Mask-Agnostic Posterior Sampling MRI via Conditional GANs with Guided Reconstruction

Matthew C. Bendel
Dept. ECE
The Ohio State University
Columbus, OH 43210
bendel.8@osu.edu

Rizwan Ahmad
Dept. BME
The Ohio State University
Columbus, OH 43210
ahmad.46@osu.edu

Philip Schniter
Dept. ECE
The Ohio State University
Columbus, OH 43201
schniter.1@osu.edu

Abstract

For accelerated magnetic resonance imaging (MRI), conditional generative adversarial networks (cGANs), when trained end-to-end with a fixed subsampling mask, have been shown to compete with contemporary diffusion-based techniques while generating samples thousands of times faster. To handle unseen sampling masks at inference, we propose "guided reconstruction" (GR), wherein the cGAN code vectors are projected onto the measurement subspace. Using fastMRI brain data, we demonstrate that GR allows a cGAN to successfully handle changes in sampling mask, as well as changes in acceleration rate, yielding faster and more accurate recoveries than the Langevin approach from (Jalal et al., 2021) and the DDRM diffusion approach from (Kawar et al., 2022). Our code will be made available at `https://github.com/matt-bendel/rcGAN-agnostic`.

1 Introduction

We consider multi-coil accelerated magnetic resonance imaging (MRI), where the goal is to recover the N-pixel MR image $\boldsymbol{x} \in \mathbb{C}^N$ from multi-coil k-space measurements $\boldsymbol{y}$, where [1]

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \text{ with } \boldsymbol{A} = \begin{bmatrix} \boldsymbol{M}\boldsymbol{F}\boldsymbol{S}_1 \\ \vdots \\ \boldsymbol{M}\boldsymbol{F}\boldsymbol{S}_C \end{bmatrix}. \tag{1}$$

In (1), $\boldsymbol{M} \in \mathbb{R}^{M\times N}$ is a subsampling operator containing rows from the $N \times N$ identity matrix, $\boldsymbol{F} \in \mathbb{C}^{N\times N}$ is the unitary 2D discrete Fourier transform, $\boldsymbol{S}_c \in \mathbb{C}^{N\times N}$ is a diagonal matrix containing the sensitivity map of the cth coil (estimated via ESPIRiT [2]), C is the number of coils, and $\boldsymbol{w} \in \mathbb{C}^{MC}$ is noise. The ratio $R \triangleq N/M$ is known as the acceleration rate.

For MRI recovery, classical methods like SENSE [1] or GRAPPA [3] work well at low acceleration rates. To deal with higher R, compressed-sensing (CS) algorithms exploit image sparsity in a transformed domain [4]. Plug-and-play methods offer improved performance by replacing the proximal step in CS algorithms with a sophisticated image denoiser [5]. More recently, deep neural networks have been employed, either as denoisers in PnP algorithms [6], unrolled versions of those algorithms [7, 8], or as direct-estimation networks [9, 10].

The aforementioned MRI recovery techniques return "point estimates," i.e., a single best estimate of $\boldsymbol{x}$ from $\boldsymbol{y}$. A fundamental challenge with point estimates is the *perception-distortion tradeoff* [11], which says that the estimate can have either low distortion (i.e., $\widehat{\boldsymbol{x}}$ close to $\boldsymbol{x}$) or high perceptual quality (i.e., $\widehat{\boldsymbol{x}}$ close to the manifold of clean images), but not both. Consequently, neural networks trained to minimize distortion metrics like L2 or L1 loss give blurry images, while those trained to give realistic, sharp images (e.g., using adversarial loss) tend to hallucinate. Another fundamental shortcoming of point estimators is that they do not acknowledge or report reconstruction uncertainty [12].

Posterior sampling. To address the aforementioned limitations of point estimation, *posterior sampling* has been proposed. There, the goal is to generate samples $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^P$ from the posterior distribution $p_{\mathsf{x}|\mathsf{y}}(\cdot|\boldsymbol{y})$. Posterior sampling facilitates various forms of uncertainty quantification, e.g., pixel-wise standard-deviation maps or confidence bounds on the results of downstream tasks like classification or segmentation [13, 14]. Furthermore, when there is a need to construct a single "best" estimate $\widehat{\boldsymbol{x}}$, the posterior sample-average $\widehat{\boldsymbol{x}}_{(P)} \triangleq \frac{1}{P}\sum_{i=1}^P \widehat{\boldsymbol{x}}_i$ allows one to straightforwardly navigate the perception-distortion tradeoff. For instance, as P increases, $\widehat{\boldsymbol{x}}_{(P)}$ has lower distortion (with $P \to \infty$ yielding the minimum mean-squared error estimate) but lower perceptual quality. Posterior sampling is also useful for adaptive acquisition [15] and counterfactual diagnosis [16].

There are several ways to design posterior samplers for imaging inverse problems. Methods like conditional generative adversarial networks (cGANs) [17–20], conditional variational auto-encoders (cVAEs) [21–23], and conditional normalizing flows (cNFs) [24–27] train networks in an end-to-end (E2E) manner to map random code vectors to posterior samples. Once trained, these approaches generate samples rapidly. However, they are usually trained on a specific forward model $\boldsymbol{A}$ (e.g., a specific MRI sampling mask), in which case they don't generalize well to other forward models. Langevin/diffusion methods [28–33] avoid these generalization issues by training a denoiser and generating posterior samples using an iterative algorithm that invokes the forward model $\boldsymbol{A}$. However, they tend to be orders-of-magnitude slower at sample generation than cGANs, cVAEs, or cNFs.

Our contribution. In this work, we focus on cGANs, which have recently been shown to yield fast and accurate posterior sampling in accelerated multicoil MRI [20]. Although one could train a cGAN (or any E2E method) to perform well *on average* over a wide range of sampling masks, doing so degrades performance on any fixed mask (see, e.g., [5, Table 2]). Thus, we propose a novel approach to agnostically training cGANs that we dub "guided reconstruction" (GR). When used in conjunction with the data-consistency procedure from [34], we find that GR allows the cGAN to generalize to a wide range of masks and sampling rates R without significant loss in performance. In particular, we demonstrate speed and accuracy improvements over recent Langevin/diffusion methods on the recovery of fastMRI [10] brain data.

2 Proposed method

Let $G_{\boldsymbol{\theta}}(\boldsymbol{A}^+\boldsymbol{y}, \cdot)$ be a conditional generator with parameters $\boldsymbol{\theta}$ that maps random code vectors $\boldsymbol{z}_i$ to samples $\widehat{\boldsymbol{x}}_i$ of the posterior $p_{\mathsf{x}|\mathsf{y}}(\cdot|\boldsymbol{y})$. Here, $\boldsymbol{A}^+$ denotes the pseudo-inverse of $\boldsymbol{A}$, and so the generator is given aliased image-space data $\boldsymbol{A}^+\boldsymbol{y}$, rather than measurement-space data $\boldsymbol{y}$, as this is standard practice for end-to-end reconstruction networks (see, e.g., the UNet in [10]). Note that, for $\boldsymbol{A}$ in (1) with ESPIRiT-estimated coil maps $\{\boldsymbol{S}_c\}$, we have that $\boldsymbol{A}^+ = \boldsymbol{A}^{\mathsf{H}}$ [2].

