Self-supervised Deep Learning for Image Reconstruction: A Langevin Monte Carlo Approach*

Ji Li†, Weixi Wang‡, and Hui Ji§

Abstract. Deep learning has proved to be a powerful tool for solving inverse problems in imaging, and most of the related work is based on supervised learning. In many applications, collecting truth images is a challenging and costly task, and the prerequisite of having a training dataset of truth images limits its applicability. This paper proposes a self-supervised deep learning method for solving inverse imaging problems that does not require any training samples. The proposed approach is built on a re-parametrization of latent images using a convolutional neural network (CNN), and the reconstruction is motivated by approximating the minimum mean square error (MMSE) estimate of the latent image using a Langevin-dynamics-based Monte Carlo (MC) method. To efficiently sample the network weights in the context of image reconstruction, we propose a Langevin MC scheme called Adam-LD, inspired by the well-known optimizer in deep learning, Adam. The proposed method is applied to solve linear and nonlinear inverse problems, specifically, sparse view CT image reconstruction and phase retrieval. Our experiments demonstrate that the proposed method outperforms existing unsupervised or self-supervised solutions in terms of reconstruction quality.

Key words. Self-supervised learning, Inverse problems, Image reconstruction, Langevin dynamics, Bayesian inference

AMS subject classifications. 68U10, 94A08

1. Introduction.

An inverse problem in imaging concerns the estimation of an image $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$ related by

$$\boldsymbol{y} = \Phi(\boldsymbol{x}) + \boldsymbol{n}, \tag{1.1}$$

where Φ denotes the forward model of image acquisition and $\boldsymbol{n}$ denotes measurement noise. Inverse imaging problem is an important problem with a wide range of applications in practice. For example, compressed sensing (CS) [32], computed tomography (CT) [18] and magnetic resonance imaging (MRI) [54] in medicine, photography restoration in optics [8], and phase retrieval in crystallography and optics [75]. All these inverse problems can be expressed in the form of (1.1) but with different definitions of Φ. In most cases, the problem (1.1) is either ill-posed with many solutions, or ill-conditioned such that a direct inversion of $\Phi(\cdot)$ will magnify measurement noise in the solution.

To address the ill-posedness of an inverse problem, one widely-used practice is to impose certain prior on the solution, the latent image, when solving (1.1). By expressing the problem as an optimization problem, such an image prior is introduced as a regularization term in the objective function. In the last few decades, many image priors have been proposed for

*Submitted to the editors August 18, 2023.

Funding: This work was funded by Singapore MOE AcRF Tier 1 Research Grant R146000315114.

†Department of Mathematics, National University of Singapore, Singapore (matliji@nus.edu.sg). The author is now affiliated to Academy for Multidisciplinary Studies of Capital Normal University, Beijing.

‡Department of Mathematics, National University of Singapore, Singapore (wangweixi@u.nus.edu).

§Department of Mathematics, National University of Singapore, Singapore (matjh@nus.edu.sg).

solving various inverse imaging problems. The most prominent one is the ℓ_1-norm relating regularization which prompts the sparsity of certain measurement of the solution, *e.g.* TV regularization and its variations [72, 19, 20], and wavelet-based regularization [15, 31].

1.1. Deep learning and inverse imaging problems. In recent years, deep learning has become a powerful tool for image recovery with very promising performance. Most existing studies on deep learning for image recovery focus on supervised learning, *i.e.*, a deep neural network (DNN), denoted by $f(\cdot;\theta)$, is trained over a dataset $\Omega = \{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^N$ where $\boldsymbol{x}_i$ denotes a latent image and $\boldsymbol{y}_i$ its noisy partial measurement. The network weights are often determined by minimizing the prediction error of the network over the dataset:

$$\theta^* = \mathrm{argmin}_\theta \frac{1}{N}\sum_{i=1}^{N} \|f(\boldsymbol{y}_i;\theta) - \boldsymbol{x}_i\|_2^2. \tag{1.2}$$

It can be seen that the supervised deep learning is about constructing a DNN-based estimator that approximates the so-called minimum mean square error (MMSE) estimator:

$$\boldsymbol{x}_{\mathrm{MMSE}} = f(\boldsymbol{y};\theta^*) = \min_{\boldsymbol{u}} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y}} \|\boldsymbol{u} - \boldsymbol{x}\|_2^2. \tag{1.3}$$

It is known that the success of a supervised learning method depends on the quality and quantity of training samples. For image recovery, in order to generalize well on unseen test data, a supervised learning method requires a large set of training samples which are highly related to test data for recovery. Such a prerequisite on training samples limits the applicability of supervised learning methods in many data-limited applications, specially the ones in medicine and science. For medical imaging, accessing a large amount of high-quality real-world medical images of the patients is often not possible. For scientific imaging, the image for reconstruction might contain important structures which do not exist in existing images.

One popular approach to relax the prerequisite on training samples is introducing some pre-trained DNN model to regularize image reconstruction. As a regularization method often is expressed as an iterative scheme, the so-called Plug-and-Play (PnP) method [88, 70, 59, 73, 41] replaces the operation related to the pre-defined image prior in such an iterative scheme by a pre-trained denoising network. Although the PnP method can be conveniently called in practice, its performance depends on how well the called pre-trained denoising network generalizes on test data. In other words, real-world images of high quality related to test images are still required for pre-training a denoising network.

Generative adversarial network (GAN) is another type of network whose pre-trained model can be used for regularizing image reconstruction. A GAN model is a generative model which generates an image of interest from an random initial seed. Then, for image reconstruction, one can re-formulate the estimation of the latent image as the estimation of the initial seed whose corresponding output image fits the measurement; see *e.g.* [12, 5, 6]. One issue of GANs is that they often suffer from training instability and model collapse, where the generator tends to generate a limited variety of samples [74, 4, 21]. Another issue is that a pre-trained GAN model usually is domain-specific, *e.g.* face images or text images. It remains an open problem to have a universal pre-trained GAN model that works well for all types of images. Indeed, to

the best of our knowledge, there is no a pre-trained GAN model that can faithfully generate images in medicine or science. where it is challenging to collect a large number of images for training a GAN.

Recently, there is an increasing interest on developing unsupervised or self-supervised deep learning methods for image recovery/reconstruction, which does not require any training sample or a pre-trained network model. One pioneering work is the so-called deep image prior (DIP) [86], which shows that the architecture of an untrained CNN has an implicit regularization effort on the output, *i.e.*, when training a CNN using gradient descent, the network will prefer the output with regular patterns over the output with random noise in the earlier stages. Thus, by using early stopping, one can train a CNN by fitting its output to the noisy image. While the DIP provides certain regularization, for ill-posed inverse problems, it still suffers from the over-fitting caused by the absence of ground-truth images in the loss function. Built on the DIP, there are some works on unsupervised deep learning for addressing the over-fitting issue from different perspectives; See *e.g.* [40, 53, 77, 28].

1.2. Our approach. This paper aims at studying self-supervised deep learning for solving general inverse imaging problems. The problem setting considered in this paper is as follows.

(i) There is one testing data $\boldsymbol{y}$ to be processed.
(ii) There is no any external image to provide prior information of the test data.
(iii) There is no any pre-trained model to use.

The goal is to develop a self-supervised deep learning method for reconstructing the latent image $\boldsymbol{x}$ from the testing measurement $\boldsymbol{y}$.

Like most DIP-relating self-supervised methods, we also consider using an untrained CNN, denoted by $f(\epsilon_0;\theta)$, to re-parameterize the latent image $\boldsymbol{x}$:

$$\boldsymbol{x} = f(\epsilon_0;\theta), \tag{1.4}$$

where ϵ_0 has the same height and width as the unknown $\boldsymbol{x}$, and is filled with random noise uniformly sampled in the range of 0 to 0.1, as described in [86]. Such a re-parametrization introduces the implicit regularization brought by the DIP. Same as [65], our approach is also about using an untrained CNN to approximate the MMSE estimate of the solution. That is, by re-expressing the MMSE estimate by the conditional expectation:

$$\boldsymbol{x}_{\mathrm{CE}} = \int \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}, \tag{1.5}$$

where $p(\boldsymbol{x}|\boldsymbol{y})$ denotes the conditional probability of $\boldsymbol{x}$ given $\boldsymbol{y}$. Under certain condition on $p(\boldsymbol{x})$, we have then

$$\boldsymbol{x}_{\mathrm{CE}} = \int f(\epsilon_0;\theta) p(\theta|\boldsymbol{y},\epsilon_0) d\theta. \tag{1.6}$$

In general, the conditional probability function $p(\theta|\boldsymbol{y},\epsilon_0)$ is computationally intractable. Then, the remaining question is how to calculate the integral above. In [65], a variational approximation method is proposed which approximates $p(\theta|\boldsymbol{y},\epsilon_0)$ by a class of computational tractable Gaussian distributions. Instead of using variational approximation in the calculation of the

integral, we propose to directly calculate the integral using the MC sampling method:

$$\widehat{\boldsymbol{x}} = \frac{1}{K}\sum_{k=1}^{K} f(\epsilon_0;\theta_k), \quad \theta_k \sim p(\theta|\boldsymbol{y},\epsilon_0). \tag{1.7}$$

Remark 1.1. The DNN-based re-parametrization of an image is not a generative model. The DNN-based re-parametrization $\boldsymbol{x} = f(\epsilon;\theta)$ learns a mapping between network weights θ and the image $\boldsymbol{x}$ for some random initial ϵ_0: $\theta \xrightarrow{f(\cdot,\epsilon_0)} \boldsymbol{x}$. Each image has its own network weights. In contrast, a generative model $\boldsymbol{x} = g(\epsilon;\theta^*)$ learns a mapping between the initial seed ϵ and the target image: $\epsilon \xrightarrow{f(\cdot,\theta^*)} \boldsymbol{x}$. All images correspond to the same network weights.

For sufficient expression capability, an over-parameterized network $f(\epsilon_0;\theta)$ is used for representing the image. Then, the success of such a MC method depends on whether we can efficiently sample the posterior distribution $p(\theta|\boldsymbol{y},\epsilon_0)$ in a very high dimensional space. Motivated by the effectiveness of the Langevin MC method, a Markov Chain Monte Carlo (MCMC) sampler, in the context of Bayesian inference and neural network training, this paper studies the Langevin MC method for efficiently sampling the posterior distribution $p(\theta|\boldsymbol{y},\epsilon_0)$ of network weights. Such a Langevin MC method will then be used for approximating the MMSE estimate of the image $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$. In the context of solving inverse problems, one viable technique [38] is adopting conditional stochastic normalizing flow, which transforms a simple base distribution into the desired target distribution using a trainable network. This transformation facilitates efficient sampling from the target distribution, often achieved through methods like Markov Chain samplers. However, to keep the implementation simplicity, this paper chooses not to employ this technique for facilitating the sampling process.

The plain over-damped Langevin Monte Carlo (MC) method [89, 68, 69, 33] is designed for general Monte Carlo sampling, which is closely related to the stochastic gradient descent (SGD) method used to train deep neural networks (DNNs). In the context of deep learning, by exploiting the connection between sampling and gradient descent optimization, we develop more efficient sampling methods for sampling network weights that are built on improved optimizers for training network. While stochastic gradient descent (SGD) serves as the fundamental optimizer for deep learning, numerous extensions have been proposed that exhibit better empirical performance. Currently, the most prominent optimizer utilized in deep learning is adaptive moment optimization (**Adam**) method, which extended the SGD with techniques. One is the adaptive learning rate for each iteration, which resembles the existing Root Mean Square Propagation (RMSprop) algorithm. The other is the momentum term to accumulate the gradient of the past, when defining the updating direction. The superior empirical performance of Adam over plain SGD inspires us to study new Langevin MC methods from the perspective of the optimizer for deep learning. In this paper, we proposed a pre-conditioned under-damped Langevin dynamics for MC sampling of network weights, named as **Adam-LD**. The resulting Langevin MC method is closely connected to the Adam optimizer for deep learning.

With such an efficient Langevin MC method, one then can apply it to sample the posterior distribution $p(\theta|\boldsymbol{y},\epsilon_0)$, and the samples are used via (1.7) to approximate the MMSE estimate of the truth image $\boldsymbol{x}$. Together with the implicit regularization induced by the DIP, we

have a powerful self-supervised deep learning method for image recovery/reconstruction. To evaluate its empirical performance, extensive experiments are conducted on two representative inverse imaging problems: (1) sparse-view computerized tomography (CT) reconstruction for medical imaging and (2) phase retrieval for scientific imaging. The experiments showed that the method outperformed existing self-supervised deep learning solutions by a large margin.

2. Related works.

2.1. Regularization method with pre-defined/learnable prior. Regularization methods impose a pre-defined image prior to address the ill-posedness in solving (1.1). One widely used prior is sparsity-based, assuming the output of the latent image after applying some operator is sparse. The TV regularization method assumes the gradient of the image is sparse, while wavelet-based regularization assumes the wavelet coefficients of an image are sparse. These ℓ_1-norm related regularizations are widely used in image recovery tasks [10, 71, 14, 43]. These methods can also be interpreted as a maximum a posteriori (MAP) estimator in Bayesian inference. Instead of using a pre-defined sparsity-related image prior, some works learn the sparsity-based prior for image recovery. For image denoising, K-SVD [1] learns an over-complete dictionary for sparse coding of image patches, and Cai *et al*. [16] learn a wavelet tight frame for sparse approximation of an image. These learnable sparsity-based priors also have applications in other image restoration tasks.

2.2. Supervised deep learning method. In recent years, deep learning has shown promising performance in many image recovery applications, including denoising, image deblurring, compressive sensing, and computerized tomography, among others (*e.g*. [61, 93, 56, 92, 91, 76, 63, 30]). The majority of existing deep learning methods for general image recovery are based on supervised learning, requiring a training dataset with many pairs of measurements and corresponding latent images. An end-to-end deep neural network is then trained to have the least prediction error over the dataset. While earlier works trained an end-to-end neural network to predict the latent image directly from the measurement in a model-free way, for image recovery with a non-trivial forward model, a better approach is to encode the knowledge of the forward operator in the network architecture via unrolled optimization techniques [92, 91, 63, 30, 55]. The trained network models, however, are problem-specific and cannot be adapted to consider changes in the inverse problems, including the forward operator and instrumental configurations.

2.3. Deep learning with pre-trained DNN model. Training a DNN from scratch can be a troublesome and costly process. To avoid this, one approach is to use a pre-trained network model for regularizing image recovery. In the past, various types of pre-trained network models have been employed in image recovery tasks. One such method is the Plug-and-Play (PnP) approach, which utilizes a pre-trained denoising network model in image recovery. Similar to most optimization unrolling schemes, the PnP method unrolls an iterative scheme, where the prior-relating operation can be interpreted as a denoising process for removing artifacts. In the PnP method, this denoising process is replaced by a pre-trained denoising network model, which has shown promising empirical performance in various inverse imaging problems [45, 82, 94]. The PnP can be applied and interpreted within the context of MAP [80, 62] and approximate MMSE estimator [2, 37, 44, 50].

There have been some studies on the convergence properties of the PnP method with pre-trained denoising networks. However, most existing studies are limited to denoisers with very specific properties. For instance, in the context of the MAP estimator, differentiable and non-expansive denoisers with symmetric Jacobian [80] and linear denoisers [62] have been studied. Certain conditions on the denoising network, hyper-parameters, and Lipschitz continuity have been imposed for the convergence of the PnP method from the viewpoint of fixed-point iterations in [82, 83, 73, 90]. It is difficult to learn a good denoiser that approximates the MMSE estimator, as Gribonval [36] showed. Recently, the theoretical convergence of PnP as the MMSE estimate was studied in [50]. However, the applicability of these studies is limited, and in practice, most widely used denoising network models, such as DnCNN, are not applicable. The empirical performance of these PnP methods depends on the generalization performance of the pre-trained denoising network model over test data. In order to have good generalization performance, the pre-training of the denoising network still requires a large number of truth images related to the test data. As a result, its applicability remains limited in data-limited environments

Generative adversarial network (GAN) has also been used as a pre-trained model for image recovery. A pre-trained GAN model is usually domain-specific, taking a random initial seed as input and outputting an image in the specific domain. With such a pre-trained GAN model, one can convert the estimation of the latent image to the estimation of the initial seed, whose corresponding latent image fits the measurement. This GAN-based method has been employed in various applications, such as compressive sensing [12], denoising, and inpainting [5], image deconvolution [6], and phase retrieval [39]. However, GANs often suffer from training instability and model collapse, where the generator tends to generate a limited variety of samples [74, 4, 21]. Such training problems limit their application in general inverse problems. The existing successful GAN models are mostly in specific domains, such as face images or text images. For applications where it is challenging to have a large-scale dataset of latent images, it remains an open problem to have a generative model with good performance.

