Neumann Network with Recursive Kernels for Single Image Defocus Deblurring

Yuhui Quan[1,2] Zicong Wu[1,2] Hui Ji[3]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2] Pazhou Lab, Guangzhou 510335, China
[3] Department of Mathematics, National University of Singapore, 119076, Singapore
csyhquan@scut.edu.cn, cszicongwu@mail.scut.edu.cn, matjh@nus.edu.sg *

Abstract

Single image defocus deblurring (SIDD) refers to recovering an all-in-focus image from a defocused blurry one. It is a challenging recovery task due to the spatially-varying defocus blurring effects with significant size variation. Motivated by the strong correlation among defocus kernels of different sizes and the blob-type structure of defocus kernels, we propose a learnable recursive kernel representation (RKR) for defocus kernels that expresses a defocus kernel by a linear combination of recursive, separable and positive atom kernels, leading to a compact yet effective and physics-encoded parametrization of the spatially-varying defocus blurring process. Afterwards, a physics-driven and efficient deep model with a cross-scale fusion structure is presented for SIDD, with inspirations from the truncated Neumann series for approximating the matrix inversion of the RKR-based blurring operator. In addition, a reblurring loss is proposed to regularize the RKR learning. Extensive experiments show that, our proposed approach significantly outperforms existing ones, with a model size comparable to that of the top methods.

1. Introduction

Defocus blurring is a type of degradation that can occur in optical systems when capturing an image with varying scene depths. In an optical system, in order to project an image of objects onto film, a lens is used to bend the incoming light inwards, causing the light rays to trace out the shape of a cone and converge at the cone apex. When the distance between an object and the lens makes the cone apex just touch the film, the light rays will produce tiny illuminated circles and the object will appear in focus. In other words, the captured image is only clear for objects that are at a certain distance from the lens, known as the focal plane, which is determined by the distance between the lens and the film. For the objects away from the focal plane, the light rays will spread out, resulting in big illuminated circles (called circle(s) of confusion, CoC) that overlap with others. As a result, these objects appear blurry in the image.

As defocus blurring can cause the loss of image details that are crucial for subsequent vision tasks, many applications can benefit from the recovery of all image details from a defocused image, such as photo refocusing, semantic segmentation, text recognition, and object detection; see *e.g.* [25]. Single image defocus deblurring (SIDD) is a technique for this purpose, *i.e.*, recovering an all-in-focus (AIF) image with sharp and clear details from a defocused image.

In a defocused image, each pixel is associated with a defocus kernel $\boldsymbol{K}_{h,w}$ related to CoC. As the CoC of a pixel is determined by the distance of its scene point to the focal plane, pixels with different scene depths will have different defocus kernels, *i.e.*, the defocus blurring is spatially-varying. The relation between a defocused image $\boldsymbol{Y}$ and its AIF counterpart $\boldsymbol{X}$ can be expressed as $\boldsymbol{Y} = \mathcal{D} \circ \boldsymbol{X}$, *i.e.* applying a blurring operator $\mathcal{D}$ to $\boldsymbol{X}$ as follows:

$$\boldsymbol{Y}[h,w] = (\mathcal{D} \circ \boldsymbol{X})[h,w] := \sum_{r,c} \boldsymbol{K}_{h,w}[r,c]\boldsymbol{X}[h-r, w-c].$$

SIDD then needs to estimate $\boldsymbol{X}$ and $\{\boldsymbol{K}_{h,w}\}_{h,w}$, which is a challenging non-uniform blind deblurring problem.

Most existing studies on SIDD are a by-product of defocus map estimation [4,7,9,16,17,20,26,29,37,42,44,50,51]. That is, a defocused image is deblurred by calling some non-blind deblurring method using the defocus kernels derived from an estimated defocus map; see *e.g.* [10, 18, 19, 22, 26, 33, 43]. Such an approach often suffers from the sensitivity of non-blind deblurring to kernel estimation errors and is usually computationally expensive. In recent years, it has emerged as a promising approach to train an end-to-end deep neural network (DNN) for SIDD using one or more datasets; see *e.g.* [1, 14, 21, 25, 27, 32, 35, 49]. However, existing work still has a large room for performance improvement. The aim of this paper is to develop an end-to-end

*This work was supported by Natural Science Foundation of Guangdong Province (Grant No. 2022A1515011755 and 2023A1515012841), and Singapore MOE AcRF Tier 1 Grant (WBS No. A-8000981-00-00).

deep learning-based approach for SIDD that brings noticeable performance improvement over existing methods.

1.1. Main Idea

Given a defocused image, estimating the defocus kernel $\boldsymbol{K}_{h,w}$ for each pixel suffers from severe solution ambiguity. As their sizes have significant variations, a plain matrix expression of these kernels with the maximum size will lead to an overwhelming number of unknowns and overfitting. It is critical to have a compact representation of defocus kernels with a much reduced number of unknowns, yet sufficient to express a wide range of defocus kernels. Our solution is the so-called recursive kernel representation (RKR) model.

The RKR model expresses a defocus kernel via a linear combination over a learnable dictionary composed by a set of atom kernels. The atom kernels are defined using the tensor product of 1D positive kernels in a recursive scheme. Such a recursive and separable structure leads to a compact parametrization of defocus kernels whose sizes may vary over a wide range, as well as a fast computational scheme for the spatially-varying blurring operator, involving only a few convolutions and point-wise multiplications. In addition, RKR also encodes implicit physical priors of defocus kernels, such as strong correlation among defocus kernels of different sizes in the same image, and blob-type kernel supports. Further, RKR provides a learnable and more expressive representation for real-world defocus kernels, compared to predefined Gaussian-based models [32, 37].

Different from uniform blurring modeled by a convolution, whose inversion can be efficiently calculated via fast Fourier transform (FFT), the inversion of a defocus blurring operator has no known transform for fast calculation. Inspired by the approximation of truncated Neumann series (NS) to matrix inversion, we utilize truncated NS expansion and the RKR model to express such an inversion, leading to an efficient formulation that only involves standard convolutions and point-wise multiplications. Then, we develop a DNN called NRKNet (Neumann Recursive Kernel Network) for SIDD, with a cross-scale fusion structure inspired from the truncated NS expansion expressed in a coarse-to-fine fashion. The NRKNet mainly contains a learnable atom kernel set for RKR and a learnable module for predicting coefficient matrices at different scales for cross-scale fusion-based deblurring, both of which are efficient. In addition, we introduce a reblurring loss for further regularizing the learning of RKR-based defocus kernel prediction, which measures the reconstruction error w.r.t. input image using learned atoms kernels and predicted coefficients.

1.2. Contributions

Modeling defocus kernels and designing a deblurring process are two key parts for developing an effective SIDD approach. This paper provides new solutions to both. Compared to existing Gaussian-based models of defocus kernels, our RKR model is applicable to more-general non-Gaussian and non-isotropic defocus kernels. Moreover, our proposed cross-scale fusion structure inspired from the truncated NS approximation leads to a computationally-efficient physics-driven DNN for SIDD. See below for a summary of our technical contributions:

- An RKR model for compact parametric representation of defocus kernels, with physical priors encoded;
- An efficient DNN for SIDD with a cross-scale fusion structure inspired from truncated NS approximation;
- A reblurring loss for regularizing the learning of defocus kernel prediction.

The results in extensive experiments show that our proposed approach brings noticeable performance gain over existing ones, with a relatively-small model size.

2. Related Work

2.1. Two-Stage Methods for SIDD

SIDD can be done via a two-stage approach: estimate a defocus map from input and then apply some non-blind deblurring method with the defocus kernels estimated from a defocus map; see *e.g.* [4, 18, 22, 26, 33, 43]. One closely-related work is the generalized Gaussian model proposed by Liu *et al.* [26] for representing defocus kernels. This model needs to estimate two parameters embedded in a non-linear function, and its resulting blurring operator cannot be efficiently computed. In comparison, our RKR model allows calculating the blurring operator via standard convolutions and point-wise multiplications, which is very efficient.

