Spring 5-21-2021

Data-Driven Approaches to Solve Inverse Problems

Peijie Qiu
Washington University in St. Louis

Washington University in St. Louis

Mckelvey School of Engineering

Department of Computer Science and Engineering

Thesis Examination Committee:
Umberto Villa, Co-Chair
Ulugbek Kamilov, Co-Chair
Netanel Raviv

Data-Driven Approaches to Solve Inverse Problems

by

Peijie(Ricardo) Qiu

A thesis presented to the McKelvey School of Engineering
of Washington University in partial fulfillment of the
requirements for the degree of

Master of Science

May 2021
Saint Louis, Missouri

Contents

List of Tables

List of Figures

Acknowledgments

A special thanks goes to the many graduate students and distinguished faculty within my department who have reviewed this thesis and helped support the related research.

Peijie(Ricardo) Qiu

Washington University in Saint Louis
May 2021

ABSTRACT OF THE THESIS

Data-Driven Approaches to Solve Inverse Problems

by

Peijie(Ricardo) Qiu

Master of Science in Computer Science

Washington University in St. Louis, May 2021

Research Advisor: Professor Umberto Villa & Ulugbek Kamilov

The main purpose of this thesis is to discuss data-driven approaches to solve inverse problems in image reconstruction. In the Bayesian framework, the image prior serves as a regularizer in the computation of a maximum-a-posterior estimation of the reconstructed image. Classical image priors include Gaussian random space(e.g. Tikhonov regularization) or Besov prior (e.g. Total Variation regularization). Inspired by generative adversarial networks, a critic (discriminator) can serve as a regularizer, because of its capability of distinguishing the distribution of the ground-truth images from the distribution of the naively reconstructed images with classical regularization functional. Another data-driven approach, regularization by denoising (RED), provides a flexible and effective way to combine the state-of-the-art denoisers and model-based methods with a variety of optimization strategies to solve the inverse problem. Unlike traditionally hand-crafted regularizers, the data-driven regularization has the potential to learn an optimal regularizer from the data. In this thesis, we will consider two widely used linear forward models, and two data-driven approaches to solve inverse problem: adversarial regularizer and regularization by denoising.

Notation

The notations which will be used in the remain of this thesis.

$d \leftarrow$ *the measurement data*

$m \leftarrow$ *the sought-after parameter or image*

$A \leftarrow$ *the linear forward operator*

$n \leftarrow$ *the additive noise*

$(n_x, n_y) \leftarrow$ *pixel index for each image pixel*

$\theta \leftarrow$ *trainable parameters in the paraterized adversarial regularizer*

$R_\theta \leftarrow$ *the parameterized regularizer*

$f \leftarrow$ *denoiser*

$\sigma \leftarrow$ *the additive Gaussian noise standard deviation*

Chapter 1

Introduction

In this section, the inverse problem is introduced. We also provide analysis of the ill-posedness of inverse problem and how it can be formulated as optimization problem. Next, modern data-driven approaches will be introduced to improve image quality including the construction of advanced regularization functionals. Related work in solving the inverse problem using data-driven approaches will also be reviewed as the motivation for this project. Finally, the outline of the whole thesis will be presented.

1.1 Inverse Problems

In many medical imaging modalities, for example computed tomography, the direct measurement that we can obtain through the imaging system is not the image itself. The mapping from the sought-after parameter (image) to the measurement data is generally defined as forward model. The measurement data is often corrupted by noise. Thus, the measurement data are often modeled as

$$d = F(m) + n, \tag{1.1}$$

where d is the measurement data, F is the forward model, m is the sought-after parameter, n is the noise. The goal of an inverse problem is to reconstruct m given the noisy measurement d and the forward model F.

1.1.1 Ill-posedness

Solving the inverse problem is non-trivial because of its ill-posedness. According to the Hadmard's definition, a well-posed inverse problem satisfies the three conditions below,

1. The solution m always exists for all the measurement d. (Existence)
2. the solution m is unique for all d. (Uniqueness)
3. The solution is stable with respect to small perturbations, which means the solution m changes continuously with respect to the measurement d. (Stability)

In contrast, if one of the above properties is not satisfied, the inverse problem is ill-posed.

For the linear forward operators which are used in this thesis, the goal is to find the solution m such that

$$d \approx Am, \tag{1.2}$$

where A denotes the linear forward operator. This problem is generally ill-posed, because in practical applications, the stability of the solution is hard to achieve.

1.2 Inverse Problems as Optimization

To overcome the instability of the ill-posed inverse problems, we reformulate the inverse problem as optimization problem by incorporating a regularization term. For a properly chosen regularization function, this approach gives raise to a convex deterministic optimization problem, which then admits a unique solution. However, the quality of the solution depends on the choice of the regularization. This motivates us to find an advanced regularization functional based on our prior information about the image.

1.2.1 Data Fidelity & Regularization

The variational formulation of an inverse problem can be written as

$$m^* = \arg\min_m \frac{1}{2} \parallel Am - d \parallel_2^2 + \lambda R(m), \tag{1.3}$$

where $\frac{1}{2} \parallel Am - d \parallel_2^2$ is the data fidelity term, $R(x)$ is the regularization term, and λ controls the regularization strength. The data fidelity term measures the data consistency of the solution m^* to the observation d. The regularization term imposes the desired prior knowledge onto the solution. For example, Tikhonov regularization induces a solution with small norm, generalized Tikhonov regularization produces a smooth solution, Total Variation regularization preserves edges. However, these regularization also introduce bias in the reconstructed images. Tikhonov does not capture well edges, Total Variation leads to prefer piece-wise constant images, thus introducing "cartoon" like artifact in the reconstruction.

1.2.2 Data-Driven Approaches for Finding Optimal regularization

To overcome the above challenges, in this thesis we explore approaches to learn optimal regularization functionals from the data. The methods we introduce are related to the adversarial learning regularizer and regularization by denoising (RED).

The basic hypothesis of adversarial regularizer is that the ground-truth images m follows an unknown distribution $\mathbb{P}_{m_r}$. The images m_{recon} reconstructed using any classical method follows a distribution $\mathbb{P}_{m_{recon}}$. According to the theory of adversarial learning, we can learn a critic in a semi-supervised manner to differentiate the true distribution $\mathbb{P}_{m_r}$ from $\mathbb{P}_{m_{recon}}$, outputting small values for samples from true distribution $\mathbb{P}_{m_r}$ while larger values for samples from $\mathbb{P}_{m_{recon}}$. Then the learned critic can be used as a regularization functional in the variational regularization method. The solution can be obtained using the numerical optimization methods that will be discussed in Chapter 2.

Regularization by denoising (RED)combines state-of-the-art denoiser with the model-based method for image reconstruction. RED enables the flexibility to choose among many denoisers, from median filter to CNN-based denoiser, to regularize the inverse problem.

Although the CNN-based denoiser is generally trained in a supervised way, adding the noise to the ground-truth image to generate the training dataset is a trivial process.

Another advantage of RED is that the cost function in the reconstruction problem is well-defined and convex, which guarantees converge to the global optimum. A family of convex optimization methods can then be used to solve this convex minimization problem, for example, fixed-point solution, gradient-based method, etc.

1.2.3 Related Work

Learning image priors has gained a lot of attention in solving inverse problems in the past decade. Previous work can be characterized as either discriminative supervised learning methods or the adversarial unsupervised learning methods. Both approaches incorporate physical models and numerical optimization methods to produce the final solution.

Supervised learning

Zhang et al.[1] proposed learning the residual mapping between the true image and the observation. The learned network can serve as a denoising prior. It can also be plugged into the variational regularization model to solve the inverse problem by variable splitting optimization strategies, for example half-quadratic splitting.

Jin et al.[2] proposed learning a deep convolutional neural network (U-net) which inputs a direct inversion of the computed tomography using filtered back projection (FBP) method and regresses the FBP result directly to the ground-truth images. Once the network is trained, it is plugged into different iterative method,(ISTA[3], FISTA[4], ADMM[5]) to solve the inverse problem. The learned network is able to remove the artifacts caused by the direct inversion using filtered back project method, and hence, is effective to solve the inverse problem.

Venkatakrishnan et al.[6] introduced a powerful and flexible framework, Plug-and-Play(PnP), for inverse problems in image reconstruction. This model-based method is able to combine

different state-of-the-art priors (for example, priors learned by a neural network) with different forward models in different tasks to simplify the design of image reconstruction methods while providing a variety of possible solutions. The underlying principle of this methodology is to take advantage of the Alternating Direction Method of Multipliers (ADMM) that decouples the optimization of data fidelity terms and regularization terms. The resulting method minimizes each term separately and alternatively. However, the PnP framework has no explicit cost function making the convergence and convexity analysis difficult.

Romano et al. [7] mitigated this issue by proposing the regularization by denoising(RED) method. This method builds on the top of adaptive image filters. By deploying this methodology, we can formulate an explicit convex cost function, while can be minimized with different optimization algorithms, such as gradient descent, ADMM, Fixed point iteration. With this approach, a variety of state-of-the-art denoisers, such as deep denoising convolutional neural network(DnCNN)[8, 9], can be chosen to be a regularizer.

Unsupervised learning

Lunz et al.[10] proposed the concept of learning an adversarial regularizer in an unsupervised setting. This work is inspired by the critic in Wasserstein generative adversarial networks (WGANs [11, 12]), which is able to tell apart the true distribution of the ground-truth images from a fake distribution. Therefore, a critic can be learned to replace the variational regularization term in the physical model to solve the inverse problem iteratively.

Mukherjee1 et al.[13] extends the idea of adversarial regularizer to learn convex adversarial regularizers. They propose a network called the data-adaptive input-convex neural network (ICNN) which serves as the regularizer. Because the data fidelity term is convex in the physical model, if the regularizer is convex as well, the whole problem will be a convex optimization problem. Families of convex optimization methods can be hence applied to solve the inverse problem iteratively. The convergence of the iterative optimization method can be analyzed with standard methods.

1.3 Contribution of this thesis

The main contribution of this thesis is the implementation of a computational framework for solving inverse problems using data-driven regularization functionals using PyTorch[14]. Automatic differentiation of PyTorch eases the way to hand-craft the gradient of the cost function, especially when the neural networks are involved in solving the inverse problem iteratively. Besides, we compare the performance of several regularization methods from classical regularizations to data-driven approaches.

1.4 Thesis Outline

In the remaining of this thesis, we will introduce two widely used forward models and model-based reconstruction method using standard regularization in Chapter 2. In chapter 3, adversarial regularizer will be introduced. In chapter 4, we will introduce the regularization by denoising (RED). In the chapter 5, the conclusion of this research will be discussed.

Chapter 2

Model-based Approaches to Image Reconstruction

In this chapter, we present two commonly used imaging operators (also called forward models): the Gaussian Blurring operator, and the Radon transform. Reconstructing images from their measurements is formulated as an optimization problem. Finally, different combinations of optimizers and regularization terms are presented to solve the corresponding optimization problem via PyTorch[14]. The image reconstruction algorithms are of vital importance to our proposed method on Chapter 3 to generate the training dataset online.

2.1 Forward Model

The forward model is a map from the sought image or parameter to a measurable quantity. The forward model is usually a physical model, such as the equations govern propagation of energy, the absorption of certain substance, and etc. For example, the X-ray is used to capture the measurable quantity of computed tomography. The energy of the X-ray is attenuated by the tissues, organs, and other objects within a human body, then the measurable quantity through any inhomogeneous object can be used to reconstruct the X-ray image. In this section, two common linear forward operators, Gaussian blurring operator and Radon transform, are presented.

2.1.1 Gaussian Blurring

Image blurring is one of the most common forward models or imaging operators encountered while capturing images. It is caused by motion, out of focus, or other physical variations. Generally, image blurring is modeled by convolving the image m with a kernel k:

$$d(x,y) = \int_{\Omega} k(x-x', y-y')m(x',y')dx'dy', \tag{2.1}$$

where Ω is the domain of the image m, d denotes the measurement data. Mathematically, a Gaussian kernel is defined as:

$$k(x,y) = C \ \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \tag{2.2}$$

$\sigma > 0$ controls the width of the kernel, and C is normalization parameter.

Convolving an image with a Gaussian kernel is equivalent to weighted average over a patch of neighbor pixels. For numerical computation, a Gaussian kernel can be discretized as a $N \times N$ 2-dimensional matrix K where each element

$$K_{i,j} = h \ C \ \exp\left(-\frac{((i-j)h)^2}{2\sigma^2}\right) \ 1 \leq i,j \leq N, \tag{2.3}$$

where $h = \frac{1}{N}$, N is the kernel size. A Gaussian kernel is also separable which means convolution with a 2D Gaussian kernel can be implemented by convolving with two 1D Gaussian kernels, separately, reducing computational cost.

As a result, the blurring measurement d can be computed as the discretized convolution between m and a Gaussian kernel K

$$d[n_x, n_y] = \sum_i \sum_j m[n_x - i, n_y - j]K[i,j], \tag{2.4}$$

where (n_x, n_y) is the index of each pixel. The discrete convolution, however, is in essence a matrix multiplication by taking advantage of the Toeplitz matrix and unrolling m and d into 1-dimensional. Therefore, the measurement d can be expressed as the linear combination of

the ground-truth image m

$$d = A_k m, \tag{2.5}$$

where A_k is a doubly block circulant matrix, a special case of Toeplitz matrix.

2.1.2 Radon Transform

The Radon transform is used to obtain computed tomographic scans, which is a projection of the original sought-after image. Therefore, the inversion of Radon transform provides a way to reconstruct the original image and hence, the tomographic tomography.

The Radon transform is defined as an set of integrals along paths. The most straightforward path is a set of straight lines, which corresponds to the X-ray transform. The Radon transform is defined as

$$\mathcal{R}f(L) = \int_L f(x)\, dx, \tag{2.6}$$

where f is an unknown attenuation coefficients, L is a straight line, commonly the X-ray path.

For a X-ray tomography, the received energy I at the detector location can be expressed as an attenuation process according to Beer' Law,

$$I = I_0 e^{m(x)\Delta x}, \tag{2.7}$$

where I_0 is the incident energy at the source, m is the attenuation coefficient(tomographic image), Δx is the length of the X-ray through the object. According to equation(1.7), for any point x in any inhomogeneous object, the attenuation of a monochromatic X-ray can be modeled as

$$I = I_0 \exp\left\{ - \int_L m(l) dl \right\}, \tag{2.8}$$

where L dentoes the path of the X-ray through the object. In practice, q parellel X-rays are used. Assuming each source energy is identical, for each X-ray the measured quantity is the ratio between source energy I and the incident energy I_i on the receiver plane, that is

$$\int_{L_i} m(l) dl = -\ln\left(\frac{I_i}{I_0}\right), \ i = 1, \dots q, \tag{2.9}$$

where L_i is the path of X-ray i through the object. As indicated by equation(1.9), the Radon Transform of the attenuation coefficients m equals to ratio that we can measure.

To discretize the integral, an inhomegeneous object is approximated by N homogeneous materials in the object. Then ratio between the incident energy and source energy is discretized as

$$\begin{aligned} -\ln\left(\frac{I_i}{I_0}\right) &= \int_{L_i} m(l)dl \\ &= \sum_{j=1}^{N} m_j \Delta x_j \end{aligned}, \tag{2.10}$$

where Δx_j is the length of the X-ray through material j. Let us further assume, for each pixel at location (n_x, n_y) on the image place, the attenuation coefficients m_{n_x,n_y} is constant. Then the ratio can be further rewritten as

$$-\ln\left(\frac{I_i}{I_0}\right) = \sum_{(n_x,n_y)\in ray_i} m_{n_x,n_y} \Delta L^i_{n_x,n_y}, \tag{2.11}$$

where $\Delta L^i_{n_x,n_y}$ denotes the length of ray i at pixel (n_x, n_y). If the pixels on the image are stacked into a single vector with each element $m_j = m_{n_x,n_y}$, the lengths of all X-rays through an object can be represented as a matrix $A \in \mathbb{R}^{q\times N}$, those entries are given by

$$A_{i,j} = \begin{cases} \Delta L^i_{n_x,n_y} & \text{if } m_{n_x,n_y} \in ray_i \\ 0 & \text{otherwise} \end{cases}. \tag{2.12}$$

Therefore, the forward operator can be linearized by the equation

$$d = Am. \tag{2.13}$$

Notice that the matrix A is sparse. It has only $O(\sqrt{N})$ non-zero elements per row.

2.2 Reconstruction as Optimization

As we described in Chapter 1; the image formation process can be modeled as ,

$$d = F(m) + n, \tag{2.14}$$

where the d is the measurement data, m is the sought-after image, F is the forward model, n is the noise. The reconstruction problem can then be casted as maximizing the posterior probability of ground-truth image m given its measurements d by use of the Bayesian framework, we have

$$\begin{aligned} m^* &= \arg\max_m \ \mathbf{Pr}_{post}(m|d) \\ &= \arg\max_m \ \mathbf{Pr}_{like}(d|m)\mathbf{Pr}_{prior}(x) \\ &= \arg\min_m \ -\log \mathbf{Pr}_{like}(d|m) - \log \mathbf{Pr}_{prior}(m), \end{aligned} \tag{2.15}$$

where the negative log-likelihood $-\log \mathbf{Pr}_{like}$ is related to data fidelity term measures the difference between m and $F(x)$, while the negative log-prior $-\log \mathbf{Pr}_{prior}$ is related to the regularization term $R(x)$. Then the equation (2.3) can be rewritten as

$$m^* = \arg\min_m \frac{1}{2} \parallel F(m) - d \parallel_2^2 + \lambda R(m), \tag{2.16}$$

which turns the image reconstruction as an optimization problem. The parameter λ controls the strength of the regularization and $R(m)$ corresponds to the image prior. Two widely used regularization methods are discussed in this section: Tikhonov regularization and Total Variational regularization.

2.2.1 Tikhonov Regularization

Tikhonov regularization is one of the most commonly used regularization terms in ill-posed inverse problems. Tikhonov regularization can be derived probabilistically by making two assumptions: the ground-truth discrete image or parameter m is a Gaussian random varible with zero mean and covariance matrix $C \in \mathbb{R}^{N\times N}$, $m \sim \mathcal{N}(0,\, C)$; the noise term n

is additive Gaussian noise with zero mean and a covariance matrix $\Sigma \in \mathbb{R}^{q\times q}$, $n \sim \mathcal{N}(0, \Sigma)$. Then according to equation (1.15), the reconstruction problem can be rewritten as

$$\begin{aligned} m^* &= \arg\min_m \; -\log \mathbf{Pr}_{like}(d|m) - \log \mathbf{Pr}_{prior}(m) \\ &= \arg\min_m \; -\log \frac{1}{\sqrt{(2\pi)^q det(\Sigma)}} \exp\left(-\frac{1}{2} \parallel F(m) - d \parallel^2_{\Sigma^{-1}}\right) \\ &- \log \frac{1}{\sqrt{(2\pi)^N det(C)}} \exp\left(-\frac{1}{2} \parallel m \parallel^2_{C^{-1}}\right) \\ &= \arg\min_m \frac{1}{2} \parallel F(m) - d \parallel^2_{\Sigma^{-1}} + \frac{1}{2} \parallel m \parallel^2_{C^{-1}} \end{aligned} . \tag{2.17}$$

The Tikhonov regularization term corresponds to $R(m) = \frac{1}{2} \parallel m \parallel^2_{C^{-1}} = \frac{1}{2} \parallel \Gamma m \parallel^2_2$ with $\Gamma = C^{-\frac{1}{2}}$, which is usually defined as generalized Tikhonov regularization.

The generalized Tikhonov regularization term is therefore formally defined as

$$R_{Tik}(m) = \frac{1}{2} \parallel \Gamma m \parallel^2_2, \tag{2.18}$$

when Γ is an identity matrix, the Tikhonov regularization is reduced to a L_2 penalty term. In cases where Γ is the discretization of a differential equation, it can impose smoothness of the image. The drawbacks of Tikhonov regularization is also obvious. The Tikhonov regularization loses sharp edges and details as it smoothes out them.

2.2.2 Total Variation Regularization

Unlike Tikhonov regularization term which uses the L_2 norm, the total variation regularization takes advantage of the L_1 norm to impose sparsity. Sparsity in the gradients is used to preserve edge information. The total variation is defined as

$$R_{tv}(m) = \int_\Omega |\nabla_x m(x)| dx, \tag{2.19}$$

This discretion of T.V. in 2D space domain is given by

$$R_{tv}(m) = \sum_{n_x, n_y} \sqrt{|m_{n_x+1, n_y} - m_{n_x, n_y}|^2 + |m_{n_x, n_y+1} - m_{n_x, n_y}|^2}, \tag{2.20}$$

where (n_x, n_y) denotes the index of each pixel. In the optimization phase, this discretized total variation method is intractable by gradient-based iterative methods, because of its non-differentiability. Thus, the sub-gradient method or primal-dual method must be used. A variant of T.V. is the anisotropic total variation, defined as

$$R_{tv-aniso}(m) = \sum_{n_x,n_y} |m_{n_x+1,n_y} - m_{n_x,n_y}| + |m_{n_x,n_y+1} - m_{n_x,n_y}|, \tag{2.21}$$

, which is easier to optimize. While as all the anisotropic function, the anisotropic total-variation norm is variant to rotations preserving the edges aligned with the coordinate axes. Efficient proximal methods exists for anisotropic T.V., however, this functional is not rationally invariant.

