Self-supervised Bayesian Deep Learning for Image Recovery with Applications to Compressive Sensing

Tongyao Pang[1], Yuhui Quan[2], and Hui Ji[1]

[1] Department of Mathematics, National University of Singapore, 119076, Singapore
[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
matpt@nus.edu.sg, csyhquan@scut.edu.cn, matjh@nus.edu.sg

Abstract. In recent years, deep learning emerges as one promising technique for solving many ill-posed inverse problems in image recovery, and most deep-learning-based solutions are based on supervised learning. Motivated by the practical value of reducing the cost and complexity of constructing labeled training datasets, this paper proposed a self-supervised deep learning approach for image recovery, which is dataset-free. Built upon Bayesian deep network, the proposed method trains a network with random weights that predicts the target image for recovery with uncertainty. Such uncertainty enables the prediction of the target image with small mean squared error by averaging multiple predictions. The proposed method is applied for image reconstruction in compressive sensing (CS), i.e., reconstructing an image from few measurements. The experiments showed that the proposed dataset-free deep learning method not only significantly outperforms traditional non-learning methods, but also is very competitive to the state-of-the-art supervised deep learning methods, especially when the measurements are few and noisy.

Keywords: Self-supervised learning, Bayesian neural network, Compressive sensing, Image recovery

1 Introduction

Image recovery is about recovering an image of high quality from its related measurement. Many image recovery tasks are to solve a linear inverse problem:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y}$ denotes the available measurement, $\boldsymbol{x}$ denotes the latent image to recover, $\boldsymbol{n}$ denotes the measurement noise which is often modeled by i.i.d. random variables componently. The operator $\boldsymbol{A}$ denotes a linear degradation/measuring process on the latent image, which is usually non-invertible/ill-conditioned. When $\boldsymbol{A}$ is non-invertible such that the number of unknowns is larger than the number of independent equations, the solution is not unique. How to resolve such solution ambiguity is the main challenge when solving (1). When $\boldsymbol{A}$ is invertible but ill-conditioned, how to suppress the magnification of measurement noise $\boldsymbol{n}$ during the recovery becomes the main concern. In the past, the most prominent approach is the regularization method, which imposes certain image prior on the solution for resolving solution ambiguities and suppressing noise amplification.

Recently, deep learning emerges as a powerful tool for solving many image recovery problems ; see *e.g.* [32, 46, 26, 45, 44, 38, 33, 34, 25, 11]. These deep-learning-based solutions are all using supervised learning, *i.e.*, a DNN (deep neural network) is trained on a labeled dataset which contains a large amount of the pairs of measurement and truth image. In supervised learning, the performance of the model will be significantly impacted by the characters of training dataset, including the amount of training samples and the correlation between the dataset and target image. In many scenarios, it is often very costly or infeasible to build a large-scale high-quality dataset closely related to the data for processing, *e.g.* magnetic resonance imaging (MRI) and computed tomography (CT) scanning for medical imaging. Certainly, there is a need to develop deep-learning methods for image recovery that provide state-of-the-art performance, while not requesting any additional training data.

In comparison to active ongoing studies on supervised learning methods, there are few works on unsupervised deep learning for solving ill-posed linear systems arising from imaging systems. Recently, image denoising, often served as one sub-module in most image recovery tasks, saw rapid progresses along this line. For example, deep image prior (DIP) [41], SURE-Net[39], and Self2Self denoising [35]. Nevertheless, image denoising is very different from general image recovery tasks. With $\boldsymbol{A} = \mathbf{I}$, image denoising is not an ill-posed problem, and its focus is on noise suppression. In contrast, most image recovery problems require solving an ill-posed linear inverse system. In addition to noise suppression, how to resolve solution ambiguity is another main concern. It is non-trivial to generalize these denoisers to solving ill-posed image recovery problems. One might use these denoisers for post-processing to refine the estimate from an image recovery method. However, this straightforward way does not work well in practice, as the artifacts in estimates are rather different from measurement noises.

There is great practical value of a deep learning method for solving image recovery problems without the need of constructing any training dataset. Thus, this paper aims at developing a self-supervised deep learning method for image recovery and applying it for image reconstruction in CS.

1.1 Main Idea

In our setting, there is no training dataset available for unsupervised learning, and only the sensed measurement $\boldsymbol{y}$ and the sensing matrix $\boldsymbol{A}$ are given. It is shown in DIP [41] that, if an early stop is adopted to avoid overfitting, a convolutional neural network (CNN) tends to predict structured results even on random input. Similar to DIP, we also learn a deep neural network (DNN) $\mathcal{F}_{\boldsymbol{\theta}}$, parameterized by weights $\boldsymbol{\theta}$, to predict $\boldsymbol{x}$, by taking a random initialization $\boldsymbol{\epsilon}_0$

as the DNN input:

$$\boldsymbol{x} = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0). \tag{2}$$

As $\boldsymbol{x}, \boldsymbol{y}$ are related by (2), we have then

$$\boldsymbol{y} = \boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) + \boldsymbol{n}. \tag{3}$$

The maximum likelihood estimate (MLE) for $\boldsymbol{x}$ is given by

$$\min_{\boldsymbol{\theta}} \operatorname{dist}(\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0), \boldsymbol{y}), \tag{4}$$

where $\operatorname{dist}(\cdot, \cdot)$ is determined by the statistical model of the measurement noise, *e.g.* $\|\cdot\|_2^2$ for Gaussian white noise. As there is significant redundancy in the parameters $\boldsymbol{\theta}$, MLE is vulnerable to overfitting. The maximum A posterior (MAP) estimation addresses such an issue by imposing prior knowledge on $\boldsymbol{\theta}$. Let $p(\boldsymbol{\theta})$ denote the prior probability distribution of $\boldsymbol{\theta}$ and consider Gaussian white noise with noise variance $\tilde{\sigma}^2$. Then, an MAP estimator is given by

$$\min_{\boldsymbol{\theta}} \frac{1}{2\tilde{\sigma}^2} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2 - \log p(\boldsymbol{\theta}). \tag{5}$$

It can be seen that an MAP estimator is the MLE estimator regularized by the term relating to the prior of $\boldsymbol{\theta}$. Indeed, the early stopping used in DIP for avoiding overfitting can be interpreted as adding regularization on the MLE with some implicit prior. Other explicit priors are used as well in image classification, *e.g.* the sum-of-squares-based weight decay regularization which assumes the network weights to follow i.i.d. normal distribution [17]. While an early-stopping-based regularization is used in DIP to avoid overfitting in the case $\boldsymbol{A} = \boldsymbol{I}$, the matrix $\boldsymbol{A}$ in many image recovery problems is non-invertible, *i.e.*, there are more unknowns than independent linear equations. In such a case, an MAP estimator with the form of (5) is sometimes not efficient enough to resolve the solution ambiguity, arising from the non-trivial null space of $\mathbf{A}$:

$$\operatorname{null}(\boldsymbol{A}) := \{\boldsymbol{x} \neq 0 : \boldsymbol{A}\boldsymbol{x} = 0\}. \tag{6}$$

Aiming at addressing such ineffectiveness of the MLE/MAP estimator, this paper proposed a self-supervised deep learning method for image recovery, which is built on the framework of Bayesian Neural Network (BNN). Briefly, the weights of a BNN are not deterministic, but random variables following certain probability distributions. Instead of learning deterministic weights, we learn the parameters of the probability distributions of these random weights. The motivation of our approach to tackle ill-posed image recovery problems comes from

(1) Model uncertainty (*i.e.* weight uncertainty) is helpful to correct the prediction bias caused by the network architecture; see *e.g.* [3, 5, 20, 15].
(2) An ensemble of multiple realizations of a Bayesian model provides more accurate inferences than a single deterministic model; see *e.g.* [2, 22].