In addition to denoising models and GANs, pre-trained diffusion-based generative models can also be employed for solving inverse problems. These pre-trained diffusion models enable the generation of new samples from the empirical distribution of the given dataset [42, 79]. In inverse problems, the goal is to sample from the conditional density $p(\boldsymbol{x}|\boldsymbol{y})$. By decoupling this to the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ and the prior $p(\boldsymbol{x})$ in Bayesian inference, diffusion-based generative models allow one to sample the prior distribution and thus solve the problem [47, 46, 78]. However, diffusion models are often large and domain-specific, which can lead to some issues associated with diffusion-based methods. These issues may include high computational demands and limited generalization performance, particularly when the target domain is significantly different from the domain for which the diffusion model was specifically trained.

2.4. Self-supervised or unsupervised deep learning method. Self-supervised or unsupervised deep learning refer to deep learning without using any training sample, which refers to the pair of a latent image and its measurement in the context of image recovery. Recently, for its practical value in data-limited environments, self-supervised or unsupervised deep learning for image recovery is receiving an increasing interest in the community. There has been rapid progress on image denoising. The pioneering work is DIP [86] which shows that regular image

structures appear before random noise when training a CNN-based denoising network. Then, early stopping can be used to have a self-supervised image denoising network. Another approach to self(un)-supervised deep denoising network is to construct the loss functions without using truth images that can simulates the loss function supervised over truth images; See *e.g.* [49, 9, 51, 67, 66]. Metzler *et al.* [58] is based on Stein's unbiased risk estimator (SURE) for regularizing the training of a denoising network without truth images.

In comparison to image denoising with $\Phi = \boldsymbol{I}$, existing studies are much fewer on self-supervised learning for ill-posed inverse imaging problems with other operators Φ. How to address ill-posedness of the problem is the main focus of the study, and the techniques for image denoising cannot be trivially generalized to solve these ill-posed problems.

In addition to image denoising, DIP also has been using for solving inverse problems such as inpainting [86], which addresses the over-fitting by using early stopping. However, its performance is not very competitive as the implicit regularization effect by DIP is not very powerful. Some extensions of DIP are developed solving various inverse problems. Heckel *et al.* [40, 28] proposed to use an under-parameterized network deep decoder as the untrained NN for solving inverse problems. While the over-fitting is alleviated by using fewer network weights, its representative capacity is also reduced and it leads to performance loss. Shi *et al.* [77] proposed to accelerate the training and correct spectral bias of DIP, by adopting a Lipschitz-controlled convolution layer and a Gaussian-controlled up-sampling layer in the network architecture.

Bostan [13] proposed to use an untrained deep decoder for phase retrieval, where the overfitting is handled by dimension reduction induced by the decoder. Zhussip *et al.* [97] proposed to regularize the denoiser by SURE in the iterative approximate message passing (AMP) scheme for image reconstruction from under-sampled measurements. For compressing sensing, Pang *et al.* [65] proposed to train a Bayesian neural network (BNN) whose network weights following a normal distribution. The BNN is trained as the approximating distribution to the posterior distribution in the variational approximation method of an MMSE estimator. The BNN method is quite computationally expensive. For phase retrieval, Chen *et al.* [24] proposed to use dropout-based network for approximating the MMSE estimator. Chen *et al.* [22] proposed a task-dependent data augmentation to train the network model via compensating the under-sampled observations in CT or image inpainting.

2.5. Organization. The paper is organized as follows. In section 3, we give an introduction to the DNN-based re-parametrization, the MMSE estimator, and the plain Langevin MC sampling. Section 4 presents a new method related to the Langevin MC method motivated from the Adam optimizer, for efficiently sampling network weights. Then they are used for developing a self-supervised deep learning solution for general image recovery. Section 5 is devoted to the experimental evaluation of the proposed self-supervised method on two representative image recovery problems: linear sparse-view CT image reconstruction and nonlinear phase retrieval. Section 6 concludes the paper.

3. CNN-based re-parametrization, MMSE estimator, and MCMC sampling.

This section is devoted to the discussion of the MMSE estimator for a DNN-based re-parametrization of latent image.

3.1. CNN-based re-parametrization and MMSE estimator. In this paper, the inverse problem $\boldsymbol{y} = \Phi(\boldsymbol{x}) + \boldsymbol{n}$ is discussed from the perspective of Bayesian inference, *i.e.*, the latent image $\boldsymbol{x}$ is treated as one sample from a probability density $p(\boldsymbol{x})$. Then, the MAP estimator is about finding a single mode which maximizes the density $p(\boldsymbol{x}|\boldsymbol{y})$. By Bayes' rule, the MAP estimator can be formulated as solving the following optimization problem:

$$\boldsymbol{x}_{\text{MAP}} := \operatorname*{argmax}_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}} -\log p(\boldsymbol{x}|\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}} -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{x}), \tag{3.1}$$

where $p(\boldsymbol{x})$ denotes the prior distribution of latent image $\boldsymbol{x}$. In traditional regularization methods, $p(\boldsymbol{x})$ is pre-defined. For example, by assuming

$$p(\boldsymbol{x}) \propto \exp(-\lambda \|\nabla \boldsymbol{x}\|_1), \tag{3.2}$$

the resulting MAP estimator is indeed the solution of the classic TV regularization method.

In addition to MAP, MMSE is another estimator whose estimate minimizes the mean squared error

$$\boldsymbol{x}_{\text{MMSE}} = \min_{\boldsymbol{u}} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y}} \|\boldsymbol{u} - \boldsymbol{x}\|_2^2, \tag{3.3}$$

whose solution is indeed the conditional expectation:

$$\boldsymbol{x}_{\text{MMSE}} = \boldsymbol{x}_{\text{CE}} = \int \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}. \tag{3.4}$$

Same as the MAP estimator, the MMSE estimator also needs to know the distribution $p(\boldsymbol{x}|\boldsymbol{y})$. Moreover, the computational complexity of the MMSE estimator is often higher than the MAP estimator as it needs to calculate an integral in a high-dimensional space.

In this paper, we propose to use an untrained CNN for representing latent image, and estimate the corresponding MMSE estimate. Such an approach is motivated from DIP [86], the implicit regularization induced by the network architecture of CNN. Such an implicit regularization can be beneficial for tackling the ill-posedness of the problem, as observed in many existing works. Thus, we consider an untrained CNN $f(\epsilon_0; \theta)$ such that the image $\boldsymbol{x}$ can be replicated by the network:

$$\boldsymbol{x} = f(\epsilon_0; \theta) \tag{3.5}$$

for some fixed initial seed ϵ_0. In other words, the image $\boldsymbol{x}$ is re-parameterized by the vector of network weights θ such that

$$\theta \xrightarrow{f(\epsilon_0, \cdot)} \boldsymbol{x}. \tag{3.6}$$

Remark 3.1. For a function $f(\epsilon_0; \theta)$ expressed by a CNN with ReLU/LeakyReLU activation function, universal approximation theorems [7, 96] have shown that, $f(\epsilon_0; \theta)$ can approximate any continuous function to an arbitrary accuracy when the depth of the network is large enough. Despite not being differentiable in only a set with zero measure, such a function can still be optimized with gradient descent-based optimizers. The function is not injective, meaning there are many sets of network weights that will give the same output. This property is desirable in deep learning, as it increases the likelihood of finding a good solution among the many possible solutions of a highly non-convex problem.

With such a re-parametrization, under certain condition on the distribution functions of $\boldsymbol{x}$ and θ, the conditional mean still have an analytic:

$$\int \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = \int f(\epsilon_0;\theta) p(\theta|\boldsymbol{y},\epsilon_0) d\theta. \tag{3.7}$$

For the completeness, we outlines its derivation in the following. For two measurable distribution function $p_x(\boldsymbol{x})$ and $p_\theta(\theta)$, define their cumulative distribution functions by

$$dP_x(\boldsymbol{x}) = p_x(\boldsymbol{x})dx, \quad dP_\theta(\theta) = p_\theta(\theta)d\theta \tag{3.8}$$

respectively. Suppose that both cumulative distribution functions $P_x(\boldsymbol{x})$ and $P_\theta(\theta)$ are absolutely continuous. Then, we have

$$\int \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = \int \boldsymbol{x} \frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})} p_x(\boldsymbol{x}) d\boldsymbol{x} = \int \boldsymbol{x} \frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})} dP_x(\boldsymbol{x}) \tag{3.9}$$

Using $\boldsymbol{x} = f(\epsilon_0;\theta)$, we have

$$\int \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = \int \boldsymbol{x} \frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})} dP_x(\boldsymbol{x}) = \int f(\epsilon_0;\theta) \frac{p(\boldsymbol{y}|f(\epsilon_0,\theta))}{p(\boldsymbol{y})} dP_\theta(\theta) = \int f(\epsilon_0;\theta) p(\theta|\boldsymbol{y},\epsilon_0) d\theta. \tag{3.10}$$

As $p(\theta|\boldsymbol{y},\epsilon_0)$ in (3.7) is not available, we re-express it by Bayes' rule:

$$p(\theta|\boldsymbol{y},\epsilon_0) \propto p(\boldsymbol{y}|\theta,\epsilon_0) p(\theta). \tag{3.11}$$

In this paper, we impose a Gaussian prior on network weights θ:

$$\theta \propto \exp(-\frac{\|\theta\|_2^2}{2\sigma_\theta^2}). \tag{3.12}$$

Then, in the presence of i.i.d. Gaussian white noise $\boldsymbol{n} \propto \exp(-\frac{\|\boldsymbol{n}\|_2^2}{2\sigma_n^2})$, we have

$$-\log p(\theta|\boldsymbol{y},\epsilon_0) = \frac{M}{2\sigma_n^2} L(\theta) + \text{const.}, \tag{3.13}$$

where

$$L(\theta) := \frac{1}{M} \|\Phi((f(\epsilon_0;\theta)) - \boldsymbol{y}\|_2^2 + \frac{\sigma_n^2}{\sigma_\theta^2 M} \|\theta\|_2^2 . \tag{3.14}$$

Here, we employ the conventional loss definition normalized by the dimension M of the data $\boldsymbol{y}$. The regularization parameter $\frac{\sigma_n^2}{\sigma_\theta^2 M}$ is referred to as the parameter for weight decay. Then, the conditional probability distribution function $p(\theta|\boldsymbol{y},\epsilon_0)$ can be expressed as

$$p(\theta|\boldsymbol{y},\epsilon_0) := \frac{1}{\int \exp(-\frac{1}{2\sigma_n^2/M} L(\theta)) d\theta} \exp(-\frac{1}{2\sigma_n^2/M} L(\theta)). \tag{3.15}$$

In summary, to utilize the implicit regularization induced by DIP of a CNN, we re-parameterize the image $\boldsymbol{x} = f(\epsilon_0;\theta)$ by a untrained CNN model. Then, the MMSE estimator for $\boldsymbol{x}$ can be computed via calculating the integral defined in (3.7) where $p(\theta|\boldsymbol{y},\epsilon_0)$ is given by (3.15).

3.2. Review on overdamped Langevin MC sampling and Plain SGD. For an MMSE estimator expressed by (3.7), one needs to calculate an integral in a very high-dimensional space. Thus, a natural choice is using the MC method for calculating the integral. There are many MC sampling methods. In this paper, we propose to use Langevin MC sampling method for our purpose. The motivations come from the deep connection between Langevin MC sampling and stochastic gradient descent (SGD) method widely used in the optimizer for deep learning. For the completeness, we first give a brief introduction on plain Langevin MC method.

The plain overdamped Langevin MC method is connected to the following stochastic differential equation (SDE):

$$d\theta_t = -\nabla L(\theta_t)dt + \sqrt{2}dW_t, \tag{3.16}$$

where W_t denotes Brownian motion. For each time stamp t, the stochastic dynamics (3.16) defines a random variable θ_t. The evolution of the distributions of θ_t can be described by the Fokker-Planck equation. To facilitate the analysis of other dynamics, we present the Fokker-Planck equation for a general SDE.

Lemma 3.2 (Fokker-Planck equation [35]). *Considering the SDE:*

$$dx_t = g(x_t)dt + \sqrt{2D(x_t)}dW_t,$$

where $x_t \in \mathbb{R}^n, g(x_t) \in \mathbb{R}^n, D(x_t) \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix. Assume that the element functions $g_i(\cdot), D_{ij}(\cdot)$ belong to $C^2(\mathbb{R}^n)$, then the distribution $p_t(x)$ is governed by the following equation

$$\partial_t p_t(x) = -\sum_i^n \partial_{x_i}[g_i(x)p_t(x)] + \sum_i^n\sum_j^n \partial_{x_i}\partial_{x_j}[D_{ij}(x)p_t(x)]$$

where $D_{ij}(x)$ denotes the (i,j)-th entry of the matrix D.

Interested readers are referred to [35] for more details. For our purpose, consider the specific Langevin dynamics (3.16). By Lemma 3.2, the governed equation of the distribution $p_t(x)$ is

$$\partial_t p_t(\theta) = \frac{\partial}{\partial\theta}[\nabla L(\theta)p_t(\theta)] + \frac{\partial^2}{\partial\theta^2}[p_t(\theta)].$$

It can be shown [35] that, under mild conditions, that the θ_t is a random variable of the distribution $p_t(\theta)$, which converges to a stationary distribution $p_\infty(\theta) \propto \exp(-L(\theta))$ by taking $\partial_t p_t(\theta) \to 0$.

The discretization of the over-damped Langevin dynamics (3.16) gives an MCMC sampling scheme (SGLD): for $k = 0, 1, \ldots,$

$$\theta_{k+1} = \theta_k - \gamma_k \cdot \nabla L(\theta_k) + \sqrt{2\gamma_k}\cdot\epsilon, \tag{SGLD}$$

where $\epsilon \sim \mathcal{N}(0, I)$. It can be seen that the iteration (SGLD) indeed is a plain SGD method defined by adding i.i.d. Gaussian white noise to the gradient descent method for minimizing

the loss function $L(\theta)$. Though the SGLD (a.k.a. ULA, unadjusted Langevin algorithm) can converge to the stationary distribution, its convergence rate is quite slow [68, 69, 33], which might result in very high computational cost. To meet practice needs in computational efficiency, we need to consider more efficient Langevin diffusions to enable its application to practical sampling task.

Now, with in hand an MCMC sampling method such as the over-damped Langevin MC scheme given by (SGLD), one can efficiently sample the distribution $p(\theta|\boldsymbol{y}, \epsilon_0)$ defined by (3.15). One common practice is running (SGLD) for total T iterations, and we assume after T' (burning-in) iterations, the distribution of the samples is close to the distribution $p(\theta|\boldsymbol{y}, \epsilon_0)$. By MC-based integration scheme, we can approximate the MMSE estimate by

$$\boldsymbol{x}_{\text{MMSE}} = \int f(\epsilon_0; \theta) p(\theta|\boldsymbol{y}, \epsilon_0) d\theta \simeq \frac{1}{T - T'} \sum_{i=T'}^{T} f(\epsilon_0; \theta_i). \tag{3.17}$$

Similarly, by the same scheme, we can also compute the variance of the estimated image

$$\text{Var}(\boldsymbol{x}_{\text{MMSE}}) \sim \int f^2(\epsilon_0; \theta) p(\theta|\boldsymbol{y}, \epsilon_0) d\theta - (\int f(\epsilon_0; \theta) p(\theta|\boldsymbol{y}, \epsilon_0) d\theta)^2. \tag{3.18}$$

It can be used for measuring the uncertainty of the estimate.

4. Langevin MC method motivated from the Adam optimizer. The classic Langevin MC method (SGLD) is connected to plain SGD. Nevertheless, plain SGD is a basic optimizer for training a DNN, and there are many other extensions of SGD for better efficiency and effectiveness when training a DNN. For example, plain SGD uses a constant learning rate for all network weights, which is not only sub-optimal but also has a negative impact on the generalization performance of the trained model. In this section, for the purpose of effectively sampling network weights for calculating MMSE estimate, we presented one Langevin MC method which are motivated by one prominent optimizer widely adopted in deep learning, the Adam method. There are two parts in Adam. One is adaptive learning rate borrowed from RMSprop, and the other is the momentum-based acceleration. While it is not clear how the training efficiency of an SGD-based optimizer for deep learning is connected to the sampling efficiency of the corresponding Langevin MC method, extensive experiments showed that there is a close correlation between two. Before proceeding, we would like to note that the stationary distribution is derived from the continuous SDE, while the algorithm is based on its discretization. It is important to recognize that the discretized stochastic process may not necessarily converge to the continuous one, as pointed out in [29]. Therefore, the discretization introduces a bias to the target distribution, which asymptotically approaches zero with a small and decreasing step size.