The total variation is intuitive. Statistically, the absolute gradients of the majority of the natural image are close to zero except for the high-frequency edges, details and other unwanted components. Therefore, by reducing the total variation, the reconstructed signal has a high probability to eliminate possibly unwanted components and hence to reconstruct the ground-truth.

2.2.3 L-Curve

Finding the regularization strength is an empirical question, because there is no canonical way to measure the quality of the reconstructed images. The quality of reconstructed signal is traded-off between the data fidelity term and regularization term. The higher the regularization is, the more the information will be lost. In the contrast, smaller regularization may cause to instability. Ideally, an equilibrium should be achieved by balancing out the data fidelity term and regularization term to obtain both visually accurate and quantitatively accurate reconstructions. In addition, different images may need different regularization strengths which make choosing the regularization strength a challenging task. L-curve criterion will be introduced in this part to mitigate this problem.

The methodology of L-curve criterion is straight-forward. For different regularization strengths, the norm of the data residual is plotted against the norm of the regularization term in a logarithmic scale. As informed by its name, this curve is generally L-shaped. The

corner on the L-curve suggests the optimal regularization strengths among all the candidates. Although no theoretical guarantees exist for this method, it provides a heuristic way to search out the regularization strength.

2.3 Overview of Numerical Optimization Methods

To solve the optimization problem, many iterative methods exist. The most commonly used is gradient-based method, specially, when the cost function is convex, for example, using either Tikhonov regularization or Total Variation regularization. The gradient method can be guaranteed to converge to its global optimum.

2.3.1 L-BFGS Optimizer

L-BFGS[15] is a powerful numerical optimization method for solving large-scale optimization problems. It is a Quasi Newton's method which efficiently approximates the inverse of the Hessian using historical information. The L-BFGS also extends the Broyden–Fletcher–Goldfarb–Shanno (BFGS[16]) algorithm while mitigating its shortcoming of explicitly stores a dense inverse Hessian matrix. The update rule of BFGS is formally defined as

$$x_{k+1} = x_k - \alpha_k B_k g_k, \tag{2.22}$$

where α_k is the step size at the kth iteration, B_k is the inverse Hessian matrix, g_k is the derivative. The inverse Hessian matrix is further updated as

$$B_{k+1} = V_k^{\mathrm{T}} B_k V_k + \rho_k s_k s_k^{\mathrm{T}}, \tag{2.23}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, $\rho_k = \frac{1}{y_k^{\mathrm{T}} s_k}$, $V_k = I - \rho_k y_k s_k^{\mathrm{T}}$. In the BFGS, a dense inverse Hessian approximation $(n \times n)$ has to be stored. When n is large, the inverse Hessian approximation is prohibitively impractical to be stored in memory, which leads to the L-BFGS algorithm.

The key idea of L-BFGS is to store only a few vectors in the history, the pair of $\{s_k, g_k\}$, used to construct the Hessian matrix. Both s_k and g_k are vectors with dimension of $n \times 1$. According to equation (1.23), the inverse Hessian approximation can be expressed as

$$\begin{aligned} B_k =& (V_{k-1}^{\mathrm{T}} \dots V_{k-m}^{\mathrm{T}}) B_k^0 (V_{k-m} \dots V_{k-1}) \\ &+ \rho_{k-m} (V_{k-1}^{\mathrm{T}} \dots V_{k-m+1}^{\mathrm{T}}) s_{k-m} s_{k-m}^{\mathrm{T}} (V_{k-m+1} \dots V_{k-1}) \\ &+ \dots \\ &+ \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}}, \end{aligned} \tag{2.24}$$

where m is the number of stored pairs $\{s_k, g_k\}$ in the history, B_k^0 is the initial guess of inverse Hessian approximation. A two-loop recursion can be derived to compute the $H_k g_k$ efficiently as is shown in Algorithm 1. B_k^0 can be obtained by solving the system of equations that

Algorithm 1 two-loop recursion Hessian approximation

$q \leftarrow g_k$
for $i = k-1, \dots, k-m$ **do**
  $\alpha_i = \rho_i s_i^{\mathrm{T}} q$
  $q = q - \alpha_i y_i$
$r \leftarrow B_k^0 q$
for $i = k-m, \dots, k-1$ **do**
  $\beta = \rho_i y_i^{\mathrm{T}} r$
  $r = r + s_i(\alpha_i - \beta)$
$B_k g_k = \mathrm{r}$

$r = B_k^0 q$. Practically, $B_k^0 = \frac{s_{k-1}^{\mathrm{T}} y_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}} I$ has been proven to be effective. Therefore, the whole optimization of L-BFGS is formulated as Algorithm 2.

Algorithm 2 L-BFGS update

Require: pre-defined m, initial guess x_0
while x has not converged **do**
  $B_k^0 \leftarrow \frac{s_{k-1}^{\mathrm{T}} y_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}} I$
  $p_k \leftarrow B_k g_k$ (where $B_k g_k$ is obtained by Algorithm 1)
  Line search for step size α_k
  $x_{k+1} \leftarrow x_k - \alpha_k p_k$
  update the vector pair $\{s_k, g_k\}$ in current iteration

Table 2.1: **Tissue Labels with Corresponding Distributions**.

Tissue	Labels	Distribution
Water	0	0.
Fat	1	U(.005, .015)
Skin	2	U(.002, .070)
Glandular	29	U(.005, .015)
Nipple	33	U(.002, .070)
Muscle	40	U(.010, .020)
Ligament	88	U(.005, .015)
TDLU	95	U(.005, .015)
Duct	125	U(.005, .015)
Artery	150	U(.360, .465)
Tumor	200	U(.030, .050)
Vein	225	U(.360, .465)

U(a,b) denotes a uniform distribution between a and b.

2.4 Image Dataset

The dataset used in this thesis is from Virtual Imaging Clinical Trial for Regulatory Evaluation(VITRE)[17]. We used VITRE to generate 973 3D numerical breast phantoms with size of 1024×1024. However, in this thesis, we only focus on certain structure of the images for example, vein, tumor, artery, duct, and etc, while discarding the majority of the background such as water. Each tissue with specific label will be assigned intensity as shown in Table 2.1. Then, we took maximum intensity projection and down-sample the image to the size of 128×128. However, the images in this dataset are inherently hard to reconstruct because of their thin tubules structure.

2.5 Numerical Examples

All the implementations are built on the top of pytorch. For the Gaussian Blurring forward operator, the convolutional function provided by pytorch is used to perform blurring. For the radon transform, a matrix form of Radon transform is generated by a Matlab code[18] and then take advantage of the sparse tensor in PyTorch to accelerate the computation. In the reconstruction phase, both Gaussian Deblurring and the inversion of Radon transform are formulated as an optimization problem and solved by L-BFGS optimizer in PyTorch with Thihonov and Total Variation regularization. A batch implementation of reconstruction in PyTorch provides a way to solve multiple inverse problems at once.

2.5.1 Gaussian Deblurring

In this section, the numerical experiments of Gaussian Deblurring problem are conducted. First, the L-curve is used to select the Tikhonov regularization parameter with a canonical Gaussian kernel with kernel size=5, standard deviation=2 and an additive Gaussian noise with zero mean and standard deviation=0.01 are used. Different scenarios including moderate versus severe blurring, high noise versus low noise are considered using both Tikhonov and Total Variation regularization with optimal parameters selected by the L-curve. The numerical experiments for Gaussian deblurring task is conducted as shown in table 2.2.

Choice of Regularization Parameter

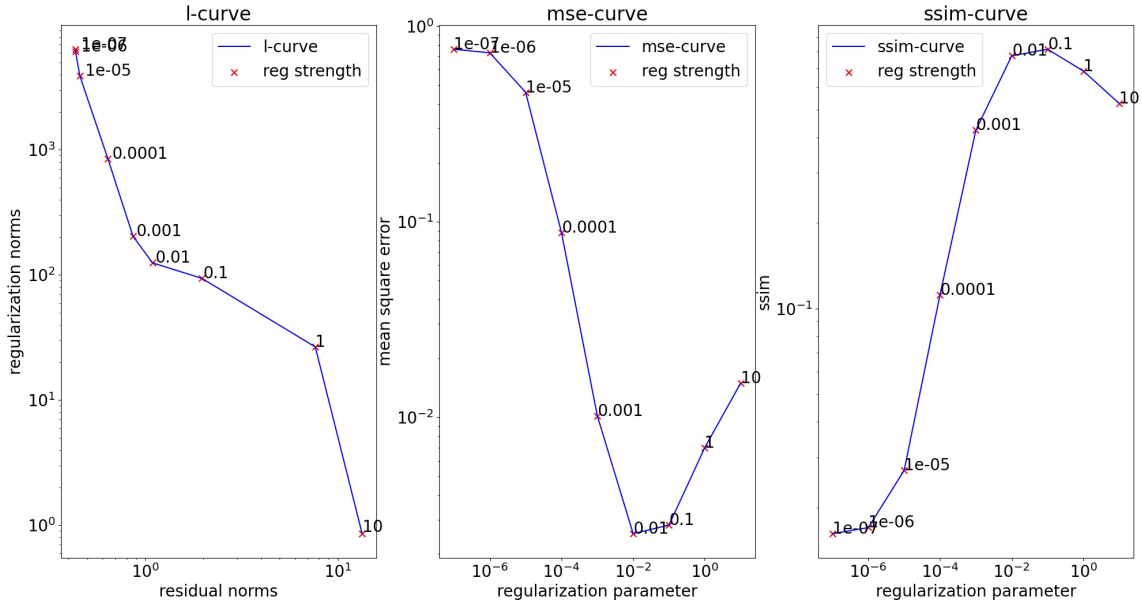

Figure 2.1: Regularization parameter selection experiment for Gaussian deblurring using Tikhonov regularization with kernel size=5, standard deviation=2, noise is an additive Gaussian noise with zero mean and standard deviation=0.01. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.01.

Table 2.2: **Numerical Experiments for Gaussian Deblurring**.

kernel-size	kernel-std	noise std	Regularization	strength	MSE(10^{-3})	SSIM	Curves	Results
5	2.0	0.01	Tikhonov	0.01	2.6, (1.9, 3.4)	0.77, (0.76, 0.79)	2.1	2.2
5	2.0	0.01	TV	0.001	2.4, (1.6, 3.7)	0.85, (0.80, 0.89)	2.1	2.2
5	2.0	0.05	Tikhonov	0.1	4.0, (2.8, 5.7)	0.65, (0.61, 0.67)	2.4	2.3
5	2.0	0.05	TV	0.01	3.5, (2.2, 5.1)	0.75, (0.69, 0.80)	2.5	2.3
5	1.0	0.01	Tikhonov	0.01	1.9, (1.5, 2.3)	0.82, (0.81, 0.83)	2.10	2.9
5	1.0	0.01	TV	0.001	1.6, (1.1, 2.3)	0.90, (0.88, 0.92)	2.11	2.9
5	1.0	0.05	Tikhonov	0.1	3.8, (3.1, 4.8)	0.65, 0.63, 0.66)	2.7	2.6
5	1.0	0.05	TV	0.01	2.9, (1.9, 4.3)	0.78, (0.74, 0.82)	2.8	2.6

The MSE and SSIM columns, formatted as [average, (minimum, maximum)], are produced by reconstructing 100 images with either Tikhonov or Total Variation regularization.

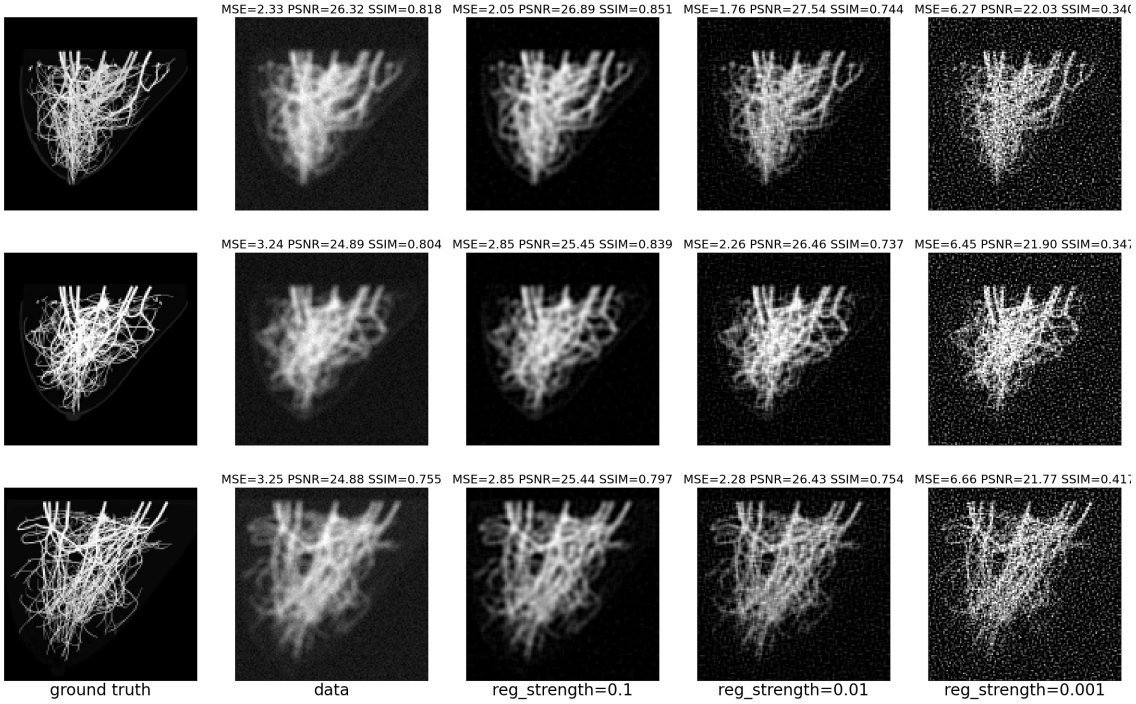

Figure 2.2: The Gaussian deblurring results with kernel size=5, standard deviation=2, and additive Gaussian noise with zero mean and standard deviation=0.01. The regularization term used is Tikhonov regularization with parameter 0.1,0.01,0.0001.

As shown in Figure 2.1, optimal regularization parameter suggested by the L-curve is 0.01. This is consistent with the parameter suggested by the mean square error curve. However, the structural similarity curve indicates that the parameter 0.1 maximizes the structural similarity between ground-truth image and the reconstructed image. The high structural similarity generally suggests a visually appealing reconstruction, which is consistent with the reconstruction result on Figure 2.2.

Moderate v.s. Severe Blurring

The moderate versus severe blurring reconstruction experiments were conducted with kernel size=5, additive Gaussian noise with zero mean and standard deviation=0.05. The standard deviation of the Gaussian kernel is 1 and 2, respectively. The optimal regularization parameter selected by the L-curve for both moderate and severe blurring are 0.1 for Tikhonov and 0.01 for Total Variation. As shown in Figure 2.4 and 2.7, for the Tikhonov regularization, the parameter selected by the L-curve may not be consistent with the SSIM-curve, which means the parameter selected by the L-curve may not be of high perceptual quality.

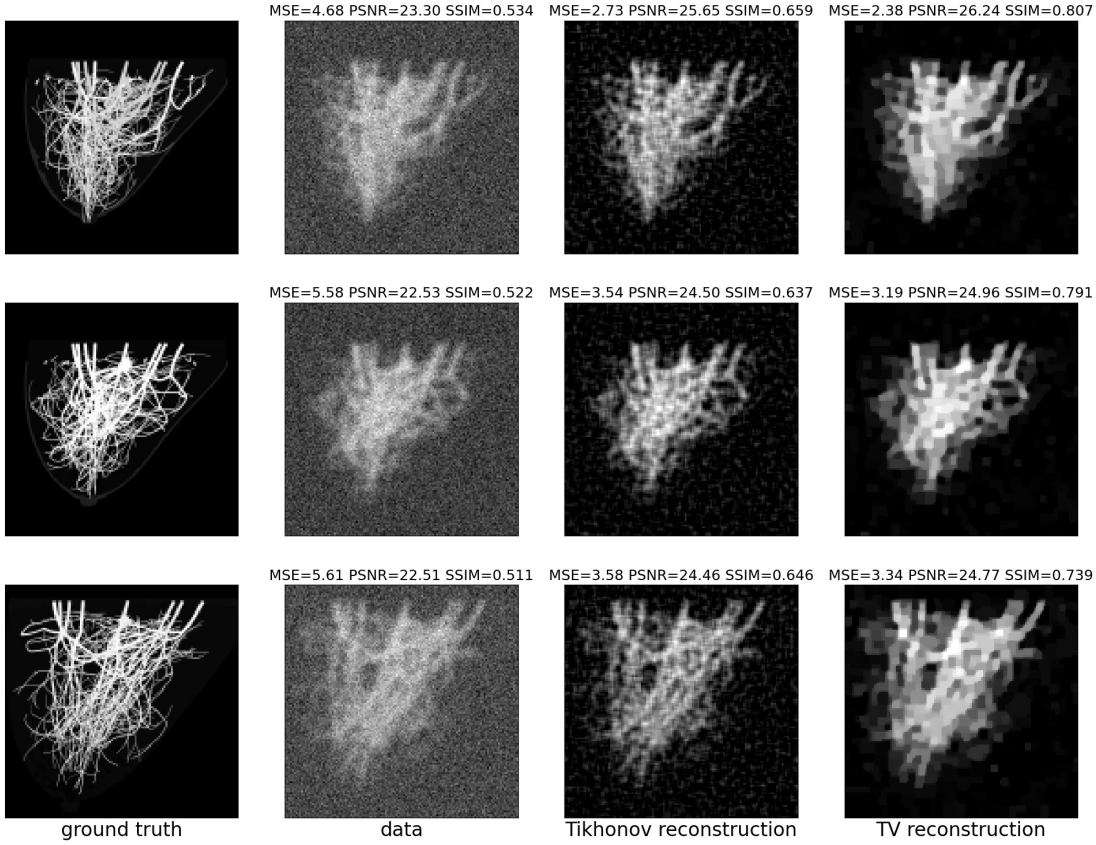

Figure 2.3: The Gaussian deblurring results with kernel size=5, standard deviation=2, noise is an additive Gaussian noise with zero mean and standard deviation=0.05.

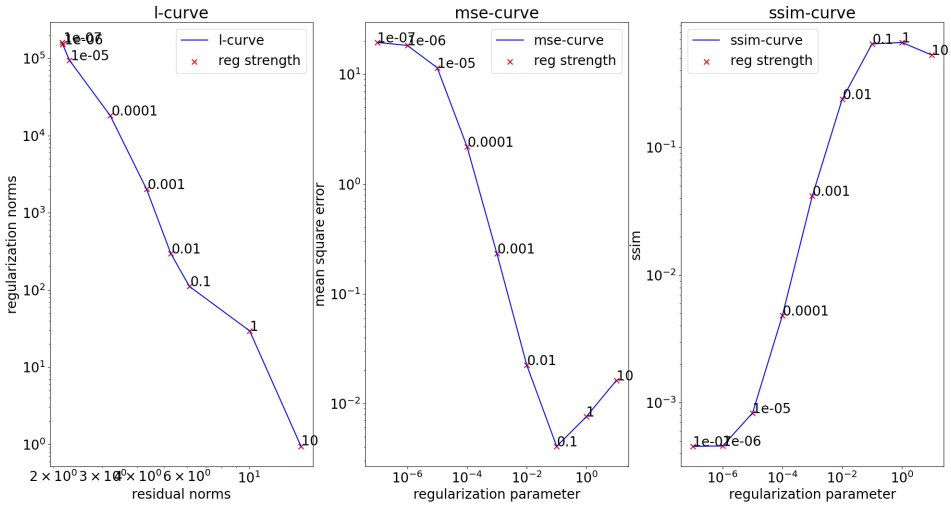

Figure 2.4: Curves for Gaussian deblurring using Tikhonov with kernel size=5, standard deviation=2, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.1.

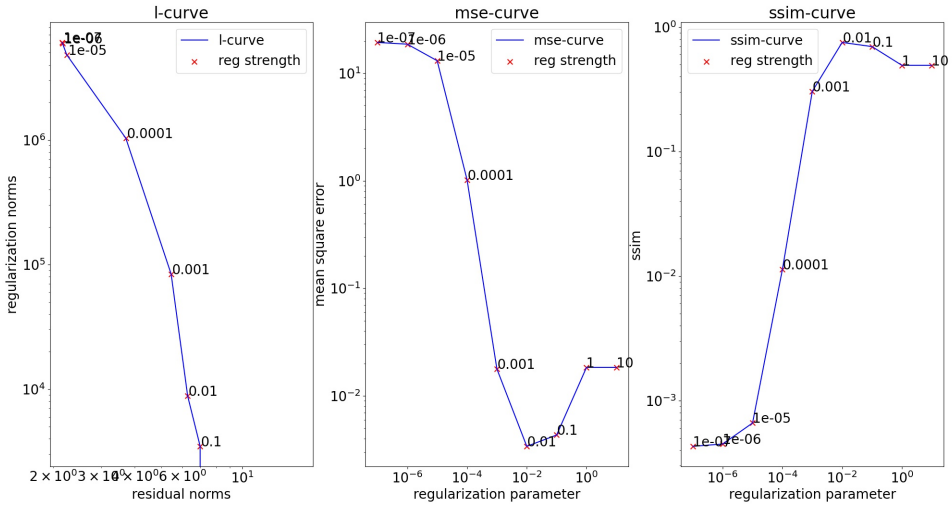

Figure 2.5: Curves for Gaussian deblurring using Total Variation with kernel size=5, standard deviation=2, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.001.