In a nutshell, from the perspective of Bayesian approximation, the proposed BNN-based approach is about learning an approximation to the minimum mean square error (MMSE) estimate defined by

$$\widehat{\boldsymbol{x}} := \arg\min_{\boldsymbol{u}} \mathbb{E}_{(\boldsymbol{x}|\boldsymbol{y})}\|\boldsymbol{u}-\boldsymbol{x}\|_2^2 = \mathbb{E}_{(\boldsymbol{x}|\boldsymbol{y})}(\boldsymbol{x}|\boldsymbol{y}) = \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x} = \int \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0)p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}, \tag{7}$$

where $p(\boldsymbol{\theta}|\boldsymbol{y})$ is the posterior probability distribution function of $\boldsymbol{\theta}$. Considering the number of weights and the non-linear structure of the network, the computation of $p(\boldsymbol{\theta}|\boldsymbol{y})$ is intractable, and thus we use the joint distribution of independent normal distributions $q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})$ to approximate $p(\boldsymbol{\theta}|\boldsymbol{y})$:

$$q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma}): \quad \theta_i \sim \mathcal{N}(\mu_i,\sigma_i^2),\ \boldsymbol{\theta}=\{\theta_i\},\ \boldsymbol{\mu}=\{\mu_i\},\ \boldsymbol{\sigma}=\{\sigma_i\}. \tag{8}$$

The cost is defined by the KL divergence between $q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})$ and $p(\boldsymbol{\theta}|\boldsymbol{y})$:

$$(\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma}} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\|p(\boldsymbol{\theta}|\boldsymbol{y})). \tag{9}$$

Once the model is trained with learned distribution parameters $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^*$, we have a prediction that approximates the MMSE estimate:

$$\boldsymbol{x}^* = \int \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0)q(\boldsymbol{\theta}|\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*)d\boldsymbol{\theta}. \tag{10}$$

See Section 3 for a more detailed discussion.

1.2 Main Contributions

Built on BNN, this paper proposed a self-supervised learning method for image recovery and applied to solve image reconstruction in CS. The proposed method is dataset-free without requiring any external training sample. The experiments showed that the proposed approach not only significantly outperformed representative traditional non-learning methods, but also is very competitive to supervised deep learning methods with state-of-the-art performance. Indeed, the proposed method has its advantages when the measurement is noisy.

The results of this paper have significance in both theoretical research and practical applications. In the dataset-free setting, existing works showed that a DNN can effectively learn meaningful image structures from a noisy image by avoiding overfitting. However, for solving an ill-posed linear problem, the solution ambiguity requires new techniques to avoid more likely overfitting (with perturbations from null space), when training a DNN using only the measurement itself. This paper is the first work that showed the weight uncertainty induced by BNN is effective to handle overfitting, and learning ensemble can lead to accurate prediction. These results showed great potential of BNN in solving ill-posed linear inverse problems arising from imaging systems.

CS is one powerful sensing modality for designing imaging systems with faster sampling and lower energy consumption. It is about reconstructing signals/images, which are sparse in certain transform domain, using the measurements much less than that traditional uniform sampling (see *e.g.* [13, 7]). It has received great attention in a wide range of applications, including medical imaging [27, 16, 8] and computational photography [14, 1]. Image reconstruction is one key module in CS-based imaging systems. Existing supervised learning methods requires a large amount of training samples to provide state-of-the-art performance, which often is a challenging task in practice. For instance, it takes a long time to collect fully-sampled true images in MRI. The performance of the proposed self-supervised method is very competitive to state-of-the-art supervised learning method, while there is no any prerequisite on training samples. Such a dataset-free setting makes the proposed method very appealing in practice.

2 Related Work

As this paper is about deep learning for image recovery, we only give a very brief review on those non-learning methods and focus more on deep learning methods.

2.1 Regularization Methods With Pre-defined Image Prior

Regularization method is one widely-used technique for image recovery, which imposes certain image prior on the image to resolve solution ambiguities. In most regularization methods, the problem of image recovery is re-formulated to some optimization problem, and its minimizer is defined as the estimate of the truth. Many regularization methods have been proposed for solving various image recovery problems, including CS image reconstruction. The ℓ_1-norm relating regularization methods (*e.g.* [40, 6, 27, 23, 24]) assume the image gradients are sparse and use ℓ_1-norm relating regularizations for image recovery. The non-local methods exploit the recurrence prior of local image patches for image reconstruction. For instance, the low-rank regularization method [12] assumes that the stack of matched patches is of low rank; the BM3D-based regularization method [30, 10] employs the BM3D denoiser [9] for regularizing patch stacks. Non-local wavelet frame method [36] regularizes the image in non-local wavelet tight frame when recovering the image.

2.2 Supervised Deep Learning Methods

Recently, deep learning has became one powerful technique for solving ill-inverse problems in imaging. Most existing such solutions are based on supervised learning, *i.e.*, the DNN is trained on a dataset with the pairs of measurement/image. Earlier works take an end-to-end approach that learns the direct map between the measurement and the image, and the main difference among them is NN architecture, *e.g.* image recovery [37, 43] and CS image reconstruction [32, 21]. As imaging physics encoded in $\boldsymbol{A}$ is not utilized in an end-to-end approach, their performance is not significantly better than traditional regularization methods.

A more promising approach is the optimization unrolling with learnable prior. The idea is unrolling the iterative scheme of a regularization method (e.g. ℓ_1-norm relating method) and replacing the operations related to image prior by a

CNN. For image deconvolution, Meinhardt *et al.* [29] unrolled the primal-dual hybrid gradient method, Zhang *et al.* [46] unrolled a half-quadratic splitting method, and Nan *et al.* [33] unrolled a VEM-based iterative scheme. Proximal forward backward splitting scheme and Douglas-Rachford iteration are unrolled in [11] and [26] for medical image reconstruction. ADMM-Net [44] unrolled the alternating direction method of multipliers (ADMM) for MRI image reconstruction, and Liu *et al.* [26] proposed another scheme of the ADMM method with different variable splitting scheme. For CS image reconstruction, ISTA-Net [45] unrolled the iterative shrinkage-thresholding algorithm (ISTA). A scalable Laplacian pyramid reconstructive adversarial network (LAPRAN) [42] was proposed for CS image reconstruction which is adaptive to different CS ratios. In SCSNet [38], sensing and reconstruction of CS is integrated in one network.