Indeed, defocus map estimation itself is a challenging task, with abundant literature on it, *e.g.*, non-learning methods [7, 17, 37, 42] and deep learning-based methods [4, 5, 9, 16, 20, 29, 44, 50, 51]. A defective defocus map will yield blur kernels with large errors. Robust deblurring with erroneous kernels is another challenging task [6, 15, 28, 31, 40]. Gilton *et al.* [11] unrolled the NS expansion to design a DNN for non-blind robust deblurring but with restriction on uniform blur. In contrast, we combine the NS expansion with both a multi-scale scheme and the RKR model to design an efficient DNN to tackle non-uniform defocus blur.

2.2. Deep Defocus Deblurring Using Dual Pixels

In existing studies on end-to-end learning for defocus deblurring, some consider using two view images captured by a dual-pixel (DP) sensor as the DNN's input. One seminal work is done by Abuolaim and Brown [2]. They proposed to train a U-Net for predicting an AIF image from DP images and contributed a dataset of quadruples: a defocused image, its AIF counterpart, and a DP image pair. Abuolaim *et al.* [3] proposed an effective method to generate realistic DP

data synthetically to address the data capture bottleneck of a DP sensor, together with a recurrent CNN to improve deblurring results. Zhang and Wang [45] combined a CNN and a transformer for further improvement. When applying a DNN designed for DP images to a single defocused image, the performance may decrease significantly; see *e.g.* [2].

2.3. End-to-End DNNs for SIDD

Using a single defocused image as input, a few works trained a DNN with an auxiliary task defined on defocus maps [25, 27] or DP data [1, 21]. Ruan *et al*. [25] proposed a two-stage DNN where the first stage for defocus map prediction is supervised by the defocus maps provided in [21], and the second stage for deblurring is supervised by the paired data they constructed from light-field images. Ma *et al*. [27] constructed a dataset with both defocused/AIF pairs and the defocus maps, for better training a two-stage DNN. Abuolaim *et al*. [1] trained a single-encoder multi-decoder DNN to predict the AIF image and its associated DP views respectively. Lee *et al*. [21] proposed an auxiliary task that predicts the defocus disparity between DP views. In comparison to these methods, ours does not require additional data but only the pairs of defocused/AIF images for training, with wider applicability.

Regarding DNN architectures, some existing works focus on improving the spatially-variant or multi-scale processing ability of DNNs to handle spatially-varying blur. Lee *et al*. [21] proposed a DNN with iterative adaptive convolutional layers to generate pixel-wise separable filters for deblurring. Ruan *et al*. [35] proposed a DNN composed of dynamic filtering layers in a multi-scale cascade manner, together with a training strategy to utilize both synthetic and real-world data. Zhang and Zhai [47] proposed a generative adversarial DNN with an attention disentanglement mechanism to distinguish blurry and clear regions. Compared to the DNNs in [21, 35], ours can be viewed as combining separable kernels via the RKR model for deblurring, or as a specific kind of dynamic filtering with physical kernel priors. Unlike ours, the three methods above do not explicitly encode the physics of defocus blurring into their DNNs.

By assuming that defocus kernels have nearly the same shapes, Son *et al*. [14] proposed a DNN with kernel-sharing parallel atrous convolutions and channel attention to model inverse filtering. In comparison, our approach models defocus blurring via RKR that allows more variations in the shapes of defocus kernels. Quan *et al*. [32] modeled defocus kernels by Gaussian scale mixture and proposed a DNN by unrolling the fixed-point iteration. While their model allows defocus kernels to have different shapes, the base Gaussian kernels are fixed and isotropic whose mixture might not fit real-world defocus kernels well. In contrast, our RKR models defocus kernels by the mixture of a set of atom kernels learned from training data, leading to higher expressivity.

3. Proposed Approach

3.1. RKR-Based Modeling for Defocus Blurring

RKR model for defocus kernels Suppose a DNN is trained to predict spatially-varying defocus kernels $\boldsymbol{K}_{h,w}$ for each location (h, w). As the kernel sizes vary in a wide range, a compact yet expressive parametric model of $\boldsymbol{K}_{h,w}$ is necessary. Our RKR model expresses $\boldsymbol{K}_{h,w}$ by a linear expansion over a set of atom kernels $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_J\}$:

$$\boldsymbol{K}_{h,w} = \boldsymbol{\Gamma}_1[h, w]\boldsymbol{A}_1 + \cdots + \boldsymbol{\Gamma}_J[h, w]\boldsymbol{A}_J, \qquad (1)$$

where $\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_J$ denote the expansion coefficient matrices. To reduce the number of learned parameters, RKR defines the atom kernels by the tensor product of 1D kernels. Then, the separable atom kernels are defined via a recursive scheme to generate kernels of wide-range support sizes in an economic manner. This leads to a recursive and separable structure for the atom kernels:

$$\boldsymbol{A}_1 = \boldsymbol{\Delta},\ \boldsymbol{A}_j = \boldsymbol{A}_{j-1} \otimes (\boldsymbol{a}_j \otimes \boldsymbol{a}_j^\top),\ \ j > 1, \qquad (2)$$

where $\boldsymbol{a}_j \in \mathbb{R}^3_+$ is an 1D positive kernels with three taps, $\boldsymbol{\Delta}$ denotes the Dirac delta, and $\otimes$ represents convolution.

Properties of RKR model The atom set $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_J\}$ has a multi-scale structure, where $\boldsymbol{A}_j$ (current scale) is the composition of $\boldsymbol{A}_{j-1}$ (previous scale) and a 3×3 atom $\boldsymbol{a}_j \otimes \boldsymbol{a}_j^\top$, with an increasing support size of $(2j-1) \times (2j-1)$ over scale j. One can rewrite $\boldsymbol{A}_j$ by a cascade form:

$$\boldsymbol{A}_j = (\boldsymbol{a}_1 \otimes \boldsymbol{a}_1^\top) \otimes \cdots \otimes (\boldsymbol{a}_j \otimes \boldsymbol{a}_j^\top). \qquad (3)$$

By direct calculation, the total number of the parameters to learn for all atom kernels is only $3J$, while the largest kernel support size is $(2J-1) \times (2J-1)$, implying a noticeable reduction of parameter number via RKR.

The RKR model also encodes physical priors of defocus kernels. Empirically, defocus kernels of the same image vary mainly in their sizes [14], while their shapes are quite similar, exhibiting blob-type structures. By imposing positivity on atom kernels, the recursive definition in RKR includes such priors in a weak sense implicitly. Indeed, the RKR can be viewed as a diffusion process and the kernels $\boldsymbol{A}_j$ will recursively isotropically expand their supports over j, leading to blob-type kernel supports.

Recall that any 2D Gaussian kernel can be factorized using two 1D Gaussian kernels or several 2D Gaussian kernels with smaller variances. Thus, RKR can be viewed as a generalization of the classic Gaussian model [17, 20, 29, 37, 42] and the Gaussian mixture model [32], which can represent a larger class of defocus kernels (*e.g.* non-isotropic ones). When defocus kernels of the data show strong symmetry, as assumed in the Gaussian or Gaussian mixture-based representations, we can use symmetric atom kernels to impose symmetry onto the represented kernels.

To conclude, the RKR model can both exploit the physical priors of defocus kernels and compactly represent defocus kernels of wide-range sizes.