Guided reconstruction. Typically, the generator is fed with codes $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. To provide side-information on the $\boldsymbol{A}$-specific aliasing pattern in $\boldsymbol{A}^+\boldsymbol{y}$, we propose to project $\boldsymbol{z}_i$ onto the measurement subspace via

$$\boldsymbol{v}_i = \boldsymbol{A}^+\boldsymbol{A}\boldsymbol{z}_i, \tag{2}$$

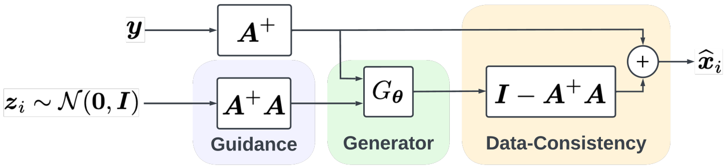
and feed $\boldsymbol{v}_i$ to the generator instead of $\boldsymbol{z}_i$. Then we train the generator using many random realizations of $\boldsymbol{A}$, as well as many $(\boldsymbol{x}, \boldsymbol{y})$ pairs, so that it learns to associate the statistics of the modified code vectors $\boldsymbol{v}_i$ with the statistics of the aliasing in $\boldsymbol{A}^+\boldsymbol{y}$. To our knowledge, using (2) in a cGAN is novel.

Data-consistency. In addition, we employ the data-consistency (DC) procedure proposed in [34]:

$$\widehat{\boldsymbol{x}}_i = (\boldsymbol{I} - \boldsymbol{A}^+\boldsymbol{A})\widehat{\boldsymbol{x}}_i^{\mathsf{raw}} + \boldsymbol{A}^+\boldsymbol{y}, \tag{3}$$

where $\widehat{\boldsymbol{x}}_i^{\mathsf{raw}}$ is the raw generator output. This ensures that the generated samples are consistent with the measurements in the sense that $\boldsymbol{A}\widehat{\boldsymbol{x}}_i = \boldsymbol{y}$. The DC approach (3) is recommended only for low-noise settings, since it makes no attempt to remove measurement noise. In high-noise settings, the dual-decomposition approach from [35] could be used instead.

A diagram of our proposed approach is provided in Fig. 1.



$\boldsymbol{y}$
$\boldsymbol{A}^+$
$\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\boldsymbol{A}^+\boldsymbol{A}$
$G_{\boldsymbol{\theta}}$
$\boldsymbol{I} - \boldsymbol{A}^+\boldsymbol{A}$
$+$
$\widehat{\boldsymbol{x}}_i$
Guidance
Generator
Data-Consistency

Figure 1: The proposed approach.

Table 1: Average PSNR, SSIM, LPIPS, and DISTS vs. R. **Bold** indicates the best score in each column and *italics* the second-best. (PE) denotes a point-estimator and (PS) denotes a posterior sampler. Time was measured on a server with 4 NVIDIA A100 GPUs, each with 82 GB RAM.

	PSNR↑		SSIM↑		LPIPS↓		DISTS↓		CFID↓		FID↓		
Model	R=4	R=8	R=4	R=8	R=4	R=8	R=4	R=8	R=4	R=8	R=4	R=8	Time (4)↓
E2E-VarNet [8] (PE)	**38.93**	**35.34**	**0.9629**	**0.9450**	0.0344	0.0602	0.0908	0.1227	7.47	8.89	8.84	*13.96*	*316ms*
Langevin [29] (PS)	38.27	34.17	0.9292	0.8958	*0.0316*	*0.0572*	*0.0777*	*0.1120*	*5.29*	*7.34*	*6.12*	14.32	14 min
DDRM [33] (PS)	34.97	32.48	0.9289	0.9046	0.0609	0.0943	0.1193	0.1392	16.88	18.99	22.91	24.59	11s
Ours (PS)	*38.58*	*35.21*	*0.9510*	*0.9254*	**0.0216**	**0.0354**	**0.0583**	**0.0807**	**2.13**	**3.92**	**2.57**	**7.33**	**217ms**

3 Experiments

Generator. We use the MRI architectures from the rcGAN paper [20], leveraging the authors' code implementation [36] but doubling the number of channels in the generator.

Data. We use the first 8 slices of all fastMRI [10] T2 brain training volumes with at least 8 coils, cropping to 384×384 pixels and compressing to 8 virtual coils [37]. We apply the same procedure to the fastMRI T2 brain validation volumes, yielding 12 200 training, 2 376 testing, and 784 validation images. From the 2 376 testing images, we randomly select 72 on which to compute metrics due to computational limits imposed by the long sample-generation time of the Langevin method [29].

Training/Validation. For our cGAN, we use the regularized adversarial training approach from [20]. For training and validation, we use randomly generated vertical-line subsampling masks with acceleration R uniformly distributed in $\{2, 3, 4, 5, 6, 7, 8\}$. Further details are given in Appendix A.

Testing. When testing, we use the GRO [38] sampling mask at specific R values. Multi-coil outputs are combined using SENSE-based coil combining and converted to magnitude images for performance evaluation. For the posterior samplers, we leverage the perception-distortion tradeoff, choosing the $P \in \{1, 2, 4, 8, 16, 32\}$ for which $\widehat{\boldsymbol{x}}_{(P)}$ has the best performance for the given metric. A comprehensive table of performance metrics versus P and R can be found in Appendix B.

Competitors. As a baseline, we consider the state-of-the-art point estimator E2E-VarNet [8]. For competing posterior samplers, we consider the Langevin approach from [29] and the DDRM diffusion approach from [33]. As suggested by the DDRM paper [33], we used $\eta = 0.85$, and as suggested by the DDRM application to fastMRI knee data in [39], we used 100 steps and the "oracle" DDPM denoiser from [40]. We trained DDPM on the fastMRI brain data using advice gained from personal communication with the authors of [39]. Further details are given in Appendix A.

Main results. Table 1 shows PSNR, SSIM [41], LPIPS [42], DISTS [43], CFID [44], FID [45], and the generation time for 4 posterior samples or one E2E-VarNet sample at $R = 4$. CFID is the only metric that aims to directly measure posterior-sampling performance, and we computed it using VGG-16 (not Inception-v3) to better align with radiologists' perceptions [46]. Our proposed cGAN performed best in all metrics but PSNR and SSIM, where it was second-best. Also, our cGAN is 3 800 times faster than the Langevin approach and 50 times faster than DDRM. We note that DDRM performs relatively poorly, which we conjecture is due to the DDPM denoiser we trained. Although we spent significant effort tuning DDPM (see Appendix A), the authors of [39] warned us (in an email) that it is "very sensitive to training hyperparameters."