2.3 Unsupervised Deep Learning Method for Image Denoising

There are few existing works on unsupervised deep learning methods for image recovery problems. Most existing unsupervised deep learning methods focus on image denoising, one sub-module often seen in image recovery. Based on the observation that regular image structures appear before random patterns during the training, Ulyanov *et al.* [41] proposed the deep image prior (DIP) method for image denoising by using early stopping to avoid overfitting. By simplifying a deep decoder, Heckel and Hand [19] proposed to use an under-parameterized NN for avoiding overfitting. Based on Stein's unbiased risk estimator (SURE), Soltanayeva and Chun [39] proposed a regularization on NN weights to train a denoising NN without training data. Quan *et al.* [35] proposed to use dropout in both training and testing for learning a denoising NN without training data.

Image denoising does not need to consider the ill-posedness of the matrix $\boldsymbol{A}$. Thus, these denoising methods cannot be easily generalized to solve ill-posed image recovery problems with good performance. Using the SURE denoiser, Zhussip *et al.* [47] developed a SURE-AMP method for CS image reconstruction based on the denoiser approximate message passing (AMP) framework. However, its performance is not competitive to other supervised learning methods.

3 Main Body

In this section, we give a detailed discussion on the proposed self-supervised BNN for CS image reconstruction. Let $\boldsymbol{y}$ denote the measurement, $\boldsymbol{x}$ denote the image to predict, $\boldsymbol{n}$ denote the measurement noise, and they are related by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}. \tag{11}$$

Let $\mathcal{F}_{\boldsymbol{\theta}}$ denote the BNN whose weights $\boldsymbol{\theta} = \{\theta_i\}$ are random variables. Consider a random initialization $\boldsymbol{\epsilon}_0$. As shown in (7), provided the posterior probability distribution function $p(\boldsymbol{\theta}|\boldsymbol{y})$, the MMSE estimate of the truth $\boldsymbol{x}$ is given by

$$\widehat{\boldsymbol{x}} = \int \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) p(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta}. \tag{12}$$

The amount of weights and complexity of a DNN makes $p(\boldsymbol{\theta}|\boldsymbol{y})$ computationally intractable. Thus, we take a Bayesian approximation approach to approximate $p(\boldsymbol{\theta}|\boldsymbol{y})$ by using the following independent joint normal distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$:

$$\theta_i \sim \mathcal{N}(\mu_i, \sigma_i), \tag{13}$$

where $\boldsymbol{\mu} = \{\mu_i\}, \boldsymbol{\sigma} = \{\sigma_i\}$, mean and s.t.d., are distribution parameters.

3.1 Training

The training of the BNN is done by minimizing the distance between the prediction of the NN and the MMSE estimate in (12). As we use $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ defined by (13) to approximate $p(\boldsymbol{\theta}|\boldsymbol{y})$ in the MMSE estimate, we proposed to train the BNN by minimizing the KL divergence between $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ and $p(\boldsymbol{\theta}|\boldsymbol{y})$:

$$\min_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \| p(\boldsymbol{\theta}|\boldsymbol{y})). \tag{14}$$

The minimum of the KL-divergence is difficult to find for general distributions. Thus, we further simplify the optimization problem by assuming that the prior distribution $p(\boldsymbol{\theta})$ can be well approximated by the joint distribution of i.i.d. normal distribution with zero mean and standard deviation $\overline{\sigma}$. Then, we have

Proposition 1. *Suppose that the measurement noise $\boldsymbol{n}$ is Gaussian white noise such that $p(\boldsymbol{n}) \sim \prod_i \exp(\frac{-\boldsymbol{n}_i^2}{2\tilde{\sigma}^2})$ and $p(\boldsymbol{\theta}) \sim \prod_i \exp(\frac{-\boldsymbol{\theta}_i^2}{2\overline{\sigma}^2})$. Then, we have*

$$\begin{aligned} &\min\nolimits_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \| p(\boldsymbol{\theta}|\boldsymbol{y})) \\ \Leftrightarrow &\min\nolimits_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2 + \lambda_1(\|\boldsymbol{\mu}\|_2^2 + \|\boldsymbol{\sigma}\|_2^2) - \lambda_2 \sum\nolimits_i \log \sigma_i, \end{aligned} \tag{15}$$

where $\lambda_1 = \tilde{\sigma}^2/\overline{\sigma}^2$ and $\lambda_2 = 2\tilde{\sigma}^2$.

Proof. See the supplementary for the detailed derivation.

For the data-dependent term $\mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2$ in (15), we only sample one instance of $\boldsymbol{\theta}$ from the distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ to approximate the expectation at each iteration for computational efficiency, which can be interpreted as a variation of the stochastic gradient descent (SGD) algorithm. It is noted that each σ_i denotes the standard deviation, which should be always positive. Thus, we adopt the same re-parameterization trick as [5] that re-expresses σ_i by

$$\sigma_i = \log(1 + \exp(\rho_i)). \tag{16}$$

Then, at every iteration of BNN training, we first randomly draw sample ϵ from standard normal distribution $\mathcal{N}(0, 1)$ and then generate the network weights by

$$\theta_i = \mu_i + \log(1 + \exp(\rho_i)) \cdot \epsilon. \tag{17}$$

More details on back-propagation for BNN can be found in the related materials.

3.2 Testing

Once the BNN is trained by (15) with estimated distribution parameters $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^*$, we have now an approximation to the posterior probability distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, *i.e.* $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$. The approximate MMSE estimate is then given by

$$\boldsymbol{x}^* = \int \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*) d\boldsymbol{\theta}. \tag{18}$$

Although both $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0)$ and $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$ have explicit forms, the above integration is still intractable. Instead, we use Monte Carlo (MC) integration in practice

$$\boldsymbol{x}^* \approx \frac{1}{T} \sum_{j=1}^{T} \mathcal{F}_{\boldsymbol{\theta}^j}(\boldsymbol{\epsilon}_0), \tag{19}$$

where $\{\boldsymbol{\theta}^j\}$ are the realizations of random variable $\boldsymbol{\theta}$ from the distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$ and T is the total sampling number.

3.3 Network Structure

We adopt the decoder part of a plain encoder-decoder NN as our NN with the following motivations. (a) Different from classic encoder-decoder network which takes images as inputs, our network input is a random vector which is very similar to the "code" generated by the encoder. Thus, the encoder does not see its function in our setting. (b) Owing to the need to learn both $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, the parameter number of the BNN is twice as that of its deterministic counterpart. Using only the decoder results in the reduction of model size, which is helpful for avoiding overfitting and reducing computational cost.

