓ_1-regularized formulation:

$$\boldsymbol{x}_{t-1} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \boldsymbol{x}'_{t-1}\|_1. \tag{88}$$

The resulting LASSO problem is solved using a single ADMM update, following the fast variant of the original method. We set $\lambda = 1$ for MRI. This modality-specific regularization strategy empirically provides better reconstruction quality in our experiments.

DPS (Chung et al., 2023a) This method incorporates measurement information by approximating the intractable likelihood $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. Specifically, DPS first estimates the clean image as $\hat{\boldsymbol{x}}_0 = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ and approximates the marginal likelihood by evaluating the measurement likelihood at this denoised estimate:

$$p_t(\boldsymbol{y}|\boldsymbol{x}_t) \approx p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0). \tag{89}$$

The corresponding likelihood gradient is then used to correct the reverse diffusion update. Under Gaussian measurement noise, this yields the following least-squares correction:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} - \lambda_t \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2^2 . \tag{90}$$

We adaptively scale λ according to the data-fidelity residual, setting

$$\lambda_{\mathrm{MPI}} = \frac{5 \times 10^{-1}}{\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_1}, \qquad \lambda_{\mathrm{MRI}} = \frac{2 \times 10^{3}}{\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_1}. \tag{91}$$

For CT measurements corrupted by Poisson noise, the standard least-squares discrepancy does not adequately capture the signal-dependent noise statistics. Following the Poisson likelihood formulation in DPS, we therefore adopt a weighted data-fidelity term:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} - \lambda_t \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_{\boldsymbol{\Lambda}}^2 . \tag{92}$$

DDS (Chung et al., 2024) It is also a likelihood-based posterior sampling method. Given the denoised estimate $\hat{\boldsymbol{x}}'_0 = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$, DDS incorporates the measurement likelihood through a regularized least-squares data-consistency problem. Under Gaussian measurement noise, the data-fidelity term corresponds to the negative log-likelihood. In contrast to the explicit likelihood-gradient correction used in DPS, DDS incorporates the measurement likelihood by solving the resulting regularized linear system using conjugate gradient (CG):

$$\hat{\boldsymbol{x}}_0 = \mathrm{CG}(\boldsymbol{A}^H\boldsymbol{A} + \lambda\boldsymbol{I}, \boldsymbol{A}^H\boldsymbol{y} + \lambda\hat{\boldsymbol{x}}'_0, \hat{\boldsymbol{x}}'_0, k_{\mathrm{CG}}) \tag{93}$$

Specifically, DDIM (Song et al., 2021a) and DPM++ (Lu et al., 2025) can be naturally combined with DDS because both samplers update the reverse process through the denoised estimate $\hat{\boldsymbol{x}}_0$, while the CG-based measurement-conditioning step in DDS is also formulated on $\hat{\boldsymbol{x}}_0$. For MPI,

Table 4: Quantitative comparison of compressed-sensing brain MRI reconstruction at $8\times$ undersampling. All methods are trained on fastMRI Knee and evaluated on fastMRI Brain to assess out-of-distribution generalization.

Method		NFEs	$8\times$ undersampling		
			nRMSE ↓	PSNR ↑	SSIM ↑
Zero-Filled		–	0.033 ± 0.01	25.7 ± 2.37	0.694 ± 0.05
DiffusionMBIR		2000	0.068 ± 0.02	28.3 ± 3.51	0.823 ± 0.07
DPS		1000	0.033 ± 0.02	28.5 ± 3.62	0.833 ± 0.06
DDS	Baseline	1000	0.036 ± 0.02	27.6 ± 3.20	0.744 ± 0.07
	DDIM	100	0.034 ± 0.02	28.2 ± 3.40	0.818 ± 0.06
	DPM++	100	$\underline{0.032} \pm 0.02$	$\underline{28.8} \pm 3.57$	$\mathbf{0.859} \pm 0.05$
SCD		600	0.035 ± 0.02	27.9 ± 3.42	0.803 ± 0.06
DEQ-Solver (ours)		10	$\mathbf{0.025} \pm 0.01$	$\mathbf{30.3} \pm 2.25$	$\underline{0.857} \pm 0.05$

we set $\lambda = 1 \times 10^2$ at $\mathrm{SNR} = 20\,\mathrm{dB}$, and $\lambda = 1$ at $\mathrm{SNR} = 10\,\mathrm{dB}$ and $30\,\mathrm{dB}$. For DPM++ at $\mathrm{SNR} = 30\,\mathrm{dB}$, we set $\lambda = 1 \times 10^{-2}$. For MRI, we set $\lambda = 10$.

SCD (Barbano et al., 2025) This method is specifically designed for out-of-distribution (OOD) medical image reconstruction by adapting a pre-trained diffusion model to each test-time measurement. In essence, SCD remains likelihood-based, as the measurement likelihood guides the LoRA-based test-time adaptation to improve measurement consistency:

$$s_{\boldsymbol{\theta}'} = \mathrm{LoRA}(s_{\boldsymbol{\theta}}; \boldsymbol{y}, \boldsymbol{A}), \tag{94}$$

$$\hat{\boldsymbol{x}}_0' = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, s_{\boldsymbol{\theta}'}}[\boldsymbol{x}_0 | \boldsymbol{x}_t], \tag{95}$$

$$\hat{\boldsymbol{x}}_0 = \mathrm{CG}\left(\boldsymbol{A}^H\boldsymbol{A} + \lambda \boldsymbol{I},\ \boldsymbol{A}^H\boldsymbol{y} + \lambda \hat{\boldsymbol{x}}_0',\ \hat{\boldsymbol{x}}_0',\ k_{\mathrm{CG}}\right). \tag{96}$$

Deep Unrolling Methods Deep unrolling methods unfold an iterative reconstruction algorithm into a fixed number of trainable stages, where learned regularization is combined with explicit data consistency. The learned prior is usually embedded in a deterministic reconstruction network and is optimized for a specific inverse problem. This differs from score-based generative models, which learn a generative prior over the image distribution and perform reconstruction through posterior sampling. For deep unrolling baselines, we adopt DEQ-MPI (Güngör et al., 2023) for MPI, and RegFormer (Xia et al., 2024) and PAIL (Zhang et al., 2025) for CT.

