Self-Supervised Blind Image Deconvolution via Deep Generative Ensemble Learning

Mingqin Chen, Yuhui Quan*, Yong Xu and Hui Ji

***Abstract*—Blind image deconvolution (BID) is about recovering a latent image with sharp details from its blurred observation generated by the convolution with an unknown smoothing kernel. Recently, deep generative priors from untrained neural networks (NNs) have emerged as a promising deep learning approach for BID, with the benefit of being free of external training samples. However, existing untrained-NN-based BID methods may suffer from under-deblurring or overfitting. In this paper, we propose an ensemble approach to better exploit the priors from untrained NNs for BID, which aggregates the deblurring results of multiple untrained NNs for improvement. To enjoy both the effectiveness and computational efficiency in ensemble learning, the untrained NNs are designed with a specific shared-base and multi-head architecture. In addition, a kernel-centering layer is proposed for handling the shift ambiguity among different predictions during ensemble, which also improves the robustness of kernel prediction to the setting of the kernel size parameter. Extensive experiments show that the proposed approach noticeably outperforms both exiting dataset-free methods and dataset-based methods.**

***Index Terms*—Blind image deconvolution, Ensemble learning, Image Deblurring, Dataset-free learning**

I. Introduction

Image restoration is to recover a degraded image back to its original form (*i.e.* the latent sharp image). There are many types of image degradation in real-world scenarios. One often-seen type is image blurring which erases the details of the latent image. Usually, the uniform blurring effect on an image is modeled by a convolution process:

$$\boldsymbol{Y} = \boldsymbol{K} \otimes \boldsymbol{X} + \boldsymbol{N}, \tag{1}$$

where $\boldsymbol{Y}$ and $\boldsymbol{X}$ denote the degraded image and latent image, respectively, $\boldsymbol{K}$ is a smoothing kernel, $\boldsymbol{N}$ is the measurement noise, and $\otimes$ is the discrete 2D convolution operation. When $\boldsymbol{K}$ is unknown, the task to recover $\boldsymbol{X}$ from $\boldsymbol{Y}$ by solving (1) is called blind image deconvolution (BID). One application of BID in digital photography is removing the motion blur caused by the camera shake dominated by image-plane translations. In this case, the convolution kernel is determined by the motion trajectory and motion speed of the camera shake during shutter time, which varies among different images.

Mingqin Chen, Yuhui Quan and Yong Xu are with the School of Computer Science and Engineering at South China University of Technology, Guangzhou 510006, China, and with Pazhou Lab, Guangzhou 510335, China. (email: csmingqinchen@mail.scut.edu.cn; csyhquan@scut.edu.cn; yxu@scut.edu.cn).

Hui Ji is with the Department of Mathematics at National University of Singapore, Singapore 119076 (email: matjh@nus.edu.sg).

Corresponding author: Yuhui Quan.

This work was supported in part by National Natural Science Foundation of China under Grants 61872151 and 62072188, in part by Natural Science Foundation of Guangdong Province under Grant 2022A1515011755, and in part by Singapore MOE AcRF under Grant R146000229114.

A. Solution Ambiguity in BID

BID is a challenging nonlinear inverse problem. Mathematically speaking, there are many sound solutions to it. For simplicity, we consider the noise-free case where

$$\boldsymbol{Y} = \boldsymbol{K} \otimes \boldsymbol{X}. \tag{2}$$

1) Under-deblurring: One class of solution ambiguity in BID comes from the kernel factorization on $\boldsymbol{K}$ given by

$$\boldsymbol{K} = \boldsymbol{K}_1 \otimes \boldsymbol{K}_2, \tag{3}$$

where $\boldsymbol{K}_1, \boldsymbol{K}_2$ are another two kernels. Then, we have

$$\boldsymbol{Y} = (\boldsymbol{K}_1 \otimes \boldsymbol{K}_2) \otimes \boldsymbol{X} = \boldsymbol{K}_1 \otimes (\boldsymbol{K}_2 \otimes \boldsymbol{X}). \tag{4}$$

In other words, $(\boldsymbol{K}_1, \boldsymbol{K}_2 \otimes \boldsymbol{X})$ is also a valid solution to (1). Such a factorization always exists. For instance, let $\boldsymbol{\delta}$ denote the delta kernel with only the center entry being one and other entries being zeros. Then, we have

$$\boldsymbol{K} = \boldsymbol{K}_1 \otimes \boldsymbol{K}_2, \quad \text{for} \quad \boldsymbol{K}_1 = \boldsymbol{\delta}, \boldsymbol{K}_2 = \boldsymbol{K}, \tag{5}$$

which gives a trivial solution with the pair of the kernel $\boldsymbol{\delta}$ and a blurred image $\boldsymbol{K} \otimes \boldsymbol{X}$. Another example is the image blurred by a Gaussian kernel. As a Gaussian kernel always can be factorized into two Gaussian kernels with smaller standard deviations and smaller sizes. Then, the image blurred by one of the smaller Gaussian kernels, paired with another smaller Gaussian kernel, is also a solution to (4).

2) Over-deblurring: Another class of solution ambiguity in BID occurs in the following form:

$$\boldsymbol{K} \otimes \boldsymbol{X} = (\boldsymbol{K} + \Delta\boldsymbol{K}) \otimes (\boldsymbol{X} + \Delta\boldsymbol{X}), \tag{6}$$

where $\Delta\boldsymbol{K}, \Delta\boldsymbol{X}$ are the errors on the kernel and on the image respectively, which are related by

$$\Delta\boldsymbol{K} \otimes \boldsymbol{X} + \boldsymbol{K} \otimes \Delta\boldsymbol{X} = -\Delta\boldsymbol{K} \otimes \Delta\boldsymbol{X}. \tag{7}$$

We can see that there are many solutions of $(\Delta\boldsymbol{K}, \Delta\boldsymbol{X})$ to (7). For instance, when imposing certain prior on the kernel for avoiding under-deblurred solutions, if the prior is not accurate, a small perturbation $\Delta\boldsymbol{K}$ will then yield a noticeable $\Delta\boldsymbol{X}$ which is often referred to as artifacts. Then, the result $\boldsymbol{X}+\Delta\boldsymbol{X}$ often appears over-deblurred.

B. Deep Learning for BID

In recent years, supervised deep learning has become one prominent tool for BID, which trains an NN on a dataset with many triplets of (blurred image, latent image, kernel) or pairs of (blurred image, latent image) to learn the priors on images

and/or kernels; see *e.g.* [1]–[5]. The model trained over the dataset is then used for predicting the latent image from an input blurred one. For supervised learning, the generalization performance of the resulting model is known to be determined by many aspects of training data, such as scale, quality, and distributional consistency to test data. In many scenarios, it is expensive to construct a large-scale, high-quality dataset with sufficient coverage of the statistical characteristics of test data, including images and kernels.

To avoid the cost and possible bias caused by an external training dataset in deep learning, there have been a few recent studies [6], [7] on developing dataset-free deep learning methods for BID which have no prerequisite on training datasets. These methods are built upon the so-called deep image prior (DIP) [8], [9] which states that, a convolutional NN (CNN) has implicit regularizations such that its output prioritizes regular structures over random noise when it is trained to fit an image with random seeds as input. Motivated by the practical value and potential of a dataset-free approach, this paper focuses on the dataset-free deep learning for BID.

C. Discussions on Existing DIP-Based BID Methods

One pioneering work exploiting DIP for BID with impressive results is the *SelfDeblur* proposed by Ren *et al.* [6], which uses two untrained NNs, denoted by $\mathcal{G}_{\boldsymbol{K}}, \mathcal{G}_{\boldsymbol{X}}$, as the generators (estimators) for the blur kernel and latent image respectively:

$$\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}}) \to \boldsymbol{K}, \ \mathcal{G}_{\boldsymbol{X}}(\boldsymbol{z}_{\boldsymbol{X}}) \to \boldsymbol{X}, \tag{8}$$

where $\boldsymbol{z}_{\boldsymbol{K}}, \boldsymbol{z}_{\boldsymbol{X}}$ denote two random seeds. Based on DIP, the two generators are optimized by

$$\min_{\mathcal{G}_{\boldsymbol{K}}, \mathcal{G}_{\boldsymbol{X}}} \|\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}}) \otimes \mathcal{G}_{\boldsymbol{X}}(\boldsymbol{z}_{\boldsymbol{X}}) - \boldsymbol{Y}\|_2^2, \tag{9}$$

such that $\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}}) \otimes \mathcal{G}_{\boldsymbol{X}}(\boldsymbol{z}_{\boldsymbol{X}})$ approximates $\boldsymbol{Y}$ well.

The DIP can effectively handle the solution ambiguity *w.r.t.* over-deblurring. Since $\mathcal{G}_{\boldsymbol{X}}$ tends to output the result with regular structures, it can exclude the error patterns of $\Delta\boldsymbol{K} \otimes \Delta\boldsymbol{X}$ with early stopping [8]. However, the DIP cannot handle the solution ambiguity *w.r.t.* under-deblurring where the estimate of the latent image has a lower blur degree. As DIP favors the output with simpler and smoother structures, the prediction of the model (9) will then be biased to the solution $(\boldsymbol{K}_1, \boldsymbol{K}_2 \otimes \boldsymbol{X})$, under the factorization (3). In the extreme case, the NN might lead to the prediction *w.r.t.* the trivial solution $(\boldsymbol{\delta}, \boldsymbol{Y})$. In [6], a total-variation (TV) regularization on the latent image, defined as $\|\nabla\boldsymbol{X}\|_1$, is added to the loss function (9) for regularizing the latent image. However, such a TV regularization cannot effectively address the bias to over-smoothed images [10].

D. Main Idea of Proposed Approach

In this paper, we propose a deep ensemble learning-based approach to exploit the priors from untrained NNs for the BID, which is more effective in resolving solution ambiguity than existing DIP-based methods. See Fig. 1 for an illustration of the basic idea of the proposed approach. Briefly, we do not consider the early stopping used in DIP to reduce the possibility of under-deblurring, but consider the following loss instead:

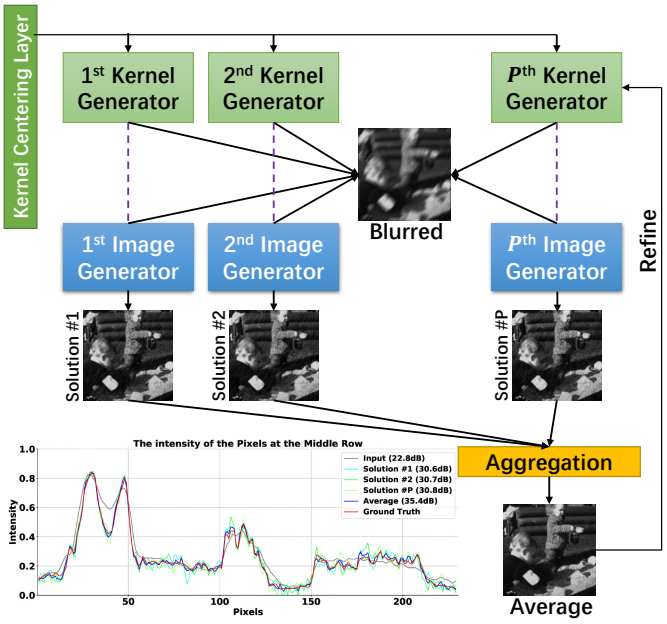


Fig. 1. Diagram of the basic idea of proposed deep generative ensemble learning approach for BID. The proposed approach learns several kernel/image generator pairs. The results of the image generators are aggregated for final prediction. The curve at the left-down corner plots the intensity of the pixels at the middle row for each predicted image, where we can see the aggregation leads to more than 4.7dB PSNR gain over individual results.

$$\min_{\mathcal{G}_{\boldsymbol{K}}, \mathcal{G}_{\boldsymbol{X}}} \|\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}}) \otimes \mathcal{G}_{\boldsymbol{X}}(\boldsymbol{z}_{\boldsymbol{X}}) - \boldsymbol{Y}\|_2^2 + \lambda\|\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}})\|_2^2, \tag{10}$$

where $\lambda\|\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}})\|_2^2$ is the squared ℓ_2-regularization on the kernel which is a common practice in traditional methods to avoid the solution converging to the delta kernel.

