Robustness of Deep Learning in MRI Reconstruction

Haoxu Huang, **Buyun Liang**
Department of Computer Science & Engineering
University of Minnesota, Twin Cities

Abstract

Magnetic Resonance Imaging (MRI) is an important medical imaging technique to obtain the anatomy and the physiological processes of the body. MRI examination usually takes a long time, which increases patients stress and diagnostic cost. Decreasing sampling number in Fourier domain could efficiently reduce the acquisition time, hence accelerate the examination process. As the reconstruction process violates Nyquist-Shannon sampling theorem, how to reconstruct high quality image from the subsampled Fourier domain is a difficult problem. The rapid development of deep learning techniques shows the potential to solve this problem. However, researchers found the deep learning techniques have a robustness issue: hardly perceptible perturbation in input data could lead to serious artifacts or enhanced noise, which causes the diagnosis untruthful. A novel adversarial training strategy was used in this projects to improve the stability of the reconstruction model. The result shows this training strategy successfully minimized the effect of perturbations.

1 Introduction

Magnetic Resonance Imaging (MRI) is widely used in radiology to form anatomy pictures of organ and tissue. The advantage of MRI is the lack of ionizing radiation and superiority in detecting soft tissues [1]. However, MRI examination may take more than 30 minutes, which increases the diagnostic cost and patients stress [2]. Thus, accelerating MRI is widely investigated by many researchers to minimize the drawbacks. MRI is an indirect process, during which the images are generated from frequency and phase measurements by using Fourier transform. Lowering the number of samples acquired in Fourier-space (*i.e.,* k-space) could improve the MRI efficiency, but make the reconstruction a challenging problem.

The reconstruction problem is a finite-dimensional, underdetermined inverse problem [3]:

$$\text{Recover image } x \in \mathbb{C}^N \text{ from noisy measurements } y = Ax + e, \tag{1}$$

where $x \in \mathbb{C}^N$ is the image to recover, $y \in \mathbb{C}^m$ denotes the vector of measurements (k-space), and $A \in \mathbb{C}^{m \times N}$ is the sampling operator (*i.e.,* subsampled discrete Fourier transform). Also $m \ll N$ in the underdetermined setting.

The classical method for solving image reconstruction problem in pre 2000's is directly applying inverse Fourier transform $\mathcal{F}^{-1}$ on the zero padding k-space measurements $\tilde{y}$ [3][4]:

$$\hat{x} = \mathcal{F}^{-1}\tilde{y} \tag{2}$$

The classical method is efficient, but it cannot achieve high quality reconstruction from the subsampled data.

In 2006, Candès et al. [5] and Donoho et al. [6] proposed a revolutionary technique, Compressive sensing (CS), which could reconstruct signal from subsampled data based on its sparsity. From the theory of CS, the sparsity of signal under certain sparsifying transforms like wavelet or discrete gradient transform could be used for solving sparse regularization problems, such as a quadratically-constrained basis pursuit problem:

$$\hat{x} \in \underset{x \in \mathbb{C}^N}{\operatorname{argmin}} \|Hx\|_1 \quad \text{s.t.} \quad \|Ax - y\|_2 \leq \eta, \tag{3}$$

CSCI 8980 Think Deep Learning: Project Final Report.

where $x \in \mathbb{C}^N$ is an image, H is the sparsifying transform, $y = Ax$ is the underdetermined inverse problem and η is a fitting parameter. Many robust and efficient optimization methods such as fast iterative shrinkage-thresholding algorithm (FISTA) [7] and alternating direction method of multipliers (ADMM) [8] show good performance in solving problem (3).

Deep Learning in Image Reconstruction Deep learning (DL), a class of machine learning algorithms, is the state-of-the-art technique on solving various imaging tasks such as classification, segmentation, deconvolution,denoising and so on [9]. It could progressively extract high-level feature from the input image. The rapid development of deep learning in recent years showed the the potential to change the field of image reconstruction. Although CS is a mature technique in MRI reconstruction, Gottschling et al. [3] still demonstrated that deep learning could outperform CS is following aspects: implicitly encoding of complex image structures of DL never requires fixed image model as in CS; direct learning of reconstruction procedure in DL is not 'handcrafted' like CS; DL doesn't need the delicate tuning of parameters in the optimization algorithm.