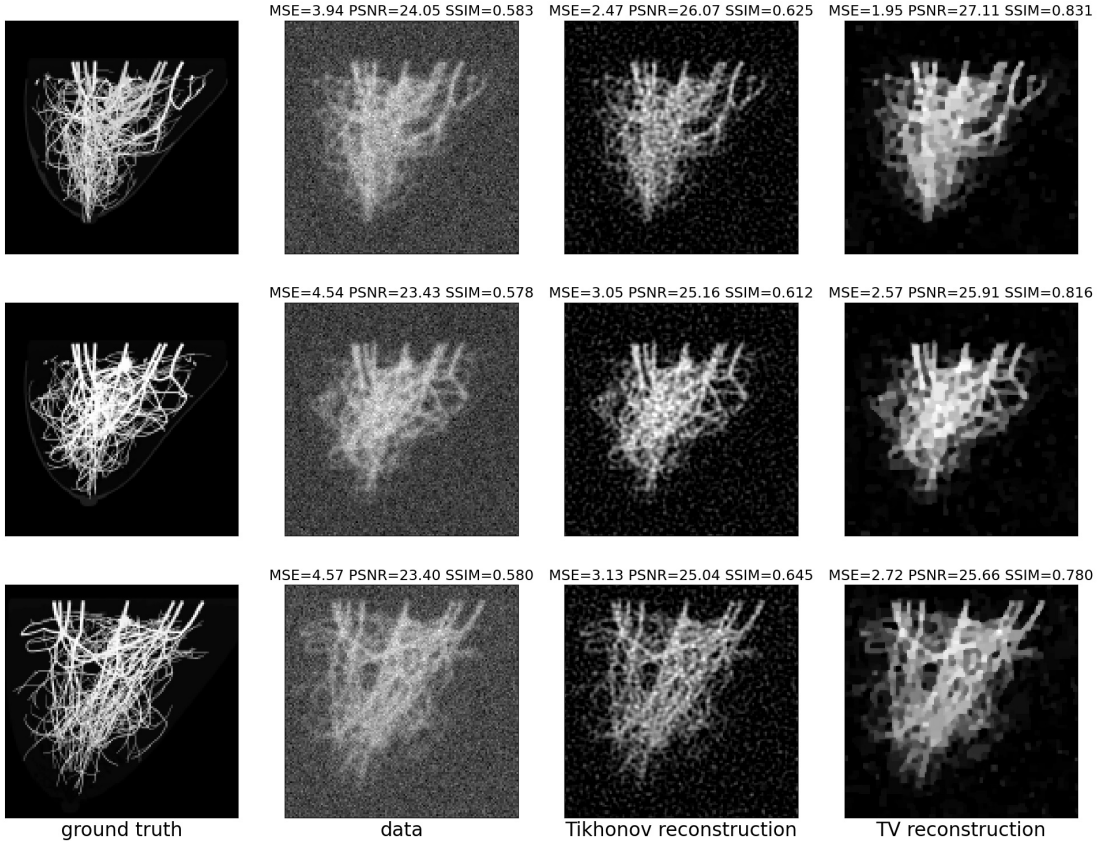

Figure 2.6: The Gaussian deblurring results with kernel size=5, standard deviation=1, noise is an additive Gaussian noise with zero mean and standard deviation=0.05.

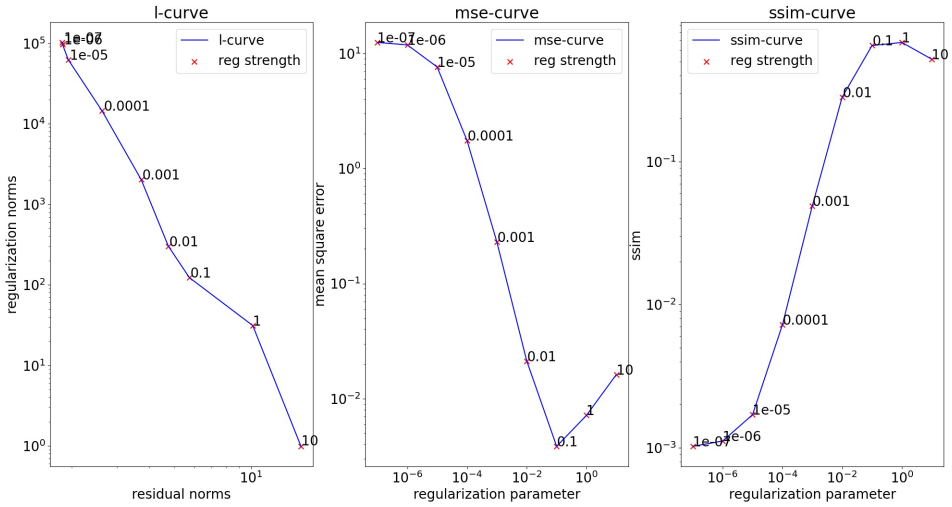

Figure 2.7: Curves for Gaussian deblurring using Tikhonov with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.1.

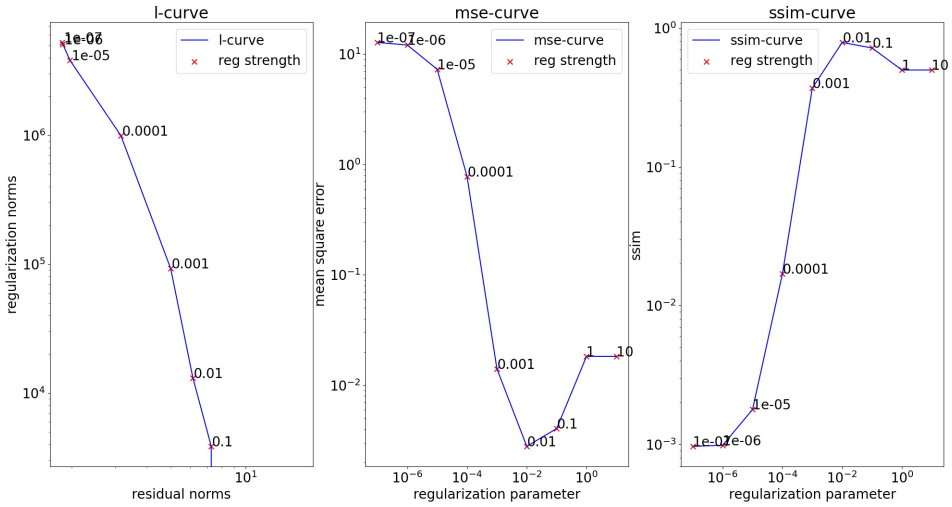

Figure 2.8: Curves for Gaussian deblurring using Total Variation with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.01.

High v.s. Low Noise

The high noise versus low noise Gaussian deblurring were conducted with kernel size=5, standard deviation=1. The additive noise is Gaussain noise with zero mean and standard deviation=0.01 and 0.05, respectively. The optimal regularization parameter selected by the L-curve of low noise are 0.01 for Tikhonov and 0.001 for Total Variation. The optimal regularization parameter selected by the L-curve of high noise are 0.1 for Tikhonov and 0.01 for Total Variation. As shown in Figure 2.10 and 2.13, for the Tikhonov regularization, the parameter selected by the L-curve does not neccessarily look visually appealing.

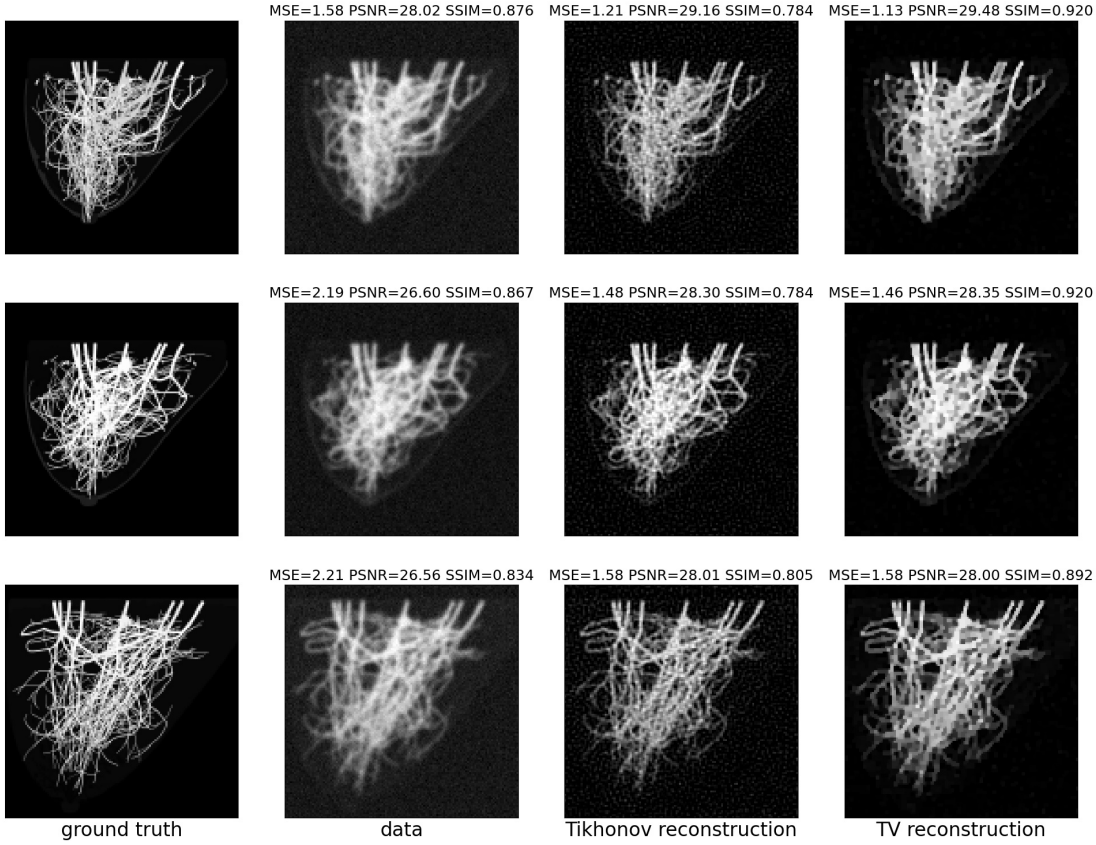

Figure 2.9: The Gaussian deblurring results with kernel size=5, standard deviation=1, noise is an additive Gaussian noise with zero mean and standard deviation=0.01.

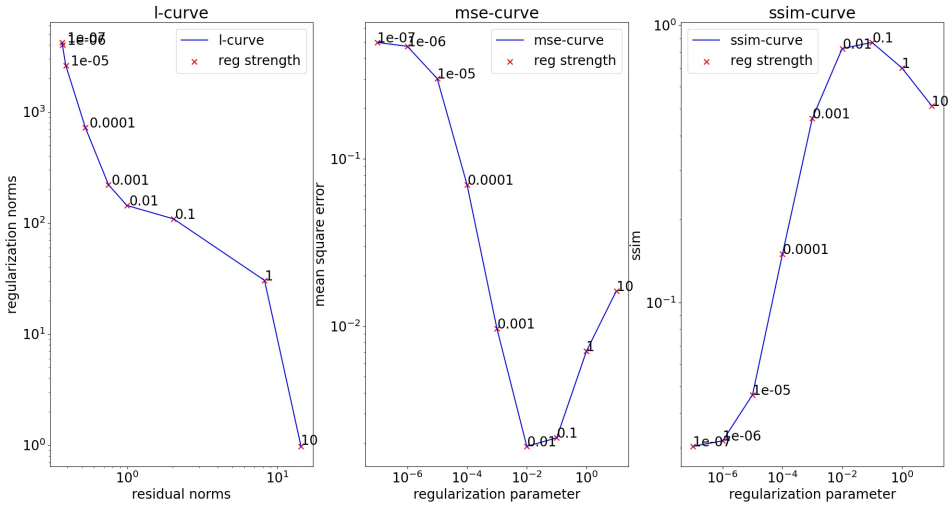

Figure 2.10: Curves for Gaussian deblurring using Tikhonov with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.01. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.01.

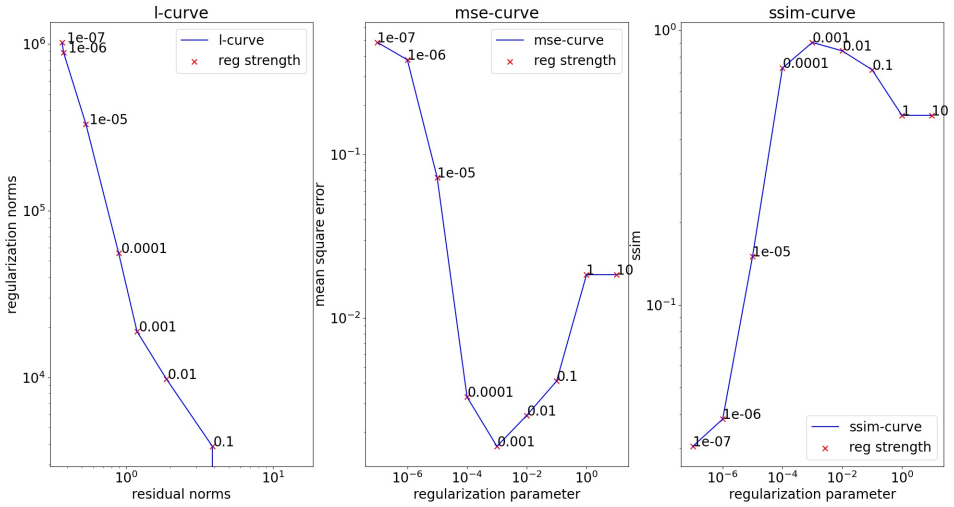

Figure 2.11: Curves for Gaussian dDeblurring using Total Variation with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.01. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10].The optimal regularization strength in this setting is 0.001.

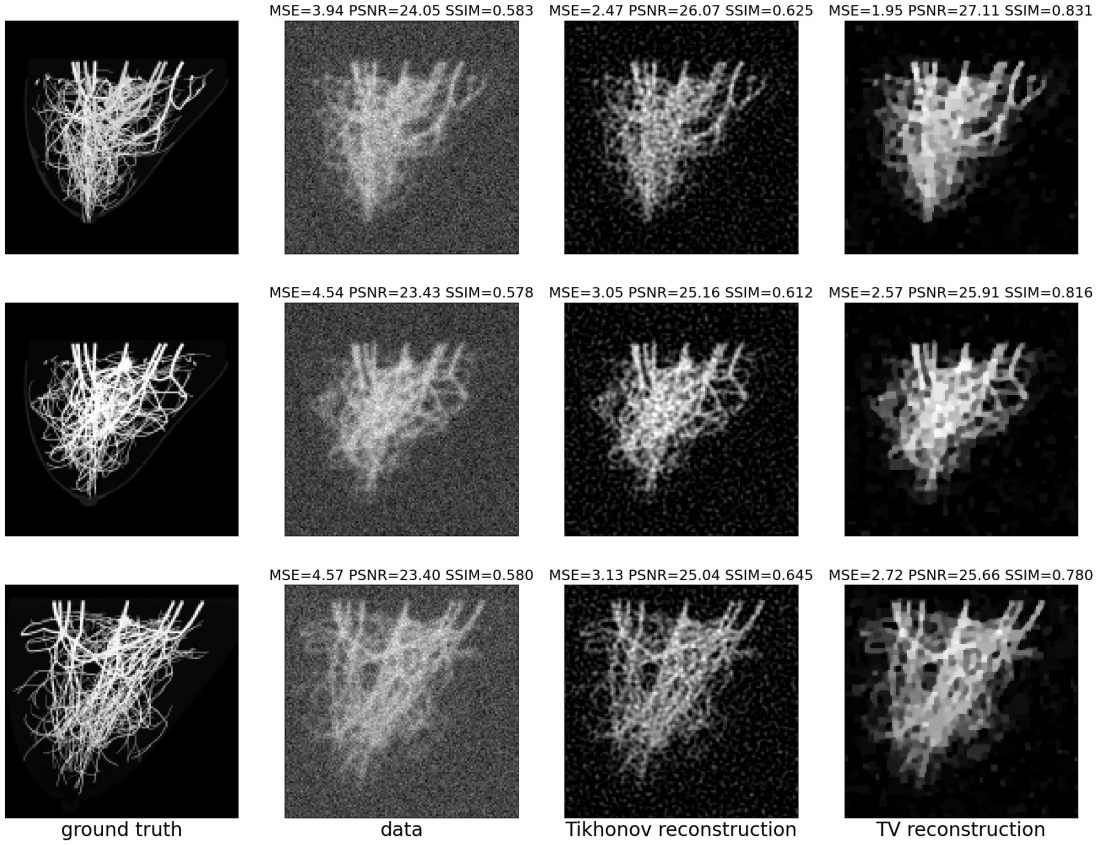

Figure 2.12: The Gaussian deblurring results with kernel size=5, standard deviation=1, noise is an additive Gaussian noise with zero mean and standard deviation=0.05.

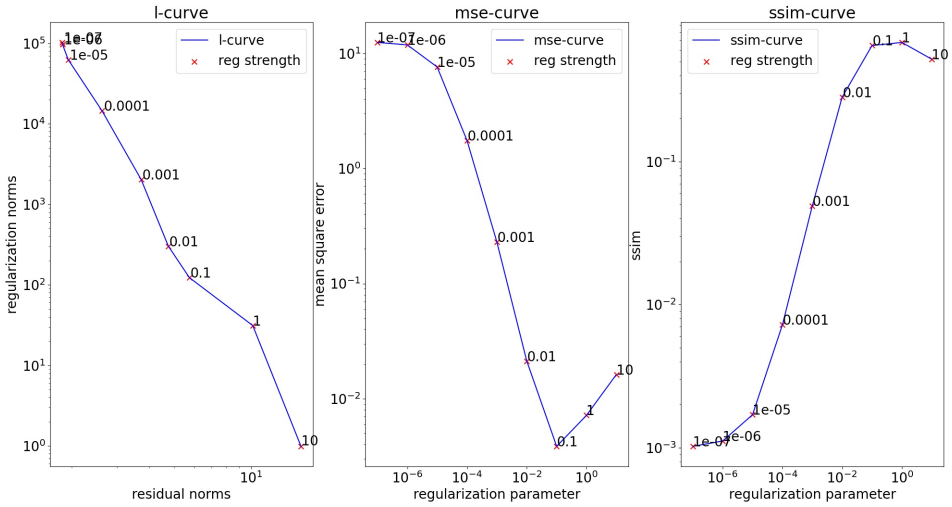

Figure 2.13: Curves for Gaussian deblurring using Tikhonov with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10].

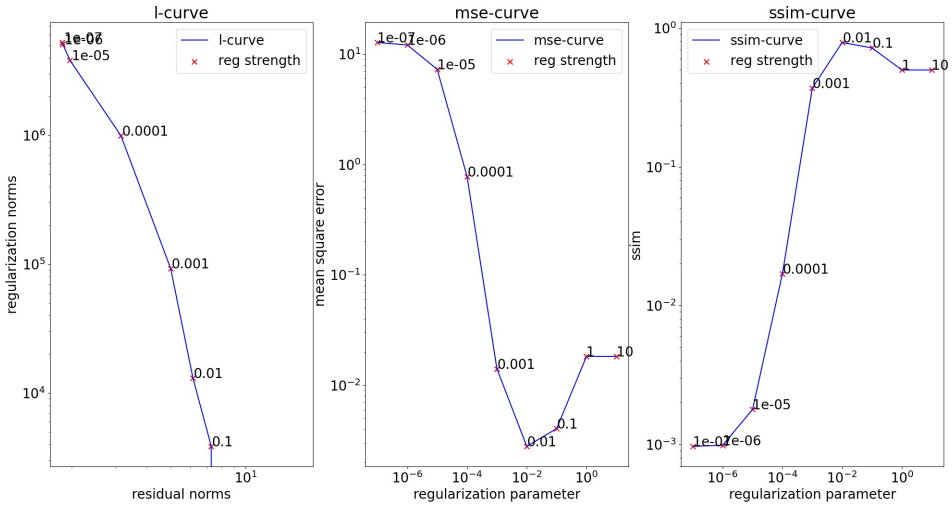

Figure 2.14: Curves for Gaussian deblurring using Total Variation with kernel size=5, standard deviation=1, additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10].

2.5.2 Radon Transform

In this section, the numerical experiments with Radon transform are conducted. Different scenarios including sparse versus dense view, high noise versus low noise are considered using both Tikhonov and Total Variation regularization with optimal parameters selected by the L-curve. The numerical experiments for Radon transform are conducted as shown in Table 2.3.