The regularization from $\lambda\|\mathcal{G}_{\boldsymbol{K}}(\boldsymbol{z}_{\boldsymbol{K}})\|_2^2$ indeed gives a biased kernel estimate $\boldsymbol{K}' = \boldsymbol{K} + \boldsymbol{U}$ with kernel errors $\boldsymbol{U}$, which will also result in an erroneous image estimate $\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{V}$ with artifacts $\boldsymbol{V}$. Then, setting $\lambda = \lambda_1, \cdots, \lambda_P$ respectively in (10) will result in a series of estimations denoted by

$$\boldsymbol{K}'_p = \boldsymbol{K} + \boldsymbol{U}_p, \ \boldsymbol{X}'_p = \boldsymbol{X} + \boldsymbol{V}_p, \ \text{for } p = 1, \cdots, P, \tag{11}$$

where $\boldsymbol{V}_p$ is the image artifacts caused by the kernel errors $\boldsymbol{U}_p$. The highly non-linear nature of NNs, together with the randomness in the NN's initialization and input, will introduce certain degrees of statistical independence among these artifacts. As a result, the aggregation of these estimations on the latent image is likely to suppress the portion of the artifacts that are statistically independent, which in turn leads to a refined deblurred image with less artifacts. An image with less artifacts will then benefit a more accurate estimation of the kernel as well. Note that the benefits from the ensemble is a specific advantage of deep models over traditional methods, as traditional methods (even setting various regularization parameters) lack sufficient randomness to generate independent solutions for effective aggregation.

Inspired by the discussion above, we propose to learn P pairs of (image, kernel) generators denoted by

$$(\mathcal{G}^p_{\boldsymbol{K}}, \mathcal{G}^p_{\boldsymbol{X}}) \ \text{for } p = 1, \cdots, P, \tag{12}$$

based on the basic training loss (10). The generator $\mathcal{G}_{\boldsymbol{X}}^{p}$ then produces different hypotheses on $\boldsymbol{X}$. Then, we can average the outputs of $\mathcal{G}_{\boldsymbol{X}}^{p}$ over p for a more accurate prediction. Note that learning multiple pairs of (image, kernel) generators independently can be computationally expensive. To have a computationally-efficient ensemble learning scheme, as well as to further improve the effectiveness, we propose a multi-head shared-base NN architecture, where the image generators are defined by the U-Nets with a shared encoder, and the kernel generators are modeled by the decoder NNs with shared front and middle layers as well as with a compact design.

In addition to solution ambiguity, there also exists the so-called *shift ambiguity*. That is, a shift on the latent image together with a reversed shift on the kernel will not change the convolution result:

$$\boldsymbol{K}(\cdot - \Delta h, \cdot - \Delta w) \otimes \boldsymbol{X}(\cdot + \Delta h, \cdot + \Delta w) = \boldsymbol{K} \otimes \boldsymbol{X}. \quad (13)$$

Such a shift ambiguity is not a concern in traditional methods, as the resulting image is the latent image with sharp details, up to a shift. However, it can be an issue when we aggregate multiple estimations of the latent image, if the image estimations are not aligned. In our approach, this shift ambiguity is addressed by introducing a kernel centering layer in the NN. It is empirically observed that such a kernel centering layer not only can handle the shift ambiguity, but also can reduce the sensitivity of our approach to the initialization of kernel support and to the overestimation of kernel sizes, leading to better kernel representation and performance improvement.

E. Contributions

This paper proposes an ensemble learning-based approach to effectively exploit the deep generative priors from untrained NNs for solving the problem of BID, which aggregates the implicit image priors and kernel priors from multiple untrained NNs to improve the estimation accuracy. The proposed approach shows state-of-the-art (SOTA) performance in extensive experiments, with better results than existing dataset-free methods (*i.e.* DIP-based or non-learning-based methods) and dataset-based deep learning methods. The technical contributions of this paper are listed as follows:

- Deep model ensemble with untrained NN priors is leveraged to handle better the solution ambiguity in BID than existing DIP-based methods, which results in a dataset-free deep learning solution with SOTA performance.
- An elaborate NN design with a multi-head and shared-base architecture is provided for image/kernel generators to improve the computational efficiency of the ensemble learning-based BID.
- A kernel centering layer is proposed to handle the shift ambiguity among predictions in ensemble and to improve the robustness to overestimation of kernel sizes.

II. Related Work

A. Non-Learning-Based BID

There have been plenty of non-learning-based methods for BID. A majority of them can be viewed as some maximum a posterior estimator with statistical priors on images or kernels, which can be formulated as some regularization model. The statistical image priors are varied in existing studies, such as (a) sparsity of latent sharp images in the form of total variation (TV) [11], graph TV [12], transform-specific ℓ_1-norm regularization [1], [13]–[15], normalized sparsity measured by ℓ_1/ℓ_2-norm ratios [10], or enhanced gradient sparsity [16]; (b) recurrence of image patches over space or scale, implemented by low-rank regularization on grouped similar parches [17], [18] or reconstruction via local self-example matching [19]; (c) dark/bright channel priors [20]–[22], which exploit the distributive difference of local extreme pixels under some channel measure between latent sharp images and their blurred versions; (d) local maximum gradient priors [23], [24] that the maximum value of a local patch gradient will diminish after blurring; and (e) segmentation prior [25] that sharp images contain more image details, which results in rougher superpixel segmentation boundaries measured by entropy. Edge selection/enhancement is one technique often called in these methods for better performance; see *e.g.* [26]–[29].

Another formulation for BID is built upon variational Bayes (VB) estimators. VB-based methods estimate the blur kernel by maximizing the marginalized distribution, *i.e.* seeking a kernel that is most likely with respect to the distribution of possible clear images; see *e.g.* [30]–[35]. The statistical image priors mentioned above are also applicable to VB-based methods. There are some studies working on other aspects of BID, such as the estimation of kernel size [36]. In comparison to all above methods, ours is learning-based which exploits the implicit priors encoded in untrained NNs and their ensemble for improvement, with a well-designed scheme to achieve the robustness to over-estimation of kernel sizes.

B. Supervised Deep Learning for BID

Many existing deep learning methods for BID train an NN using an organized dataset with latent images and true kernels. These methods can be divided into three categories. The first category learns an end-to-end NN that maps image patches to kernels; see *e.g.* [37], [38]. The second category unrolls some iterative scheme of a regularization-based BID model to construct the deep NN for BID, with explicit kernel estimation performed within the NN; see *e.g.* [1], [3], [4], [39]. The third category uses an NN to improve traditional non-learning-based methods by providing better image priors, initializations or parameters; see *e.g.* [2], [5].

There are also methods training blurring-model-free NNs to directly predict the blurred images to the corresponding latent images; see *e.g.* [2], [40]–[49]. By avoiding explicitly using the uniform blurring model, these methods are applicable to handling general non-uniform blur. However, their performance is usually not satisfactory when dealing with uniform blur.

C. Unsupervised Learning for BID

Generative adversarial networks (GANs) have been recently exploited for BID to relax the prerequisite on paired training data of supervised learning. Xia *et al.* [50] proposed to train a GAN on the pairs of blurry images. Lu *et al.* [51] proposed

to train GANs with the cycle consistency loss on unpaired images from two domains, *i.e.*, the clear image domain and the blurred image domain. These GAN-based methods are usually designed for domain-specific BID, *e.g.*, face/text images.

Unlike GAN-based methods, a few studies exploited the generative image priors from untrained NNs for "zero-shot" self-supervised learning of BID. Based on DIP [8], Ren *et al.* [6] proposed SelfDeblur using two untrained NN-based generators learned via a TV-regularized self-supervised loss, where a CNN and and a multi-layer perceptron (MLP) are used as the image and kernel generators respectively. SelfDeblur achieved impressive results on general natural images, even only using the input blurry image itself for learning. A refined version of SelfDeblur has also been released on GitHub [52]. It includes an additional training stage using the structural similarity index (SSIM) loss for improvement, with SOTA performance achieved. Later, Kotera *et al.* [53] improved SelfDeblur with better initialization and multi-scale processing. In parallel, Asim *et al.* [7] also proposed a DIP-based BID method with impressive performance on face images.

Without accessing external training data, SelfDeblur and its related methods [6], [7], [52] avoid the possible bias induced in an external dataset. In addition, they apply to different types of images in general. Same as SelfDeblur, the proposed approach uses two untrained NNs to represent the latent image and blur kernel, respectively. Different from [6], [7], [52], the proposed approach introduces the concept of the ensemble for a more accurate estimation of images and kernels. Several designs on the NN architecture are introduced for an effective implementation of such an ensemble concept, such as a different kernel centering layer for handling possible misalignment of different kernel estimates for a better ensemble, which also reduces the sensitivity to kernel support initialization, as well as a multi-head shared-based structure to reduce the computational cost of ensemble. It is worth mentioning that our ensemble scheme shares a similar spirit with the averaging scheme over different individual results of DIP [54], but is more effective for BID (as it considers the characteristics of BID) and computationally efficient (as it does not train multiple individual models).

III. Methodology

The overview of our NN architecture for ensemble learning-based BID is shown in Fig. 2. There are three main parts:

- A multi-head shared-base NN for image generators to have multiple predictions on the latent image.
- A multi-head shared-base NN for kernel generators to have multiple predictions on the kernel.
- A kernel centering layer for handling the shift ambiguity existing in the predictions.

A. Architectures of Image Generators and Kernel Generators

To have an effective ensemble, the image generators and kernel generators are expected to yield the predictions with sufficient statistical independence. However, separately training multiple individual NNs for prediction can be very expensive in computation, as the training time will increase linearly with the ensemble size. To address this, we propose a shared-base and multi-head architecture for both the image generators and kernel generators.