4.1. RMSprop and Preconditioned overdamped Langevin MC sampling. When training a deep network, for each iteration, the plain SGD uses a constant learning rate for all network weights. The learning rate plays an important role in deep learning, in both computational efficiency and generalization performance. The RMSprop is an unpublished yet well-known

adaptive learning rate method. For a loss function $L(\theta)$, RMSprop updates the estimates by

$$v_t = \beta v_{t-1} + (1-\beta)(\nabla L(\theta_t))^2; \tag{4.1}$$

$$G(\theta_t) = \mathrm{diag}(\frac{1}{\sqrt{v_t}+\epsilon}); \tag{4.2}$$

$$\theta_{t+1} = \theta_t - \eta G(\theta_t)\nabla L(\theta_t), \tag{4.3}$$

where η is a constant learning rate and β is called the moving average parameter. RMSprop optimizer is closely related to a preconditioned overdamped Langevin MC method, referred as p-LD method in [52]. Li *et al.* [52] considered the following governed stochastic dynamics

$$d\theta_t = -G(\theta_t)\nabla L(\theta_t)dt + \Gamma(\theta_t)dt + \sqrt{2G(\theta_t)}dW_t, \tag{4.4}$$

where $G(\theta_t) > 0$ is the preconditioned matrix and $\Gamma(\theta_t) \in \mathbb{R}^n$ is a vector and its i-th entry $\Gamma_i(\theta_t) = \sum_j \frac{\partial G_{i,j}(\theta_t)}{\partial \theta_{t,j}}$, where $G_{i,j}(\theta_t)$ is the (i,j) entry of $G(\theta_t)$.

By Lemma 3.2, one can verify that the distribution evolution of the dynamics (4.4) satisfies

$$\begin{aligned} \partial_t p_t(\theta) &= \frac{\partial}{\partial\theta}\{[G(\theta)\nabla L(\theta) - \Gamma(\theta)]p_t(\theta)\} + \frac{\partial^2}{\partial\theta^2}[G(\theta)p_t(\theta)] \\ &= \frac{\partial}{\partial\theta}\{G(\theta)\nabla L(\theta)p_t(\theta) + G(\theta)\frac{\partial}{\partial\theta}p_t(\theta)\}. \end{aligned} \tag{4.5}$$

Let $\partial_t p_t(\theta) = 0$. By $G(\theta) > 0$, we have the stationary distribution $p_\infty(\theta) \propto \exp(-L(\theta))$. When the preconditioned matrix $G(\theta) = I$, the dynamics (4.4) degenerates to the classical over-damped Langevin dynamics (3.16).

The discrete preconditioned overdamped Langevin dynamics scheme can be written as

$$\theta_{t+1} = \theta_t - \eta G(\theta_t)\nabla L(\theta_t) + \eta\Gamma(\theta_t) + \sqrt{2\eta G(\theta_t)}\mathcal{N}(0,1), \tag{4.6}$$

where η denotes the learning rate. Li *et al.* [52] leveraged the following diagonal matrix whose entries are the moving average of squared gradient, *i.e.*,

$$G(\theta_t) = \mathrm{diag}(\frac{1}{\sqrt{v_t}+\epsilon}),\ \text{where } v_t = \beta v_{t-1} + (1-\beta)(\nabla L(\theta_t))^2, \tag{4.7}$$

where $\epsilon > 0$ is to avoid the overflow of the division by zero. As a comparison to RMSprop, there are two differences: the inclusion of $\Gamma(\theta_t)$ term and the scaled Gaussian injection noise $\sqrt{2\eta G(\theta_t)}\mathcal{N}(0,1)$.

According to the finite-time convergence analysis presented in [52], the MSE between the expectation average and the practical sample average can be upper bounded, and the error asymptotically approaches zero. By omitting $\Gamma(\theta_t)$, an extra term is introduced in the upper bound that is controlled by the moving average hyperparameter β and is close to zero if the parameter is close to 1. In this case, the effect of $\Gamma(\theta_t)$ is negligible and then this term is neglected for practical implementation. The resulting Langevin dynamics is refereed to p-LD,

which runs as

$$\text{(p-LD)}\qquad \begin{aligned} v_t &= \beta v_{t-1} + (1-\beta)(\nabla L(\theta_t))^2; \\ G(\theta_t) &= \operatorname{diag}\Big(\frac{1}{\sqrt{v_t}+\epsilon}\Big); \\ \theta_{t+1} &= \theta_t - \eta G(\theta_t)\nabla L(\theta_t) + \sqrt{2\eta G(\theta_t)}\mathcal{N}(0,1). \end{aligned}$$

In short, the preconditioned matrix for the equalized gradient makes the update of weights to be dependent on the local curvature of the landscape. That is, for flat regions, the effective learning rate is larger while for a curved region, the effective learning rate is smaller.

4.2. SGD with momentum and underdamped Langevin dynamics. Another modification to plain SGD is to include the momentum in the gradient update scheme, which help gradient descent method avoiding being trapped at local minima or a saddle point [84]. For a loss function $L(\theta)$, the SGD with momentum updates the parameter by

$$\begin{aligned} v_k &= (1-\alpha)v_{k-1} - \eta\nabla L(\theta_{k-1}); \\ \theta_k &= \theta_{k-1} + v_k, \end{aligned} \tag{4.8}$$

where α denotes the hyper-parameter and η denotes the learning rate. The SGD with momentum (4.8) is closely related to an underdamped Langevin dynamics.

Considering the underdamped second-order Langevin dynamics

$$\frac{d^2\theta_t}{dt^2} = -\frac{d\theta_t}{dt} - \nabla L(\theta_t) + \sqrt{2}z(t), \tag{4.9}$$

where $z(t)$ denotes Gaussian random variable. It can be re-expressed as

$$\begin{aligned} d\theta_t &= m_t dt \\ dm_t &= -\nabla L(\theta_t)dt - m_t dt + \sqrt{2}dW_t, \end{aligned} \tag{4.10}$$

where the variable m_t is called the momentum variable. Recall a Hamilton system reads

$$\frac{d\vartheta_t}{dt} = \frac{\partial \mathcal{H}}{\partial \mu_t}, \quad \frac{d\mu_t}{dt} = -\frac{\partial \mathcal{H}}{\partial \vartheta_t}, \tag{4.11}$$

where the Hamilton is defined as $\mathcal{H}(\vartheta_t, \mu_t) := L(\vartheta_t) + \frac{1}{2}\mu_t^T\mu_t$. Then, the first-order dynamics system (4.10) can be interpreted as a Hamilton system with an artificial fraction term and an additional Brownian motion. This interpretation implies that the underdamped Langevin dynamics performs similarly as the HMC sampling method. Indeed, the stationary distribution of the underdamped Langevin dynamics is $p_\infty(\theta, m) \propto \exp(-\mathcal{H}(\theta, m))$. Hence its marginalization distribution of θ is proportional to $\exp(-L(\theta))$.

Theorem 4.1 (Stationary distribution). *Considering an underdamped Langevin dynamics system defined by (4.10). Assume $L(\theta) \in C^2(\mathbb{R}^n)$. Then, its stationary distribution $p_\infty(\theta, m)$ satisfies*

$$p_\infty(\theta, m) \propto \exp(-\mathcal{H}(\theta, m)),$$

where the Hamilton is defined as $\mathcal{H}(\theta, m) := L(\theta) + \frac{1}{2}m^T m$. Furthermore, the marginalized distribution $p_\infty(\theta) \propto \exp(-L(\theta))$.

Proof. Let $F = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}$ and $D = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$, and consider the stacked variable (θ, m). Then, the dynamics system can be written as follows.

$$d\begin{pmatrix} \theta \\ m \end{pmatrix} = -\begin{pmatrix} 0 & -I \\ I & I \end{pmatrix}\begin{pmatrix} \nabla L(\theta) \\ m \end{pmatrix} dt + \sqrt{2}dW_t = -[D+F]\nabla\mathcal{H}(\theta, m)dt + \sqrt{2D}dW_t,$$

where $\sqrt{D}$ denotes the Hadamard element-wise operation of taking square root.

The evolution of the joint distribution $p_t(\theta, m)$ under the dynamics is governed by the Fokker-Planck equation

$$\partial_t p_t(\theta, m) = \nabla^T[(D+F)\nabla\mathcal{H}(\theta, m)p_t(\theta, m)] + \nabla^T[Dp_t(\theta, m)]\nabla, \tag{4.12}$$

where the operator in the last term is defined as

$$\begin{aligned} \nabla^T[Dp_t(\theta, m)]\nabla = &\sum_{i,j} \partial_{\theta_i}\partial_{\theta_j}[D_{ij}^{11}p_t(\theta, m)] + \sum_{i,j} \partial_{\theta_i}\partial_{m_j}[D_{ij}^{12}p_t(\theta, m)] \\ &+ \sum_{i,j} \partial_{m_i}\partial_{\theta_j}[D_{ij}^{21}p_t(\theta, m)] + \sum_{i,j} \partial_{m_i}\partial_{m_j}[D_{ij}^{22}p_t(\theta, m)], \end{aligned} \tag{4.13}$$

where $D = \begin{pmatrix} D^{11} & D^{12} \\ D^{21} & D^{22} \end{pmatrix}$. The right-hand side of (4.12) can be re-written as

$$\text{RHS} = \nabla^T\left\{[D+F][\nabla\mathcal{H}(\theta, m)p_t(\theta, m) + \nabla p_t(\theta, m)]\right\},$$

since D, F are independent of both (θ, m), and we have $\nabla^T(F\nabla p_t(\theta, m)) = \partial_m\partial_\theta p_t(\theta, m) - \partial_\theta\partial_m p_t(\theta, m) = 0$. Then, let $\partial_t p_t(\theta, m) = 0$. We conclude that the invariant stationary distribution $p_\infty(\theta, m) \propto \exp(-\mathcal{H}(\theta, m))$. The marginalized distribution can be obtained similarly. ∎

The discrete scheme of the system (4.10) is closely related to the SGD with momentum [25]. We write down the discredited scheme of the system (4.10). Suppose that the step-size for the system is set to ϵ, using backward difference and forward difference formula for the derivative, the discretization scheme of (4.10) is

$$\begin{aligned} \theta_k - \theta_{k-1} &= \epsilon m_k \\ m_k - m_{k-1} &= -\epsilon\nabla L(\theta_{k-1}) - \epsilon m_{k-1} + \sqrt{2\epsilon}\mathcal{N}(0, 1). \end{aligned}$$

By setting $v = \epsilon m$ and $\alpha = \epsilon, \eta = \epsilon^2$, we have the Langevin dynamics with the name SGm-LD:

$$\begin{aligned} v_k &= (1-\alpha)v_{k-1} - \eta\nabla L(\theta_{k-1}) + \sqrt{2\alpha\eta}\mathcal{N}(0, 1) \\ \theta_k &= \theta_{k-1} + v_k. \end{aligned} \tag{SGm-LD}$$

The difference between SGm-LD and the SGD with momentum optimizer (4.8) is the injection of Gaussian noise along the iteration.

4.3. Adam and Preconditioned underdamped Langevin dynamics. In the previous sections, we discussed two extensions of plain SGD and their connections to the related Langevin dynamics: adaptive learning rate and the usage of momentum. The combination of these two extensions leads to the Adam optimizer, a widely used one in training DNN. In this section, we present a complete discussion on the Langevin dynamics that is related to Adam, which results in a Adam-motivated Langevin MC method for effectively calculating the integral of the MMSE estimator.

The dynamics related to Adam is defined as follows.

$$\begin{aligned} d\theta_t &= G(\theta_t)m_t dt; \\ dm_t &= -G(\theta_t)\nabla L(\theta_t)dt - m_t dt + \Gamma(\theta_t)dt + \sqrt{2}dW_t, \end{aligned} \tag{4.14}$$

where $\Gamma(\theta_t) := \frac{\partial}{\partial\theta}G(\theta_t)$. The stationary distribution associated to the new dynamics (4.14) is given in the following theorem.

Theorem 4.2 (Stationary distribution). *Consider the dynamics* (4.14)*.Assume that* $L(\theta) \in C^2(\mathbb{R}^n)$ *and* $G(\theta) > 0$*, then its stationary distribution of* θ *satisfies* $p_\infty(\theta) \propto \exp(-L(\theta))$*.*

Proof. Denote $F = \begin{pmatrix} 0 & -G(\theta_t) \\ G(\theta_t) & 0 \end{pmatrix}$ and $D = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$. Consider the stacked variable (θ, m). Then, the dynamics system (4.14) can be re-written as follows.

$$\begin{aligned} d\begin{pmatrix} \theta \\ m \end{pmatrix} &= -\begin{pmatrix} 0 & -G(\theta_t) \\ G(\theta_t) & I \end{pmatrix}\begin{pmatrix} \nabla L(\theta_t) \\ m \end{pmatrix} dt + \begin{pmatrix} 0 \\ \Gamma(\theta_t) \end{pmatrix} dt + \sqrt{2D}dW_t \\ &= -[D+F]\begin{pmatrix} \nabla L(\theta_t) \\ m \end{pmatrix} dt + \begin{pmatrix} 0 \\ \Gamma(\theta_t) \end{pmatrix} dt + \sqrt{2D}dW_t. \end{aligned}$$

By Lemma 3.2 , the evolution of the joint distribution $p_t(\theta, m)$ is governed by the Fokker-Planck equation:

$$\partial_t p_t(\theta,m) = \nabla^T[(D+F)\begin{pmatrix} \nabla L(\theta) \\ m \end{pmatrix} p_t(\theta,m) - \begin{pmatrix} 0 \\ \Gamma(\theta_t) \end{pmatrix} p_t(\theta,m)] + \nabla^T[Dp_t(\theta,m)]\nabla.$$

Using the notation (4.13), we have

$$\nabla^T[Fp_t(\theta,m)]\nabla = -\sum_{i,j}\frac{\partial^2}{\partial\theta_i\partial m_j}\left(G(\theta_t)p_t(\theta,m)\right) + \sum_{i,j}\frac{\partial^2}{\partial m_i\partial\theta_j}\left(G(\theta_t)p_t(\theta,m)\right). \tag{4.15}$$

By the interchange property of partial derivatives of a smooth function, we conclude that $\nabla^T[Fp_t(\theta,m)]\nabla = 0$. Thus we have

$$\partial_t p_t(\theta,m) = \nabla^T[(D+F)\begin{pmatrix} \nabla L(\theta) \\ m \end{pmatrix} p_t(\theta,m) - \begin{pmatrix} 0 \\ \Gamma(\theta_t) \end{pmatrix} p_t(\theta,m)] + \nabla^T[(D+F)p_t(\theta,m)]\nabla.$$

Notice that the matrix F only depends on the variable θ_t. By the chain rule, we have

$$\nabla^T[(D+F)p_t(\theta,m)]\nabla = \nabla^T[(D+F)\nabla p_t(\theta,m)] + \nabla^T[((D+F)\nabla)\,p_t(\theta,m)].$$

Therefore, we have the Fokker-Planck equation

$$\partial_t p_t(\theta,m) = \nabla^T\left[(D+F)\left(\begin{pmatrix}\nabla L(\theta)\\ m\end{pmatrix}p_t(\theta,m)+\nabla p_t(\theta,m)\right)\right]$$
$$-\nabla^T[\begin{pmatrix}0\\ \Gamma(\theta_t)\end{pmatrix}p_t(\theta,m)]+\nabla^T[((D+F)\nabla)\,p_t(\theta,m)].$$

Notice that

$$(D+F)\nabla=\begin{pmatrix}0 & -G(\theta_t)\\ G(\theta_t) & I\end{pmatrix}\begin{pmatrix}\frac{\partial}{\partial\theta}\\ \frac{\partial}{\partial m}\end{pmatrix}=\begin{pmatrix}0\\ \Gamma(\theta_t)\end{pmatrix},$$

Thus, the Fokker-Planck equation becomes

$$\partial_t p_t(\theta,m) = \nabla^T\left[(D+F)\left(\begin{pmatrix}\nabla L(\theta)\\ m\end{pmatrix}p_t(\theta,m)+\nabla p_t(\theta,m)\right)\right].$$

Let $\partial_t p_t(\theta,m)=0$. Suppose the matrix $D+F$ is positive semi-definite. The stationary distribution of the dynamics is then proportional to $\exp(-L(\theta)-\frac{1}{2}\|m\|_2^2)$. Then, one can conclude the marginalized distribution of θ. ∎

Same as [52], we also neglect the $\Gamma(\theta_t)$ term in the practical implementation of the dynamics (4.14). The computation of $\Gamma(\theta_t)$ is not manageable in high-dimensional problem such as training a deep neural network. Although the finite-time convergence of our Adam-LD and the effect of $\Gamma(\theta_t)$ on convergence are still unknown, we expect similar convergence properties as p-LD to hold. Then, the discrete dynamics of (4.14) is closely connected to Adam optimizer in deep learning.