RKR-based blurring operator Using the RKR model, we express the blurring operator as

$$\mathcal{D}(\cdot) = \sum_{j=1}^{J} \mathbf{\Gamma}_j \odot (\cdot \otimes \boldsymbol{A}_j), \tag{4}$$

where $\odot$ represents point-wise multiplication. It can be seen that the RKR model brings acceleration to the calculation of the blurring operator, as the spatially-variant convolution now reduces to J spatially-invariant convolutions by $\{\boldsymbol{A}_j\}_{j=1}^{J}$ and the weighted summation of the convolution results. In addition, further acceleration can be implemented by noting that

$$\boldsymbol{Y} \otimes \boldsymbol{A}_j = \big((\boldsymbol{Y} \otimes \boldsymbol{A}_{j-1}) \otimes \boldsymbol{a}_j\big) \otimes \boldsymbol{a}_j^{\top}, \; j > 1. \tag{5}$$

We can sequentially calculate $\boldsymbol{Y} \otimes \boldsymbol{A}_j$ for $j = 1, \cdots, J$. Then, with $\boldsymbol{Y} \otimes \boldsymbol{A}_{j-1}$ pre-computed in the previous step, the 2D convolution $\boldsymbol{Y} \otimes \boldsymbol{A}_j$ reduces to two 1D convolutions of a small size, which is faster. Consider an N-pixel image and a kernel size of $S \times S$, the computational complexity of spatially convolution vs. RKR-based blurring operation is $\mathcal{O}(NS^2)$ vs. $\mathcal{O}(6NJ)$, as RKR involves $2J$- 1D convolutions with kernel length of 3. In practical SIDD, S^2 is much larger than $6J$, *e.g.*, 39^2 vs. 6×72 in our practice.

3.2. NS-Based Efficient Approximate Inversion

The RKR-formulated defocus blurring operator allows a DNN to exploit the physics of image formation, which is typically done by unfolding an iterative numerical scheme for solving a regularization model. In a typical unfolding DNN for image recovery, there are two steps in each stage and one is the inversion of the forward process $\mathcal{D}$. In uniform image deblurring (*e.g.* [23,24,46]), it can be efficiently calculated using FFT-based deconvolution. However, it is not the case for non-uniform defocus blurring. Considering computational efficiency, we need a physics-aware DNN without involving the inversion of $\mathcal{D}$.

Our solution is inspired by the truncated NS approximation to matrix inversion. Let $\mathcal{P} : \mathbb{R}^N \to \mathbb{R}^N$ denote a linear operator with spectral norm $\rho(\mathcal{P}) < 1$ and $\mathcal{I} : \mathbb{R}^N \to \mathbb{R}^N$ an identity operator. Then, one can express the inversion of $\mathcal{I} - \mathcal{P}$ by its NS expansion [13] as follows:

$$(\mathcal{I} - \mathcal{P})^{-1} = \sum_{k=0}^{\infty} \mathcal{P}^k = \mathcal{I} + \mathcal{P} + \mathcal{P}^2 + \mathcal{P}^3 + \cdots. \tag{6}$$

For a blurring operator $\mathcal{P}$ with $0 < \rho(\mathcal{P}) \leq 1$, we have $\rho(I - \mathcal{P}) < 1$. Substituting $\mathcal{P}$ by $\mathcal{I} - \mathcal{D}$ in (6) gives

$$\mathcal{D}^{-1} = \mathcal{I} + (\mathcal{I} - \mathcal{D}) + (\mathcal{I} - \mathcal{D})^2 + (\mathcal{I} - \mathcal{D})^3 + \cdots, \tag{7}$$

A truncated NS, *e.g.*, the 4-term expansion that reads

$$\tilde{\mathcal{D}} : \boldsymbol{Y} \to \boldsymbol{Y} + \sum_{k=1}^{3} (\mathcal{I} - \mathcal{D})^k \circ \boldsymbol{Y}, \tag{8}$$

will then approximate $\mathcal{D}^{-1}$. Based on (8), we have an efficient computational scheme which only involves convolutions and point-wise multiplications. Note that when $\rho(\mathcal{D})$ does not satisfy the condition for NS, the truncated NS still can be viewed as an approximation to the inversion with some implicit regularization. Together with the implicit regularization from a CNN structure and supervised training, it is still capable of learning an effective deblurring process.

Multi-scaling processing is an effective strategy for blind image deblurring to achieve efficiency and stability. Let $\boldsymbol{Y}_t$ denote the downsampled version of $\boldsymbol{Y}$ with the factor of 2^{t-1} (*i.e.* $\boldsymbol{Y}_1 = \boldsymbol{Y}$). A plain scheme for multi-scaling processing is to separately calculate (8) in different scales:

$$\boldsymbol{Z}_t = \boldsymbol{Y}_t + \sum_{k=1}^{3} (\mathcal{I} - \mathcal{D}_t)^k \circ \boldsymbol{Y}_t, \quad t = 1, 2, \cdots. \tag{9}$$

where $\mathcal{D}_t$ denotes the blurring operator at the tth scale. Then all $\boldsymbol{Z}_t$ are integrated for estimating $\boldsymbol{X}$. To improve the computational efficiency of inference, we propose the following cross-scale fusion pipeline:

$$\boldsymbol{Z}_0 = \boldsymbol{Y}_1 + (\mathcal{I} - \mathcal{D}_1) \circ \boldsymbol{Y}_1 + \big((\mathcal{I} - \mathcal{D}_2)^2 \circ \boldsymbol{Y}_2\big)_{\uparrow 2} + \big((\mathcal{I} - \mathcal{D}_3)^3 \circ \boldsymbol{Y}_3\big)_{\uparrow 4}, \tag{10}$$

where the notation $_{\uparrow c}$ denotes the upsampling operator with the factor of c. In (10), we apply higher-order terms of $(\mathcal{I} - \mathcal{D})$ to images with smaller sizes for acceleration.

3.3. DNN Architecture

Our NRKNet is built upon the cross-scale fusion pipeline of (10) and the RKR-formulated blurring operator of (4). We extend $\mathbf{\Gamma}_j$ in (4) to the scale-dependent version $\mathbf{\Gamma}_{j,t}$ and define $\mathcal{D}_t$ in (10) by

$$\mathcal{D}_t(\cdot) := \sum_{j=1}^{J} \mathbf{\Gamma}_{j,t} \odot (\cdot \otimes \boldsymbol{A}_j), \tag{11}$$

where $\boldsymbol{A}_j$ parameterized by $\{\boldsymbol{a}_j\}_j$ via (2) is shared across different scales. All $\boldsymbol{a}_j$ are trainable parameters, and all $\mathbf{\Gamma}_{j,t}$ are estimated by a trainable module. The NRKNet infers an AIF image from an input image $\boldsymbol{Y}$ using (10). At scale t, it

1. estimates $\mathbf{\Gamma}_{j,t}$ from $\boldsymbol{Y}_t$ to construct $\mathcal{D}_t$; and
2. calculates $(\mathcal{I} - \mathcal{D}_t)^t \circ \boldsymbol{Y}_t$.

Afterwards, the results from all scales are upsampled to the original size and summed together with $\boldsymbol{Y}_1$ ($\boldsymbol{Y}$) to obtain the final output $\boldsymbol{Z}_0$. See Figure 1 for the outline of NRKNet.

Module for estimating $\Gamma_{j,t}$ Inspired by [32, 39], a U-Net [34] with weights shared across scales is first employed to extract features from $\boldsymbol{Y}_t$ for each scale t. Then, convolutional long short-term memory (ConvLSTM) [38] units across different scales are used for estimating $\mathbf{\Gamma}_{j,t}$ from the extracted features. As a result, NRKNet relates the learning at different scales not only by passing the result from one scale to the next, but also by the recurrence mechanism of the ConvLSTM units. The U-Net sequentially connects 3

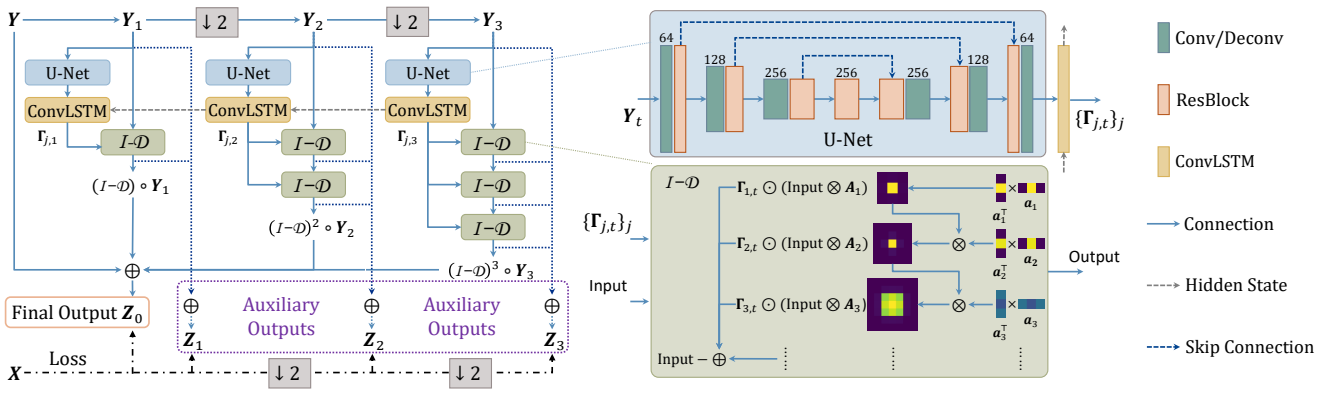

Figure 1. Outline of proposed NRKNet for SIDD.

