Un-supervised Learning for Blind Image Deconvolution via Monte-Carlo Sampling

Ji Li[†], **Yuesong Nan**[†], **and Hui Ji**[†]

[†] Department of Mathematics, National University of Singapore, Singapore

E-mail: matliji@nus.edu.sg, nanyuesong@u.nus.edu, matjh@nus.edu.sg

Oct 2021

Abstract. Deep learning has been a powerful tool for solving many inverse imaging problems. The majority of existing deep-learning-based solutions are supervised on an external dataset with many blurred/latent image pairs. Recently, there has been an increasing interest on developing dataset-free deep learning methods for image recovery without any prerequisite on external training dataset, including blind deconvolution. This paper aims at developing an un-supervised learning method for blind image deconvolution, which does not call any training sample yet provides very competitive performance. Based on the re-parametrization of latent image using a deep network with random weights, this paper proposed to approximate the maximum-a-posteriori (MAP) estimator of the blur kernel using the Monte-Carlo (MC) sampling method. The MC sampling is efficiently implemented by using dropout and random noise layer, which does not require conjugate model as traditional variational inference does. Extensive experiments on popular benchmark datasets for blind image deconvolution showed that the proposed method not only outperformed existing non-learning methods, but also noticeably outperformed existing deep learning methods, including both supervised and un-supervised ones.

AMS classification scheme numbers: 68U10, 94A08

PACS numbers:

Submitted to: *Inverse Problems*

1. Introduction

Image blurring is one type of image degradation in photography that causes the loss of image details. For example, the so-called motion blurring happens when there is camera shake during shutter time. When the variation of the scene depth is small and the camera motion is dominated by in-image translation, such motion blurring can be modeled by a convolution process:

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

where $\otimes$ denotes 2D discrete convolution operator, $\boldsymbol{y}$ denotes the input blurred image, $\boldsymbol{k}$ denotes the blur kernel, $\boldsymbol{x}$ denotes the latent image with sharp details for recovery,

and $\boldsymbol{n}$ denotes measurement noise. Then, the so-called blind image deconvolution is about recovering the latent image with sharp details from an input motion-blurred image. In other words, blind image deconvolution concerns the estimation of both the kernel $\boldsymbol{k}$ and the latent image $\boldsymbol{x}$ from the input $\boldsymbol{y}$.

Blind image deconvolution is an ill-posed inverse problem with severe solution ambiguity, *i.e.*, there exists many kernel/image pairs that satisfy the equation (1), as long as the kernel can be factorized by $\boldsymbol{k} = \boldsymbol{k}_1 \otimes \boldsymbol{k}_2$. Such factorization leads to another solution $(\boldsymbol{k}_1, \boldsymbol{k}_2 \otimes \boldsymbol{x})$ since

$$\boldsymbol{y} = \boldsymbol{k} \otimes \boldsymbol{x} + \boldsymbol{n} = (\boldsymbol{k}_1) \otimes (\boldsymbol{k}_2 \otimes \boldsymbol{x}) + \boldsymbol{n}. \tag{2}$$

One well-known example is the no-blur pair [42]: $(\boldsymbol{\delta}, \boldsymbol{k} \otimes \boldsymbol{x})$, where $\boldsymbol{\delta}$ denotes the (Dirac) Delta kernel. The no-blur pair gives a trivial solution to the problem such that the result remains blurred. How to resolve solution ambiguity is one main concern when designing a method to solve blind image deconvolution. Certain priors need to be imposed on both latent image and blur kernel to resolve solution ambiguity existing in (1). These priors are introduced either in the formulation of Bayesian inference or regularized optimization model. The problem of blind image deconvolution has been extensively studied in the past. In the next, we give a brief review of the works are related to the approach presented in this paper.

1.1. Related works

1.1.1. Hand-crafted regularizations and Bayesian inference Considering an MAP estimator for solving (1), we have

$$p(\boldsymbol{k}, \boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{k}, \boldsymbol{x}) p_x(\boldsymbol{x}) p_k(\boldsymbol{k}), \tag{3}$$

where $p(\boldsymbol{y}|\boldsymbol{k}, \boldsymbol{x})$ denotes the likelihood function, and $p_x(\boldsymbol{x}), p_k(\boldsymbol{k})$ denote the probability density functions (statistical priors) of $\boldsymbol{x}, \boldsymbol{k}$. In the case of i.i.d. Gaussian white noise $\boldsymbol{n}$ with s.t.d. σ, the negative logarithm on both sides of (3) leads to an MAP estimator, which solves the problem:

$$\min_{\boldsymbol{x},\boldsymbol{k}} \quad \frac{1}{2\sigma^2} \|\boldsymbol{k} \otimes \boldsymbol{x} - \boldsymbol{y}\|_2^2 - \log p_x(\boldsymbol{x}) - \log p_k(\boldsymbol{k}), \tag{4}$$

where $\log p_x(\boldsymbol{x})$ and $\log p_k(\boldsymbol{k})$ are the logarithm of the prior distributions of $\boldsymbol{x}$ and $\boldsymbol{k}$. The $\text{MAP}_{\boldsymbol{x},\boldsymbol{k}}$ estimator (4) can also be viewed as a regularization model:

$$\min_{\boldsymbol{x},\boldsymbol{k}} \quad \frac{1}{\sigma^2} \|\boldsymbol{k} \otimes \boldsymbol{x} - \boldsymbol{y}\|_2^2 + \phi(\boldsymbol{x}) + \psi(\boldsymbol{k}), \tag{5}$$

where $\phi(\boldsymbol{x})$ and $\psi(\boldsymbol{k})$ are regularization terms induced by the priors imposed on image and kernel. As blurring effect happens mostly on high-frequencies of image, in most recent approaches, the problem is solved in the domain of image gradients $\nabla\boldsymbol{x}$. There are some works which considers the problem in the dark/extreme channel [40, 55]. In the past, there have been extensive studies on the regularizations for image (image gradient) in various image recovery tasks, including blind deconvolution, as well as the regularizations for motion-blur kernels. For the regularization on motion-blur kernel, the squared ℓ_2-norm $\psi(\boldsymbol{k}) = \lambda_2 \|\boldsymbol{k}\|_2^2$, induced by Gaussian prior on kernel, is the most popular one (*e.g.* [24,52,54]). The curvelet-based ℓ_1-norm regularization is used in [5].

The most-often seen regularization for $\boldsymbol{x}$ is the ℓ_p-norm relating regularization, including total variation (TV) regularization $\phi(\nabla \boldsymbol{x}) = \lambda_1 \|\nabla \boldsymbol{x}\|_1$ (*e.g.* [2, 7]), normalized TV regularization $\phi(\nabla \boldsymbol{x}) = \lambda_1 \|\frac{\nabla \boldsymbol{x}\|_1}{\nabla \boldsymbol{x}\|_2}$ [24], wavelet-based regularization [4], and ℓ_0-norm relating regularization (*e.g.* [54]). In addition to ℓ_p-norm relating regularizations, there are also the regularizations derived from other image priors, including recurrence prior of image patches, [35, 44, 46]. There are several works to infer the blur kernel by the spectral map of the blurred image with the power law decay of clean natural images [12, 17]. To efficiently solve the resultant nonconvex optimization and avoid the convergence to degenerated solutions, edge selection as a powerful technique in blind image deconvolution is widely used, see *e.g.* [8,13,39,52,56].

Instead of jointly estimating both image and kernel via the MAP framework, another formulation of Bayesian inference is based on $\text{MAP}_{\boldsymbol{k}}$. Its theoretical advantages over the joint MAP estimator is discussed in [30], in terms of avoiding the convergence to the no-blur solution. As the posterior distribution $p(\boldsymbol{k}|\boldsymbol{y})$ is computationally intractable, variational Bayesian inference in [1,30,36,51] is adopted, which approximates $p(\boldsymbol{k}|\boldsymbol{y})$ by a mean-field distribution. Fergus *et al.* [9] modeled gradient images using i.i.d. mixture of zero-mean Gaussians. Babacan *et al.* [1] used the super-Gaussian image priors. Wipf and Zhang [51] adopted Gaussian scale mixture (GSM) prior modeling, and linked the variational Bayesian algorithm to an specific MAP reformulation. Note that to make the variational Bayesian tractable, the conjugacy of the probability modeling is required.

1.1.2. Supervised deep learning methods In recent years, deep learning has been a powerful tool for blind image drblurring. Most existing deep-learning-based solutions are based on *supervised learning*, *i.e.*, a deep neural network (DNN) is trained over many blurred/latent image pairs. There are two types of approaches to supervised learning. One type of methods explicitly calls the convolution process in the network. Chakrabarti *et al.* [6] learned the deblur kernel in the frequency domain. Schuler *et al.* [45] and Li *et al.* [32] unrolled an alternative minimization scheme of an MAP estimator with its image-prior-related part replaced by a learnable DNN. Kaufman and Fattal [19] proposed a two-stage approach which first learns a DNN for predicting blur kernel and then learns another DNN for deblurring the image using the predicted kernel. There are also some other variations, including learning a discriminator or the fitting term (*e.g.* [31, 38]). These methods are based on the convolution-based blurring model. Thus, there are more suitable for processing the images with uniform blurring. Another type of methods trains the DNN in an end-to-end way that directly maps an blurred image to a sharp one. The main differences among these works are on the design of network architectures; see *e.g.* [26, 37, 47, 53]. Xu *et al.* [53] is the first work that proposes the end-to-end training for a deblurring DNN. Tao *et al.* [47] employed a coarse-to-fine and recurrence structures in network. As these methods do not reply on the convolution-based blurring model, they are applicable to the images with spatially-varying blurring effect (*e.g.* [57]).