Deep neural network architecture based on convolutional neural network (CNN) has been widely used in reconstructing MRI from subsampled k-space data [10][11][12][13]. U-Net architecture proposed by Ronneberger et al. [14] was the foundation of many deep learning methods for MRI reconstruction. It will be used as baseline model for image reconstruction in this project.

Instabilities in Deep Learning DL techniques are not perfect, as it has a serious robustness issue. In 2014, Szegedy et al. [15] found that a hardly perceptible perturbation in the input image could lead to misclassification of neural networks. This finding means generalization of DL to be difficult. Besides, Antun et al. [3][4] found that deep learning could yield unstable results in image reconstruction. Stability of deep learning is important, since the false positives and negatives in reconstruction would make the diagnosis unreliable. In their study, there are three types of instabilities: 1). Tiny worst-case perturbations in the original images could result in serious distortions in the recovered image. These perturbations comes from image domain (movement of patients and anatomic difference between people) and sampling domain (malfunctioning of equipment and inevitable noise from the physical model of scanning machine); 2). Small structural changes in the image could get removed from the recovered image; 3). Increasing the number of samples in the sampling device may deteriorates or stagnates the quality of recovered image. In this project, we will focus on the first case. Recently, Raj et al. [16] proposed an adversarial training strategy of end-to-end deep-learning-based image reconstruction model to improve robustness. The min-max formulation used our algorithm is based on this work.

2 Proposed Methods

2.1 Performance Metrics

The performance metrics [2] for the reconstruction results are Normalized Mean Square Error (NMSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM). The introduction of SSIM will be presented in section 4 because we also used it as the final training loss function.

$$\text{NMSE}\left(\hat{m}, m\right) = \frac{\left\|\hat{m} - m\right\|_2^2}{\left\|m\right\|_2^2}, \tag{4}$$

where $\hat{m}$ is the reconstructed image volume and m is the reference image volume.

$$\text{PNSR}\left(\hat{m}, m\right) = 10 \log_{10} \frac{\left[\max(m)\right]^2}{\text{MSE}\left(\hat{m}, m\right)}, \tag{5}$$

where $\max(m)$ is the maximum possible pixel value of the image and MSE $\left(\hat{m}, m\right) = \frac{1}{n}\left\|\hat{m} - m\right\|_2^2$ is the mean squared error (MSE) between reconstructed image $\hat{m}$ and reference image m.

2.2 Loss Function

Although MSE Loss is the most common loss function for deep learning training, where MSE$(\hat{m}, m) = ||\hat{m} - m||_2^2$ for two images $\hat{m}$ and m with pixel-wise comparison, for standard

UNet training, we found that SSIM [17] can be used as loss function and it usually outperform MSE Loss from many computer vision literature. Different to MSE, which estimates absolute errors, SSIM will compute inter-dependencies of pixels by calculating luminance (l), contrast (c) and structure (s). The formula for SSIM between sliding windows x and y of two images $\hat{m}$ and m is

$$\text{SSIM}(x,y) = [l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}], \tag{6}$$

where α, β, γ are weights usually set to 1 and

$$l(x,y) = \frac{2\mu_x \sigma_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \; c(x,y) = \frac{2\sigma_x \sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \; s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3}, \tag{7}$$

where μ and σ are mean and variance and the constant $c's$ are included to avoid instability with definition $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ and $c_3 = c_2/2$, where L is the dynamic range of the pixel values (e.g., 255 for 8-bit grayscale images). Then, the SSIM Loss can be defined as Structural Dissimilarity (DSSIM), where

$$\text{DSSIM}(x,y) = \frac{1 - \text{SSIM}(x,y)}{2} \tag{8}$$

Our experiment setting has windows size to be standard 11×11.

2.3 Adversarial Training Strategy

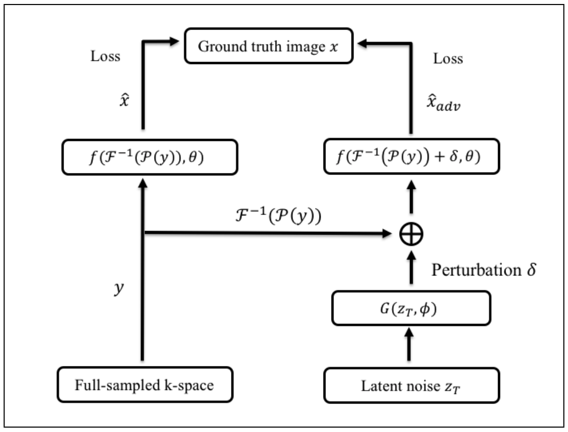


Figure 1: Adversarial training framework.