encoder blocks, 1 residual block and 3 decoder blocks. Each encoder/decoder block contains a Conv layer with down/up-upsampling, and a residual block with 2 Conv layers and a residual connection. The output of the i-th encoder block is added to the input of the $(3-i)$-th decoder block via a skip connection. The ConvLSTM units capture dependencies of blurring among different scales, and their hidden states capture useful information from different scales and benefit deblurring across different scales. This can progressively improve the estimation of coefficient matrices.

Implementation of $(\mathcal{I}-\mathcal{D}_t)^t \circ \boldsymbol{Y}$ We sequentially apply $\mathcal{I}-\mathcal{D}_t$ for the calculation of $(\mathcal{I}-\mathcal{D}_t)^t$ and compute $(\mathcal{I}-\mathcal{D}_t)\circ\boldsymbol{Y} = \boldsymbol{Y}-\mathcal{D}_t\circ\boldsymbol{Y}$ via (4) and (5). A softmax layer is attached behind each $\boldsymbol{a}_j$ to enforce non-negativeness and ℓ_1-normalization on $\boldsymbol{a}_j$. The summation in (11) is implemented by 1×1 convolution.

3.4. Loss Functions

Let $\boldsymbol{Z}_0$ denote the DNN's output and $\boldsymbol{Y}$ the input. Let $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3$ denote the images at different scales: $\boldsymbol{Z}_t = \sum_{k=1}^{t}(\mathcal{I}-\mathcal{D})^k \circ \boldsymbol{Y}$, which are only for training and not calculated in inference. Let $\boldsymbol{X}$ be the ground truth and $\boldsymbol{X}_t$ the downsampled version of $\boldsymbol{X}$ with factor $2^{t-1}(t>0)$. For convenience, we define $\boldsymbol{Y}_0=\boldsymbol{Y}, \boldsymbol{X}_0=\boldsymbol{X}$. The training loss is defined in a multi-scale manner as follows:

$$\mathcal{L} := \sum_{t=0}^{3}\lambda_t \mathcal{L}_{\text{predict}}(\boldsymbol{Z}_t,\boldsymbol{X}_t) + \alpha\sum_{t=1}^{3}\lambda_t\mathcal{L}_{\text{reblur}}(\boldsymbol{Y}_t,\boldsymbol{X}_t), \tag{12}$$

where the weights $[\lambda_0,\cdots,\lambda_3]$ at different scales are set to a decreasing sequence: $[1, 0.75, 0.5, 0.25]$. Note that t starts from 0 and 1 in the two terms respectively. The function $\mathcal{L}_{\text{predict}}$ is the prediction loss defined as:

$$\mathcal{L}_{\text{predict}}(\boldsymbol{Z}_t,\boldsymbol{X}_t) := \|\boldsymbol{Z}_t-\boldsymbol{X}_t\|_2^2 + \beta\|\mathcal{F}(\boldsymbol{Z}_t)-\mathcal{F}(\boldsymbol{X}_t)\|_1, \tag{13}$$

where $\mathcal{F}$ denotes FFT. The first term of $\mathcal{L}_{\text{predict}}$ is the standard mean-squared error, and the second term is the frequency-domain reconstruction (FDR) loss [8] for better restoring corrupted high frequency information. The function $\mathcal{L}_{\text{reblur}}$ is our proposed reblurring loss. It is defined as

$$\mathcal{L}_{\text{reblur}} := \|\mathcal{D}_t(\boldsymbol{X}_t)-\boldsymbol{Y}_t\|_2^2. \tag{14}$$

The reblurring loss encourages the estimated blurring operator $\mathcal{D}_t$ at each scale to synthesize the original blurred image $\boldsymbol{Y}_t$ back using the ground truth $\boldsymbol{X}_t$. As a result, it can regularize the learning of both the atom set $\{\boldsymbol{A}_j\}_{j=1}^{J}$ of RKR and the prediction module of $\boldsymbol{\Gamma}_{j,t}$. This loss differs from the reblurring loss of [21] that needs to train another DNN for reblurring.

4. Experiments

4.1. SIDD Datasets and Implementation Details

Benchmark datasets Experiments for performance evaluation are conducted on five benchmark datasets, including DPDD [2], LFDOF [25], RealDOF [21], RTF [9] and CUHK-BD [36]. These datasets are captured by different devices with different resolutions. Since only DPDD and LFDOF provide train-test split, following the protocols of existing works on them, we train two models using the training sets from DPDD and LFDOF, respectively. The DPDD-trained model is evaluated on DPDD (test set), RealDOF, and RTF. The LFDOF-trained model is evaluated on LFDOF (test set) and RTF. CUHK-BD that has no ground truths is used to evaluate both the models qualitatively.

Implementation details The hyper-parameters of our approach are consistently set through all experiments. We fix $J=20$ for RKR and $\alpha=\beta=0.1$ for the training loss. All learnable DNN parameters are initialized by Xavier [12]. The Adam optimizer [30] is called for training, with epoch number 4000 and batch size 4. Similar to [35], the learning rate is fixed at 10^{-4} in the first 2000 epochs and decayed with a factor of 0.5 for every 1000 epochs. Random flipping, rotation, and cropping (to 256×256 pixels) are used for data augmentation. Our approach is implemented in PyTorch and run on an NVIDIA RTX 2080Ti GPU.

Method	DPDD			RealDOF			RTF			# Parameters (Million)	# MACCs (Billion)	Time (Second)
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS			
Input	23.890	0.725	0.349	22.333	0.633	0.524	24.200	0.717	0.248	-	-	-
DPDNet-S	24.348	0.747	0.277	22.870	0.670	0.425	23.608	0.591	0.296	32.3	485	0.3
AIFNet	24.213	0.742	0.309	23.093	0.680	0.413	24.041	0.758	0.289	41.6	985	0.5
MDP	25.347	0.763	0.268	23.500	0.681	0.444	24.012	0.738	0.312	46.9	1081	0.5
KPAC	25.221	0.774	0.226	23.975	0.762	0.338	24.618	0.777	0.236	2.1	197	**0.1**
IFANet	25.366	0.789	0.217	24.712	0.748	**0.306**	24.924	0.801	0.227	10.5	363	0.3
GKMNet	25.468	0.789	0.219	24.257	0.729	0.390	24.972	0.791	0.262	**1.4**	**148**	0.2
DRBNet	25.485	0.792	0.254	24.884	0.751	0.376	24.463	0.773	0.311	11.7	347	0.2
NRKNet	**26.109**	**0.810**	**0.210**	**25.148**	**0.768**	0.338	**25.931**	**0.829**	**0.215**	6.1	553	0.3

Table 1. Quantitative comparison of DPDD-trained models on three datasets. Best (second best) results are boldfaced (underlined).

