Optimizing Sampling Patterns for Compressed Sensing MRI with Diffusion Generative Models

Sriram Ravula
University of Texas at Austin
Electrical and Computer Engineering
sriram.ravula@utexas.edu

Brett Levac
University of Texas at Austin
Electrical and Computer Engineering
blevac@utexas.edu

Ajil Jalal
University of California, Berkeley
Electrical Engineering and Computer Sciences
ajiljalal@berkeley.edu

Jonathan I. Tamir
University of Texas at Austin
Electrical and Computer Engineering
jtamir@utexas.edu

Alexandros G. Dimakis
University of Texas at Austin
Electrical and Computer Engineering
dimakis@austin.utexas.edu

Abstract

Diffusion-based generative models have been used as powerful priors for magnetic resonance imaging (MRI) reconstruction. We present a learning method to optimize sub-sampling patterns for compressed sensing multi-coil MRI that leverages pre-trained diffusion generative models. Crucially, during training we use a single-step reconstruction based on the posterior mean estimate given by the diffusion model and the MRI measurement process. Experiments across varying acceleration factors and pattern types show that sampling operators learned with our method lead to competitive, and in the case of 2D patterns, improved reconstructions compared to baseline patterns.

1 Introduction

Compressed sensing (CS) [7, 11] has been used to accelerate magnetic resonance imaging (MRI) beyond the Nyquist rate by sampling a pseudo-random subset of Fourier coefficients in k-space and imposing a sparse prior on the image [20]. More recently, deep learning has been used as a powerful tool to solve ill-posed inverse problems such as MRI reconstruction beyond the capabilities of sparsity priors [21]. In particular, several approaches have been introduced to optimize the sampling pattern for MRI either separately or jointly with the reconstruction network through end-to-end learning [4, 1, 29, 23, 33, 2]. However, these approaches are only suitable when the gradient of the sampling operator can be calculated through the full reconstruction process. Thus, while these techniques have been successful for optimizing sampling patterns for end-to-end reconstruction networks, it is unclear how to extend them to unsupervised methods such as CS with generative models (CSGM) [6].

Our Contributions In this work we optimize the sampling operator for CSGM MRI. We use the recently proposed unsupervised method of posterior sampling with diffusion models [15, 9, 19, 25], since it has been shown to be robust to changes in imaging anatomy, acceleration factor, and sub-sampling patterns without requiring retraining. Since diffusion-based posterior sampling is not amenable to algorithm unrolling, we propose a simple and effective alternative to the full gradient of

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

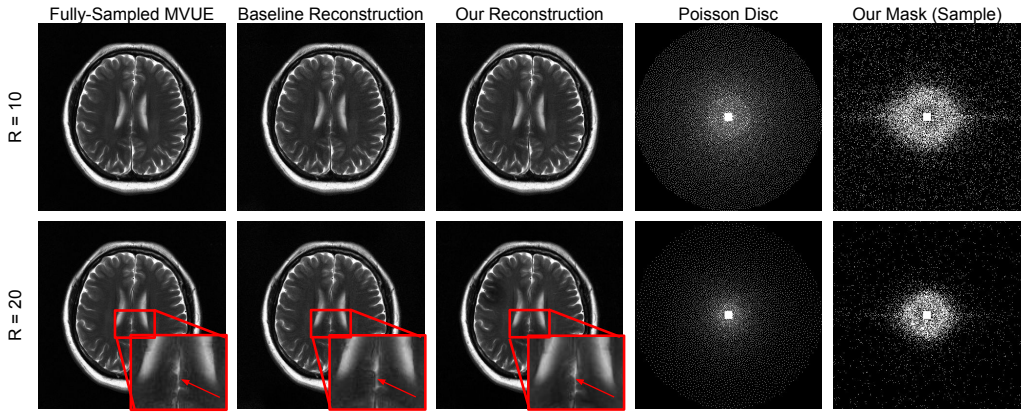


Figure 1: **Reconstructions using baseline masks vs masks learned with our method.** We perform posterior sampling using a diffusion model to reconstruct MRI scans using varying pattern types and acceleration factors (R). Artifacts in the reconstructions are highlighted with red arrows and insets (zoom may be needed). Masks learned with our method produce reconstructions with fewer artifacts.

the reconstruction process that can be used to optimize the selection of which samples to keep. We apply our approach to multi-coil Cartesian MRI and demonstrate that optimized sampling patterns can be used to reduce the reconstruction error for a given acceleration factor.