Fig. 2 shows examples of $\widehat{\boldsymbol{x}}_{(P)}$, along with the corresponding pixel-wise absolute errors $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$ and pixel-wise standard deviation (SD) $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_{(P)} - \widehat{\boldsymbol{x}}_i)^2)^{1/2}$ at $P = 4$ and $R = 4$. The error images of the proposed approach look similar to those of the E2E-VarNet and quite different from those of the Langevin/diffusion techniques. Relative to the proposed approach, the pixel-wise SD maps show no variability for the E2E-VarNet (which is expected as it is a point estimator), lower

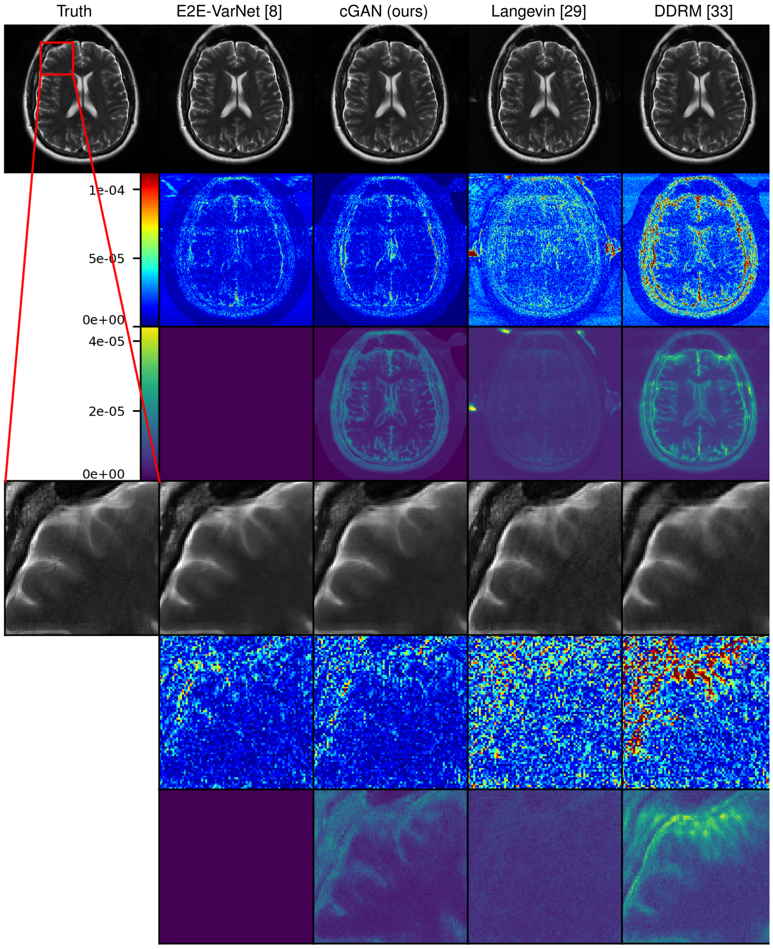


Figure 2: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.

Table 2: Ablation study on the effect of GR and DC. **Bold** indicates the best score in each column.

	PSNR↑				SSIM↑				LPIPS↓				DISTS↓			
Model	R=2	R=4	R=6	R=8	R=2	R=4	R=6	R=8	R=2	R=4	R=6	R=8	R=2	R=4	R=6	R=8
Ours w/o GR, w/o DC	37.86	35.37	33.19	32.87	0.9433	0.9194	0.9006	0.8915	0.0368	0.0610	0.0713	0.0782	0.0948	0.1249	0.1310	0.1397
Ours w/o GR	42.46	37.79	35.87	34.66	0.9741	0.9488	0.9340	0.9230	0.0121	0.0366	0.0444	0.0579	0.0485	0.0931	0.1044	0.1171
Ours w/o DC	38.70	36.08	33.80	33.33	0.9537	0.9295	0.9146	0.9003	0.0346	0.0539	0.0649	0.0721	0.0929	0.1219	0.1255	0.1359
Ours	**43.01**	**38.58**	**36.41**	**35.21**	**0.9748**	**0.9507**	**0.9360**	**0.9249**	**0.0110**	**0.0310**	**0.0413**	**0.0517**	**0.0475**	**0.0891**	**0.1017**	**0.1137**

variability and strange hot-spots for the Langevin method (also seen in [20]), and higher variability for DDRM (expected due to its larger estimation error). Additional examples are given in Appendix C.

Ablation study. Table 2 shows the performance of the proposed approach with and without GR and DC at $P = 32$. There we see that both GR and DC are important ingredients of the proposed method.

4 Conclusion

We proposed a "guided reconstruction" technique that helps to make posterior-sampling cGANs agnostic to the MRI mask and acceleration factor. Experiments with fastMRI brain data show our method outperforming the Langevin method from [29] and the DDRM diffusion method from [33] in all accuracy/perceptual metrics while generating samples orders-of-magnitude faster.

Limitations. Although the proposed approach helps to make the cGAN agnostic to the forward model, it still requires that the cGAN is trained over a representative range of forward models and may not generalize well to forward models outside that range.

Acknowledgments and Disclosure of Funding

The authors are funded in part by the National Institutes of Health under grant R01-EB029957 and the National Science Foundation under grant CCF-1955587.

References

[1] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: Sensitivity encoding for fast MRI," *Magn. Reson. Med.*, vol. 42, no. 5, pp. 952–962, 1999. 1

[2] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala, and M. Lustig, "ESPIRiT–an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA," *Magn. Reson. Med.*, vol. 71, no. 3, pp. 990–1001, 2014. 1, 2

[3] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magn. Reson. Med.*, vol. 47, no. 6, p. 1202–1210, 2002. 1

[4] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007. 1

[5] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug and play methods for magnetic resonance imaging," *IEEE Signal Process. Mag.*, vol. 37, pp. 105–116, Mar. 2020. 1, 2

[6] T. Meinhardt, M. Möller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1781–1790, 2017. 1

[7] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, vol. 79, no. 6, pp. 3055–3071, 2018. 1

[8] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson, "End-to-end variational networks for accelerated MRI reconstruction," in *Proc. Intl. Conf. Med. Image Comput. Comput. Assist. Intervent.*, pp. 64–73, 2020. 1, 3, 4, 8, 9, 10, 11, 12, 13, 14

[9] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, pp. 4509–4522, Sept. 2017. 1

[10] J. Zbontar, F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. Rabbat, P. Vincent, J. Pinkerton, D. Wang, N. Yakubova, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, "fastMRI: An open dataset and benchmarks for accelerated MRI," *arXiv:1811.08839*, 2018. 1, 2, 3

[11] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 6228–6237, 2018. 1

[12] E. Begoli, T. Bhattacharya, and D. Kusnezov, "The need for uncertainty quantification in machine-assisted medical decision making," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 20–23, 2019. 1

[13] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur, "Confidence calibration and predictive uncertainty estimation for deep medical image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 12, pp. 3868–3878, 2020. 2

[14] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021. 2

[15] T. Sanchez, I. Krawczuk, Z. Sun, and V. Cevher, "Uncertainty-driven adaptive sampling via GANs," in *Proc. Neural Inf. Process. Syst. Workshop*, 2020. 2

[16] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, "Explaining image classifiers by counterfactual generation," in *Proc. Int. Conf. on Learn. Rep.*, 2019. 2

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 1125–1134, 2017. 2

[18] H. Zhao, H. Li, S. Maurer-Stroh, and L. Cheng, "Synthesizing retinal and neuronal images with generative adversarial nets," *Med. Image Analysis*, vol. 49, 07 2018. 2

[19] J. Adler and O. Öktem, “Deep Bayesian inversion,” *arXiv:1811.05910*, 2018. 2

[20] M. Bendel, R. Ahmad, and P. Schniter, “A regularized conditional GAN for posterior sampling in inverse problems,” *arXiv:2210.13389*, 2022. 2, 3, 4, 8

[21] V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly, “Uncertainty quantification in deep MRI reconstruction,” *IEEE Trans. Med. Imag.*, vol. 40, pp. 239–250, Jan. 2021. 2

[22] F. Tonolini, J. Radford, A. Turpin, D. Faccio, and R. Murray-Smith, “Variational inference for computational imaging inverse problems,” *J. Mach. Learn. Res.*, vol. 21, no. 179, pp. 1–46, 2020. 2

[23] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Proc. Neural Inf. Process. Syst. Conf.*, 2015. 2