Suppose that the step-size for the system is set to ϵ and the $\Gamma(\theta_t)$ term is neglected. Using the backward difference and forward difference, one can express a discretization scheme as the following:

$$\theta_k-\theta_{k-1}=\epsilon G(\theta_k)m_k;$$
$$m_k-m_{k-1}=-\epsilon G(\theta_{k-1})\nabla L(\theta_{k-1})-\epsilon m_{k-1}+\sqrt{2\epsilon}\mathcal{N}(0,1)$$

Define $v=\epsilon m$ and $\alpha=\epsilon,\eta=\epsilon^2$. One have

$$v_k=(1-\alpha)v_{k-1}-\eta G(\theta_{k-1})\nabla L(\theta_{k-1})+\sqrt{2\alpha\eta}\mathcal{N}(0,1);$$
$$\theta_k=\theta_{k-1}+G(\theta_{k-1})v_{k-1}.$$

Equivalently, it can be rewritten as

$$\begin{aligned}v_k&=(1-\alpha)v_{k-1}+\alpha\nabla L(\theta_{k-1});\\ \theta_k&=\theta_{k-1}-\alpha G(\theta_k)G(\theta_{k-1})v_k+\sqrt{2\alpha}G(\theta_k)\mathcal{N}(0,1).\end{aligned}\tag{4.16}$$

The discretization scheme (4.16) is related to Adam optimizer by replacing the preconditioned matrix $G(\theta_{k-1})$ by $G(\theta_k)$ in (4.16) and generating it by the moving average of squared gradient:

$$G^2(\theta_t)=\mathrm{diag}(\frac{1}{\sqrt{\tilde{m}_k}+\epsilon}),\ \text{where } \tilde{m}_t=\beta\tilde{m}_{t-1}+(1-\beta)(\nabla_\theta L(\theta_{t-1}))^2.$$

The iterative scheme (4.16) clearly resembles Adam optimizer. The parameter α affects both the learning rate and the weighting parameter $1-\alpha$ for the gradient to update v_k, while β serves as the weighting parameter for the squared gradient update of the preconditioned matrix. To minimize bias error in the expectation average and practical sample average, it is recommended to set the parameter β close to 1 [52]. Consequently, in our experiments, we set β to 0.99. The stepsize α plays an important role in controlling both the asymptotic accuracy and the convergence speed to the stationary state of Adam-LD. In our experiments, we set α to $1e-2$ to achieve fast convergence to the stationary regime, even though it comes at the expense of some asymptotic bias. This practice closely follows the default setting used in the Adam optimizer to smoothen the convergence path for nonconvex learning problems and enhance learning efficiency.

The main difference between Langevin dynamics and Adam lies in the additional scaled noise term in Langevin dynamics. See Alg. 4.1 for the outline of the proposed Langevin MC algorithm motivated from Adam, which is called **Adam-LD** in this paper. Alg. 4.1 is for sampling from the distribution proportional to $\exp(-h^{-1}L(\theta))$. The scaling parameter h is the temperature of the posterior distribution density. In the presence of measurement noise, the scaling parameter h corresponds to the temperature of the posterior distribution density, which is proportional to the variance of measurement noise σ_n^2. Motivated by the effectiveness of the step-size correction of Adam, we also include such a correction step in Alg. 4.1, *i.e.* Step 7 in Alg. 4.1. Different from Adam, the moving average parameter for the gradient of the Langevin sampling scheme depends on the learning rate α.

Remark 4.3 (Hyper-parameter h for noise-free measurement). The theoretical derivation for the hyper-parameter h is not applicable to noise-free data. Strictly speaking, the averaging is not the MMSE prediction for deterministic noiseless scenario. Nevertheless, empirical results show that setting the value of h to 0 performs well. In this case, the resulting algorithm becomes deterministic, relying on the deterministic Adam algorithm for training the network and regularization provided by DIP [86]. However, our approach differs from DIP in that we use sample averaging for estimation, while DIP uses exponential moving averaging for prediction.

Remark 4.4 (Connection to other langevin MC samplers in deep learning). Based on a standard overdamped Langevin dynamics, SGLD has been employed in [26] for image denoising and inpainting using an untrained network. The proposed Adam-LD methods extended the dynamics to a preconditioned underdamped dynamics, which is connected to the Adam optimizer. The motivation of such an extension is for faster convergence rate and more efficient sampling. The proposed Adam-LD is also applicable to general image recovery problem with arbitrary measurement matrix. There are also some Langevin MC samplers for deep learning with the extension to a preconditioned dynamics; See *e.g.* *Adam SGLD*[11] and *Adaptively preconditioned SGLD* [48]. Both the proposed Adam-LD and Adam SGLD introduce adaptive bias to the drift term. The injected Gaussian noise in the proposed Adam-LD is modified with a pre-conditioned matrix, and injected Gaussian noise in the Adam SGLD method is plain Gaussian noise without any pre-conditioning. Both Adam-LD and adaptively preconditioned SGLD inject pre-conditioned Gaussian noise. The proposed Adam-LD utilizes the drift term, while adaptively preconditioned SGLD method does not consider the drift term.

Algorithm 4.1 Adam motivated Langevin MC sampling method (Adam-LD)

Input: Scaling h, step-size α, moving average parameter β, tolerance ϵ, objective function $L(\theta)$, maximum iteration number T;

Output: Samples from $\pi(\theta) \propto \exp(-h^{-1}L(\theta))$: $\{\theta_t\}_{t=T',\dots,T}$, where T' is the number for "burn-in" iterations

Initialization: $t \leftarrow 0$, $v_0 \leftarrow 0$
while $t < T$ **do**
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_\theta L(\theta_{t-1})$
  $m_t = \beta m_{t-1} + (1-\beta) g_t^2$, %Moving average of the moving squared gradient
  $v_t = (1-\alpha) v_{t-1} + \alpha g_t$, %Moving average of the gradient
  $\hat{m}_t = m_t/(1-\beta^t)$, $\hat{v}_t = v_t/(1-\alpha^t)$, %Step correction
  $G = \mathrm{diag}(\frac{1}{\sqrt{\hat{m}_t}+\epsilon})$, %Preconditioned matrix
  $\theta_t \leftarrow \theta_{t-1} - \alpha G \hat{v}_t + \sqrt{2\alpha h G}\mathcal{N}(0, I)$ %Parameter updating
end while

5. Experiment. The proposed method, Adam-LD, is applicable to any network architecture with DIP property. To focus on the evaluation of sampling effectiveness, the same U-Net as DIP [86] is adopted for performance evaluation through all experiments. The DNN is a 5-layer auto-encoder with skip connections where each layer contains 128 channels. See [86] for more details on network architecture. Recall that the regularized squared ℓ_2-norm of the fitting error in the domain of measurement, normalized by the dimension of measurement, is used as the loss function for training the DNN:

$$L(\theta) = \frac{1}{M} \left\| \boldsymbol{y} - \Phi(f(\theta; \epsilon_0)) \right\|_2^2 + \lambda_{\mathrm{wd}} \left\| \theta \right\|_2^2, \tag{5.1}$$

where weight decay parameter $\lambda_{\mathrm{wd}} = \frac{\sigma_n^2}{\sigma_\theta^2 M}$ affects the final performance. They vary with the problem and the dataset. DIP [86] is as the baseline in this study. For DIP, the learning loss is given by

$$L_{\mathrm{DIP}}(\theta) = \frac{1}{M} \left\| \boldsymbol{y} - \Phi(f(\theta; \epsilon_0)) \right\|_2^2. \tag{5.2}$$

For both DIP and the sampling methods, we use the same strategy for network weights initialization. The weights are randomly initialized using the Xavier method, which is a default initialization technique in the PyTorch library. For DIP, the setting is the same as [86] where Adam is used for training with learning rate $1e-2$ and no weight decay is imposed. In addition, for noisy measurement, early stopping is needed in DIP for avoiding over-fitting, thus we terminate the iteration once the loss objective $L_{\mathrm{DIP}}(\theta)$ in (5.2) is less than $0.9h$, where h is the noise level. Due to the normalization in the loss objective, the optimal value for h is $h = 2\sigma_n^2/M$, where σ_n denotes the standard variance of measurement noise.

The proposed Adam-LD method is evaluated on two image reconstruction problems: linear sparse-view CT reconstruction problem and nonlinear phase retrieval problem. For the

experimental data, we used the CT100 dataset [27] for CT image reconstruction and Natural-6 and Unnatural-6 for phase retrieval [59]. CT100 dataset contains 100 real-world in-vivo CT images where the last 10 images are used for testing, and the remaining is used as training data for supervised learning methods. Using the same configuration in [22], the CT images are resized to 128×128 pixels and we apply Radon transform on them to generate the 50-view sinograms. For phase retrieval, there are six images in each Unnatural-6 and Natural-6 category set [59] and they are of size 256×256. We used three bipolar random masks to generate the coded diffraction patterns as the measurements in frequency domain. Both the noiseless and noisy measurements (corrupted by Gaussian white noise) are evaluated in the experiments. For noisy measurement, its noise level is measured by signal-to-noise ratio (SNR). The larger the SNR is, the less noise the measurement contains, and SNR$= \infty$ for noiseless measurements. For noisy measurement $\boldsymbol{y} = \Phi(\boldsymbol{x}) + \boldsymbol{n}$, the Gaussian white noise

$$\boldsymbol{n} = \boldsymbol{n}' \left\| \boldsymbol{y}^{\text{noiseless}} \right\|_2 / \left\| \boldsymbol{n}' \right\|_2 / \sqrt{10^{\text{SNR}/10}}, \quad \boldsymbol{n}' \text{ is Gaussian white noise vector,}$$

is added to noiseless measurement $\boldsymbol{y}^{\text{noiseless}}$ to yield to noisy measurement. For CT, three noise levels $30, 40, 50, 60, 70$ in SNR are evaluated. For phase retrieval, three SNR levels $10, 15$ and 20 are considered.

Other parameters. Except for the investigation of the convergence analysis, in the experiments, the iteration number for sparse-view CT is set to $T = 20,000$, the burn-in iteration is set to $T' = 14,000$. For phase retrieval, the maximum iteration T is set to 5000 and burn-in iteration number $T' = 2000$.

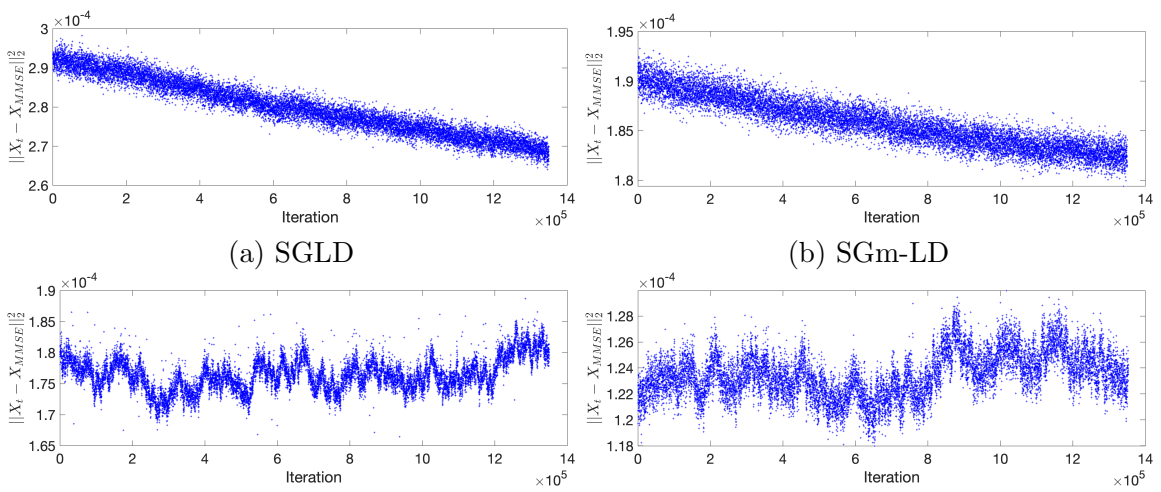
For the noisy sparse-view CT experiment, the weight decay is set to $1e-6$, and $\alpha = 1e-2$ and $\beta = 1e-2$. For the noisy phase retrieval experiment, the weight decay varies with the dataset and the noise level. Specially, for Unnatural-6 dataset, weight decay is set to $1e-4$, $1e-5$, $1e-6$ for noise levels $10, 15, 20$ respectively. For Natural-6 dataset, weight decay is set to $1e-3$, $1e-4$, $1e-5$ for noise levels $10, 15, 20$ respectively. The hyperparameters α and β are still set to $1e-2$. For the noiseless case for both experiments, the weight decay is set to zero. Langevin dynamics requires to set the scaling parameter h proportional to the noise level of the measurements. In the experiments, we use the noisy measurement and its SNR level to set the hyper-parameter $h = \frac{2\|\boldsymbol{y}\|_2^2}{M^2 10^{\text{SNR}/10}}$. For noiseless case, we set $h = 0$.

Computing machine specifications. We implemented the algorithms in Python, using the PyTorch library. Our experiments were conducted on a server equipped with an Intel(R) Xeon(R) Gold 6246 CPU and a NVIDIA TITAN RTX GPU. For the sparse-view CT application, with 20,000 iterations, the algorithm took 1,663 seconds to process an image of size 128×128. In the case of phase retrieval with 5,000 iterations, the running time is 282 seconds for an image of size 256×256.

5.1. Study on statistical properties of the different samplers . When applying the sampling algorithm to inverse problems, it is important to examine how the state space is randomly explored. In pursuit of this objective, we present a partial empirical study on the convergence behavior of four samplers: SGLD, SGm-LD, p-LD, and Adam-LD. We subject these samplers to a comparative analysis under the same computational procedure and employing the same neural network (NN).

To check the exploration ability of the Markov chain, we run the four sampling methods for the sparse-view CT experiment over a long time. The experiment is conducted on sparse-view CT image reconstruction for one sample image (img_98) with (SNR=60dB). We run the four algorithms for 3.85×10^6 iterations, where the first 2.5×10^6 iteration as the burn-in phase. Only the $1.35e^6$ samples after the burn-in period is taken to compute the MMSE estimation.

For high-dimensional image estimation problems, such as those involving images of size 256×256, computing multivariate autocorrelation functions (ACFs) of the Markov chain to study correlations between samples is not feasible. Therefore, following the approach in [50], we tracked only the evolution of the mean squared error (MSE) between the MMSE $\boldsymbol{x}_{\text{MMSE}}$, computed using the 100,000 samples after the burn-in phase, and the samples generated by $\boldsymbol{x}_i = f(\theta_i; \epsilon_0)$. This provides a partial view of the correlations between the samples. Figure 5.1 shows the MSE Euclidean distance between the final MMSE estimate and the generated samples of the chain for the four sampling methods, including SGLD, SGm-LD, p-LD and Adam-LD. It is observed that the samples fluctuate around the average values. The presence of the structure of the MSE distance shows that the convergence is not achieved. The MSE are decreasing for the SGLD and SGm-LD methods, which indicates that the samples are not close to the stationary regime. The MSE plot of p-LD and Adam-LD exhibits more unstructured, which means the samples are almost uncorrelated. Compared to p-LD, the concentration belt of the samples from Adam-LD is narrower, which shows that the proposed Adam-LD converges faster.