A.5 Further Experimental Results

Out-of-Distribution As our method is training-based, its robustness to out-of-distribution (OOD) data warrants further investigation. We therefore evaluate our model on fastMRI Brain, which is completely unseen during training. The quantitative results are reported in Tab. 4, and the corresponding reconstructions are shown in Fig. 15 and Fig. 16. Specifically, at $8\times$ undersampling, DEQ-Solver achieves a PSNR of $30.3\,\mathrm{dB}$ using only 10 NFEs. Our method also achieves the lowest nRMSE and highest PSNR among all evaluated methods. It should be emphasized that our MRI experiments use the original acquired k-space measurements. Therefore, the lower nRMSE achieved by our method reflects higher fidelity to the real measured signal and, consequently, better quantitative accuracy. This is particularly important for clinical assessment, where accurate quantitative measurements can support longitudinal comparison, disease monitoring, and evaluation of treatment response. Compared with DDS using DPM++, DEQ-Solver requires $10\times$ fewer NFEs and improves PSNR by $1.5\,\mathrm{dB}$, with only a slight decrease in SSIM (0.857 vs. 0.859). As for SCD (Barbano et al., 2025), which is specifically designed for OOD reconstruction, our method achieves a $2.4\,\mathrm{dB}$ improvement in PSNR, reduces nRMSE from 0.035 to 0.025, increases SSIM from 0.803 to 0.857, and reduces the number of NFEs by $60\times$. These results suggest that substantial reductions in sampling cost can be achieved without compromising OOD generalization under the distribution shift from fastMRI Knee to fastMRI Brain.

Ablation on the Trajectory Formulation and Solving Strategy To understand where the performance gains of our trajectory-level DEQ formulation come from, we separately examine the effects

Table 5: Comparison of trajectory formulations and solving strategies for MPI reconstruction at 20 dB SNR.

Method	Implicit	Global	Trajectory Length			
			10	20	50	100
DeqIR	✗	✓	33.8 ± 4.27	35.2 ± 4.12	35.3 ± 4.17	35.1 ± 4.17
DEQ-Seq	✓	✗	36.1 ± 4.60	36.7 ± 4.83	37.2 ± 4.91	37.4 ± 4.93
DEQ-Solver	✓	✓	38.5 ± 4.97	38.2 ± 4.86	37.5 ± 4.76	37.9 ± 5.31

(a) PSNR Performance

Method	Implicit	Global	Trajectory Length			
			10	20	50	100
DeqIR	✗	✓	0.873 ± 0.06	0.871 ± 0.05	0.853 ± 0.06	0.839 ± 0.06
DEQ-Seq	✓	✗	0.956 ± 0.03	0.968 ± 0.02	0.972 ± 0.02	0.973 ± 0.02
DEQ-Solver	✓	✓	0.980 ± 0.02	0.982 ± 0.02	0.974 ± 0.02	0.980 ± 0.03

(b) SSIM Performance

of implicit state dependence and global trajectory solving. We consider three trajectory-solving configurations. DeqIR (Cao et al., 2024) represents the explicit-global setting, where an explicit diffusion sampling chain is reformulated as a joint fixed-point system for parallel evaluation. DEQ-Seq adopts our implicit backward Euler formulation but solves the trajectory sequentially over time. DEQ-Solver solves the global trajectory equilibrium that naturally arises from the implicit backward Euler discretization of the posterior dynamics.

*1) **Implicit vs. Explicit Trajectory Dependencies*** As discussed in the main text, our method introduces an implicit dependence on the state being solved at each time step, whereas DeqIR (Cao et al., 2024) constructs a joint fixed-point system from an explicit diffusion sampling chain. As shown in Table 5, DEQ-Seq consistently outperforms DeqIR across different trajectory lengths. This suggests that the implicit backward Euler formulation provides a more stable approximation of the posterior dynamics, especially under coarse temporal discretization.

*2) **Global vs. Sequential Trajectory Sampling*** To further isolate the effect of global trajectory solving, we compare DEQ-Seq with DEQ-Solver, both of which use the same implicit backward Euler formulation. DEQ-Solver achieves consistently higher reconstruction quality, particularly for short trajectories. One possible explanation is that sequential solving propagates local approximation errors along the reverse trajectory, whereas the global equilibrium solver repeatedly refines all temporal states and enforces trajectory-wide consistency. This allows DEQ-Solver to maintain accurate reconstruction with fewer temporal states.

*3) **Effect of Trajectory Length*** Trajectory length introduces a trade-off between memory usage and temporal discretization accuracy. A longer trajectory requires more temporal states and therefore higher memory consumption, while also providing a finer temporal discretization with smaller step sizes. As shown in Table 5, increasing the trajectory length does not consistently improve reconstruction performance, and $N = 10$ achieves the best results. One possible explanation is that a longer trajectory enlarges the global fixed-point system and increases the range of temporal coupling, which can make equilibrium solving more difficult. Under a fixed DEQ iteration budget, this increased solving difficulty may offset the benefit of finer temporal discretization.

Ablation on Timestep Scheduling We compare logarithmically and uniformly spaced time schedules with a trajectory length of $N = 100$, as shown in Tab. 6. The logarithmic schedule outperforms the uniform schedule. One possible explanation is that, under the VP-SDE, logarithmic spacing better accommodates the non-uniform evolution of the posterior vector field over diffusion time. It allocates more temporal states to regions where the dynamics vary rapidly, while using fewer states in more slowly varying regions. This more adaptive allocation of temporal states may therefore provide a more accurate discretization of the reverse-time dynamics under a fixed trajectory length.

Ablation on the Number of Function Evaluations We further examine the effect of the NFEs, defined here as the iteration budget of the DEQ fixed-point solver. As shown in Table 7, increas-

Table 6: Comparison of time step schemes for MPI reconstruction at 20 dB SNR.

Method		Time Scheme	
		Log	Linear
Ours	PSNR ↑	37.9 ± 5.31	37.0 ± 5.72
	SSIM ↑	0.980 ± 0.03	0.979 ± 0.03

Table 7: Ablation study on the number of function evaluations (NFEs).