The NN architecture is illustrated in Figure 1. To recover an image of size $H \times W \times C$, the size of our NN input is $H/32 \times W/32 \times 128$. The input is forwarded into our decoder NN which contains five decoder blocks. Each of the first four decoder blocks sequentially connects an upsampling layer with a scaling factor of 2, and two Bayesian convolution layers both of which have 128 channels and are equipped with the leaky ReLU. The last decoder block contains an upsampling layer with a scaling factor of 2, and three aforementioned Bayesian convolution layers whose numbers of output channels are 64, 32, C with two LReLUs and one Sigmoid layer followed respectively. A Bayesian convolution layer is a convolution layer such that its weights are generated by (17) during training and testing.

4 Image Reconstruction in CS

In this section, we apply the proposed self-supervised image recovery method to solve image reconstruction problem in CS. Mathematically, CS image reconstruction can be formulated as solving an under-determined linear system:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{20}$$

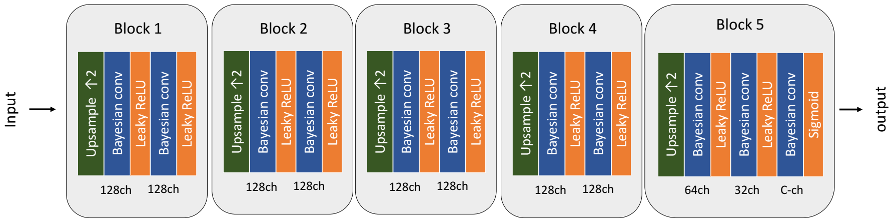


Fig. 1: Structure of the BNN used in the proposed method.

where $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ (or $\mathbb{C}^{M\times N}$) denotes the sensing matrix with $M \ll N$, $\boldsymbol{y}$ denotes the measurements collected by sensors, and $\boldsymbol{n}$ denotes noise. The experiments are conducted on two settings of CS: one is the block-wise random Gaussian CS problem in natural image acquisition, and the other is the random Fourier downsampling CS problem in magnetic resonance imaging (MRI).

4.1 Implementation Details

Our method is implemented using Pytorch. For convolution layers, the kernel size is 3×3 and both the stride and padding number is 1. The bi-linear interpolation is used for upsampling layers. For leaky ReLUs, the negative slope is fixed to 0.01. The BNN parameter $\boldsymbol{\mu}$ is initialized using the normal distribution as [18]. The initial value of $\vec{\rho}$ is drawn from the uniform distribution on $[-5, -4]$. The model is trained by the Adam optimizer with fixed learning rate 10^{-4}. The parameter λ_1 and λ_2 in (15) are updated as follows:

$$\lambda_1 = \gamma_1(\tilde{\sigma} + 10^{-3})^2,\ \ \lambda_2 = \gamma_2(\tilde{\sigma} + 10^{-3})^2, \tag{21}$$

with $\gamma_1 = 0.05$ and $\gamma_2 = 0.25$. The training procedure is stopped either the maximum iteration number 10^5 is reached or the residual $\|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2/M$ is less than $(\tilde{\sigma} + 10^{-3})^2$. The sampling number T used in the MC approximation during prediction is set to 100. For comparison to other methods, we cite the results directly from the literature if possible; otherwise, we run the codes from the authors with the effort on the tuning-up of parameters to reproduce. If none is available, we leave it blank in the table.

4.2 CS on Natural Image Acquisition

For the CS-based reconstruction on natural images, we follow the setting of one recent deep-learning-based method, *i.e.* ISTA-Net [45]. Two datasets are used for testing. One is "Set11" [45] with 11 images and the other is "BSD68" [28] with 68 images. These images are cropped into non-overlapped blocks of size 33×33 to generate the measurements. The sensing matrix $\boldsymbol{A}$ of size $M \times N$ ($N = 1089$) is first sampled from independent standard normal distribution entry-wisely and then orthogonalized row-wisely. The CS ratio, *i.e.* M/N, is set to $4\%, 10\%, 25\%, 40\%$ respectively. In the noisy case, Gaussian white noise with s.t.d. 10 are added to the measurements. See Table 1 for the computational times of our method. The time varies for different settings as our method stops the iteration when the residual meets tolerance.

Table 1: Computational time (in hours) of our method for processing images in Set11 and Set68 in different settings, on a TITAN RTX GPU.

Dataset	$\tilde{\sigma}$	40%	25%	10%	4%
Set11	0	4.7	4.9	5.1	5.8
	10	1.2	1.2	0.7	0.4

Dataset	$\tilde{\sigma}$	40%	25%	10%	4%
Set68	0	25.0	24.9	25.3	40.9
	10	11.3	10.7	7.3	5.3

Table 2: Average PSNR(dB)/SSIM results of different methods on Set11 [45] and BSD68 [28] in noiseless CS-based natural image reconstruction.

Dataset	Method	40%	25%	10%	4%
Set11	TVAL3	30.52/0.90	26.44/0.80	21.35/0.59	17.45/0.41
	DAMP	33.49/0.93	28.21/0.85	21.16/0.60	15.69/0.35
	ReconNet	-/-	25.54/0.76	22.68/0.64	19.98/0.53
	ISTA	**35.97/0.96**	**32.59/0.93**	26.64/0.81	21.59/0.62
	DIP	33.28/0.92	31.33/0.91	27.40/**0.83**	23.15/0.69
	Ours	35.71/0.95	32.30/0.92	**27.49/0.83**	**23.26/0.70**
BSD68	TVAL3	29.39/0.86	26.48/0.77	22.49/0.58	19.10/0.42
	DAMP	28.03/0.79	25.57/0.70	21.92/0.52	17.11/0.33
	ReconNet	-/-	25.31/0.71	23.16/0.60	21.28/0.50
	ISTA	**32.17/0.92**	**29.36/0.85**	**25.32**/0.70	22.40/0.56
	DIP	30.10/0.87	27.78/0.80	24.82/0.69	22.51/**0.58**
	Ours	31.28/0.90	28.63/0.84	25.24/**0.71**	**22.52/0.58**

Four methods are included in the comparison, *i.e.* TVAL3 [23], D-AMP [31], ReconNet [21] and ISTA [45]. The first two are regularization methods while the last two are supervised deep learning methods. In addition, we also include the DIP method [41], which is a recent unsupervised learning technique for image recovery. There is no work that directly extends the original DIP for CS image reconstruction. Thus, following the DIP for image super-resolution [41], we implement a DIP-based CS reconstruction method by using the cost function

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2. \tag{22}$$

The NN used for CS image reconstruction is the same as that for super-resolution, *i.e.* an encoder-decoder NN with skip-connections whose model size is comparable to ours. The NN stops if it hits the maximum iteration number 2×10^4 or

Table 3: Average PSNR(dB)/SSIM results of different methods on Set11 [45] and BSD68 [28] in CS-based natural image reconstruction with noise level $\tilde{\sigma} = 10$.

