Uncertainty-Driven Adaptive Sampling via GANs

Thomas Sanchez, Igor Krawczuk, Zhaodong Sun and Volkan Cevher
École Polytechnique Fédérale de Lausanne, Switzerland
{thomas.sanchez,igor.krawczuk,zhaodong.sun,volkan.cevher}@epfl.ch

Abstract

We propose an adaptive sampling method for the linear model, driven by the uncertainty estimation with a generative adversarial network (GAN) model. Specifically, given a forward observation model that provides partial measurements $\boldsymbol{y}$ about an unknown parameter $\boldsymbol{x}$, we show how to build a GAN model to estimate its posterior $p(\boldsymbol{x}|\boldsymbol{y})$. We then leverage our approximate posterior to perform sequential adaptive sampling by actively selecting the measurement with the current maximal uncertainty. We empirically demonstrate that our posterior estimate contracts rapidly towards the correct mode, while outperforming the state-of-the-art approaches even for other criteria for which they are specifically trained, such as PSNR or SSIM.

1 Introduction

This work focuses on the following inverse problem, where we seek to recover a signal $\boldsymbol{x} \in \mathbb{C}^P$ from partial observations $\boldsymbol{y} \in \mathbb{C}^N$ obtained by subsampling a unitary transform matrix $\boldsymbol{A} \in \mathbb{C}^{P \times P}$:

$$\boldsymbol{y} = \boldsymbol{P}_\omega \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{\eta} \in \mathbb{C}^N$ is a noise vector; $\omega \subseteq [P] := \{1, \dots, P\}$ is an index set of allowable sampling locations with cardinality N; and $\boldsymbol{P}_\omega$ is a diagonal matrix such that $(\boldsymbol{P}_\omega)_{ii} = 1$ if $i \in \omega, 0$ otherwise.

The compressed sensing (CS) problem above is classical [1, 2], where the literature revolves around using statistical or geometric priors to circumvent its ill-posed nature since $N \ll P$. In particular, the Bayesian perspective on CS imposes prior distributions $p(\mathbf{x})$ on a random vector $\mathbf{x}$, aiming then at constructing a tractable posterior distribution $p(\mathbf{x}|\boldsymbol{y})$ which captures the probability for a sample $\boldsymbol{x}$ to have generated the observed data $\boldsymbol{y}$ [3, 4]. The geometric perspective in contrast leverages convex models based on the ℓ_1-norm, and then seeks to obtain point estimates [1, 2].

Contemporary approaches to posterior modeling in the CS setting can be roughly split into two camps, with one computing point estimates of the mean and the variance, assuming an underlying Gaussian distribution, such as [5], and the other approximating the posterior with a generative adversarial network (GAN) or a variational autoencoder, such as [6, 7, 8].

These data-driven, deep learning-based methods have also significantly improved practical reconstruction performance in CS problems [9, 10, 11, 12] as compared to the Bayesian or ℓ_1-norm based estimates. By data-driven, we mean that these approaches explicitly build underlying prior or posterior models using training data, as opposed to explicitly relying on carefully-constructed mathematical prior and geometric models, such as Laplace priors, the ℓ_1-norm, or Besov smoothness.

The successes of these approaches have led the broader medical imaging community to re-think the problem of optimizing the undersampling patterns also in a data-driven, adaptive (closed-loop) fashion [5, 13, 14, 15], instead of considering fixed (open-loop), random undersampling [1, 2, 16]. While there are earlier CS works that also use a data-driven perspective, they still perform the sampling in an open-loop fashion [17, 18, 19]. There are also closed-loop, but not data-driven sequential CS approaches, [3, 20, 21], which rely on mathematical models and do not use no training data.

In this work, we argue that the practical benefits of the generative methods have not been fully exploited in the CS setting. To this end, we propose a closed-loop, data-driven, Generative Adaptive

NeurIPS 2020 Workshop on Deep Learning and Inverse Problems, virtual.

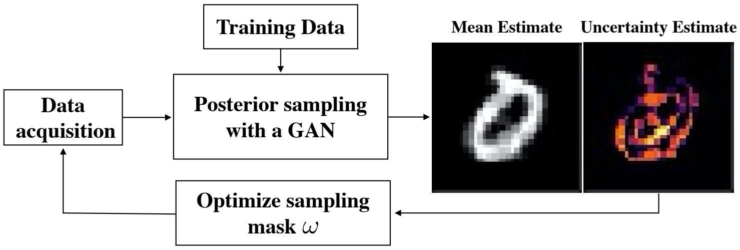

Figure 1: Overview of our Generative, Adaptive Sampling approach.

Sampling (GAS) method based on the recently proposed approaches of [6, 7]. In contrast, the GAN approaches in CS so far has been only used to obtain point estimates for recovery [22, 23].

By leveraging the flexible posterior modeling of GANs [6, 8] with their ability to exploit conditionals [24, 7], we develop a GAN model where the generator can simultaneously provide (i) a good image reconstruction, and (ii) uncertainty quantification in the image domain, thus naturally leading to an *end-to-end*, sequential adaptive sampling approach.

Notation. Throughout the text, we refer to random vectors as boldface straight letter $\mathbf{x}$, $\mathbf{y}$ and vector as boldface italic, $\boldsymbol{x}$, $\boldsymbol{y}$. We use capital boldface letters for matrices.

2 Methods

2.1 Reconstruction via conditional sampling

Given subsampled measurements via Equation (1), we consider two approaches to estimate $p(\mathbf{x}|\boldsymbol{y})$. In the first approach, we obtain the posterior mean and its variance, as a summary estimate of the posterior distribution. This is typically achieved via the problem of signal reconstruction:

$$(\hat{\boldsymbol{x}}_\theta, \hat{\boldsymbol{s}}_\theta^2) = f_\theta(\boldsymbol{y}_\omega) \tag{2}$$

where $\boldsymbol{y}_\omega$ specifies the dependency of $\boldsymbol{y}$ on ω; f_θ refers to a deep neural network with parameters θ; $(\hat{\boldsymbol{x}}_\theta, \hat{\boldsymbol{s}}_\theta^2)$ denote the point estimates of the conditional mean and variance of the model, respectively. Such models are typically trained using supervised learning by minimizing a Gaussian negative log-likelihood with heteroscedastic variance [25, 5].