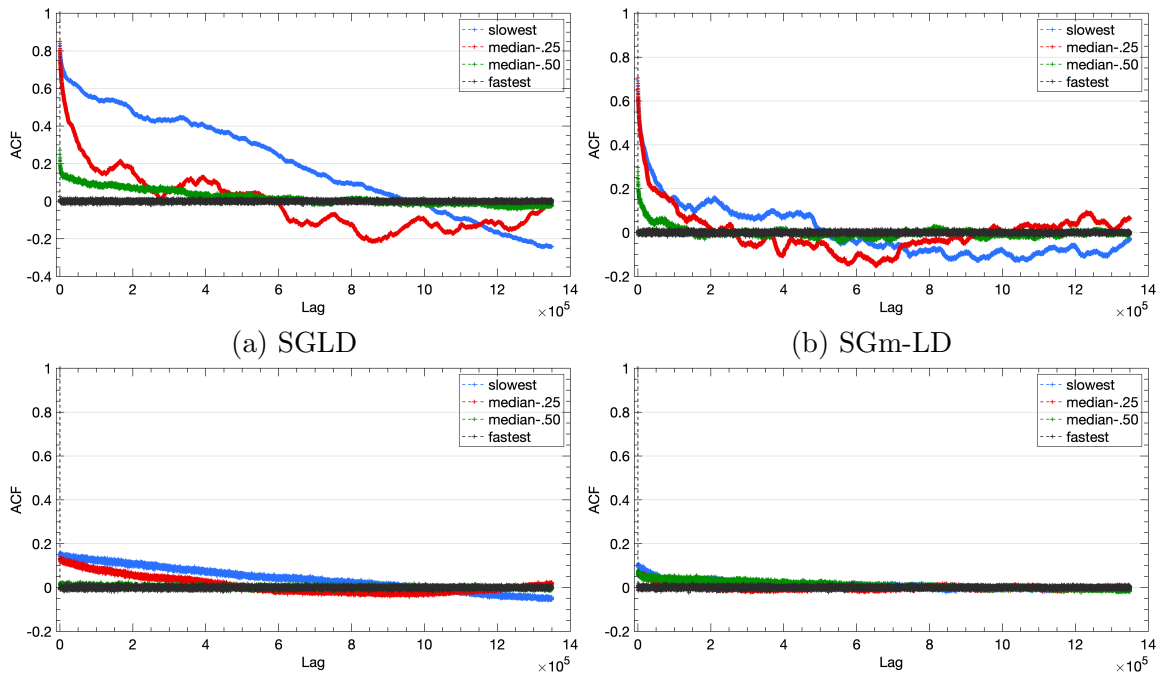
(a) SGLD (b) SGm-LD

(c) p-LD (d) Adam-LD

Figure 5.1. *Evolution of the MSE (Mean Squared Error) distance between the final MMSE estimate and the samples generated by the four samplers for the sparse-view CT application after the burn-in period. The samples randomly oscillate around the average values means that they are uncorrelated. The samples from p-LD and Adam-LD are almost uncorrelated.*

ACFs in Fourier domain. Although the posterior covariance matrix is intractable in our problem, we focus on the approximations of the posterior covariance of the reconstructed images using the Fourier frequency. We depict the sample ACFs associated with the sparse-view CT experiment. This is achieved by tracking the variance statistics of the magnitude

of Fourier coefficient along the iteration. We sort the final empirical marginal variances of the reconstruction in ascending order, and locate the indices corresponding to the smallest values, the values at the first quarter position, the values at the half position, and the largest values. Then we investigate the autocorrelation function (ACF) of the one-dimensional array, whose elements are the magnitude of the Fourier coefficient of the generated samples at a certain position. The slowest and the fastest converging direction correspond to the Fourier coefficients with the lowest and the highest marginal empirical variances. The ACFs for other two positions are called the median converging directions. More specially, we denote the median converging directions median-.25, and median-.50 respectively. See Figure 5.2 for the sample ACFs in the Fourier domain. For the four sampling methods, the ACF along the fastest direction vanishes, which indicates that the independence has achieved after the burn-in period. For Adam-LD, the ACF along the slowest direction vanishes quickly after the burn-in period. However, the slowest converging directions for SGLD, SGm-LD has a decreasing curves with oscillation, which indicates that the Markov chain does not achieve the stationary regime after the burn-in period. The comparison demonstrates the efficiency advantage of the Adam-LD for sampling the neural network.



(a) SGLD (b) SGm-LD

(c) p-LD (d) Adam-LD

Figure 5.2. *ACFs for the sparse-view CT experiment. The ACFs are shown for lags up to* 1.35e6 *for the four samplers. Independence is not achieved along the two median directions and the slowest direction (correspond to the final empirical marginal variances of the samples in the Fourier domain). The comparison demonstrates the efficiency advantage of the Adam-LD.*

Evolution of the empirical standard deviation. To the best of our knowledge, there is also lack of full investigation on the convergence of second-order moment statistics of Langevin dynamics-based method for the nonconvex learning. We only depict the evolution of the

empirical standard deviation in the revision by setting a long running time, i.e., $3.85e^6$ iteration with $2.5e6$ burn-in iteration. See Figure 5.3 the evolution of the RMSE (Root Mean Squared Error) between the current empirical standard deviation computed using all the generated samples from the burn-in iteration to the current iteration and the final empirical standard deviation. The decreasing curve demonstrates that the convergence of second order moment of the sampling method is not achieved with the running iterations.

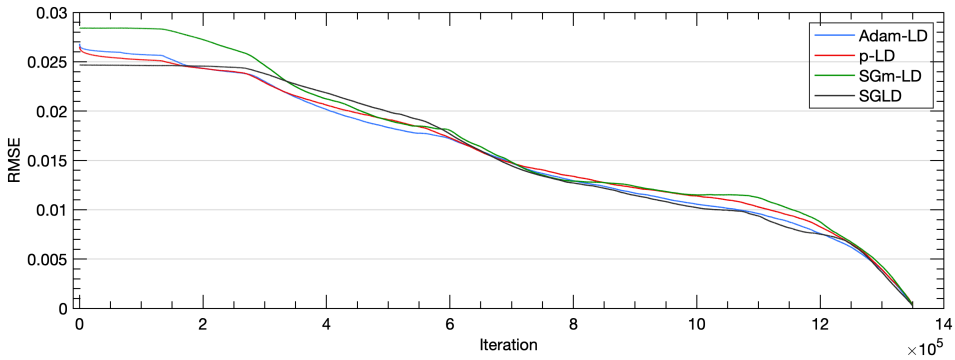


Figure 5.3. *Evolution of the RMSE (Root Mean Squared Error) between the final standard deviation and the estimated current standard deviation for the sparse-view CT experiment. The decreasing curves shows that the convergence of the second order moment is not achieved.*

5.1.1. Ablation study.

Setting of the method with different samplers. For SGLD and SGm-LD, we set the learning $\eta = 1e-3$ to avoid exploding loss. The α is set to 0.9 for SGm-LD borrowed from the literature [84]. For p-LD and Adam-LD, we set the $\beta = 0.99$ and $\epsilon = 1e-8$ by the default values in the PyTorch library. For other hyper-parameter, the learning rate η, α are set to $1e-2$ for p-LD and Adam-LD respectively.

Trade-off between reconstruction performance and computational cost. In this experiment, we study the reconstruction performance with respect to the number of iterations. Note that the setting of maximum iteration to $2e4$ in our experiment can not lead to the convergence to the stationary, and the MMSE is computed with a biased distribution.

In the remaining experiments, the total iterations setting is the same as before. The iteration number for sparse-view CT is set to $T = 20,000$, the burn-in iteration is set to $T' = 14,000$. For phase retrieval, the maximum iteration T is set to 5000 and burn-in iteration number $T' = 2000$. Despite the insufficient iteration numbers to maximize the reconstruction performance, the proposed method still provided a noticeable performance gain over existing methods.

If the computational cost permits, the performance with more iterations can be boosted. For example, when processing a single image in the sparse-view CT application with 1,200,000 iterations, the PSNR of the result is 39.57dB. In comparison, the PSNR of the result from 20,000 iterations is 36.31dB. However, it took 27 hours to run 1,200,000 iterations, whose computational cost is overwhelming in practice. To meet practical need, we make a trade-off between reconstruction performance and computational cost.

Efficiency and effectiveness of Adam-LD MC sampler. Two image reconstruction problems are set as the following: CT reconstruction with noisy measurement (SNR= 50) and phase retrieval with noisy measurement (SNR= 10). See Figure 5.4 for the comparison of different methods in terms of objective loss value and quantitative PSNR value with respect to iteration number. Here we does not recall the sample average in the MMSE estimation. It can be seen that for CT image reconstruction, Adam-LD is the one with the fastest decreasing speed among all Langevin MC methods, in terms of both loss value and reconstruction error. The decay of loss value of Adam-LD is slower than the DIP which calls Adam optimizer. However, the DIP suffers from overfitting as its reconstruction error increases after around 2000 iterations. In contrast, Adam-LD does not suffer from over-fitting, as its reconstruction error does not increase with more iterations. Note that there is a sharp decrease of the PSNR in the early stage. It may either due to abrupt mode transition of the Langevin dynamics iteration or due to insufficient burn-in time.

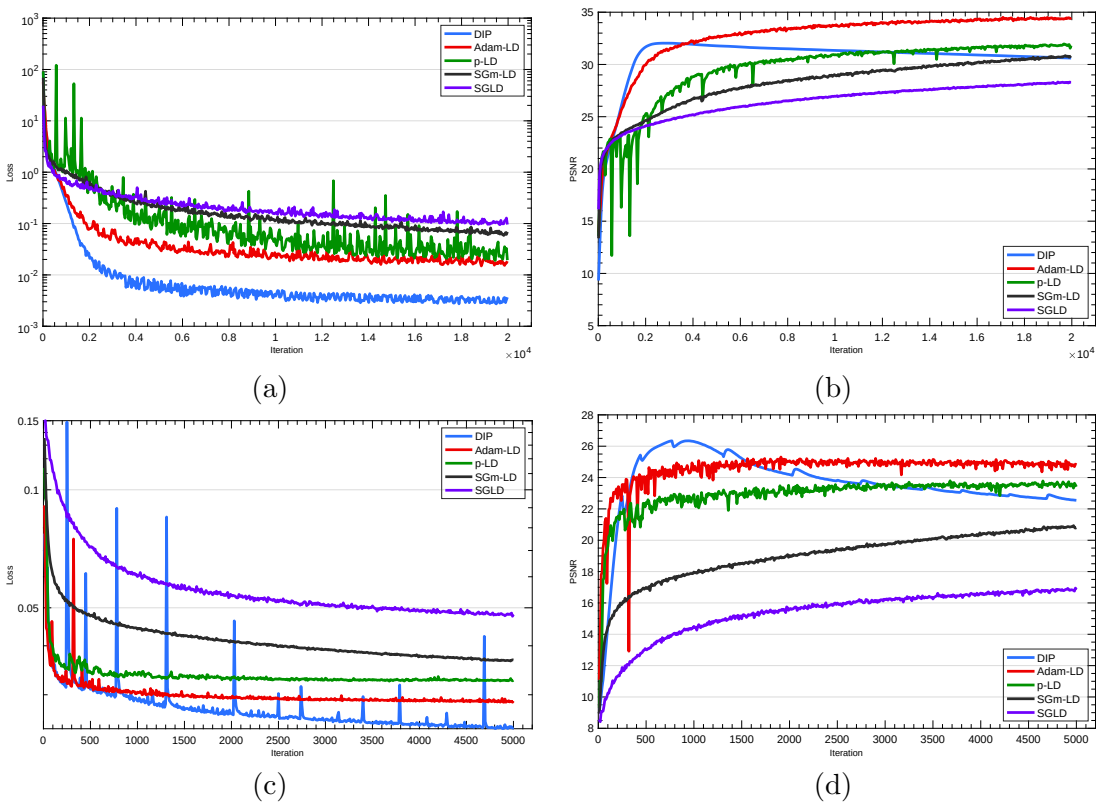


Figure 5.4. *Comparison of the evolution of the PSNR metric of the individual samples from different Langevin MC samplers for two image reconstruction problems. Top row is for CT and Bottom row for phase retrieval. (a) and (c) Loss value over iteration; (b) and (d) reconstruction accuracy in PSNR over iteration.*

Performance comparison of different Langevin MC samplers over the dataset. In this experiment, the comparison of the MMSE estimate using the four different Langevin MC samplers is conducted over the datasets for two image reconstruction tasks. See Table 5.1 for the comparison of quantitative performance among different samplers. It can be seen that the

proposed Adam-LD is clearly the best performer for both tasks, and it outperforms the other 3 samplers by a large margin. This clearly indicates the advantage of Adam-LD over other MC samplers when being used for sampling network weights to approximate MMSE estimate. In other words, the usage and momentum and the precondition in the Adam-LD is very helpful for efficient sampling in the context of this paper.

Table 5.1

Comparison of the MMSE with different Langevin dynamics sampling methods applied to sparse-view CT and phase retrieval. These sampling methods run $2e^4$ *iterations with* 14000 *burn-in iteration for sparse-view CT application. They run* 5000 *iterations with* 2000 *burn-in iteration for phase retrieval.*

Task	Dataset	SNR	SGLD	SGm-LD	p-LD	Adam-LD
CT	CT100	30	27.34	28.13	28.78	**28.94**
		40	27.94	30.66	32.02	**32.46**
		50	29.83	32.33	34.47	**36.83**
		60	29.89	32.34	34.76	**38.58**
		70	29.96	32.37	34.77	**39.04**
		∞	29.98	32.48	34.93	**39.26**
PR	Unnat.-6	10	22.09	26.00	30.24	**31.23**
		15	22.12	26.05	33.68	**34.29**
		20	22.15	26.15	36.69	**37.48**
		∞	22.16	26.15	42.33	**43.16**
PR	Natural-6	10	19.60	23.07	24.57	**25.91**
		15	19.70	23.25	29.42	**29.86**
		20	20.10	24.01	32.06	**32.76**
		∞	20.13	24.02	41.38	**44.24**

5.2. Sparse-view CT image reconstruction. X-ray computed tomography (CT) imaging is an important application in medical imaging. The measurement is a collection of the discrete line integral (*i.e.* Radon transform) along a set of scanning lines. The measurement $\boldsymbol{y}$, also called sinograms, are related to the underlying image $\boldsymbol{x}$ of internal organs by:

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

where $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ denotes the corresponding Radon transform. In this experiment, we concern the image reconstruction problem in sparse-view CT. Sparse-view CT aims at reduce the number of measured projections for several benefits in practice, such as reducing patient dose and reducing scan time. In comparison to traditional CT, image reconstruction for sparse-view CT is more challenging due to fewer measurement available for estimating the image.

For experimental data of sparse-view CT imaging, the implementation of Radon transformation is from the open-sourced code [1]. As the focus of this paper is on dataset-free methods for image reconstruction, the performance evaluation of the proposed Adam-LD

[1] https://github.com/matteo-ronchetti/torch-radon

method focuses on the comparison to other dataset-free methods, including both traditional regularization methods and self-supervised or unsupervised deep learning methods.

The following additional methods are included: classic Filter back projection (FBP) method with ramp filter, TV regularization method [2], four self-supervised learning methods: DIP, SBNN [65]), and (R)EI [23]. In addition, the supervised learning method [22] using the same UNet architecture as [23], denoted as S-EI, is included for showing the performance gap between supervised and unsupervised learning methods. As a baseline, we also include the results of Adam for direct minimization of $L(\theta)$ in (5.1), without employing sampling or averaging. It is worth noting that overfitting is observed for Adam, similar to DIP. To address this concern, we employ the early stopping strategy for Adam, similar to DIP, to report the quantitative results.

The impact of weight decay parameter to the results. In this section, we conduct an experiment to demonstrate the impact of different weight decay parameters on the overall performance. For this purpose, we test the weight decay parameters from a set $1E+0, \ldots, 1E-9$ at three different SNR levels (40, 60, and 70) in the context of the sparse-view CT experiment. The comparison of performance with different choices of the weight decay parameter is presented in Table 5.2. The experiments showed that the best performance is consistently achieved when the parameter value is set to $1E-6$ for the sparse-view CT experiment, regardless of the SNR levels.

Table 5.2
Performance of Adam-LD for sparse-view CT with different weight decay parameters

SNR	1E+00	1E-01	1E-02	1E-03	1E-04	1E-05	1E-06	1E-07	1E-08	1E-09
40	25.47	28.19	30.35	31.52	32.17	32.40	32.46	32.41	32.32	32.32
60	26.40	30.67	33.08	35.81	37.49	37.76	38.58	38.15	38.22	38.15
70	26.69	30.43	33.58	36.68	38.73	38.78	39.04	38.70	38.77	38.70

Quantitative and visual results. See Table 5.3 for the performance comparison between Adam-LD and other methods for image reconstruction of sparse-view CT. Clearly, the proposed Adam-LD outperformed all other dataset-free methods by a noticeable margin. The advantage of the proposed Adam-LD method in quantitative metric is also consistent with its advantage in visual quality. See Figure 5.5 for visual comparison of an exemplary instance with zoomed-in regions. See Figures 5.6 and 5.7 for visual comparison of more reconstructed results from different methods. Upon visualizing the reconstructions, it becomes evident that the proposed Adam-LD method yields the best results, featuring sharper edges. Conversely, the images reconstructed by REI, DIP, and SBNN exhibit oversmoothing effect in their results. This observation aligns with the fact that neural networks trained using the ℓ_2 loss tend to promote smoothed images [95, 87]. Regarding the FBP reconstruction, it is afflicted by amplified noise artifacts, which is a well-known issue for FBP. On the other hand, the TV solution suffers from staircasing issues, stemming from oversimplified assumptions about the

[2]The regularization parameter is set by the non-Bayesian discrepancy principle [85]. Here TV method only performs MAP estimation and not MMSE estimation. There are indeed MMSE restoration method involving TV regularization imaging problems fusing proximal MCMC sampling methods, such as [57, 34]

anisotropic total variation prior [3] concerning real-world images.