4.2. Performance Comparison

Methods for comparison Seven latest deep learning-based SIDD methods are used for comparison, including DPDNet-S [2], AIFNet [25], KPAC [14], IFANet [21], GKMNet [32], MDP [1] and DRBNet [35]. All these methods, except AIFNet and DRBNet, provide the models pre-trained on DPDD. We directly use their models for the comparison with our DPDD-trained model, while retraining their models on LFDOF using their released codes for the comparison of LFDOF-trained models. AIFNet provides a model trained on LFDOF and the SYNDOF dataset [20] (used for its defocus map estimation stage). We use it for the comparison with LFDOF-trained models and retrain it with DPDD and SYNDOF for the comparison of DPDD-trained models. DRBNet provides a model trained on LFDOF and fine-tuned on DPDD. For fair comparison, we retrain its model on two datasets, respectively.

Quantitative comparison of DPDD-trained models Table 1 compares the results of the DPDD-trained models on three datasets, in terms of three widely-used metrics: PSNR (Peak Signal to Noise Ratio, dB), SSIM (Structural SIMilarity index) [41], and LPIPS (Learned Perceptual Image Patch Similarity) [48]. Pixel values of deblurred images are cropped to $[0, 255]$ before evaluation. Our NRKNet achieves the highest PSNR and SSIM values among all the compared methods across all the three datasets. In terms of LPIPS, NRKNet also performs the best on DPDD and RTF, and the second best on RealDOF. The PSNR gain of NRKNet over other DNNs is quite noticeable, *e.g.*, 0.624dB on DPDD, 0.264dB on RealDOF, and 0.959dB on RTF. Such improvement on the datasets captured by different devices indeed shows that the superior generalization performance of NRKNet. Table 1 also compares the computational complexity of different methods, in terms of number of model parameters, number of MACCs (Multiply-ACCumulate operations), and running time on a 1280×720 image. NRKNet has the third smallest model size, with a medium level of MACCs and running time.

Quantitative comparison of LFDOF-trained models See Table 2 for the quantitative comparison of the LFDOF-trained models on LFDOF and RTF. On both the datasets, NRKNet achieves the best results among all the compared methods in terms of all three metrics. The PSNR gain of NRKNet over the second best is 0.228dB on LFDOF and 0.495dB on RTF. These results have again demonstrated the effectiveness of our proposed approach.

Method	LFDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Input	25.874	0.777	0.320	24.200	0.717	0.248
AIFNet	29.677	0.884	0.202	27.552	0.882	0.176
MDP	28.069	0.834	0.185	25.580	0.809	0.228
KPAC	28.942	0.857	0.174	25.959	0.803	0.230
IFANet	29.787	0.872	0.156	26.437	0.838	0.238
GKMNet	29.081	0.867	0.171	26.989	0.855	0.247
DRBNet	30.253	0.883	**0.147**	26.717	0.853	0.200
NRKNet	**30.481**	**0.884**	**0.147**	**28.047**	**0.889**	**0.145**

Table 2. Quantitative comparison of LFDOF-trained models. Best (second best) results are boldfaced (underlined).

Qualitative comparison Figure 2 shows some deblurred images from the previous experiments for visual comparison. The deblurred images on CUHK-BD are separately shown in Figure 3. The visual quality of the deblurred images consists with the quantitative results above. Overall, in comparison to other methods, NRKNet achieves higher visual quality, *e.g.*, NRKNet recovers the image structures and details better, restores clearer text, and produces less artifacts. *See also supplementary materials for the visualization of the learned atom kernels and predicted coefficients*, where regions with larger blur amount tend to have larger coefficients on large-size kernels, and vice versa.

4.3. Ablation Studies ans Analysis

RKR learning We construct three baselines to analyze the proposed RKR learning. *1)* Non-adaptive: all the atoms $\boldsymbol{A}_j$ are fixed to Gaussian kernels with the same number and the same size, using the scheme suggested in [32]. *2)* Non-recursive: removing the recursive structure from RKR by defining independent atoms: $\boldsymbol{A}_j = \hat{\boldsymbol{a}}_j \otimes \hat{\boldsymbol{a}}_j, \bar{\boldsymbol{a}}_j \in \mathbb{R}_+^{2j-1}$. *3)* Non-separable: disabling the separability of atoms by directly defining $\boldsymbol{A}_j \in \mathbb{R}_+^{(2j-1)\times(2j-1)}$.

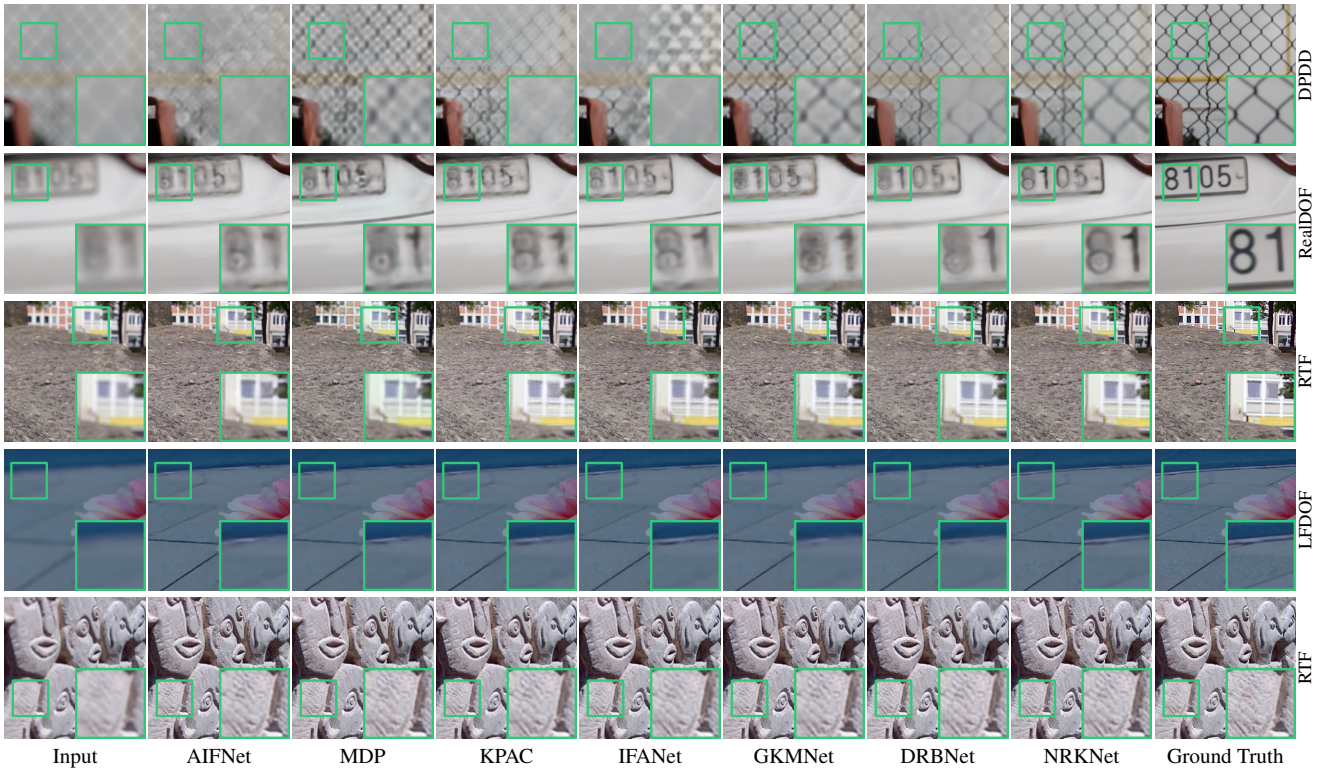

Figure 2. Deblurred images from DPDD-trained models (first three rows) and LFDOF-trained models (last two rows).

Model	PSNR	SSIM	LPIPS	# MACCs(B)	# Para(M)	Time(s)
NonAdaptive	25.309	0.797	0.255	76.81	9261	**0.3**
NonRecursive	25.994	0.807	0.218	2.03	440	0.4
NonSeparable	26.071	0.807	0.221	12.99	12340	0.5
Full RKR	**26.109**	**0.810**	**0.210**	**1.60**	**60**	**0.3**

Table 3. Results (DPDD) of ablation study on RKR.