1.1.3. Unsupervised deep learning for image denoising In order to have good generalization performance, a supervised learning method requires an external dataset with many training samples for training the network. Such a prerequisite on training data with truth images limits its application in certain domains, including medical imaging and scientific images. It is of great value to develop deep-learning-based

method which does not require the access to external truth images, while still providing good performance.

While there are few works on unsupervised learning methods for blind deconvolution, there have been many works on unsupervised deep learning methods for image denoising. Ulyanov *et al*. [48] proposed the seminal work, deep image prior (DIP), which shows there exist implicit regularizations introduced by the structures of a convolutional neural network (CNN), which prefers regular image structures over random noising when training a denoising network. The Noise2Noise method presented in [29] showed that one can train a denoising network using the noisy/noisy image pairs, instead of noisy/clean image pair as those supervised denoising network. Afterward, many methods are developed to augment noisy images to image pairs for training a denoising network; see *e.g*. [3, 25, 28, 41].

1.1.4. Unsupervised deep learning methods for blind image deblurring In comparison to image denoising, the works on unsupervised deep learning for blind deconvolution are few. A method is proposed in Ren *et al*. [43], which is motivated by the DIP [11,48] and the double-DIP [11]. The double-DIP proposed to use two NNs to predict two layers, cartoon and texture, of an image for decomposition. Similarly, Ren *et al*. [43] also used two NNs for predicting kernel and image: One CNN for predicting the image and an FCN for predicting blur kernel. While DIP is very effective on image denoising, it cannot resolve solution ambiguity in blind deconvolution, *i.e*. the convergence to a blurred solution. Thus, an additional TV-regularization is imposed on the latent image in the cost function in Ren *et al*. [43] for resolving solution ambiguity. Another approach is based on GAN. Lu *et al*. [34] proposed a GAN-based disentangle representation framework with unpaired blurry and sharp images to deblur domain-specific images such as facial images. Such a GAN-based method will encounter domain shift issues for wider adoption. Liu *et al*. [33] proposed an optical-flow-based self-supervised method for processing images or videos, where the network is trained on video sequences.

1.2. Discussion on existing un-supervised methods

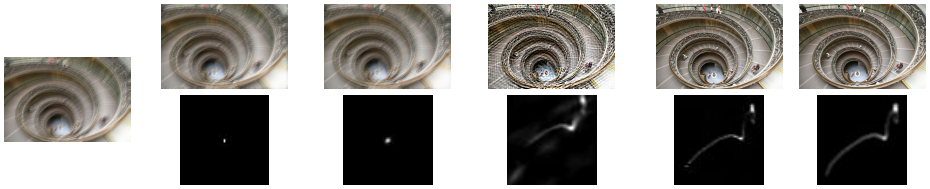
The idea of double-DIP for blind image deconvolution [43] is to train two generative NNs $\mathcal{G}_k(\cdot;\boldsymbol{\theta}_k)$ and $\mathcal{G}_x(\cdot;\boldsymbol{\theta}_x)$ for predicting the kernel $\boldsymbol{k}$ and the latent image $\boldsymbol{x}$ respectively. These two NNs map the two fixed initial seeds $\boldsymbol{z}_x, \boldsymbol{z}_k$ to the unknown image and kernel:

$$\mathcal{G}_x(\cdot;\boldsymbol{\theta}_x):\ \boldsymbol{z}_x \to \boldsymbol{x}, \quad \text{and} \quad \mathcal{G}_k(\cdot;\boldsymbol{\theta}_k):\ \boldsymbol{z}_k \to \boldsymbol{k}, \tag{6}$$

where $\boldsymbol{\theta}_x, \boldsymbol{\theta}_k$ are the parameter sets of two NNs. Then, one might train the two networks to have a maximum likelihood estimation (MLE) by using the following loss function:

$$\min_{\boldsymbol{\theta}_k,\boldsymbol{\theta}_x} \frac{1}{2\sigma^2}\|\mathcal{G}_k(\boldsymbol{z}_k;\boldsymbol{\theta}_k) \otimes \mathcal{G}_x(\boldsymbol{z}_x;\boldsymbol{\theta}_x) - \boldsymbol{y}\|_2^2. \tag{7}$$

DIP cannot resolve solution ambiguity of blind deconvolution we discussed in the previous section, when training the networks using (7). What DIP can avoid is the introduction of random noise in the prediction. However, the pairs of image/kernel which cause solution ambiguities do not contain such noisy pattern. For example, the image in the trivial no-blur pair $(\boldsymbol{y}, \boldsymbol{\delta})$ is the smoothed-out version of the latent $\boldsymbol{x}$



(a) Input (b) w/o regularization (c) Early stopping (d) TV regularization (e) Our model (27) (f) Groundtruth

Figure 1. Estimating the image/kernel for the image shown in (a) using the same NN trained by different methods. (b) Training by the loss (7) without early stopping; (c) Training by (7) with early stopping in DIP; (d) Training by TV regularized loss (8) in Ren *et al*. [43]; (e) Training by the proposed method; (f) Truth image/kernel.

without random noise. As a result, the smoothed-out version of the image $\boldsymbol{x}$ tends to appear before the truth image $\boldsymbol{x}$ during the training. In other words, DIP itself is not sufficient for handling possible overfitting when training the networks by (7). See Figure 1 (c) for an illustration of the convergence to the no-blur pair when using (7) to train two networks. To avoid possible overfitting when using double-DIP to solve blind deconvolution problem, an additional TV regularization is introduced in [43] on the image predicted by the network. The resulting loss function is then:

$$\min_{\boldsymbol{\theta}_k,\boldsymbol{\theta}_x} \frac{1}{2\sigma^2}\|\mathcal{G}_k(\boldsymbol{z}_k;\boldsymbol{\theta}_k)\otimes\mathcal{G}_x(\boldsymbol{z}_x;\boldsymbol{\theta}_x)-\boldsymbol{y}\|_2^2+\lambda\|\nabla\mathcal{G}_x(\boldsymbol{z}_x;\boldsymbol{\theta}_x)\|_1, \tag{8}$$

where λ is a hyper-parameter to be tuned-up. With additional TV regularization on the prediction, the loss function (8) partially addressed the likely overfitting. However, such a regularization also have a negative impact on the result, as it is not adaptively optimized for individual images.

See Figure 1 for an illustration of the predictions from the network trained using MLE-based (7), TV-regularized loss (8), and the proposed method. It can be seen that the network trained using (7) will converge to the prediction of the no-blur pair, and the network trained using (8) does not provide an accurate estimation of the blur kernel due to the non-adaptive regularization on the sample image. Overall, existing DIP-based unsupervised methods for blind image deconvolution, *e.g*. DIP-based Ren *et al*. [43], leave a lot of room for further improvement. For GAN-based methods, *e.g*. Lu *et al*. [34], they are more suitable for processing domain-specific images (*e.g*. text or face images), not general natural images.

1.3. Main idea and contribution

In this paper, we proposed a new un-supervised deep learning method for blind image deconvolution. The method is not about the design of new deterministic network architecture for blind deconvolution, but is about introducing deep-NN-based re-parametrization [16,20,21] technique, in the framework of Bayesian inference, to tackle the overfitting caused by the absence of training samples.

The proposed method is built on the well-known $\text{MAP}_{\boldsymbol{k}}$ estimator for blind deconvolution [30]. The estimator $\text{MAP}_{\boldsymbol{k}}$ requires the marginalization of the images over the prior distribution $p_x(\boldsymbol{x})$, which is in general computationally intractable. The so-called *variational Bayesian approximation method* tackles such an issue by

approximating the distribution $p_x(\boldsymbol{x})$ using another distribution $q(\boldsymbol{x}, \boldsymbol{\gamma})$ to make the computation tractable. For example, the following factorization distribution is used in [9, 36] for mean field approximation: $q(\boldsymbol{x}; \boldsymbol{\gamma}) = \prod_i q(\boldsymbol{x}_i; \gamma_i)$, where $q(\boldsymbol{x})$ (or $q(\nabla \boldsymbol{x})$) and $q(\boldsymbol{k})$ is defined by a conjugate hierarchical distribution,

$$q(\boldsymbol{x_i}) \sim \mathcal{N}(x_i; 0, \gamma_i), \text{ where } \gamma_i \sim \Gamma(\gamma_i; a, b). \tag{9}$$

where $\mathcal{N}$ denotes normal distribution and Γ denotes Gamma distribution.