The NN for image generators is a multi-head U-Net with one shared encoder and multiple decoders. The pair of the encoder and any decoder in the U-Net maps a random seed of size $H_{\boldsymbol{X}} \times W_{\boldsymbol{X}} \times 8$ to an image of size $H_{\boldsymbol{X}} \times W_{\boldsymbol{X}} \times C$, where $(H_{\boldsymbol{X}}, W_{\boldsymbol{X}})$ is the latent image size, and C is 1 for gray-scale images or 3 for color images. Concretely, the encoder maps the input seed to a feature cube of size $H/32 \times W/32 \times 48$. All the decoders have the same structure. Each decoder symmetrically maps the feature cube back to the original size. We use Sigmoid for the activations on the decoders' outputs to map the pixel values to $[0, 1]$ for general images and use LReLU (leaky rectified linear unit) for the activations on saturated images. Skip connections are inserted at each (encoder, decoder) block pair. See Table I for more details. In implementation, we use group convolution for further acceleration.

The NN for kernel generators is a multi-head decoder with a shared base. The NN takes 1D random seeds in $\mathbb{R}^{200}$ as input and outputs different kernel estimates. The input seed is first mapped to a feature vector denoted by $\boldsymbol{v}$ via a fully-connected (FC) layer. To make the model compact yet with

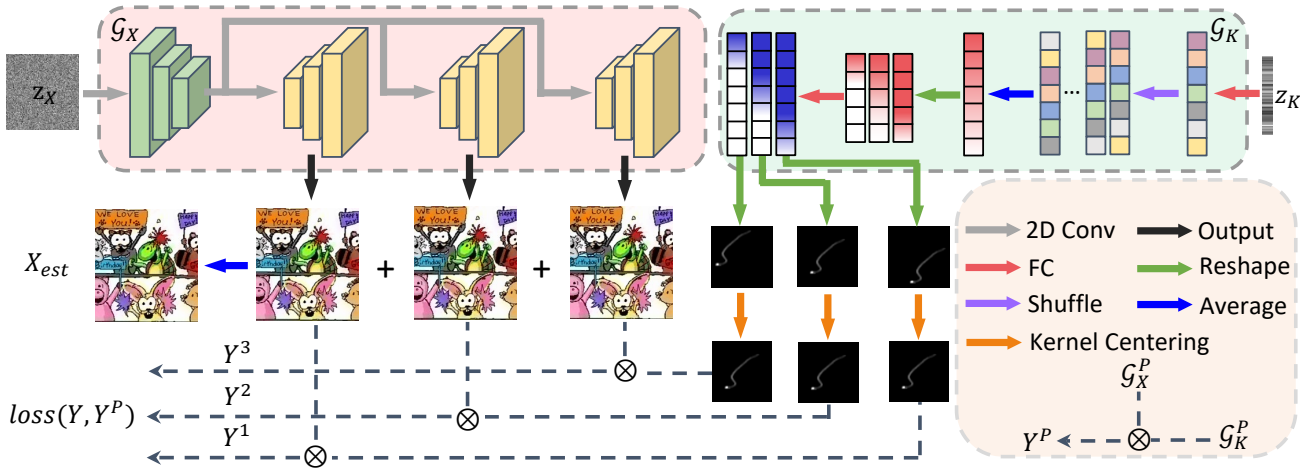


Fig. 2. Overview of proposed NN architecture for BID.

TABLE I

DETAILS OF $\mathcal{G}_{\boldsymbol{X}}$ AND $\mathcal{G}_{\boldsymbol{K}}$. 2D CONV: GROUP CONVOLUTION (3×3, STRIDE 1, GROUP NUMBER IS 1 FOR ENCODER AND 3 FOR DECODER). GELU: GAUSSIAN ERROR LINEAR UNITS [55]. MAX_POOL: MAX POOLING (2×2, STRIDE 2). $\uparrow_2$: 2X BI-LINEAR UPSAMPLING. FC: FULLY-CONNECTED LAYER.

	SN	Layer	#Channels	SN	Layer	#Channels	SN	Layer	#Channels
$\mathcal{G}_{\boldsymbol{X}}$	Encoder (Shared)			Decoder (3 Heads)			27	Concatenation (No.25, No.26)	144×3
	1		8	14	Replicate + Tile (No.12)	48×3	28	2D Conv + GeLU	96×3
	2	2D Conv + GeLU	48	15	$\uparrow_2$	48×3	29	2D Conv + GeLU	96×3
	3	2D Conv + GeLU	48	16	Replicate + Tile (No.10)	48×3	30	$\uparrow_2$	96×3
	4	Max_Pool	48	17	Concatenation (No.15, No.16)	96×3	31	Replicate + Tile (No.4)	48×3
	5	2D Conv + GeLU	48	18	2D Conv + GeLU	96×3	32	Concatenation (No.30, No.31)	144×3
	6	Max_Pool	48	19	2D Conv + GeLU	96×3	33	2D Conv + GeLU	96×3
	7	2D Conv + GeLU	48	20	$\uparrow_2$	96×3	34	2D Conv + GeLU	96×3
	8	Max_Pool	48	21	Replicate + Tile (No.8)	48×3	35	$\uparrow_2$	96×3
	9	2D Conv + GeLU	48	22	Concatenation (No.20, No.21)	144×3	36	Replicate + Tile (No.2)	48×3
	10	Max_Pool	48	23	2D Conv + GeLU	96×3	37	Concatenation (No.36, No.37)	144×3
	11	2D Conv + GeLU	48	24	2D Conv + GeLU	96×3	38	2D Conv + GeLU	72×3
	12	Max_Pool	48	25	$\uparrow_2$	96×3	39	2D Conv + GeLU	36×3
	13	2D Conv + GeLU	48	26	Replicate + Tile (No.6)	48×3	40	2D Conv + Sigmoid	$C \times 3$
$\mathcal{G}_{\boldsymbol{K}}$	SN	Function	Size	3	Replicate	999×5	6	Reshape	3×333
	1	Initialization	200×1	4	Shuffle	999×5	7	FC + SoftMax	$3 \times (H_{\bar{\boldsymbol{K}}} \cdot W_{\bar{\boldsymbol{K}}})$
	2	FC + GeLU	999×1	5	Mean	999×1	8	Reshape	$_3 \times H_{\bar{\boldsymbol{K}}} \times W_{\bar{\boldsymbol{K}}}$

sufficient expressibility, instead of applying a subsequent FC layer to $\boldsymbol{v}$ as what an MLP does, we introduce a shuffle layer that circularly shifts $\boldsymbol{v}$ multiple times with a fixed order randomly generated at the beginning, so as to generate multiple instances. These instances are averaged to have a new feature, which is then split into three parts. Such a series of operations corresponds to multiplying $\boldsymbol{v}$ with a structured spare matrix, by which the efficiency in terms of number of parameters and time can be improved over the dense matrix used by an FC layer. Afterwards, each part is then mapped to a kernel via a head FC layer. The last layer of each head is configured with the Softmax activation in order to impose two physical constraints on the estimated kernel [6]: $\sum_{h,w} \boldsymbol{K}(h,w) = 1$ and $\forall h, w, \boldsymbol{K}(h,w) \geq 0$. See Table I for more details. We also leverage group convolution for acceleration in practice.

B. Kernel Centering Layer for Handling Shift Ambiguity

Recall that there exists shift ambiguity in BID as shown in (13). The multiple generators can have the image estimates and kernel estimates with different centers. Without special treatments, the shift ambiguity can cause significant errors in ensemble-based prediction. We address this by proposing the so-called *kernel centering (KC) layer*.

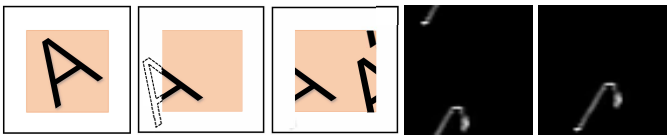
The KC layer is based on a spatial shifting operator implemented in the frequency domain. Let $(h_{\bar{\boldsymbol{K}}}, w_{\bar{\boldsymbol{K}}})$ denote the centroid of a kernel $\bar{\boldsymbol{K}} \in \mathbb{R}^{H_{\bar{\boldsymbol{K}}} \times W_{\bar{\boldsymbol{K}}}}$:

$$(h_{\bar{\boldsymbol{K}}}, w_{\bar{\boldsymbol{K}}}) = \frac{\sum_{h,w} \bar{\boldsymbol{K}}(h,w) \cdot (h,w)}{\sum_{h,w} \bar{\boldsymbol{K}}(h,w)}. \tag{14}$$

Recall that the center of $\bar{\boldsymbol{K}}$ lies at $(\frac{1}{2}H_{\bar{\boldsymbol{K}}}, \frac{1}{2}W_{\bar{\boldsymbol{K}}})$. Then, the KC layer aligns $\bar{\boldsymbol{K}}$ by aligning its centroid of to its center, *i.e.*, circularly shifting it with the offset $(\frac{1}{2}H_{\bar{\boldsymbol{K}}} - h_{\bar{\boldsymbol{K}}}, \frac{1}{2}W_{\bar{\boldsymbol{K}}} - w_{\bar{\boldsymbol{K}}})$, where the circular shift ensures information-losslessness. Since the shifting operation defined in the spatial domain is non-differentiable, we define the KC layer in the frequency domain:

$$\begin{aligned} \boldsymbol{K}(h - h_{\bar{\boldsymbol{K}}} \bmod H_{\bar{\boldsymbol{K}}}, w - w_{\bar{\boldsymbol{K}}} \bmod W_{\bar{\boldsymbol{K}}}) = \\ \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{K}(h,w)) e^{i2\pi(\frac{hh}{H_{\bar{\boldsymbol{K}}}} + \frac{ww}{W_{\bar{\boldsymbol{K}}}})}), \end{aligned} \tag{15}$$

where $\mathcal{F}, \mathcal{F}^{-1}$ denote the discrete Fourier transform (DFT) and inverse DFT respectively, and i is the imaginary symbol. Clearly, the shifting operation defined in the frequency domain is differentiable, and back-propagation is now applicable.



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

Fig. 3. Illustration of benefit of KC layer. (a) Support of a kernel; (b) Kernel support exceeding boundaries; (c) Kernel support when using circular shift; (d) An estimated kernel before KC; (e) Estimated kernel after KC.

The KC layer not only resolves the shift ambiguity, but also makes our approach insensitive to the initialization of the kernel support. In practice, the true kernel size is unknown, and one often used strategy is setting a larger size for safety. Due to the shift ambiguity, when the estimated kernel has its mess noticeably away from the center and crossing over the boundary of the array, the part outside the boundary cannot be used for estimating the image; see Fig. 3 for an illustration. Thus, it will lead to a very erroneous estimation, which then will be very hard to correct in the later stage as observed empirically. In traditional regularization-based methods, the issue caused by the shift ambiguity is often handled by taking a multi-scale approach and initializing the kernel with its mess in the center, or by centering the kernel in the spatial domain after each estimation. Such a practice cannot be used in an NN-based generator which takes a random initialization and requires all functions are differentiable for back-propagation. As a result, DIP-based methods such as [6], [7] require the kernel size to be known in their implementations.