Algorithm 1 Minibatch training for robust reconstructor with generative perturbation

```
Input: Mini-batch samples $y_T, G_{T-1}, f_{T-1}$
Output: $G_T$ and $f_T$
$G_T = G_{T-1}$
Generate latent noise $z_T$
$G_T = \underset{G}{\text{argmax}}\ [\lambda_1 \text{Loss}(f_{T-1}(\mathcal{F}^{-1}\mathcal{P}(y_T) + G(z_T;\phi);\theta), \mathcal{F}^{-1}(y_T)) + \lambda_2 \max\{0, ||G(z_T;\phi)||_2^2 - \epsilon\}]$
$\delta_T = G_T(z_T)$
$f_T = \underset{f}{\text{argmin}}\ [\text{Loss}(f(\mathcal{F}^{-1}\mathcal{P}(y_T);\theta), \mathcal{F}^{-1}(y_T)) + \lambda_1 \text{Loss}(f(\mathcal{F}^{-1}\mathcal{P}(y_T) + \delta_T;\theta), \mathcal{F}^{-1}(y_T))]$
return $G_T, f_T$
```

Our new training strategy contains an auxiliary network which could generate adversarial examples during training. When MSE Loss is used, the loss function has the following form:

$$\begin{aligned} \min_{\theta} \max_{\phi} \text{Loss}\left(f(\mathcal{F}^{-1}\mathcal{P}(y);\theta), \mathcal{F}^{-1}(y)\right) + \text{Loss}\left(f(\mathcal{F}^{-1}\mathcal{P}(y) + G\left(z_T;\phi\right);\theta), \mathcal{F}^{-1}(y)\right) \\ + \lambda_2 \max\left\{0, \|G\left(z_T;\phi\right)\|_2^2 - \epsilon\right\}, \end{aligned} \tag{9}$$

where y, and $\mathcal{F}$ are described in previous section, $\mathcal{P}$ is the mask operation for full sampled k-space, z_T is the latent noise, f is the image reconstruction network, G is the network to generate additive perturbation, and ϵ is the perturbation radius for G. Besides, Loss can be any loss function to quantify the quality of two input in image domain (We tried MSE Loss and SSIM Loss in our project), θ stands for parameters for reconstructor and ϕ stands for parameters for generator. The first term is the standard MSE Loss or DSSIM Loss in x-space. The second term represents an adversarial game between f and G, where G will generate optimal perturbation which maximize the reconstruction loss while f will make itself robust to the perturbations. The third term is a hinge loss penalty to discourage large perturbation. Besides, some empirical changes are added into the training algorithm to build an adversarial robust f. Figure 1 shows the adversarial training framework, which consists of image reconstruction network f (U-Net) jointly with network G (Transposed Convolution) to generate additive perturbation.

Algorithms The algorithm with arbitrary loss function then can be written as Algorithm 1. The algorithm is inspired and modified from [16]. Firstly, we maximize the generator w.r.t ground truth images with an added regularization term to avoid the case that the perturbation becomes too large hence maximizing the possible perturbation to produce. Then, we calculate the perturbation from generator and minimize the loss between output from reconstructor with original image and ground truth image as well as the loss between output from reconstructor with original image plus perturbation and ground truth image. In the Algorithm 1, G, f, $\mathcal{P}$, y, z_{T}, Loss, θ and ϕ are defined in last paragraph. In real training case of the algorithm in deep learning framework, the gradients of generator will be computed by optimizing negative Loss with certain optimizer (e.g. Adam). For reconstructor, the gradient will be computed by optimizing positive Loss.

2.4 Network Architecture

Reconstructor: U-Net The training network is in an end-to-end manner to minimize the MSE with respect to the ground truth images [2]:

$$\min_{\theta} \frac{1}{2} \sum_{i=0}^{n} \left\| B_\theta \left(\tilde{m}^{(i)} \right) - m^{(i)} \right\|_2^2, \tag{10}$$

where $m^{(i)}$ is the ground truth image, $\tilde{m}^{(i)} = \mathcal{C}\left(\left| \mathcal{F}^{-1} \left(\mathcal{P} \left(y \right) \right) \right| \right)$ is the zero-filled image, and B_θ is the function to be learned by U-Net. Here $\mathcal{P}$ is the projection function makes all masked entries to be zero, $\mathcal{F}^{-1}$ is inverse Fourier transform and $\mathcal{C}$ is the linear operator for center cropping.