[24] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, “Guided image generation with conditional invertible neural networks,” *arXiv:1907.02392*, 2019. 2

[25] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, “Learning likelihoods with conditional normalizing flows,” *arXiv preprint arXiv:1912.00042*, 2019. 2

[26] H. Sun and K. L. Bouman, “Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging,” in *Proc. AAAI Conf. Artificial Intell.*, vol. 35, pp. 2628–2637, 2021. 2

[27] J. Wen, R. Ahmad, and P. Schniter, “A conditional normalizing flow for accelerated multi-coil MR imaging,” in *Proc. Int. Conf. Mach. Learn.*, 2023. 2

[28] Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” in *Proc. Neural Inf. Process. Syst. Conf.*, 2020. 2

[29] A. Jalal, M. Arvinte, G. Daras, E. Price, A. Dimakis, and J. Tamir, “Robust compressed sensing MRI with deep generative priors,” in *Proc. Neural Inf. Process. Syst. Conf.*, 2021. 2, 3, 4, 8, 9, 10, 11, 12, 13, 14

[30] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *Proc. Int. Conf. on Learn. Rep.*, 2021. 2

[31] Y. Song, L. Shen, L. Xing, and S. Ermon, “Solving inverse problems in medical imaging with score-based generative models,” in *Proc. Int. Conf. on Learn. Rep.*, 2022. 2

[32] H. Chung and J. C. Ye, “Score-based diffusion models for accelerated MRI,” *Med. Image Analysis*, vol. 80, p. 102479, 2022. 2

[33] B. Kawar, M. Elad, S. Ermon, and J. Song, “Denoising diffusion restoration models,” in *Proc. Neural Inf. Process. Syst. Conf.*, 2022. 2, 3, 4, 8, 9, 10, 11, 12, 13, 14

[34] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, “Amortised MAP inference for image super-resolution,” in *Proc. Int. Conf. on Learn. Rep.*, 2017. 2

[35] D. Chen and M. E. Davies, “Deep decomposition learning for inverse imaging problems,” in *Proc. European Conf. Comp. Vision*, pp. 510–526, 2020. 2

[36] M. Bendel, R. Ahmad, and P. Schniter, “A regularized conditional GAN for posterior sampling in inverse problems.” Downloaded from `https://github.com/matt-bendel/rcGAN`, May 2023. 3, 8

[37] T. Zhang, J. M. Pauly, S. S. Vasanawala, and M. Lustig, “Coil compression for accelerated imaging with Cartesian sampling,” *Magn. Reson. Med.*, vol. 69, no. 2, pp. 571–582, 2013. 3

[38] M. Joshi, A. Pruitt, C. Chen, Y. Liu, and R. Ahmad, “Technical report (v1.0)–pseudo-random cartesian sampling for dynamic MRI,” *arXiv:2206.03630*, 2022. 3, 8

[39] B. Kawar, N. Elata, T. Michaeli, and M. Elad, “GSURE-based diffusion model training with corrupted data,” *arXiv:2305.13128*, 2023. 3, 8

[40] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. Neural Inf. Process. Syst. Conf.*, vol. 33, pp. 6840–6851, 2020. 3, 8

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, Apr. 2004. 3

[42] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 586–595, 2018. 3

[43] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, “Image quality assessment: Unifying structure and texture similarity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, 2020. 3

[44] M. Soloveitchik, T. Diskin, E. Morin, and A. Wiesel, “Conditional Frechet inception distance,” *arXiv:2103.11521*, 2021. 3

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in *Proc. Neural Inf. Process. Syst. Conf.*, vol. 30, 2017. 3

[46] S. Kastryulin, J. Zakirov, N. Pezzotti, and D. V. Dylov, "Image quality assessment for magnetic resonance imaging," *arXiv:2203.07809*, 2022. 3

[47] A. Jalal, M. Arvinte, G. Daras, E. Price, A. Dimakis, and J. Tamir, "csgm-mri-langevin." `https://github.com/utcsilab/csgm-mri-langevin`, 2021. Accessed: 2021-12-05. 8

[48] B. Kawar, N. Elata, T. Michaeli, and M. Elad, "GSURE-based diffusion model training with corrupted data." Downloaded from `https://github.com/bahjat-kawar/gsure-diffusion`, May 2023. 8

[49] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models." Downloaded from `https://github.com/bahjat-kawar/ddrm`, May 2022. 8

Supplementary Materials

A Implementation details

A.1 Masks

We use vertical-line subsampling masks in all experiments. To facilitate ESPIRiT-based coil sensitivity estimation, these masks include an auto-calibration signal (ACS) region that takes the form of an N_{ACS}-wide collection of vertical lines in the center of k-space. At acceleration $R = 2$ we use $N_{\mathsf{ACS}} = 32$, at acceleration $R = 8$ we use $N_{\mathsf{ACS}} = 16$, and at any other acceleration, we choose N_{ACS} by linearly extrapolating these values.

For training and validation, we use *randomly* generated vertical-line subsampling masks, where the acceleration R is uniformly distributed in $\{2, 3, 4, 5, 6, 7, 8\}$ and, for a given R, the lines are uniformly distributed over the non-ACS region. Each batch element gets a different mask realization. For testing, we use *(deterministic) GRO* [38] sampling masks, with the same ACS region, at fixed values of R. With very high probability, these GRO masks are unseen during training/validation.

A.2 Models

cGAN. For our proposed method, we build on the regularized cGAN from [20] and the authors' implementation in [36]. This model, dubbed "rcGAN," was shown to produce state-of-the-art MRI posterior samples, outperforming several existing cGANs, as well as the score-based diffusion technique from [29], in all accuracy metrics, while generating samples thousands of times faster than [29].

For our implementation, we use the same UNet generator from [36], but we double the number of channels to increase network capacity, since the network has a harder problem to solve. Also, we use the same discriminator and training hyperparameters as [36], except that we reduce the batch size from 32 to 20. Running PyTorch on a server with 4 Tesla A100 GPUs, each with 82 GB of memory, the training of our cGAN takes approximately 2 days.

E2E-VarNet. We use the E2E-VarNet [8] to exemplify a state-of-the-art point estimator. For this, we use the same E2E-VarNet hyperparameter settings that were used in [29], including the use of the SENSE-based coil-combined images as ground-truth when training. Like our cGAN, the E2E-VarNet is trained with random masks as described above. However, it is trained to minimize a weighted combination of L1 and (negative) SSIM loss, as described in [8].

Langevin. For the Langevin approach to MRI from [29], we use the authors' implementation from [47] but with the sampling mask changed to the GRO mask. We generated $P = 32$ samples for each of the 72 different test images using a batch-size of 4, which took approximately 6 days for each tested acceleration rate R on a server with 4 NVIDIA V100 GPUs, each with 32 GB of memory. The reconstruction times in Table 1, however, correspond to our A100 server, for fair comparison to the other methods.

DDRM. To configure DDRM [33], we take guidance from [39] and the accompanying codebase [48], where DDRM was used to recover fastMRI knee data. Using minor changes to the code from [48], we first trained a DDPM denoiser [40] for our fastMRI T2 brain data using the suggested hyperparameters for the "oracle" model. We then found that the DDPM performance could be significantly improved by reducing the batch size to 20 and increasing the training time to 400 epochs (or 262k iterations), which took about 3 days on our A100 server. Once DDPM was trained, we integrated it into the DDRM code from [49]. This required making only a small modification to the inpainting forward model, as confirmed through emails with the authors. Beyond this, we used the default DDRM hyperparameters from [33] (e.g., $\eta = 0.85$) and we ran DDRM for 100 time steps to be consistent with the fastMRI knee implementation in [39]. Using this DDRM setup, we generated $P = 32$ samples for each of the 72 different test images using a batch-size of 4, which took roughly 2 hours for each tested acceleration rate R on our A100 server.