Different from [6], [7], the KC layer can handle this issue effectively without knowing the exact kernel size. See Fig. 3 for an illustration. As circular shifting is applied to the kernel estimate, when the support of the kernel estimate partially goes out of the boundary (*e.g.* left), the part outside the boundary will not vanish, but just goes into the other side (*e.g.* right) of

the kernel. In other words, the support of the kernel outside the boundary is not an issue after centered back to proper place and can still be utilized in our case as the normal ones. As a result, our approach is not sensitive to the initialization of kernel support. Such an advantage enables one to use an universal model for different kernel sizes within some range.

C. Learning and Prediction

Similar to many existing regularization-based methods, the prediction error is measured in both the image domain and the gradient domain in our approach. We adopt eight high-pass filters from B-spline wavelet frame [56] for measuring image gradients with different orientations and different differentiation orders. Let $\mathbb{H} = \{\boldsymbol{h}_{j_1}\boldsymbol{h}_{j_2}^\top\}_{0\le j_1,j_2\le 2} \setminus \{\boldsymbol{h}_0\boldsymbol{h}_0^\top\}$ denote a set of eight 2D high-pass filters constructed by the tensor product of following 1D filters: $\boldsymbol{h}_0 = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]^\top, \boldsymbol{h}_1 = [-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}]^\top, \boldsymbol{h}_2 = [\frac{\sqrt{2}}{4}, 0, -\frac{\sqrt{2}}{4}]^\top$. Recall that $\mathcal{G}^p_{\boldsymbol{X}}$, $\mathcal{G}^p_{\boldsymbol{K}}$ denote the p^{th} image generator and the p^{th} kernel generator. The training is done with the loss below:

$$\begin{aligned}\mathcal{L} := &\sum_p \sum_{\boldsymbol{H}\in\mathbb{H}} \|\boldsymbol{H}\otimes\boldsymbol{Y} - \boldsymbol{H}\otimes\mathcal{G}^p_{\boldsymbol{K}}\otimes\mathcal{G}^p_{\boldsymbol{X}}\|_2^2 \\ &+\gamma\|\boldsymbol{Y} - \mathcal{S}(\mathcal{G}^p_{\boldsymbol{K}}\otimes\mathcal{G}^p_{\boldsymbol{X}})\|_2^2 + \lambda_p\|\mathcal{G}^p_{\boldsymbol{K}}\|_2^2,\end{aligned} \tag{16}$$

where $\mathcal{S}$ is the thresholding operator for simulating possible over-saturation:

$$\mathcal{S} : \boldsymbol{X}(h,w) \to \min(\max(\boldsymbol{X}(h,w), 0), 1). \tag{17}$$

Recall that the range of $\boldsymbol{X}(h,w)$ from LReLU is not constrained to $[0,1]$ which simulates the high range of image pixel values. Then the projection done by $\mathcal{S}$ is shrinking $\boldsymbol{X}(h,w)$ which may generate saturated pixels. To further enhance the diversity among the generators, we vary λ_p over p and set $(\lambda_1,\lambda_2,\lambda_3) = (\lambda-\Delta\lambda, \lambda, \lambda+\Delta\lambda)$. Note that the first term in (16) does not include $\mathcal{S}$, as the high-pass filter $\boldsymbol{H}$ can eliminate saturated regions from the loss owing to their gradients are often close to 0. We introduce a 2-stage scheme for training: first train using $\gamma = 0$ with sufficient iterations for concentrating the kernel estimation on image edges, and then train with $\gamma > 0$ for including low-frequency image components into consideration. For better handling image boundaries, the loss on boundary pixels is only counted for the image generators and is ignored for the kernel generators.

Since each generator is an over-parameterized model, its prediction is sensitive to the initialization of NN's parameters. Together with the varied regularization weights for different generators and separate gradient back-propagation on each head during training, our scheme suffices to make the generators converge to different results empirically. Afterwards, we aggregate the output images from different image generators as the final estimate of the latent image via averaging. Recall that with the use of the KC layer, the images output by different generators are likely to align with each other well. For further improvement, we apply half-pixel alignment within 3-pixel offset to those output images before averaging. Formally, the final estimate of the latent image $\boldsymbol{X}_{\text{est}}$ is given by

$$\boldsymbol{X}_{\text{est}} = \frac{1}{P}\sum_{p=1}^{P} \mathcal{A}_{\mathcal{G}^{p_0}_{\boldsymbol{X}}}(\mathcal{G}^p_{\boldsymbol{X}}(\boldsymbol{z}^p_x)), \tag{18}$$

where $\mathcal{A}_{\mathcal{G}^{p_0}_{\boldsymbol{X}}}$ denotes the half-pixel image alignment operation towards the image from $\mathcal{G}^{p_0}_{\boldsymbol{X}}$. Once having $\boldsymbol{X}_{\text{est}}$, we fine-tune the kernel generators via

$$\min_{\mathcal{G}^p_{\boldsymbol{K}}} \|\mathcal{G}^p_{\boldsymbol{K}}\otimes\boldsymbol{X}_{\text{est}} - \boldsymbol{Y}\|_2^2, \text{ for } p = 1,\cdots,P, \tag{19}$$

with a few iterations. The final predicted kernel is given by

$$\boldsymbol{K}_{\text{est}} = \frac{1}{P}\sum_{p=1}^{P}\mathcal{G}^p_{\boldsymbol{K}}. \tag{20}$$

While there are other schemes for aggregating the predictions of different generators, empirically the simple average operation suffices to achieve satisfactory performance.

IV. Experiments

A. Protocols and Implementation

1) Protocols for quantitative evaluation: Following most existing works, for a BID method with explicit kernel output, we first run it to have an estimated kernel. Its performance is then evaluated based on the quality of the deblurred image from an existing nonblind image deconvolution method using the estimated kernel. For a BID method without explicit kernel estimation, we use its deblurred images for evaluation. As for the proposed approach, both cases are covered to provide a better understanding on its performance.

Some representative BID methods are selected for comparison, including non-learning regularization-based methods [10], [11], [15], [20]–[22], [26], [27], [29], [33], [35], [57], [58] and deep-learning-based methods [1], [6], [37], [43], [44], [49], [59]. The results are reported with the following priority: quoted from existing literature, generated by the pre-trained model from author(s), and generated by the model trained with its published code. If these conditions cannot be satisfied, we will exclude that method in comparison. A closely-related competitor to our approach is SelfDeblur [6] which is also based on untrained NN priors and showed SOTA performance. For this competitor, we not only report its results published in its original work, but also report the results obtained from its refined version published on GitHub [52], which is denoted by †SelfDeblur. The improved version of SelfDeblur [53] is not compared due to the lack of official released code.

2) Implementation details: We jointly train the image generators and kernel generators for 5000 epochs with the Adam optimizer. The first 3000 epochs are with $\gamma = 0$ to mimic edge-selection in traditional methods, and the last 2000 epochs are with $\gamma = 2e{-}2$. The learning rate is initialized to 10^{-4} for each kernel generator and 0.01 for each image generator, and all the learning rates are dropped with a rate of 0.5 after 3000 epochs. The random input seeds are sampled from the uniform distribution $\mathcal{U}(0, 0.1)$ and with perturbations drawn from the normal distribution $\mathcal{N}(0, 1e{-}3)$ during training. The weights $\lambda_1, \lambda_2, \lambda_3$ for the kernel prior terms in (16) are set to $0.05, 0.1, 0.15$ respectively. The image generated by $\mathcal{G}^2_{\boldsymbol{X}}$ is chosen as the alignment reference, *i.e.* $p_0 = 2$ in (18). The number of fine-tuning epochs for final kernel estimation is 100. Since the multi-head architecture involves additional convolution operations with the blur kernel, we

TABLE II
COMPARISON ON LEVIN *et al.*'S DATASET IN TERMS OF PSNR(DB) AND SSIM. THE FIRST (SECOND) ROW CONTAINS NON-LEARNING-BASED (LEARNING-BASED) METHODS. BEST (SECOND BEST) RESULTS ARE BOLDFACED (UNDERLINED). THE METHODS CALLING AN ADDITIONAL NONBLIND SOLVER TO DEBLUR IMAGES WITH ESTIMATED KERNELS ARE MARKED BY *. THE NONBLIND SOLVER IS FIXED TO BE THE ONE OF [33].

Metric	Cho & Lee* [57]	Xu & Jia* [26]	Levin *et al.** [33]	Sun *et al.** [27]	Perrone & Favaro* [11]	Pan *et al.** [20]	Gong *et al.** [29]	Yan *et al.** [21]	Pan *et al.** [58]	Yang & Ji† [35]	Wen *et al.** [22]
PSNR↑	30.79	31.74	31.09	32.38	30.64	32.69	34.07	31.28	32.96	32.04	32.91
SSIM↑	0.875	0.917	0.915	0.910	0.899	0.928	0.943	0.912	**0.961**	0.912	0.938

Metric	Chakrabarti [37]	Zuo *et al.* [1]	Pan *et al.* [59]	Kupyn *et al.* [43]	Cho *et al.* [49]	SelfDeblur* [6]	SelfDeblur [6]	†SelfDeblur* [6]	†SelfDeblur [6]	DEBID* [Ours]	DEBID [Ours]
PSNR↑	25.21	32.66	30.42	32.96	25.18	33.32	33.07	33.79	33.91	<u>34.21</u>	**34.92**
SSIM↑	0.785	0.933	0.907	0.961	0.773	0.943	0.931	0.928	0.934	0.946	<u>0.960</u>