Metric	FullExp	2Scales	4Scales	w/o LSTM	Original
PSNR	26.077	25.894	26.020	24.371	**26.109**
SSIM	0.809	0.805	0.806	0.774	**0.810**
LPIPS	0.230	0.240	0.235	0.267	**0.210**
Time(s)	0.4	**0.3**	0.4	**0.3**	**0.3**

Table 4. Results (DPDD) of ablation study on DNN pipeline.

See Table 3 for their results on DPDD. For highlighting the differences on computational efficiency, the number of MACCs and the number of parameters are calculated only on the RKR-related modules. We have the following observations. *1)* Using pre-defined Gaussian kernels as atom kernels leads to the largest and noticeable PSNR decrease, indicating that our learned RKR provides a better model than the fixed Gaussian-based model for representing real-world defocus kernels. *2)* Using independent non-recursive atoms yields worse performance and higher computational complexity. The reason for its worse performance is probably that the recursive form for compactness does not impact its expressibility noticeably while encoding some implicit prior. *3)* Using non-separable atoms increases the time cost noticeably and yields a little performance degradation, indicating that separability also brings some regularization. *4)* Benefiting from recursive modeling, the full RKR has the lowest computational complexity.

Cross-scale fusion pipeline To analyze the cross-scale fusion pipeline in NRKNet, we construct four baselines as follows. *1)* Full expansion: At each scale of NRKNet, all the four terms (three scales) of truncated NS expansion are calculated, *i.e.*, up to $(\mathcal{I} - \mathcal{D})^3$ for every scale. *2)* Two scales: Only using the first two scales of NRKNet, *i.e.*, calculation up to $(\mathcal{I}-\mathcal{D})^2$. *3)* Four scales: Following the same way in the original design, we extend the model by adding one more scale and introducing $(\mathcal{I} - \mathcal{D})^4$ and $\boldsymbol{Y}_4$ to the 4th scale. *4)* w/o LSTM: Removing all ConvLSTM units and increasing the number of layers of U-Net to have a similar depth and parameter number of the original model.

See Table 4 for the results. *1)* Full expansion brings no performance gain (even worse) but much more additional computational cost. Our cross-scale fusion scheme provides an efficient alternate. *2)* The two-scale version of NRKNet leads to noticeable performance drop, which is probably due to its weakened deblurring power w/o using $(\mathcal{I} - \mathcal{D})^3$. *3)* The four-scale version also leads to performance drop, but not big. This is not surprising as the 4th scale is too

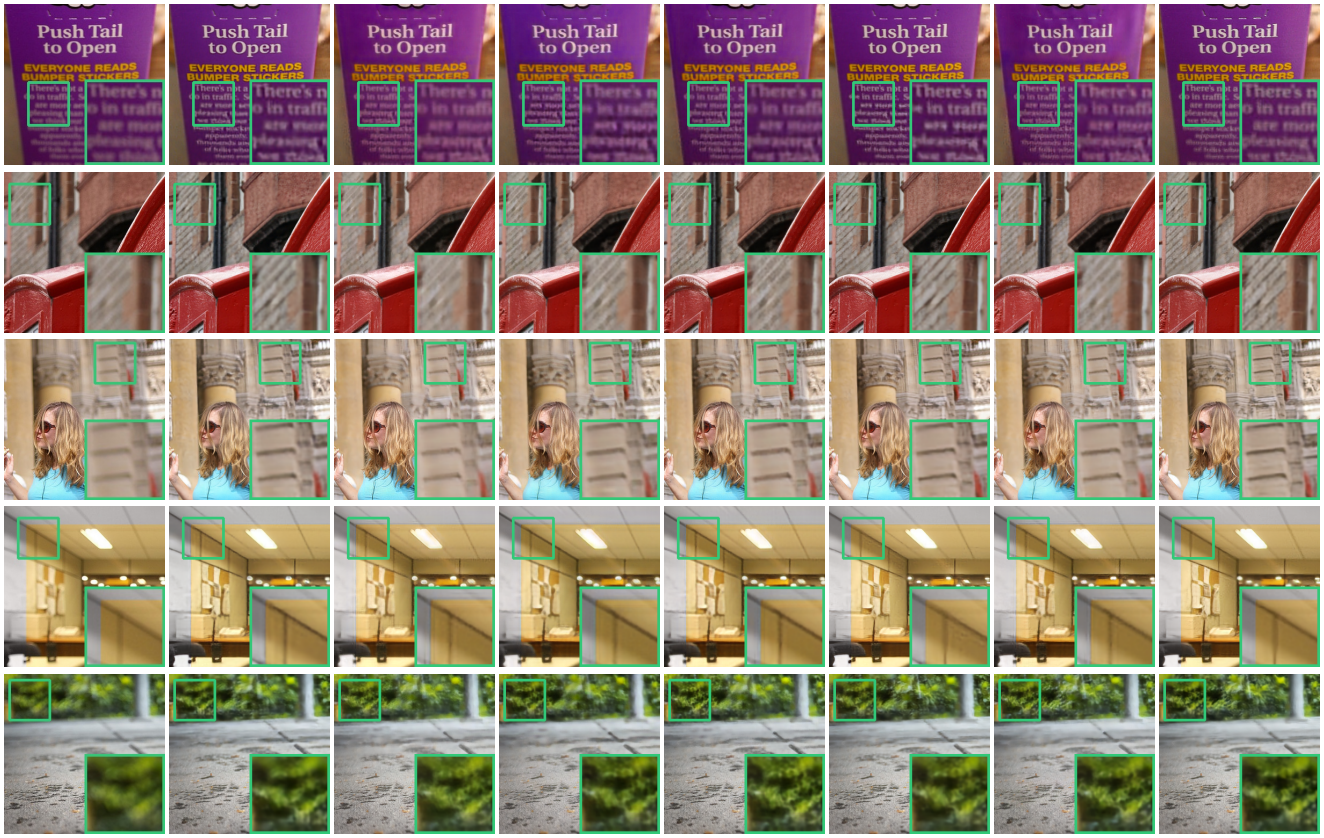

Figure 3. SIDD results from DPDD-trained models on selected images from CUHK-BD dataset. *See also our supplementary materials for the results of LFDOF-trained models.*

Metric	w/o $\mathcal{L}_{\text{reblur}}$	w/o FDR loss	Single-scale $\mathcal{L}$	Full Loss
PSNR	25.816	25.887	25.973	**26.109**
SSIM	0.804	0.805	0.806	**0.810**
LPIPS	0.240	0.235	0.234	**0.210**

Table 5. Results (DPDD) of ablation study on loss functions.

rough to provide useful information, but just increasing the running cost. *4)* The recurrent mechanism provided by the CovnLSTM units is very useful for coefficient prediction, which has noticeable contribution to the performance.

Loss functions We remove the reblurring loss and the FDR loss respectively from the total training loss to train the NRKNet, so as to verify their effectiveness. We also simplify the total training loss to a single-scale version. See Table 5 for the results. Our proposed reblurring loss and the FDR loss have similar amount of contribution to the deblurring performance in terms of all the three metrics. The multi-scale scheme in the total loss also has a notable contribution to the performance.

Effect of atom number To study the influence of the atom number J of the RKR model to the deblurring performance of NRKNet, we vary J to be several values respectively and retrain the NRKNet accordingly. The PSNR results on DPDD are: 25.942 ($J = 15$), 26.042 ($J = 18$), 26.109 ($J = 20$, original setting), 26.122 ($J = 22$), and 26.131 ($J = 25$). We can see that the PSNR decreases as J decreases, and it gains just a bit with $J > 20$.

5. Conclusion and Discussion

This paper proposed an end-to-end learning approach for SIDD with three innovative components: an efficient learnable RKR model for defocus kernels, an efficient cross-scale fusion DNN architecture inspired from NS approximation, and a reblurring loss for regularizing defocus kernel prediction learning. Extensive experiments on five datasets demonstrated the effectiveness brought by these components. While the proposed approach empirically achieved the overall best deblurring results, it takes a relatively long time for inference despite its small model size. In addition, though the atom kernels of RKR are learned from training data, they lack adaptivity to each test sample. Reduction on inference time and investigation of a more adaptive approach will be the focus of our future work.