2 Background

Diffusion-Based Generative Models Diffusion-based models learn to generate new signals by reversing a corruption process, such as removing additive Gaussian noise. Early work treated various discrete-time diffusion processes [24, 13], which Song et al. [26] unify under the framework of continuous-time Stochastic Differential Equations (SDEs). We focus on the Variance Exploding (VE) class of SDEs.

The noisy signal at time t in the forward diffusion process is given by $\mathbf{x}_t \in \mathbb{R}^n$. The diffusion process is modeled as the solution to an Itô SDE of the form

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

Here, $\mathbf{w}$ is the standard n-dimensional Wiener process, and in the VE case, $\mathbf{f}(\mathbf{x}, t) = \mathbf{0}$ and $g(t) = \sqrt{d\sigma_t^2/dt}$. The variance σ_t^2 is a monotonically increasing function that defines the distribution of the diffused signal, with the property that at time $t = 0$ we recover the data distribution: $\mathbf{x}_0 \sim p_0 = p_{data}$. The goal of Diffusion-based models is to start from samples $\mathbf{x}_T \sim p_T$ consisting of pure Gaussian noise and reverse the forward diffusion given by Eq. (1) to arrive at samples $\mathbf{x}_0 \sim p_0$. Conveniently, the reverse of the forward SDE is also an SDE [3], with the form

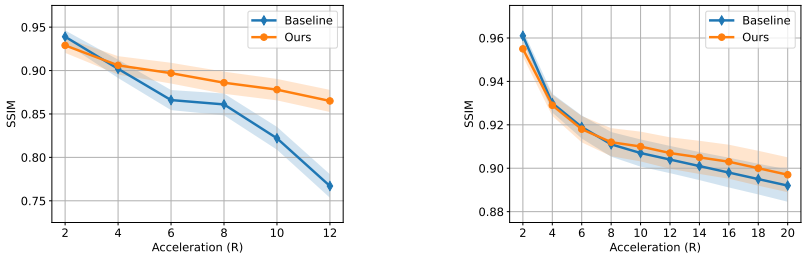
$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)]dt + g(t)d\bar{\mathbf{w}}, \tag{2}$$

where dt is now a negative time step and $\bar{\mathbf{w}}$ is the standard Wiener process when time flows backward. The reverse SDE depends on the *score function* $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ of the marginal distribution at time t, which can be estimated by training a *score network* $\mathbf{s}_\theta(\mathbf{x}_t, t)$ with denoising score matching [28] loss so that $\mathbf{s}_\theta(\mathbf{x}_t, t) \simeq \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$.

Posterior Sampling We are given measurements of the form $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \epsilon$, where $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a known forward operator, $\epsilon \in \mathbb{R}^m$ is some additive noise, and $\mathbf{x}_0 \in \mathbb{R}^n$ is a signal we want to recover. Using Bayes' rule, we observe that the score of the posterior distribution can be decomposed as $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})$. We can reconstruct the signal using posterior sampling via the reverse SDE in Eq.(2), replacing the score term with our decomposed posterior score:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2(\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t))]dt + g(t)d\bar{\mathbf{w}}. \tag{3}$$

While we can train a score network to approximate $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, the time-dependent likelihood $p_t(\mathbf{y}|\mathbf{x}_t)$ is not easy to obtain. In graphical terms, $\mathbf{x}_t \leftarrow \mathbf{x}_0 \rightarrow \mathbf{y}$, but there is no explicit dependence between the measurements $\mathbf{y}$ and the noisy signal $\mathbf{x}_t$.



(a) 2D (b) 3D

Figure 2: **Mean test SSIM [30] for 2D and 3D patterns.** We compare reconstructions with masks learned using our method to those with fixed baseline masks across a range of acceleration factors. The shaded areas indicate a 95% confidence interval. Masks learned with our method consistently lead to better reconstructions than baseline masks for 2D patterns. For 3D patterns, our masks offer competitive performance with baselines.

Diffusion Posterior Sampling (DPS) [8] is a technique that approximates the time-dependent score of the likelihood as $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \simeq \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\hat{\mathbf{x}}_0)$, where $\hat{\mathbf{x}}_0$ is a "one-step" denoised estimate given by Tweedie's formula [12] as the posterior mean

$$\hat{\mathbf{x}}_0 \coloneqq \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t). \tag{4}$$

We can replace the score function in Eq.(4) using our trained score network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to approximate $\hat{\mathbf{x}}_0$. Finally, using the DPS approximation, we can sample from the posterior using the reverse SDE from Eq.(3).