Method		NFEs		
		10	50	100
Ours	PSNR ↑	38.5 ± 4.97	38.5 ± 5.25	38.6 ± 5.16
	SSIM ↑	0.980 ± 0.02	0.983 ± 0.02	0.979 ± 0.02
	FPS ↑	6.33	1.40	0.71

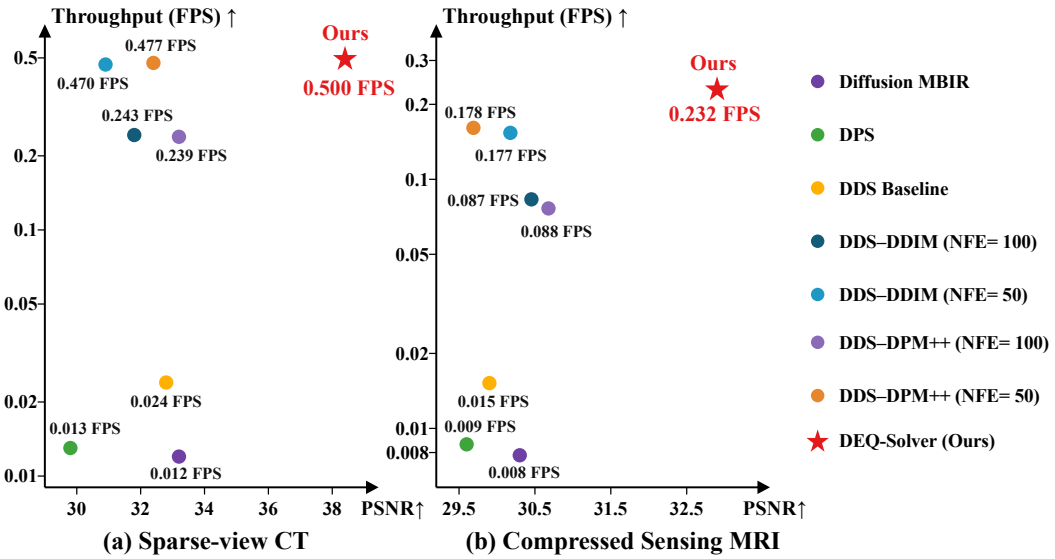

Figure 7: FPS (log scale) vs. PSNR for CT and MRI reconstruction. The PSNR values are evaluated under 32-view CT and 8× MRI undersampling, respectively.

ing the NFEs from 10 to 100 provides only marginal improvements in reconstruction quality, with PSNR remaining around $38.5 - 38.6$ dB and SSIM showing only minor variations. In contrast, the inference speed decreases substantially, from 6.33 FPS at 10 NFEs to 0.7 FPS at 100 NFEs. These results indicate that the global equilibrium can be approximated sufficiently well with a small number of fixed-point iterations, while further iterations yield diminishing reconstruction gains at a considerable computational cost. We therefore use 10 NFEs as a practical trade-off between reconstruction quality and sampling efficiency in all comparison experiments throughout this work. Fig. 7 further compares the resulting FPS of our method with those of other methods under different NFE settings. On both sparse-view CT and compressed-sensing MRI, our method reaches the highest PSNR and runs roughly 2× as fast as the best accelerated sampler at 100 NFEs.

Ablation on the spatial weighting map $\boldsymbol{W}_t$ We investigate the effect of the spatial weighting map $\boldsymbol{W}_t$ in Table 8. When $\boldsymbol{W}_t = \boldsymbol{I}$, no spatially adaptive weighting is introduced, and all image locations are treated equally during the data-consistency update, resulting in the lowest reconstruction quality. Scaling the identity matrix to $1 \times 10^{-2} \cdot \boldsymbol{I}$ brings only a limited improvement, which suggests that adjusting a single global weight is insufficient. In contrast, for the learnable settings, allowing $\boldsymbol{W}_t$ to vary spatially leads to substantial gains in both PSNR and SSIM. Moreover, expanding the admissible range from $[10^{-2}, 1]$ to $[10^{-3}, 10^{1}]$ further improves performance and achieves the best results. This indicates that spatially adaptive weighting is important for modulating the relative contributions of the denoising estimate and measurement information across different spatial locations, while a sufficiently wide dynamic range provides greater flexibility for such adaptation.

A.6 Qualitative Results

In this section, we provide additional qualitative reconstruction results to facilitate a better visual comparison.

Table 8: Effect of the spatial weighting map $\boldsymbol{W}_t$ on MPI reconstruction quality at an SNR of 10 dB. The intervals denote the bounds $[\lambda_{\min}, \lambda_{\max}]$ used in the sigmoid parameterization of $\boldsymbol{W}_t$

$\boldsymbol{W}_t$	PSNR ↑	SSIM ↑
$\boldsymbol{I}$	30.8 ± 4.83	0.751 ± 0.11
$1 \times 10^{-2} \cdot \boldsymbol{I}$	31.3 ± 4.65	0.762 ± 0.08
$[1 \times 10^{-2}, 1]$	32.1 ± 5.13	0.871 ± 0.05
$[1 \times 10^{-3}, 10]$	32.9 ± 5.16	0.958 ± 0.03