In the second approach, we can have the ability draw samples of the posterior distribution:

$$\hat{\mathbf{x}}_\theta = G_\theta(\boldsymbol{y}_\omega, \mathbf{z}), \tag{3}$$

where $\mathbf{z}$ is a random vector drawn from a simple distribution; $G_\theta(\boldsymbol{y}_\omega, \mathbf{z})$ is a deep generative model mapping the input to a sample from $p(\mathbf{x}|\boldsymbol{y}_\omega)$. Such models typically rely on conditional generative adversarial networks [26, 6, 7] or variational autoencoders [27, 24].

2.2 Sampling optimization

The ideal sampling optimization algorithm would tailor the mask to each instance of $\mathbf{x} \sim p(\mathbf{x})$ solving

$$\operatorname*{arg\,min}_{\omega:|\omega|\leq N} \ell(\mathbf{x}, \hat{\boldsymbol{x}}_\theta(\mathbf{y}_\omega = \mathbf{P}_\omega \boldsymbol{A}\mathbf{x})), \tag{4}$$

which is impossible since this requires using the unknown ground truth signal $\mathbf{x}$ at testing time. Two main approaches have been explored to address this problem.

Fixed (open-loop) sampling. A majority of data-driven mask design approaches use fixed masks [28, 29, 30, 31]. Indeed, the subsampling mask is built using training data and kept fixed at inference time. Formally, the problem of choosing the subsampling pattern corresponds to finding a subset ω that satisfies

$$\operatorname*{arg\,min}_{\omega:|\omega|\leq N} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left[\ell(\mathbf{x}, \hat{\boldsymbol{x}}_\theta(\mathbf{y}_\omega = \boldsymbol{P}_\omega \boldsymbol{A}\mathbf{x}))\right], \tag{5}$$

where we are constrained with a maximal sampling budget N and want to find a mask that minimizes a given loss function ℓ. We estimate the expected loss using the empirical estimate with the training samples. We let $\hat{\boldsymbol{x}}_\theta(\mathbf{y}_\omega = \boldsymbol{P}_\omega \boldsymbol{A}\mathbf{x})$ denote an estimate of the mean, obtained either directly through Equation (2) or by averaging on $\mathbf{z}$ for Equation (3), i.e., $\hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_\omega) = \mathbb{E}_{\mathbf{z}}[\hat{\mathbf{x}}_\theta(\boldsymbol{y}_\omega, \mathbf{z})] = \mathbb{E}_{\mathbf{z}}[G_\theta(\boldsymbol{y}_\omega, \mathbf{z})]$.

Adaptive (closed-loop) sampling. In contrast with Equation 5, adaptive sampling optimizes a heuristic ℓ available at test time. For a fixed, unknown data sample $\boldsymbol{x}$, we use information of the previously obtained measurements $\boldsymbol{y}_{\omega_{t-1}}$ to determine what should be acquired at time t. The heuristic can be human crafted or learned from data, e.g. a score assigned by a neural network[5]. By conditioning on the previous samples and using the heuristic ℓ, we solve

$$v_t = \underset{v:v\in[P]}{\arg\min}\, \ell(\boldsymbol{x}, \hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_{\omega_t} = \mathbf{P}_{\omega_t}\boldsymbol{A}\boldsymbol{x})) \quad \text{subject to} \quad \omega_t = \{\omega_{t-1}, v\}, \tag{6}$$

where the variables v_t typically refer to individual pixels to be sequentially observed from the reference image $\boldsymbol{x} \sim p(\mathbf{x})$. Here we adopt a greedy approach, without trying to plan on a longer horizon. This has been explored, for instance, in [32, 13, 14]. Such approaches rely on performing additional computation or training an additional network in a supervised fashion in order to guide the planning. The work we present in the next section is based on a greedy approach, but could easily be combined with a long-term planning approach in future work.

2.3 GAS: Generative adaptive sampling

Departing from the idea of explicitly training a policy to guide adaptive sampling [32, 13, 5], GAS simply minimizes the variance of an approximate posterior. This turns out to yield a strong sampling policy *despite the networks involved being trained soley for conditional generation.*

We start by modeling an approximate posterior $p_\theta(\mathbf{x}|\boldsymbol{y}_\omega)$ on any mask ω by a conditional GAN [26, 7] or its Wasserstein counterpart [33, 6]. Given the trained model $G_{\theta^*}(\boldsymbol{y}_\omega, \cdot)$ that gives samples from the approximate posterior $p(\mathbf{x}|\boldsymbol{y}_\omega)$, adaptive sampling follows naturally. Starting at $t = 0$ from unconditional sampling, i.e. sampling from the prior, we iteratively sample the locations v_t with the largest variance in the transformed domain $\boldsymbol{A}$:

$$v_t = \underset{v:v\in[P]}{\arg\max}\, \mathbf{P}_{\{v\}} \text{Var}[\boldsymbol{A}\mathbf{x}|\mathbf{y}_{\omega_t}] \tag{7}$$

where $\text{Var}[\boldsymbol{A}\mathbf{x}|\mathbf{y}_{\omega_t}]$ denotes the point-wise variance on the vector $\boldsymbol{A}\mathbf{x}$. It is estimated at each stage by computing the empirical variance using samples from $G_{\theta^*}(\mathbf{y}_{\omega_t}, \cdot)$. The chosen element is then added in the sampling mask as $\omega_{t+1} = \{\omega_t, v_t\}$.