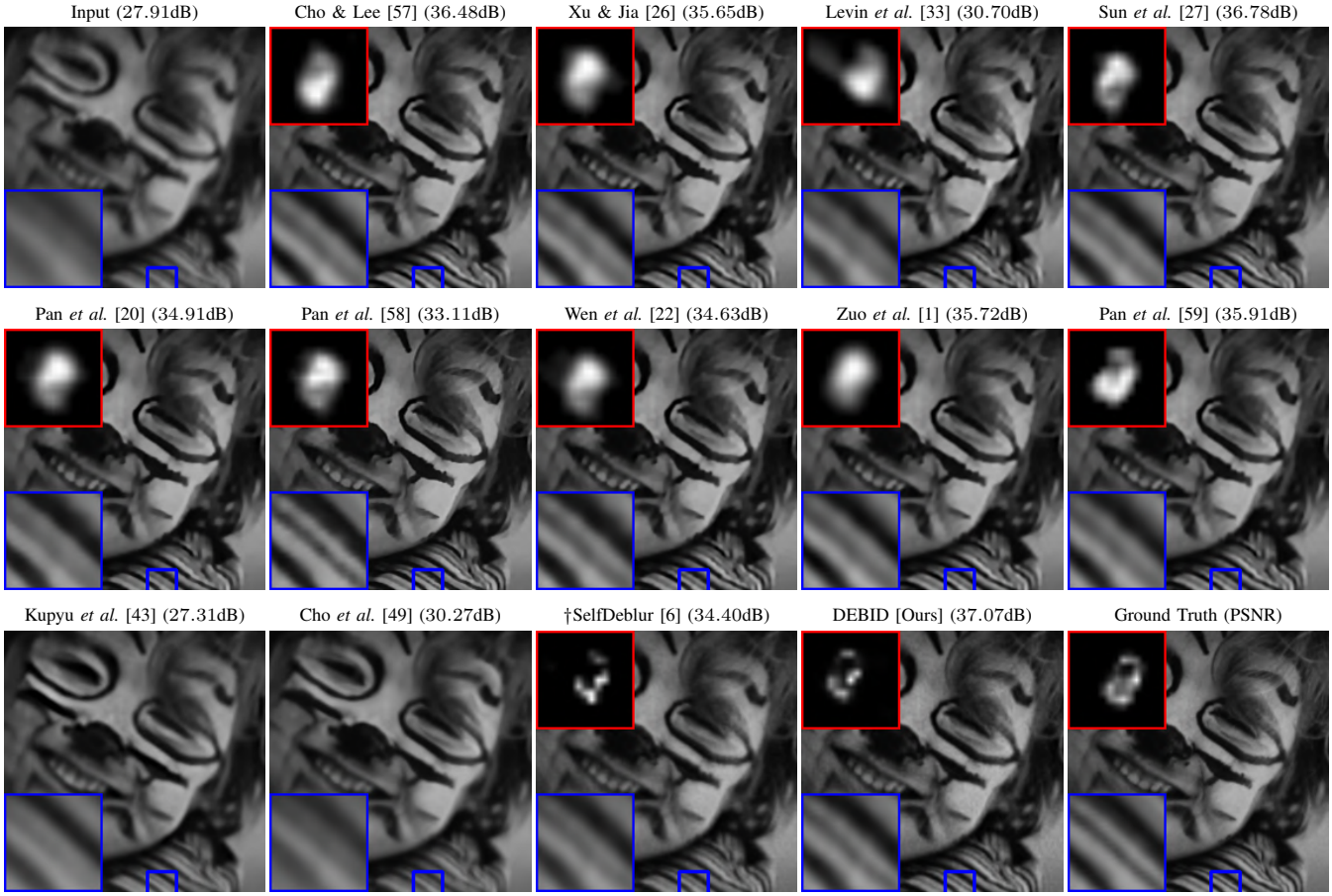


Fig. 4. Deblurring results of different methods on a sample image from Levin *et al.*'s dataset. The estimated kernels are shown if available.

use FFT-based convolution implementation to accelerate the computation. Such an implementation leads to faster speed on large images or large kernels. Our code will be released on our homepage upon paper's acceptance. For convenience, we name our approach as DEBID (Deep Ensemble-based BID).

B. Evaluation on Levin et al.'s Real Dataset [32]

The Levin *et al.*'s benchmark dataset [32] is a real motion deblurring dataset containing 32 blurry and sharp image pairs captured by a camera which is placed on a tripod with Z-axis rotation handle locked and with X/Y handles loosened. The sizes of these kernels are quite small, ranging from 11×11 to 25×25. In this experiment, the output sizes of the kernel generators in DEBID are fixed to 35×35 for all images. Following [1], [27], the non-blind deconvolution method [33] is called in the evaluation protocol.

See Table II for the quantitative results in terms of average PSNR and average SSIM over the deblurred images. There are totally 18 methods selected for comparison. It shows that DEBID outperforms all others in terms of PSNR. The PSNR gain of DEBID over the second best performer (*i.e.* †SelfDeblur) is around 0.61dB. Note that DEBID uses the same kernel size parameter for all images, while †SelfDeblur uses different kernel size parameters for different images. In terms of SSIM, DEBID is the second best with negligible performance gap to the best performer, Pan *et al.* [58]. See Fig. 4 for a visual

TABLE III
COMPARISON ON LEVIN *et al.*'S DATASET IN TERMS OF MSE AND MNC. BEST (SECOND BEST) RESULTS ARE BOLDFACED (UNDERLINED).

Metric	Cho & Lee [57]	Xu & Jia [26]	Krishnan *et al.* [10]	Levin *et al.* [33]	Sun *et al.* [27]	Perrone & Favaro [11]	Zuo *et al.* [1]	Pan *et al.* [20]	Yan *et al.* [21]	SelfDeblur [6]	DEBID [Ours]
MSE ↓	340.8	402.1	675.4	307.0	254.7	370.4	351.9	347.2	317.8	<u>142.1</u>	**127.7**
MNC ↑	0.864	0.881	0.826	0.869	0.931	0.865	0.904	0.894	0.859	<u>0.946</u>	**0.954**

TABLE IV
COMPARISON ON LAI *et al.*'S DATASET IN TERMS OF PSNR(DB)/SSIM. BEST RESULTS ARE BOLDFACED. THE METHODS CALLING AN ADDITIONAL NONBLIND SOLVER TO DEBLUR IMAGES WITH ESTIMATED KERNELS ARE MARKED BY *. THE NONBLIND SOLVER IS FIXED TO BE THE ONE OF [61].

Category	Cho & Lee* [57]	Xu & Jia* [26]	Krishinan *et al.** [10]	Sun *et al.** [27]	Xu *et al.** [15]	Perrone & Favaro* [11]	Pan *et al.** [20]	Yan *et al.** [21]	Yang & Ji* [35]	Wen *et al.** [22]
Manmade	16.11/0.388	19.56/0.546	15.67/0.435	19.30/0.530	17.87/0.494	17.53/0.497	17.33/0.476	19.32/0.579	19.99/0.599	18.80/0.606
Natural	20.09/0.512	23.38/0.623	19.24/0.576	23.69/0.662	22.14/0.581	22.08/0.615	21.47/0.600	23.69/0.678	24.33/0.692	23.40/0.775
People	19.89/0.639	26.50/0.824	21.34/0.634	26.13/0.832	25.72/0.785	24.04/0.781	24.33/0.775	27.01/0.842	27.22/0.861	25.09/0.845
Saturated	14.23/0.474	15.59/0.532	14.11/0.501	14.95/0.531	15.00/0.518	13.89/0.465	15.11/0.537	16.46/0.588	17.04/0.605	17.09/0.657
Text	14.82/0.490	19.68/0.764	15.11/0.597	18.35/0.723	18.61/0.749	16.80/0.654	17.56/0.692	18.64/0.689	20.35/0.762	16.89/0.567
All	17.03/0.501	20.97/0.658	17.09/0.549	20.48/0.656	19.87/0.625	18.87/0.602	19.16/0.616	21.02/0.675	21.79/0.704	20.27/0.690

Category	Pan *et al.* [59]	Kupyn *et al.* [43]	Kaufman & Fattal [44]	Cho *et al.* [49]	SelfDeblur* [6]	SelfDeblur [6]	†SelfDeblur* [6]	†SelfDeblur [6]	DEBID* [Ours]	DEBID [Ours]
Manmade	15.58/0.419	15.93/0.321	18.94/0.517	16.69/0.330	20.08/0.538	20.35/0.509	20.37/0.755	20.29/0.755	19.62/0.692	**22.14/0.803**
Natural	20.83/0.575	18.95/0.429	22.05/0.586	19.83/0.447	22.50/0.581	22.05/0.514	22.58/0.728	22.13/0.726	24.12/0.807	**26.18/0.894**
People	22.50/0.711	21.53/0.694	27.05/0.831	20.87/0.591	27.41/0.580	25.94/0.737	26.02/0.889	25.58/0.778	28.23/0.890	**31.25/0.923**
Saturated	13.60/0.490	13.79/0.488	15.18/0.599	15.16/0.506	16.58/0.654	16.35/0.520	16.56/0.648	15.97/0.504	17.12/0.692	**18.43/0.714**
Text	16.48/0.599	14.82/0.519	17.85/0.717	15.45/0.425	19.06/0.731	20.16/0.699	20.48/0.794	19.72/0.770	19.44/0.711	**23.00/0.822**
All	17.80/0.559	17.04/0.490	20.22/0.650	17.60/0.460	21.13/0.671	20.97/0.596	21.14/0.763	20.74/0.726	21.71/0.758	**24.20/0.831**

inspection of the deblurring results, where DEBID achieved the best visual quality among the compared methods, in terms of sharpness degree and artifact amount.

Following [6], we also run the performance evaluation on two other metrics on the accuracy of estimated kernels: the best aligned mean squared error (MSE) and the maximum of normalized convolution cross-correlation (MNC) [60]. The boundaries of an estimated kernel are cut to make it has the same size as the true kernel for evaluation. See Table III for the quantitative comparison with 10 methods. Our DEBID is the best performer among all the compared methods in terms of both metrics. To conclude, all above results have demonstrated that DEBID provides SOTA performance in processing motion-blurred images with small kernels.

C. Evaluation on Lai et al.'s Synthetic and Real Datasets [62]

The Lai *et al.*'s benchmark dataset [62] contains 100 blurry images classified into five categories. These images are synthesized using real blur kernels with sizes ranging from 31×31 to 75×75, which are larger than the kernels in Levin *et al.*'s dataset. We fix the output size of each kernel generator in DEBID to 55 when the true kernel size is less than that value. Otherwise, we set it to 81. Following the same procedure as [62], the non-blind deconvolution method [61] is called in the evaluation protocol for all categories except "Saturation", for which the method [63] with outlier handling is called.

Totally 16 methods are selected for comparison. See Table IV for the quantitative results measured in average PSNR and average SSIM of deblurred images. Again, DEBID is the best performer, with nearly 1dB PSNR advantage over the second best performer, †SelfDeblur. This clearly indicates that DEBID also provides SOTA performance in the case of large kernels. It is not surprising that DEBID outperforms existing methods by a large margin on large-size kernels. Recall that the larger the kernel size is, the more severe the solution ambiguity will be. Benefiting from the use of ensemble of deep priors, DEBID can better address such solution ambiguity. In addition, the proposed KC layer can make the learning of kernels more effective. These two techniques lead to the superior performance of DEBID. See also Fig. 5 and Fig. 6 for the visual comparison of some results. The deblurred image from DEBID is sharper with less artifacts compared to those of other methods in Fig. 5, and DEBID also works better than other methods in handing the saturated image in Fig. 6.

Lai *et al.*'s [62] also collected a number of real-world blurry images for test. As there is no truth image available for quantitative evaluation, we only provide visual comparison on some samples in Fig 7. It can be seen that DEBID achieved higher visual quality than other compared methods.