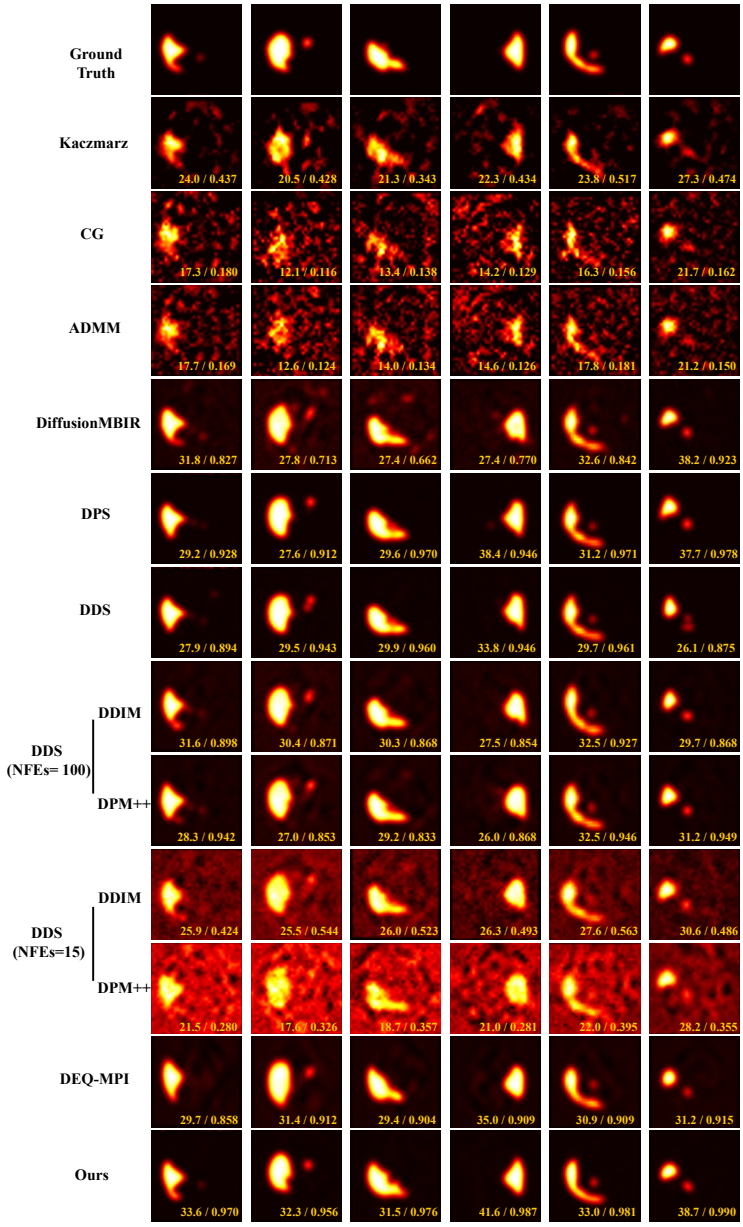

Figure 8: MPI reconstruction results at SNR $= 10$dB.

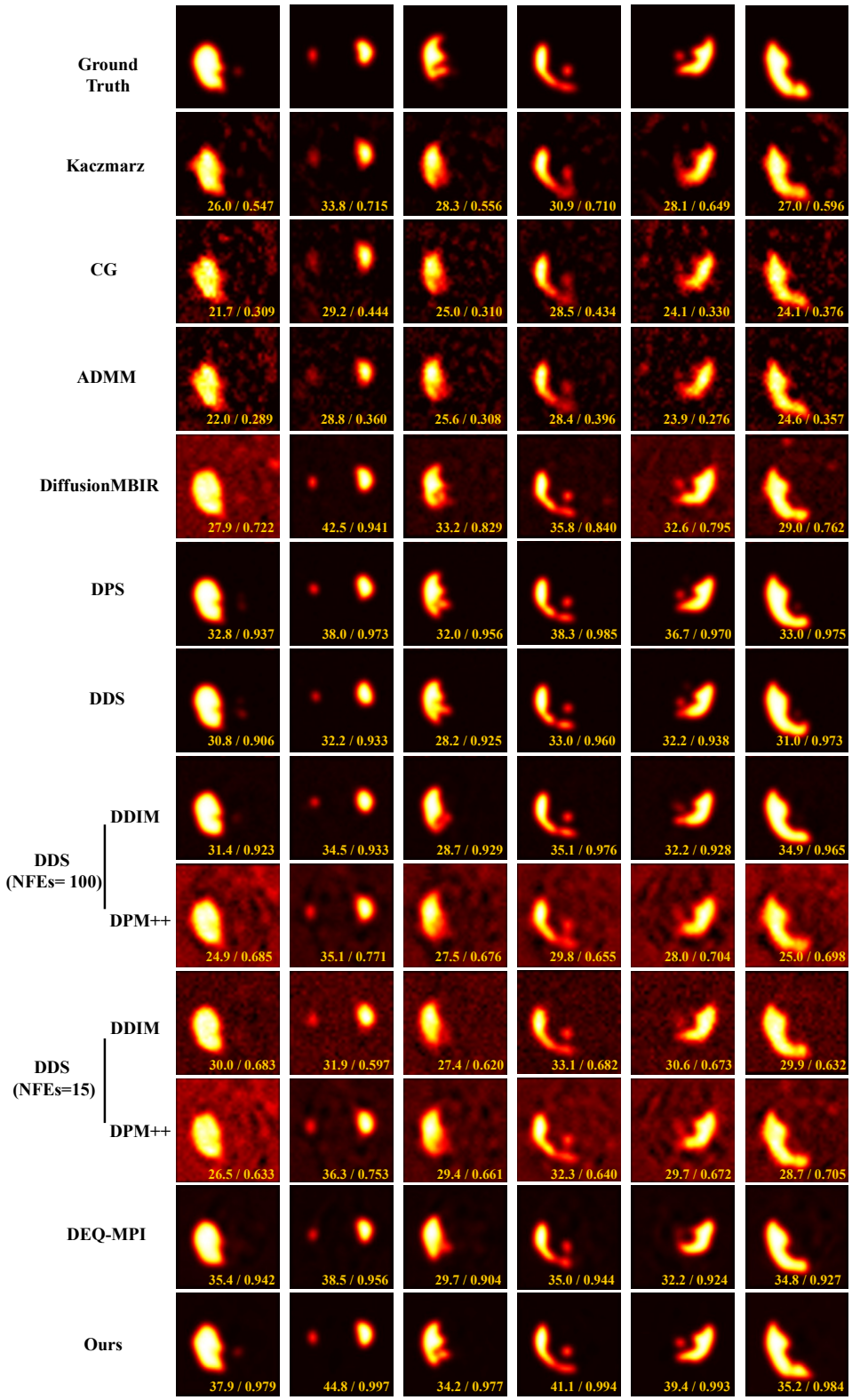

Figure 9: MPI reconstruction results at SNR $= 20\text{dB}$.