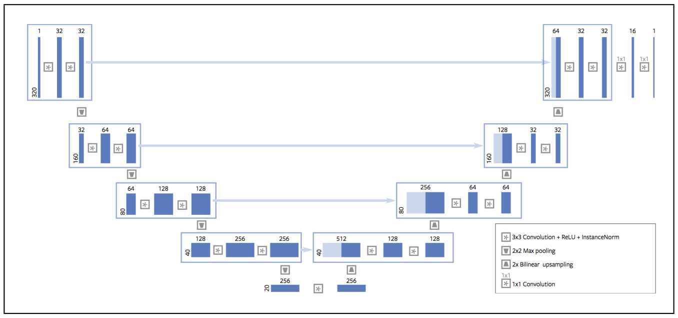


Figure 2: Reconstructor U-Net architecture [2][14].

Figure 2 [2][14] shows the U-Net architecture used for single-coil knee MRI reconstruction, which consists of a down-sampling (contracting) path and an up-sampling (expansive) path. The down-sampling path is based on the typical convolutional neuron network architecture, which could extracted features from dataset and reduce the number of parameters to be learned by the network. It consists of blocks of two 3×3 convolutions, each followed by a rectified linear unit (ReLU) and instance normalization. Each blocks are connected by a 2×2 max pooling operation with stride 2 to achieve down-sampling, which halve each spatial dimension. After the original image being down-sampled into latent space, the up-sampling path, which shows similar structure to the down-sampling path, will be used to obtain a image with same size as input. In each block of up-sampling path, there is a concatenation of the up-sampled activation from previous block, and an activation follows skip connection from the down-sampling path. Each block also consists of blocks of two 3×3 convolutions followed by a rectified linear unit (ReLU) and instance normalization. And blocks are interleaved with $2\times$ Bilinear up-sampling operator, which doubles the resolution of the result of block. At the final layer, a series of 1×1 convolutions map the 32-components feature to 1 without changing the spatial resolution of original image. As some small adjustments from original U-Net to make more stable training, we increased number of channels, pool layers to 64 and 5 respectively, PReLU [18] is used as activation function rather than ReLU and average pooling is used rather than max pooling.

Generator: Transposed Convolution Transposed Convolution first proposed by Zeiler, et al. [19] is used as our generator architecture with regularization layers. The architecture is also used by Radford et al. [20] to achieve the deep convolutional generative adversarial networks (DCGAN) in minimax application. It could map the latent space z_T to the image space, which means noise $G(z)$ with the same shape as the input image of U-Net will be generated. In our project, $G(z)$ has shape $(n_c \times 320 \times 320) = (1 \times 320 \times 320)$, where number of channels of the input image n_c is set to be 1 as grayscale images are used.

Figure 3 shows the Transposed Convolution example from DCGAN for adversarial attack, which consists of a sequence of fractionally-strided convolutions, each followed with a 2D batch norm layer and a PReLU activation. Last layer is paired with a tanh function to achieve a data range of $[-1, 1]$ to match our normalized pixel values. In our projects, 6 fractionally-strided convolutions were used to convert the latent space into a 320×320 pixel image.

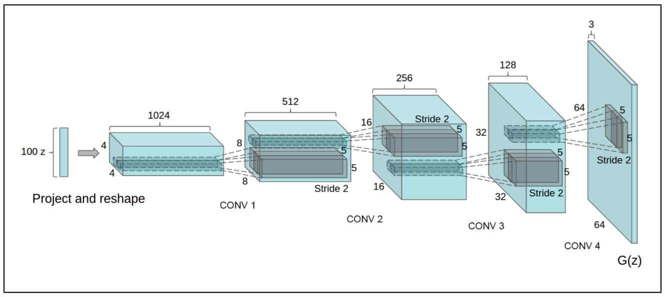


Figure 3: Generator DCGAN architecture [20].