Table 2.3: **Numerical Experiments for Radon Transform**.

num-rayse	num-views	noise std	Regularization	Strength	MSE(10^{-3})	SSIM	Curves	Results
181	90	0.1	Tikhonov	1.0	0.7, (0.4, 0.9)	0.93, (0.92, 0.95)	2.16	2.15
181	90	0.1	TV	0.1	0.2, (0.1, 0.4)	0.99, (0.98, 0.99)	2.17	2.15
181	90	0.01	Tikhonov	0.1	1.2, (0.8, 1.5)	0.87, (0.86, 0.88)	2.22	2.21
181	90	0.01	TV	0.01	0.5, (0.2, 0.7)	0.97, (0.96, 0.98)	2.23	2.21
181	45	0.1	Tikhonov	1	2.3, (1.5, 3.4)	0.79, (0.76, 0.83)	2.19	2.18
181	45	0.1	TV	0.1	1.6, (0.8, 2.6)	0.90, (0.85, 0.93)	2.20	2.18

The MSE and SSIM columns, formatted as [average, (minimum, maximum)], are obtained by reconstructing 100 images using either Tikhonov or Total Variation regularization.

Sparse view v.s. Dense view

The sparse versus dense views experiments were conducted with a Radon transform with 181 rays for each view, an additive Gaussian noise with zero mean and standard deviation=0.1. The number of views are 90 and 45 for dense and sparse view, respectively. The selected parameter of L-curve for both sparse and dense view are 1.0 for Tikhonov and 0.1 for Total Variation. As indicated by Figure 2.16 and 2.19, the selected parameter by L-curve does not neccessarily the best choice of MSE-curve and SSIM-curve.

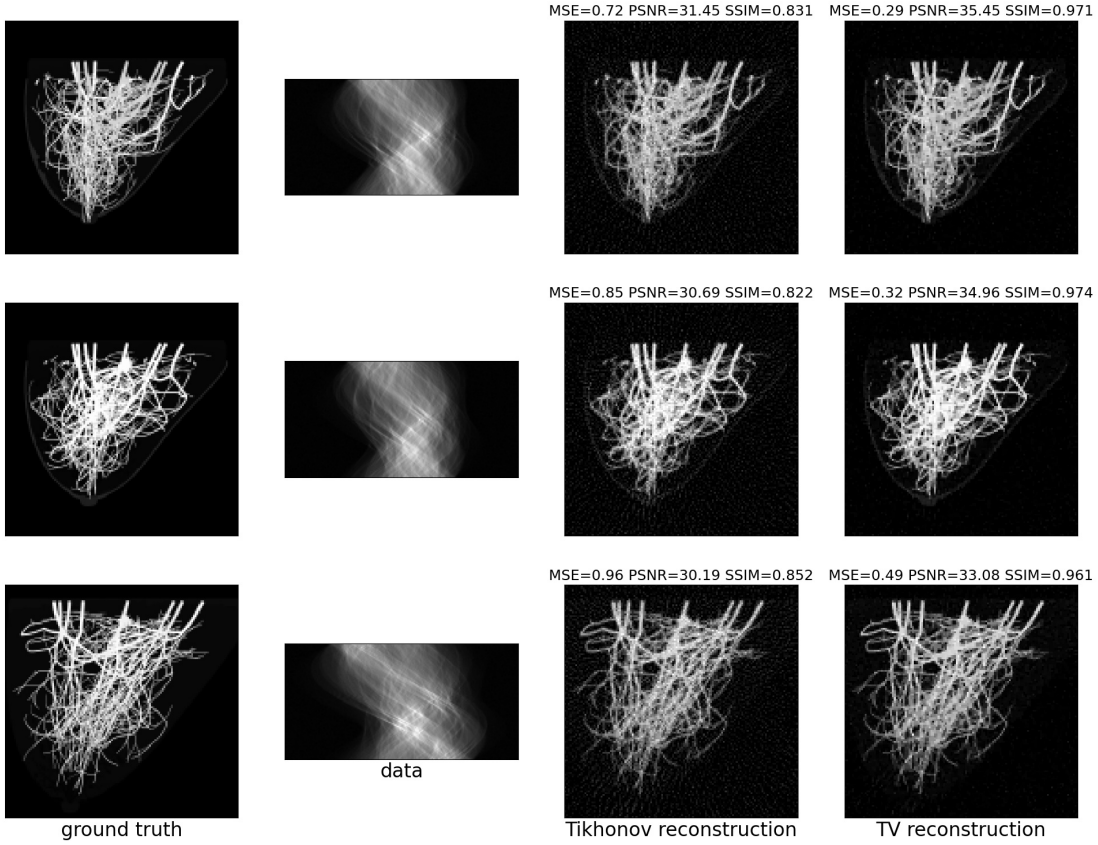

Figure 2.15: The Radon transform reconstruction results with number of rays=181, number of views=90, noise is an additive Gaussian noise with zero mean and standard deviation=0.01.

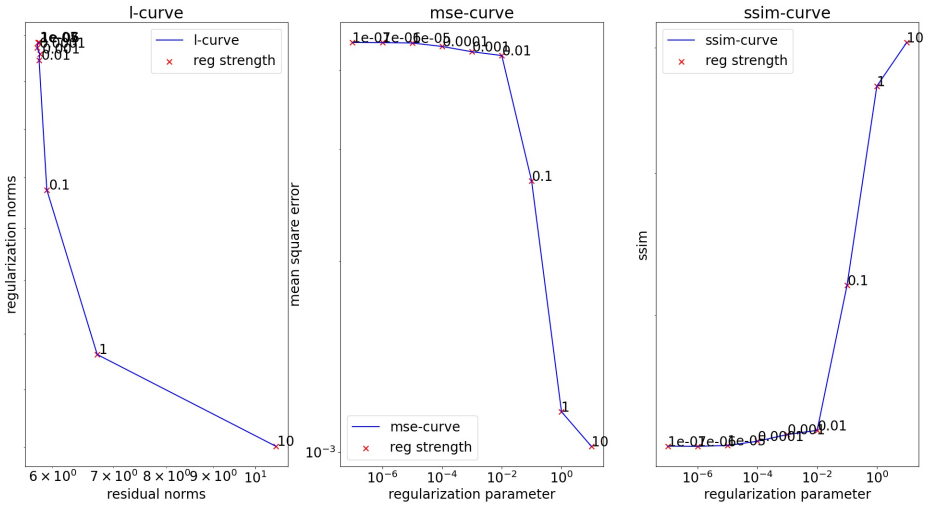

Figure 2.16: Curves for Radon transform using Tikhonov regularization with number of views=90, additive Gaussian noise with zero mean and standard deviation=0.01. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 1.0.

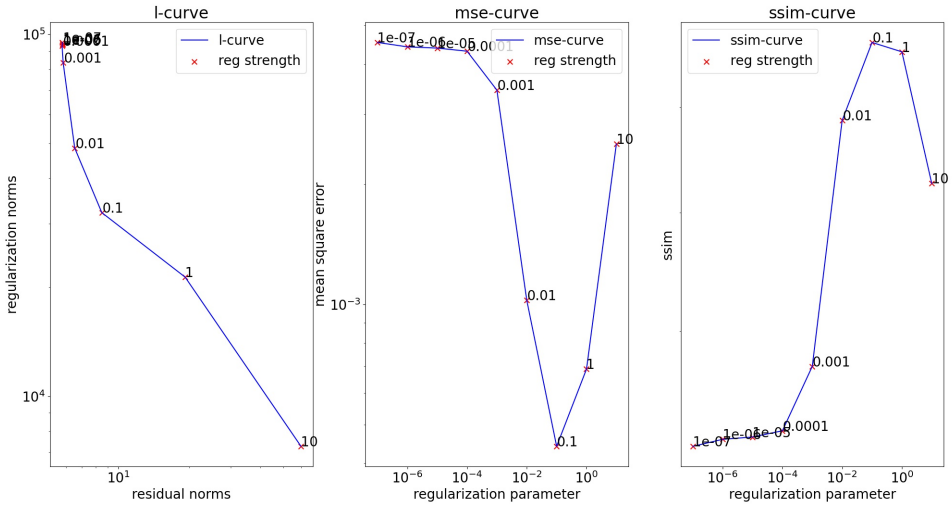

Figure 2.17: Curves for Radon transform using Total Variation regularization with number of views=90, additive Gaussian noise with zero mean and standard deviation=0.01. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10].The optimal regularization strength in this setting is 0.1.

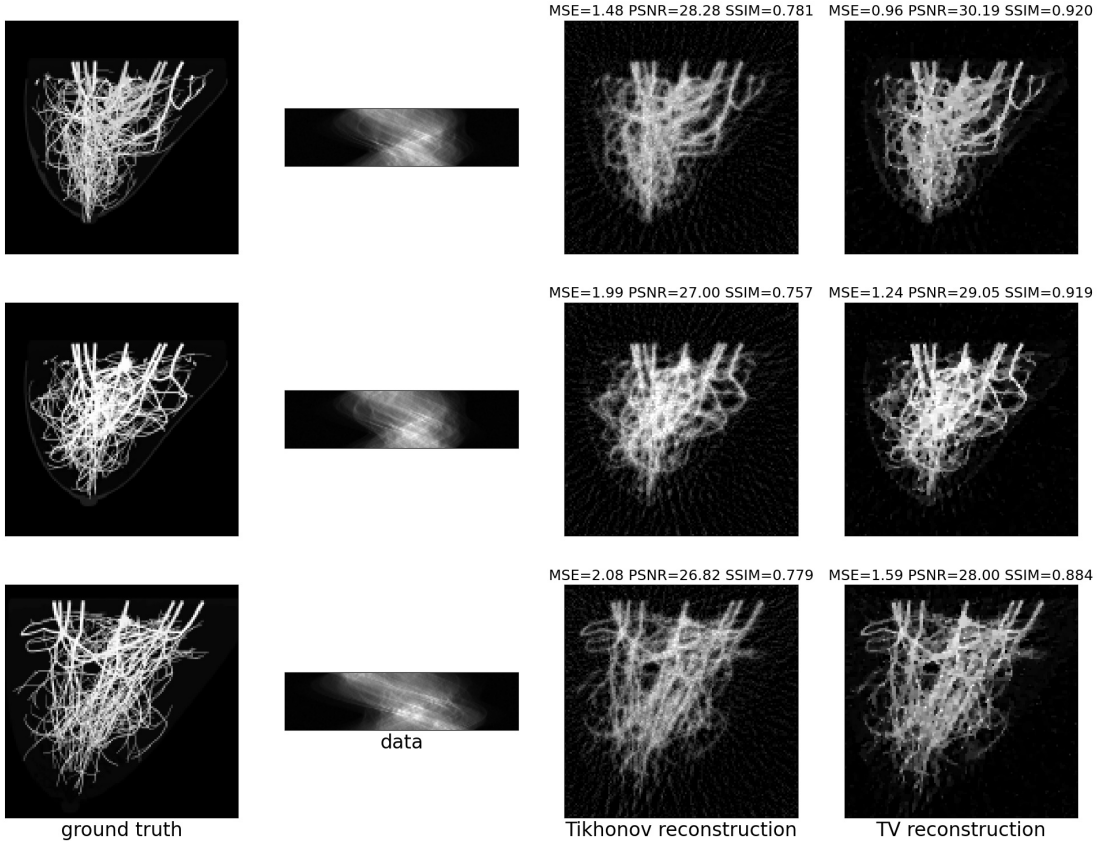

Figure 2.18: The Radon transform reconstruction results with number of rays=181, number of views=45, noise is an additive Gaussian noise with zero mean and standard deviation=0.1.

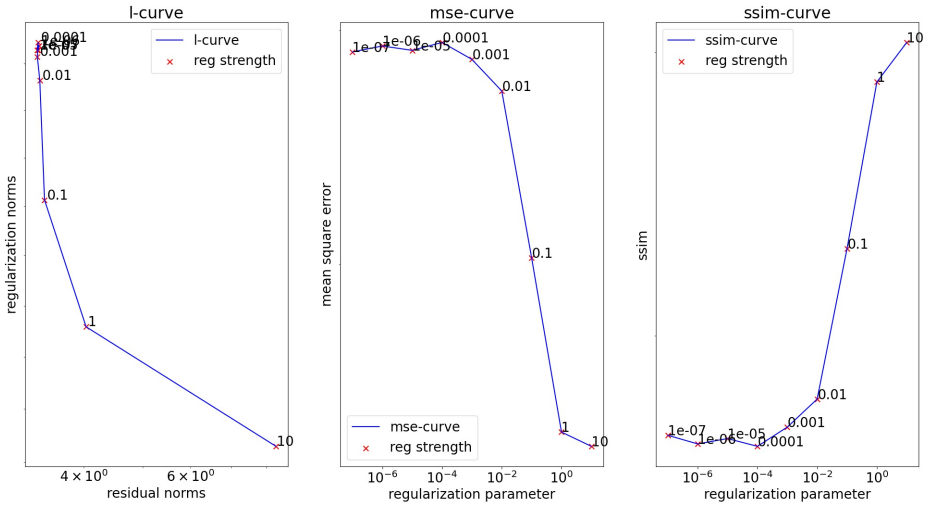

Figure 2.19: Curves for Radon transform using Tikhonov regularization with number of views=45, additive Gaussian noise with zero mean and standard deviation=0.1. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 1.0.

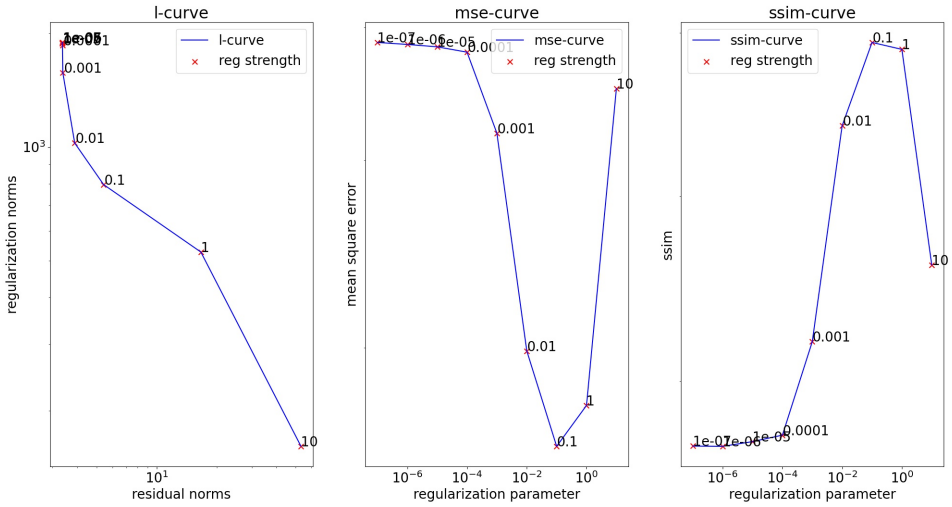

Figure 2.20: Curves for Radon transform using Total Variation regularization with number of views=45, additive Gaussian noise with zero mean and standard deviation=0.1. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.1.

High v.s. Low noise

The high noise versus low noise experiments were conducted with a Radon transform with 90 views and 181 rays for each view. The noise is an additive Gaussian noise with zero mean and standard deviation=0.01 and 0.1, respectively. The selected parameter of L-curve for both sparse and dense view are 0.1 for Tikhonov and 0.01 for Total Variation. As indicated by Figure 2.16 and 2.22, the selected parameter by L-curve does not neccessarily agrees with MSE-curve and SSIM-curve.

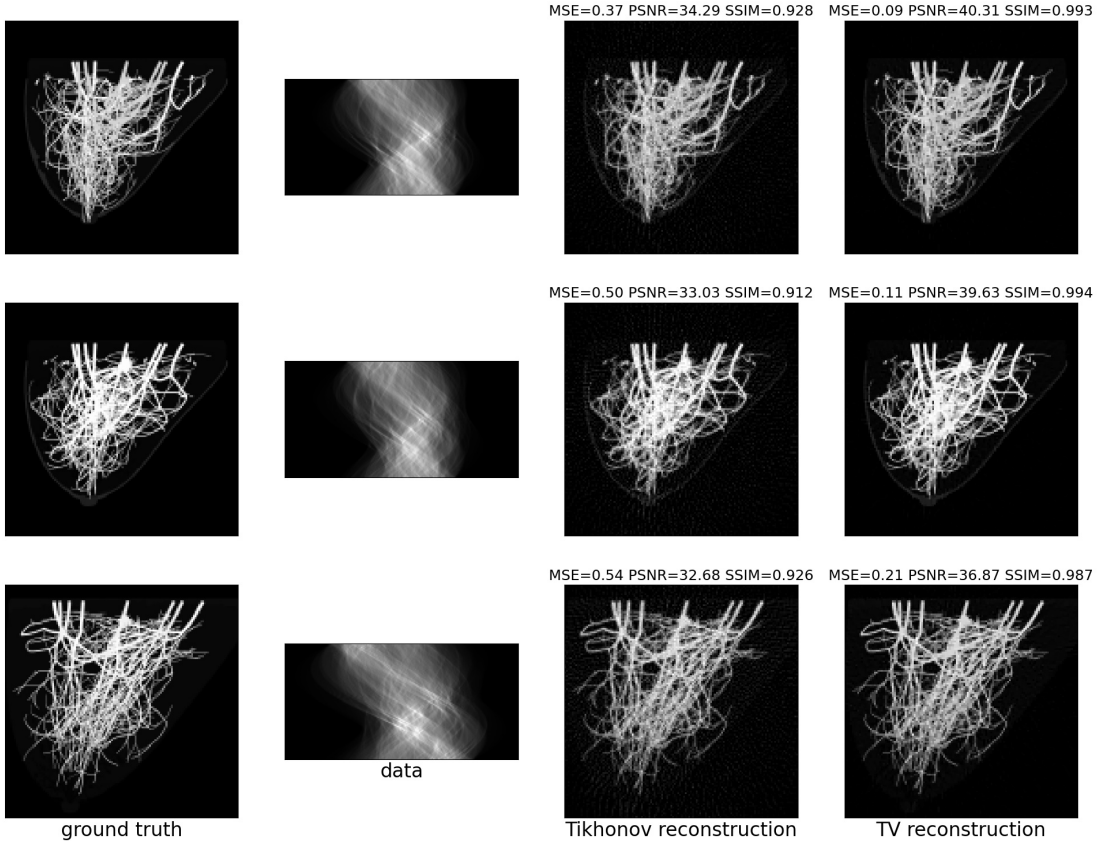

Figure 2.21: The Radon transform reconstruction results with number of rays=181, number of views=90, noise is an additive Gaussian noise with zero mean and standard deviation=0.01.

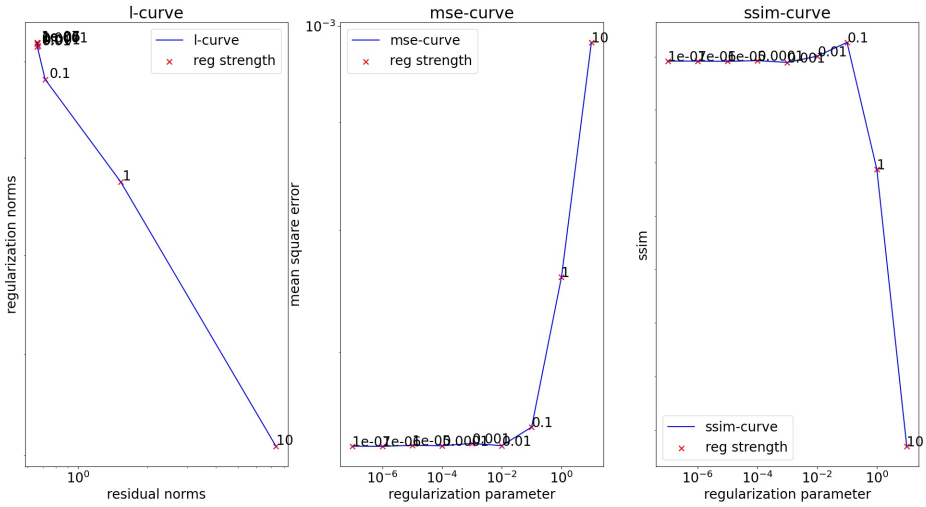

Figure 2.22: Curves for Radon transform using Tikhonov regularization with number of views=90, additive Gaussian noise with zero mean and standard deviation=0.1. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.1.

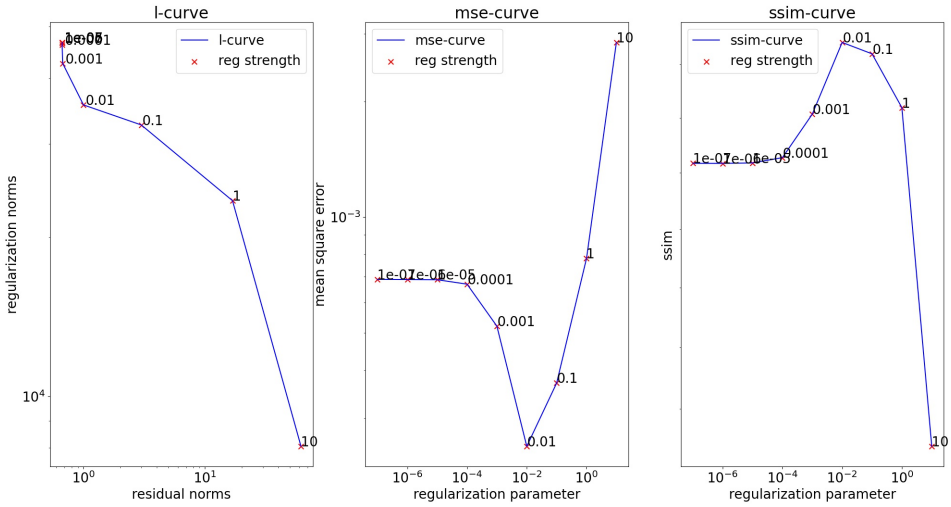

Figure 2.23: Curves for Radon using Total Variation regularization with number of views=90, additive Gaussian noise with zero mean and standard deviation=0.1. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 10]. The optimal regularization strength in this setting is 0.01.