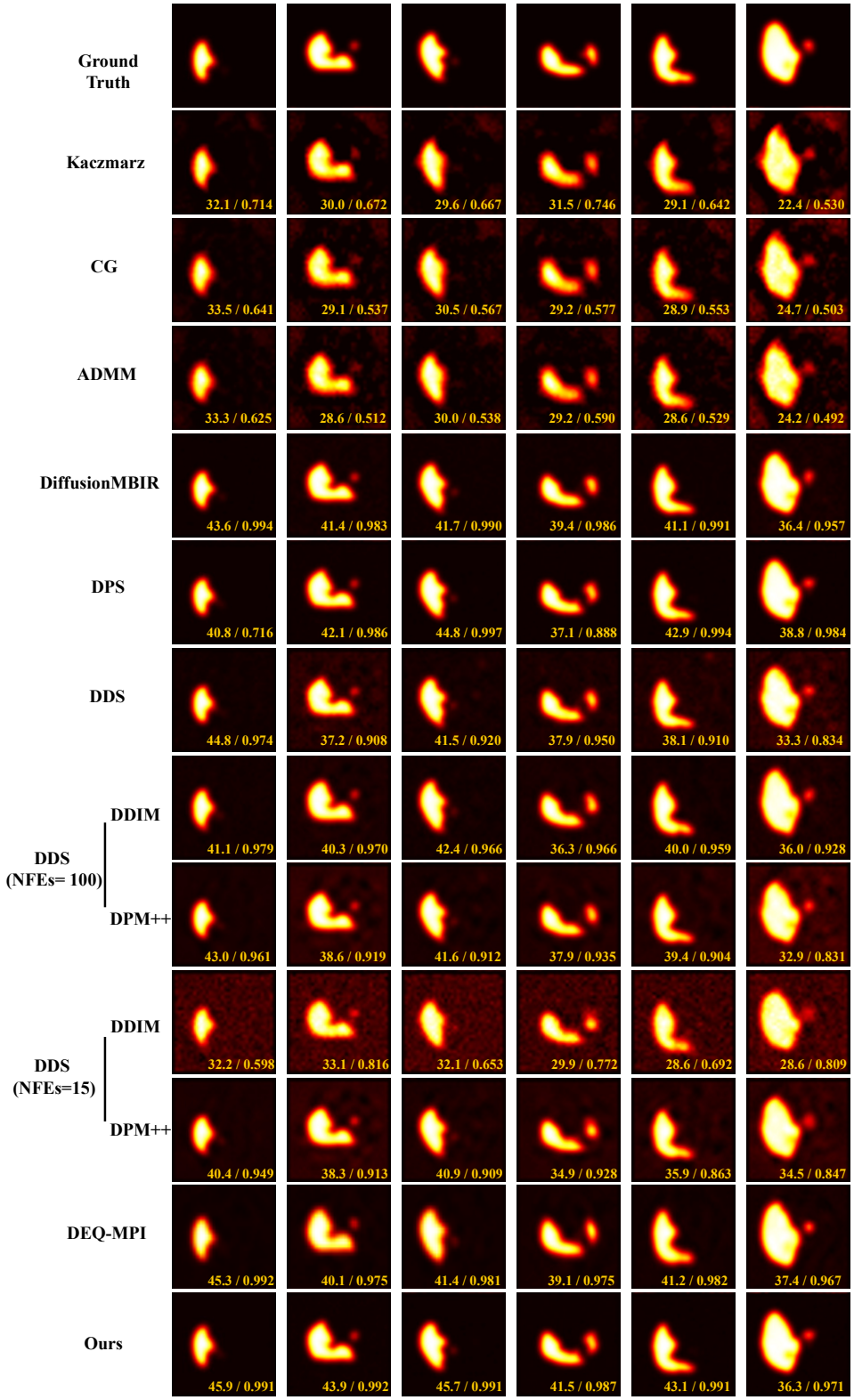

Figure 10: MPI reconstruction results at SNR $= 30\mathrm{dB}$.

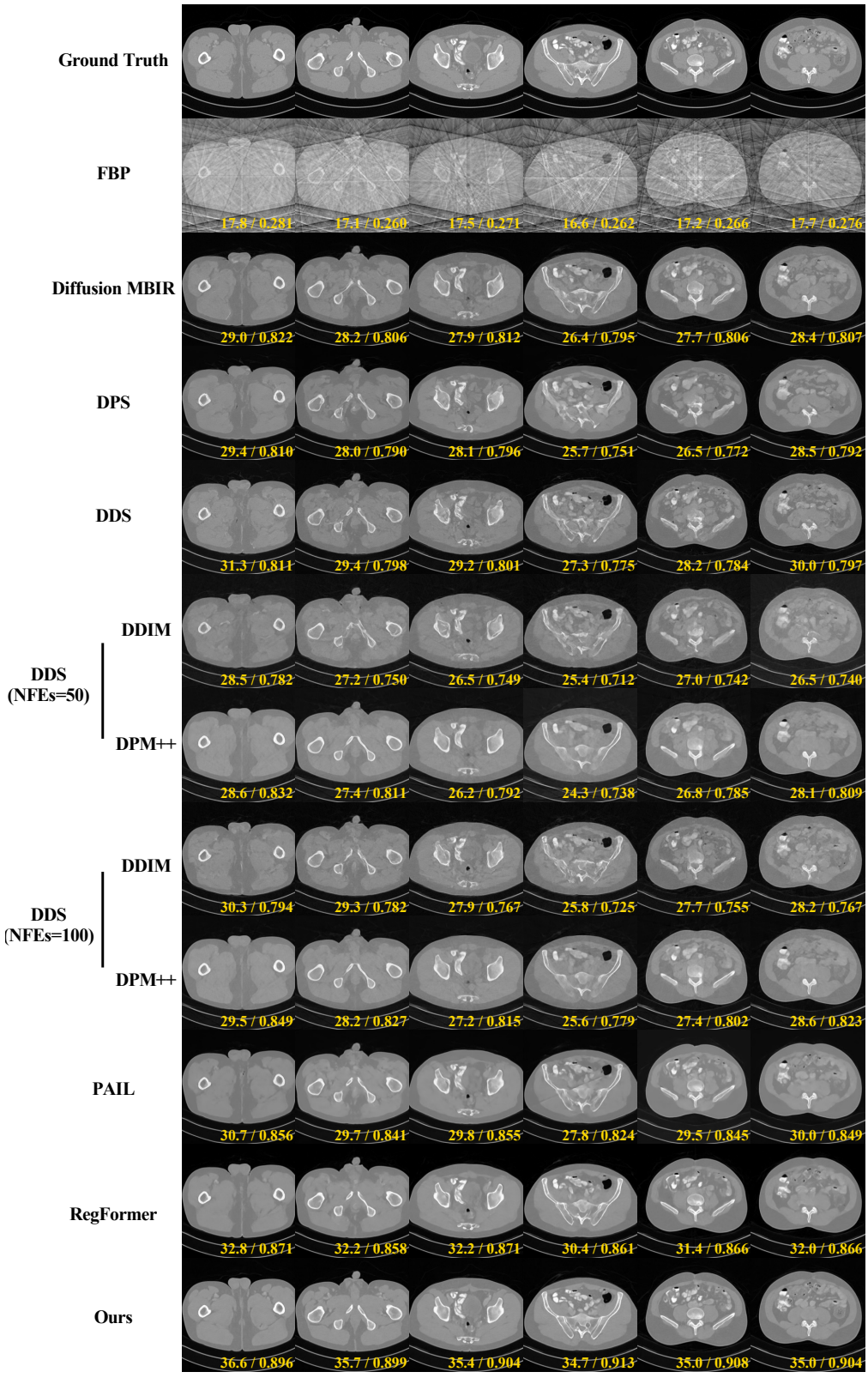

Figure 11: 16-View CT reconstruction results.