B Additional results

Table B.1: Average PSNR, SSIM, LPIPS, and DISTS of $\widehat{\boldsymbol{x}}_{(P)}$ versus number of posterior samples P for $R \in \{2, 4, 6, 8\}$-accelerated fastMRI T2 brain recovery. The point estimators produce only $P = 1$ sample. The UNet has the same architecture as our cGAN generator but is a point estimator trained with the same L1+SSIM loss as the E2E-VarNet. **Bold** indicates the best score for each metric, for each R, across all P.

		PSNR↑						SSIM↑					
Accel	Model	P=1	P=2	P=4	P=8	P=16	P=32	P=1	P=2	P=4	P=8	P=16	P=32
$R = 2$	E2E-VarNet [8]	41.77	-	-	-	-	-	0.9646	-	-	-	-	-
	UNet	42.49	-	-	-	-	-	**0.9755**	-	-	-	-	-
	Ours w/o GR, w/o DC	34.96	36.22	37.03	37.48	37.73	37.86	0.9301	0.9412	0.9442	0.9443	0.9438	0.9433
	Ours w/o GR	39.81	40.99	41.72	42.13	42.35	42.46	0.9660	0.9713	0.9732	0.9738	0.9741	0.9741
	Ours w/o DC	35.14	36.64	37.60	38.22	38.54	38.70	0.9343	0.9467	0.9513	0.9530	0.9536	0.9537
	Ours	40.17	41.42	42.20	42.64	42.89	**43.01**	0.9671	0.9721	0.9738	0.9744	0.9747	0.9748
	DDRM [33]	36.23	37.31	37.97	38.34	38.55	38.65	0.9277	0.9416	0.9488	0.9525	0.9544	0.9554
$R = 4$	E2E-VarNet [8]	**38.93**	-	-	-	-	-	**0.9629**	-	-	-	-	-
	UNet	38.27	-	-	-	-	-	0.9542	-	-	-	-	-
	Ours w/o GR, w/o DC	33.19	34.18	34.78	35.10	35.28	35.37	0.9114	0.9216	0.9233	0.9219	0.9205	0.9194
	Ours w/o GR	35.51	36.56	37.17	37.52	37.70	37.79	0.9379	0.9464	0.9488	0.9491	0.9490	0.9488
	Ours w/o DC	33.44	34.63	35.32	35.75	35.98	36.08	0.9170	0.9288	0.9313	0.9311	0.9303	0.9295
	Ours	35.82	37.05	37.81	38.23	38.46	38.58	0.9391	0.9481	0.9506	0.9510	0.9509	0.9507
	Langevin [29]	36.04	37.02	37.65	37.99	38.17	38.27	0.8989	0.9138	0.9218	0.9260	0.9281	0.9292
	DDRM [33]	32.62	33.68	34.32	34.68	34.87	34.97	0.8864	0.9073	0.9186	0.9244	0.9274	0.9289
$R = 6$	E2E-VarNet [8]	**36.71**	-	-	-	-	-	**0.9531**	-	-	-	-	-
	UNet	35.74	-	-	-	-	-	0.9366	-	-	-	-	-
	Ours w/o GR, w/o DC	30.45	31.66	32.41	32.84	33.06	33.19	0.8770	0.8978	0.9039	0.9038	0.9020	0.9006
	Ours w/o GR	33.15	34.34	35.09	35.52	35.75	35.87	0.9160	0.9295	0.9336	0.9344	0.9342	0.9340
	Ours w/o DC	30.63	31.98	32.87	33.38	33.67	33.80	0.8767	0.9007	0.9107	0.9139	0.9145	0.9146
	Ours	33.61	34.84	35.61	36.05	36.29	36.41	0.9187	0.9314	0.9354	0.9362	0.9361	0.9360
	DDRM [33]	31.16	32.30	33.02	33.43	33.65	33.76	0.8675	0.8922	0.9056	0.9127	0.9163	0.9182
$R = 8$	E2E-VarNet [8]	**35.34**	-	-	-	-	-	**0.9450**	-	-	-	-	-
	UNet	34.70	-	-	-	-	-	0.9257	-	-	-	-	-
	Ours w/o GR, w/o DC	30.14	31.35	32.10	32.54	32.75	32.87	0.8721	0.8920	0.8970	0.8961	0.8935	0.8915
	Ours w/o GR	32.27	33.35	34.01	34.36	34.56	34.66	0.9052	0.9191	0.9232	0.9237	0.9234	0.9230
	Ours w/o DC	30.30	31.62	32.44	32.94	33.19	33.33	0.8769	0.8967	0.9023	0.9024	0.9013	0.9003
	Ours	32.54	33.72	34.45	34.87	35.10	35.21	0.9068	0.9206	0.9247	0.9254	0.9252	0.9249
	Langevin [29]	32.42	33.08	33.71	34.01	34.10	34.17	0.8607	0.8772	0.8866	0.8918	0.8943	0.8958
	DDRM [33]	29.17	30.51	31.41	31.94	32.23	32.38	0.8410	0.8704	0.8877	0.8972	0.9021	0.9046