3 Experiment

We test the applicability of GAS on MNIST, where we explore subsampling in pixel ($\boldsymbol{A} = \mathbf{I}$) We use the same architecture and optimizer for both the Wasserstein GAN (WGAN) [34] and the Neural Conditioner (NC) [7]. The architectures used are a Unet [35] with residual blocks (ResUNet), used in [7], and a cascade of residual networks (c-ResNet) used in [5]. Both architecture are use data-consistency layers [10], that integrate the observations into the final image, preventing the networks from discarding them. We generate undersampled images by dynamically sampling masks at random or by sampling from a Gaussian centered in the middle of the image (variable-density sampling), using various sampling rates between 0.5% and 20% sampling rate. Throughout the experiments, we used 5 samples from the posterior to compute empirical posterior means and variances. In addition to these baselines, we also trained reconstruction networks to further expand the comparison. We refer the model as *Recon.* Further details of the implementation are provided in the appendix, where. we also included additional experiments on CIFAR10.

Our results are summarized in Table 1 below, and we compare the following sampling models.

- VDS: Variable-Density sampling. The model is evaluated on the variable-density masks used during training, a common baseline in CS works [36].
- LBC: Learning-based compressive sensing [19]. A fixed sampling mask is trained and applied on all testing data. We use their stochastic version for scalability reasons [37].
- GAS: Our generative, uncertainty-driven adaptive compressed sensing approach.

In the sampling experiments, the baselines are evaluated on various sampling rates up to 20% sampling rates. The LBC and adaptive methods are evaluated on the whole range from 0 to 20%. The models are test on several sampling rates and sampling masks that they never saw during training. We report the PSNR, SSIM [38] as well as the downstream classification accuracy, using a small residual network trained on the original MNIST digits, and evaluated on a batch of samples from the conditional generator. The final classification result is ensembled from this batch of predictions. As we consider results at different sampling rates, we summarize the overall performance by computing the area under curve (AUC) of the different metrics. We then compute the average and standard deviations of these test curves. The curves, illustrating the performance of the models at different sampling rates, are provided in the appendix.

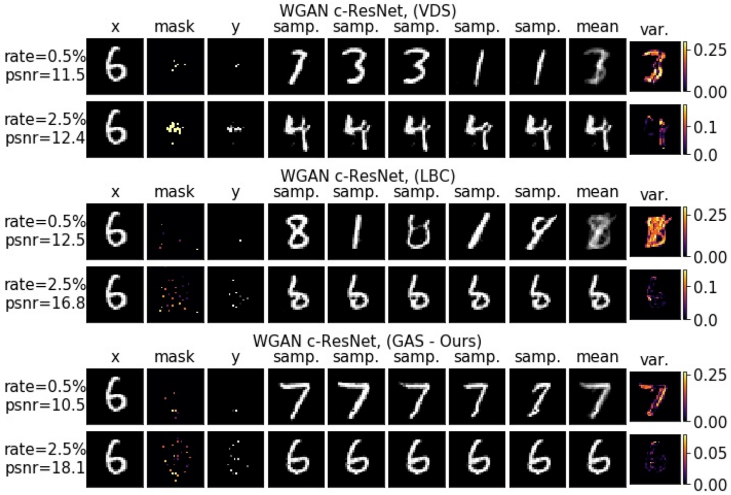

Figure 2: Image domain illustration of the WGAN c-ResNet sampling using three different methods. The columns contain respectively: the ground truth image, the mask (with brighter color meaning that the location has been selected recently), the observation, several samples and the empirical posterior mean and variances. The first row shows the result obtained with 0.5% and 2.5% sampling respectively.

We see on Tables 1 that both GAN models strongly outperform the LBC baseline in the image domain. In the Fourier domain, the performance gap is smaller, due to the more homogeneous distribution of the different spectra. The WGAN c-ResNet adaptive is still matches the LBC performance, but the LBC is much more competitive in this setting and tends to outperform GAS. This is likely due to the fact that the LBC mask is trained to minimize PSNR, whereas our model uses the uncertainty in an unsupervised fashion. We would also like to emphasize that the model learns to perform unconditional sampling despite never seeing an empty mask during training, as shown in the Appendix.

Algorithm	Model	Architecture	PSNR [dB] (↑)	SSIM (↑)	Accuracy [%] (↑)
VDS	NC	ResUNet	17.10	0.76	0.67
	WGAN	c-ResNet	16.41	0.76	0.67
VDS	Recon.	ResUNet	18.87 ± 2.31	0.77 ± 0.05	0.70 ± 0.21
	Recon.	c-ResNet	18.85 ± 2.18	0.70 ± 0.05	0.75 ± 0.20
LBC [37]	NC	ResUNet	21.39 ± 2.34	0.90 ± 0.04	0.92 ± 0.14
	NC	c-ResNet	20.33 ± 2.20	0.88 ± 0.04	0.93 ± 0.15
	WGAN	ResUNet	20.29 ± 2.09	0.89 ± 0.04	0.90 ± 0.14
	WGAN	c-ResNet	20.48 ± 2.14	0.89 ± 0.04	0.92 ± 0.14
GAS (Ours)	NC	ResUNet	$\mathbf{40.15 \pm 15.78}$	$\mathbf{0.94 \pm 0.04}$	0.91 ± 0.15
	NC	c-ResNet	31.47 ± 6.69	0.93 ± 0.03	$\mathbf{0.93 \pm 0.14}$
	WGAN	ResUNet	31.73 ± 9.10	0.92 ± 0.06	0.88 ± 0.17
	WGAN	c-ResNet	$\mathbf{35.81 \pm 10.34}$	$\mathbf{0.96 \pm 0.03}$	$\mathbf{0.95 \pm 0.13}$

Table 1: Average test set AUC (one AUC per image) with standard deviation on MNIST, in image domain.

4 Outlook

We presented a novel approach, rooted in Bayesian modeling, for adaptive sampling. We showed that, in line with recent advances [7], GANs performs well at the task of learning a large number of different conditional distributions. Having been trained on a limited set of sampling rates and masks, the model is able to convincingly generalize to unseen masks and sampling rates, and is even able to perform unconditional sampling without having been explicitly trained on this task. In addition, the samples from the conditional are able to provide meaningful uncertainty information, and was successfully leveraged as a greedy, adaptive sampling policy. This success is due to the unique ability of generative models to capture complex, multimodal distributions and cheaply sample from them at inference time. This could make them prime candidates for a more uncertainty-aware deep learning approach to inverse problems. Our future work will demonstrate the applicability of this method to problems where sampling does not occur in image domain, such as MRI, with Fourier domain sampling.