It can be seen that the performance of such a variational approximation method depends on how accurately the approximating distribution can approximate the prior distribution $p_x(\boldsymbol{x})$. In order to make q computationally tractable, the distribution q are greatly simplified in existing methods. As a result, the approximating distribution does not accurately characterize the properties of natural images. For example, the mean-field assumption of q assumes the independence of all image pixels, which certainly is sub-optimal. This motivates us to study a different Bayesian approximation method to the $\text{MAP}_{\boldsymbol{k}}$ estimator. Our solution is based on the so-called re-parametrization technique [16, 20, 21], which approximates the prior distribution of $p_x(\boldsymbol{x})$ by a NN with random weights

$$\mathcal{G}_x(\cdot; \boldsymbol{\theta}_x, \boldsymbol{\gamma}_x) : \ \boldsymbol{z}_x \rightarrow \text{samples from } p_x(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{\theta}_x$ denotes the set of deterministic weights of the network and $\boldsymbol{\gamma}_x$ denotes the set of the weights randomly sampled from standard distribution (*e.g.* normal or Bernoulli distribution). The distribution derives from (10) will provide better characterization of the correlations among image pixels, in comparison to existing mean-field models. For the representation of kernel $\boldsymbol{k}$, we also use a generative network with learnable weights to express it:

$$\mathcal{G}_k(\cdot; \boldsymbol{\theta}_k) : \ \boldsymbol{z}_k \rightarrow \boldsymbol{k}.$$

Note that, unlike image generative network $\mathcal{G}_x(\cdot)$, there is no need to impose randomness on $\mathcal{G}_k(\cdot)$, as the marginalization is only for the image in our approach.

Remark (Prior distribution of targe image in un-supervised learning). It is noted that the definition of prior distribution $p(\boldsymbol{x})$ in our setting has different meaning from that used in supervised learning or GAN. In supervised learning or GAN, a model is trained over many images in the same domain, which is supposed to learn the prior distribution of all images in that domain. Then, such a prior distribution over all images allows one to use a pre-trained model to process unseen images in the same domain. In our unsupervised case, the network is trained only for one specific target image $\boldsymbol{x}$. The resulting prior distribution $p(\boldsymbol{x})$ is then about the probability of such an image. As a result, in contrast to that in supervised learning, the prior distribution function $p(\boldsymbol{x})$ that we are approximating is a Gaussian-like function that centers at the target image x with small variance.

Based on the proposed image generative network with random weights, one can approximate the $\text{MAP}_{\boldsymbol{k}}$ using the MC sampling method. In this paper, we proposed an efficient MC sampling scheme for approximating the $\text{MAP}_{\boldsymbol{k}}$. Such a MC sampling scheme is integrated in an alternating iteration scheme, which leads to an efficient algorithm for training the network. See below for the summary of our contributions.

- **Deep-NN-based re-parametrization for variational approximation**. An deep-NN-based re-parametrization is introduced to provide more accurate approximation to the distribution of images over existing related methods.

- **Efficient training scheme with the integration of an efficient MC sampling and alternating iteration scheme**. An efficient MC sampling is proposed to approximate the $\text{MAP}_{\boldsymbol{k}}$ estimator, which is integrated into an alternating iteration scheme for training the network.
- **Noticeable performance improvement over existing solutions.** Extensive experiments on three benchmark datasets show that the proposed method not only noticeably outperformed existing non-learning methods and un-supervised methods, but also outperformed existing supervised deep learning methods.
- **Potential application in other non-linear inverse problems.** The techniques presented in this paper are easy to implement. It can see its potential applications to solve other linear or non-linear inverse problems.

In short, we proposed to model the prior distribution of the image by a deep-NN-based re-parametrization technique, which is implemented by including both additive Gaussian noise layer and dropout layer in the network. Then, one can sample the distribution $p_x(\boldsymbol{x})$, expressed by the trained network with random variables. One image sample can be obtained with one instance of Gaussian noise and dropout [21] for the generative network. In other words, we proposed to train a generative network for image with random process, which enables an efficient MC-sampling based approximation to the integral over $\boldsymbol{x}$ involved in the $\text{MAP}_{\boldsymbol{k}}$. Note that, in comparison to the pre-defined TV regularization on image [43] which assumes the gradients of target image follows a Laplacian distribution, the prior distribution of target image in our approach is learned via training a generative networks with random process.

1.4. Organization

The organization of the paper is as follows. In Section 2, the main results are presented with all details. In Section 3, extensive experiments are conducted to evaluate the performance of the proposed method. Section 4 concludes the paper.

2. Monte-Carlo sampling for MAP_k

The section is devoted to the detailed discussion of the proposed MC-sampling-based self-supervised deep learning method for blind image deconvolution. The method is built on the following $\text{MAP}_{\boldsymbol{k}}$ for $\boldsymbol{k}$:

$$\begin{aligned}\widehat{\boldsymbol{k}} &= \underset{\boldsymbol{k}}{\operatorname{argmax}}\, p(\boldsymbol{k}|\boldsymbol{y}) = \underset{\boldsymbol{k}}{\operatorname{argmax}} \int p(\boldsymbol{k},\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x} \\ &= \underset{\boldsymbol{k}}{\operatorname{argmax}}\, p_k(\boldsymbol{k}) \int p(\boldsymbol{y}|\boldsymbol{k},\boldsymbol{x})p_x(\boldsymbol{x})d\boldsymbol{x}, \quad \text{(Bayes' rule)} \end{aligned} \tag{11}$$

where p_x/p_k denotes the prior distribution of image/kernel. Once the kernel is predicted, the latent image $\boldsymbol{x}$ can be efficiently computed by calling a non-blind deconvolution method. It is shown in [30] that, in terms of avoiding the convergence to the no-blur pair, the $\text{MAP}_{\boldsymbol{k}}$ is more effective than $\text{MAP}_{\boldsymbol{k},\boldsymbol{x}}$ which jointly estimate $\boldsymbol{k}$ and $\boldsymbol{x}$ as the following:

$$(\widetilde{\boldsymbol{k}},\widetilde{\boldsymbol{x}}) = \underset{\boldsymbol{k},\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{k},\boldsymbol{x}|\boldsymbol{y}). \tag{12}$$

The reason comes from the size difference between the kernel and image: kernel size is usually much smaller than image size. Thus, the $\text{MAP}_{\boldsymbol{k}}$ is much less likely biased to Dirac Delta $\boldsymbol{\delta}$ than $\text{MAP}_{\boldsymbol{k},\boldsymbol{x}}$. Note that the MAP estimator $\text{MAP}_{\boldsymbol{k}}$ requires the marginalization of the images over the posterior distribution $p(\boldsymbol{k}, \boldsymbol{x}|\boldsymbol{y})$. In our approach, such a marginalization is approximated by an MC sampling method.

2.1. MC sampling with NN-based re-parametrization for approximate the MAP estimator (11)

In this section, we present the detailed derivation of the proposed MC-sampling based approximation to $\text{MAP}_{\boldsymbol{k}}$. By Bayesian rule, we have

$$\log p(\boldsymbol{y}|\boldsymbol{k}) = \log \int p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{k}) d\boldsymbol{x} \propto \log \int p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k}) d\boldsymbol{x}. \tag{13}$$

Then, by the Jensen's inequality,

$$-\log \int p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k}) dx = -\log \mathbb{E}_{p(\boldsymbol{x})}[p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k})] \leq -\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k})]. \tag{14}$$

Note that $\text{MAP}_{\boldsymbol{k}}$ is the minimum of the following cost function:

$$\min_{\boldsymbol{k}} \quad \mathcal{L}(\boldsymbol{k}) := -\log p(\boldsymbol{k}) - \log \int p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k}) dx. \tag{15}$$

Then, we approximate the $\text{MAP}_{\boldsymbol{k}}$ by minimizing its upper bound given by (14):

$$\min_{k} \quad \mathcal{L}_{MC} := -\log p(\boldsymbol{k}) - \mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k})]. \tag{16}$$

In our approach, the distribution of kernel is assumed to be normal distribution such that $-\log(p(\boldsymbol{k})) = \lambda \|\boldsymbol{k}\|_2^2$. The expectation in (16) is approximated by the MC sampling method:

$$\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k})] \simeq \frac{1}{N} \sum_{i=1}^{N} \log p(\boldsymbol{y}|\boldsymbol{k}, \boldsymbol{x}_i), \tag{17}$$

where $\{\boldsymbol{x}_i\}_{i=1}^N$ are randomly sampled from the distribution $p_x(\boldsymbol{x})$. As $p_x(\boldsymbol{x})$ is not available, the remaining question is then how to simulate MC samples $\{\boldsymbol{x}_i\}_{i=1}^N$ without knowing the exact form of $p_x(\boldsymbol{x})$.

In our approach, we use a deep generative network with random weights to approximate the prior distribution of $\boldsymbol{x}$, such that we can simulate MC samples by the map

$$\mathcal{G}_x(\cdot, \boldsymbol{\theta}_x; \boldsymbol{\epsilon}, \boldsymbol{M}) : \boldsymbol{z}_x \rightarrow \text{Samples from } \boldsymbol{p}_x(\boldsymbol{x}), \tag{18}$$

where $\boldsymbol{\theta}_x$ are deterministic weights and $\boldsymbol{\epsilon}, \boldsymbol{M}$ are random variables. In other words, the network $\mathcal{G}_x$ is a network with random weights for approximating the prior distribution of the image $\boldsymbol{x}$, while the deterministic generative network used in [11, 43] is for predicting the latent image. Suppose that after training the network $\mathcal{G}_x$ with sufficient iterations, the model can well approximate $p(\boldsymbol{x})$, the Dirac Delta distribution. Then, the images predicted by the random instances of the trained model are the ones very close to the latent image $\boldsymbol{x}$.