3 Experiments

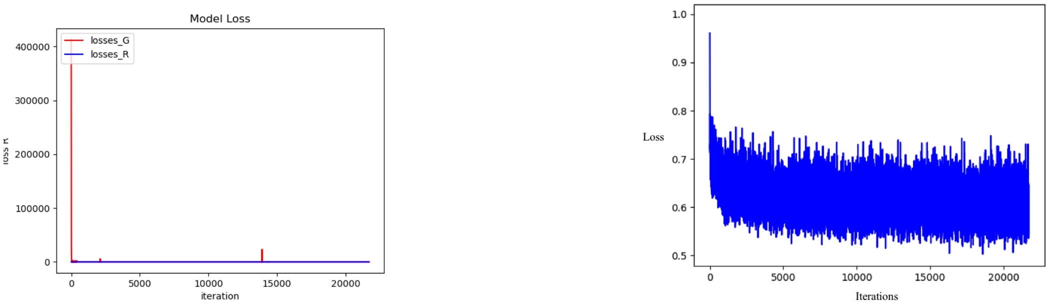
3.1 Datasets

We conduct our experiments on single-coil FastMRI dataset [2], which is stored as H5 with full sampled k-space. A mask (either random or equispaced) is applied to the original k-space to produce under-sampled k-space of our training sample x. The image domain is obtained by applying inverse Fourier transform on both under-sampled k-space and full sampled k-space. For training set, the dataset has 973 volumes with corresponding 34742 slices. For validation set, the dataset has 199 volumes with corresponding 7135 slices.

The H5 files in both training and validation set contain tensors `kspace`, `reconstruction_rss` and `reconstruction_esc`. `kspace` is the emulated single-coil k-space data, which has the tensor shape (number of slices, height, width). For example, shape can be $(38, 640, 368)$. The exact tensor shapes are different among files. `reconstruction_rss` and `reconstruction_esc` are two ground truth reconstructions for the single-coil datasets. `reconstruction_rss` is the root-sum-of-squares computed from multi-coil data, and `reconstruction_esc` is computed by the inverse Fourier transform on the single-coil data. All ground truth images are cropped to the central 320×320 pixel region.

3.2 Training Details

All networks were trained on two Nvidia Tesla V100 GPU. We used PyTorch for training and implementation of all models. Our generator has 7 Transposed convolution layers to upsample the latent noise from size 100 to worst-case noise in image domain with shape 320×320. Learning Rate of both models were set to 1×10^{-4} with AdamW optimizer [21][22] for reducing over-fitting with weight decay rate 0.01. Batch size of all training is set to be 8. From our attempts of getting stable results, the hyperparameters are set as followings: α_1 for weighting perturbation reconstruction loss is set as 1, α_2 for weighting hinge loss impact is set as -1 and ϵ for regularizing numerical value of perturbation squared norm in hinge loss is set as 1×10^3 or 5×10^3. Early stopping is applied when validation loss is not decreasing for 3 consecutive epochs. As the training is very time consuming (close to 2 hrs for one epoch for minimax training) and we observed that the algorithm converges fast with few epochs, we set maximum epoch to be 5. This leads to roughly 10 hrs of training for each attempt. Figure 4 shows the loss of generator and reconstructor vs. iterations and loss of reconstructor vs. iteration alone during adversarial training from the case $\epsilon = 1 \times 10^3$. Since the hinge loss regularization is applied, the generator loss (red) starts large and the loss change in figure (a) is hard to see. However, from numerical value observation of our experiment, we see that the reconstructor loss decreases and converge to equilibrium with generator loss after several iterations.



(a) Loss of reconstructor (blue) and generator (red) vs. iterations

(b) Loss of reconstructor vs. iterations

Figure 4: Loss changes on reconstructor and generator

4 Results and Discussion

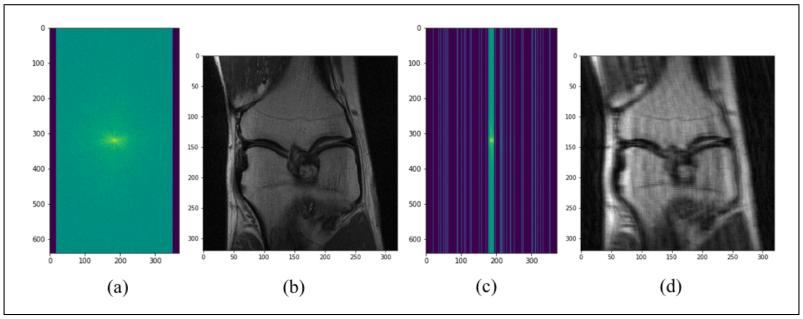
Source code of adversarial training and supporting information are submitted with the report.