Acknowledgments

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n° 725594 - time-data)m from Hasler Foundation Program: Cyber Human Systems (project number 16066) and from the Office of Naval Research (ONR) (grant n° N62909-17-1-2111).

References

[1] D. L. Donoho, “Compressed sensing,” *IEEE transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans, on Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” *IEEE Transactions on signal processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[4] S. D. Babacan, R. Molina, and A. K. Katsaggelos, “Bayesian compressive sensing using laplace priors,” *IEEE Transactions on image processing*, vol. 19, no. 1, pp. 53–63, 2009.

[5] Z. Zhang, A. Romero, M. J. Muckley, P. Vincent, L. Yang, and M. Drozdzal, “Reducing uncertainty in undersampled MRI reconstruction with active acquisition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2049–2058.

[6] J. Adler and O. Öktem, “Deep bayesian inversion,” *arXiv preprint arXiv:1811.05910*, 2018.

[7] M. Belghazi, M. Oquab, and D. Lopez-Paz, “Learning about an exponential amount of conditional distributions,” in *Advances in Neural Information Processing Systems 32*, 2019. [Online]. Available: http://papers.nips.cc/paper/9523-learning-about-an-exponential-amount-of-conditional-distributions.pdf.

[8] F. Tonolini, A. Lyons, P. Caramazza, D. Faccio, and R. Murray-Smith, “Variational inference for computational imagsing inverse problems,” *arXiv preprint arXiv:1904.06264*, 2019.

[9] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[10] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, “A deep cascade of convolutional neural networks for dynamic MR image reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.

[11] Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, and A. Ozcan, “Phase recovery and holographic image reconstruction using deep learning in neural networks,” *Light: Science & Applications*, vol. 7, no. 2, pp. 17 141–17 141, 2018.

[12] J. Adler and O. Öktem, “Learned primal-dual reconstruction,” *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[13] K. H. Jin, M. Unser, and K. M. Yi, “Self-supervised deep active accelerated MRI,” *arXiv preprint arXiv:1901.04547*, 2019.

[14] L. Pineda, S. Basu, A. Romero, R. Calandra, and M. Drozdzal, “Active mr k-space sampling with reinforcement learning,” *arXiv preprint arXiv:2007.10469*, 2020.

[15] I. A. Huijben, B. S. Veeling, K. Janse, M. Mischi, and R. J. Van Sloun, “Learning sub-sampling and signal recovery with applications in ultrasound imaging,” *IEEE Transactions on Medical Imaging*, 2020.

[16] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, “Breaking the coherence barrier: A new theory for compressed sensing,” in *Forum of Mathematics, Sigma*, Cambridge University Press, vol. 5, 2017.

[17] S. Ravishankar and Y. Bresler, “MR image reconstruction from highly undersampled k-space data by dictionary learning,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, 2011.

[18] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, “Learning-based compressive subsampling,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 809–822, 2016.

[19] B. Gözcü, R. K. Mahabadi, Y.-H. Li, E. Ilıcak, T. Çukur, J. Scarlett, and V. Cevher, “Learning-based compressive MRI,” *IEEE Transactions on Medical Imaging*, 2018.

[20] J. Haupt, R. Nowak, and R. Castro, “Adaptive sensing for sparse signal recovery,” in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, IEEE, 2009, pp. 702–707.

[21] M. Seeger, H. Nickisch, R. Pohmann, and B. Schölkopf, “Optimization of k-space trajectories for compressed sensing by bayesian experimental design,” *Magn. Reson. Med.*, vol. 63, no. 1, pp. 116–126, 2010.

[22] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” *arXiv preprint arXiv:1703.03208*, 2017.

[23] F. Latorre, A. Eftekhari, and V. Cevher, “Fast and provable ADMM for learning with generative priors,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 12 027–12 039. [Online]. Available: http://papers.nips.cc/paper/9371-fast-and-provable-admm-for-learning-with-generative-priors.pdf.

[24] O. Ivanov, M. Figurnov, and D. Vetrov, “Variational autoencoder with arbitrary conditioning,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=SyxtJh0qYm.

[25] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” In *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[26] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[27] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[28] S. Ravishankar and Y. Bresler, “Adaptive sampling design for compressed sensing MRI,” in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, IEEE, 2011, pp. 3751–3755.

[29] C. D. Bahadir, A. V. Dalca, and M. R. Sabuncu, “Learning-based optimization of the undersampling pattern in MRI,” in *International Conference on Information Processing in Medical Imaging*, Springer, 2019, pp. 780–792.

[30] Y. Wu, M. Rosca, and T. Lillicrap, “Deep compressed sensing,” in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, Long Beach, California, USA: PMLR, Sep. 2019, pp. 6850–6860. [Online]. Available: http://proceedings.mlr.press/v97/wu19d.html.

[31] I. A. Huijben, B. S. Veeling, and R. J. van Sloun, “Deep probabilistic subsampling for task-adaptive compressed sensing,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJeq9JBFvH.

[32] X. Huan and Y. M. Marzouk, “Sequential bayesian optimal experimental design via approximate dynamic programming,” *arXiv preprint arXiv:1604.08320*, 2016.

[33] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.

[34] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[35] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[36] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[37] T. Sanchez, B. Gözcü, R. B. van Heeswijk, E. Ilıcak, T. Çukur, *et al.*, “Scalable learning-based sampling optimization for compressive dynamic MRI,” *arXiv preprint arXiv:1902.00386*, 2019.

[38] Z. 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*, vol. 13, no. 4, pp. 600–612, 2004.

A Implementation details

We carried out our experiments on the MNIST dataset. The training set contains 60000 handwritten digits of size 28×28. We held out 10000 digits for validation purposes. The test set consists of 10000 digits. We zero-padded all images to 32×32 for convenience. We ran all experiments on a Tesla K40 GPU, using batch size 64, and initialized the networks using the default Pytorch values.