The network with random process is implemented on an encoder-decoder backbone network, where several dropout layers are inserted in the decoder part and an additive Gaussian noise layer is attached in the beginning of the encoder. Let $\mathcal{B}(p_0)$ denotes the Bernoulli distribution with probability p_0. Then, the distribution represented by $\mathcal{G}_x(\cdot, \boldsymbol{\theta}_x)$ can be expressed as

$$\mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}; \boldsymbol{\theta}_x, \boldsymbol{M}), \tag{19}$$

where $\boldsymbol{\theta}_x$ are deterministic network weights, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$, $\boldsymbol{M} \sim \mathcal{B}(p_0)$. The distribution represented by (19) is then used to approximate the density function of the latent image $p_x(\boldsymbol{x})$. Once the network is trained with optimal weights $\boldsymbol{\theta}^*$, the MC samples from $p_x(\boldsymbol{x})$ can be simulated by sampling the network as follows.

$$\{\boldsymbol{x}_i\}_{i=1}^N = \{\mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}_i; \boldsymbol{\theta}_x^*, \boldsymbol{M}_i)\}_{i=1}^N, \tag{20}$$

where $\{\boldsymbol{\epsilon}_i, \boldsymbol{M}_i\}$ are independently drawn from $\mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$ and $\mathcal{B}(p_0)$ respectively.

In our approach, same as [11,43], we also train a deterministic network to predict the kernel $\boldsymbol{k}$, which is implemented by a small-size encoder-decoder backbone network. The reason of no randomness in the $\mathcal{G}_k(\cdot)$ is that there is no gain obtained as illustrated in our experiments, see Section 3.5.

Remark. While double-DIP [11], Ren *et al.* [43], and our methods all train two deep NNs for the latent image and the blur kernel. The NN for image in [11, 43] is deterministic which predicts the latent image. In contrast, the NN for image in our approach is the one with random weight which represents the prior distribution of images.

2.2. Alternating iterative scheme for network training

Note that most existing non-learning regularization methods take an alternating iterative scheme to alternatively update the estimation of $\boldsymbol{x}$ (or $\nabla \boldsymbol{x}$) and the kernel $\boldsymbol{k}$. Such an alternating iteration scheme works surpassingly well in practice. Thus, we also propose to use the same alternatively iterative scheme to updates the weights of two NNs:

$$\boldsymbol{\theta}_k^0 \to \boldsymbol{\theta}_x^1 \to \boldsymbol{\theta}_k^1 \to \boldsymbol{\theta}_x^2 \to \cdots \to \boldsymbol{\theta}_k^t \to \boldsymbol{\theta}_x^{t+1} \to \boldsymbol{\theta}_k^{t+1} \to \cdots, \tag{21}$$

which is equivalent to alternatively update the estimations of image and kernel. Recall that the $\text{MAP}_{\boldsymbol{k}}$ estimation of $\boldsymbol{k}$ at iteration t reads

$$\boldsymbol{k}^t = \underset{\boldsymbol{k}}{\text{argmax}} \quad p(\boldsymbol{k}|\boldsymbol{y}; \boldsymbol{\theta}_x^t). \tag{22}$$

We minimize

$$\begin{aligned}
\mathcal{L}_{MC} &= -\log p(\boldsymbol{k}) - \mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{k})] \\
&= -\log p(\boldsymbol{k}) - \mathbb{E}_{p(\boldsymbol{x})}[-\frac{1}{2\sigma^2} \|\boldsymbol{k} \otimes \boldsymbol{x} - \boldsymbol{y}\|_2^2] \\
&\approx -\log p(\boldsymbol{k}) + \frac{1}{N} \sum_{i=1}^N \frac{1}{2\sigma^2} \left\|\boldsymbol{k} \otimes \mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}_i^t; \boldsymbol{\theta}_x^t, \boldsymbol{M}_i^t) - \boldsymbol{y}\right\|_2^2 \\
&= \lambda \|\boldsymbol{k}\|_2^2 + \frac{1}{N} \sum_{i=1}^N \frac{1}{2\sigma^2} \left\|\boldsymbol{k} \otimes \mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}_i^t; \boldsymbol{\theta}_x^t, \boldsymbol{M}_i^t) - \boldsymbol{y}\right\|_2^2.
\end{aligned} \tag{23}$$

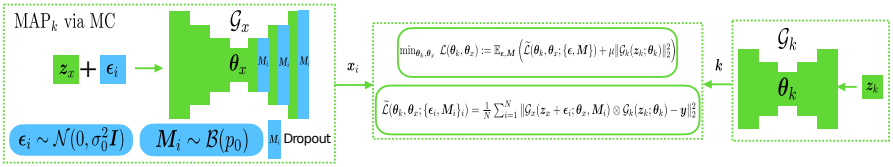


Figure 2. The diagram of the un-supervised training approach to blind deconvolution via randomized deep network.

It can be seen that the MC sampling is used to approximate the expectation in the derivation above. Notice that the kernel $\boldsymbol{k}$ is generated by the deterministic NN $\mathcal{G}_k(\boldsymbol{z}_k; \boldsymbol{\theta}_k)$. Defining the following loss function:

$$\widetilde{\mathcal{L}}(\theta_k, \theta_x; \{\epsilon_i, \boldsymbol{M}_i\}_i) = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{G}_k(\boldsymbol{z}_k; \boldsymbol{\theta}_k) \otimes \mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}_i; \boldsymbol{\theta}_x, \boldsymbol{M}_i) - \boldsymbol{y}\|_2^2. \tag{24}$$

Then, minimizing $\mathcal{L}_{MC}$ is to equivalently solve

$$\boldsymbol{\theta}_k^t = \underset{\boldsymbol{\theta}_k}{\operatorname{argmin}} \quad \left(\widetilde{\mathcal{L}}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_x^t; \{\boldsymbol{\epsilon}_i^t, \boldsymbol{M}_i^t\}_i) + 2\sigma^2 \lambda \|\mathcal{G}_k(\boldsymbol{z}_k; \boldsymbol{\theta}_k)\|_2^2 \right). \tag{25}$$

After the kernel $\boldsymbol{k}^t$ is updated at iteration t, we can update the estimation of prior distribution $p_x(\boldsymbol{x})$, which is determined by the Bayesian NN $\mathcal{G}_x(\boldsymbol{z}_x + \boldsymbol{\epsilon}; \boldsymbol{\theta}_x, \boldsymbol{M})$. The update on $\boldsymbol{\theta}_x$ is then defined by

$$\boldsymbol{\theta}_x^{t+1} = \underset{\boldsymbol{\theta}_x}{\operatorname{argmin}} \quad \widetilde{\mathcal{L}}(\boldsymbol{\theta}_k^t, \boldsymbol{\theta}_x; \{\boldsymbol{\epsilon}_i^{t+1}, \boldsymbol{M}_i^{t+1}\}_i). \tag{26}$$

In summary, the alternating update on two weight sets $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_x$ can be viewed as solving minimization problem

$$\min_{\boldsymbol{\theta}_k, \boldsymbol{\theta}_x} \quad \mathcal{L}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_x) = \min_{\boldsymbol{\theta}_k, \boldsymbol{\theta}_x} \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{M}} \left(\widetilde{\mathcal{L}}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_x; \{\boldsymbol{\epsilon}, \boldsymbol{M}\}) + \mu \|\mathcal{G}_k(\boldsymbol{z}_k; \boldsymbol{\theta}_k)\|_2^2 \right), \tag{27}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$, $\boldsymbol{M} \sim \mathcal{B}(p_0)$ and $\mu = 2\sigma^2\lambda$. The function (27) is the cost function used for training two networks. See Figure 2 for the diagram of the work flow. In practice, we observed that at each iteration, just sampling one image instance (*i.e.*, $N = 1$) by the randomized image generative network performs well in our experiments. See Algorithm 1 for the outline of the algorithm for network training.

Recall that to have an accurate estimation of the truth image $\boldsymbol{x}$, we would like to have the prior distribution defined by the trained image generative network concentrated on the truth image $\boldsymbol{x}$ with small variance. See Figure 3 for the visualization of the images generated from the network trained after iteration $100, 1000$ and 5000 respectively. It can be seen that the images generated by the network trained with more iterations are more close to the truth images. Also, after sufficient iterations, these images generated from the network are well concentrated on the truth images without significant image artifacts (small variance). This illustration showed the approximation effectiveness of the proposed method to the truth prior distribution of the latent image $\boldsymbol{x}$.

Algorithm 1 Blind image deconvolution via approximate Monte-Carlo sampling

Input: Blurry image $\boldsymbol{y}$, regularization parameter μ, fixed input noise s.t.d σ_x, σ_k, injected noise s.t.d σ_0, dropout rate p_0, number of iteration T.

Output: Estimated sharp image $\boldsymbol{x}$ and blur kernel $\boldsymbol{k}$.