Performance Summary Table 1 summarizes the reconstruction performance without or with adversarial training. Results are obtained by applying U-Net model on single-coil training and validation data. Different perturbation radiuses ϵ are used here. These results show our training could give reasonable reconstruction quality from subsampled data.

Perturbation Radius ϵ	Process & Methods	NMSE	PNSR	SSIM
	Training-S	0.21083	25.64586	0.64354
	Validation-S	0.23234	25.35955	0.62848
1×10^3	Training-A	0.19351	25.87200	0.63597
	Validation-A	0.21883	25.19485	0.60633
5×10^3	Training-A	0.22061	25.10864	0.61624
	Validation-A	0.23862	24.87306	0.60115

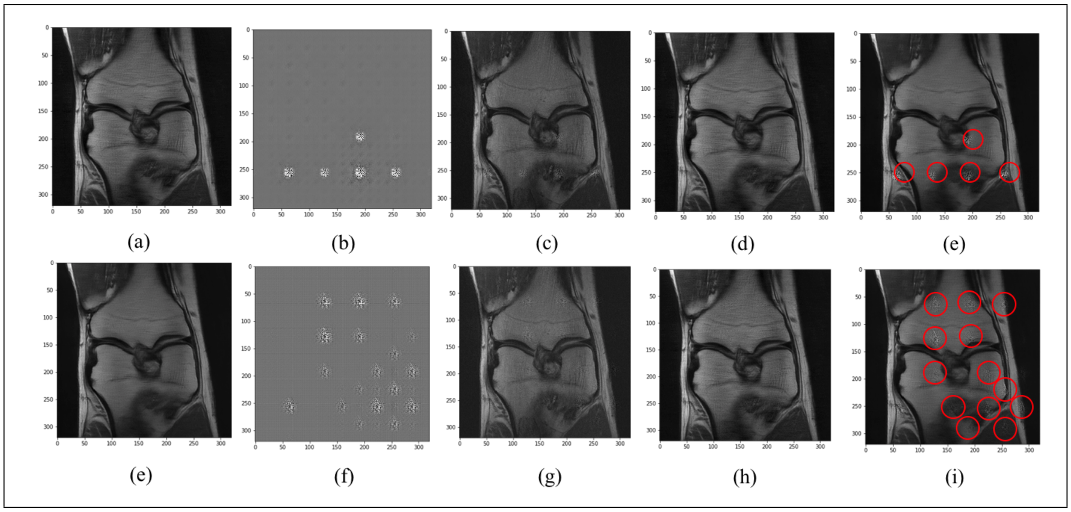
Table 1: Reconstruction performance on single-coil knee MRI datasets. Training-S and Validation-S represent the training and validation performance of standard U-Net without adversarial training. Training-A and Validation-A represent the training and validation performance of standard U-Net with adversarial training.

Visualization of Input Data Figure 5 (a) and (c) show the absolute value of full-sampled and subsampled k-space data, respectively. The acceleration rate is four, which requires other columns in k-space are selected randomly with probability $P = (\frac{N}{4} - N * 8\%)/(N - N * 8\%)$, where N is the number of columns in k-space. This is to ensure the expected number of columns in k-space is equivalent to $N/4$. Figure 5 (b) and (d) represent the inverse Fourier transform of (a) and (c), respectively. The low quality image (d) will be cropped into a 320 × 320 image and used as the input of U-Net.



(a) (b) (c) (d)

Figure 5: Visualization of k-space and input image. (a) Absolute value of full-sampled k-space data (b) Inverse Fourier transform of (a). (c) Absolute value of subsampled k-space data (d) Inverse Fourier transform of (c).



(a) (b) (c) (d) (e)

(e) (f) (g) (h) (i)

Figure 6: Results visualization. Top row represent results with perturbation radius $\epsilon = 1 \times 10^3$, while bottom row represent results with perturbation radius $\epsilon = 5 \times 10^3$: (a)(e) Reconstruction from standard input by using U-Net with adversarial training. (b)(f) Visualization of perturbations. (c)(g).Ground-truth image with perturbations. (d)(h) Reconstruction from input with perturbations by using U-Net with adversarial training. (e)(i) Reconstruction from input with perturbations by using U-Net without adversarial training. Red circles represent the noise in the reconstruction image.