A.1 Experiment setting

We experiment on two main generator architectures, a UNet[1] with residual blocks, called ResUNet for short, as used in [2]. The architecture used is detailed in Table 2 below. We also used a cascade of residual networks, called c-ResNet [3], where one block of the architecture is given in Table 3. The network was composed of 3 such blocks. Our choice was motivated by the fact that encoder-decoder architectures are required by the high dimensional input of this type of conditional GAN. Regarding the discriminator, we used a ResNet-like architecture, following [2, 4]. The architecture is reported in Table 4.

The cWGAN models were trained with gradient penalty [4] and the generators are followed with a sigmoid to keep the range in $[0, 1]$, and followed by a data-consistency layer [5, 6], which has the role of replacing the generated data with the observed ones at the last layer.

All models were trained using Adam [7] with learning rate 10^{-4}, $\beta = (0.5, 0.999)$ and weight decay 10^{-4}. All models were trained for 300 epochs. The images were undersampled using masks that were either drawn at random or drawn with a probability decaying away from the center. The settings used are given in greater detail in the Table 5 below.

The Neural Conditioner (NC) [2] generator and discriminator were updated every other round. In the WGAN case, we optimized the generator once, followed by four updates of the discriminator, as done in [8].

Similarly to [2], we found gradient regularization of the discriminator to be essential. However, we obtained very poor training results using the authors proposed gradient penalty, and observed that the gradient penalty of [4] led to a stable training. We used the gradient penalty with a weight of 10, following [8] in the WGAN case, and a value of 1 for the NC, but did not observe significant differences between these two weights.

We note in passing that contrarily to [2], we did not observe any particular effect of spectral normalization of the encoder part of the generator.

Reconstruction networks. Using the same architectures as the GAN generators (ResUNet and c-ResNet), we also trained two reconstruction networks by minimizing the ℓ_2 distance between ground truth images and reconstructed images. Formally, given a training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_{\omega_i,i})\}_{i=1}^n$, the network f_θ is trained by minimizing $\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{x}_i - f_\theta(\boldsymbol{y}_{\omega_i,i})\|_2^2$.

We used Adam [7] with learning rate $2 \cdot 10^{-4}$, $\beta = (0.0, 0.9)$ and weight decay $= 10^{-4}$. We trained the models for 200 epochs using a batch size 128. The masks used are the same as the GANs, described in Table 5 below.

A.2 Sampling experiments

After training, we carried out several rounds of evaluation for different types of sampling as reported in Table 1. We hereafter detail the different setting we used. As mentioned in the main body of the paper, we chose to estimate the posterior mean and variances using 5 samples from the posterior. We found that 2 samples were sufficient for the adaptive sampling with GAS to yield good results. We also observed that more samples 20 to 100 made little improvement in the sampling quality or image quality, except that it suppressed the noise of the empirical posterior mean and variance a bit more efficiently.

For the baseline, variable-density sampling, we constructed both random and Gaussian masks using the settings of Table 5, but including also sampling rates that were never seen during training, including rates above 20%.

	Type	Output size	Comments
Input	–	$2 \times 32 \times 32$	
Encoder	$\text{ResBlock}_1(3,1,1)$	$128 \times 32 \times 32$	
	Avg Pool +$\text{ResBlock}_2(3,1,1)$	$128 \times 16 \times 16$	
	Avg Pool +$\text{ResBlock}_3(3,1,1)$	$128 \times 8 \times 8$	
	Avg Pool +$\text{ResBlock}_4(3,1,1)$	$128 \times 4 \times 4$	
	Avg Pool +$\text{ResBlock}_5(3,1,1)$	$128 \times 2 \times 2$	
Bottleneck	Avg Pool +$\text{ResBlock}_6(3,1,1)$	$128 \times 1 \times 1$	
Decoder	$\text{TrConv}_1(2,2,0)$	$128 \times 2 \times 2$	
	$\text{ResBlock}_7(3,1,1)$	$128 \times 2 \times 2$	$\text{Cat}[\text{TrConv}_1, \text{ResBlock}_5]$
	$\text{TrConv}_2(2,2,0)$	$128 \times 4 \times 4$	
	$\text{ResBlock}_8(3,1,1)$	$128 \times 4 \times 4$	$\text{Cat}[\text{TrConv}_2, \text{ResBlock}_4]$
	$\text{TrConv}_3(2,2,0)$	$128 \times 8 \times 8$	
	$\text{ResBlock}_9(3,1,1)$	$128 \times 8 \times 8$	$\text{Cat}[\text{TrConv}_3, \text{ResBlock}_3]$
	$\text{TrConv}_4(2,2,0)$	$128 \times 16 \times 16$	
	$\text{ResBlock}_{10}(3,1,1)$	$128 \times 16 \times 16$	$\text{Cat}[\text{TrConv}_4, \text{ResBlock}_2]$
	$\text{TrConv}_5(2,2,0)$	$128 \times 32 \times 32$	
	$\text{ResBlock}_{11}(3,1,1)$	$128 \times 32 \times 32$	$\text{Cat}[\text{TrConv}_5, \text{ResBlock}_1]$
	$\text{Conv}(1,1,0) + \text{Sigmoid} + \text{DC}$	$1 \times 32 \times 32$	

Table 2: ResUNet architecture used in the experiments, inspired by the architecture in [2]. The activation used throughout is a LeakyReLu with slope 0.2. We had $n_{\text{channels,in}}$ containing 1 channel for the image, plus 1 channel for the noise. $\text{ResBlock}(3,1,1)$ denotes a residual block with a 3×3 kernel size, stride 1 and padding 1. $\text{TrConv}_4(2,2,0)$ denotes a transposed convolution block with a 2×2 kernel size, stride 2 and padding 0