%% *Generate fix noise .*
$\boldsymbol{z}_x \sim \mathcal{N}(0, \sigma_x^2 \mathbf{I})$; $\boldsymbol{z}_k \sim \mathcal{N}(0, \sigma_k^2 \mathbf{I})$
for $t = 1 : T$ **do**
  $\boldsymbol{\epsilon}^t \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$, $\boldsymbol{M}^t \sim \mathcal{B}(p_0)$
  $\boldsymbol{\theta}_k^t = \text{Adam}(\boldsymbol{\theta}_k^{t-1}, \nabla_{\boldsymbol{\theta}_k} \mathcal{L})$, where $\mathcal{L}$ in (27)
  $\boldsymbol{\theta}_x^t = \text{Adam}(\boldsymbol{\theta}_x^{t-1}, \nabla_{\boldsymbol{\theta}_x} \mathcal{L})$, where $\mathcal{L}$ in (27)
end for
$\boldsymbol{x} = \mathcal{G}_x(\boldsymbol{z}_x; \boldsymbol{\theta}_k^T)$, $\boldsymbol{k} = \mathcal{G}_k(\boldsymbol{z}_k; \boldsymbol{\theta}_k^T)$

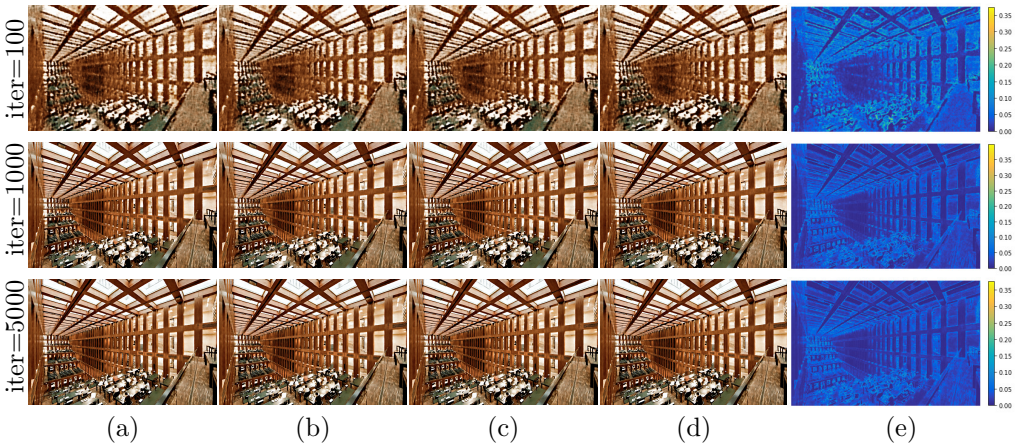


Figure 3. The image samples generated from the network $\mathcal{G}_{\boldsymbol{x}}$ trained after $100, 1000$ and 5000 iterations respectively. (a)-(d) show some image samples and (e) shows the standard deviation of 10 such samples.

2.3. Discussion on the treatment of kernel estimation

In the previous section, we present a MC-sampling-based approach to approximate the $\text{MAP}_{\boldsymbol{k}}$. The generative NN $\mathcal{G}_x$ is a deep network with random process which is trained to model the posterior distribution of $\boldsymbol{x}$. Different from the treatment for image, the generative network for the kernel $\boldsymbol{k}$ is deterministic.

From the derivation of the $\text{MAP}_{\boldsymbol{k}}$ estimator, there seems to be no need for training a network for predicting the kernel. One reason we train a network for predicting the kernel is for computational efficiency. Note that the integral involved in $\text{MAP}_{\boldsymbol{k}}$ is approximated by the MC method. While more samples will give better approximation accuracy, only few samples are used during the iteration for computational efficiency. As a result, direct calculation of the kernel will suffer from approximation error. By training a generative network for predicting the kernel, we can help alleviate such approximation error by learning the prediction from only few samples. In addition, there are two widely used physical constraints on the kernel:

- *Non-negativity:* $\boldsymbol{k}[n] \geq 0$ for all n.
- *Normality:* $\|\boldsymbol{k}\|_1 = 1$.

These two physical constraints can be implemented by appending a Softmax layer in the end of the NN, same as [43]. Therefore, there is no need to add such constraints on the cost function.

As the generative NN used for kernel is deterministic and the one for image is randomized, a natural question is then why not treat two in the same way, *e.g.* both NNs are randomized. Then, the image can also be estimated by marginalizing over the distribution of kernels, which gives the MAP estimator for image $\text{MAP}_{\boldsymbol{x}}$. The reason why we don't adopt such a strategy is that the dimension of the kernel space is much smaller than that of the image space. As a result, such a marginalization does not bring noticeable benefit, while it will significantly increase computational cost. In Section 3, an ablation study is conducted to show that there is no performance gain if we also randomize the NN for the kernel in the same way as we did for the image.

In summary, we present an un-supervised deep learning method for blind image deconvolution, which does not require any training samples with ground truth image. The proposed method is built on the $\text{MAP}_{\boldsymbol{k}}$ estimator which estimates the kernel by marginalizing over all images in its posterior distribution. In our approach, the $\text{MAP}_{\boldsymbol{k}}$ estimator is approximated by MC sampling on the approximating distribution modeled by a generative NN with random process. Based on such an MC sampling method, two generative networks for image and kernel are trained in an alternating iterative scheme. The corresponding cost function can be solved by standard deep learning solvers, *e.g.* Adam.

3. Experiments

This section is devoted to the performance evaluation of the proposed method, in comparison to existing solutions to the application of blind image deconvolution.

3.1. Implementation details

Both generative networks, $\mathcal{G}_x(\cdot, \boldsymbol{\theta}_x)$ for image and kernel $\mathcal{G}_k(\cdot, \boldsymbol{\theta}_k)$ for kernel, are based on U-Net and are jointly trained. For $\mathcal{G}_x(\cdot, \boldsymbol{\theta}_x)$, we implemented a U-net with 6 levels. Channel number was set to 128. For $\mathcal{G}_k(\cdot, \boldsymbol{\theta}_k)$, the U-net contained 4 levels with channel number 64. The physical constraints on blur kernel is implicitly implemented by appending a Softmax layer to the output of the U-net. Both NNs used LeakyReLU with slope 0.1; see supplementary file for more details. To make the image generative network enable sampling multiple image instances, we add a Gaussian noise layer to perturb the input $\boldsymbol{z}_x$. In addition, we also include a dropout layer before each re-scaling in the Decoder sub-network.

We set the fixed inputs $\boldsymbol{z}_x$, $\boldsymbol{z}_k$ sampling from the uniform distribution $\mathcal{U}(0, 0.1)$. We trained the NN for 5×10^3 epochs. For the generative NN for images $\mathcal{G}_x(\cdot, \boldsymbol{\theta}_x)$, we set the dropout rate $p_0 = 0.3$ at the first 1.5×10^3 epochs and drop to $p_0 = 0.001$ for later iteration. A decreasing dropout rate is for further improving computational efficiency. The motivation is that after the kernel is far away from the Dirac delta function, the randomness of kernel is not long necessary for keeping the network converging to the Dirac delta function.

For noise $\boldsymbol{\epsilon}$, its s.t.d. is set to $\sigma_0 = 0.05$ for larger kernel size or 0.01 for relative small kernel size initially and 0.001 after 3.5×10^3 iterations. The learning rate is 1×10^{-2} . For $\mathcal{G}_k(\cdot, \boldsymbol{\theta}_k)$, the learning rate was set to 1×10^{-4}. Both leaning rates

dropped with rate 0.5 after 3×10^3 steps. The parameter μ was set to be 1×10^{-4}. All experiments are conducted on a single NVIDIA Titan RTX GPU.

3.2. Dataset and protocol for performance evaluation

The experiments are evaluated on both synthesized dataset for quantitative evaluation and real-world dataset for qualitative evaluation. Quantitative evaluation is conducted in three benchmark datasets:

- Levin's dataset‡ with mild uniform blurring effect. Levin *et al.*'s dataset contains 32 images generated by convolving 4 clear images using 8 motion-blur kernels and adding Gaussian white noise with s.t.d. 1%. The size of these kernels is small, ranging from 11×11 to 27×27.
- Lai's dataset§ with severe blurring effect. Lai *et al.*'s [27] contains 100 blurry images falling into 5 categories and covers 4 different kernels whose size ranges from 31×31 to 75×75.
- Köhler's dataset‖ with non-uniform blurring effect. Köhler [22] contains 48 motion-blurred images. This dataset is produced by recording the samplings from the six-dimensional camera motion trajectory. Thus, the blurring of the images are not uniform. However, the variations of blurring effect are not very large. The size of dominant kernels ranges from 41×41 to 141×141.

Performance evaluation on real-world images are conducted on one dataset with real-world images:

- Lai *et al.* 's real-world dataset [27], which contains 100 real blurred images captured in the real-world scenarios with different cameras and settings. They are categorized into 5 attributes as the synthetic images form Lai's dataset [27].

As no ground truths are available for real-world images, only qualitative evaluation via visual inspection is possible on these images.

In comparison to non-blind image deconvolution, the focus of blind image deconvolution is about kernel estimation. A popular evaluation protocol is to measure the accuracy of the estimated kernel, which is done in a two-stage protocol. Firstly, the blind deconvolution algorithm for evaluation is called to estimate the kernel. Then, using the estimated kernel, some standard non-blind deconvolution method [30, 50] is called to deblur the image, whose quality is used for measuring the estimation accuracy of the kernel. However, many recent deep learning methods simply restore the blurred image without predicting the blur kernels, and thus we only can evaluate them using the output of these methods. In our experiments, when evaluating a method, the two-stage protocol is called whenever the kernel is available. For the method without outputting the kernel, its direct output of image is used for evaluation and the results are colored in blue. For the proposed method, both protocols are used for evaluation. Two metrics are used for quantitative evaluation: PSNR (peak signal-to-noise ratio) and SSIM (structural similarity index measure).