2.6 Remarks

The forward model is of vital importance to project the sought-after image onto a measurable quantity. The reconstruction using classical regularization methods with the optimally selected regularization strength is able to reconstruct the ground-truth image with a reasonably good quality. Especially, for the Radon transform with dense views, the reconstruction using Total Variation regularization is able to approximate the ground-truth image very well. Furthermore, the batch implementation of reconstruction in PyTorch provides us a feasible way to online generate the training dataset for training adversarial regularizer in Chapter 3.

Chapter 3

Adversarial Regularizer

In this chapter, the proposed adversarial regularizer will be introduced along with its training strategy. We first present how the learned regularizer can be used to solve the image reconstruction problem in Chapter 2. Network architecture can be formulated as a critic in the adversarial generative network. Secondly, we will explain how the proposed adversarial regularizer will be implemented. We will discuss the loss function used to train the proposed networks. Finally, the proposed regularizer will be applied to solve the inverse problem discussed in Ch 2.

3.1 Regularizer as a Critic

Conventionally, a direct mapping from the measurement d to the ground-truth image m, or a residual mapping from the residual term $F(m) - d$ to the noise n is learned in a supervised setting. However, there are two main limitations when it comes to solve the inverse problem in medical imaging reconstruction. The first is that the limited number of training dataset may not achieve a good performance in a supervised learning setting. The second is that the inversion of a complex forward operator in an inverse problem is always hard to learn.

One possible solution to those two obstacles is the combination of the physical model and the neural network. As demonstrated in Chapter 2, the image reconstruction problem can be formulated as an optimization problem with a variational regularization. In this chapter, a parametric form of a regularization functional $R_\theta(m)$ is learned from the training

data. Then, we plug in the learned regularizer in equation (2.16) and solve the minimization problem iteratively to approach the ground-truth image.

Ideally, a good regularization functional $R_\theta(m)$ is able to differentiate the ground-truth image m from its degraded or partially reconstructed version m_d. For a ground-truth image m, the value of a good regularization functional $R_\theta(m)$ should be small; while for a degraded image, the value of $R_\theta(m)$ should be large. In a statistical view, the ground-truth image m comes from a distribution $m_r \in \mathbb{P}_{m_r}$, and the degraded image also comes from a distribution $m_d \in \mathbb{P}_{m_d}$. This leads to the discriminator or critic in a adversarial generative network(GAN), which is capable of differentiating the ground-truth image from any other images. Mathematically, the quantity

$$\mathbb{E}_{X\in\mathbb{P}_{m_r}} R_\theta(X) - \mathbb{E}_{X\in\mathbb{P}_{m_d}} R_\theta(X) \tag{3.1}$$

should be as small as possible for a reasonably good regularization functional R_θ. A good regularization functional can then be found by solving

$$R_\theta^* = \arg\min \ \mathbb{E}_{X\in\mathbb{P}_{m_r}} R_\theta(X) - \mathbb{E}_{X\in\mathbb{P}_{m_d}} R_\theta(X), \tag{3.2}$$

where θ is a set of parameters of regularization functional. Furthermore, the critic is trained in an unsupervised way, which mitigates the limited size of training images in our case. We will further describe the network architecture, loss function, and training strategy of training a critic as regularizer in the following sections.

3.2 Structure of the Network

The proposed neural network is implemented as a convolutional neural network, because the convolutional neural network can extract the hierarchical features of an image[19]. Considering training a deep neural network is computationally expensive, in order to validate the propose method, a shallow network is developed at the first phase. The simple architecture aims to mimic the functionality of a total variation regularization. While the complex architecture works like a discriminator in the Deep convolutional GANs.

Simple Architecture

Table 3.1: **Simple Network Architecture**.

Name	**Layer Type**	**Filter Size** ($K_1 \times K_2 \times C_{in} \times C_{out}$)	**Feature Map Size** ($H_{out} \times W_{out} \times C_{out}$)
	Input		$128 \times 128 \times 1$
Conv1	convolution	$3 \times 3 \times 1 \times 4$	$126 \times 126 \times 4$
Conv2	convolution	$1 \times 1 \times 4 \times 8$	$126 \times 126 \times 8$
avgpool1	avgpooling		1

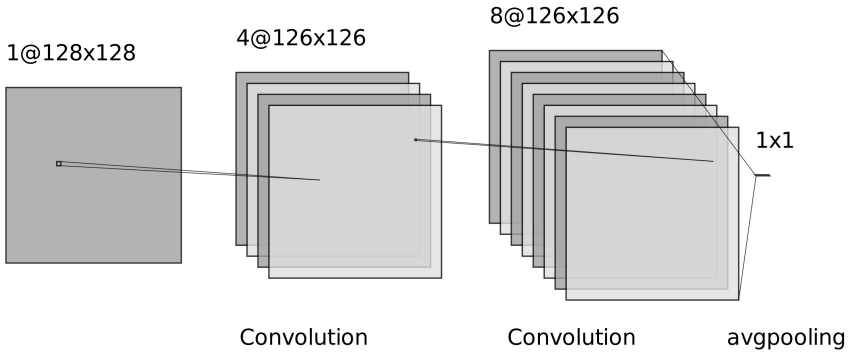

Figure 3.1: The simple neural network architecture.

Complex Architecture

The complex architecture makes use of the architecture used in the deep convolutional GANs.

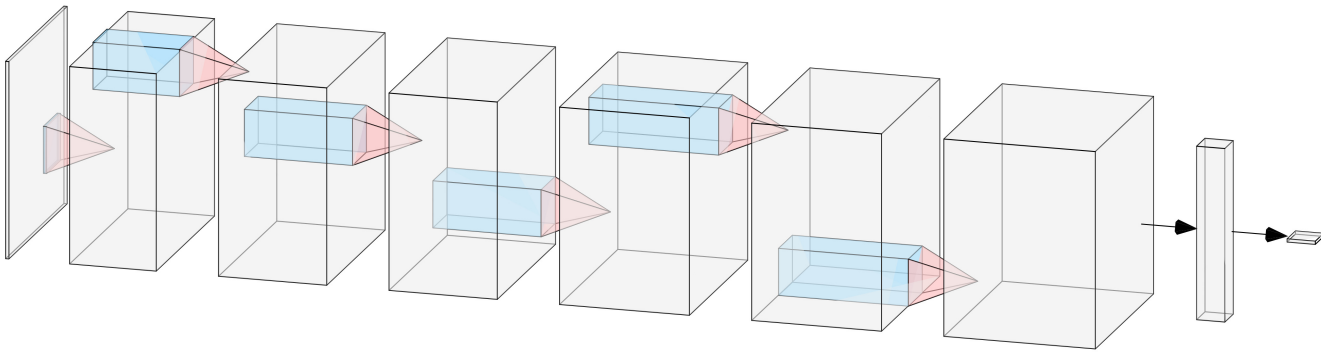

Figure 3.2: The complex neural network architecture.

Table 3.2: **Complex Network Architecture**.

Name	**Layer Type**	**Filter Size** ($K_1 \times K_2 \times C_{in} \times C_{out}$)	**Feature Map Size** ($H_{out} \times W_{out} \times C_{out}$)
	Input		$128 \times 128 \times 1$
Conv1	convolution	$5 \times 5 \times 1 \times 16$	$128 \times 128 \times 4$
Conv2	convolution	$5 \times 5 \times 16 \times 32$	$128 \times 128 \times 32 \times 8$
Conv3	convolution	$5 \times 5 \times 32 \times 32$	$64 \times 64 \times 32$
Conv4	convolution	$5 \times 5 \times 32 \times 64$	$32 \times 32 \times 64$
Conv5	convolution	$5 \times 5 \times 64 \times 64$	$16 \times 16 \times 64$
Conv6	convolution	$5 \times 5 \times 64 \times 128$	$8 \times 8 \times 128$
FC1	fully-connected	$1 \times 1 \times 8192 \times 256$	$1 \times 1 \times 256$
FC2	fully-connected	$1 \times 1 \times 256 \times 1$	$1 \times 1 \times 1$

3.3 Wasserstein Loss Function

According to our aim indicated in equation (1.2), to train our network is equivalent to maximize the distance between two distributions. While measuring the distance between two distributions remain intractable, Martin et al[11]., proposed a good distance called Wasserstein distance to measure how close two distributions are when training GANs. Based on the Kantorovich-Rubinstein duality[20], the Wasserstein distance is then formally defined as

$$W(\mathbb{P}_{m_r}, \mathbb{P}_{m_d}) = \sup_{f \in 1-Lip} \mathbb{E}_{X \in \mathbb{P}_{m_d}} f(X) - \mathbb{E}_{X \in \mathbb{P}_{m_r}} f(X), \tag{3.3}$$

where f is a functions with Lipschitz constant equal to 1. The negative Wasserstein distance is then the loss function to minimize in equation(3.2), so-called the wasserstein loss function.

Several methods exist to impose the 1-Lipschitz constrain including weight clipping[11], gradient penalty[12], spectral normalization[21]. The weight clipping, which is a heuristic method to clip the weight of the entire neural network, has no theoretical guarantees. In this work, the gradient penalty will be added to the loss function to constrain the Lipschitz constant in the form of

$$\mathcal{L}(\theta) = \mathbb{E}_{X \in \mathbb{P}_{m_d}} f(X) - \mathbb{E}_{X \in \mathbb{P}_{m_r}} f(X) + \chi(R_\theta), \tag{3.4}$$

where $\mathcal{L}(\theta)$ is the loss function, $\chi(R_\theta)$ is the indicator of the set $1 - Lip$ of all Lipschitz continuous function with constant 1. That is

$$\chi(f) = \lambda(|\nabla f|^2 - 1)_+^2 \tag{3.5}$$

3.4 Training Strategy

The training strategy to the proposed network is quite different from that of the majority of the neural networks. The first concern is that the original dataset only contains ground-truth images. Therefore, the very first step is to generate the measurements using a forward operator and add noise to them. Subsequently, the partially reconstructed images

are generated online using the optimization strategy proposed in Chapter 2, which means the partially reconstructed training images are generated at each training epoch. There are two main reasons by doing so. On one hand, a richer training set will be obtained, even if the ground-truth images m are repeated each epoch, the noise n will not remain the same. This is a useful data augmentation when it comes to the limited training dataset. Furthermore, generating the training set online can save the memory, only one batch of the image reconstruction is computed. Even though the computation needed for per epoch is slightly increased because of the forward and reconstruction process, the batch implementation introduced in Chapter 2 in conjunction with GPU acceleration partially mitigates this problem.

The whole algorithm of learning an adversarial regularizer R_θ is shown in Algorithm 3. The regularization term $R(x)$ used to compute the partially reconstructed images is either Tikhonov regularization or Total Variation regularization in this thesis. The forward operator A used in this thesis is linear forward operators discussed in Chapter 2, which is either Gaussian blurring or Radon transform.

Algorithm 3 Learning an Adversarial Regularizer

Require: the ground-truth images m, forward operator A, noise standard deviation σ_n, regularization strength λ, the number of batches b, Adam hyperparameters α

while θ has not converged **do**

for $i = 1, \dots, b$ **do**

Sample ground truth image $m_i \sim \mathbb{P}_{m_r}$

$d_i = Am_i + \mathcal{N}(0, \sigma_n^2 I)$

$m_i^{recon} \leftarrow \arg\min_{m_i} \frac{1}{2} \parallel Am_i - d_i \parallel_2^2 + \lambda R(m_i)$

$L_i \leftarrow R_\theta(m_i) - R_\theta(m_i^{recon}),\ R_\theta \in 1 - Lip$

$\theta \leftarrow Adam(\nabla_\theta L_i, \alpha)$

3.5 Adversarial Regularizer for Inverse Problems

Once the adversarial regularizer $R_\theta(x)$ is learned from the network, it can be plugged in the equation (2.16) to serve as the regularization term, hence, solving the inverse problem as

$$m^* = \arg\min_m \frac{1}{2} \parallel F(m) - d \parallel_2^2 + \alpha R_\theta(m) + \beta R(m), \tag{3.6}$$

Table 3.3: **Numerical Experiments for Gaussian Deblurring**.

model	training set	critic curves	Reconstruction	Regularization	strength	mse avg(10^{-3})	ssim avg
Naive	Tikhonov	3.3, 3.4	3.5, 3.6	Tikhonov	0.1	4.17	0.641
				TV	0.01	4.02	0.727
				Adv	(0.1,1000)	4.35	0.711
Deep	Tikhonov	3.7, 3.8	3.9, 3.10	Tikhonov	0.1	4.17	0.641
				TV	0.01	4.02	0.727
				Adv	(0.1,0.1)	4.2	0.641

The MSE and SSIM are obtained by reconstructing 100 images using adversarial regularizer. The results are produced by Gaussian deblurring problem with kernel size=5, standard deviation=2.0, noise deviation=0.05. The regularization α and β of using adversarial regularizer is formatted as (α, β).

where α is the regularization strength for adversarial regularizer, $R(x)$ is the regularization term other than the adversarial regularizer, β is the regularization strength for $R(x)$.

To solve this optimization problem, we can still use the same iterative methods proposed in Chapter 2.

3.6 Numerical Results

3.6.1 Gaussian Deblurring

For Gaussian deblurring problem, the training dataset is reconstructed from Gaussian blurring with kernel size=5, standard deviation=2.0, noise deviation=0.05 using Tikhonov regularization regularization or Total Variation regularization.

Shallow Model

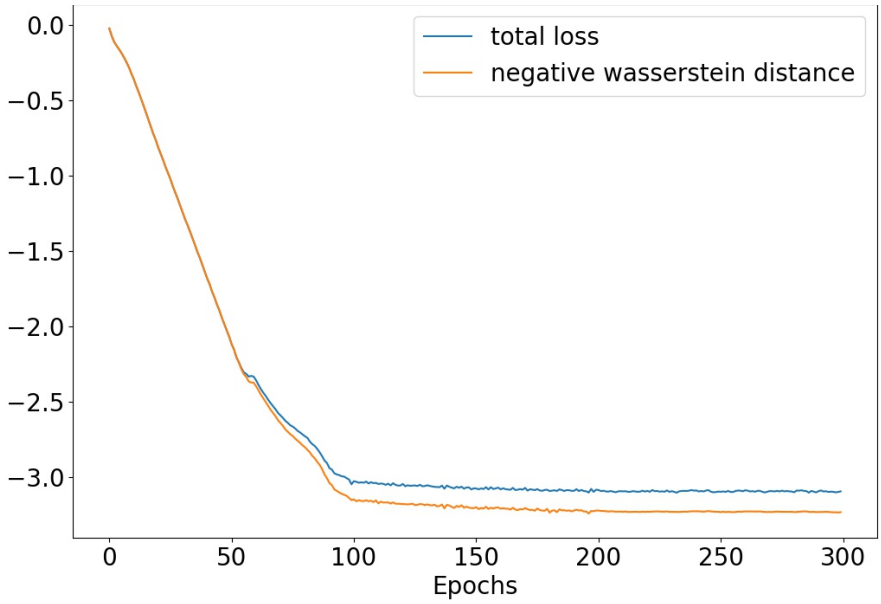

Figure 3.3: The total training losses and Wasserstein distance between true images and their classical reconstruction with Tikhonov regularization over 300 training epochs.

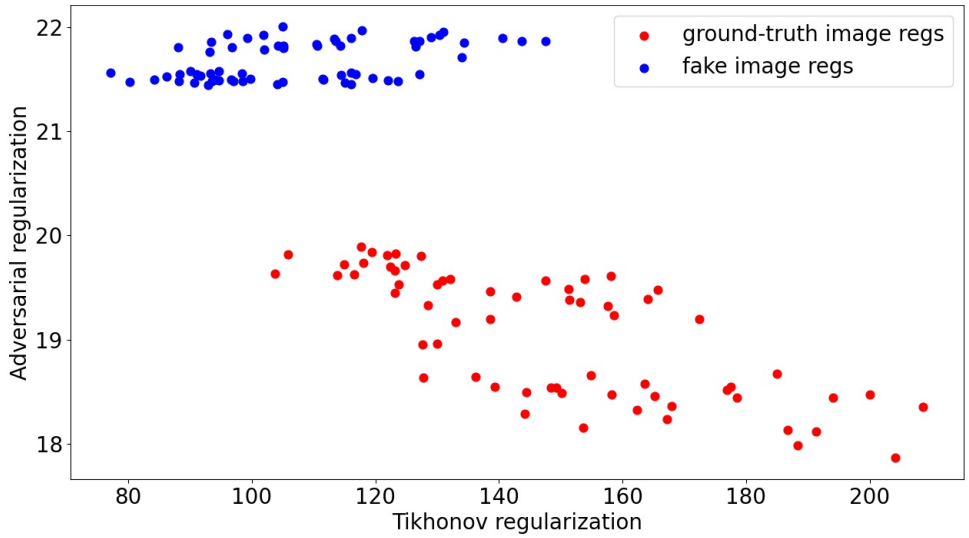

Figure 3.4: The Tikhonov regularization against the learned adversarial regularization over 100 ground-truth images and their corresponding reconstructions using Tikhonov regularization.

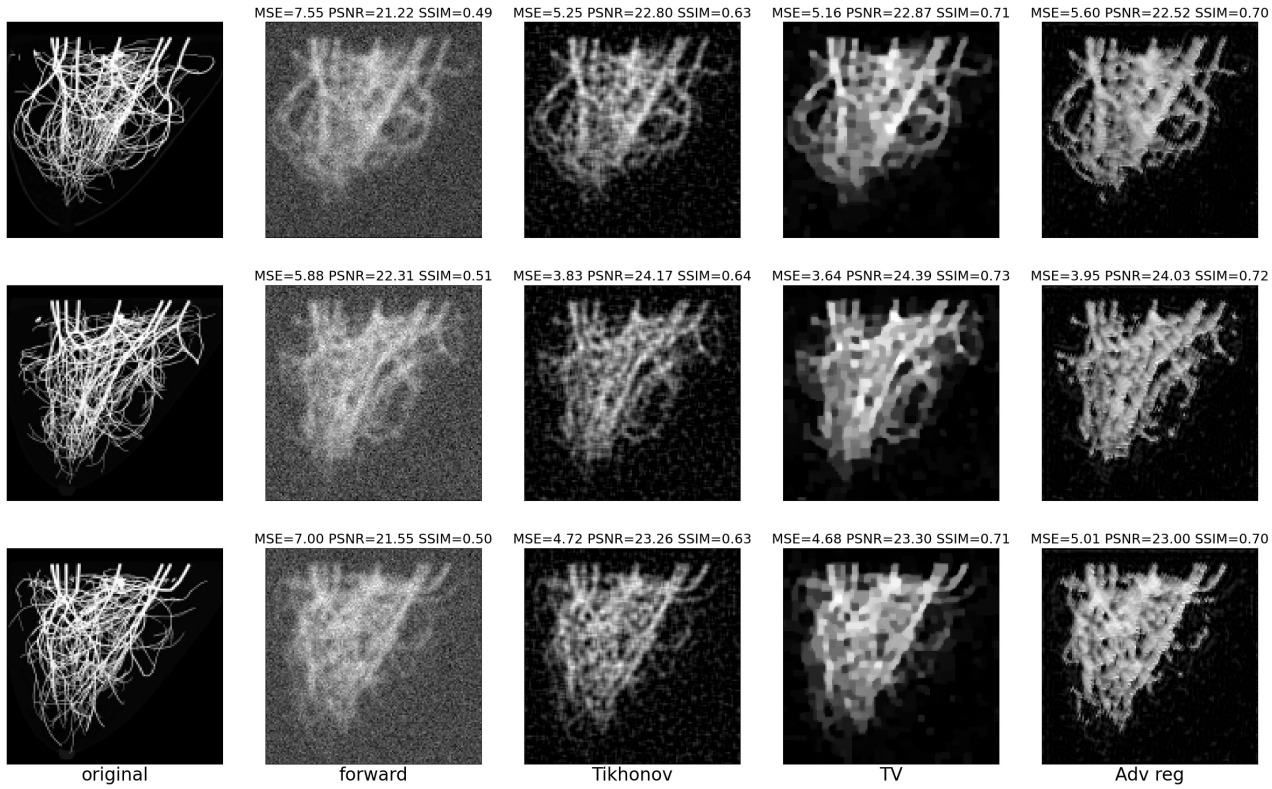

Figure 3.5: The reconstruction from Gaussian blurring with kernel size=5, standard deviation=2.0, noise deviation=0.05 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