References

[1] Mahmoud Afifi Abdullah Abuolaim and Michael S Brown. Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning. In *Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1231–1239, 2022. 1, 3, 6

[2] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. In *Proceedings of European Conference on Computer Vision*, 2020. 2, 3, 5, 6

[3] Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 2289–2298, 2021. 2

[4] Saeed Anwar, Zeeshan Hayder, and Fatih Porikli. Deblur and deep depth from single defocus image. *Machine Vision and Applications*, 32:1–13, 2021. 1, 2

[5] Stanley H Chan and Truong Q Nguyen. Single image spatially variant out-of-focus blur removal. In *Proceedings of IEEE International Conference on Image Processing*, pages 677–680. IEEE, 2011. 2

[6] Mingqin Chen, Yuhui Quan, Tongyao Pang, and Hui Ji. Nonblind image deconvolution via leveraging model uncertainty in an untrained deep neural network. *International Journal of Computer Vision*, 130(7):1770–1789, 2022. 2

[7] Sunghyun Cho and Seungyong Lee. Convergence analysis of map based blur kernel estimation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4818–4826, 2017. 1, 2

[8] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021. 5

[9] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *IEEE Transactions on Image Processing*, 25:1660–1673, 2016. 1, 2, 5

[10] D. A. Fish, A. M. Brinicombe, E.R. Pike, and J. G. Walker. Blind deconvolution by means of the richardson-lucy algorithm. *Journal of The Optical Society of America A-optics Image Science and Vision*, 12:58–65, 1995. 1

[11] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019. 2

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 5

[13] Israel Gohberg and Seymour Goldberg. *Basic operator theory*. Birkhäuser, 2013. 4

[14] Sunghyun Cho Hyeongseok Son, Junyong Lee and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 2642–2650, 2021. 1, 3, 6

[15] Hui Ji and Kang Wang. Robust image deblurring with an inaccurate blur kernel. *IEEE Transactions on Image Processing*, 21(4):1624–1634, 2011. 2

[16] Ali Karaali, Naomi Harte, and Claudio R Jung. Deep multiscale feature learning for defocus blur estimation. *IEEE Transactions on Image Processing*, 31:1097–1106, 2022. 1, 2

[17] Ali Karaali and Claudio Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing*, 27:1126–1137, 2018. 1, 2, 3

[18] Ali Karaali and Cláudio Rosito Jung. Svbr-net: A non-blind spatially varying defocus blur removal network. In *Proceedings of IEEE/CVF International Conference on Image Processing*, pages 566–570. IEEE, 2022. 1, 2

[19] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in Neural Information Processing Systems*, 22, 2009. 1

[20] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12214–12222, 2019. 1, 2, 3, 6

[21] Junyong Lee, Hyeongseok Son, Jaesung Rin, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021. 1, 3, 5, 6

[22] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3):70–es, 2007. 1, 2

[23] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C Eldar. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Transactions on Computational Imaging*, 6:666–681, 2020. 4

[24] Yuelong Li, Mohammad Tofighi, Vishal Monga, and Yonina C Eldar. An algorithm unrolling approach to deep image deblurring. In *Proceedings of IEEE/CVF International Conference on Acoustics, Speech and Signal Processing*, pages 7675–7679. IEEE, 2019. 4

[25] Jizhou Li Lingyan Ruan, Bin Chen and Miuling Lam. Aifnet: All-in-focus image restoration network using a light field-based dataset. *IEEE Transactions on Computational Imaging*, 7:675–688, 2021. 1, 3, 5, 6

[26] Yu-Qi Liu, Xin Du, Hui-Liang Shen, and Shu-Jie Chen. Estimating generalized gaussian blur kernels for out-of-focus image deblurring. *IEEE Transactions on Circuits and Systems for Video Technology*, 31:829–843, 2020. 1, 2

[27] Haoyu Ma, Shaojun Liu, Qingmin Liao, Juncheng Zhang, and Jing-Hao Xue. Defocus image deblurring network with defocus map estimation as auxiliary task. *IEEE Transactions on Image Processing*, 31:216–226, 2021. 1, 3

[28] Yuesong Nan and Hui Ji. Deep learning for handling kernel/model uncertainty in image deconvolution. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2388–2397, 2020. 2

[29] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2017. 1, 2, 3

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[31] Yuhui Quan, Zhuojie Chen, Huan Zheng, and Hui Ji. Learning deep non-blind image deconvolution without ground truths. In *Proceedings of European Conference on Computer Vision*, pages 642–659. Springer, 2022. 2

[32] Yuhui Quan, Zicong Wu, and Hui Ji. Gaussian kernel mixture network for single image defocus deblurring. *Advances in Neural Information Processing Systems*, 34:20812–20824, 2021. 1, 2, 3, 4, 6

[33] Wenqi Ren, Jiawei Zhang, Lin Ma, Jinshan Pan, Xiaochun Cao, Wangmeng Zuo, Wei Liu, and Ming-Hsuan Yang. Deep non-blind deconvolution via generalized low-rank approximation. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 2

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015. 4

[35] Lingyan Ruan, Bin Chen, Jizhou Li, and Miuling Lam. Learning to deblur using light field generated and real defocus images. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16304–16313, 2022. 1, 3, 5, 6

[36] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. 5

[37] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015. 1, 2, 3

[38] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 2015. 4

[39] Xin Tao, Hongyun Gao, Yi Wang, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 4

[40] Subeesh Vasu, Venkatesh Reddy Maligireddy, and AN Rajagopalan. Non-blind deblurring: Handling kernel uncertainty with cnns. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3272–3281, 2018. 2

[41] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 6

[42] Guodong Xu, Yuhui Quan, and Hui Ji. Estimating defocus blur via rank of local patches. In *Proceedings of IEEE International Conference on Computer Vision*, pages 5371–5379, 2017. 1, 2, 3

[43] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *ACM Transactions on Graphics*, 2008. 1, 2

[44] Anmei Zhang and Jian Sun. Joint depth and defocus estimation from a single image using physical consistency. *IEEE Transactions on Image Processing*, 30:3419–3433, 2021. 1, 2

[45] Dafeng Zhang and Xiaobing Wang. Dynamic multi-scale network for dual-pixel images defocus deblurring with transformer. In *Proceedings of IEEE/CVF International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2022. 3

[46] Jiawei Zhang, Jinshan Pan, Wei-Sheng Lai, Rynson WH Lau, and Ming-Hsuan Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3817–3825, 2017. 4

[47] Jie Zhang and Wanming Zhai. Blind attention geometric restraint neural network for single image dynamic/defocus deblurring. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 3

[48] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[49] Wenda Zhao, Fei Wei, You He, and Huchuan Lu. United defocus blur detection and deblurring via adversarial promoting learning. In *Proceedings of European Conference on Computer Vision*, pages 569–586. Springer, 2022. 1

[50] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3080–3088, 2018. 1, 2

[51] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:1884–1897, 2020. 1, 2

Neumann Network with Recursive Kernels for Single Image Defocus Deblurring (Supplementary Materials)