‡ `https://webee.technion.ac.il/people/anat.levin/papers/LevinEtalCVPR2011Code.zip`
§ `http://vllab.ucmerced.edu/wlai24/cvpr16_deblur_study/data/cvpr16_deblur_study_synthetic_dataset.zip`
‖ `http://people.kyb.tuebingen.mpg.de/rolfk/BenchmarkECCV2012/BlurryImages.zip`

Table 1. Average PSNR/SSIM of the results from different methods on Levin *et al.*'s dataset [30].

	Non-learning Methods							Supervised			Un-supervised			
Metric	Cho & Lee [8]	Xu & Jia [52]	Sun *et al.* [46]	Xu *et al.* [54]	Pan *et al.* [40]	Yan *et al.* [55]	Yang & Ji [56]	Pan *et al.* [38]	Tao *et al.* [47]	Kupyn *et al.* [26]	Ren *et al.* [43]	Ren *et al.* [43]	Ours	Ours
PSNR	30.79	31.74	32.38	29.93	32.69	31.28	32.04	30.42	25.97	25.7	33.32	33.07	<u>34.42</u>	**34.71**
SSIM	0.875	0.917	0.91	0.895	0.928	0.912	0.912	0.907	0.795	0.79	<u>0.943</u>	0.931	**0.948**	**0.948**

Input | Xu *et al.* [52] | Tao *et al.* [47] | Kupyn *et al.* [26] | Ren *et al.* [43] | Ours | Groundtruth

Figure 4. Deconvolution results for two images from Levin *et al.* [30]'s dataset with noise level 1 %.

3.3. Experiments on blind image deconvolution

Our method is compared to existing representative blind image deconvolution methods. Whenever possible, we directly cited the results from the literature. Otherwise, we used the pre-trained models from the authors to generate the results. If only code was available, we made effort to train it for optimal parameters. If none was available, we left it blank.

Experiments on Levin *et al.*'s dataset [30] with small kernel size In this experiments, totally 11 methods are included for comparison, including 7 non-learning methods, such as [8, 46, 52, 54], 3 supervised deep learning methods [26, 38, 47] and 1 unsupervised learning method [43]. See Table 1 for quantitative comparisons of the results from different methods. It can be seen that our method outperformed the second-best by around <u>0.64dB</u> in terms of PSNR, and by around <u>0.005</u> in terms of SSIM. The results showed that our method provides very competitive performance when processing motion-blurred images with small kernel sizes. See Figure 4 for the visual comparison of the results from several methods. It can be seen that the results from ours and Ren's have overall the best visual quality in terms of sharpness and artifacts. Overall, the proposed method provides the SOTA performance on the images with small kernel size.

Experiments on Lai *et al.*'s dataset [27] with large kernel size Following [27], the non-blind deconvolution method [23] is called in the evaluation protocol for all categories except "saturation", in which the method [50] with outlier handling is called. For this dataset, totally 17 methods are included for comparison. See Table 2 and 3 for the quantitative comparison of the results from different methods in terms

Table 2. Average PSNR of the results from different methods on Lai *et al.*'s dataset [27].

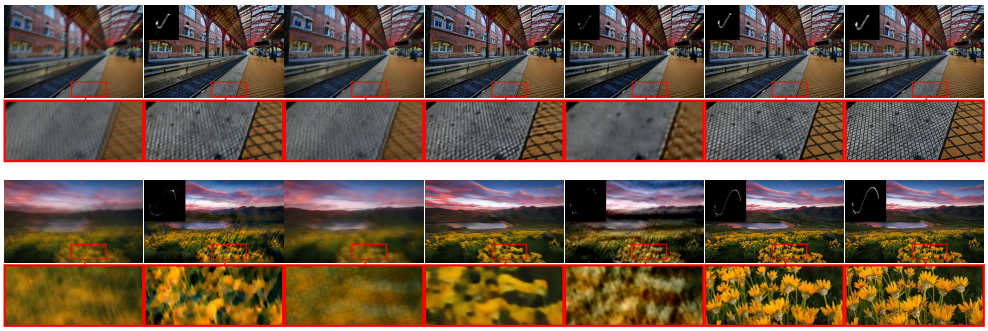
	Non-learning Methods							Supervised			Un-supervised			
Category	Cho & Lee [8]	Xu & Jia [52]	Sun *et al.* [46]	Xu *et al.* [54]	Pan *et al.* [40]	Yan *et al.* [55]	Yang & Ji [56]	Tao *et al.* [47]	Kupyn *et al.* [26]	Kaufman &Fattal [19]	Ren *et al.* [43]	Ren *et al.* [43]	Ours	Ours
Manmade	16.11	19.56	19.3	17.87	17.33	19.32	19.99	15.61	15.93	18.94	20.08	20.35	21.01	**23.06**
Natural	20.09	23.38	23.69	22.14	21.47	23.69	24.33	18.61	18.95	22.05	22.5	22.05	24.67	**26.00**
People	19.89	26.5	26.13	25.72	24.33	27.01	27.22	21	21.53	27.05	27.41	25.94	28.17	**31.02**
Saturated	14.23	15.59	14.95	15	15.11	16.46	17.04	13.78	13.79	15.18	16.58	16.35	16.63	**17.21**
Text	14.82	19.68	18.35	18.61	17.56	18.64	20.35	14.42	14.82	17.85	19.06	20.16	20.51	**25.46**
Average	17.03	20.97	20.48	19.87	19.16	21.02	21.79	16.68	17.04	20.22	21.13	20.97	22.20	**24.55**

Table 3. Average SSIM of the results from different methods on the dataset Lai *et al.* [27].

	Non-learning Methods							Supervised			Un-supervised			
Category	Cho & Lee [8]	Xu & Jia [52]	Sun *et al.* [46]	Xu *et al.* [54]	Pan *et al.* [40]	Yan *et al.* [55]	Yang & Ji [56]	Tao *et al.* [47]	Kupyn *et al.* [26]	Kaufman &Fattal [19]	Ren *et al.* [43]	Ren *et al.* [43]	Ours	Ours
Manmade	0.388	0.546	0.53	0.494	0.476	0.579	0.599	0.3	0.321	0.517	0.538	0.509	0.682	**0.751**
Natural	0.512	0.623	0.662	0.581	0.6	0.678	0.692	0.412	0.429	0.586	0.581	0.514	0.751	**0.774**
People	0.639	0.824	0.832	0.785	0.775	0.842	0.861	0.681	0.694	0.833	0.85	0.737	0.863	**0.902**
Saturated	0.474	0.532	0.531	0.518	0.537	0.588	0.605	0.488	0.488	0.599	0.654	0.52	0.651	**0.679**
Text	0.49	0.764	0.723	0.749	0.692	0.689	0.762	0.489	0.519	0.717	0.731	0.699	0.76	**0.892**
Average	0.501	0.658	0.656	0.625	0.616	0.675	0.704	0.474	0.49	0.65	0.671	0.596	0.741	**0.800**

of PSNR and SSIM. It can be seen that our method outperformed all other methods by a large margin, with about 2.7 dB advantage over the second-best performer. The results indicate that the proposed method can estimate the kernel of large size much more accurately than existing ones. It is not surprising to see such a significant performance gain on the estimation of the kernel of large size. Notice that in the case of large kernel size, the deterministic generative NN trained in Ren *et al.* [43] is prone to the convergence to the no-blur pair with kernel close to Dirac delta. Thanks to randomized NN trained in the proposed method, the proposed method provides a MC-based sampling approximation to the MAP estimator of the kernel, which effectively avoids the convergence to the no-blur pair. As a result, our approach outperformed Ren *et al.*'s method by a large margin. The improvement demonstrates the benefit of MC-based approximation to the $\text{MAP}_{\boldsymbol{k}}$ estimator. See Figure 5 for visual comparison of some results from different methods. More comprehensive visual comparison of the results from different methods can be found in the supplementary file. Overall, the results from the proposed method have the best visual quality among all, with sharper image details and less artifacts.

Experiments on Köhler *et al.*'s dataset [22] with non-uniform motion blurring The dataset from Köhler [22] is not exactly uniform blurred. The blurring effect of the images is can roughly viewed as a uniform blurring with mild variations all over the images. It is a dataset for evaluating the robustness of the uniform deblurring



Input | Xu *et al.* [52] | Kupyn *et al.* [26] | Kaufman & Fattal [19] | Ren *et al.* [43] | Ours | Groundtruth

Figure 5. Visual comparisons of two examples of the deconvolution results from the dataset Lai *et al.* [27]. First row comes from the "manmade" category and second row comes from the "natural" category.

method to small variations on blurring effect. For this dataset, totally 12 methods are included for comparison. See Table 4 for quantitative comparison of different methods in terms of PSNR and MSSIM¶. It can be seen that our proposed method is the best among all in terms of PSNR, and is close to the best performer in terms of MSSIM. See Figure 6 for visual comparison of sample results from different method. The experiment shows the robustness of the proposed method when being applied to deblur an image whose blurring effect is only approximately uniform.