Dataset	Method	40%	25%	10%	4%
Set11	TVAL3	26.66/0.72	24.75/0.67	21.02/0.54	17.28/0.39
	DAMP	29.25/0.86	26.35/0.80	20.84/0.58	15.56/0.35
	ReconNet	-/-	24.36/0.66	22.00/0.57	19.62/0.49
	ISTA	27.98/0.75	27.26/0.75	24.55/0.70	20.79/0.56
	DIP	28.87/0.83	27.36/0.79	24.19/0.68	21.27/0.55
	Ours	**30.39/0.88**	**28.67/0.84**	**25.23/0.76**	**21.91/0.64**
BSD68	TVAL3	26.15/0.68	24.80/0.63	22.03/0.52	18.93/0.39
	DAMP	26.55/0.72	24.87/0.65	21.70/0.51	16.96/0.33
	ReconNet	-/-	24.12/0.61	22.36/0.53	20.77/0.46
	ISTA	26.68/0.70	25.84/0.68	**23.86**/0.60	**21.64**/0.50
	DIP	25.24/0.64	24.07/0.59	22.46/0.51	21.13/0.45
	Ours	**28.13/0.81**	**26.47/0.75**	23.79/**0.64**	21.54/**0.53**

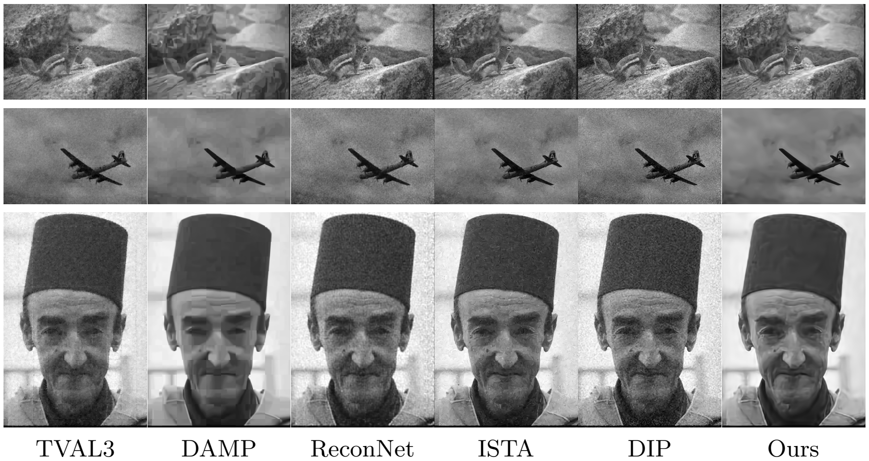


Fig. 2: Results of Gaussian CS image reconstruction using noisy input with ratio 25%.

the residual $\|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2/M$ reaches the same tolerance $(\tilde{\sigma} + 10^{-3})^2$ as ours.

See Table 2 and Table 3 for the quantitative results in both noiseless and noisy cases. See Figure 2 for visual comparison of some examples. Generally, our method outperformed two traditional non-learning methods (TVAL3 and DAMP) by a large margin and the unsupervised deep-learning method DIP. Even compared to the state-of-the-art (SOTA) supervised learning methods, our method remains very competitive. The SOTA supervised learning methods (*e.g.* ISTA) have small advantages when the measurements are noise-free and ours has noticeable advantages when the measurements are noisy.

Additionally, we compared the behavior of the BNN method to the DIP method over iterations. We run 10^5 steps for both without early stopping on a sample image. See Figure 3 for the trace of PSNR and residual value over iteration. It can be seen that our method is more stable to the iteration number than the DIP method in terms of PSNR value. Moreover, the residual of DIP decreases to zero eventually even in the presence of noise, which causes overfitting, while the residual of our method does not vanish. This indicates the advantages of our BNN over the DIP method when processing noisy measurements.

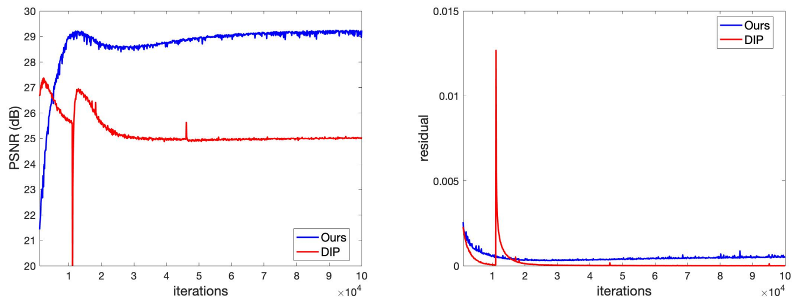


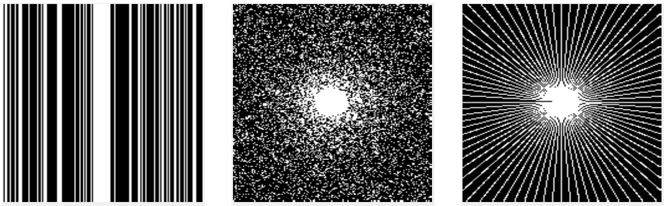
Fig. 3: PSNR (left) and residual(right) over iterations of DIP and our method with CS ratio 25% and noise level $\tilde{\sigma} = 10$ on the natural image "boats".

4.3 CS Image Reconstruction in MRI

For CS in MRI, we use the down-sampled data in the k-space. The sensing matrix $\boldsymbol{A}$ is the dot production of a random down-sampling mask $\boldsymbol{M}$ and the discrete Fourier transform $\boldsymbol{F}$. Following the setting in [26], the contaminated measurements are generated by $\boldsymbol{y} = \boldsymbol{M} \odot \boldsymbol{F}(\boldsymbol{x} + \boldsymbol{n}_1 + i\boldsymbol{n}_2)$, where the entries of $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ follow i.i.d. normal distribution of mean zero and s.t.d. $\tilde{\sigma}$. Then the noise $\boldsymbol{n}$ in (20) takes the form of $\boldsymbol{n} = \boldsymbol{M} \odot \boldsymbol{F}(\boldsymbol{n}_1 + i\boldsymbol{n}_2)$, which is complex and also follows i.i.d. Gaussian distribution entry-wisely. The dataset is the same as [26] with 21 MRI images from ADNI (Alzheimer's Disease Neuroimaging Initiative). We test three types of down-sampling masks of sampling ratio 25%, namely, 1D Gaussian mask, 2D Gaussian mask, and radial mask shown in Figure 4. In the noisy case, Gaussian white noise with s.t.d. $\tilde{\sigma}$ as 10% of the maximum pixel value of the MRI image are added to the down-sampled k-space measurements. We compare the performance of our method with the simple zero-filling method (ZF) [4], TV-regularization-based method [27], ADMM-Net [44], the plug-in methods in [26] with three different networks: SCAE, SNLAE, and GAN, and DIP [41]. See Table 4 for the quantitative comparison of different method and Figure 5 for visual comparison on some sample images. It can be seen that our method outperformed all the other methods in all settings, except that DIP performs best in the noiseless case with 1D Gaussian mask.