MRI Reconstruction A common approach for accelerating MRI scans is to collect fewer measurements and solve the resulting ill-posed inverse problem. The measurement process for multi-coil MRI can be written in the form

$$\mathbf{y}_i = \mathbf{PFS}_i\mathbf{x}_0 + \epsilon, \tag{5}$$

where $\mathbf{y}_i \in \mathbb{C}^m$ are the measurements in the spatial frequency domain (or *k-space*) for the i^{th} coil, $\mathbf{x}_0 \in \mathbb{C}^n$ is the image of interest, $\mathbf{S}_i \in \mathbb{C}^{n\times n}$ is the coil sensitivity map for the i^{th} coil (c coils in total), $\mathbf{F} \in \mathbb{C}^{n\times n}$ is the Fourier transform matrix, $\mathbf{P} \in \mathbb{C}^{m\times n}$ is a sub-sampling operator whose rows are a subset of the rows of the $n \times n$ identity matrix, and $\epsilon \in \mathbb{C}^m$ is i.i.d Gaussian noise. We also define the *acceleration factor* $R \coloneqq m/n$ as the under-sampling ratio.

3 Methods

We would like to learn a distribution $p_{\theta_\mathcal{A}}(\mathcal{A})$, parameterized by weights $\theta_\mathcal{A}$, over forward operators $\mathcal{A}$ that produce measurements $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \epsilon$. Our goal is to minimize the reconstruction error between the true signal $\mathbf{x}_0$ and the estimates $\tilde{\mathbf{x}}_0 \sim p_0(\mathbf{x}_0|\mathbf{y})$ produced by sampling from the posterior using a diffusion-based generative model. We can write the problem as an optimization that can be solved using gradient descent-based methods:

$$\theta^*_\mathcal{A} = \underset{\theta_\mathcal{A}}{\operatorname{argmin}}\ \mathbb{E}_{\mathbf{x}_0 \sim p_0(\mathbf{x}_0),\, \tilde{\mathbf{x}}_0 \sim p_0(\mathbf{x}_0|\mathbf{y}),\, \mathbf{y} \sim p(\mathbf{y}|\mathbf{x}_0),\, \mathcal{A} \sim p_{\theta_\mathcal{A}}(\mathcal{A})} \|\mathbf{x}_0 - \tilde{\mathbf{x}}_0\|_2^2. \tag{6}$$

In practice, however, sampling from the posterior to get $\tilde{\mathbf{x}}_0$ involves an iterative application of an SDE solver or ancestral sampling. This makes differentiating with respect to $\theta_\mathcal{A}$ challenging, as naïvely backpropagating through the sampling procedure is infeasible due to memory constraints.

To motivate our solution, recall that Tweedie's formula allows us to use Eq. (4) to get a one-step approximation of the denoised posterior mean $\mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)}[\mathbf{x}_0|\mathbf{x}_t]$. We show in Proposition 3.1 that we can extend Tweedie's formula to include measurements $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}_0)$.

Proposition 3.1 (Tweedie's formula with additional measurements)**.** *Let $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$ be an unknown signal, $\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I})$ a version of $\mathbf{x}_0$ corrupted by additive Gaussian noise, and $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}_0)$ some additional measurements of $\mathbf{x}_0$. Furthermore, let $\mathbf{x}_t$ and $\mathbf{y}$ be conditionally*

independent given $\mathbf{x}_0$*:* $p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) = p_t(\mathbf{x}_t|\mathbf{x}_0)$*. Finally, assume that* $p_t(\mathbf{x}_t|\mathbf{y})$ *is supported everywhere. Then, the posterior mean of* $\mathbf{x}_0$ *conditioned on* $\mathbf{x}_t$ *and* $\mathbf{y}$ *is given by*

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}). \tag{7}$$

We give the proof in Appendix A. Recalling that $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$, our result essentially allows us to leverage the score of the likelihood *in addition to* the prior to obtain a finer estimate of $\mathbf{x}_0$ than using the prior alone. The assumption of conditional independence of $\mathbf{x}_t$ and $\mathbf{y}$ given $\mathbf{x}_0$ is satisfied in the inverse problem setting, as we have that $\mathbf{x}_t \leftarrow \mathbf{x}_0 \rightarrow \mathbf{y}$ with no other dependencies.