	Type	Output size	Comments
Input	–	$2 \times 32 \times 32$	
Encoder	$\text{Conv}_1(3,2,1)$	$32 \times 16 \times 16$	
	$\text{Conv}_2(3,2,1)$	$64 \times 8 \times 8$	
	$\text{Conv}_3(3,2,1)$	$128 \times 4 \times 4$	
Bottleneck	$\text{ResBlock}_1(3,1,1)$	$128 \times 4 \times 4$	Skip-add from module $i-1$
	$\text{ResBlock}_2(3,1,1)$	$128 \times 4 \times 4$	
	$\text{ResBlock}_3(3,1,1)$	$128 \times 4 \times 4$	Skip to module $i+1$
Decoder	$\text{TrConv}_1(4,2,1)$	$64 \times 8 \times 8$	
	$\text{TrConv}_2(4,2,1)$	$32 \times 16 \times 16$	
	$\text{TrConv}_3(4,2,1)$	$16 \times 32 \times 32$	
	$\text{Conv}(1,1,0) + (\text{Sigmoid}) + \text{DC}$	$1 \times 32 \times 32$	Sigmoid only at the last block

Table 3: One block of c-ResNet, used in [3]. In the paper, we used 3 blocks with 3 residual blocks in the Bottlecneck layer. We had $n_{\text{channels,in}}$ containing 1 channel for the image, plus 1 channel for the noise. $n_{\text{channels,out}}$ 1 or 2.

Type	Output size
Input	$2 \times 32 \times 32$
$\text{ResBlock}_1(3,2,1)$	$128 \times 16 \times 16$
$\text{ResBlock}_2(3,2,1)$	$128 \times 8 \times 8$
$\text{ResBlock}_3(3,2,1)$	$128 \times 4 \times 4$
Global mean pooling	$128 \times 1 \times 1$
$\text{Conv}(1,1,0) + \text{Sigmoid}$	$1 \times 1 \times 1$

Table 4: Discriminator ResNet. This was inspired by the discriminator of [2].

Regarding the LBC approach [10], there are two parameters to be adjusted, namely the number of training samples to be used at each stage of the greedy mask optimization, and the size of candidate locations to be used. We set these to 64 and 256 respectively. The criterion used for the optimization was to select at each round the sample that maximally decreased the mean squared error in the empirical posterior mean. The mask was then evaluated at different stages on the testing set and the results reported.

Our adaptive sampling procedure, GAS, was also averaged on 5 samples from the generator, and the optimization was carried out by selecting at each round the point with the largest empirical variance in the transformed domain.

Random	Rates	0.005	0.01	0.025	0.03	0.05	0.1	0.15	0.2
Gaussian	Rates	0.005	0.01	0.025	0.05	0.1	0.15	0.2	
	Center fraction	0.0	0.0	0.0	0.04	0.05	0.06	0.07	
	Density standard deviation	0.06	0.06	0.06	0.08	0.12	0.16	0.16	
	Scaling	0.01	0.01	0.02	0.05	0.05	0.05	0.05	

Table 5: Parameters used for sampling random masks used throughout the training. *Rates* corresponds to different sampling rates used randomly. For the **Gaussian**, centered in the middle of the image, there are additional parameters for each sampling rate. The *center fraction* is the fraction of elements that are systematically acquired in the center of the image. The *density standard deviation* will control how far the masks will spread and the *scaling* parameter will scale the density up so that there remains a chance to sample high frequencies even when the density is narrow. These parameters have been commonly used to empirically tune variable density sampling in compressed sensing MRI, and can be found in the codes of [9] for instance.

B Additional results

B.1 Additional MNIST results

We provide additional results that further illustrate the performance of our models and of the different baselines considered. In Figure 3, we show additional image domain results, using the NC model [2], using the ResUNet architecture. We also present some quantitative results on Figure 4 are respectively summarized by AUC curves in Table 1. We then provide some images obtained with our reconstruction algorithms on Figure 5.

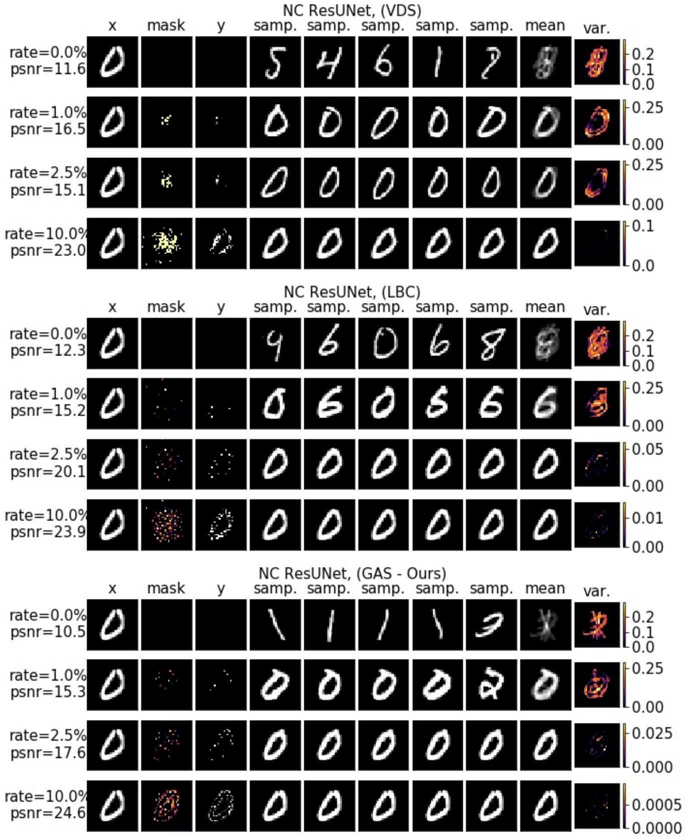

Figure 3: Additional Image domain results, featuring different sampling rates, using a NC ResUNet model for conditional sampling.