D. Evaluation on Complexity and Sensitivity

1) Comparison on complexity: The computational complexity of DEBID is compared with †SelfDeblur, the closely-related competitor. The parameter number of DEBID's model is about 3.12M while that of SelfDeblur is 3.76M. On an RTX 2080Ti GPU with auto mixed precision and parallel computation, our PyTorch code takes around 15.23 minutes for a 512x512 image under a 75x75 blur kernel, which is slightly slower than SelfDeblur. See Table V for a detailed comparison of running time using different image sizes and kernel sizes. Due to the FFT-based implementation, DEBID is even faster than SelfDeblur. We also replace the standard

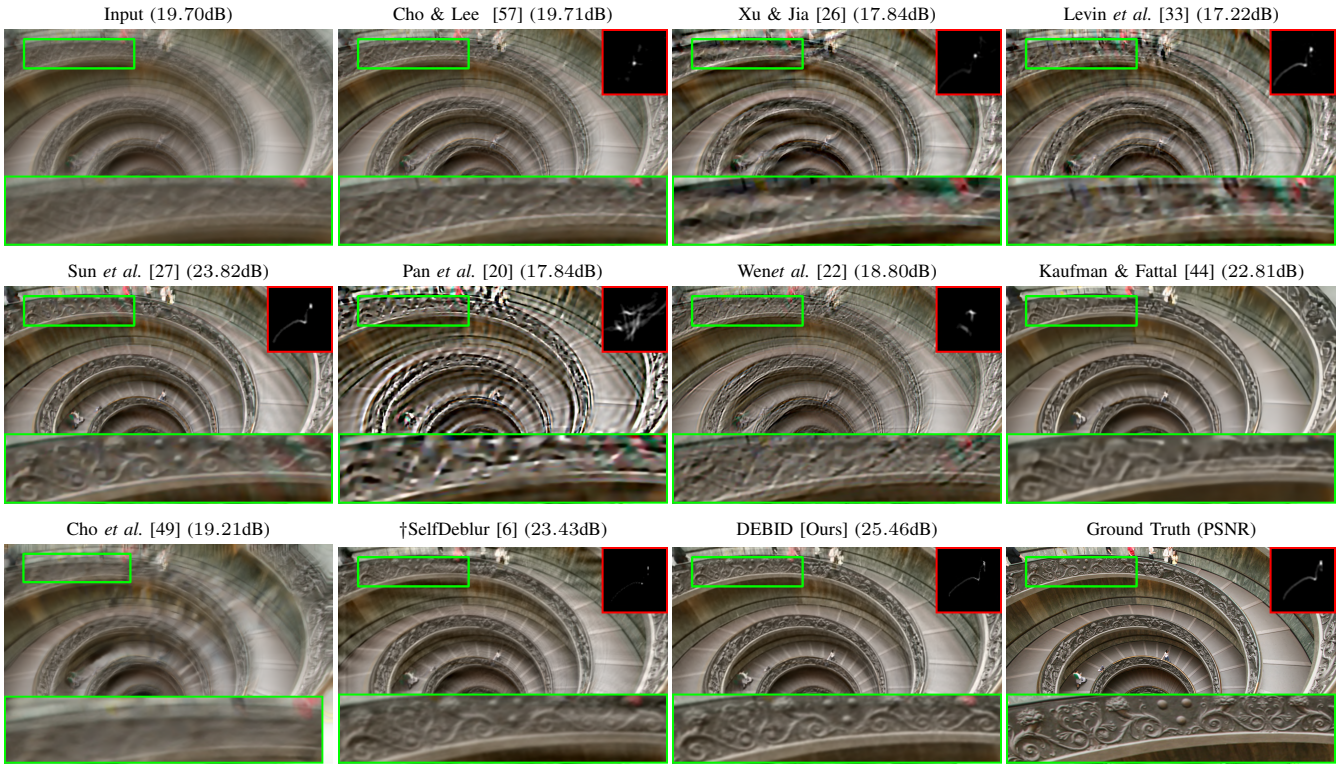


Fig. 5. Deblurring results of different methods on a sample image from Lai *et al.*'s dataset. The estimated kernels are shown if available.

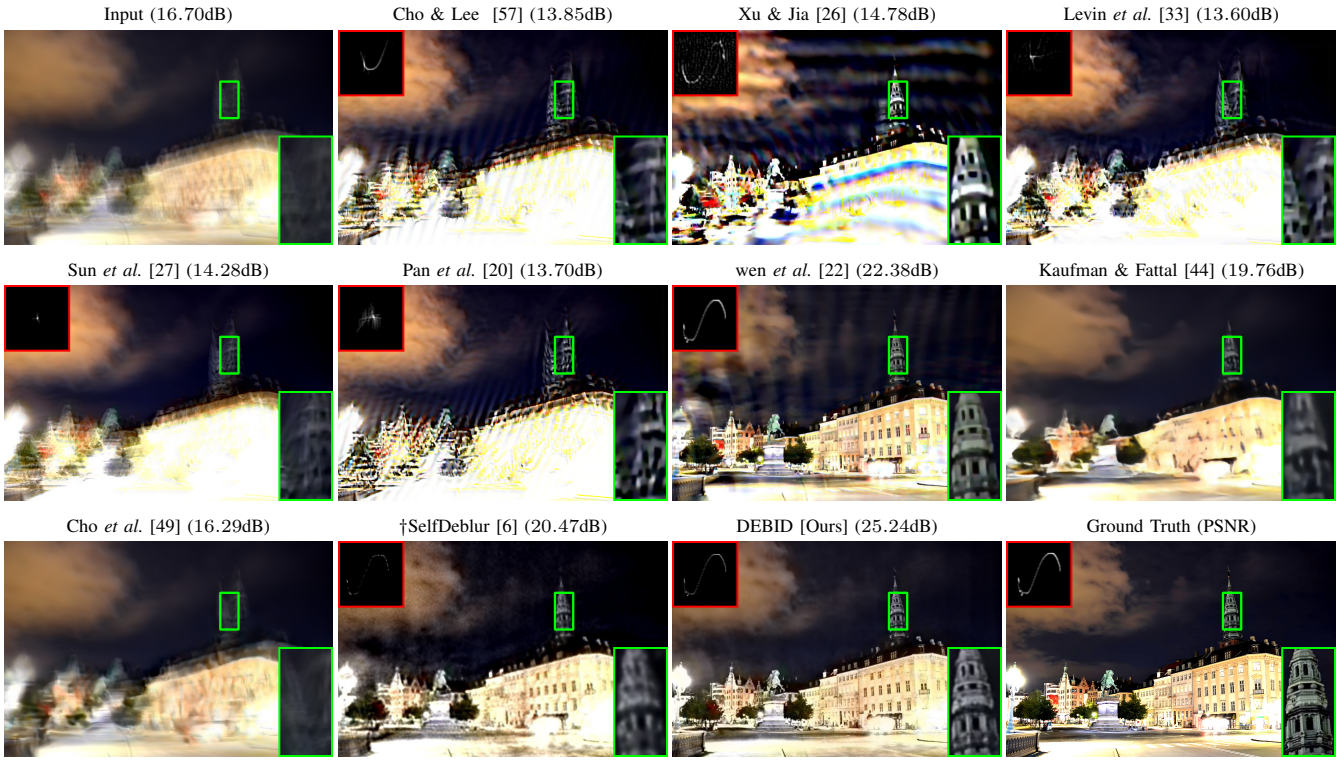


Fig. 6. Deblurring results of different methods on a saturated image from Lai *et al.*'s dataset. The estimated kernels are shown if available.

Fig. 7. Deblurring results of different methods on real blurry images from Lai *et al.*'s real-world dataset.

convolution in SelfDeblur with the FFT-based implementation for a fairer comparison. Even in such a case, the running time of DEBID is less than 1.51 times of SelfDeblur. Considering the number of generator trebles compared to SelfDeblur, such a time cost is not high yet acceptable, which also demonstrates the efficiency of the proposed NN architecture.

2) Study on the sensitivity to the kernel size parameter: The setting of kernel size parameter plays an important role when processing images of unknown blurring degrees. We evaluate the sensitivity of DEBID to the kernel size parameter (*i.e.* the output sizes of the kernel generators) on Levin *et al.*'s dataset, by varying that parameter to be $35, 41, 45, 51, 55, 61$ and the true kernel size respectively. See Table VI for the quantitative results. We also include †SelfDeblur for comparison, which is tested under the same configuration as ours. It can be seen that †SelfDeblur is sensitive to the kernel size parameter. If the kernel size parameter is fixed to a constant for all images,

TABLE V
RUNNING TIME FOR PROCESSING AN IMAGE WITH DIFFERENT IMAGE SIZES AND KERNEL SIZES ON AVERAGE.

Time	DEBID			SelfDeblur / SelfDeblur-FFT		
(min)	51×51	75×75	101×101	51×51	75×75	101×101
384×384	9.3	10.4	10.8	7.5/6.2	9.2/6.9	11.8/7.3
512×512	13.8	15.2	16.4	11.7/9.3	14.5/10.2	18.2/10.9
768×768	29.2	30.1	32.7	26.3/19.5	31.9/20.2	40.3/21.7

its performance has a dramatic drop. In contrast, DEBID is much more robust to the kernel size parameter setting.

TABLE VI
PSNR(DB) RESULTS ON LEVIN *et al.*'S DATASET WITH DIFFERENT VALUES OF THE KERNEL SIZE PARAMETER.

Kernel size parameter	Truth	35	41	45	51	55	61
†SelfDeblur [6]	33.87	32.11	28.12	26.84	25.26	17.58	16.11
DEBID [Ours]	35.13	34.92	34.54	34.25	34.04	33.73	33.43

3) Study on the sensitivity of reference selection in image alignment: In previous experiments, we use the output of the middle generator $\mathcal{G}_{\boldsymbol{X}}^2$ (*i.e.* $p_0 = 2$) as the reference for the half-pixel image alignment in (18). We also try using other generators for the reference respectively. See Table VII for the results. There is minor difference observed, which implies that DEBID is insensitive to the reference selection.

TABLE VII
PSNR(DB)/SSIM RESULTS ON LEVIN *et al.*'S DATASET USING DIFFERENT ALIGNMENT REFERENCES.

$p_0 = 1$	$p_0 = 2$	$p_0 = 3$
34.91 / 0.960	34.92 / 0.960	34.90 / 0.960

E. Ablation Studies

1) Contributions of main components: The following ablation studies are conducted on Levin *et al.*'s dataset: *(i) "w/o HighPass"*: Remove the loss defined in the wavelet domain from (16); *(ii) "w/o KC"*: Disable all KC layers; *(iii) "w/o Ensemble"*: Pick the best from the outputs of all generators as the final deblurring result, rather than average them; (iv) "w/o Alignment": Disable half-pixel alignment (*i.e.* setting $\mathcal{A}$ to an identity mapping) in (18); and (v) "w/o Shuffle": remove the shuffle layers from the kernel generators. See Table VIII for the results. Obviously, each component of DEBID makes a noticeable contribution to the performance gain. Among them, the ensemble mechanism that leverages model diversity contributes the most and substantially, the alignment scheme done by the KC layer contributes the second most, and the loss on high-pass responses contributes the least yet noticeable. All these components are important to DEBID as they are designed for handling the solution ambiguity arsing from result shift, agnostic kernel size, and solution uncertainty.

TABLE VIII
RESULTS OF ABLATION STUDY ON LEVIN *et al.*'S DATASET.