In reality, we only have access to a score network and an approximation of the likelihood score from DPS. Therefore, a tractable approximation of the expectation in Eq. (7) is

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] \simeq \mathbf{x}_t + \sigma_t^2 [\mathbf{s}_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\hat{\mathbf{x}}_0)], \tag{8}$$

where $\hat{\mathbf{x}}_0$ is given by Tweedie's formula as in Eq. (4). In practice, we assume that $p_t(\mathbf{y}|\hat{\mathbf{x}}_0) \sim \mathcal{N}(\mathbf{y}; \mathcal{A}(\hat{\mathbf{x}}_0), \sigma_t^2 \mathbf{I})$ and calculate the posterior mean as

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] \simeq \mathbf{x}_t + \sigma_t^2 \mathbf{s}_\theta(\mathbf{x}_t, t) - \gamma \nabla_{\mathbf{x}_t} \|\mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y}\|_2^2, \tag{9}$$

where γ is a likelihood step size.

Using the result from Proposition 3.1, we finally present our training objective:

$$\begin{aligned} \theta_{\mathcal{A}}^* &= \underset{\theta_{\mathcal{A}}}{\operatorname{argmin}}\ \mathbb{E}_{\mathbf{x}_0 \sim p_0(\mathbf{x}_0)} \|\mathbf{x}_0 - \tilde{\mathbf{x}}_0\|_2^2 \\ \tilde{\mathbf{x}}_0 &= \mathbb{E}_{\mathcal{A} \sim p_{\theta_{\mathcal{A}}}(\mathcal{A}),\, \mathbf{y} \sim \mathcal{N}(\mathcal{A}(\mathbf{x}_0), \sigma_{\mathbf{y}}^2 \mathbf{I}),\, \mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0),\, t \sim q_t(t)} [\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}], \end{aligned} \tag{10}$$

where $q_t(t)$ is a distribution over time steps and we use the approximation from Eq. (9) to calculate the posterior mean estimate $\tilde{\mathbf{x}}_0$. During training, we calculate the gradient w.r.t $\theta_{\mathcal{A}}$ with automatic differentiation (e.g. using PyTorch [22]), which involves backpropagating through $\tilde{\mathbf{x}}_0$ and therefore through the score network.

4 Experiments

Dataset We experiment on the fastMRI multi-coil brain dataset [31]. We create single-coil minimum variance unbiased estimator (MVUE) images from the fully-sampled k-space data and sensitivity maps (calculated using ESPIRiT [27]) to use as our ground truth reference. For training and testing sub-sampling patterns, we create a subset of 200 training and 50 validation scans from the fastMRI training set and 100 test scans from the fastMRI validation set.

Diffusion Model We train a score-based network using the ADM architecture [10] with the EDM repo [17]. We use the EDM [17] default pre-conditioning and training parameters, with a learning rate of 1×10^{-4}, batch size of 15, and an exponential moving average (ema) half-life of 100K images, and train the models for 140 epochs. For posterior sampling, we modify the the stochastic sampling algorithm from EDM to have no second-order correction and add a log-likelihood step. We present our sampler in Algorithm 1 in Appendix C. Following the implementation of DPS [8], we set the sampling likelihood step size parameter as $\rho_{dps} = \rho / \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)\|$, where ρ is a tuneable hyperparameter. We use 100 sampling steps with $\rho = 10$ and the stochasticity parameter $S_{churn} = 0$.

Learning Sampling Patterns We describe our choice of sampling pattern parameterization in Appendix B. When learning sub-sampling patterns, we use the noise schedule from EDM with $\sigma_t = t$ and noise distribution $q_t(t) = q_{\sigma_t}(\sigma_t)$ such that $\ln \sigma_t \sim \mathcal{N}(P_{mean}, P_{std}^2)$, where we use the default values $P_{mean} = -1.2$ and $P_{std} = 1.2$ [17]. We set the measurement noise $\sigma_{\mathbf{y}} = 0$, training likelihood step size to $\gamma = 1$, training batch size to 1, and Gumbel temperature to $\tau = 1$. We use the Adam optimizer [18] with a learning rate of 1×10^{-2} and optimize for 10 epochs over all 200 training samples. We fix a central 16-pixel wide region that is always fully sampled for calibration purposes on both our masks and baselines, and initialize pattern weights as $\theta_{\mathbf{P}} = \mathbf{0}$.