Table 5.3

Average PSNR(dB) of the results from different methods for CT image construction w.r.t. different SNR. Adam-LD with $\alpha = \beta = 1e-2$ *runs* $2e^4$ *iterations with* 14000 *burn-in iteration.*

Method	Regularization		Supervised	(Un)Self-supervised				
SNR	FBP [64]	TV [19]	S-EI [22]	DIP [86]	SBNN [65]	(R)EI [23]	Adam	Adam-LD
30	13.35	26.14	26.57	24.63	24.02	25.51	27.04	**28.94**
40	21.42	29.79	29.53	29.11	29.29	29.36	31.91	**32.46**
50	29.54	34.67	33.28	32.54	30.31	32.61	35.69	**36.83**
60	30.16	35.55	37.04	33.27	30.22	36.56	36.43	**38.58**
70	30.23	35.65	37.52	32.76	30.03	36.80	37.65	**39.04**
∞	30.24	35.66	38.17	34.76	33.87	36.94	37.67	**39.26**

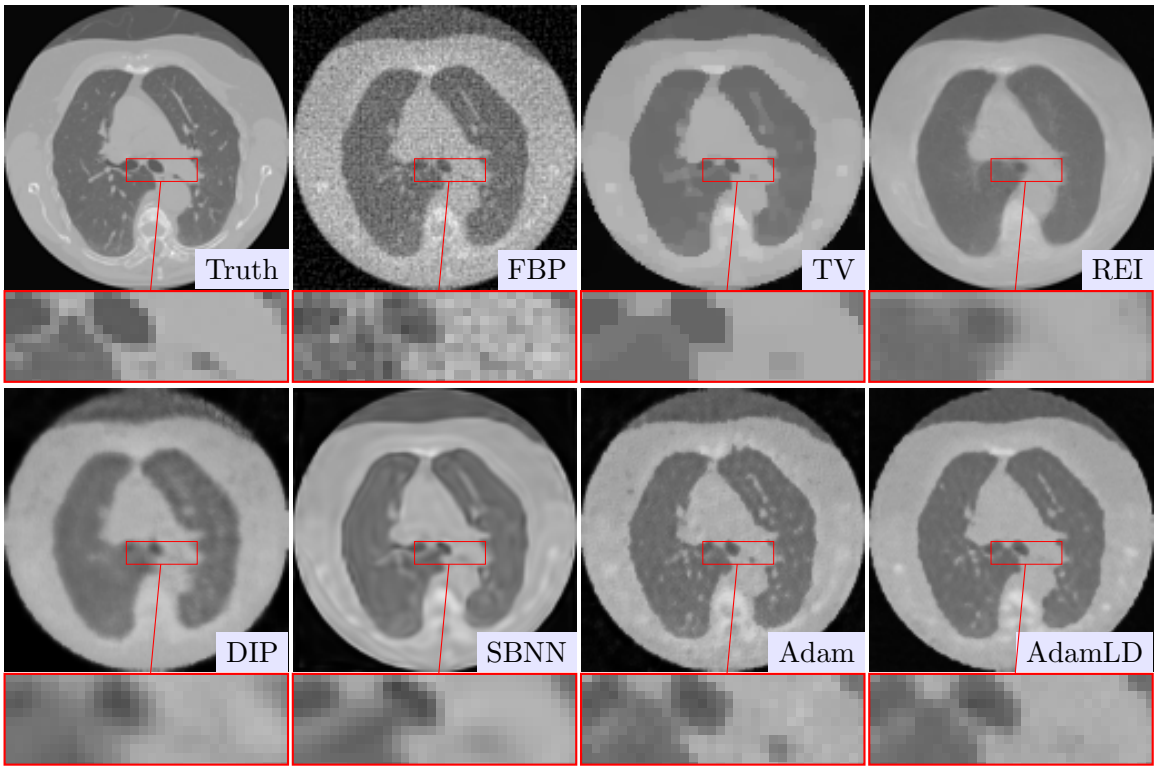


Figure 5.5. *Reconstruction of sparse-view CT using Gaussian noisy observation (SNR* $= 40$*) for 'img_90'. Adam-LD with* $\alpha = \beta = 1e-2$ *runs* $2e^4$ *iterations with* 14000 *burn-in iteration. MMSE from Adam-LD produce a noiseless restoration with sharper edges.*

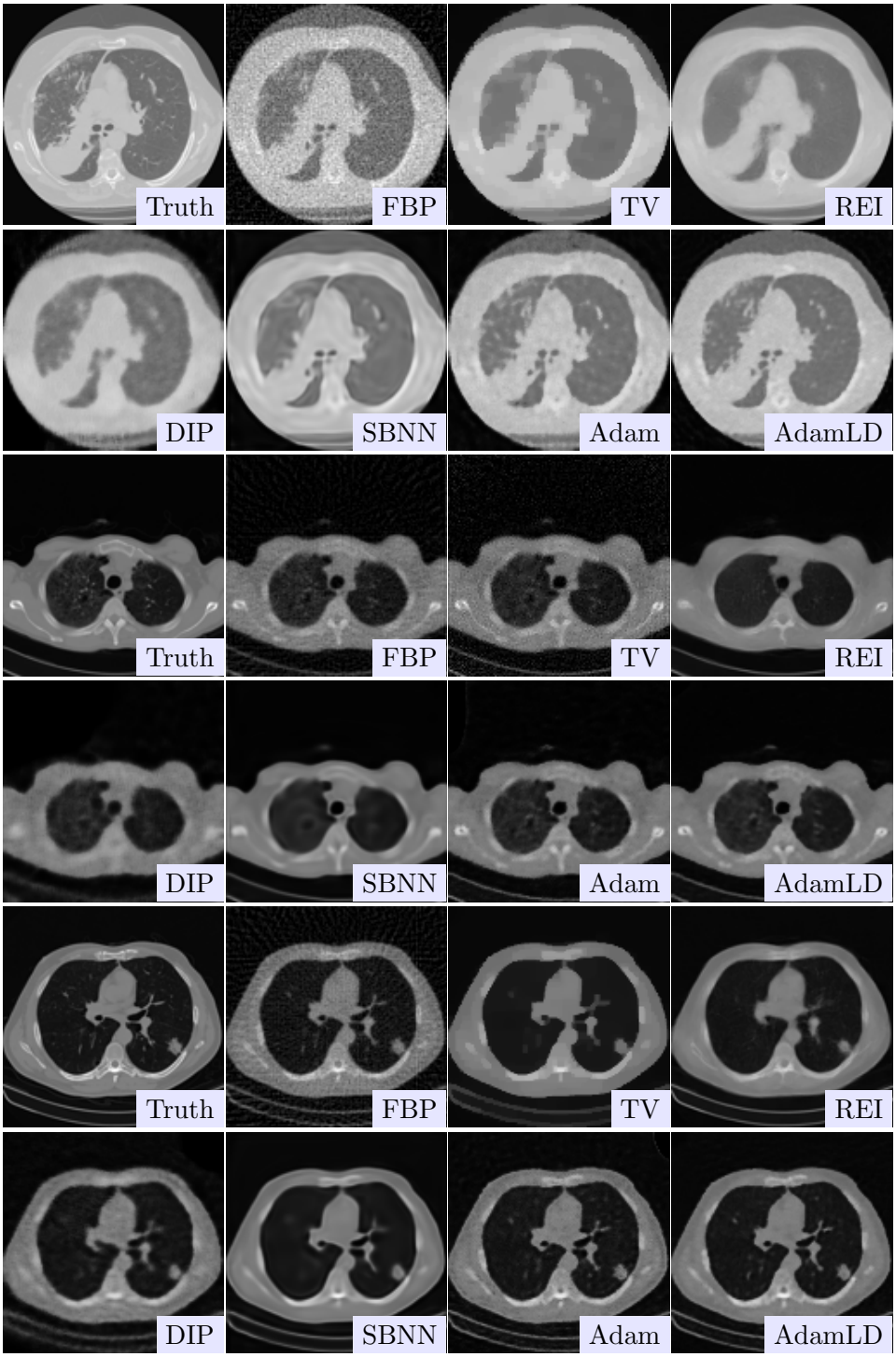


Figure 5.6. *Reconstruction of sparse-view CT using Gaussian noisy observation (SNR* $= 40$*). Adam-LD with* $\alpha = \beta = 1e{-}2$ *runs* $2e^4$ *iterations with* 14000 *burn-in iteration. MMSE from Adam-LD produce a noiseless restoration with sharper edges.*

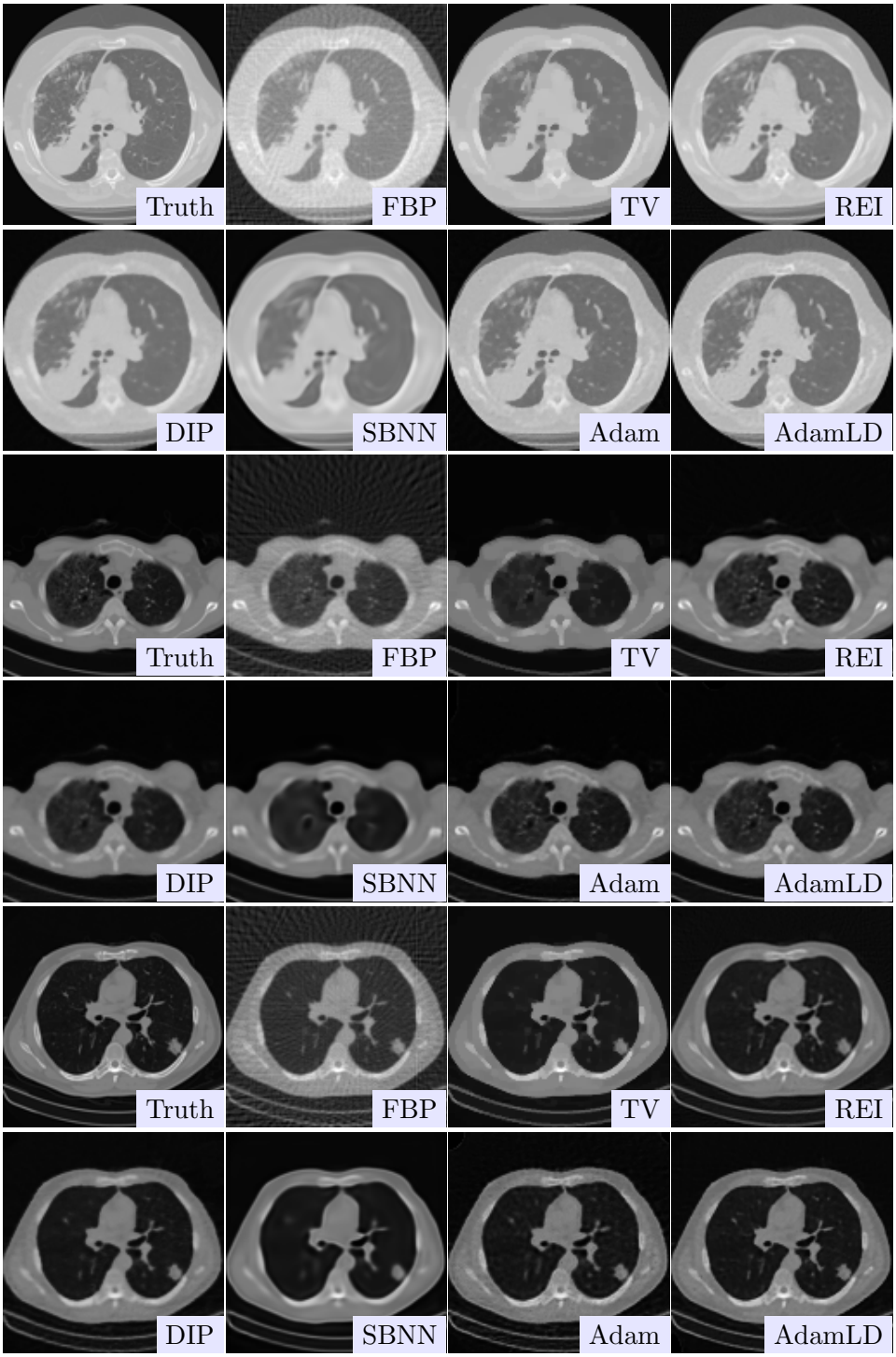


Figure 5.7. *Reconstruction of sparse-view CT using Gaussian noisy observation (SNR* $= 70$*). Adam-LD with* $\alpha = \beta = 1e{-}2$ *runs* $2e^4$ *iterations with* 14000 *burn-in iteration. MMSE from Adam-LD produce a noiseless restoration with sharper edges.*

5.3. Phase retrieval. Phase retrieval is an important imaging technique in scientific imaging applications, which reconstructs an image from the magnitude of its measurements in Fourier domain. It needs to solve an ill-posed non-linear problem:

$$\boldsymbol{b} = |\boldsymbol{A}\boldsymbol{x}| + \boldsymbol{n},$$

where $|\cdot|$ denotes absolute value and $\boldsymbol{A}$ denotes a sensing matrix composed of Discrete Fourier transform and bipolar (or uniform) random masks; See more details in [59]. Following the same setting as [59], we set the noise level of the measurement data in terms of its SNR. Note that the absolution operator in phase retrieval is not differentiable everywhere. But the set of points at which $L(\theta)$ or $L_{\text{DIP}}(\theta)$ is not differentiable is a zero probability measure. The gradient can be computed using the PyTorch library, where the subgradient is used when applicable. Same as the experiments on sparse-view CT image reconstruction, the comparison includes other existing works on phase retrieval: non-learning Wirtinger flow (WF) method [17], two plug-and-play deep learning method: prDeep [59], prGMAP [60] and three self-supervised learning methods: DIP [86], SBNN [65] and the baseline Adam which directly minimizing the objective.

The impact of weight decay parameter to the results. In this section, we conduct an experiment to demonstrate the impact of different weight decay parameters on the overall performance. For this purpose, we test the weight decay parameters from a set $\{1E+0, \ldots, 1E-9\}$ at three SNR levels (*i.e.*, 10, 15 and 20) for phase retrieval experiment on atural-6 dataset. Please refer to Table 5.4 for a comparison of performance using different values of the weight decay parameters. The optimal weight decay for phase retrieval varies depending on the noise levels. In other words, finding the optimal weight decay setting requires prior knowledge of the noise level in the data.

Table 5.4
Performance of Adam-LD for phase retrieval with different weight decay parameters

	SNR	1E+00	1E-01	1E-02	1E-03	1E-04	1E-05	1E-06	1E-07	1E-08	1E-09
Nat.-6	10	15.14	15.17	15.18	25.91	24.01	20.67	19.01	18.45	18.44	18.35
	15	15.45	15.49	15.50	25.00	29.86	28.30	25.89	24.68	24.55	24.55
	20	15.26	15.30	15.31	25.15	30.92	32.76	31.86	30.88	30.50	30.47

Quantitative and visual results. See Table 5.5 for a quantitative comparison of different methods over two datasets. The Wirtinger flow, in fact, optimizes the nonlinear least squares objective in the pixel domain using the measurement data. In scenarios with noiseless data, this nonconvex objective exhibits a benign loss landscape, making optimization towards global minima less problematic [81]. Since the gradient flow occurs in the pixel domain, the algorithm converges much faster, resulting in good performance. Thus, for noiseless data, the gradient descent (Wirtinger flow) method is the best performer and the proposed Adam-LD is the second best. However, for noisy data, Adam-LD is the best performer with a large gap to the second best performer. The absence of regularization in the Wirtinger flow leads to inferior results when dealing with noisy data. This indicates practical value of the proposed Adam-LD method as noise is unavoidable in real-world data. See Figure 5.8 for visual comparison

of an exemplary instance with zoomed-in regions. See Figures 5.9 and 5.10 for visualization comparison of sample data from noisy measurements. It can be seen that the advantage of the proposed Adam-LD method remains in terms of visual quality. Due to the absence of proper regularization, the WF results in erroneous reconstructions, except in the noiseless case. Both the DIP and Adam methods produce erroneous reconstructions, indicating that the early stopping strategy does not yield a satisfactory result. On the other hand, the plug-and-play PrDeep approach leads to oversmoothed reconstructions, likely due to the utilization of a generic CNN-based denoiser. Similarly, the SBNN algorithm also produces oversmoothed reconstructions, similar to PrDeep. Furthermore, the reconstructions from SBNN exhibit erroneous artifacts around the silent edge.

Table 5.5

Average PSNR(dB) value of the results of different phase retrieval methods w.r.t. different SNR. Adam-LD with $\alpha = \beta = 1e-2$ *runs* 5000 *iterations with* 2000 *burn-in iteration.*

Method		GD	plug-and-play		(Un)Self-supervised learning			
Dataset	SNR	WF [17]	prDeep [59]	prGAMP [60]	DIP [86]	SBNN [65]	Adam	Adam-LD
Unnat.-6	10	20.37	30.2	30.04	28.71	29.99	28.26	**31.23**
	15	26.18	32.13	32.88	32.08	31.89	31.34	**34.29**
	20	31.47	35.44	35.87	32.49	32.22	35.05	**37.48**
	∞	**53.32**	51.13	50.76	41.24	41.86	41.94	43.16
Natural-6	10	15.33	26.11	25.38	24.54	24.65	23.68	**25.91**
	15	21.12	28.79	28.23	28.59	29.52	26.35	**29.86**
	20	26.42	31.33	30.97	31.17	30.19	29.79	**32.76**
	∞	**50.99**	46.38	46.01	41.96	38.09	39.14	44.24

5.4. The marginal posterior standard deviation of the inference. The proposed sampler methods provide samples that allow us to calculate the marginal posterior standard deviation, measured by the variance of these samples. We compared the marginal posterior standard deviation from the three samplers. In the experiments, SGLD and SGm-LD can not achieve the close-to-stationary regime with our iteration configuration. Thus for fair comparison, we set the MMSE estimate to the ground truth in the posterior standard variance computation for all three samplers. Please refer to Figures 5.11 and 5.12 for the visualization of the marginal posterior standard deviation estimation of the reconstruction of a sample for sparse-view CT and phase retrieval, respectively.