Stability with Respect to Perturbations We perturbed the zero filled image (described in section 2.4) $\tilde{m}^{(i)} = \mathcal{C}\left(\left|\mathcal{F}^{-1}\left(\mathcal{P}\left(y\right)\right)\right|\right)$ with perturbations generated from Transposed Convolution. Figure 6 is used for illustrated how the worst case perturbation could affect reconstruction results and how adversarial training can help minimize the noise. Figure 6 (b) and (f) are the worst case perturbations generated from the adversarial training with perturbation radius $\epsilon = 1 \times 10^3$ and $\epsilon = 5 \times 10^3$, respectively. Both perturbations consist of some big white circles and many small noises that are less visible than the white ones. The perturbations represented here are not as tiny as described in Antun et al.'s work [4]. In order to achieve tiny (imperceptible) perturbations, smaller perturbation radius ϵ is needed. Since the training process is expensive, we pick larger ϵ to give better demonstration of robustness issue in limited time. Figure 6 (a) and (e) are the ground-truth images with perturbations. Perturbations are visible in (a) and (e) since ϵ is large. Figure 6 (d) and (h) represents the reconstruction from the subsampled data with perturbations by using U-Net with adversarial training, while Figure 6 (e) and (i) represents the reconstruction from the subsampled data by using U-Net with adversarial training. We could clearly see that the reconstruction images from the subsampled data by using U-Net without adversarial training still have white circles from the perturbations. Taking a comparison with closer look between (c)(g) and (e)(i), it can be observed that the baseline model even enhanced the added perturbation rather than trying to remove them. However, as shown in Figure 6 (d) and (h), there are no visible noises in the reconstruction. This means the reconstructor after adversarial training is robust to the worst-case perturbation. It should be noticed that neither our experiment nor the method proposed by Antun et.al.[3][4] can perfectly simulate the perturbation in real world for patient movement or equipment malfunctioning since both methods find perturbation by gradient ascent based on the reconstructor neural networks. However, our experiment shows that adversarial training can be an effective way of mitigating the instability problem under the assumption that the distribution of perturbation is simulated. If there is enough data of perturbation generated from real world to simulate the true distribution, it could be possible that adversarial training could mitigate the instability problem to get wide adaptation on medical industry.

Novelty Involved Even though there had been many discussions about the possibility of solving instability with minimax optimization in the medical image processing community, there were no many attempts on actual experiment. We only found one recent paper [23] referred to use adversarial attack on improving MRI reconstruction robustness with a different proposed method. The successful experimental result of this kind of problem would be much more difficult to achieve than theoretical derivation because of various problem may emerge from minimax optimization such as mode collapse and non-convergence. Although the formulation we used is not completely novel, the original paper [16] that proposed this method didn't attempt their solution on MRI data and we made several adjustments to make it work on actual experimental training.

5 Future Works

In our current project, the worst-case perturbation is learned and added to image domain. However, another way to create the perturbation is that the perturbation can be learned from k-space and added to the sampled k-space directly. We considered to come up with an experiment on that, but time limit constrained our further investigation by now. For creating perturbation on k-space, since k-space has complex number with both real part and imaginary part, one way of creating the worst-case perturbation generator with CNN would be having two kernels to generate result with two channels, whereby one channel represents real part and another represents imaginary part. In this case, the worst-case perturbation can be learned by gradient ascent by loss function $||f(\mathcal{F}^{-1}(\mathcal{P}(y_T)+G(y_T;\theta))) - f(\mathcal{F}^{-1}\mathcal{P}(y_T);\theta)||_2^2 - \lambda_2||G(y_T;\phi)||_2^2$, where the definition of notations are the same as our previous experiment. As k-space perturbation is obtained, the perturbation need to be added to original sub-sampled k-space with algorithm modified to be.