Table 4. Average PSNR/MSSIM of the results from different methods on the dataset Köhler *et al.* [22].

	Non-learning Methods								Supervised			Self-sup.	
Metric	Cho & Lee [8]	Xu & Jia [52]	Whyte *et al.* [50]	Hirsch *et al.* [14]	Vasu & Ra -jagopalan [49]	Yan *et al.* [55]	Jin *et al.* [18]	Yang & Ji [56]	Tao *et al.* [47]	Kupyn *et al.* [26]	Kaufman & Fattal [19]	Ren *et al.* [43]	Ours
PSNR	28.98	29.53	28.07	27.77	29.89	28.57	29.61	29.22	27.06	26.97	<u>30.17</u>	25.85	**30.27**
MSSIM	0.933	<u>0.944</u>	0.848	0.852	0.927	**0.949**	N/A	N/A	0.84	0.83	0.915	0.792	0.936

3.4. Experiments on real images from Lai et al. *[27]*

As no ground truths are available for real images, only visual inspection is available. See Figure 7 for the illustration of some results from our methods on real image dataset [27]. More comprehensive visual comparison of the results from different methods can be found in the supplementary file.

3.5. Ablation study on MC sampling

This section is devoted to the ablation study on the proposed MC sampling method for blind image deconvolution. The main idea of the proposed method is randomizing the

¶ The comparison protocol for Köhler's dataset uses MSSIM instead of SSIM.

Input | Xu *et al.* [52] | Kupyn *et al.* [26] | Kaufman & Fattal [19] | Ren *et al.* [43] | Ours | Groundtruth

Figure 6. Visual comparisons of one example of the deconvolution results from the dataset Köhler *et al.* [22] from different methods.

Input | Output | Input | Output | Input | Output

Figure 7. Visual comparison on deconvolution results for real-world images from Lai *et al.*'s dataset [27].

generative network for the image to enable MC-sampling-based approximation to the MAP estimator of the kernel. This ablation study is for validating the performance gain brought by our training schemes.

Performance gain by the proposed MC sampling This study focuses on the performance gain introduced by the approximate Monte-Carlo sampling implemented via dropout and noise layer for input. See Table 5 for the results on the dataset Lai *et al.* [27]. It can be seen that, without dropout or noise layer, the performance will see a significant decrease. The results showed the effectiveness of the randomization to avoid the overfitting. In some categories, dropout layers made substantial contribution to the performance . In other categories, noise layer for input made substantial contribution to the performance.

Table 5. Ablation study on the proposed architecture in terms of PSNR/SSIM on Lai *et al.*'s dataset.

	w/o dropout & noise layer	w/ only drop-out	w/ only noise layer	Ours w/ both
Manmade	19.54/0.507	19.60/0.564	19.91/0.620	23.06/0.751
Natural	22.70/0.629	24.42/0.564	25.49/0.772	26.00/0.774
People	26.83/0.781	30.48/0.889	29.21/0.863	31.02/0.902
Saturated	15.71/0.566	16.58/0.619	17.35/0.677	17.21/0.679
Text	20.58/0.669	24.28/0.863	25.82/0.899	25.46/0.892
Average	21.07/0.630	23.07/0.726	23.56/0.766	24.55/0.800

Ablation study on decreasing dropout rate and noise variance The practice of using decreasing noise variance and decreasing dropout rate is based on empirical

observation through extensive experiments. In this study, we investigate how such a practice benefit the performance and efficiency of the proposed method. Totally six additional figurations are considered in this study, including (1) Only using dropout layer with rate 0.5; (2) Only using dropout layer with rate 0.001; (3) Only using dropout layer with varying rate from 0.3 to 0.001 at iteration 1500; (4) Only using input noise layer with standard deviation 0.05; (5) Only using input noise layer with standard deviation 0.001; (6) Only using input noise layer with varying standard deviation from 0.01 to 0.0001 at iteration 3500. These six figurations are compared to the configuration used in our method: the dropout rate is varying from 0.3 to 0.001 at iteration 1500 and the standard deviation of input noise is varying from 0.01 to 0.001.

See Figure 8 for the comparison of performance impact (in terms of PSNR) and of computational efficiency (in terms of decreasing speed of training loss), using the same network and same training hyper-parameters on a sample image from from Lai *et al.*'s dataset. The curves "S2" and "S5" showed that too small variance of two random sources leads to severe overfitting: small training loss but with large test error (small PSNR value). The curves "S1" and "S4" showed that large variance of either random sources provides reasonable testing performance but has a slower convergence rate. The curves "S3" and "S6" showed that the decreasing variance of either random sources provides a faster convergence rate and smaller testing error. The combination of both achieve the best in terms of both training efficiency and testing performance.

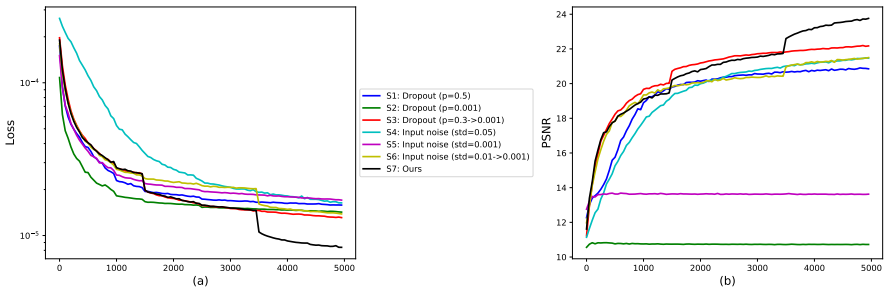
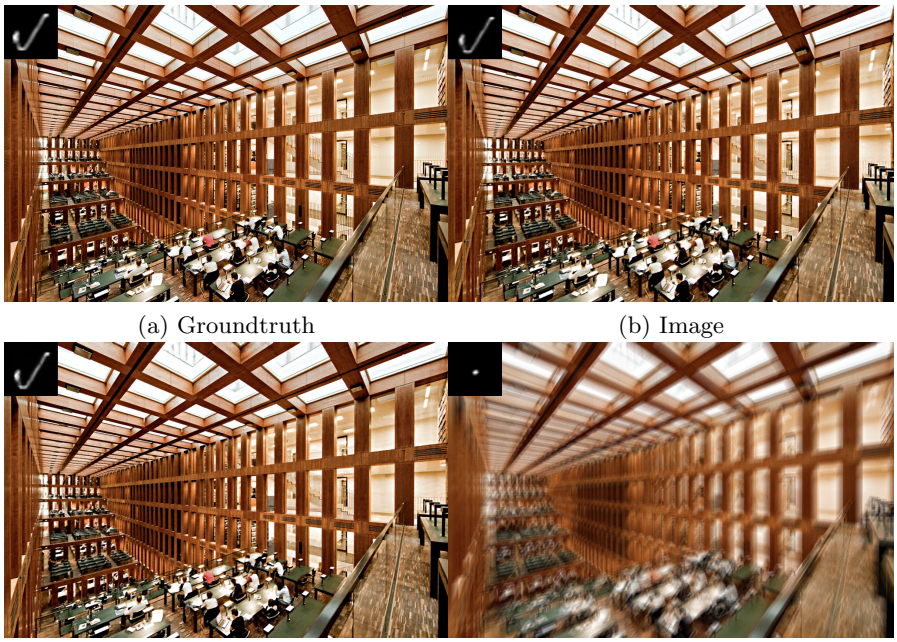


Figure 8. Comparison of the training efficiency and the testing performance of different configurations, in terms of iteration numbers. (a) The convergence rate, and (b) the testing performance in PSNR value

Ablation study on other randomization options In the proposed method, only the network for predicting the images is randomized. In this study, we investigate other randomization options. One is only randomizing the network for predicting the kernel. We also set up the dropout layers with dropout rate $p_0 = 0.1$ at the first 1.5×10^3 epochs and drop to $p_0 = 0.001$ for latter iteration, and add the noise layer $\boldsymbol{\epsilon}$ the same as the NN for predicting images in the proposed method. The average result is reported on a test image `manmade_01_kernel_01` from Lai *et al.*'s dataset with 10 trials. Another option is randomizing two networks for both image and kernel.

See Figure 9 for the illustration of the results from the proposed method and two other randomization options. It can be seen that in comparison to the proposed one which only randomizes the NN for image, the option of only randomizing the NN for predicting kernel suffers from the overfitting to the Dirac delta (All the 10 trails produce delta kernels). For another option which randomizes both NN, while it

provides comparable performance to the proposed method, it is less computationally efficient, see Figure 10 for the comparison of prediction accuracy (in PSNR and SSIM) over epochs. This study showed the soundness of the design of the proposed method in terms of both prediction accuracy and computational efficiency.