Note that there may exist a bias between the ground truth and the MMSE estimate associated to the target distribution. Moreover, a high standard deviation does not necessarily indicate a bad model or undesirable performance. Instead, it simply signifies a higher level of uncertainty in the predictions, which can be a valuable insight in certain scenarios. It is interesting to observe that the values of the uncertainty map, represented by the standard deviation, align qualitatively with the reconstruction quality. One notable observation is that, for all methods, regions with salient structures exhibit larger standard deviations than those

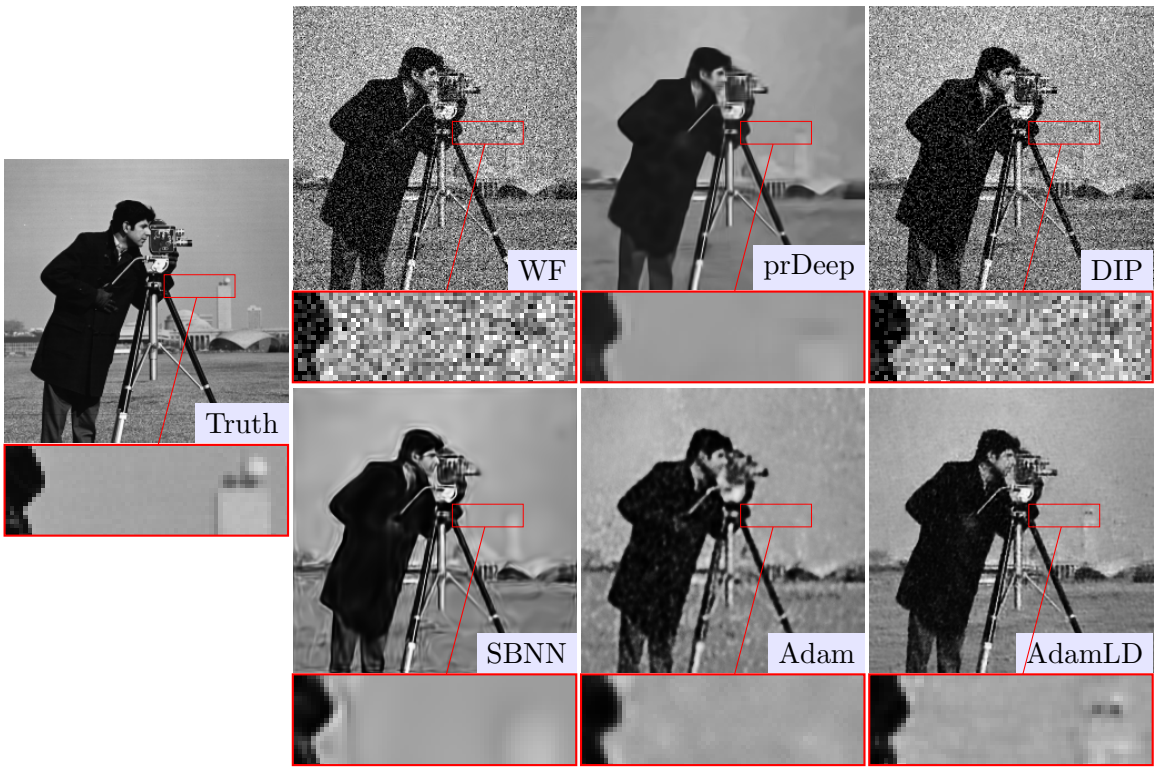


Figure 5.8. *Phase retrieval results with bipolar masks and sample data with SNR=10 for 'cameraman'. Adam-LD with* $\alpha = \beta = 1e-2$ *runs* 5000 *iterations with* 2000 *burn-in iteration.*

with smooth regions. This observation is consistent with a well-known fact in practice, smooth regions are easier to estimate than regions with salient structures in image reconstruction. The difficulty arises from distinguishing abrupt changes caused by salient edges or noise. Moreover, the results obtained from data with low noise levels exhibit smaller standard deviation values compared to those obtained from data with high noise levels. This observation is also in line with the fact that, for a given method, higher noise levels in the measurement generally lead to more erroneous results. Overall, the uncertainty map from the Adam-LD method shows the smallest standard deviation value among all methods under the same configuration. This suggests that the Adam-LD method is likely to provide more reliable and less uncertain reconstructions.

6. Conclusion and future work. This paper presented an self-supervised deep learning method for solving inverse problems in imaging. The basic idea is to re-parameterize an image by a deep network for exploiting the deep image prior and then derive an algorithm which is motivated by approximating its MMSE estimator via an MC sampling method. The key of such an approach is an effective MC sampler, and we proposed a Langevin MC method motivated by the idea behind Adam optimizer widely used in deep learning. The proposed method is applied to solve two image reconstruction problems: sparse-view CT image reconstruction and phase retrieval. The experiments showed that in most cases, the

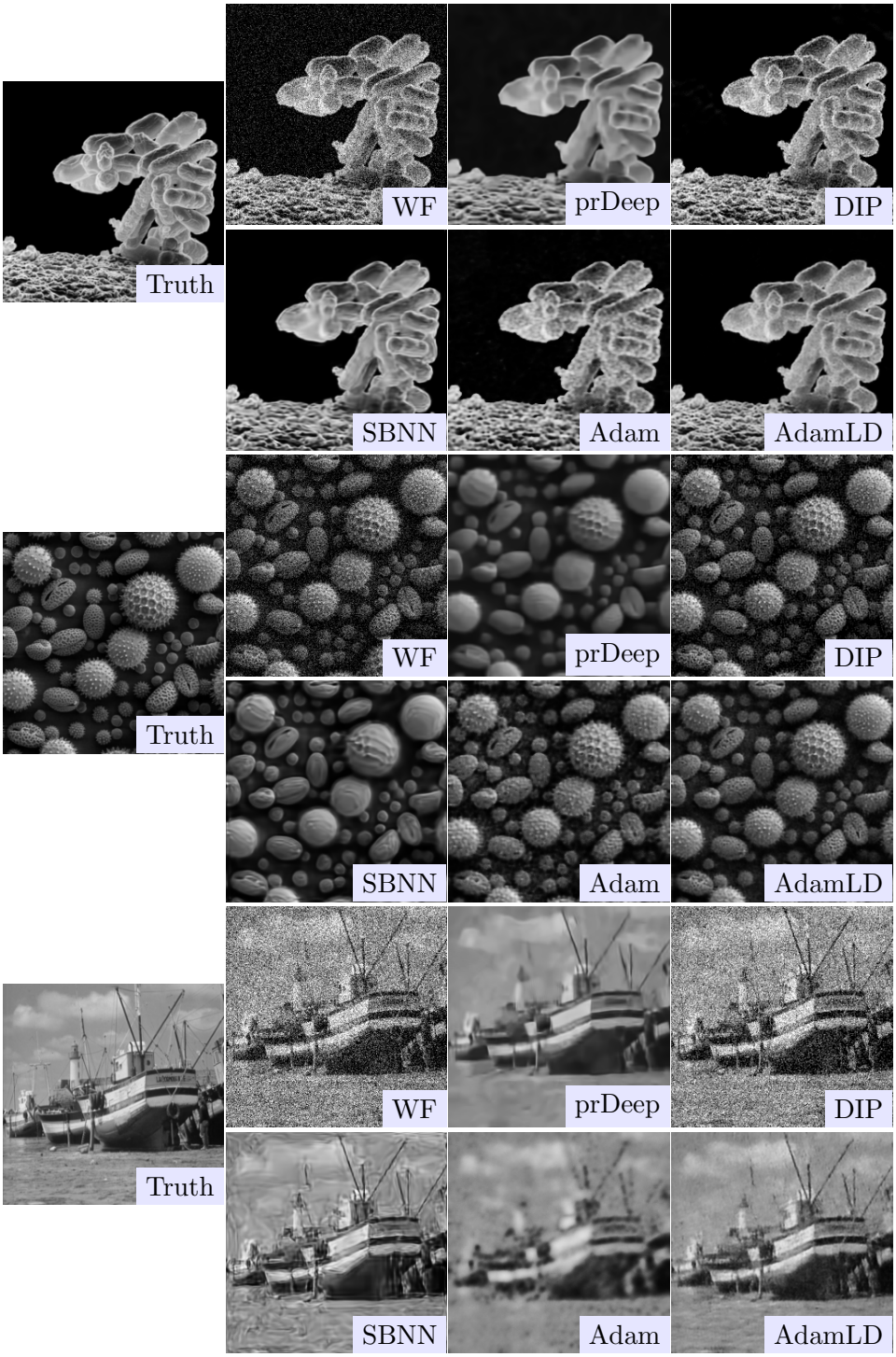


Figure 5.9. *Comparison of the reconstructions produced by different methods over the dataset [59] for the phase retrieval inverse problem with SNR* $= 10$*. Adam-LD with* $\alpha = \beta = 1e-2$ *runs* 5000 *iterations with* 2000 *burn-in iteration. A bipolar is mask is applied on the Fourier measurement.*

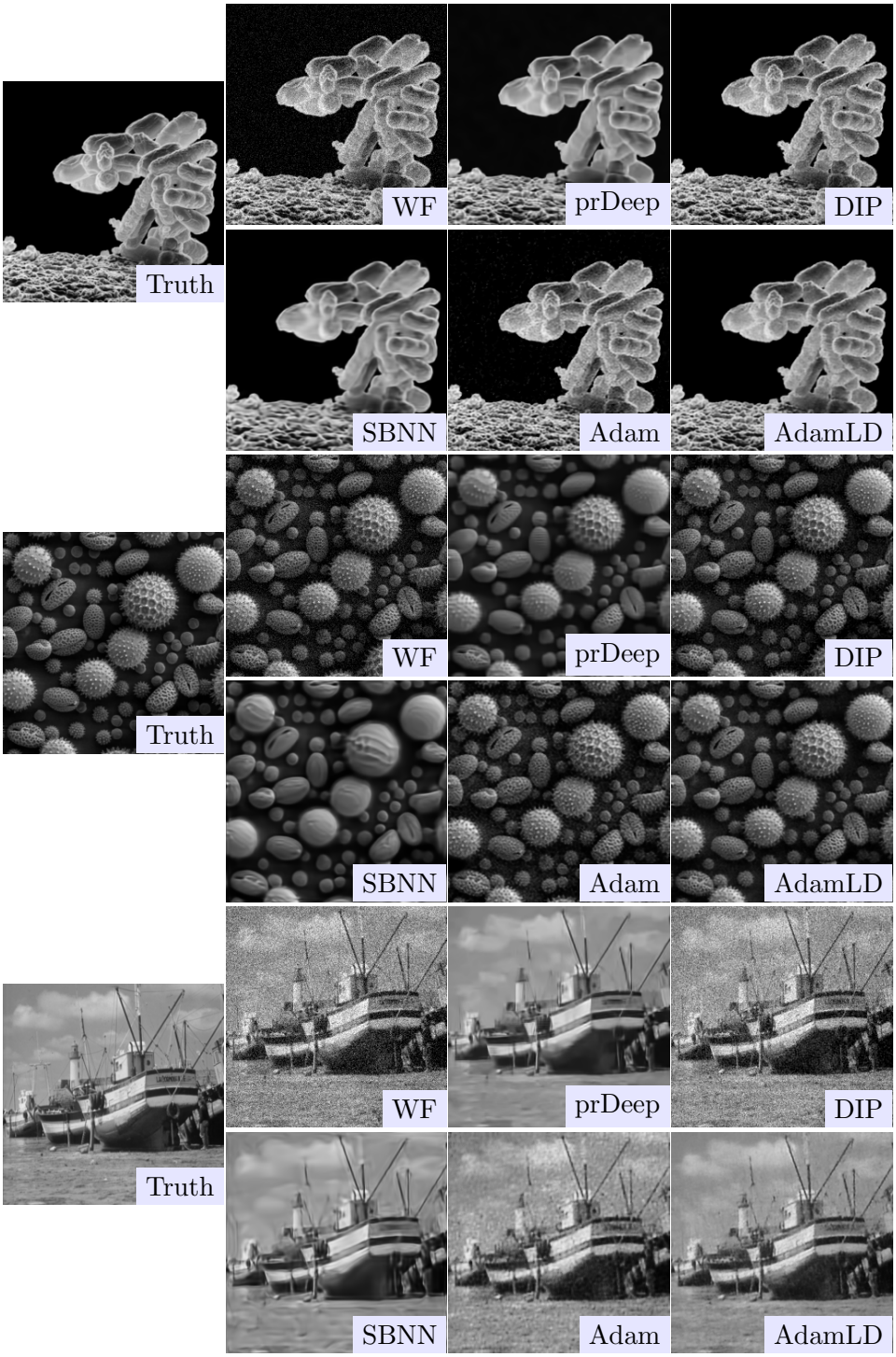


Figure 5.10. *Comparison of the reconstructions produced by different methods over the dataset [59] for the phase retrieval inverse problem with SNR* $= 15$*. Adam-LD with* $\alpha = \beta = 1e-2$ *runs* 5000 *iterations with* 2000 *burn-in iteration. A bipolar is mask is applied on the Fourier measurement.*

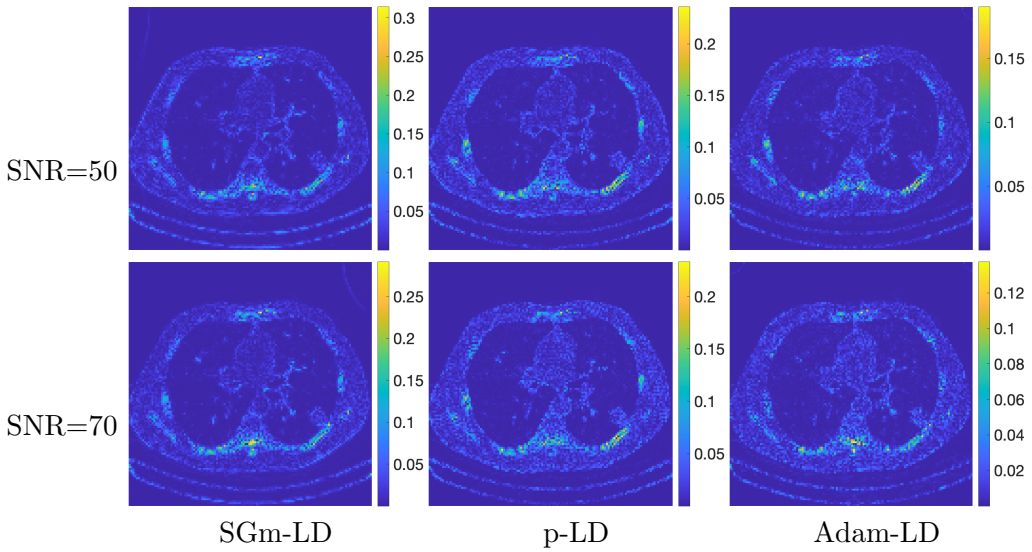


Figure 5.11. *The marginal posterior standard deviation of three samplers for sparse-view CT from a sample data with SNR*$=50, 70$ *respectively.*

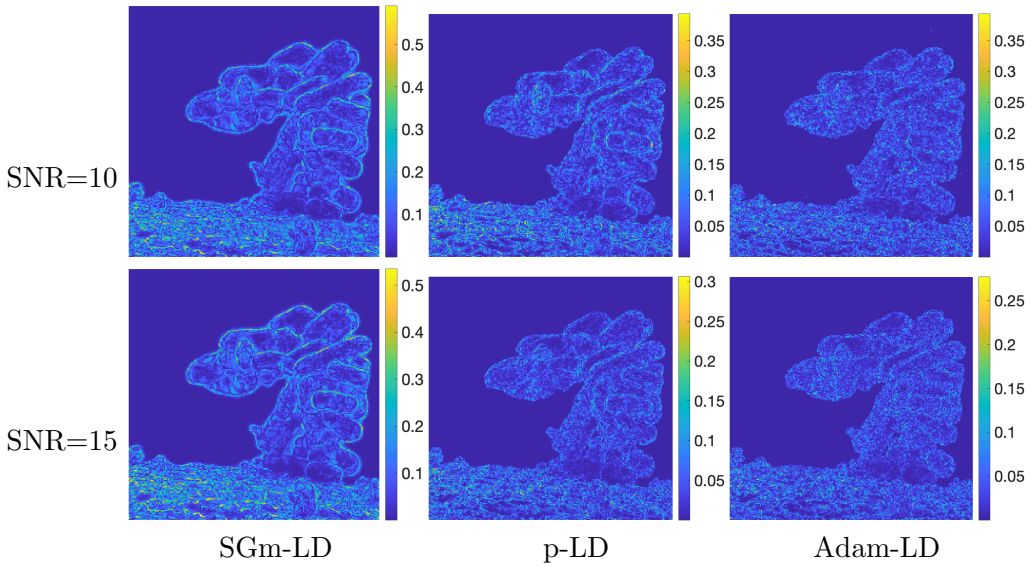


Figure 5.12. *The marginal posterior standard deviation of three samplers for phase retrieval from a sample data with SNR*$=10, 15$ *respectively.*

proposed self-supervised approach outperforms existing non-learning and dataset-free deep learning methods by a large margin.