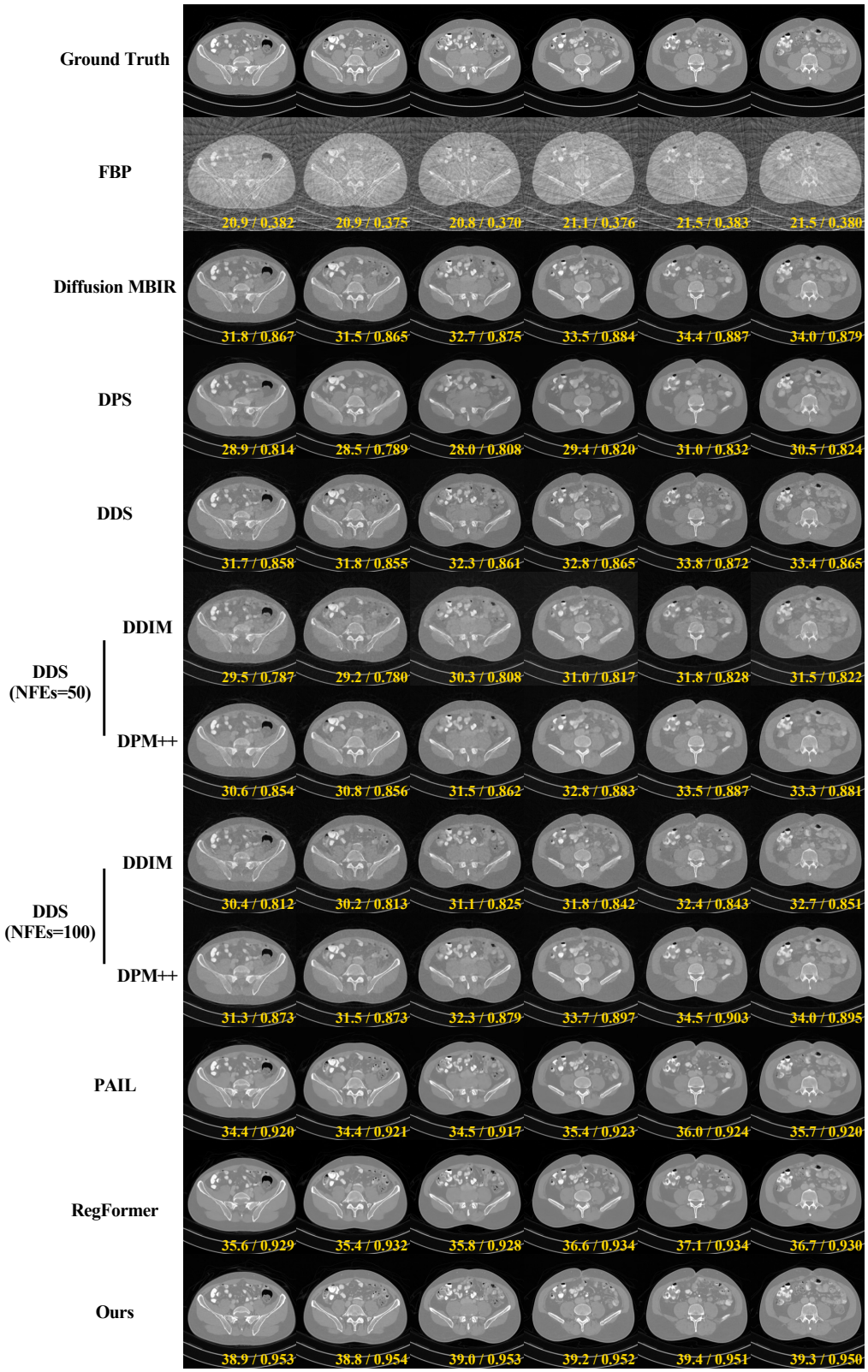

Figure 12: 32-View CT reconstruction results.