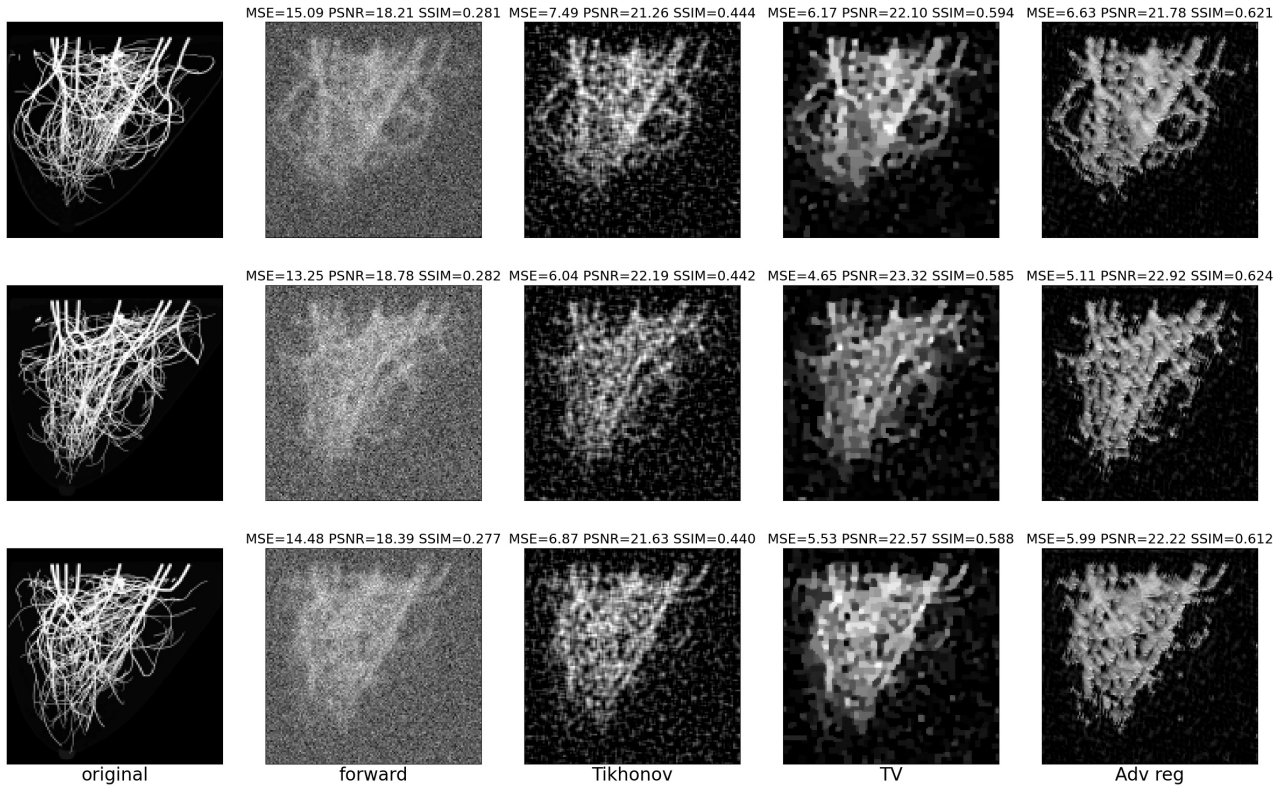

Figure 3.6: The reconstruction from Gaussian blurring with kernel size=5, standard deviation=2.0, noise deviation=0.1 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

Deep Model

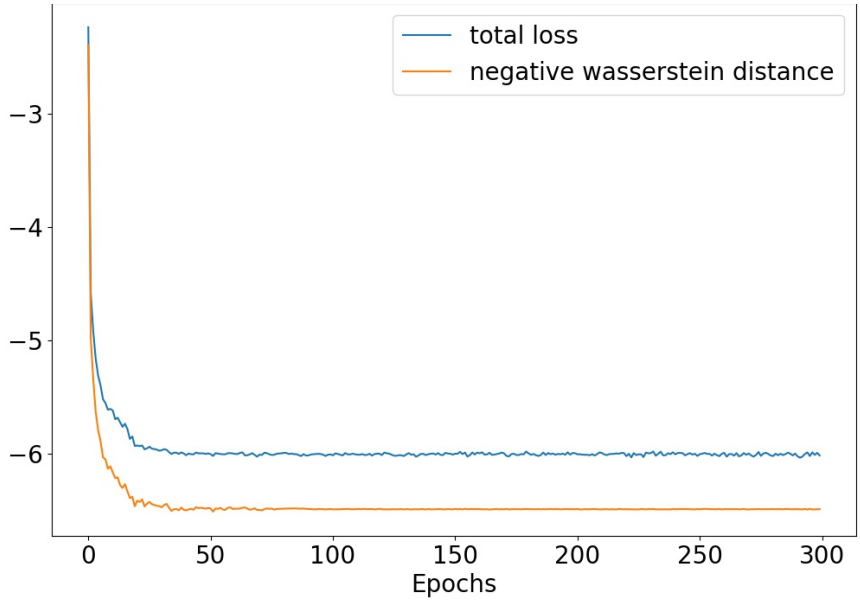

Figure 3.7: The total training losses and Wasserstein distance between true images and their coarse reconstruction with Tikhonov regularization over 300 training epochs.

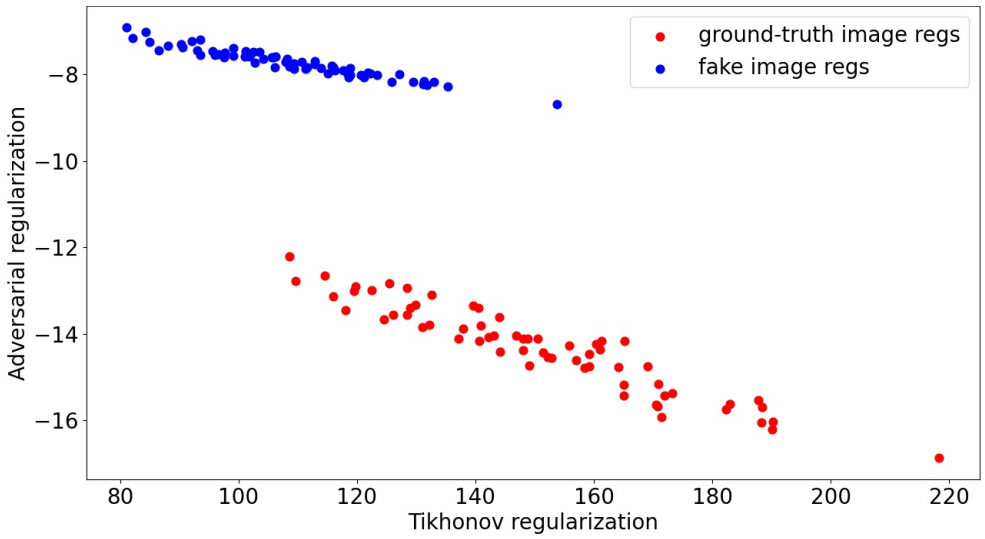

Figure 3.8: The Tikhonov regularization against the learned adversarial regularization over 100 ground-truth images and their corresponding reconstruction using standard Tikhonov regularization.

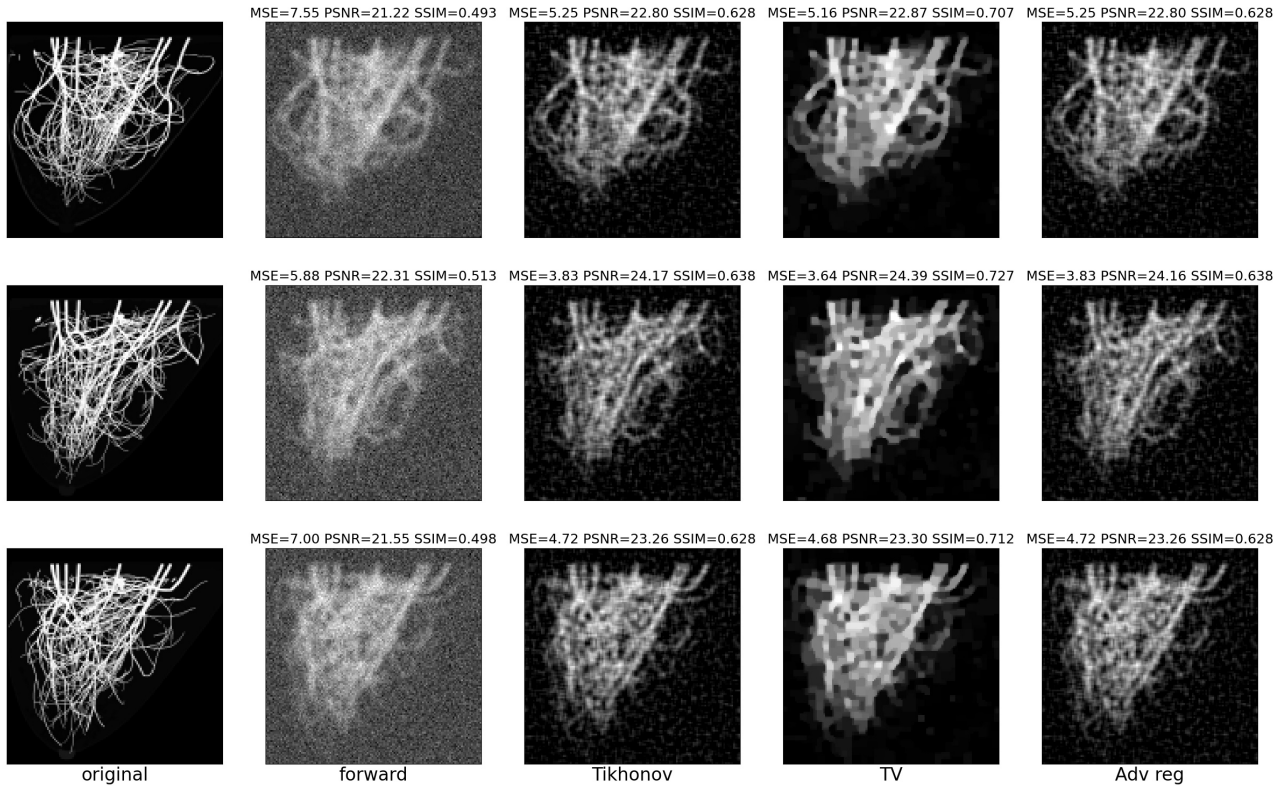

Figure 3.9: The reconstruction from Gaussian blurring with kernel size=5, standard deviation=2.0, noise deviation=0.05 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

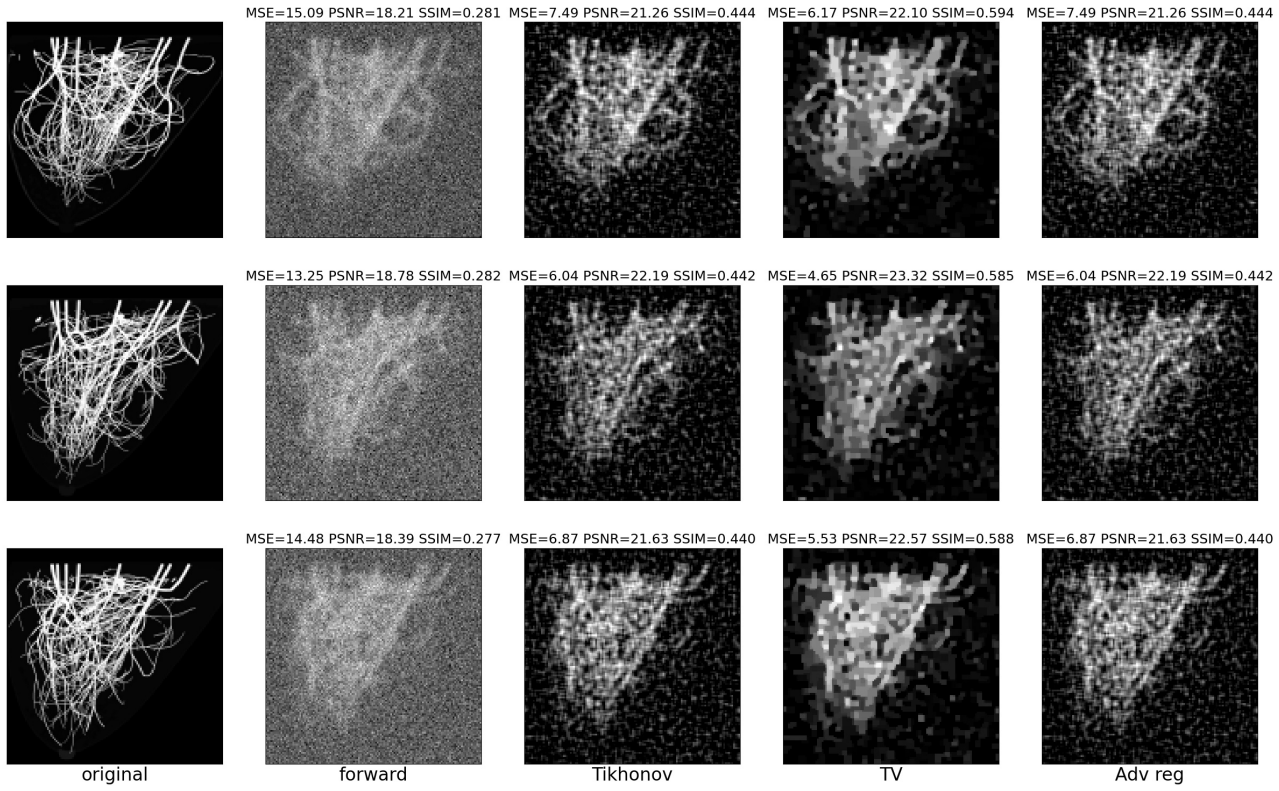

Figure 3.10: The reconstruction from Gaussian blurring with kernel size=5, standard deviation=2.0, noise deviation=0.1 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

3.6.2 Radon Transform

Shallow Model

For Radon transform reconstruction problem, the training dataset is reconstructed from Radon transform with number of views per-ray=45, number of rays=181, noise deviation=0.1 using Tikhonov regularization regularization.

Table 3.4: **Numerical Experiments for Radon Transform**.

model	training set	critic curves	Reconstruction	Regularization	strength	mse avg(10^{-3})	ssim avg
Naive	Tikhonov	3.11, 3.12	3.13,3.14	Tikhonov	1.0	2.46	0.760
				TV	0.1	1.87	0.876
				Adv	(1,1000)	2.36	0.808
Deep	Tikhonov	3.15, 3.16	3.17, 3.18	Tikhonov	1.0	2.46	0.760
				TV	0.1	1.87	0.876
				Adv	(1.0, 0.1)	2.46	0.760

The MSE and SSIM are obtained by reconstructing 100 images using Adversarial regularizer. The results are produced by Radon transform problem with number of views per-ray=45, number of rays=181, noise deviation=0.1. The regularization α and β of using adversarial regularizer is formatted as (α, β).

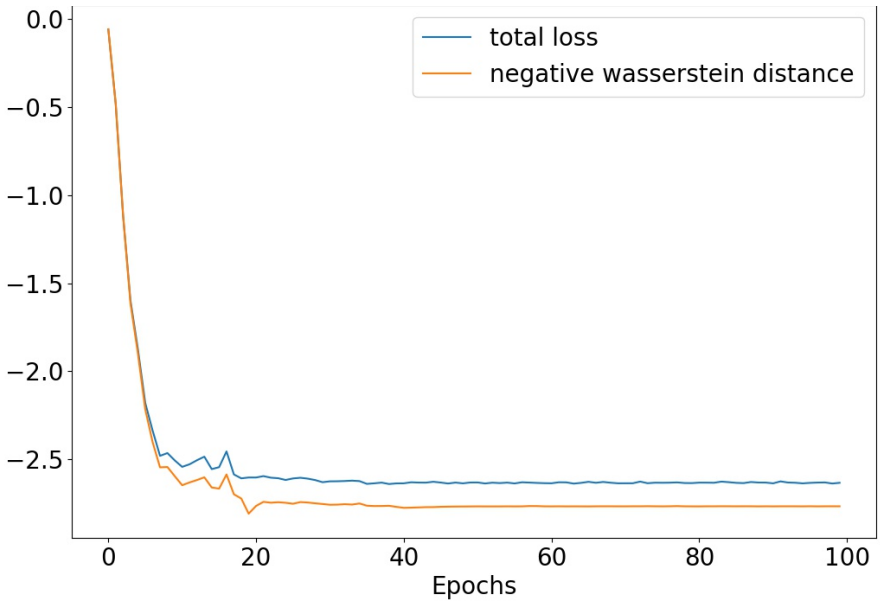

Figure 3.11: The total training losses and Wasserstein distance between true images and their coarse reconstruction with Tikhonov regularization over 200 training epochs.

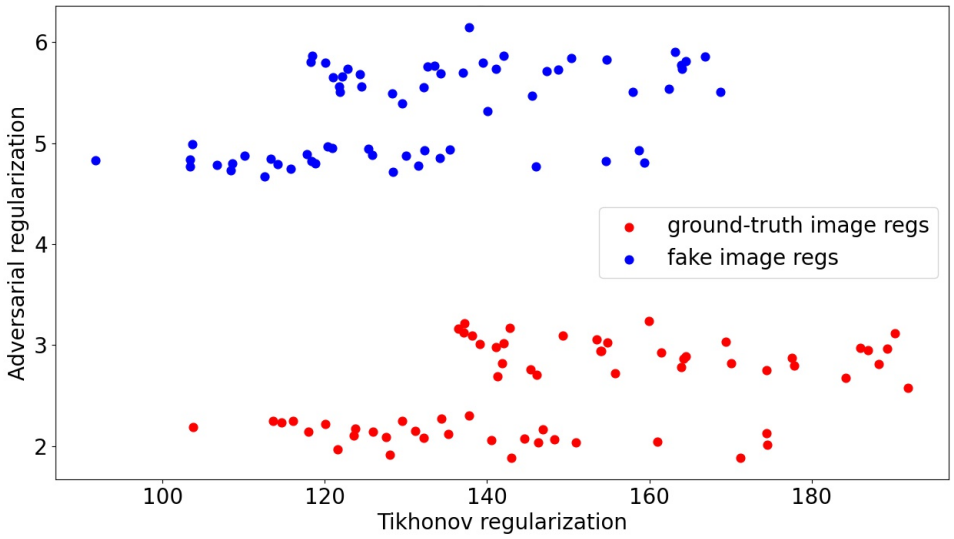

Figure 3.12: The Tikhonov regularization against the learned adversarial regularization over 100 ground-truth images and their corresponding reconstruction using standard Tikhonov regularization.

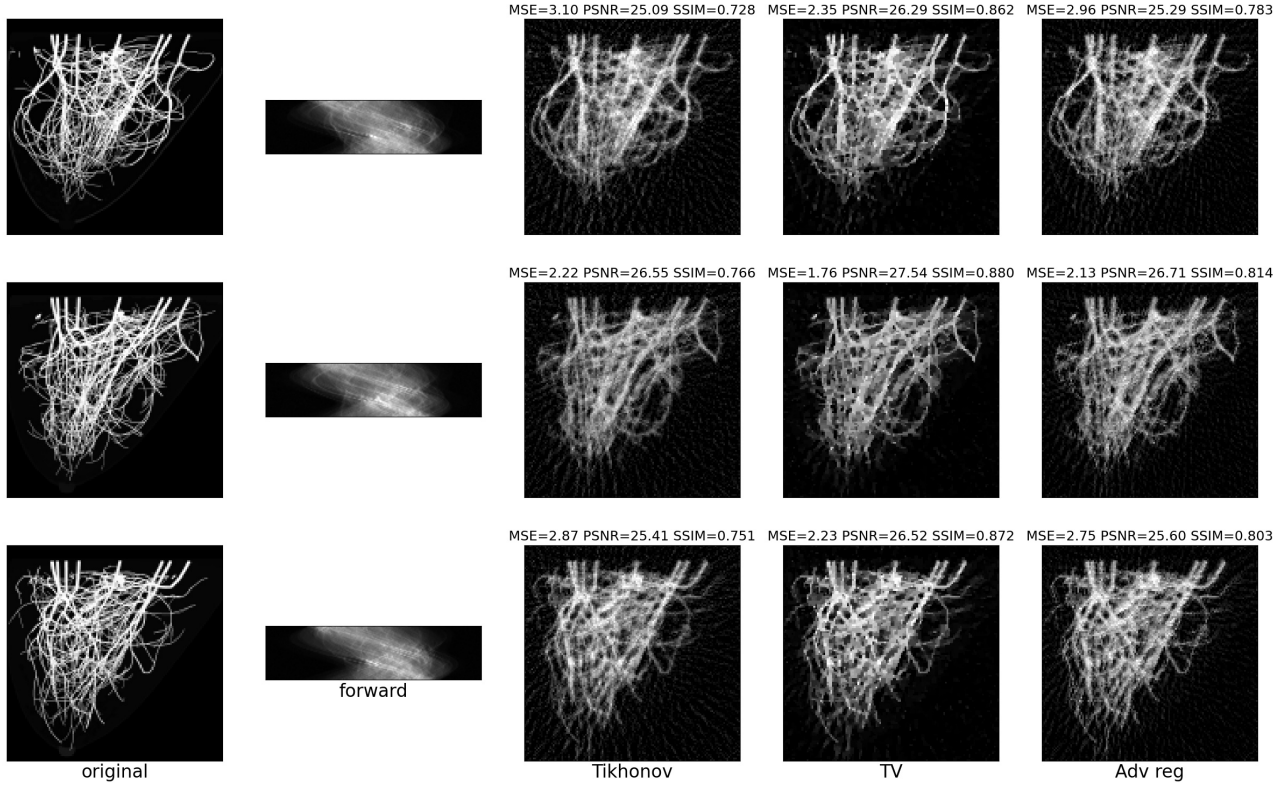

Figure 3.13: The reconstruction from Radon transform with number of views per-ray=45, number of rays=181, noise deviation=0.1 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