Results We train 2D and 3D sub-sampling patterns for brain scans across various accelerations. As a baseline, we compare to posterior sampling using fixed equispaced masks for the 2D case and Poisson disc masks for 3D. We display reconstructions in Figure 1. Our masks can reduce small- and large-scale artifacts compared to baseline masks, as shown by the areas indicated with red arrows and insets in Figure 1. We note that our reconstruction at $R = 20$ displays a shading artifact in the top-left, indicating limited quality for 2D patterns at high accelerations regardless of training.

We also evaluate the performance of each method using the structural similarity index measure (SSIM) [30] between the reference MVUE and the reconstruction and plot the results in Figure 2. Our method achieves better reconstruction error for 2D imaging at $R > 4$. Interestingly, both our masks and the Poisson disc masks perform similarly in the 3D case. We note that training a diffusion model end-to-end or fine-tuning a pre-trained model with the objective in Eq. (10) may offer better performance. However, these approaches would be more computationally expensive and couple the diffusion model with the forward operator, decreasing generalization ability.

References

[1] Hemant Kumar Aggarwal and Mathews Jacob. J-MoDL: Joint model-based deep learning for optimized sampling and reconstruction. *IEEE Journal of Selected Topics in Signal Processing*, 14(6):1151–1162, oct 2020.

[2] Cagan Alkan, Morteza Mardani, Shreyas Vasanawala, and John M. Pauly. Learning to sample MRI via variational information maximization. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020.

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

[4] Cagla D. Bahadir, Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Deep-learning-based optimization of the under-sampling pattern in mri. 2019.

[5] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013.

[6] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed sensing using generative models. In *ICML*, 2017.

[7] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

[8] 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*, 2023.

[9] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated mri. *Medical Image Analysis*, 80:102479, 2022.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[11] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[12] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106:1602–1614, 12 2011.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[15] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14938–14954. Curran Associates, Inc., 2021.

[16] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[19] Guanxiong Luo, Martin Heide, and Martin Uecker. Mri reconstruction via data driven markov chain with joint uncertainty estimation. *arXiv preprint arXiv:2202.01479*, 2022.

[20] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[21] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[23] Ferdia Sherry, Martin Benning, Juan Carlos De los Reyes, Martin J Graves, Georg Maierhofer, Guy Williams, Carola-Bibiane Schönlieb, and Matthias J Ehrhardt. Learning the sampling pattern for mri. *IEEE Transactions on Medical Imaging*, 39(12):4310–4321, 2020.

[24] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[25] 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.

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

[27] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. Espirit—an eigenvalue approach to autocalibrating parallel mri: where sense meets grappa. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.

[28] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[29] Guanhua Wang, Tianrui Luo, Jon-Fredrik Nielsen, Douglas C Noll, and Jeffrey A Fessler. B-spline parameterized joint optimization of reconstruction and k-space trajectories (bjork) for accelerated 2d mri. *IEEE Transactions on Medical Imaging*, 41(9):2318–2330, 2022.

[30] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[31] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

[32] Jinwei Zhang, Hang Zhang, Alan Wang, Qihao Zhang, Mert Sabuncu, Pascal Spincemaille, Thanh D. Nguyen, and Yi Wang. Extending loupe for k-space under-sampling pattern optimization in multi-coil mri, 2020.

[33] Marcelo VW Zibetti, Gabor T Herman, and Ravinder R Regatte. Fast data-driven learning of parallel mri sampling patterns for large scale problems. *Scientific Reports*, 11(1):19312, 2021.

A Proof of Proposition 3.1

Proposition 3.1 (Tweedie's formula with additional measurements)**.** *Let $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$ be an unknown signal, $\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I})$ a version of $\mathbf{x}_0$ corrupted by additive Gaussian noise, and $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}_0)$ some additional measurements of $\mathbf{x}_0$. Furthermore, let $\mathbf{x}_t$ and $\mathbf{y}$ be conditionally independent given $\mathbf{x}_0$: $p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) = p_t(\mathbf{x}_t|\mathbf{x}_0)$. Finally, assume that $p_t(\mathbf{x}_t|\mathbf{y})$ is supported everywhere. Then, the posterior mean of $\mathbf{x}_0$ conditioned on $\mathbf{x}_t$ and $\mathbf{y}$ is given by*

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}). \tag{7}$$

Proof. We begin by representing the distribution $p_t(\mathbf{x}_t|\mathbf{y})$ as marginalizing out $\mathbf{x}_0$ conditioned on $\mathbf{y}$:

$$p_t(\mathbf{x}_t|\mathbf{y}) = \int_{\mathbf{x}_0} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) p_0(\mathbf{x}_0|\mathbf{y}) d\mathbf{x}_0.$$