		LPIPS↓						DISTS↓					
Accel	Model	P=1	P=2	P=4	P=8	P=16	P=32	P=1	P=2	P=4	P=8	P=16	P=32
$R = 2$	E2E-VarNet [8]	0.0171	-	-	-	-	-	0.0564	-	-	-	-	-
	UNet	0.0122	-	-	-	-	-	0.0504	-	-	-	-	-
	Ours w/o GR, w/o DC	0.0289	0.0267	0.0293	0.0329	0.0353	0.0368	0.0859	0.0787	0.0821	0.0878	0.0918	0.0948
	Ours w/o GR	0.0105	0.0099	0.0106	0.0113	0.0119	0.0121	0.0386	0.0375	0.0418	0.0453	0.0474	0.0485
	Ours w/o DC	0.0296	0.0267	0.0286	0.0313	0.0333	0.0346	0.0861	0.0799	0.0819	0.0866	0.0907	0.0929
	Ours	0.0095	**0.0093**	0.0099	0.0105	0.0108	0.0110	**0.0355**	0.0369	0.0410	0.0444	0.0466	0.0475
	DDRM [33]	0.0401	0.0304	0.0288	0.0295	0.0304	0.0310	0.1014	0.0847	0.0791	0.0790	0.0801	0.0811
$R = 4$	E2E-VarNet [8]	0.0344	-	-	-	-	-	0.0908	-	-	-	-	-
	UNet	0.0323	-	-	-	-	-	0.0892	-	-	-	-	-
	Ours w/o GR, w/o DC	0.0465	0.0457	0.0503	0.0556	0.0588	0.0610	0.1055	0.1031	0.1086	0.1158	0.1210	0.1249
	Ours w/o GR	0.0261	0.0260	0.0298	0.0332	0.0354	0.0366	0.0652	0.0683	0.0773	0.0853	0.0906	0.0931
	Ours w/o DC	0.0467	0.0438	0.0463	0.0497	0.0521	0.0539	0.1096	0.1062	0.1101	0.1155	0.1191	0.1219
	Ours	0.0227	**0.0216**	0.0244	0.0275	0.0296	0.0310	**0.0583**	0.0619	0.0735	0.0817	0.0878	0.0891
	Langevin [29]	0.0545	0.0394	0.0336	0.0320	0.0317	0.0316	0.1116	0.0921	0.0828	0.0793	0.0781	0.0777
	DDRM [33]	0.0745	0.0615	0.0609	0.0640	0.0672	0.0692	0.1439	0.1246	0.1193	0.1197	0.1219	0.1242
$R = 6$	E2E-VarNet [8]	0.0466	-	-	-	-	-	0.1070	-	-	-	-	-
	UNet	0.0472	-	-	-	-	-	0.1078	-	-	-	-	-
	Ours w/o GR, w/o DC	0.0608	0.0552	0.0586	0.0642	0.0685	0.0713	0.1147	0.1051	0.1093	0.1183	0.1261	0.1310
	Ours w/o GR	0.0375	0.0317	0.0342	0.0387	0.0424	0.0444	0.0785	0.0745	0.0832	0.0934	0.1003	0.1044
	Ours w/o DC	0.0641	0.0543	0.0546	0.0586	0.0623	0.0649	0.1274	0.1155	0.1140	0.1172	0.1218	0.1255
	Ours	0.0344	**0.0289**	0.0314	0.0358	0.0392	0.0413	0.0731	**0.0713**	0.0803	0.0908	0.0976	0.1017
	DDRM [33]	0.0902	0.0776	0.0788	0.0842	0.0891	0.0923	0.1514	0.1332	0.1290	0.1313	0.1356	0.1390
$R = 8$	E2E-VarNet [8]	0.0602	-	-	-	-	-	0.1227	-	-	-	-	-
	UNet	0.0579	-	-	-	-	-	0.1175	-	-	-	-	-
	Ours w/o GR, w/o DC	0.0633	0.0586	0.0632	0.0697	0.0748	0.0782	0.1195	0.1111	0.1163	0.1255	0.1337	0.1397
	Ours w/o GR	0.0442	0.0408	0.0454	0.0510	0.0554	0.0579	0.0873	0.0859	0.0952	0.1058	0.1130	0.1171
	Ours w/o DC	0.0645	0.0566	0.0590	0.0644	0.0689	0.0721	0.1264	0.1158	0.1189	0.1248	0.1308	0.1359
	Ours	0.0421	**0.0354**	0.0389	0.0445	0.0488	0.0517	0.0824	**0.0807**	0.0910	0.1022	0.1097	0.1137
	Langevin [29]	0.0759	0.0624	0.0577	0.0572	0.0582	0.0586	0.1282	0.1160	0.1120	0.1120	0.1127	0.1135
	DDRM [33]	0.1070	0.0929	0.0943	0.1013	0.1079	0.1123	0.1598	0.1421	0.1392	0.1432	0.1485	0.1521

C Additional reconstruction plots

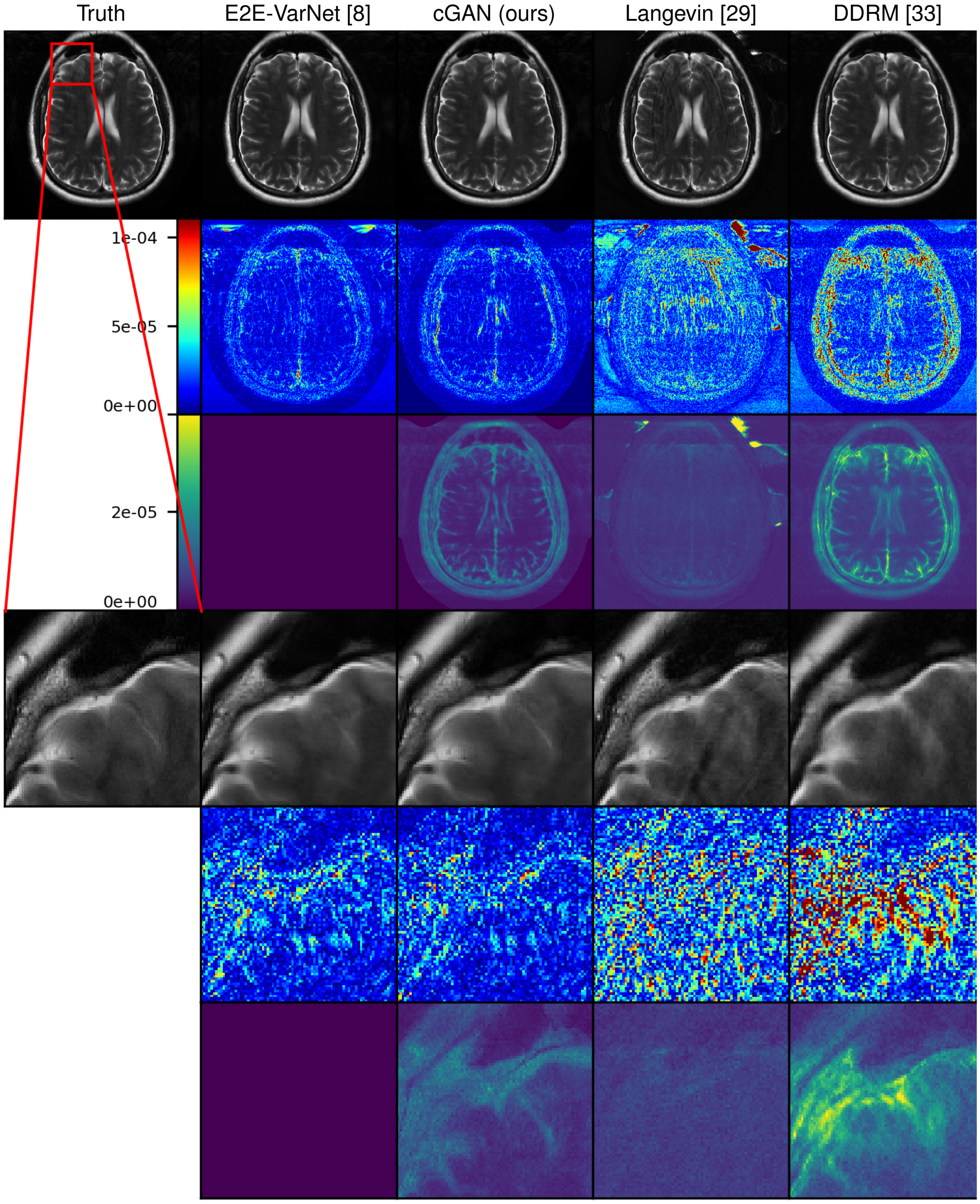


Figure C.1: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.