One limitation of our proposed method is that the noise level needs to be provided in advance. In the future, we aim to study self-supervised deep learning methods for image reconstruction that are blind to the noise level, eliminating the need for its explicit specification. Additionally, it remains a challenging question how to derive an effective Langevin MC sam-

pling method that can provide accurate estimates for MMSE estimation, within reasonable computational cost, In the future, we will delve into investigating this question and exploring approaches to achieve more precise and reliable MMSE estimates through improved sampling techniques.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Transactions on Signal Processing, 54 (2006), pp. 4311–4322.

[2] G. Alain and Y. Bengio, *What regularized auto-encoders learn from the data-generating distribution*, Journal of Machine Learning Research, 15 (2014), pp. 3563–3593.

[3] F. Andreu, V. Caselles, J. I. Díaz, and J. M. Mazón, *Some qualitative properties for the total variation flow*, Journal of Functional Analysis, 188 (2002), pp. 516–547.

[4] M. Arjovsky and L. Bottou, *Towards principled methods for training generative adversarial networks*, in International Conference on Learning Representations, 2017.

[5] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, *Invertible generative models for inverse problems: mitigating representation error and dataset bias*, in International Conference on Machine Learning, PMLR, 2020, pp. 399–409.

[6] M. Asim, F. Shamshad, and A. Ahmed, *Blind image deconvolution using deep generative priors*, IEEE Transactions on Computational Imaging, 6 (2020), pp. 1493–1506.

[7] F. Bach, *Breaking the curse of dimensionality with convex neural networks*, The Journal of Machine Learning Research, 18 (2017), pp. 629–681.

[8] M. R. Banham and A. K. Katsaggelos, *Digital image restoration*, IEEE Signal Processing Magazine, 14 (1997), pp. 24–41.

[9] J. Batson and L. Royer, *Noise2self: blind denoising by self-supervision*, in International Conference on Machine Learning, PMLR, 2019, pp. 524–533.

[10] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Transactions on Image Processing, 18 (2009), pp. 2419–2434.

[11] C. A. Bhardwaj, *Adaptively preconditioned stochastic gradient Langevin dynamics*, in Workshop on Understanding and Improving Generalization in Deep Learning, ICML 2019, PMLR, 2019.

[12] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, *Compressed sensing using generative models*, in International Conference on Machine Learning, PMLR, 2017, pp. 537–546.

[13] E. Bostan, R. Heckel, M. Chen, M. Kellman, and L. Waller, *Deep phase decoder: self-calibrating phase microscopy with an untrained deep neural network*, Optica, 7 (2020), pp. 559–562.

[14] J. Cai, H. Ji, C. Liu, and Z. Shen, *Blind motion deblurring from a single image using sparse approximation*, in IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 104–111.

[15] J.-F. Cai, B. Dong, S. Osher, and Z. Shen, *Image restoration: total variation, wavelet frames, and beyond*, Journal of the American Mathematical Society, 25 (2012), pp. 1033–1089.

[16] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, *Data-driven tight frame construction and image denoising*, Applied and Computational Harmonic Analysis, 37 (2014), pp. 89–105.

[17] E. J. Candes, X. Li, and M. Soltanolkotabi, *Phase retrieval via Wirtinger flow: theory and algorithms*, IEEE Transactions on Information Theory, 61 (2015), pp. 1985–2007.

[18] S. V. Chakhlov, S. P. Osipov, A. K. Temnik, and V. A. Udod, *The current state and prospects of X-ray computational tomography*, Russian Journal of Nondestructive Testing, 52 (2016), pp. 235–244.

[19] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to total variation for image analysis*, Theoretical Foundations and Numerical Methods for Sparse Recovery, 9 (2010), p. 227.

[20] T. Chan, S. Esedoglu, F. Park, and A. Yip, *Total variation image restoration: overview and recent developments*, Handbook of Mathematical Models in Computer Vision, (2006), pp. 17–31.

[21] T. Che, Y. Li, A. Jacob, Y. Bengio, and W. Li, *Mode regularized generative adversarial networks*, in International Conference on Learning Representations, 2017.

[22] D. Chen, J. Tachella, and M. E. Davies, *Equivariant imaging: learning beyond the range space*, in

IEEE/CVF International Conference on Computer Vision, 2021, pp. 4379–4388.
[23] D. Chen, J. Tachella, and M. E. Davies, *Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5647–5656.
[24] M. Chen, P. Lin, Y. Quan, T. Pang, and H. Ji, *Unsupervised phase retrieval using deep approximate MMSE estimation*, IEEE Transactions on Signal Processing, 70 (2022), pp. 2239–2252.
[25] X. Cheng, N. S. Chatterji, P. L. Bartlett, and M. I. Jordan, *Underdamped Langevin MCMC: a non-asymptotic analysis*, in Conference on learning theory, PMLR, 2018, pp. 300–323.
[26] Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon, *A Bayesian perspective on the deep image prior*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5443–5451.
[27] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, et al., *The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository*, Journal of Digital Imaging, 26 (2013), pp. 1045–1057.
[28] M. Z. Darestani and R. Heckel, *Accelerated MRI with un-trained neural networks*, IEEE Transactions on Computational Imaging, 7 (2021), pp. 724–733.
[29] V. De Bortoli and A. Durmus, *Convergence of diffusions and their discretizations: from continuous to discrete processes and back*, Arxiv Preprint Arxiv:1904.09808, (2019).
[30] Q. Ding, G. Chen, X. Zhang, Q. Huang, H. Ji, and H. Gao, *Low-dose CT with deep learning regularization via proximal forward backward splitting*, Physics in Medicine & Biology, (2020).
[31] B. Dong, H. Ji, J. Li, Z. Shen, and Y. Xu, *Wavelet frame based blind image inpainting*, Applied and Computational Harmonic Analysis, 32 (2012), pp. 268–279.
[32] D. L. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.
[33] A. Durmus and É. Moulines, *High-dimensional Bayesian inference via the unadjusted Langevin algorithm*, Bernoulli, 25 (2019), pp. 2854–2882.
[34] A. Durmus, E. Moulines, and M. Pereyra, *Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau*, SIAM Journal on Imaging Sciences, 11 (2018), pp. 473–506.
[35] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences*, Springer Berlin Heidelberg, 1985.
[36] R. Gribonval, *Should penalized least squares regression be interpreted as maximum a posteriori estimation?*, IEEE Transactions on Signal Processing, 59 (2011), pp. 2405–2410.
[37] B. Guo, Y. Han, and J. Wen, *AGEM: solving linear inverse problems via deep priors and sampling*, Advances in Neural Information Processing Systems, 32 (2019).
[38] P. Hagemann, J. Hertrich, and G. Steidl, *Stochastic normalizing flows for inverse problems: a Markov Chains viewpoint*, SIAM/ASA Journal on Uncertainty Quantification, 10 (2022), pp. 1162–1190.
[39] P. Hand, O. Leong, and V. Voroninski, *Phase retrieval under a generative prior*, Advances in Neural Information Processing Systems, 31 (2018).
[40] R. Heckel and P. Hand, *Deep decoder: concise image representations from untrained non-convolutional networks*, in International Conference on Learning Representations, 2018.
[41] J. Hertrich, S. Neumayer, and G. Steidl, *Convolutional proximal neural networks and plug-and-play algorithms*, Linear Algebra and its Applications, 631 (2021), pp. 203–234.
[42] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, Advances in Neural Information Processing Systems, 33 (2020), pp. 6840–6851.
[43] K. Jaganathan, Y. C. Eldar, and B. Hassibi, *Phase retrieval: an overview of recent developments*, Optical Compressive Imaging, (2016), pp. 279–312.
[44] Z. Kadkhodaie and E. Simoncelli, *Stochastic solutions for linear inverse problems using the prior implicit in a denoiser*, Advances in Neural Information Processing Systems, 34 (2021), pp. 13242–13254.
[45] U. S. Kamilov, H. Mansour, and B. Wohlberg, *A plug-and-play priors approach for solving nonlinear imaging inverse problems*, IEEE Signal Processing Letters, 24 (2017), pp. 1872–1876.
[46] B. Kawar, M. Elad, S. Ermon, and J. Song, *Denoising diffusion restoration models*, in Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, pp. 23593–23606.

[47] B. Kawar, G. Vaksman, and M. Elad, *SNIPS: solving noisy inverse problems stochastically*, Advances in Neural Information Processing Systems, 34 (2021), pp. 21757–21769.
[48] S. Kim, Q. Song, and F. Liang, *Stochastic gradient Langevin dynamics with adaptive drifts*, Journal of Statistical Computation and Simulation, 92 (2022), pp. 318–336.
[49] A. Krull, T. Buchholz, and F. Jug, *Noise2Void-learning denoising from single noisy images*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2129–2137.
[50] R. Laumont, V. D. Bortoli, A. Almansa, J. Delon, A. Durmus, and M. Pereyra, *Bayesian imaging using Plug & Play priors: when Langevin meets Tweedie*, SIAM Journal on Imaging Sciences, 15 (2022), pp. 701–737.
[51] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, *Noise2Noise: learning image restoration without clean data*, in ICML, 2018, pp. 2965–2974.
[52] C. Li, C. Chen, D. Carlson, and L. Carin, *Preconditioned stochastic gradient Langevin dynamics for deep neural networks*, in Thirtith AAAI Conference on Artificial Intelligence, 2016.
[53] T. Li, Z. Zhuang, H. Liang, L. Peng, H. Wang, and J. Sun, *Self-validation: early stopping for single-instance deep generative priors*, Arxiv Preprint Arxiv:2110.12271, (2021).
[54] Z.-P. Liang and P. C. Lauterbur, *Principles of magnetic resonance imaging: a signal processing perspective*, SPIE Optical Engineering Press, 2000.
[55] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, *Non-local recurrent network for image restoration*, Arxiv Preprint Arxiv:1806.02919, (2018).
[56] J. Liu, T. Kuang, and X. Zhang, *Image reconstruction by splitting deep learning regularization from iterative inversion*, in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 224–231.
[57] C. Louchet and L. Moisan, *Total variation denoising using iterated conditional expectation*, in 22nd European Signal Processing Conference, IEEE, 2014, pp. 1592–1596.
[58] C. Metzler, A. Mousavi, R. Heckel, and R. Baraniuk, *Unsupervised learning with Stein's unbiased risk estimator*, Arxiv Preprint Arxiv:1805.10531, (2018).
[59] C. Metzler, P. Schniter, A. Veeraraghavan, et al., *prDeep: robust phase retrieval with a flexible deep network*, in International Conference on Machine Learning, PMLR, 2018, pp. 3501–3510.
[60] C. A. Metzler, A. Maleki, and R. G. Baraniuk, *BM3D-PRGAMP: compressive phase retrieval based on BM3D denoising*, in IEEE International Conference on Image Processing, IEEE, 2016, pp. 2504–2508.
[61] A. Mousavi, A. B. Patel, and R. G. Baraniuk, *A deep learning approach to structured signal recovery*, in 53rd annual allerton conference on communication, control, and computing, IEEE, 2015, pp. 1336–1343.
[62] P. Nair, R. G. Gavaskar, and K. N. Chaudhury, *Fixed-point and objective convergence of plug-and-play algorithms*, IEEE Transactions on Computational Imaging, 7 (2021), pp. 337–348.
[63] Y. Nan, Y. Quan, and H. Ji, *Variational-EM-based deep learning for noise-blind image deblurring*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3626–3635.
[64] F. Natterer, *The mathematics of computerized tomography*, SIAM, 2001.
[65] T. Pang, Y. Quan, and H. Ji, *Self-supervised bayesian deep learning for image recovery with applications to compressive sensing*, in European Conference on Computer Vision, 2020.
[66] T. Pang, H. Zheng, Y. Quan, and H. Ji, *Recorrupted-to-Recorrupted: unsupervised deep learning for image denoising*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2043–2052.
[67] Y. Quan, M. Chen, T. Pang, and H. Ji, *Self2self with dropout: learning self-supervised denoising from single image*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1890–1898.
[68] G. O. Roberts and O. Stramer, *Langevin diffusions and Metropolis-Hastings algorithms*, Methodology and Computing in Applied Probability, 4 (2002), pp. 337–357.
[69] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, (1996), pp. 341–363.
[70] Y. Romano, M. Elad, and P. Milanfar, *The little engine that could: regularization by denoising (RED)*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 1804–1844.
[71] J. Romberg, *Imaging via compressive sampling*, Ieee Signal Processing Magazine, 25 (2008), pp. 14–20.

[72] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.
[73] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, *Plug-and-play methods provably converge with properly trained denoisers*, in International Conference on Machine Learning, PMLR, 2019, pp. 5546–5557.
[74] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, *Improved techniques for training GANs*, Advances in Neural Information Processing Systems, 29 (2016).
[75] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, *Phase retrieval with application to optical imaging: a contemporary overview*, Ieee Signal Processing Magazine, 32 (2015), pp. 87–109.
[76] W. Shi, F. Jiang, S. Liu, and D. Zhao, *Scalable convolutional neural network for image compressed sensing*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12290–12299.
[77] Z. Shi, P. Mettes, S. Maji, and C. G. Snoek, *On measuring and controlling the spectral bias of the deep image prior*, International Journal of Computer Vision, 130 (2022), pp. 885–908.
[78] Y. Song, L. Shen, L. Xing, and S. Ermon, *Solving inverse problems in medical imaging with score-based generative models*, in International Conference on Learning Representations, 2022.
[79] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, in International Conference on Learning Representations, 2021.
[80] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, *Plug-and-play priors for bright field electron tomography and sparse interpolation*, IEEE Transactions on Computational Imaging, 2 (2016), pp. 408–423.
[81] J. Sun, Q. Qu, and J. Wright, *A geometric analysis of phase retrieval*, Foundations of Computational Mathematics, 18 (2018), pp. 1131–1198.
[82] Y. Sun, B. Wohlberg, and U. S. Kamilov, *An online plug-and-play algorithm for regularized image reconstruction*, IEEE Transactions on Computational Imaging, 5 (2019), pp. 395–408.
[83] Y. Sun, Z. Wu, X. Xu, B. Wohlberg, and U. S. Kamilov, *Scalable plug-and-play ADMM with convergence guarantees*, IEEE Transactions on Computational Imaging, 7 (2021), pp. 849–863.
[84] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, *On the importance of initialization and momentum in deep learning*, in International Conference on Machine Learning, PMLR, 2013, pp. 1139–1147.
[85] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, *A study of methods of choosing the smoothing parameter in image restoration by regularization*, IEEE Transactions on Pattern Analysis & Machine Intelligence, 13 (1991), pp. 326–339.
[86] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Deep image prior*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.
[87] K. Usui, K. Ogawa, M. Goto, Y. Sakano, S. Kyougoku, and H. Daida, *Quantitative evaluation of deep convolutional neural network-based image denoising for low-dose computed tomography*, Visual Computing for Industry, Biomedicine, and Art, 4 (2021), pp. 1–9.
[88] 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, IEEE, 2013, pp. 945–948.
[89] M. Welling and Y. W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, in International Conference on Machine Learning (ICML-11), 2011, pp. 681–688.
[90] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov, *Provable convergence of plug-and-play priors with MMSE denoisers*, IEEE Signal Processing Letters, 27 (2020), pp. 1280–1284.
[91] Y. Yang, J. Sun, H. Li, and Z. Xu, *Deep admm-net for compressive sensing MRI*, in NeurIPS, 2016, pp. 10–18.
[92] J. Zhang and B. Ghanem, *ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing*, in IEEE conference on computer vision and pattern recognition, 2018, pp. 1828–1837.
[93] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang, *Learning fully convolutional networks for iterative non-blind deconvolution*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 3817–3825.
[94] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, *Plug-and-play image restoration*

with deep denoiser prior, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021), pp. 6360–6376.

[95] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, *Residual non-local attention networks for image restoration*, in International Conference on Learning Representations, 2018.

[96] D.-X. Zhou, *Universality of deep convolutional neural networks*, Applied and Computational Harmonic Analysis, 48 (2020), pp. 787–794.

[97] M. Zhussip, S. Soltanayev, and S. Chun, *Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10255–10264.