Next, we take the gradient w.r.t. $\mathbf{x}_t$ on both sides:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{y}) &= \nabla_{\mathbf{x}_t} \int_{\mathbf{x}_0} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) p_0(\mathbf{x}_0|\mathbf{y}) d\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) d\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) d\mathbf{x}_0.
\end{aligned}$$

On the last line, we use the identity $\nabla_x \log f(x) = \nabla_x f(x)/f(x)$. We note that since $p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) = p_t(\mathbf{x}_t|\mathbf{x}_0)$, and $p_t(\mathbf{x}_t|\mathbf{x}_0)$ is Gaussian, $p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y})$ has non-zero value everywhere and we avoid singularities from the denominator.

Continuing, we use the conditional independence of $\mathbf{x}_t$ and $\mathbf{y}$ given $\mathbf{x}_0$ to replace $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y})$ on the right-hand side with $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)$ and obtain:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{y}) &= \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \left(\frac{\mathbf{x}_0 - \mathbf{x}_t}{\sigma_t^2} \right) d\mathbf{x}_0.
\end{aligned}$$

Here, we use the fact that $p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I}_n)$ is a Gaussian and replace the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)$ by its exact value, $(\mathbf{x}_0 - \mathbf{x}_t)/\sigma_t^2$.

Expanding the right-hand-side, we get:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{y}) &= \frac{1}{\sigma_t^2} \left[\int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \mathbf{x}_0 d\mathbf{x}_0 - \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \mathbf{x}_t d\mathbf{x}_0 \right] \\
&= \frac{1}{\sigma_t^2} \left[\int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \mathbf{x}_0 d\mathbf{x}_0 - \mathbf{x}_t \int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) d\mathbf{x}_0 \right] \\
&= \frac{1}{\sigma_t^2} \left[\int_{\mathbf{x}_0} p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) \mathbf{x}_0 d\mathbf{x}_0 - \mathbf{x}_t p_t(\mathbf{x}_t|\mathbf{y}) \right].
\end{aligned}$$

In the previous line, we marginalize out $\mathbf{x}_0$ conditioned on $\mathbf{y}$ as in the first line of the proof to recover $p_t(\mathbf{x}_t|\mathbf{y})$.

Next, we observe that Bayes' rule tells us $p_0(\mathbf{x}_0|\mathbf{y}) p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}) = p_t(\mathbf{x}_t|\mathbf{y}) p_0(\mathbf{x}_0|\mathbf{x}_t, \mathbf{y})$ and replace the former quantity by the latter on the right-hand side:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{y}) &= \frac{1}{\sigma_t^2} \left[\int_{\mathbf{x}_0} p_t(\mathbf{x}_t|\mathbf{y}) p_0(\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}) \mathbf{x}_0 d\mathbf{x}_0 - \mathbf{x}_t p_t(\mathbf{x}_t|\mathbf{y}) \right] \\
&= \frac{1}{\sigma_t^2} \left[p_t(\mathbf{x}_t|\mathbf{y}) \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] - \mathbf{x}_t p_t(\mathbf{x}_t|\mathbf{y}) \right] \\
&= \frac{p_t(\mathbf{x}_t|\mathbf{y})}{\sigma_t^2} \left[\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] - \mathbf{x}_t \right] \\
\frac{\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{y})}{p_t(\mathbf{x}_t|\mathbf{y})} &= \frac{1}{\sigma_t^2} \left[\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] - \mathbf{x}_t \right].
\end{aligned}$$

We note that from our assumption, $p_t(\mathbf{x}_t|\mathbf{y})$ is fully supported everywhere, so we avoid singularities when dividing by this quantity.