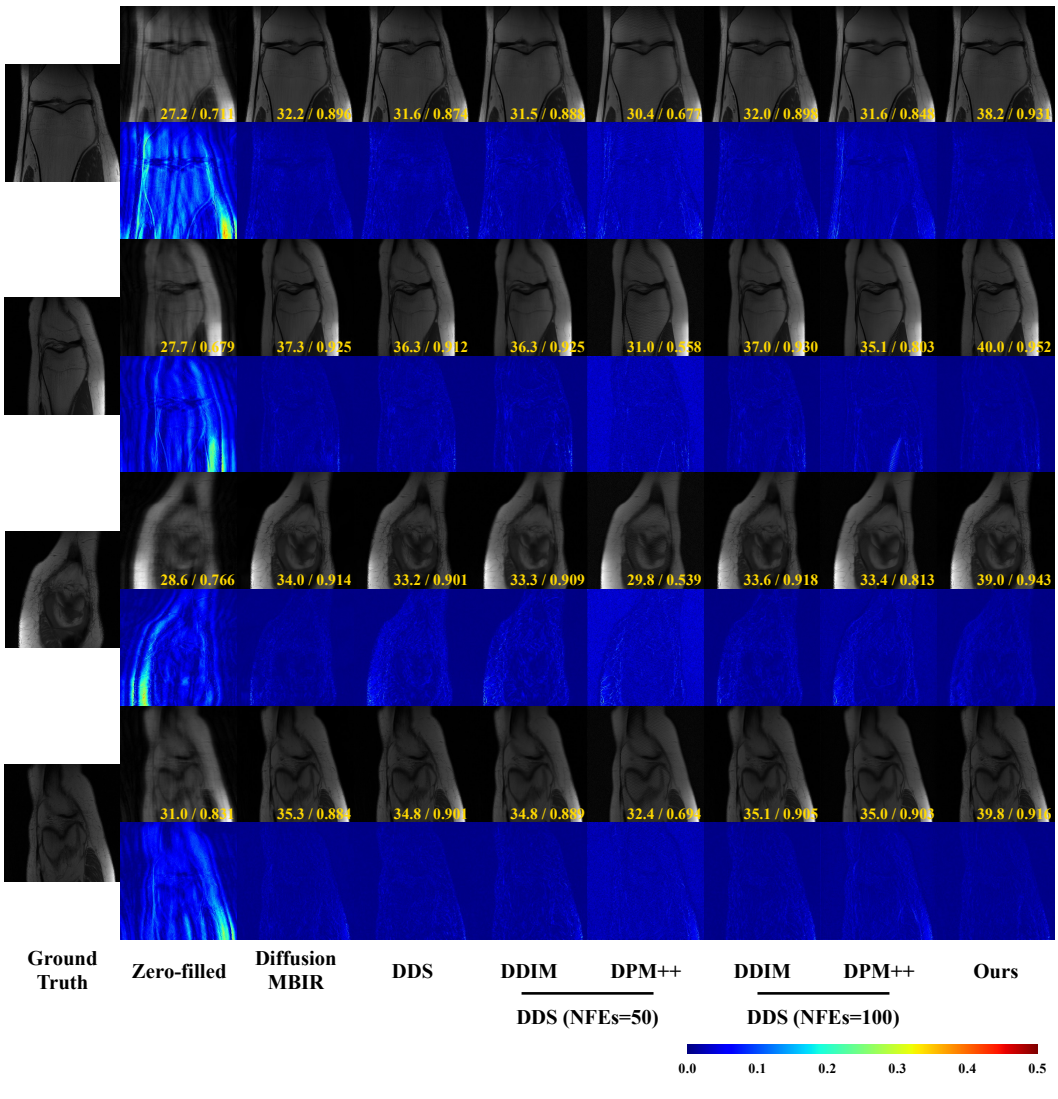

Figure 13: Representative reconstruction results on fastMRI Knee under $8\times$ undersampling.

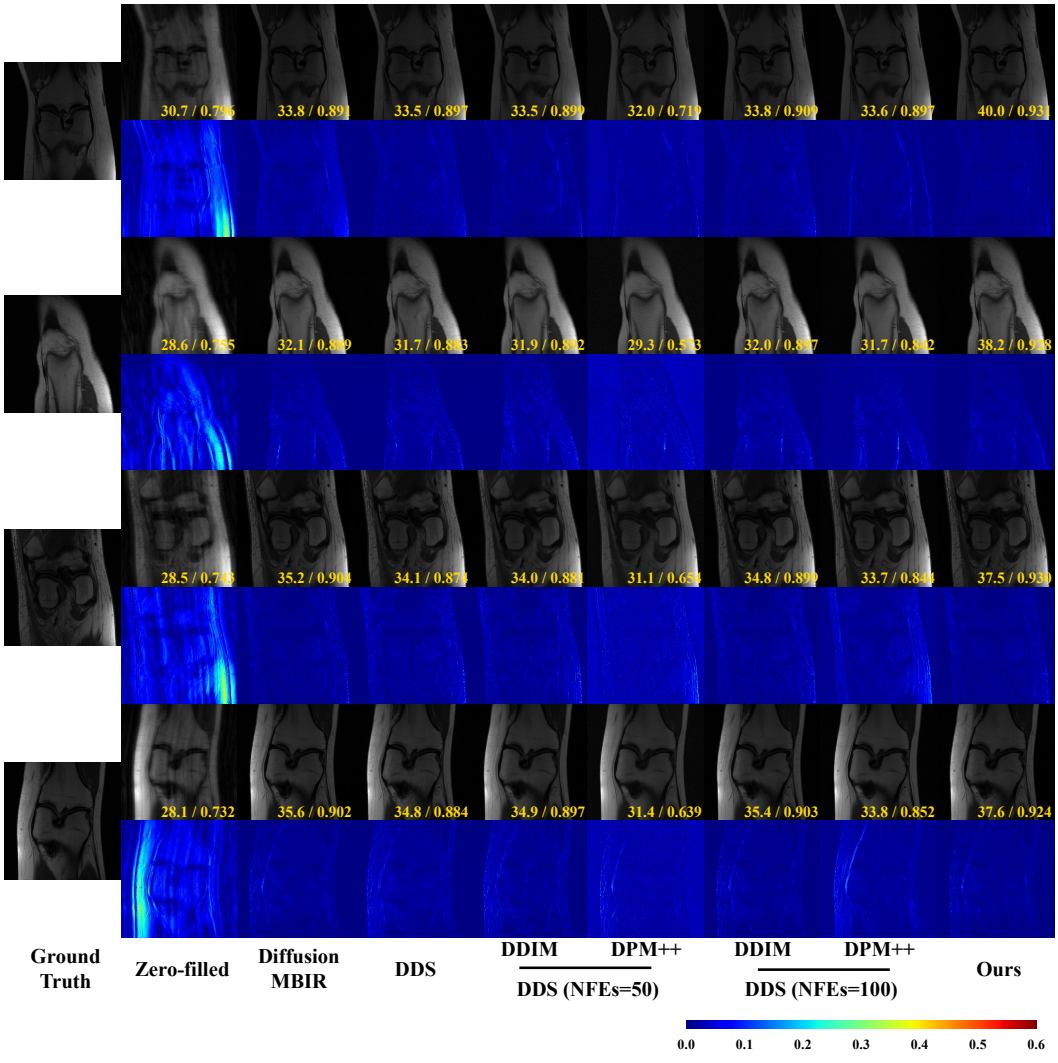

Figure 14: Additional reconstruction results on fastMRI Knee under $8\times$ undersampling.