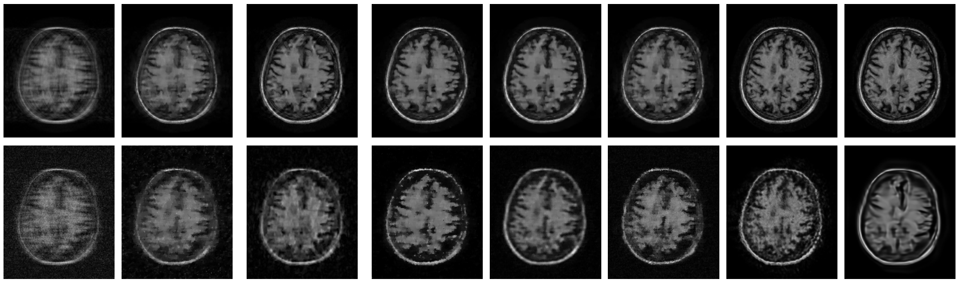
Table 4: Average PSNR(dB)/SSIM of the results of of different methods for CS-based MRI reconstruction.

Method	1D Gaussian		2D Gaussian		radial	
$\tilde{\sigma}$	0	10%	0	10%	0	10%
ZF	23.06/0.62	20.37/0.26	25.30/0.50	22.38/0.36	25.45/0.51	22.38/0.36
TV	25.77/0.76	22.25/0.37	32.79/0.90	24.92/0.49	32.32/0.90	25.16/0.49
ADMM-Net	28.99/0.87	22.98/0.44	34.97/0.94	25.84/0.60	33.67/0.93	25.96/0.61
SCAE	29.37/0.88	22.72/0.63	35.61/0.95	26.06/0.74	33.94/0.94	26.13/0.70
SNLAE	29.06/0.86	24.39/0.56	32.85/0.86	26.15/0.67	32.53/0.88	26.38/0.66
GAN	27.47/0.82	23.32/0.69	32.94/0.91	26.31/0.75	32.26/0.90	25.53/0.74
DIP	**31.80/0.92**	23.38/0.68	35.63/0.95	24.41/0.72	33.81/0.94	24.54/0.73
Ours	31.38/0.91	**25.65/0.76**	**36.10/0.96**	**27.12/0.82**	**34.08/0.95**	**27.07/0.82**



1D Gaussian 2D Gaussian radial

Fig. 4: Three different types of sampling masks of sample ratio 25%.



ZF TV ADMM-net SCAE SNLAE GAN DIP Ours

Fig. 5: MRI reconstruction results with 1D Gaussian mask of sampling ratio 25%; the first row corresponds to the noiseless case and the second row noisy case.

4.4 Ablation Study

Ablation study is conducted on CS reconstruction for image acquisition on the dataset Set11 to show how much performance improvement weight uncertainty of BNN can bring in. Two deterministic versions of the BNN are used for comparison. One is the MLE estimator which trains the NN with deterministic weights: $\min_{\boldsymbol{\theta}} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2$. The other is the MAP estimator which trains the NN with deterministic weights using (5) and a Gaussian prior on the weights:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2 - 2\tilde{\sigma}^2 \log(p(\boldsymbol{\theta})) = \|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0) - \boldsymbol{y}\|_2^2 + \gamma\tilde{\sigma}^2\|\boldsymbol{\theta}\|_2^2, \tag{23}$$

where γ is set to 0.05 after tuning-up and $\tilde{\sigma}^2$ replaced with a small perturbation $(\tilde{\sigma} + 10^{-3})^2$ as ours. All these two versions and ours use the same architecture and stopping criteria. See Table 5 for the comparison. Clearly, the BNN with random weights significantly outperformed the other two deterministic versions. This clearly indicates the effectiveness of weight uncertainty in BNN on handling the overfitting in our self-supervised learning methods for CS reconstruction.

Table 5: Average PSNR(dB)/SSIM results of ablation studies on natural image Set11.

$\tilde{\sigma}$	Method	Weights	40%	25%	10%	4%
0	MLE	Deterministic	32.34/0.92	29.43/0.87	25.13/0.75	21.13/0.61
	MAP	Deterministic	32.90/0.92	29.51/0.87	25.01/0.74	21.08/0.60
	Ours	Random	**35.71/0.95**	**32.30/0.92**	**27.49/0.83**	**23.26/0.70**
10	MLE	Deterministic	28.87/0.84	27.24/0.80	23.92/0.70	20.35/0.56
	MAP	Deterministic	28.82/0.84	27.18/0.80	23.90/0.70	20.25/0.56
	Ours	Random	**30.39/0.88**	**28.67/0.84**	**25.23/0.76**	**21.91/0.64**

5 Conclusion

Built Bayesian neural network with random weights, this paper proposed a self-supervised framework of deep learning with state-of-the-art performance for reconstructing an image from fewer and noisy measurements in CS. The work in this paper not only has its value in the applications of CS-based imaging systems, but also provides a new insight for developing dataset-free un-supervised/self-supervised deep learning methods for other image recovery problems.

Acknowledgment

Tongyao Pang and Hui Ji would like to acknowledge the support from Singapore MOE Academic Research Fund (AcRF) Tier 2 research project (MOE2017-T2-2-156), and Yuhui Quan would like to acknowledge the support of National Natural Science Foundation of China (61872151, U1611461).

References

1. Arce, G., Brady, D., Carin, L., Arguello, H., Kittle, D.: Compressive coded aperture spectral imaging: An introduction. IEEE Signal Processing Magazine **31**(1), 105–115 (2013)
2. Baldi, P., Sadowski, P.J.: Understanding dropout. In: NeurIPS. pp. 2814–2822 (2013)
3. Barber, D., Bishop, C.M.: Ensemble learning in bayesian neural networks. Nato ASI Series F Computer and Systems Sciences **168**, 215–238 (1998)
4. Bernstein, M.A., Fain, S.B., Riederer, S.J.: Effect of windowing and zero-filled reconstruction of MRI data on spatial resolution and acquisition strategy. Journal of Magnetic Resonance Imaging **14**(3), 270–280 (2001)
5. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: ICML. pp. 1613–1622 (2015)
6. Cai, J., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring from a single image using sparse approximation. In: CVPR. pp. 104–111 (2009)
7. Candes, E.J., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Transactions on Information Theory **52**(12), 5406–5425 (2006)
8. Chen, G., Tang, J., Leng, S.: Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. Medical Physics **35**(2), 660–663 (2008)
9. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
10. Danielyan, A., Katkovnik, V., Egiazarian, K.: Bm3d frames and variational image deblurring. IEEE Transactions on Image Processing **21**(4), 1715–1728 (2011)
11. Ding, Q., Chen, G., Zhang, X., Huang, Q., Ji, H., Gao, H.: Low-dose CT with deep learning regularization via proximal forward backward splitting. Physics in Medicine & Biology (2020)
12. Dong, W., Shi, G., Li, X., Ma, Y., Huang, F.: Compressive sensing via nonlocal low-rank regularization. IEEE Transactions on Image Processing **23**(8), 3618–3632 (2014)
13. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)
14. Duarte, M., Davenport, M., Takhar, D., Laska, J., Sun, T., Kelly, K., Baraniuk, R.: Single-pixel imaging via compressive sampling. IEEE Signal Processing Magazine **25**(2), 83–91 (2008)
15. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML. pp. 1050–1059 (2016)
16. Gamper, U., Boesiger, P., Kozerke, S.: Compressed sensing in dynamic mri. Magnetic Resonance in Medicine **59**(2), 365–373 (2008)
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
18. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. pp. 1026–1034 (2015)
19. Heckel, R., Hand, P.: Deep decoder: Concise image representations from untrained non-convolutional networks. arXiv preprint arXiv:1810.03982 (2018)
20. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: NeurIPS. pp. 5574–5584 (2017)

21. Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., Ashok, A.: Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In: CVPR. pp. 449–458 (2016)
22. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS. pp. 6402–6413 (2017)
23. Li, C., Yin, W., Jiang, H., Zhang, Y.: An efficient augmented lagrangian method with applications to total variation minimization. Computational Optimization and Applications **56**(3), 507–530 (2013)
24. Li, M., Fan, Z., Ji, H., Shen, Z.: Wavelet frame based algorithm for 3d reconstruction in electron microscopy. SIAM Journal on Scientific Computing **36**(1), B45–B69 (2014)
25. Liu, J., Chen, N., Ji, H.: Learnable douglas-rachford iteration and its applications in dot imaging. Inverse Problems & Imaging **14**(4), 683 (2020)
26. Liu, J., Kuang, T., Zhang, X.: Image reconstruction by splitting deep learning regularization from iterative inversion. In: MICCAI. pp. 224–231. Springer (2018)
27. Lustig, M., Donoho, D., Pauly, J.M.: Sparse MRI: The application of compressed sensing for rapid mr imaging. Magnetic Resonance in Medicine **58**(6), 1182–1195 (2007)
28. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. vol. 2, pp. 416–423. IEEE (2001)
29. Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: ICCV. pp. 1781–1790 (2017)
30. Metzler, C.A., Maleki, A., Baraniuk, R.: Bm3d-amp: A new image recovery algorithm based on bm3d denoising. In: ICIP. pp. 3116–3120. IEEE (2015)
31. Metzler, C.A., Maleki, A., Baraniuk, R.: From denoising to compressed sensing. IEEE Transactions on Information Theory **62**(9), 5117–5144 (2016)
32. Mousavi, A., Patel, A., Baraniuk, R.: A deep learning approach to structured signal recovery. In: Allerton. pp. 1336–1343. IEEE (2015)
33. Nan, Y., Quan, Y., Ji, H.: Variational-EM-based deep learning for noise-blind image deblurring. In: CVPR. pp. 3626–3635 (June 2020)
34. Nan, Y., Ji, H.: Deep learning for handling kernel/model uncertainty in image deconvolution. In: CVPR. pp. 2388–2397 (June 2020)
35. Quan, Y., Chen, M., Pang, T., Ji, H.: Self2self with dropout: Learning self-supervised denoising from single image. In: CVPR. pp. 1890–1898 (2020)
36. Quan, Y., Ji, H., Shen, Z.: Data-driven multi-scale non-local wavelet frame construction and image recovery. Journal of Scientific Computing **63**(2), 307–329 (2015)
37. Schuler, C., B., C., Harmeling, S., Scholkopf, B.: A machine learning approach for non-blind image deconvolution. In: CVPR. pp. 1067–1074 (2013)
38. Shi, W., Jiang, F., Liu, S., Zhao, D.: Scalable convolutional neural network for image compressed sensing. In: CVPR. pp. 12290–12299 (2019)
39. Soltanayev, S., Chun, S.: Training deep learning based denoisers without ground truth data. In: NeurIPS. pp. 3257–3267 (2018)
40. Tang, J., Nett, B.E., Chen, G.: Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms. Physics in Medicine & Biology **54**(19), 5781 (2009)
41. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR. pp. 9446–9454 (2018)

42. Xu, K., Zhang, Z., Ren, F.: Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction. In: ECCV. pp. 485–500 (2018)
43. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NIPS. pp. 1790–1798 (2014)
44. Yang, Y., Sun, J., Li, H., Xu, Z.: Deep admm-net for compressive sensing MRI. In: NeurIPS. pp. 10–18 (2016)
45. Zhang, J., Ghanem, B.: Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In: CVPR. pp. 1828–1837 (2018)
46. Zhang, J., Pan, J., Lai, W.S., Lau, R.W., Yang, M.H.: Learning fully convolutional networks for iterative non-blind deconvolution. In: CVPR. pp. 3817–3825 (2017)
47. Zhussip, M., Soltanayev, S., Chun, S.: Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior. In: CVPR. pp. 10255–10264 (2019)

Self-supervised Bayesian Deep Learning for Image Recovery with Applications to Compressive Sensing (Supplementary Materials)

Tongyao Pang[1], Yuhui Quan[2], and Hui Ji[1]

[1] Department of Mathematics, National University of Singapore, 119076, Singapore
[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
matpt@nus.edu.sg, csyhquan@scut.edu.cn, matjh@nus.edu.sg

1 Proof of Proposition 1

The KL divergence between $q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})$ and $p(\boldsymbol{\theta}|\boldsymbol{y})$ can be rewritten as

$$\begin{aligned}&\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\|p(\boldsymbol{\theta}|\boldsymbol{y}))\\=&\,\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\|p(\boldsymbol{\theta}))-\mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})}\log p(\boldsymbol{y}|\boldsymbol{\theta})+\mathrm{const.}\end{aligned}\tag{1}$$

Since $p(\boldsymbol{\theta})\sim\prod_i\exp(\frac{-\theta_i^2}{2\overline{\sigma}^2})$ and $q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\sim\prod_i\exp(\frac{-(\theta_i-\mu_i)^2}{2\tilde{\sigma}_i^2})$, we have

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\|p(\boldsymbol{\theta}))=&\sum_i\mathrm{KL}(q(\theta_i|\mu_i,\sigma_i)\|p(\theta_i))\\=&\frac{1}{2\overline{\sigma}^2}(\|\boldsymbol{\mu}\|_2^2+\|\boldsymbol{\sigma}\|_2^2)-\sum_i\log\sigma_i+\mathrm{const.}\end{aligned}\tag{2}$$

On the other hand, $p(\boldsymbol{n})\sim\prod_i\exp(\frac{-\boldsymbol{n}_i^2}{2\tilde{\sigma}^2})$, which gives us

$$\log p(\boldsymbol{y}|\boldsymbol{\theta})=-\frac{1}{2\tilde{\sigma}}\|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0)-\boldsymbol{y}\|_2^2+\mathrm{const.}\tag{3}$$

Finally, we obtain

$$\begin{aligned}&\min_{\boldsymbol{\mu},\boldsymbol{\sigma}}\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})\|p(\boldsymbol{\theta}|\boldsymbol{y}))\\=&\min_{\boldsymbol{\mu},\boldsymbol{\sigma}}\mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})}\|\boldsymbol{A}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_0)-\boldsymbol{y}\|_2^2+\lambda_1(\|\boldsymbol{\mu}\|_2^2+\|\boldsymbol{\sigma}\|_2^2)-\lambda_2\sum_i\log\sigma_i,\end{aligned}\tag{4}$$

where $\lambda_1=\tilde{\sigma}^2/\overline{\sigma}^2$ and $\lambda_2=2\tilde{\sigma}^2$. The proof is done.