Metric	w/o HighPass	w/o KC	w/o Ensemble	w/o Alignment	w/o Shuffle	Original
PSNR(dB)	34.62	34.31	33.93	34.58	34.57	34.92
SSIM	0.951	0.951	0.941	0.948	0.947	0.960

See Fig. 8 for a visual inspection of the benefit of the ensemble. We show the prediction residuals of the deblurred images from individual image generators and the one from their ensemble. The residuals produced by different image generators are quite diverse (*e.g.*, noticeable errors occur in different locations), and their aggregation leads to a noticeable residual reduction. We also provide a visual inspection of the benefit of the KC layer; see Fig. 9. We show the BID results from one image generator of DEBID with and without using the KC layer, respectively, with the initialization kept the same. We can observe that the kernel estimates become different between the two cases in the later iterations, and the one estimated with KC is more accurate. Also note that the rightmost support of the kernel estimate is allowed to go into the left side before applying KC.

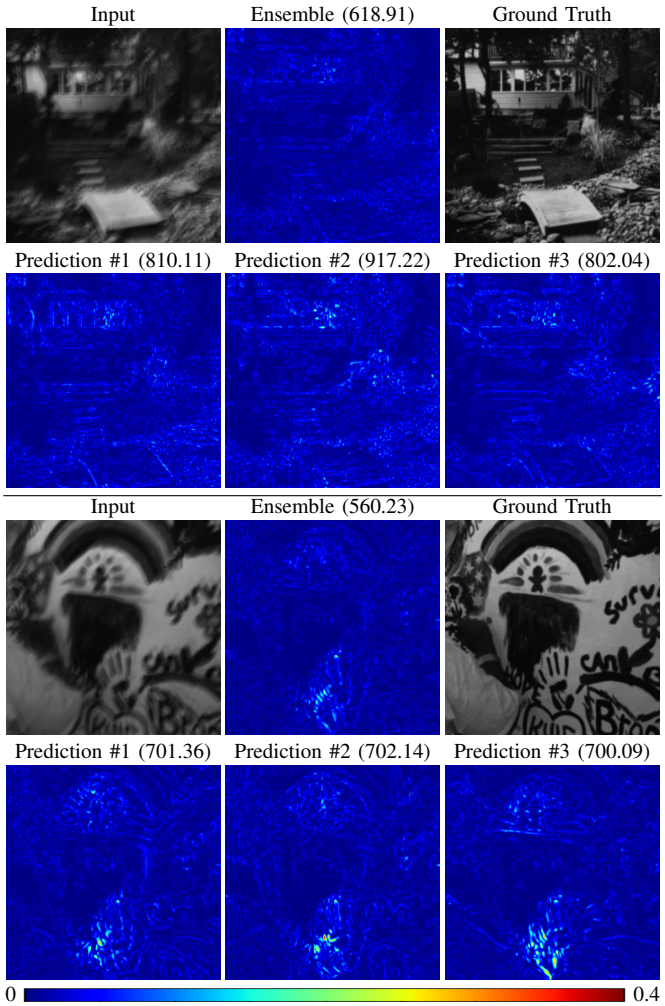


Fig. 8. Absolute residuals between estimated images and ground truths, generated by different image generators and their ensemble. Each value in the parentheses is the ℓ_1 norm of the residual, *i.e.*, the absolute error. See supplementary materials for the corresponding image estimates are shown.

2) Effectiveness of regularization parameters $\{\lambda_p\}_p$: We also evaluate the influence of the setting of $\{\lambda_p\}_p$ in the loss (16). Recall that $(\lambda_1, \lambda_2, \lambda_3)$ is set to $(\lambda - \Delta\lambda, \lambda, \lambda + \Delta\lambda)$ in DEBID where $\lambda = 0.1, \Delta\lambda = 0.05$. First, we zero out the variations in $\{\lambda_p\}_p$, *i.e.*, $\Delta\lambda = 0, \lambda = 0.1$. Then, we further set $\Delta\lambda = 0, \lambda = 0$, *i.e.*, the regularization term in the loss (16) is disabled. See Table IX for the quantitative results of these two settings on Levin *et al.*'s dataset. Comparing those results with the original ones, we can find that without the kernel regularization term, the performance drops about 1.04dB in

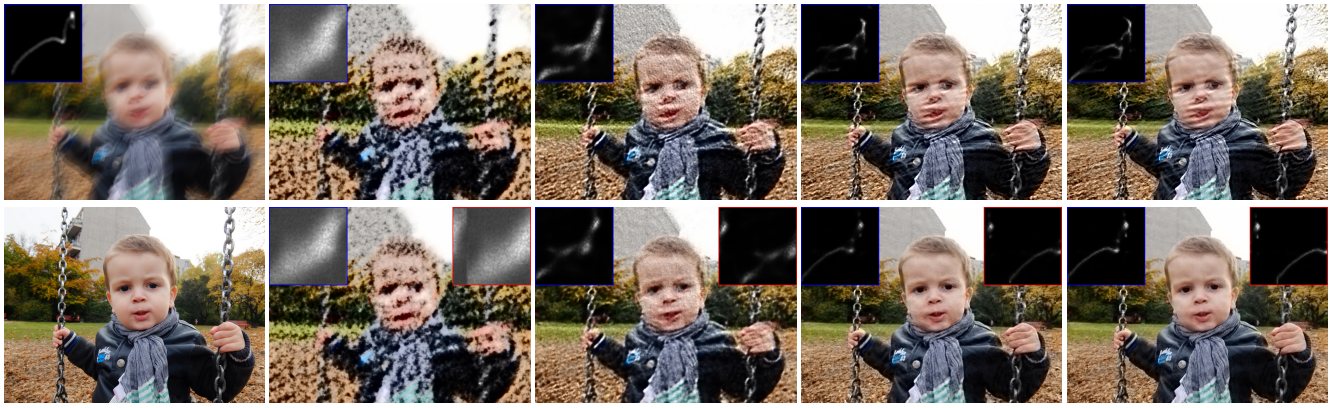


Fig. 9. Deblurring results w/o KC (upper row) and w/ KC (bottom row) on a single image generator of DEBID, using the same initialization. For the images generated with KC, the kernels after and before applying KC are shown on the left-top and right-top respectively.

PSNR. This is probably that the deblurred image from each generator tends to be under-deblurred as the kernel generator becomes more likely to output a delta kernel in this case, and for sharpness enhancement, the aggregation of under-deblurred estimations may not be as effective as the aggregation of images with artifacts. In addition, we can see that the small perturbations to $\{\lambda_p\}_p$ can improve the PSNR result by around 0.56dB, which is because the variations in $\{\lambda_p\}_p$ lead to higher diversity among the generators.

TABLE IX
PSNR(DB)/SSIM RESULTS ON LEVIN *et al.*'S DATASET WITH DIFFERENT SETTINGS OF REGULARIZATION WEIGHTS.

$\lambda = 0.1, \Delta\lambda = 0$	$\lambda = 0, \Delta\lambda = 0$	$\lambda = 0.1, \Delta\lambda = 0.05$ (original)
34.36 / 0.944	33.88 / 0.931	34.92 / 0.960

3) Effectiveness in handling saturated regions: The following ablation studies are conducted on the 'Saturated' category of Lai *et al.*'s dataset: *(i) 'Sigmoid'*: Use Sigmoid to replace the LReLU in the last output layer, following SelfDeblur [6]; *(ii) 'w/o Clip'*: Disable the thresholding operator $\mathcal{S}$ in the loss (16). See Table X for the results. It can be concluded that both the LReLU and the thresholding operation make noticeable contributions to the performance in dealing with images with saturated regions.

TABLE X
RESULTS OF ABLATION STUDY *w.r.t.* THE MECHANISMS FOR HANDLING SATURATED IMAGES ON THE 'SATURATED' CATEGORY OF LAI *et al.*'S DATASET, MEASURED IN TERMS OF PSNR(DB)/SSIM.

Sigmoid	w/o Clip	DEBID	†SelfDeblur
16.88 / 0.665	17.19 / 0.693	18.43 / 0.714	15.97 / 0.504

4) Effectiveness of NN architecture: We form three variants of †SelfDeblur to examine the effectiveness of our design on the architecture. The first version denoted by †SelfDeblur×3 is constructed by training and testing †SelfDeblur for three times separately, followed by averaging the results with sub-pixel aliment for output, as what DEBID does. The second version denoted by ‡SelfDeblur is constructed by replacing the NNs of image/kernel generators of SelfDeblur with the proposed lighter-weight ones. The last version denoted by ‡SelfDeblur×3 is constructed by separately running ‡SelfDeblur for three times, followed by averaging the results with sub-pixel alignment for output, as what DEBID does.

See Table XI for the comparison on Levin's dataset with the kernel size parameter fixed at 35, in terms of PSNR, SSIM, and number of model parameters. We also construct a baseline for comparison by setting the number P of image and kernel generator pairs to 1 in DEBID, which is denoted by DEBID(1). The results show that the proposed NN architecture improves both the accuracy of a single inference and the effectiveness of the ensemble, and the proposed multi-head base-sharing design not only reduces the model size for compactness but also keeps sufficient and even better model diversity which is important for ensemble-based prediction.

TABLE XI
COMPARISON OF DIFFERENT NN ARCHITECTURES ON LEVIN *et al.*'S DATASET IN TERMS OF THREE METRICS.

Metric	†SelfDeblur	‡SelfDeblur	DEBID(1)
#Parameters	3.76M	1.61M	1.61M
PSNR(dB) / SSIM	32.11 / 0.876	33.07 / 0.914	33.91 / 0.940
Metric	**†SelfDeblur×3**	**‡SelfDeblur×3**	**DEBID**
#Parameters	11.28M	4.83M	3.12M
PSNR(dB) / SSIM	32.59 / 0.891	33.72 / 0.925	34.92 / 0.960

5) Effect of ensemble size: We vary the ensemble size P from 1 to 7 and then evaluate the performance of DEBID. The parameters $\{\lambda_p\}_{p=1}^P$ are set to P uniform points starting at 0.05 and ending at 0.15, as we empirically observed noticeable performance decrease out of this range. See Table XII for the results, where using $P = 1, 2$ yields much worse results, and using $P > 3$ may achieve further PSNR gain but the gain becomes very minor when $P > 5$. Using $P = 3$ can achieve the balance between performance and computational cost.

F. Challenging Cases

In Fig. 10, we show a couple of challenging cases from the results of DEBID. The first case contains many severely-

TABLE XII
PSNR(DB) RESULTS OF DEBID WITH DIFFERENT ENSEMBLE SIZE P, TESTED ON LEVIN *et al.*'S DATASET.

$P=1$	$P=2$	$P=3$	$P=4$	$P=5$	$P=6$	$P=7$
33.91	34.54	34.92	35.06	35.12	35.14	35.15

blurred small image edges, which is well known as a challenging case for BID. As a result, DEBID produced noticeable artifacts. Even so, the artifacts in the result of DEBID are still noticeably less severe than that from SelfDeblur. It will be our future work on improving the estimation on such images. One possible solution is to introduce a smart edge selection mechanism during kernel estimation, similar to many traditional methods. In the second case, the kernel size parameter we set is much smaller than the ground-truth one, *i.e.*, 25 versus 51. As a result, DEBID cannot accurately predict the kernel even with the KC layer, and some text in small font surrounding "A7" in the deblurred image is not easy to recognize. Note that SeflDeblur also failed on this case with even worse results.