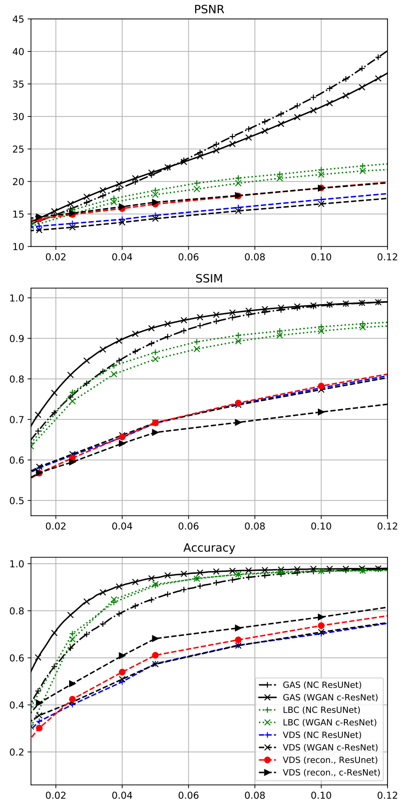

Figure 4: Image domain plots, showing PSNR, SSIM and Accuracy for some selected models

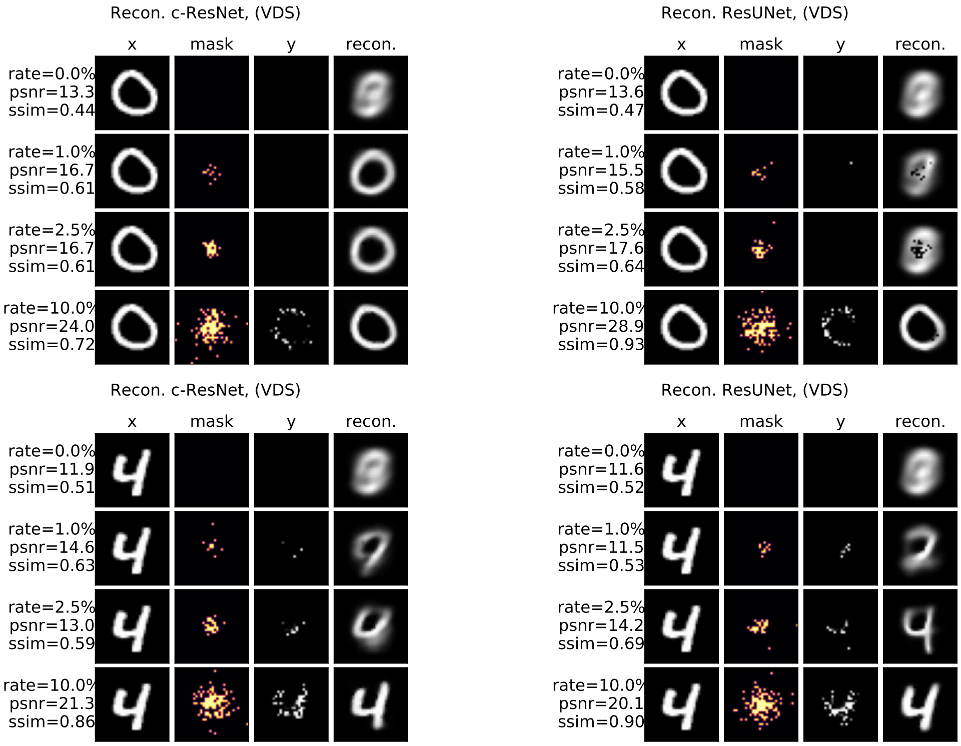

Figure 5: Additional Image domain results, featuring different sampling rates, using a c-ResNet and ResUNet trained for reconstruction.

B.2 CIFAR10 experiment

Finally, we provide results on CIFAR10, for which the pixelwise subsampling task is much more challenging. We chose to use grayscale versions of the images.

Experiment setting. For CIFAR10, we modified the training procedures slightly. For the WGAN we use ExtraAdam [11], TTUR [12] with learning rates of $lr_G = 10^{-4}$ and $lr_D = 3 \cdot 10^{-4}$, one sided gradient penalty [13] and betas of $(0, 0.9)$. Number of epochs was set to 200 with the learning rates being halved every 50 steps. Batchsize was set to 128 and we used 8 GPUs in total, yielding an effective batch size of 1024. For the NC we retain batch size 128 and the Adam settings, simply moving to 8 GPUs and increasing training time to 200 epochs.

Results. Figure 6 and fig. 7 showcase GAS results using a WGAN and NC respectively. We evaluate VDS and GAS on a subset of the test set of CIFAR10 composed of 1000 images and report the results on Table 6. The results are detailed in the plots of Figure 8.

Algorithm	Model	Architecture	PSNR [dB] (↑)	SSIM (↑)	Accuracy [%] (↑)
VDS	NC	ResUNet	9.32 ± 0.70	0.11 ± 0.03	0.10 ± 0.17
	WGAN	ResUNet	8.72 ± 0.62	0.09 ± 0.03	0.10 ± 0.17
VDS	Recon.	ResUNet	6.73 ± 0.51	0.09 ± 0.02	0.11 ± 0.17
	Recon.	c-ResNet	11.79 ± 0.72	0.13 ± 0.03	0.10 ± 0.16
GAS (Ours)	NC	ResUNet	14.36 ± 1.89	0.55 ± 0.08	$\mathbf{0.27 \pm 0.22}$
	WGAN	ResUNet	$\mathbf{15.26 \pm 2.34}$	$\mathbf{0.57 \pm 0.09}$	0.21 ± 0.18

Table 6: Average test set AUC (one AUC per image) with standard deviation on CIFAR10, in image domain. This was computed on a subset of the test set.