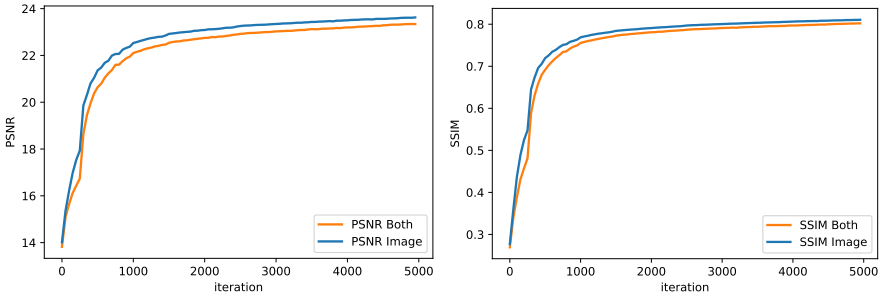


(a) Groundtruth (b) Image

(c) Both (d) Kernel

Figure 9. Results of three randomization options. (a) Groundtruth; (b) randomized image network; (c) randomized both networks; (d)randomized kernel network.

Applicability of MC-sampling technique on other network architectures The proposed MC-sampling technique is independent of network architectures. It is quite common to use the U-Net as the backbone for the generative NN of image. In the proposed method, the generative NN for kernel implemented by a CNN. There are



(a) (b)

Figure 10. Comparison of recovery progress of image averaged over 10 trials with two randomization options. (a) PSNR vs iteration (b) SSIM vs iteration.

other NN architectures for predicting kernel. For example, a fully connected network (FCN) is implemented in Ren *et al.*'s method [43] for predicting the kernel. In this study, we also applied the proposed method using the network architecture from [43]. See Table 6 for the study. It can be seen that the proposed MC sampling method also significantly boosted the performance of Ren *et al.*'s method [43]. The performance gap between the different implementation of the NN for kernel is relatively small. This study showed that the MC-sampling technique is the main contributor to the performance gain of the proposed method, and it is also applicable to other network architectures.

Table 6. The study on applying the proposed training scheme on Ren *et al.*'s [43] architecture with model size.

	Levin *et al.*	Lai *et al.*	Köhler *et al.*	# Param.
[43]	33.07/0,931	20.97/0.596	25.85/0.792	3.52M
[43] + MC	33.63/0.936	23.92/0.770	29.63/0.925	3.52M
Ours	34.71/0.948	24.55/0.800	30.27/0.936	2.65M

3.6. Comparison of running time

See Table 7 for the comparison of running time among existing blind deconvolution algorithms. It is noted that for a supervised learning method, it is time-consuming when training a model. However, once the model is trained, using it for processing an image is very fast. In comparison, our method needs to train a model for the image for processing. Thus, it falls into the category of traditional iterative methods.

It can be seen that the running time for processing an image is in the same category as many iterative optimization methods and existing self-supervised learning solutions. In comparison to supervised learning which can pre-trained a model and use it to process other images, the proposed method needs to train different networks for different images. One of our future research direction would be study how to to speed up the training process of an unsupervised learning method for blind image deblurring. One possible approach is using the scheme of meta-learning [10], which first learns a network using supervised learning and then refine the pre-trained model using an un-supervised learning method to make the model adaptive to the test image. Another possible approach is we introduce the so-called region selection technique used in existing non-learning method (*e.g.* [15]), which only select certain image regions for estimating the kernel. As the training efficiency depends on the size of input image, the incorporation of such a region selection module can speed up the training process as well.

4. Conclusion

Different from most existing deep-learning-based methods which are supervised over a large training dataset, this paper presented an un-supervised deep learning method for blind image deconvolution. The proposed method is built on the MC approximation to an $\text{MAP}_{\boldsymbol{k}}$ estimator via dropout and noise layer for the input. Despite the absence of truth images in training, the proposed method still provided SOTA performance on standard benchmark datasets of blind image deconvolution. The proposed technique

Table 7. Time comparison with existing blind deconvolution algorithms when processing a 256×256 image.

	Non-learning Methods						Supervised			Self-super.	
	Sun *et al.* [46]	Xu *et al.* [52]	Pan *et al.* [38]	Yan *et al.* [55]	Jin *et al.* [18]	Yang & Ji [56]	Tao *et al.* [47]	Kupyn *et al.* [26]	Kaufman & Fattal [19]	Ren *et al.* [43]	Ours
Time (s)	113.98	1.18	295.23	35.84	1242.97	354.03	0.21	0.24	0.17	235.84	273.12

has potential applications for solving other challenging non-linear inverse problems arising from imaging, where the collection of ground truth images is costly or challenging. In future, we will study how to further improve computational efficiency of the proposed method, as well as build its theoretical foundation.

References

[1] S Derin Babacan, Rafael Molina, Minh N Do, and Aggelos K Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *European conference on computer vision*, pages 341–355. Springer, 2012.

[2] Yuanchao Bai, Gene Cheung, Xianming Liu, and Wen Gao. Graph-based blind image deblurring from a single photograph. *IEEE Transactions on Image Processing*, 28(3):1404–1418, 2018.

[3] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019.

[4] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring from a single image using sparse approximation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 104–111, 2009.

[5] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Framelet-based blind motion deblurring from a single image. *IEEE Transactions on Image Processing*, 21(2):562–572, 2011.

[6] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, pages 221–235, 2016.

[7] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, 1998.

[8] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Transactions on graphics (TOG)*, 28(5):145, 2009.

[9] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[11] Yosef Gandelsman, Assaf Shocher, and Michal Irani. "Double-DIP": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11026–11035, 2019.

[12] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012.

[13] Dong Gong, Mingkui Tan, Yanning Zhang, Anton Van den Hengel, and Qinfeng Shi. Blind image deconvolution by automatic gradient activation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1827–1836, 2016.

[14] Michael Hirsch, Christian J Schuler, Stefan Harmeling, and Bernhard Schölkopf. Fast removal of non-uniform camera shake. In *2011 International Conference on Computer Vision*, pages 463–470. IEEE, 2011.

[15] Zhe Hu and Ming-Hsuan Yang. Good regions to deblur. In *European conference on computer vision*, pages 59–72. Springer, 2012.

[16] Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International conference on machine learning*, pages 2235–2244. PMLR, 2018.

[17] Hui Ji and Chaoqiang Liu. Motion blur identification from image gradients. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[18] Meiguang Jin, Stefan Roth, and Paolo Favaro. Normalized blind deconvolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–684, 2018.

[19] Adam Kaufman and Raanan Fattal. Deblurring using analysis-synthesis networks pair. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.

[22] Rolf Köhler, Michael Hirsch, Betty Mohler, Bernhard Schölkopf, and Stefan Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European conference on computer vision*, pages 27–40. Springer, 2012.

[23] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.

[24] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 233–240, 2011.

[25] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.

[26] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *IEEE International Conference on Computer Vision*, pages 8878–8887, 2019.

[27] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1709, 2016.

[28] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32:6970–6980, 2019.

[29] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *ICML*, 2018.

[30] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009.

[31] Lerenhan Li, Jinshan Pan, Wei-Sheng Lai, Changxin Gao, Nong Sang, and Ming-Hsuan Yang. Learning a discriminative prior for blind image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6616–6625, 2018.

[32] Yuelong Li, Mohammad Tofighi, Vishal Monga, and Yonina C Eldar. An algorithm unrolling approach to deep image deblurring. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7675–7679, 2019.

[33] Peidong Liu, Joel Janai, Marc Pollefeys, Torsten Sattler, and Andreas Geiger. Self-supervised linear motion deblurring. *IEEE Robotics and Automation Letters*, 5(2):2475–2482, 2020.

[34] Boyu Lu, Jun-Cheng Chen, and Rama Chellappa. Unsupervised domain-specific deblurring via disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10225–10234, 2019.

[35] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798, 2014.

[36] James Miskin and David JC MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in independent component analysis*, pages 123–141. Springer, 2000.

[37] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.

[38] Jinshan Pan, Jiangxin Dong, Yu-Wing Tai, Zhixun Su, and Ming-Hsuan Yang. Learning discriminative data fitting functions for blind image deblurring. In *IEEE International Conference on Computer Vision*, pages 1068–1076, 2017.

[39] Jinshan Pan, Risheng Liu, Zhixun Su, and Xianfeng Gu. Kernel estimation from salient structure for robust motion deblurring. *Signal Processing: Image Communication*, 28(9):1156–1170, 2013.

[40] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring

using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.

[41] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2043–2052, 2021.

[42] Daniele Perrone and Paolo Favaro. A clearer picture of total variation blind deconvolution. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1041–1055, 2015.

[43] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020.

[44] Wenqi Ren, Xiaochun Cao, Jinshan Pan, Xiaojie Guo, Wangmeng Zuo, and Ming-Hsuan Yang. Image deblurring via enhanced low-rank prior. *IEEE Transactions on Image Processing*, 25(7):3426–3437, 2016.

[45] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, 2015.

[46] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography*, pages 1–8, 2013.

[47] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[48] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[49] Subeesh Vasu and AN Rajagopalan. From local to global: Edge profiles to camera motion in blurred images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4447–4456, 2017.

[50] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Deblurring shaken and partially saturated images. *International journal of computer vision*, 110(2):185–201, 2014.

[51] David Wipf and Haichao Zhang. Revisiting bayesian blind deconvolution. *Journal of Machine Learning Research (JMLR)*, 2014.

[52] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, pages 157–170, 2010.

[53] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014.

[54] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural ℓ_0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013.

[55] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4003–4011, 2017.

[56] Liuge Yang and Hui Ji. A variational EM framework with adaptive edge selection for blind motion deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2019.

[57] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020.