Input and GT kernel SelfDeblur DEBID

Fig. 10. Challenging cases for the proposed method.

V. CONCLUSION

In this paper, we showed that the aggregation of deep priors from multiple untrained NNs can better handle the solution ambiguity in BID than existing DIP-based methods. To have a computationally-efficient yet effective ensemble learning framework for deep prior aggregation, we introduced a multi-head shared-base NN architecture for latent image estimation and kernel prediction. In addition, the kernel centering layer was introduced for handling the shift ambiguity in the solutions, with additional benefits brought to kernel representation and estimation. The effectiveness of the proposed approach has been justified by extensive experiments. The ideas and techniques introduced in this paper can also see their applications in solving other non-linear inverse problems. In future, we will study how to improve the performance by introducing explicit mechanisms of effective edge selection and progressive kernel size estimation, and how to generalize the proposed approach to other problems such as non-uniform blind deblurring.

REFERENCES

[1] W. Zuo, D. Ren, D. Zhang, S. Gu, and L. Zhang, "Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1751–1764, 2016.

[2] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang, "Motion blur kernel estimation via deep learning," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 194–205, 2017.

[3] Y. Li, M. Tofighi, V. Monga, and Y. C. Eldar, "An algorithm unrolling approach to deep image deblurring," in *Proc. ICASSP*. IEEE, 2019, pp. 7675–7679.

[4] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, 2015.

[5] L. Li, J. Pan, W.-S. Lai, C. Gao, N. Sang, and M.-H. Yang, "Learning a discriminative prior for blind image deblurring," in *Proc. CVPR*, 2018, pp. 6616–6625.

[6] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *Proc. CVPR*, 2020, pp. 3341–3350.

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

[8] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. CVPR*, 2018, pp. 9446–9454.

[9] Y. Gandelsman, A. Shocher, and M. Irani, "Double-dip: Unsupervised image decomposition via coupled deep-image-priors," in *Proc. CVPR*, 2019, pp. 11 026–11 035.

[10] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. CVPR*. IEEE, 2011, pp. 233–240.

[11] D. Perrone and P. Favaro, "Total variation blind deconvolution: The devil is in the details," in *Proc. CVPR*, 2014, pp. 2909–2916.

[12] Y. Bai, G. Cheung, X. Liu, and W. Gao, "Graph-based blind image deblurring from a single photograph," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1404–1418, 2018.

[13] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *Proc. CVPR*. IEEE, 2009, pp. 104–111.

[14] ——, "Framelet-based blind motion deblurring from a single image," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 562–572, 2011.

[15] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proc. CVPR*, 2013, pp. 1107–1114.

[16] L. Chen, F. Fang, S. Lei, F. Li, and G. Zhang, "Enhanced sparse model for blind deblurring," in *Proc. ECCV*. Springer, 2020, pp. 631–646.

[17] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *Proc. ECCV*. Springer, 2014, pp. 783–798.

[18] W. Ren, X. Cao, J. Pan, X. Guo, W. Zuo, and M.-H. Yang, "Image deblurring via enhanced low-rank prior," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3426–3437, 2016.

[19] Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single-image blind deblurring using multi-scale latent structure prior," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2033–2045, 2019.

[20] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Blind image deblurring using dark channel prior," in *Proc. CVPR*, 2016, pp. 1628–1636.

[21] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao, "Image deblurring via extreme channels prior," in *Proc. CVPR*, 2017, pp. 4003–4011.

[22] F. Wen, R. Ying, Y. Liu, P. Liu, and T.-K. Truong, "A simple local minimal intensity prior and an improved algorithm for blind image deblurring," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 2923–2937, 2021.

[23] L. Chen, F. Fang, T. Wang, and G. Zhang, "Blind image deblurring with local maximum gradient prior," in *Proc. CVPR*, 2019, pp. 1742–1750.

[24] Z. Xu, H. Chen, and Z. Li, "Fast blind deconvolution using a deeper sparse patch-wise maximum gradient prior," *Signal Process. Image Commun.*, vol. 90, p. 116050, 2021.

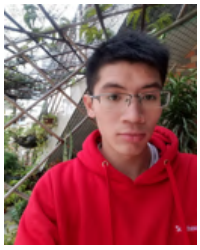
[25] B. Luo, Z. Cheng, L. Xu, G. Zhang, and H. Li, "Blind image deblurring via superpixel segmentation prior," *IEEE Trans. Circuits Syst. Video Technol.*, 2021.

[26] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. ECCV*. Springer, 2010, pp. 157–170.

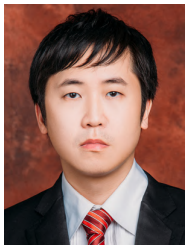
[27] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Proc. ICCP*. IEEE, 2013, pp. 1–8.

[28] J. Pan, R. Liu, Z. Su, and X. Gu, "Kernel estimation from salient structure for robust motion deblurring," *Signal Process.-Image Commun.*, vol. 28, no. 9, pp. 1156–1170, 2013.

[29] D. Gong, M. Tan, Y. Zhang, A. Van den Hengel, and Q. Shi, "Blind image deconvolution by automatic gradient activation," in *Proc. CVPR*, 2016, pp. 1827–1836.
[30] A. C. Likas and N. P. Galatsanos, "A variational approach for bayesian blind image deconvolution," *IEEE Trans. Image Process.*, vol. 52, no. 8, pp. 2222–2233, 2004.
[31] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *ACM Trans. Graph.*, 2006, pp. 787–794.
[32] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. CVPR*. IEEE, 2009, pp. 1964–1971.
[33] ——, "Efficient marginal likelihood optimization in blind deconvolution," in *Proc. CVPR*. IEEE, 2011, pp. 2657–2664.
[34] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in *Proc. ECCV*. Springer, 2012, pp. 341–355.
[35] L. Yang and H. Ji, "A variational EM framework with adaptive edge selection for blind motion deblurring," in *Proc. CVPR*, 2019, pp. 10 167–10 176.
[36] S. Liu, H. Wang, J. Wang, and C. Pan, "Blur-kernel bound estimation from pyramid statistics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 1012–1016, 2015.
[37] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. ECCV*. Springer, 2016, pp. 221–235.
[38] R. Yan and L. Shao, "Blind image blur estimation via deep learning," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1910–1921, 2016.
[39] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 666–681, 2020.
[40] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," *Proc. NIPS*, vol. 27, pp. 1790–1798, 2014.
[41] R. Aljadaany, D. K. Pal, and M. Savvides, "Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution," in *Proc. CVPR*, 2019, pp. 10 235–10 244.
[42] Y. Song, J. Zhang, L. Gong, S. He, L. Bao, J. Pan, Q. Yang, and M.-H. Yang, "Joint face hallucination and deblurring via structure generation and detail enhancement," *Int. J. Comput. Vision*, vol. 127, no. 6, pp. 785–800, 2019.
[43] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. CVPR*, 2019, pp. 8878–8887.
[44] A. Kaufman and R. Fattal, "Deblurring using analysis-synthesis networks pair," in *Proc. CVPR*, 2020, pp. 5811–5820.
[45] R. Yasarla, F. Perazzi, and V. M. Patel, "Deblurring face images using uncertainty guided multi-stream semantic networks," *IEEE Trans. Image Process.*, vol. 29, pp. 6251–6263, 2020.
[46] J. Cai, W. Zuo, and L. Zhang, "Dark and bright channel prior embedded network for dynamic scene deblurring," *IEEE Trans. Image Process.*, vol. 29, pp. 6885–6897, 2020.
[47] J. Pan, J. Dong, Y. Liu, J. Zhang, J. Ren, J. Tang, Y.-W. Tai, and M.-H. Yang, "Physics-based generative adversarial models for image restoration and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2449–2462, 2020.
[48] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. CVPR*, 2021, pp. 14 821–14 831.
[49] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proc. ICCV*, 2021, pp. 4641–4650.
[50] Z. Xia and A. Chakrabarti, "Training image estimators without image ground-truth," in *Proc. NeurIPS*, 2019.
[51] B. Lu, J.-C. Chen, and R. Chellappa, "Unsupervised domain-specific deblurring via disentangled representations," in *Proc. CVPR*, 2019, pp. 10 225–10 234.
[52] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," https://github.com/csdwren/SelfDeblur, 2020, [Online; accessed 2020].
[53] J. Kotera, F. Šroubek, and V. Šmídl, "Improving neural blind deconvolution," in *Proc. ICIP*. IEEE, 2021, pp. 1954–1958.
[54] A. Vedaldi, V. Lempitsky, and D. Ulyanov, "Deep image prior," *Int. J. Comput. Vision*, 2020.
[55] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
[56] I. Daubechies, B. Han, A. Ron, and Z. Shen, "Framelets: Mra-based constructions of wavelet frames," *Appl. Comput. Harmon. Anal.*, vol. 14, no. 1, pp. 1–46, 2003.
[57] S. Cho and S. Lee, "Fast motion deblurring," in *ACM Trans. Graph.*, 2009, pp. 1–8.
[58] L. Pan, R. Hartley, M. Liu, and Y. Dai, "Phase-only image based kernel estimation for single image blind deblurring," in *Proc. CVPR*, 2019, pp. 6034–6043.
[59] J. Pan, J. Dong, Y.-W. Tai, Z. Su, and M.-H. Yang, "Learning discriminative data fitting functions for blind image deblurring," in *Proc. ICCV*, 2017, pp. 1068–1076.
[60] Z. Hu and M.-H. Yang, "Good regions to deblur," in *Proc. ECCV*. Springer, 2012, pp. 59–72.
[61] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," *Proc. NIPS*, vol. 22, pp. 1033–1041, 2009.
[62] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang, "A comparative study for single image blind deblurring," in *Proc. CVPR*, 2016, pp. 1701–1709.
[63] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, 1995.



Minqin Chen is currently an M.Sc candidate in Computer Science at South China University of Technology. His research interests include image recovery and self-supervised learning.



Yuhui Quan received the Ph.D. degree in Computer Science from South China University of Technology in 2013. He worked as a postdoctoral research fellow in Mathematics at National University of Singapore from 2013 to 2016. He is currently an associate professor in Computer Science at South China University of Technology. His research interests include image restoration, unsupervised learning, and sparse representation.

Yong Xu received the B.S., M.S., and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He was a Post-Doctoral Research Fellow of computer science with South China University of Technology from 1999 to 2001. He is currently a professor in Computer Science at South China University of Technology. His current research interests include computer vision and image processing.

Hui Ji received the B.Sc. degree in Mathematics from Nanjing University in China, the M.Sc. degree in Mathematics from National University of Singapore and the Ph.D. degree in Computer Science from the University of Maryland, College Park. In 2006, he joined National University of Singapore as an assistant professor in Mathematics. Currently, he is an associate professor in mathematics at National University of Singapore. His research interests include image processing and machine learning.