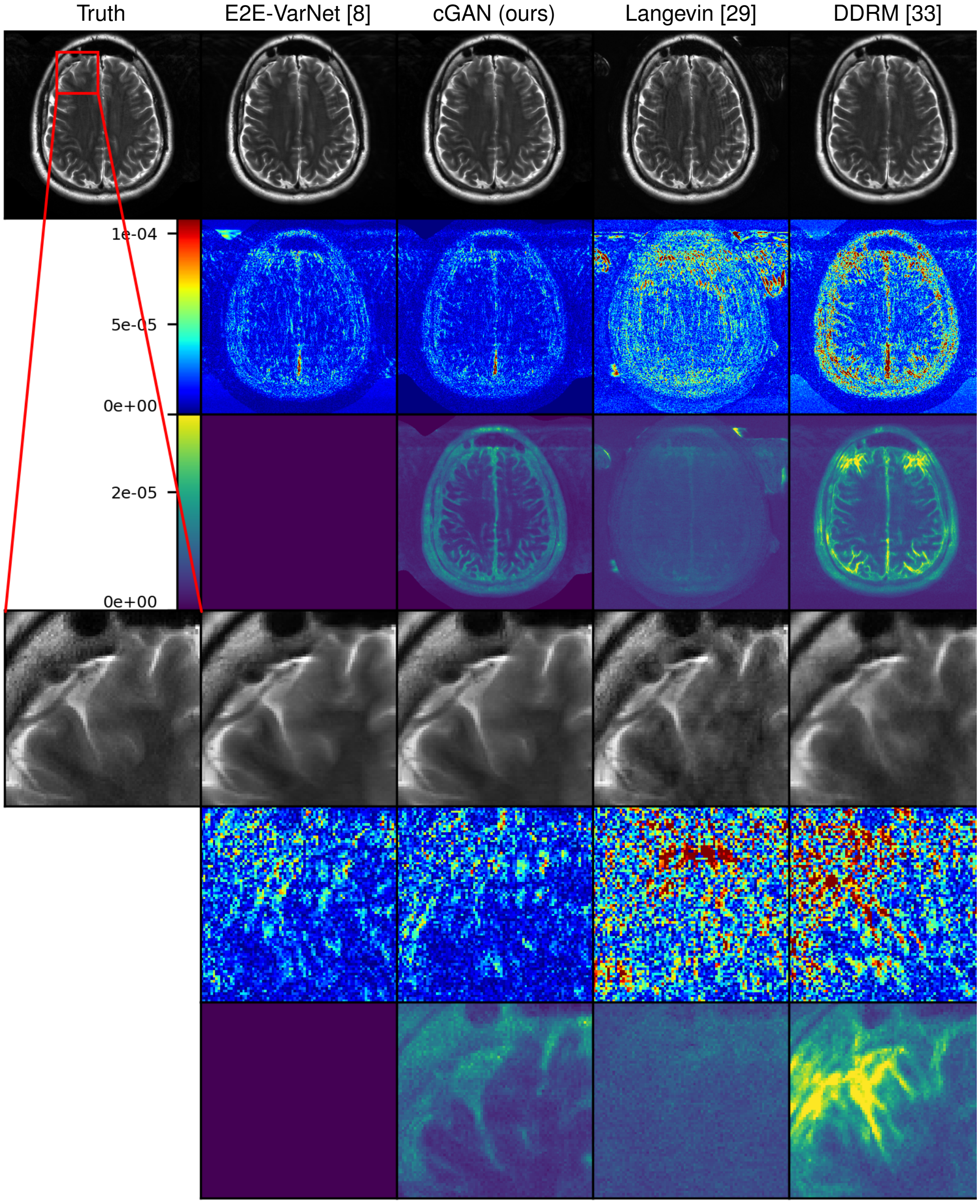


Figure C.2: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.

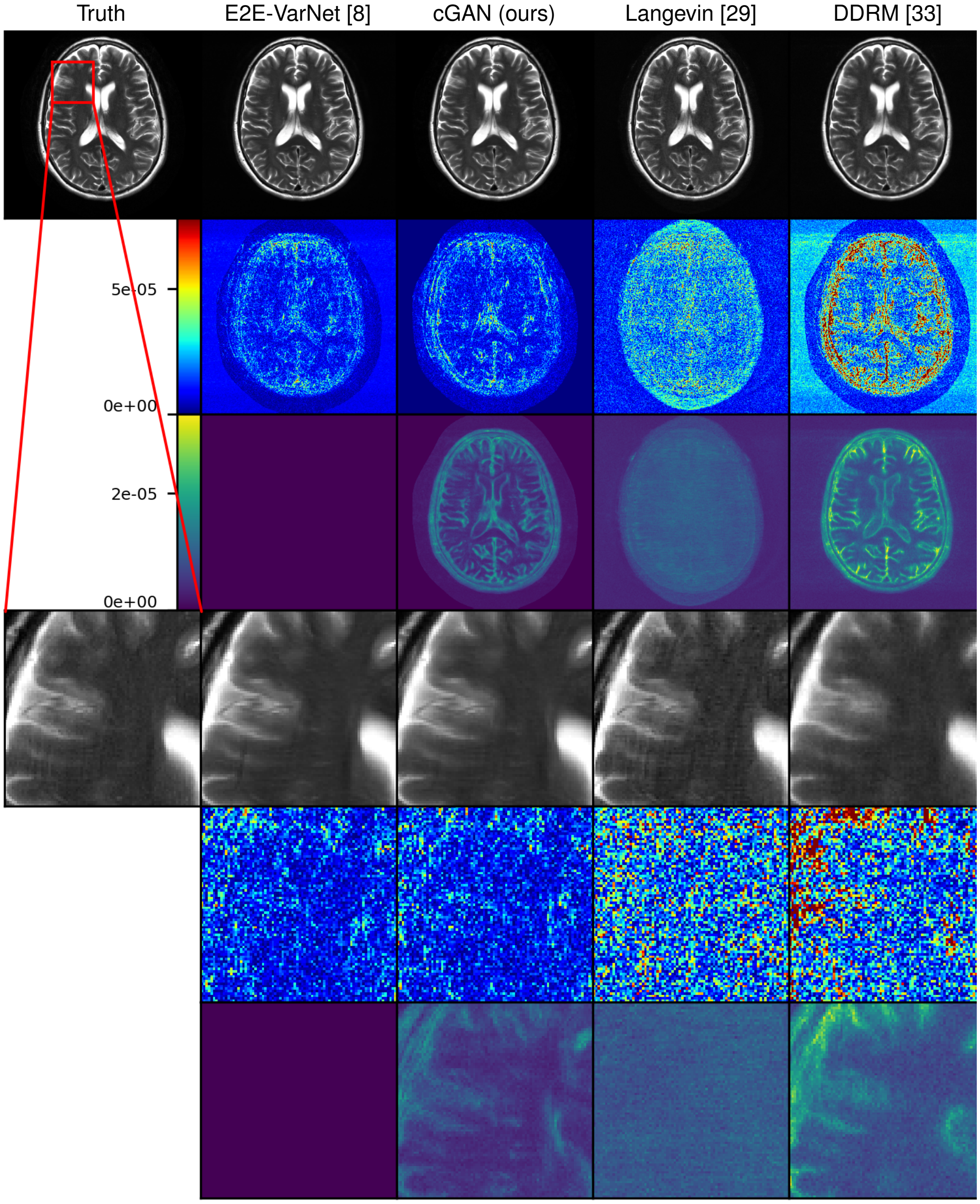


Figure C.3: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.