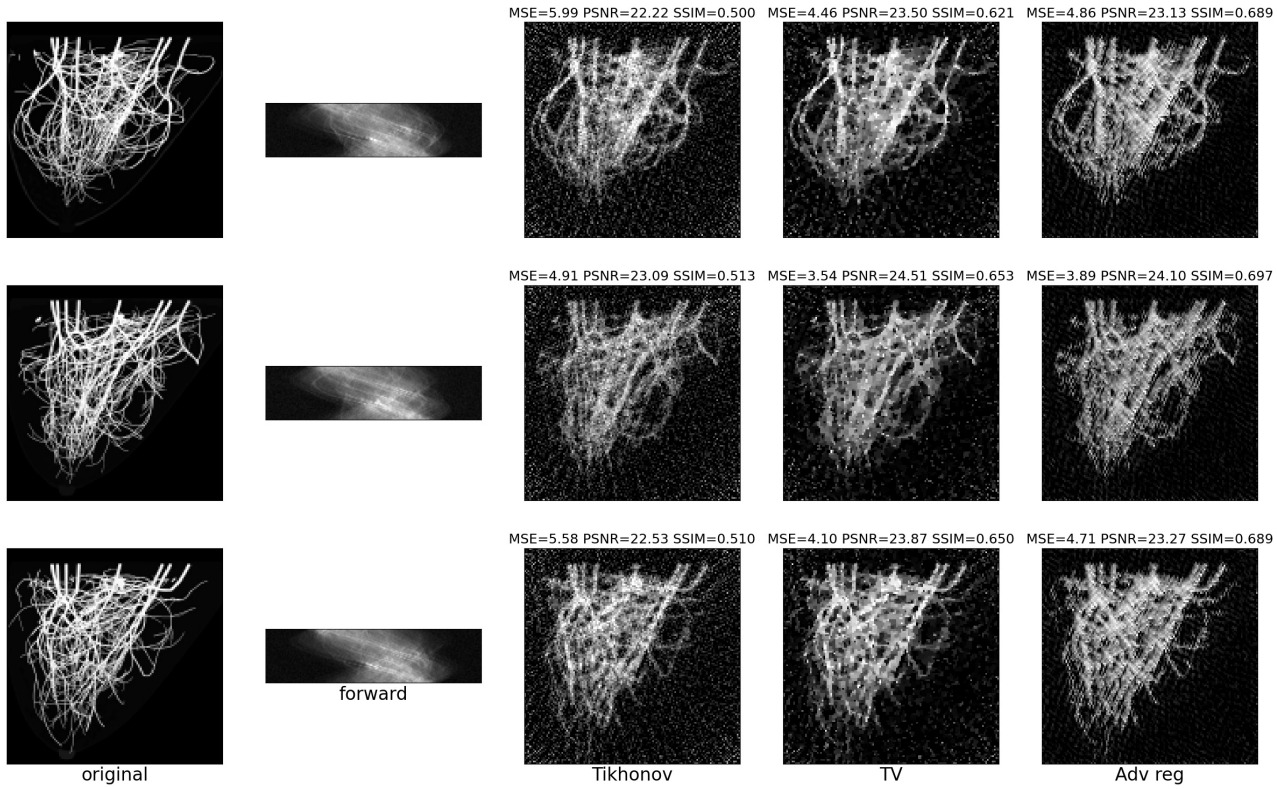

Figure 3.14: The reconstruction from Radon transform with number of views per-ray=45, number of rays=181, noise deviation=0.5 using Tikhonov regularization, Total Variation regularization and learned adversarial regularization.

Deep Model

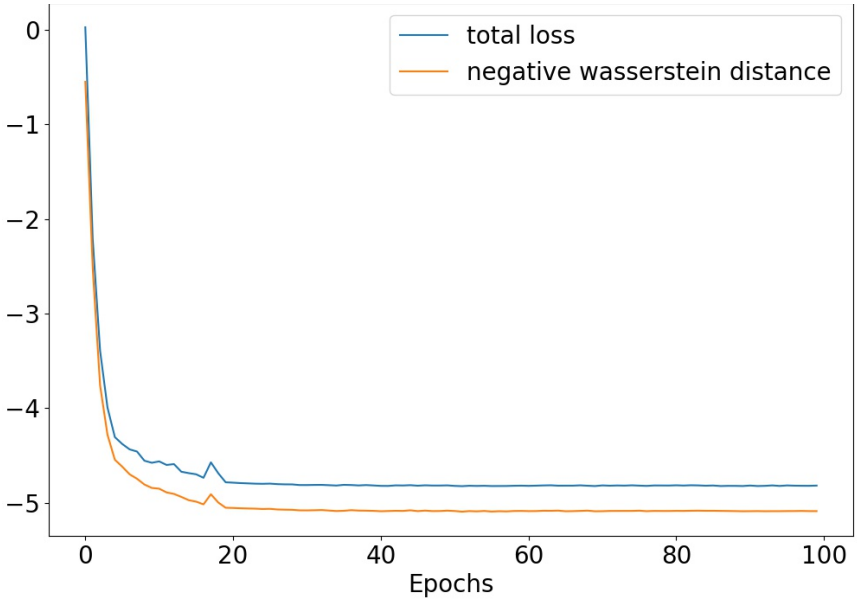

Figure 3.15: The total training losses and wasserstein distance between true images and their coarse reconstruction with Tikhonov regularization over 200 training epochs.

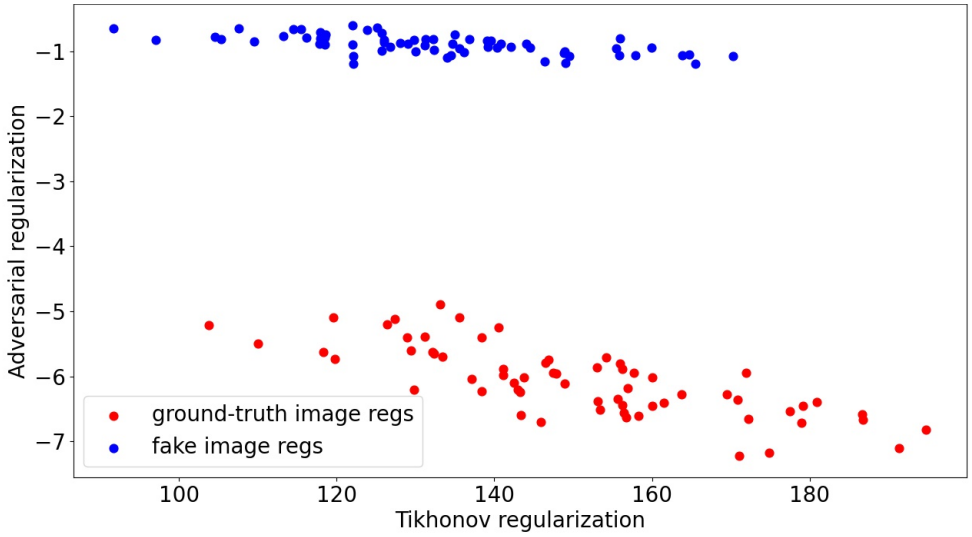

Figure 3.16: The Tikhonov regularization against the learned adversarial regularization over 100 ground-truth images and their corresponding reconstruction using standard Tikhonov regularization.

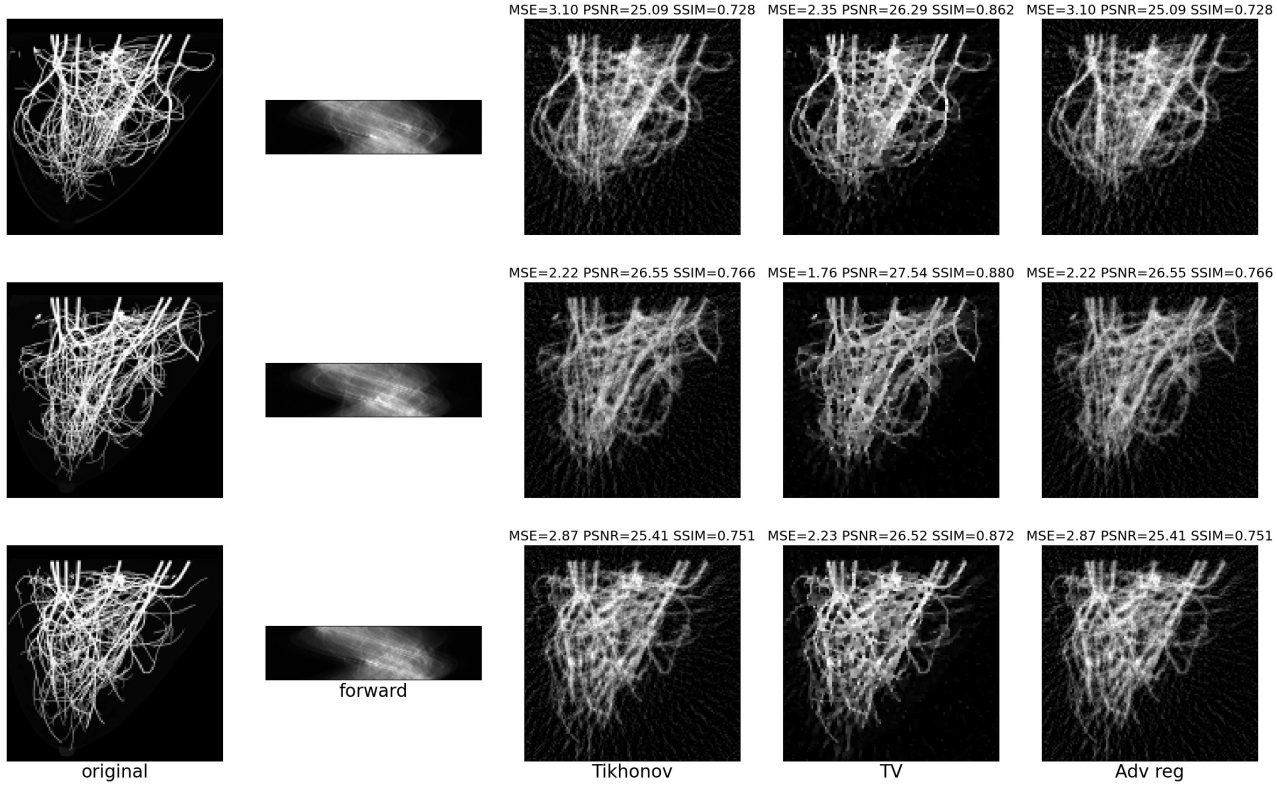

Figure 3.17: The reconstruction from Radon transform with number of views per-ray=45, number of rays=181, noise deviation=0.1 using Total Variation regularization, Total Variation regularization and learned adversarial regularization.

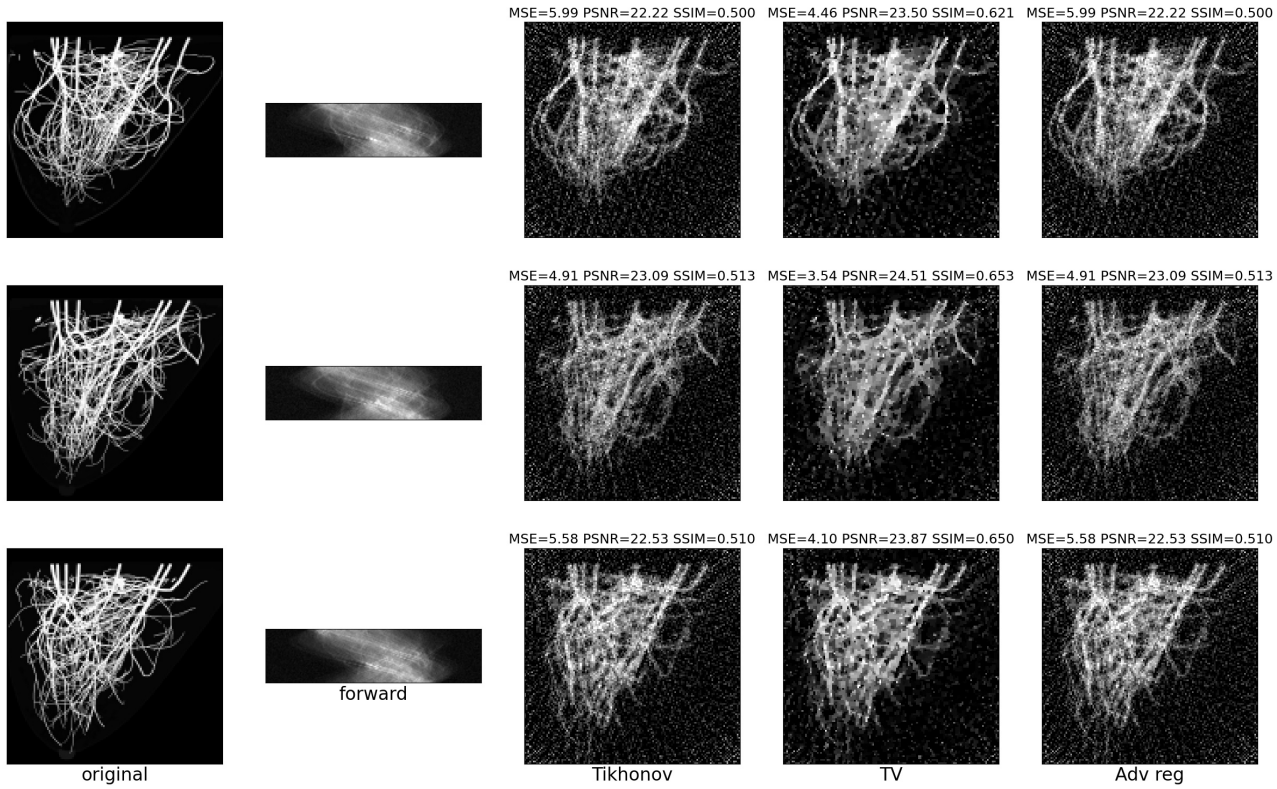

Figure 3.18: The reconstruction from Radon transform with number of views per-ray=45, number of rays=181, noise deviation=0.1 using Total Variation regularization, Total Variation regularization and learned adversarial regularization.

The experimental results in Figure 3.4, 3.8, 3.12, 3.16 demonstrated that the proposed networks, both the shallow and deep network architectures, are able to distinguish the ground-truth images from their naive reconstruction using other regularization term from Gaussian blurring. However, as is shown in Figure 3.5, 3.13, the reconstruction, which uses the same setting as the training dataset, performance using the proposed shallow adversarial regularizer is slightly better than the Tikhonov regularization while worse than the T.V. regularization. But as the noise level increases, the adversarial regularizer outperforms both Tikhonov and T.V. regularization as is shown in Figure 3.10, 3.14. Finally, as is shown in Figure 3.9, 3.17, there is no big difference between the performance of the shallow architecture and the deep architecture.

3.7 Remarks

As demonstrated by the experimental results, the trained adversarial regularizer is able to differentiate the ground-truth images from their corresponding naive reconstruction using classical regularization methods. However, the trained adversarial regularizer performs poorly to solve the inverse problem with the same setting as generating the training dataset: slightly better than the Tikhonov regularization but worse than the Total Variation regularization. While it works way better as the noise level goes higher, generally better than the Tikhonov and Total Variation regularization in both Gaussian deblurring problem and inverse Radon transform. Finally, the shallower neural network architecture performs even better than the deep architecture.

Chapter 4

Regularization by Denoising (RED)

In this chapter, a powerful and flexible image reconstruction framework, regularization by denoising (RED), will be discussed in detail. We will first discuss the basic principle of RED, which is built on the top of image-adaptive Laplacian filter. Next, we summarize the conditions under which the RED framework works. Subsequently, we will demonstrate how denoisers can be used to solve inverse problem in image reconstruction based on the RED model. Finally, we will present the image denoiser used in this thesis, which uses a deep convolutional neural network.

4.1 Laplacian Regularization

A proper denoiser f is able to remove the noise of a noisy image and create a reasonably clean noise-free image that approximates the ground-truth image. A denosier can be described as a pseudo-linear filter which is in the form of

$$f(m) = W(m)m, \tag{4.1}$$

where $f(m)$ is the denoiser, $W(m)$ denotes the pseudo-linear filter, and m is the noisy image unrolled as a vector.

Then, according to [22], the image-adaptive Laplacian, which is widely used to regularize a variety of inverse problems, can be formulated as

$$\begin{aligned} R_L(m) &= \frac{1}{2} m^{\mathrm{T}} L(m) m \\ &= \frac{1}{2} m^{\mathrm{T}} (I - W(m)) m \\ &= \frac{1}{2} m^{\mathrm{T}} (m - W(m)m). \end{aligned} \tag{4.2}$$

where $L(m) = I - W(m)$ is the image-adaptive Laplacian filter. The cost function of the inverse problem is then written as

$$m^* = \arg\min_m \frac{1}{2} \parallel F(m) - d \parallel_2^2 + \frac{\lambda}{2} m^{\mathrm{T}} (m - W(m)m). \tag{4.3}$$

One key challenge in solving equation (1.3) using gradient-based methods is the computation of the gradient of $W(m)$ with respect to m. In RED, $W(m)m$ is replaced by an arbitrary denoiser $f(m)$. The regularization term $R_L(m)$ can be rewritten as

$$R_L(m) = \frac{1}{2} m^{\mathrm{T}} (m - f(m)). \tag{4.4}$$

Using this formulation, no explicit $W(m)$ is needed, instead, a more general filter $f(m)$ is introduced. We also observe that the regularization term will equal to zero if one of the following conditions is satisfied where 1) $m = 0$, 2) $m = f(m)$, 3) m and $f(m)$ is orthogonal.

The gradient of the Laplacian regularizer can be computed as

$$\nabla_m R_L(m) = m - \frac{1}{2} \nabla_m \{m^{\mathrm{T}} f(m)\}, \tag{4.5}$$

The above expression can be further simplified as

$$\nabla_m R_L(m) = m - f(m). \tag{4.6}$$

4.2 Underlying Conditions

There are four basic underlying conditions that ensure the derivation of the gradient of $R_L(m)$ is equation (4.6) and impose the convexity of entire cost function: 1) Differentiability, 2) Local homogeneity, 3) Symmetric Jacobian, and 4) the Passivity.

Condition 1: Differentiability

Condition 1 requires the denoiser to have a directional derivative. Mathematically, the directional derivative is defined as

$$\begin{aligned}\nabla_m f(x) &= \lim_{\gamma \to 0} \frac{f(x+\gamma m) - f(x)}{\gamma} \\ &= \lim_{\gamma \to 0} \frac{f(x) + \gamma \nabla f(x)^{\mathrm{T}} m + o(\gamma) - f(x)}{\gamma} \\ &= \nabla f(x)^{\mathrm{T}} m.\end{aligned} \tag{4.7}$$

To obtain the expression above, $f(x+\gamma u)$ is expanded based on Taylor's expansion, and the definition of $o(g)$ ($\lim_{\gamma \to 0} \frac{o(\gamma)}{\gamma} \to 0$) is used.

Condition 2: Symmetry Jacobian

Condition 2 constrains the denoiser to have a symmetric Jacobian, which means

$$\nabla_m f(m) = [\nabla_m f(m)]^{\mathrm{T}} \tag{4.8}$$

Condition 3: (Local) Homogeneity

The definition of homogeneity is that given a scale factor c, the scaled image cm will result in corresponding scaled output of the denoiser $cf(m)$. Local homogeneity relaxes this condition to the case of c such that $|c-1| \leq \epsilon$ where $\epsilon \to 0$. The directional derivative of

$f(m)$ along m direction can be expressed as

$$\begin{aligned}\nabla_m f(m)m &= \frac{f((1+\epsilon)m) - f(m)}{\epsilon} \\ &= \frac{(1+\epsilon)f(m) - f(m)}{\epsilon} \\ &= f(m).\end{aligned} \tag{4.9}$$

Condition 4: Strong Passivity

The strong passivity is satisfied if the spectral radius of the Jacobian of $f(m)$ is no greater than 1.

$$\eta\{\nabla_m f(m)\} \leq 1, \tag{4.10}$$

where η denotes the spectral radius of the Jacobian of $f(m)$. Then the Hessian of $R_L(m)$,

$$\nabla\{\nabla R_L(m)\} = \nabla\{m - f(m)\} = I - \nabla f(m) \succeq 1. \tag{4.11}$$

The positive definiteness of the Hessian of $f(m)$ guarantees the convexity of $R_L(m)$.

4.3 Denosiers

In RED, $f(m)$ can be an arbitrary denoiser as long as the above conditions are satisfied. Commonly chooses of $f(m)$ include median filters, Total Variation filters, and the state-of-the-art convolutional neural network-based denoiser. In this thesis, the denoiser we used is a deep convolutional neural network (DnCNN). As suggested by the gradient of the Laplacian regularizer, instead of learning a denoiser $f(m)$ outputting the clean image, we learn the residual $m - f(m)$ via DnCNN.

The network architecture is inspired by [1, 9, 8], but no batch normalization is used in our case since training images are already scaled. The basic block is convolutional layer followed by a Relu activation. The entire architecture is shown in Table 4.1.

The network is trained is learn the residual between the ground-truth image m and its denoised image through the denoiser $f(m)$. At each training epoch, additive Gaussian noise n_σ is added to the ground-truth image and output of the network is n_σ. We also relax the conditions of the denoiser in section 4.2, as DnCNN does not have the symmetric Jacobian and passivity.