2 More Ablation Studies

In the main paper, we have conducted ablation studies to demonstrate the advantages of our BNN over deterministic NNs. Now we want to further show the effectiveness of our Monte Carlo (MC) prediction scheme

$$\boldsymbol{x}^* \approx \frac{1}{T}\sum_{j=1}^{T} \mathcal{F}_{\boldsymbol{\theta}^j}(\boldsymbol{\epsilon}_0), \tag{5}$$

where $\{\boldsymbol{\theta}^j\}_j = 1^T$ are the realizations of random variable $\boldsymbol{\theta}$ from the distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$ and T is the total sampling number. As a comparison, we test the performance of the single prediction scheme, which only uses the mean of the weights, *i.e.* $\boldsymbol{\mu}^*$, to predict as follows

$$\tilde{\boldsymbol{x}} = \mathcal{F}_{\boldsymbol{\mu}^*}(\boldsymbol{\epsilon}_0). \tag{6}$$

See Table 1 for the quantitative results on Set11 [1] in CS reconstruction of natural images. It can be seen that in noise-free case, there is no performance gain of our MC prediction (5) over the single prediction scheme (6). In contrast, in noisy case, our MC prediction significantly outperformed the single one. This phenomenon may be explained by the weight uncertainty of the trained BNN model.

Recall that weight uncertainty is measured by the variance $\boldsymbol{\sigma}^*$ and the signal-to-noise ratio $\boldsymbol{\mu}^*/\boldsymbol{\sigma}^*$ in Figure 1. It can be seen that the weight uncertainty is of large magnitude in the noisy case such that multiple predictions via MC sampling of the weights are more diverse and averaging them provides more gains in performance. For a better understanding, we select a 10×10 block from the natural image "Lena256" to visualize the diversity of the predictions via MC sampling of the weights in Figure 2. The x-axis stands for the pixel at the selected 10×10 block, which varies from 1 to 100. The y-axis is the corresponding pixel value. We plot the mean and variance of the predictions $\{\mathcal{F}_{\boldsymbol{\theta}^j}(\boldsymbol{\epsilon}_0)\}_{j=1}^{100}$, where the weights $\boldsymbol{\theta}^j$ are sampled from $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$. The mean is the central blue line and the variance is reflected by the shallow blue area. In the noiseless case, the shallow blue area even can not be observed which means that the multiple predictions via MC sampling of the weights are the same. This explains the finding that there is no difference in the performance of the single prediction (6) and our MC prediction in the noiseless case.

Table 1. Average PSNR(db)/SSIM results of ablation studies on Set11 [1].

$\tilde{\sigma}$	prediction	40%	25%	10%	4%
0	single	**35.71/0.95**	**32.30/0.92**	**27.49/0.83**	**23.26/0.70**
	ours	**35.71/0.95**	**32.30/0.92**	**27.49/0.83**	**23.26/0.70**
10	single	29.50/0.86	27.87/0.82	24.54/0.73	21.36/0.62
	ours	**30.39/0.88**	**28.67/0.84**	**25.23/0.76**	**21.91/0.64**

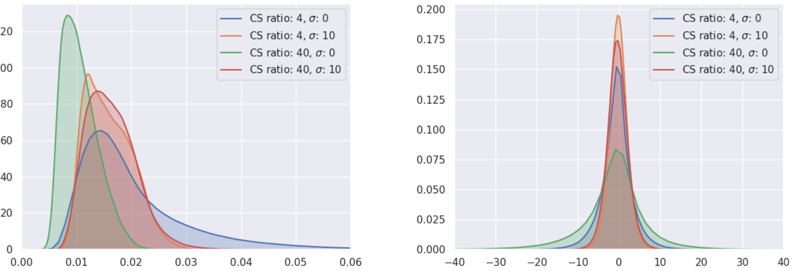
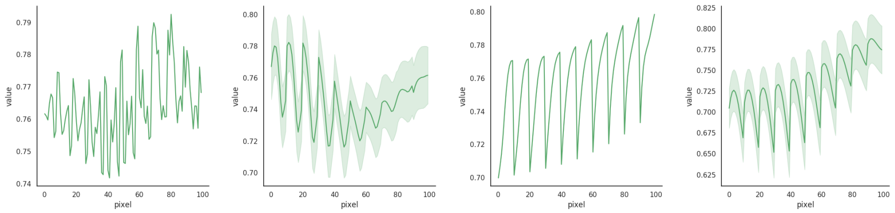


Fig. 1. Histograms of the variance $\boldsymbol{\sigma}^*$ (left) and signal-to-noise ratio $\boldsymbol{\mu}^*/\boldsymbol{\sigma}^*$ (right) of the weights of the trained BNN for natural image "boats" with different CS ratios and noise levels.



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

Fig. 2. The diversity of the predictions $\{\mathcal{F}_{\boldsymbol{\theta}^j}(\boldsymbol{\epsilon}_0)\}_{j=1}^{100}$ for a 10×10 block of the natural image "boats", where the weights $\boldsymbol{\theta}^j$ are sampled from $q(\boldsymbol{\theta}|\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$. The central blue lines are the mean of the predictions over 100 times and the shallow blue areas indicate the variance. From left to right, the settings are: (a) CS ratio = 40, $\sigma = 0$; (b) CS ratio = 40, $\sigma = 10$;(c) CS ratio = 4, $\sigma = 0$;(d) CS ratio = 4, $\sigma = 10$.

3 An Interesting Demo on Batch Processing

In the previous experiments, we only process one single image once. It is attractive to see whether our method is able to process multiple images in a batch during one period of training. In this section, we show a demo on MRI data for batch image processing. The maximum iteration for batch processing is increased to 1.5×10^5. See Table 2 for the comparison of batch image processing and separate processing on MRI data in compressive sensing. In the noisy case, the results of the batch training are even better than that of the separate training.

Table 2. Comparison of PSNR(db)/SSIM results of separate training and batch training on MRI data in compressive sensing.

mask	1D Gaussian		2D Gaussian		radial	
σ	0	10%	0	10%	0	10%
Separate	31.38/**0.91**	25.65/0.76	**36.10/0.96**	27.12/0.82	**34.08/0.95**	27.07/0.82
Batch	**31.44/0.91**	**25.99/0.80**	34.78/0.94	**27.41/0.84**	33.37/0.94	**27.36/0.84**

References

1. Zhang, J., Ghanem, B.: Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In: CVPR. pp. 1828–1837 (2018)