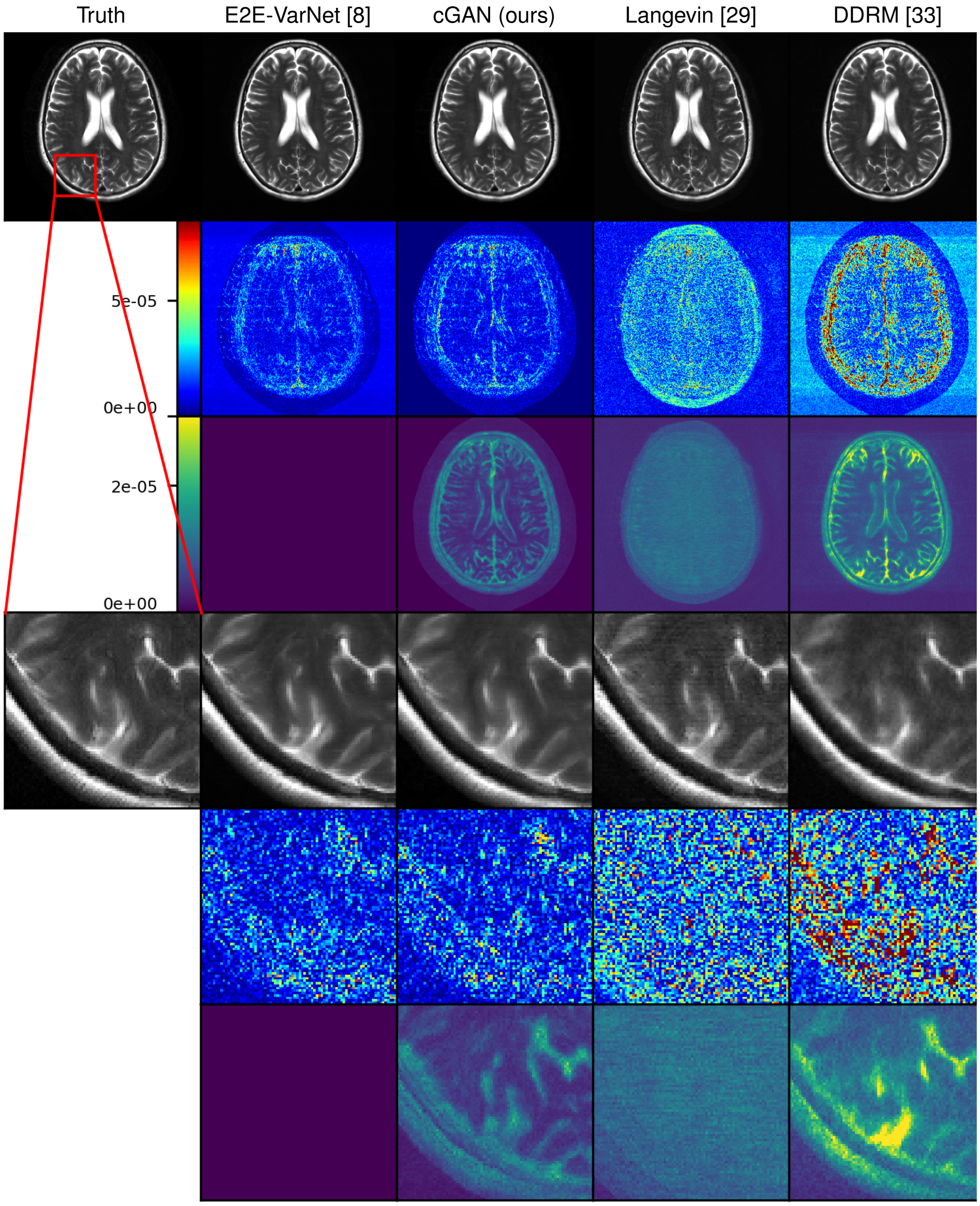


Figure C.4: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.

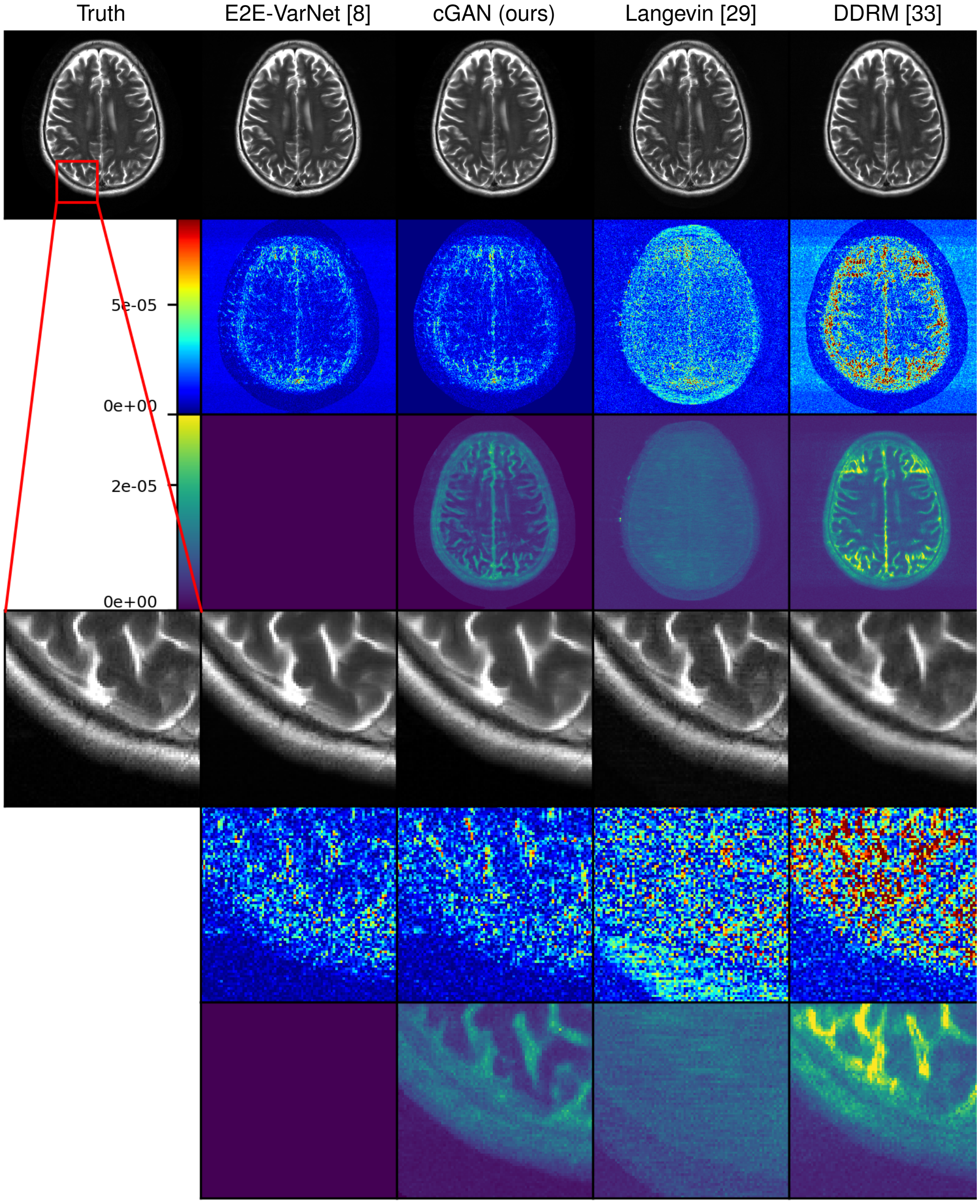


Figure C.5: Example $R = 4$ MRI reconstructions with $P = 4$. Rows 1 and 4: P-sample average $\widehat{\boldsymbol{x}}_{(P)}$. Rows 2 and 5: pixel-wise absolute error $|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}|$. Rows 3 and 6: pixel-wise SD $(\frac{1}{P}\sum_{i=1}^{P}(\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)})^2)^{1/2}$. Rows 4 through 6 show a zoomed region.