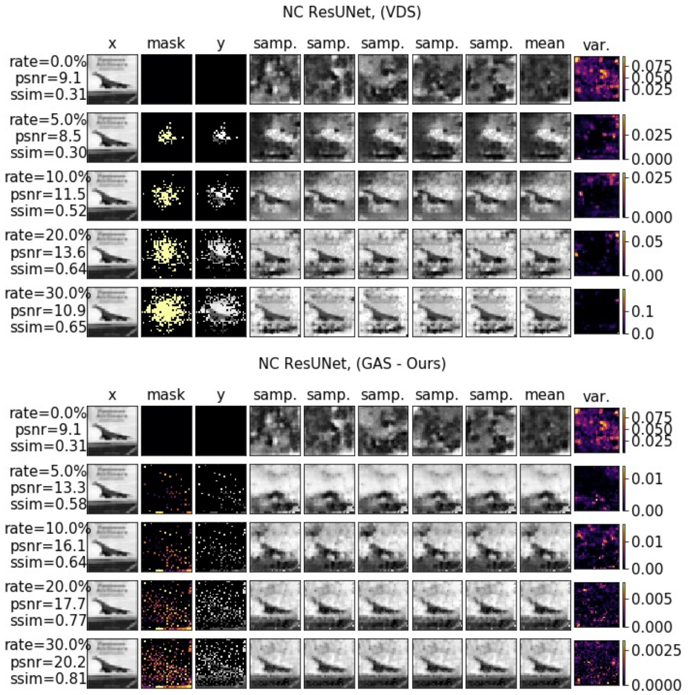

Figure 6: CIFAR10 results, featuring different sampling rates, using a NC ResUNet model for conditional sampling. Observe that while the model saw only masks going from 0.5% to 20% during training, it generalizes well up to 30%, gaining 2.5dB in the case of GAS.

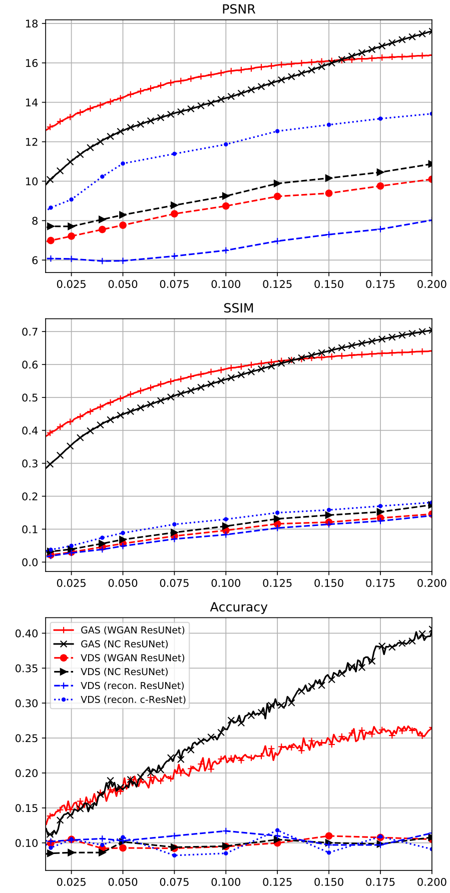

Figure 8: Image domain plots, showing PSNR, SSIM and Accuracy for some selected models

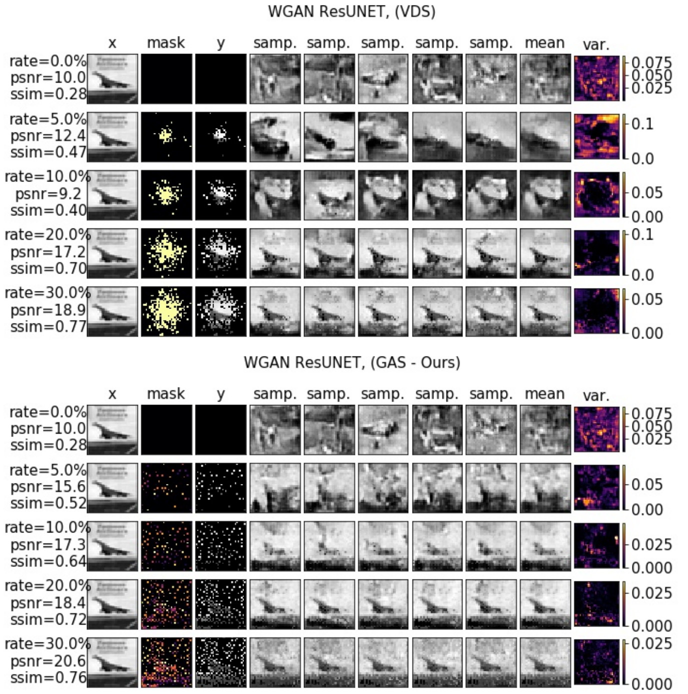

Figure 7: CIFAR10 results, featuring different sampling rates, using a WGAN ResUNet model for conditional sampling. Observe that while the model saw only masks going from 0.5% to 20% during training, it generalizes well up to 30%.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[2] M. Belghazi, M. Oquab, and D. Lopez-Paz, "Learning about an exponential amount of conditional distributions," in *Advances in Neural Information Processing Systems 32*, 2019. [Online]. Available: http://papers.nips.cc/paper/9523-learning-about-an-exponential-amount-of-conditional-distributions.pdf.

[3] Z. Zhang, A. Romero, M. J. Muckley, P. Vincent, L. Yang, and M. Drozdzal, "Reducing uncertainty in undersampled MRI reconstruction with active acquisition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2049–2058.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[5] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.

[6] M. Mardani, Q. Sun, D. Donoho, V. Papyan, H. Monajemi, S. Vasanawala, and J. Pauly, "Neural proximal gradient descent for compressive imaging," in *Advances in Neural Information Processing Systems*, 2018, pp. 9573–9583.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] J. Adler and O. Öktem, "Deep bayesian inversion," *arXiv preprint arXiv:1811.05910*, 2018.

[9] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye, "K-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI," *Magn. Reson. Med.*, vol. 61, no. 1, pp. 103–116, 2009.

[10] T. Sanchez, B. Gözcü, R. B. van Heeswijk, E. Ilıcak, T. Çukur, *et al.*, "Scalable learning-based sampling optimization for compressive dynamic MRI," *arXiv preprint arXiv:1902.00386*, 2019.

[11] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, "A variational inequality perspective on generative adversarial networks," *arXiv preprint arXiv:1802.10551*, 2018.

[12] 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 *Advances in neural information processing systems*, 2017, pp. 6626–6637.

[13] H. Petzka, A. Fischer, and D. Lukovnikov, "On the regularization of wasserstein GANs," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1hYRMbCW.