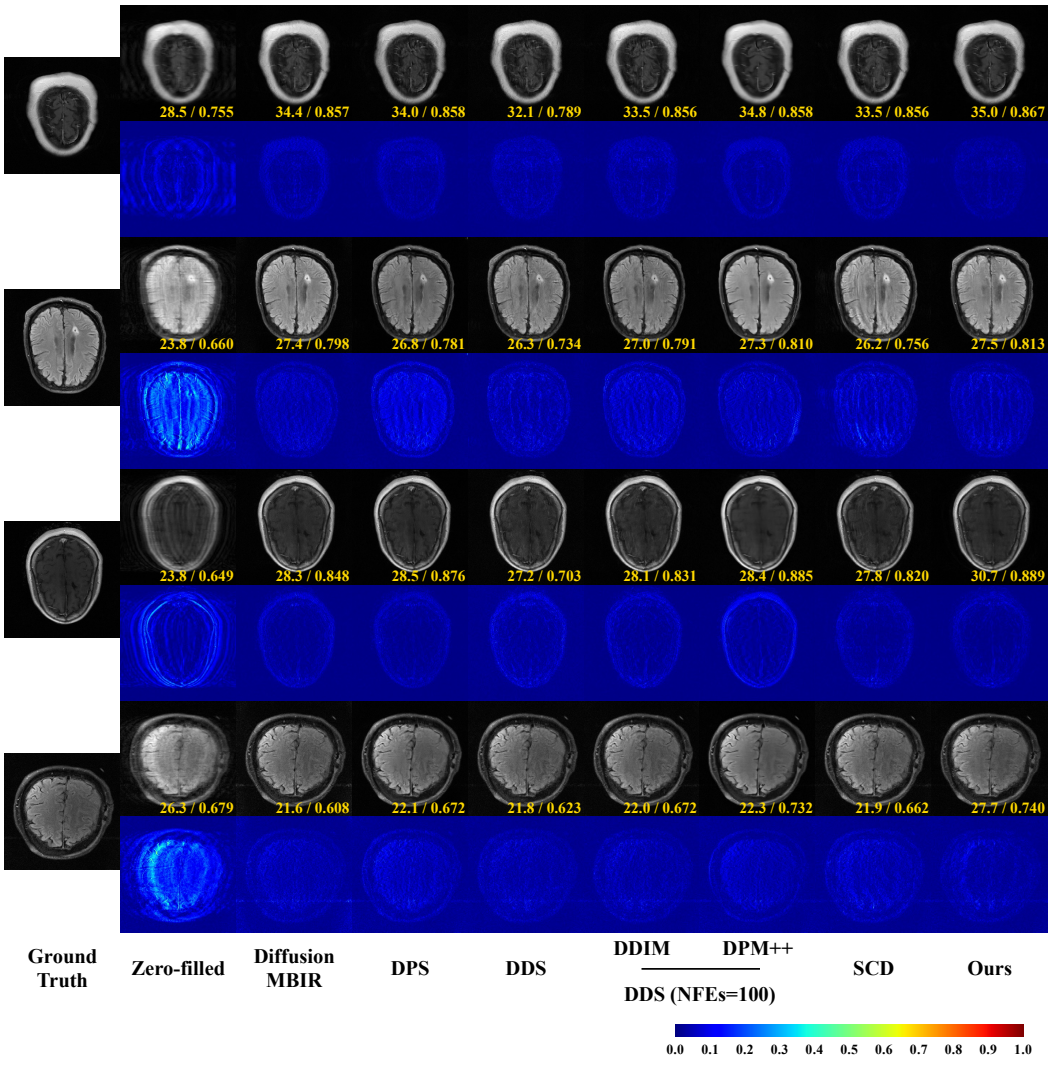

Figure 15: Representative reconstruction results on fastMRI Brain under $8\times$ undersampling.

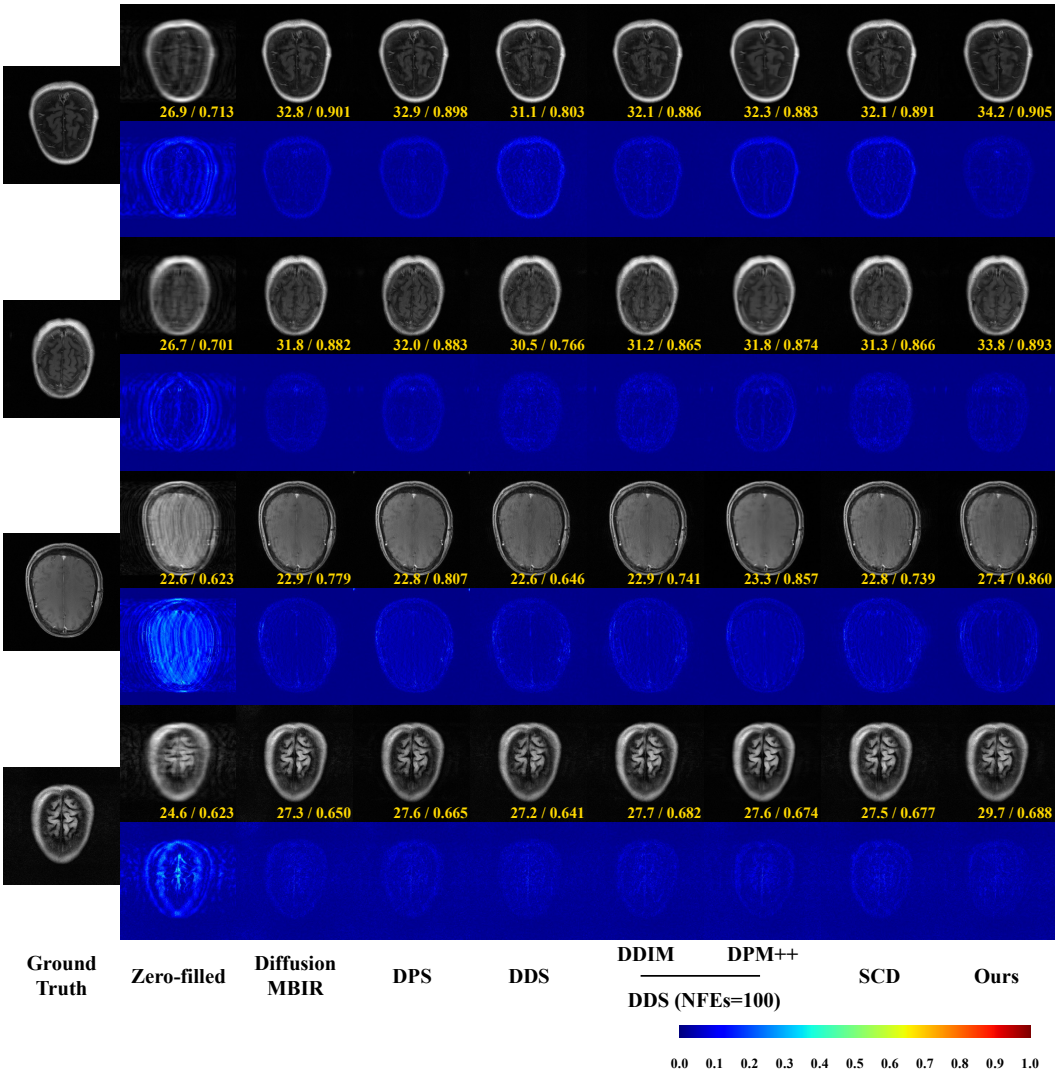

Figure 16: Additional reconstruction results on fastMRI Brain under $8\times$ undersampling.