Finally, we again invoke the identity $\nabla_x \log f(x) = \nabla_x f(x)/f(x)$ to rewrite the left-hand side and rearrange to obtain the desired result:

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) &= \frac{1}{\sigma_t^2}\Big[\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] - \mathbf{x}_t\Big] \\ \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] &= \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}).\end{aligned}$$

□

B Parameterizing the Sampling Pattern

We wish to learn a distribution over optimal sampling pattern for reconstructing MRI images for some fixed acceleration R. This amounts to learning $\mathcal{A}_i(\mathbf{x}) = \mathbf{PFS}_i\mathbf{x}$, for $i \in [c]$ with $\mathbf{F}$ and $\mathbf{S}_i$ fixed. Therefore, we only need to learn a distribution $p_{\theta_\mathbf{P}}$ parameterized by weights $\theta_\mathbf{P}$ over the sub-sampling operator $\mathbf{P}$. We base our parameterization and sampling scheme on that of LOUPE [32], with some changes. For n-dimensional images, we learn parameters $\theta_\mathbf{P} \in \mathbb{R}^n$. Each entry $\theta_{\mathbf{P},i}, i \in [n]$, defines an independent Bernouli random variable $\mathcal{B}(p(\theta_{\mathbf{P},i}))$ at each location in the k-space of an image, where $p(\theta_{\mathbf{P},i})$ is the probability that the variable takes the value 1. The sampling mask distribution is defined as $p_{\theta_\mathbf{P}} = \prod_{i=1}^{n} \mathcal{B}(p(\theta_{\mathbf{P},i}))$.

Following LOUPE, we re-scale the values $\sigma(\theta_{\mathbf{P},i})$, where $\sigma(x) = 1/(1+e^{-x})$ is the standard sigmoid function, so that they have mean $1/R$. Since the Bernouli random variables are independent, sample patterns drawn from the re-scaled distribution will have an average acceleration factor of R. Given the unnormalized mean $\bar{p} = \frac{1}{n}\sum_{i=1}^{n} \sigma(\theta_{\mathbf{P},i})$, we define $p(\theta_{\mathbf{P},i})$ as

$$p(\theta_{\mathbf{P},i}) = \begin{cases} \frac{1}{\bar{p}R}\sigma(\theta_{\mathbf{P},i}) & \text{if } \bar{p} \geq \frac{1}{R} \\ 1 - \frac{R-1}{R-\bar{p}R}(1-\sigma(\theta_{\mathbf{P},i}) & \text{otherwise} \end{cases}, \tag{11}$$

which outputs values in the range $[0, 1]$ and has the desired mean $\frac{1}{n}\sum_{i=1}^{n} p(\theta_{\mathbf{P},i}) = 1/R$.

Next, we need to sample from $p_{\theta_\mathbf{P}}$ in a way that is differentiable with respect to $\theta_\mathbf{P}$. We use the Gumbel Straight-Through estimator [16], which shows superior performance to the vanilla Straight-Through estimator [5] used by LOUPE. For samples g_1 and g_2 drawn i.i.d. from the Gumbel(0, 1) distribution and temperature $\tau > 0$, the Gumbel Straight-Through estimator generates a sample $z_i \in \{0, 1\}$ for k-space location $i \in [n]$ as

$$z_i = \mathbf{1}_{\geq 0.5}(y_i),\ y_i = \frac{u_\tau(p(\theta_{\mathbf{P},i}), g_1)}{u_\tau(p(\theta_{\mathbf{P},i}), g_1) + u_\tau(1 - p(\theta_{\mathbf{P},i}), g_2)},\ u_\tau(p, g) = \exp\left(\frac{\log p + g}{\tau}\right), \tag{12}$$

where $\mathbf{1}_{\geq 0.5}(\cdot)$ is the indicator function that takes the value 1 if its argument is ≥ 0.5 and 0 otherwise. As $\tau \to 0$, we have that $\mathbb{E}[y_i] \to p(\theta_{\mathbf{P},i})$. The Gumbel Straight-Through estimator replaces $\nabla_{\theta_\mathbf{P}} z_i$ with $\nabla_{\theta_\mathbf{P}} y_i$ during backpropagation. This trick allows us to draw realistic binary samples z_i while enjoying well-defined gradients from the smooth softmax sample y_i.

C Posterior Sampling Algorithm

Algorithm 1 Posterior sampling

Require: $\mathbf{s}_\theta(\mathbf{x}_t, t),\ \sigma_{t\in\{t_N,\dots,t_0\}},\ S_{churn}, \rho_{dps}, \mathbf{y}$