Algorithm 2 Minibatch training for robust reconstructor with generative perturbation

```
Input: Mini-batch samples $y_T, G_{T-1}, f_{T-1}$
Output: $G_T$ and $f_T$
$G_T = G_{T-1}$
$G_T = \text{argmax}_G \lambda_1||f_{T-1}(\mathcal{F}^{-1}(\mathcal{P}(y_T)+G(y_T;\phi));\theta)-\mathcal{F}^{-1}(y_T)||_2^2+\lambda_2 \max\{0, ||G(y_T;\phi)||_2^2 -\epsilon\}$
$\delta_T = G_T(y_T)$
$f_T = \text{argmin}_f||f(\mathcal{F}^{-1}\mathcal{P}(y_T);\theta) - \mathcal{F}^{-1}(y_T)||_2^2 + \lambda_1||f(\mathcal{F}^{-1}(\mathcal{P}(y_T) + \delta_T);\theta) - \mathcal{F}^{-1}(y_T)||_2^2$
return $G_T, f_T$
```

Moreover, as we observed that the single-coil images from this dataset has varying quality and some of them have very poor image quality, some more preprocessing such as removing low quality during training can further improve the performance.

References

[1] Zhang, Zizhao, et al. "Reducing uncertainty in undersampled mri reconstruction with active acquisition." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2019.

[2] Zbontar, Jure, et al. "fastMRI: An open dataset and benchmarks for accelerated MRI." *arXiv preprint arXiv:1811.08839* (2018).

[3] Gottschling, Nina M., et al. "The troublesome kernel: why deep learning for inverse problems is typically unstable." *arXiv preprint arXiv:*2001.01258 (2020).

[4] Antun, Vegard, et al. "On instabilities of deep learning in image reconstruction and the potential costs of AI." *Proceedings of the National Academy of Sciences* (2020).

[5] Candès, Emmanuel J., Justin Romberg, and Terence Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information." *IEEE Transactions on information theory 52.2* (2006): 489-509.

[6] Donoho, David L. "Compressed sensing." *IEEE Transactions on information theory 52.4* (2006): 1289-1306.

[7] Beck, Amir, and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM journal on imaging sciences 2.1* (2009): 183-202.

[8] Boyd, Stephen, Neal Parikh, and Eric Chu. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Now Publishers Inc*, 2011.

[9] Ongie, Gregory, et al. "Deep learning techniques for inverse problems in imaging." *IEEE Journal on Selected Areas in Information Theory* (2020).

[10] Hammernik, Kerstin, et al. "Learning a variational network for reconstruction of accelerated MRI data." *Magnetic resonance in medicine* 79.6 (2018): 3055-3071.

[11] Han, Yoseob, and Jong Chul Ye. "Framing U-Net via deep convolutional framelets: Application to sparse-view CT." *IEEE transactions on medical imaging* 37.6 (2018): 1418-1429.

[12] Hyun, Chang Min, et al. "Deep learning for undersampled MRI reconstruction." *Physics in Medicine & Biology* 63.13 (2018): 135007.

[13] Schlemper, Jo, et al. "A deep cascade of convolutional neural networks for MR image reconstruction." *International Conference on Information Processing in Medical Imaging.* Springer, Cham, 2017.

[14] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention.* Springer, Cham, 2015.

[15] Szegedy, Christian, et al. "Intriguing properties of neural networks." *arXiv preprint arXiv:1312.6199* (2013).

[16] Raj, Ankit, Yoram Bresler, and Bo Li. "Improving Robustness of Deep-Learning-Based Image Reconstruction." *arXiv preprint arXiv:*2002.11821 (2020).

[17] Wang, Zhou; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. (2004-04-01). "Image quality assessment: from error visibility to structural similarity". *IEEE Transactions on Image Processing.* 13 (4): 600–612

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". *arXiv:1502.01852*

[19] Zeiler, Matthew D., et al. "Deconvolutional networks." *Computer Vision and Pattern Recognition (CVPR), 2010* IEEE Conference on. IEEE, 2010.

[20] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

[21] Ilya Loshchilov, Frank Hutter. "Decoupled Weight Decay Regularization". *In Proceedings of 2019 International Conference on Learning Representations (ICLR 2019)*

[22] Diederik P. Kingma, Jimmy Ba. "Adam: A Method for Stochastic Optimization". *In Proceedings of 2015 International Conference for Learning Representations (ICLR 2015)*

[23] Francesco Calivá, Kaiyang Cheng, Rutwik Shah, Valentina Pedoia. "Adversarial Robust Training in MRI Reconstruction". *arXiv:2011.00070*