Table 4.1: **DnCNN Architecture**.

Name	**Layer Type**	**Filter Size** ($K_1 \times K_2 \times C_{in} \times C_{out}$)	**Feature Map Size** ($H_{out} \times W_{out} \times C_{out}$)
	Input		$128 \times 128 \times 1$
Conv1	convolution	$3 \times 3 \times 1 \times 64$	$128 \times 128 \times 64$
Conv2	convolution	$3 \times 3 \times 64 \times 64$	$128 \times 128 \times 64$
Conv3	convolution	$3 \times 3 \times 64 \times 64$	$128 \times 128 \times 64$
Conv4	convolution	$3 \times 3 \times 64 \times 64$	$128 \times 128 \times 64$
Conv5	convolution	$3 \times 3 \times 64 \times 64$	$128 \times 128 \times 64$
Conv6	convolution	$3 \times 3 \times 64 \times 64$	$128 \times 128 \times 64$
Conv7	convolution	$3 \times 3 \times 64 \times 1$	$128 \times 128 \times 1$

4.4 Optimization Strategy

The data fidelity term in the cost function is convex, the regularization term is also convex under the assumption of passivity. Then the entire cost function is convex, hence, a variety of convex optimization strategy can be applied to solution this minimization problem.

For a linear forward model $F(m) = Am$ (as those considered in this thesis), we write the optimization problem

$$m^* = \arg\min_m \frac{1}{2} \parallel F(m) - d \parallel_2^2 + \frac{\lambda}{2} m^{\mathrm{T}}(m - W(m)m). \tag{4.12}$$

First order optimality conditions require the gradient of the cost function to vanish, that is

$$A^{\mathrm{T}}(Am - d) + \lambda(m - f(m)) = 0. \tag{4.13}$$

Equation (1.12) can be solved using the fixed point iteration

$$m_{k+1} = (A^{\mathrm{T}}A + \lambda A)^{-1}(A^{\mathrm{T}}d + \lambda f(m_k)). \tag{4.14}$$

$$A^{\mathrm{T}}(Am - d) + \lambda(m - f(m)) = 0, \tag{4.15}$$

However, the computation of $(A^{\mathrm{T}}A + \lambda A)^{-1}$ is infeasible if the dimension of $A^{\mathrm{T}}A$ is large, which leads us to gradient-based method. For simplicity, this thesis will use vanilla gradient descent as shown in Algorithm 4.

Algorithm 4 Steepest Gradient Descent For RED

Require: the measurement d, forward operator A, regularization strength λ, step size α, noise standard deviation σ

Initialization $m_0 = d$

for $k = 1, 2 \ldots, N$ **do**

Denoising $m_k = f_\sigma(m_{k-1})$

Update $m_k = m_{k-1} - \alpha[A^{\mathrm{T}}(Am_{k-1} - d) + \lambda(m_{k-1} - m_k)]$

4.5 Numerical Results

For both Gaussian deblurring and Radon transform problem, the training dataset is generated with the additive Gaussian with zero mean and 0.001 standard deviation. The training curves of the DnCNN is shown in Figure 4.1. The total loss of the network converges.

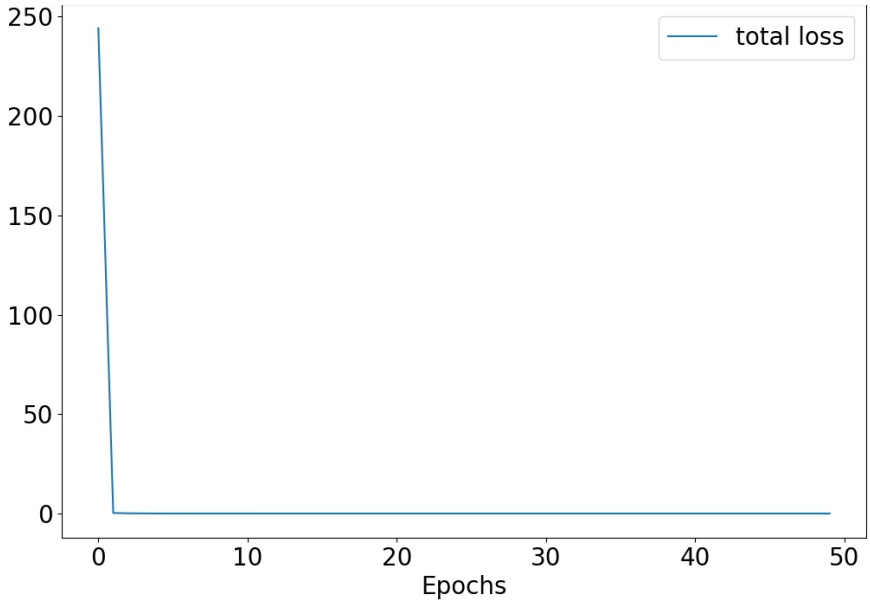

Figure 4.1: The total training losses of the DnCNN.

4.5.1 Gaussian Deblurring

The reconstruction test is performed on Gaussian Blurring image with kernel size=5, standard deviation = 2, the noise is the additive Gaussian noise with zero mean and 0.05 standard deviation.

Table 4.2: **Numerical Experiments for Gaussian Deblurring**.

kernel-size	kernel-std	noise std	Regularization	strength	MSE(10^{-3})	SSIM	Curves	Results
5	2.0	0.05	Tikhonov	0.1	4.0, (2.8, 5.7)	0.65, (0.61, 0.67)		4.2
5	2.0	0.05	TV	0.01	3.5, (2.2, 5.1)	0.75, (0.69, 0.80)		4.2
5	2.0	0.05	Adv	(0.1, 1000)	4.4, (3.1, 5.6)	0.71, (0.68, 0.75)		4.2
5	2.0	0.05	RED	10	5.0, (3.8, 6.3)	0.67, (0.51, 0.78)	4.3	4.2

The mse and ssim columns, formatted as [average, (minimum, maximum)], are computed by reconstructing 100 images. The regularization α and β in adversarial regularizer is formatted as (α, β) in the strength column.

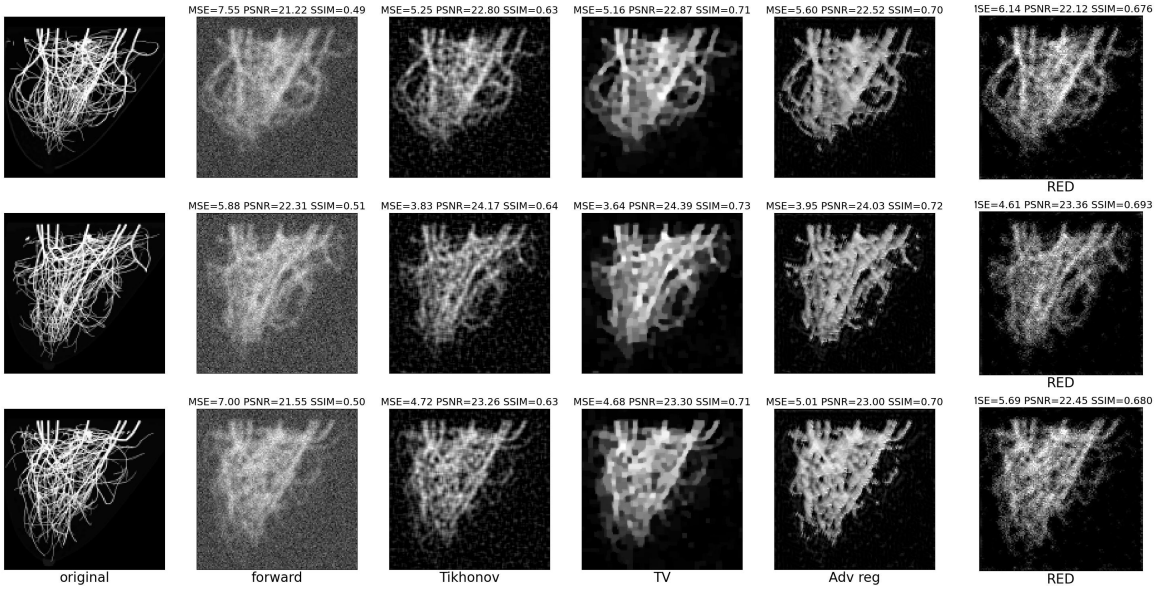

Figure 4.2: The Gaussian deblurring reconstruction samples with a kernel size=5, standard deviation=2, noise is an additive Gaussian noise with zero mean and standard deviation=0.05.

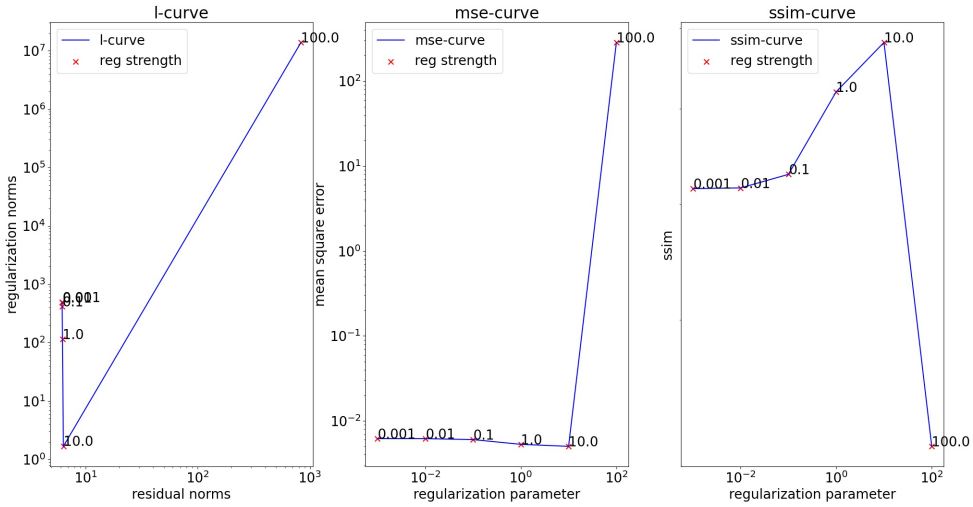

Figure 4.3: Curves for Gaussian deblurring using RED reconstruction with with a kernel size=5, standard deviation=2, noise is an additive Gaussian noise with zero mean and standard deviation=0.05. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3]. The optimal regularization strength suggested by the curves is 10.

4.5.2 Radon Transform

The reconstruction test is performed on Radon transform with number of views per-ray=45, number of rays=181, and additive Gaussian noise with zero mean and 0.1 standard deviation.

Table 4.3: **Numerical Experiments for Radon Transform**.

num-rays	num-views	noise std	Regularization	Strength	MSE(10^{-3})	SSIM	Curves	Results
181	45	0.1	Tikhonov	1.0	2.3, (1.5, 3.4)	0.79, (0.76, 0.83)		4.4
181	45	0.1	TV	0.1	1.6, (0.8, 2.6)	0.90, (0.85, 0.93)		4.4
181	45	0.1	Adv	(1.0,1000)	2.3, (1.6, 3.0)	0.80, (0.77, 0.83)		4.4
181	45	0.1	RED	100	2.0, (1.5, 2.6)	0.89, (0.84, 0.94)	4.5	4.4

The MSE and SSIM columns, formatted as [average, (minimum, maximum)], are produced by reconstructing 100 images using different regularization. The regularization α and β in adversarial regularizer is formatted as (α, β) in the strength column

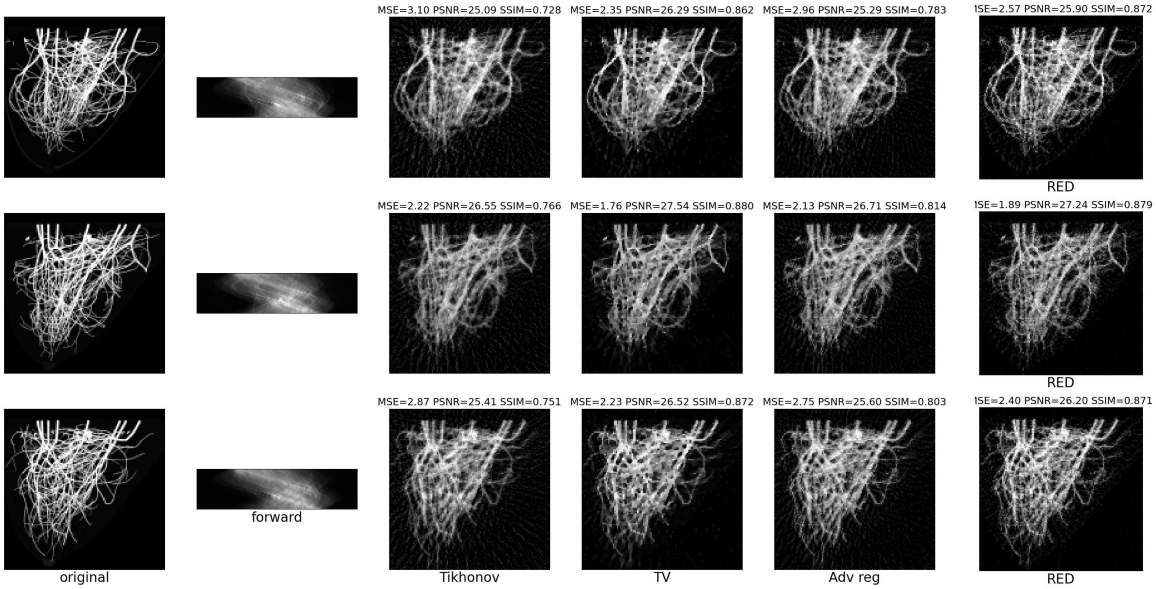

Figure 4.4: The Radon transform reconstruction samples with number of rays=181, number of views=45, noise is an additive Gaussian noise with zero mean and standard deviation=0.1.

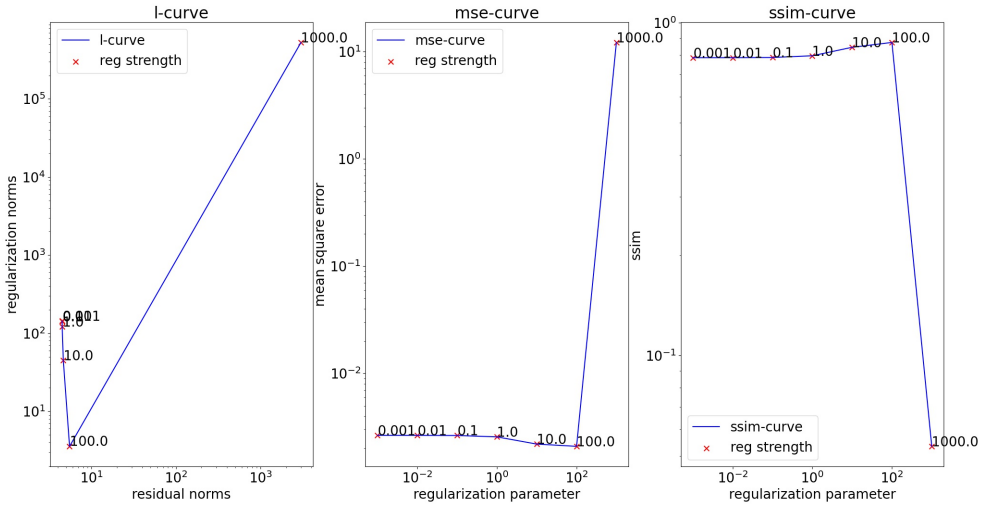

Figure 4.5: Curves for Radon transform using Total Variation regularization with number of views=45, additive Gaussian noise with zero mean and standard deviation=0.1. From left to right is 1) L-curve; 2)MSE-curve; 3) SSIM-curve. The regularization parameter candidates are [1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3]. The optimal regularization strength suggested by the curve is 100.

4.6 Remarks

The experiment demonstrated that the L-curve, MSE-curve, and SSIM-curve is able to help us select the best regularization strength. The results using RED outperforms those using the adversarial regularizer in the inversion of Radon transform task. While the adversarial regularizer outperforms the RED in the Gaussian deblurring problem.

Chapter 5

Conclusion

In this thesis, two commonly encountered inverse problems in image reconstruction have been discussed: Gaussian deblurring problem and the inversion of Radon transform. We have first presented the mathematical formulation of the inverse problem involving Gaussian blurring and Radon transform. Model-based variational regularization formulations as well as numerical optimization algorithm were introduced subsequently to solve this model-based inverse problem. Finally, we focused on the design of the regularization functional and explored both classical regularization functional (such as Tikhonov and Total Variation regularization) and modern data-driven regularization methods. The whole framework was built on the top of PyTorch taking advantage of PyTorch automatic differentiation, GPU acceleration and sparse matrix operation [1].

In Chapter 3, we implemented and numerically assessed an adversarial regularizer approach. The key idea of the method is that a proper regularizer should be able differentiate ground-truth images from the naively reconstructed images. Two neural network architectures, a shallow architecture inspired by Total Variation and a deeper architecture inspired by WGANs, were trained for this task. A penalty term was incorporated into to loss function to satisfy the 1-Lipschitz constraint during the training of the network. The numerical experiments demonstrated that the proposed adversarial regularizers were able to effectively differentiate the ground-truth images from those naively reconstructed ones. However, the proposed adversarial regularizer did not show significant improvement in the quality of the reconstructed images. In particular, the learned regularizer performed not better than Tikhonov and T.V. regularization in the low noise case. The possible reason is that

[1] https://github.com/uvilla/ganPrior

the images used in this thesis is extremely hard to reconstruct as they contains thin tubular structures. However, some improvement in image quality was observed for extremely noisy data. Furthermore, the deeper network architecture of the adversarial regularizer works even worse than the shallower architecture. One possible reason might be the loss of convexity of the deeper architecture, which makes the iterative optimization hard to converge.

In Chapter 4, we discussed the regularization by denoising (RED) approach. We briefly introduced the image-adaptive Laplacian as a precursor of RED. Next, we summarized the underlying hypotheses in the RED framework, which are needed to derive the gradient of the cost function and ensure the convexity of the cost function. The CNN-based denoiser (DnCNN) was designed to serve as the denoiser in our case. We further discuss the numerical optimization algorithms to minimize the cost function in the reconstruction phase. The L-curve, in conjunction with the mean-square-error curve and structural similarity curve were used to select the optimal regularization strength. Finally, we compared the performance of the four regularization methods (Tikhonov, T.V., Adversarial regularizer, and RED) in this thesis. The RED approach outperformed the Tikhonov and Adversarial regularizer in the inversion of Radon transform problem and was close to the T.V. approach. In the Gaussian deblurring task, the RED worked worse than the Adversarial regularizer and just slightly better than the Tikhonov regularization. One possible reason is that the DnCNN denoiser might not satisfy some of the underlying conditions of the RED framework. In fact, the DnCNN was not restricted to satisfy the strong passivity, because there was no constraint on the DnCNN to enforce the spectral radius of its Jacobian to be no greater than 1. This leads to the loss of convexity in the cost function making the numerical optimization difficult.

References

[1] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3929–3938, 2017.

[2] 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.

[3] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[4] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[5] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[6] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, IEEE, 2013.

[7] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," 2017.

[8] Y. Sun, J. Liu, and U. S. Kamilov, "Block coordinate regularization by denoising," in *Proc. Ann. Conf. Neural Information Processing Systems (NeurIPS)*, May 2019.

[9] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov, "Rare: Image reconstruction using deep priors learned without ground truth," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1088–1099, 2020.

[10] S. Lunz, O. Öktem, and C.-B. Schönlieb, "Adversarial regularizers in inverse problems," *arXiv preprint arXiv:1805.11572*, 2018.

[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[13] S. Mukherjee, S. Dittmer, Z. Shumaylov, S. Lunz, O. Öktem, and C.-B. Schönlieb, "Learned convex regularizers for inverse problems," *arXiv preprint arXiv:2008.02839*, 2020.

[14] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[15] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, p. 503–528, Aug. 1989.

[16] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[17] A. Badano, C. G. Graff, A. Badal, D. Sharma, R. Zeng, F. W. Samuelson, S. J. Glick, and K. J. Myers, "Evaluation of Digital Breast Tomosynthesis as Replacement of Full-Field Digital Mammography Using an In Silico Imaging Trial," *JAMA Network Open*, vol. 1, pp. e185474–e185474, 11 2018.

[18] P. C. Hansen and J. S. Jørgensen, "Air tools ii: algebraic iterative reconstruction methods, improved implementation," *Numerical Algorithms*, vol. 79, no. 1, pp. 107–137, 2018.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[20] C. Villani, *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, 2008.

[21] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[22] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, 2013.

Vita

Peijie(Ricardo) Qiu

Degrees

B.E. Chongqing University of Post and Telecommunication, Electronic Information Engineering, June 2018
B.E. Northern Arizona University, Electrical Engineering, May 2018

Publications

Peijie Qiu, Xilun Liu, Siwei Wen, Kyle N.Winfree, Chun-Hsing Ho, "Development of an IoT Instrumented Bike: for Assessment of Road and Bike Trail Conditions", IEEE ISSI(International Symposium on Sensing and Instrumentation in IoT Era) Conference, Sept. 2018 (**Best Student Paper Award**)

May 2021

Data-Driven Approaches to Solve Inverse Problems, Qiu, M.S. 2021