sample $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{t_N}^2\mathbf{I})$
for $i \in \{N, \dots, 1\}$ **do**
  sample $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\alpha_i \leftarrow min(S_{churn}/N, \sqrt{2}-1)$
  $\hat{\sigma}_{t_i} \leftarrow \sigma_{t_i} + \alpha_i\sigma_{t_i}$
  $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i + \sqrt{\hat{\sigma}_{t_i}^2 - \sigma_{t_i}^2}\mathbf{z}_i$
  $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_i + \hat{\sigma}_{t_i}^2\mathbf{s}_\theta(\hat{\mathbf{x}}_i, \hat{t}_i)$
  $\hat{\mathbf{x}}'_i \leftarrow \hat{\mathbf{x}}_i + (\hat{\sigma}_{t_i} - \sigma_{t_{i-1}})\mathbf{s}_\theta(\hat{\mathbf{x}}_i, \hat{t}_i)$
  $\mathbf{x}_{i-1} \leftarrow \hat{\mathbf{x}}'_i - \rho_{dps}\nabla_{\hat{\mathbf{x}}_i}\|\mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y}\|_2^2$
return $\mathbf{x}_0$

D Extended Experimental details

D.1 Dataset

All MRI data are initially stored as complex-valued multi-coil k-space measurements. We appropriately crop the raw k-space data and MVUE images for brain scans to 384×384 pixels for all experiments.

For training the score network on brain scans, we take volumes from the fastMRI multi-coil brain training set and remove the last two noisy slices from each volume for a total of 57,297 scans. We use scans from all available contrasts and field strengths for training the score networks. To make the data compatible with real-valued network weights, we represent the complex-valued MVUE images as two-channel, real-valued images.

Since the data exhibit a wide dynamic range of pixel values, we linearly scale images to the range [-1,1] when training the score networks using the minimum and maximum pixel values from the two-channel fully-sampled MVUE images. When learning sampling patterns and performing posterior sampling, we use the minimum and maximum pixel values from the MVUE calculated from the retrospectively-undersampled k-space data to perform this scaling.

For training and evaluating sampling patterns, we use T2-weighted brain scans with a field strength of 3 Teslas. We remove the last five slices from brain volumes when creating the training, validation, and test sets for learning sampling patterns with our method.

D.2 Training Diffusion Models

We train a score-based network using the ADM architecture [10] and the default parameters for ImageNet 256 with some changes. We use 128 base channels instead of 256 and self-attention in the two smallest resolution scales instead of three smallest. The model is trained with classifier-free diffusion guidance [14] with a label dropout probability of 0.1, treating each unique (contrast, field strength) pair as a different class.

D.3 Training Sampling Patterns

When training the sampling patterns using Eq. (10), we use a single forward operator $\mathcal{A}(\mathbf{x}) = \mathbf{PFx}$ instead of using $\mathcal{A}_i(\mathbf{x}) = \mathbf{PFS}_i\mathbf{x}$ for each of the c coils with $i \in [c]$. In other words, we do not use the coil sensitivity maps $\mathbf{C}_i$ when retrospectively sub-sampling training data to create measurements. We find that this method accelerates training by avoiding memory and computational costs for multiple coils. We also find that this method improves test performance and leads to more stable convergence of the learned sampling patterns. During validation and testing, we still perform reconstructions using undersampled multi-coil k-space data as measurements.

During validation, we perform posterior sampling to reconstruct each image in the validation set and track the error between the reconstructions and fully-sampled images. Before testing, we restore the weights of the sampling operator from the iteration with the lowest mean validation error.

D.4 Hyperparameters

We tune the following hyperparameters: the training likelihood step size γ, Gumbel temperature τ, learning rate for sampling patterns, sampling likelihood step size ρ, and sampling stochasticity parameter S_{churn}.

For the general experiments in Figures 1 and 2, we fix the number of sampling steps to 100 and search for the values of $\rho \in \{0.1, 0.2, \ldots, 1, 2, \ldots, 10\}$ and $S_{churn} \in \{0, 10, \ldots, 50\}$. We choose the pair of values with the smallest reconstruction error on a hold-out set of 30 scans. Once we find the optimal values $\rho = 10$ and $S_{churn} = 0$, we fix them and search for values of $\gamma \in \{0.1, 0.5, 1, 5, 10\}$ and $\tau \in \{0.1, 0.5, 1\}$. We train sampling patterns for 2 epochs on our training set using a learning rate of 0.1 and choose the pair $\gamma = 1$ and $\tau = 1$ that gives the best reconstruction error on the 30 hold-out scans using the learned pattern. Finally, we fix all previous tuned hyperparameters and search for the learning rate in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ by again training for 2 epochs and reconstructing hold-out images, finding 10^{-2} to be optimal. We tune all listed hyperparameters using 2D patterns with $R = 4$ and use the same values across all pattern types and accelerations.