1. Visual Comparison of SIDD Results of LFDOF-Trained Models on CUHK-BD Dataset

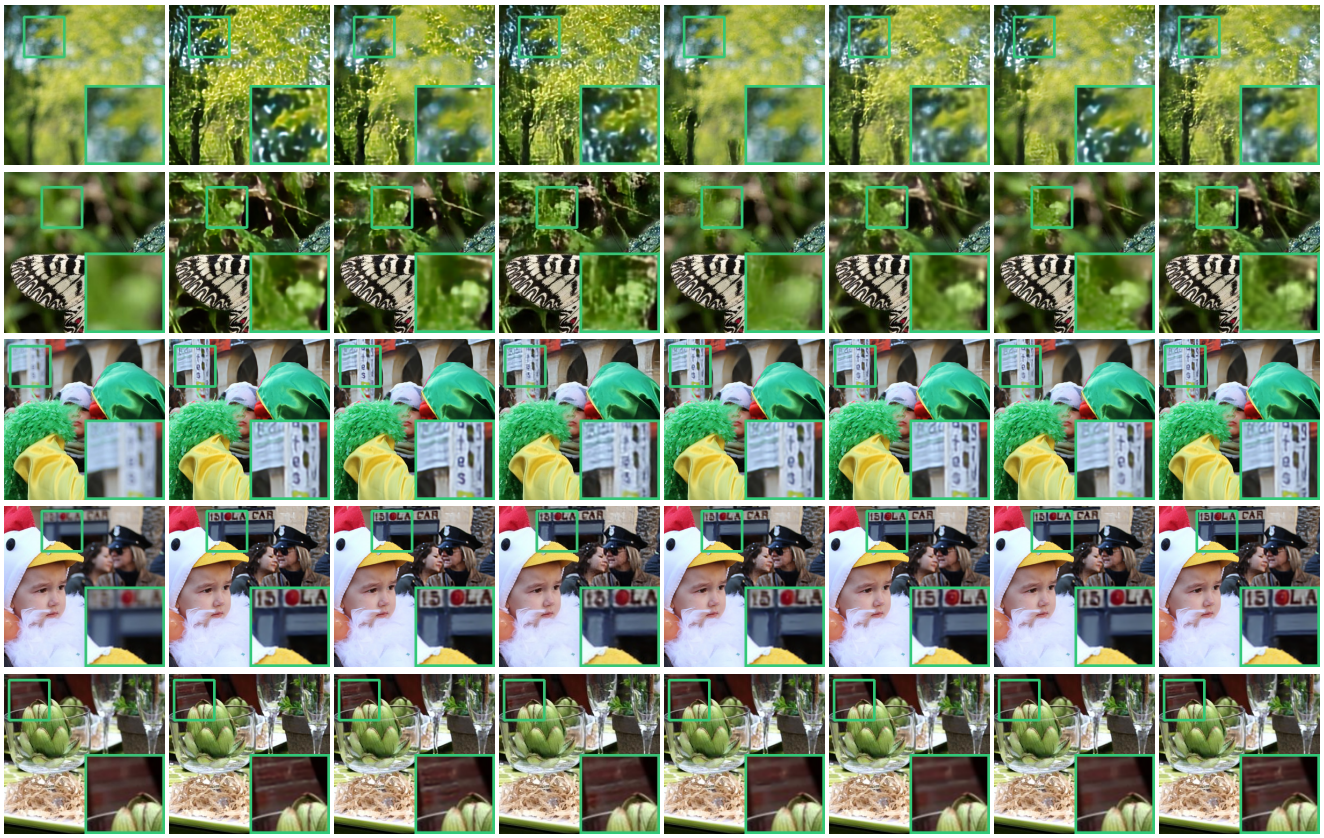

Figure 1. SIDD results from LFDOF-trained models on selected images from CUHK-BD dataset.

2. Code Link and Animated Versions of Visual Results

Our code is also published via the link https://github.com/csZcWu/NRKNet. We also supply animated versions of the visual results via this link for clarity and convenient sake.

3. Results of Ablation Studies on Other Datasets

Method	RealDOF			RTF			LFDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Input	23.890	0.725	0.349	24.200	0.717	0.248	25.874	0.777	0.320	24.200	0.717	0.248
NonAdaptive	24.501	0.756	0.372	24.738	0.807	0.246	30.360	0.881	0.154	27.753	0.883	0.166
NonRecursive	25.148	0.768	0.342	25.346	0.817	0.241	30.475	0.883	0.148	27.814	0.885	0.150
NonSeparable	**25.150**	**0.769**	**0.339**	25.538	0.825	0.231	30.450	0.883	0.150	27.968	0.888	0.149
FullRKR	25.148	0.768	0.340	**25.931**	**0.829**	**0.215**	**30.537**	**0.884**	**0.147**	**28.047**	**0.889**	**0.145**

Table 1. Results of ablation study on RKR using DPDD-trained models (left part) and LFDOF-trained models (right part).

Method	RealDOF			RTF			LFDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Input	22.333	0.633	0.524	24.200	0.717	0.248	25.874	0.777	0.320	24.200	0.717	0.248
FullExp	**25.149**	0.768	**0.340**	25.569	0.813	0.225	30.517	0.883	0.149	28.028	0.891	0.147
2Scales	24.914	0.755	0.356	25.561	0.812	0.248	29.497	0.864	0.180	26.975	0.848	0.235
4Scales	25.146	**0.769**	0.361	25.728	0.814	0.247	30.470	0.883	0.149	27.904	0.887	0.149
w/o LSTM	24.679	0.747	0.361	24.725	0.787	0.323	29.053	0.860	0.174	25.223	0.844	0.178
Original	25.148	0.768	**0.340**	**25.931**	**0.829**	**0.215**	**30.537**	**0.884**	**0.147**	**28.047**	**0.889**	**0.145**]

Table 2. Results of ablation study on DNN pipeline using DPDD-trained models (left part) and LFDOF-trained models (right part).

Method	RealDOF			RTF			LFDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Input	22.333	0.633	0.524	24.200	0.717	0.248	25.874	0.777	0.320	24.200	0.717	0.248
w/o$\mathcal{L}_{\text{reblur}}$	25.009	0.758	0.352	25.052	0.812	0.233	30.006	0.880	0.156	27.674	0.885	0.163
w/o FDR Loss	24.972	0.764	0.349	25.275	0.819	0.249	30.280	0.873	0.165	27.689	0.875	0.188
Single-scale Loss	24.785	0.749	0.363	25.173	0.797	0.278	29.296	0.861	0.185	26.487	0.832	0.275
Full Loss	**25.148**	**0.768**	**0.340**	**25.931**	**0.829**	**0.215**	**30.537**	**0.884**	**0.147**	**28.047**	**0.889**	**0.145**

Table 3. Results of ablation study on loss functions using DPDD-trained models (left part) and LFDOF-trained models (right part).

4. Visualization of Learned Atoms Kernels in RKR

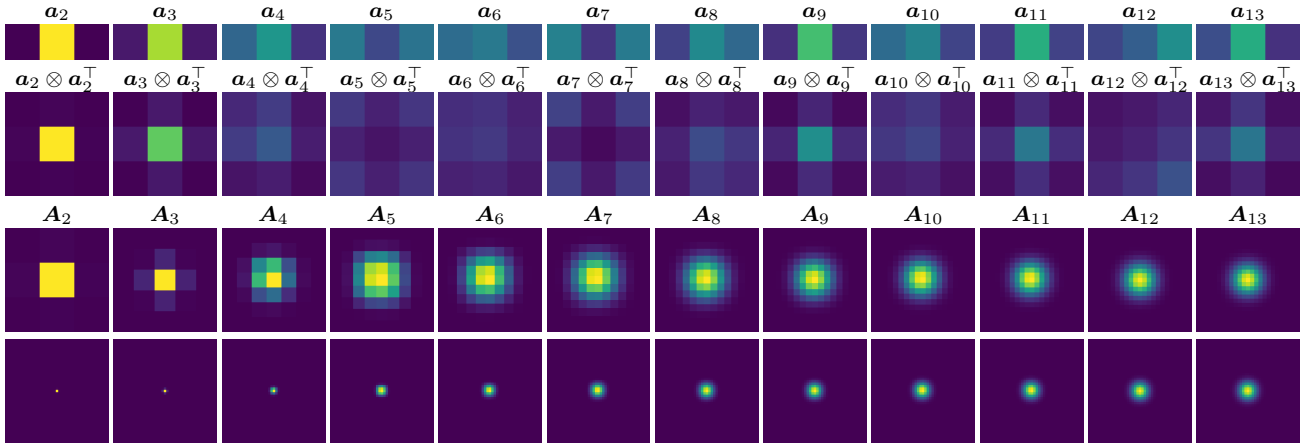

Figure 2. Visualization of learned adaptive kernels in our DPDD-trained NRKNet. The kernels shown in the 4th row are the ones of the 3rd row padded with zeros to have the same size.

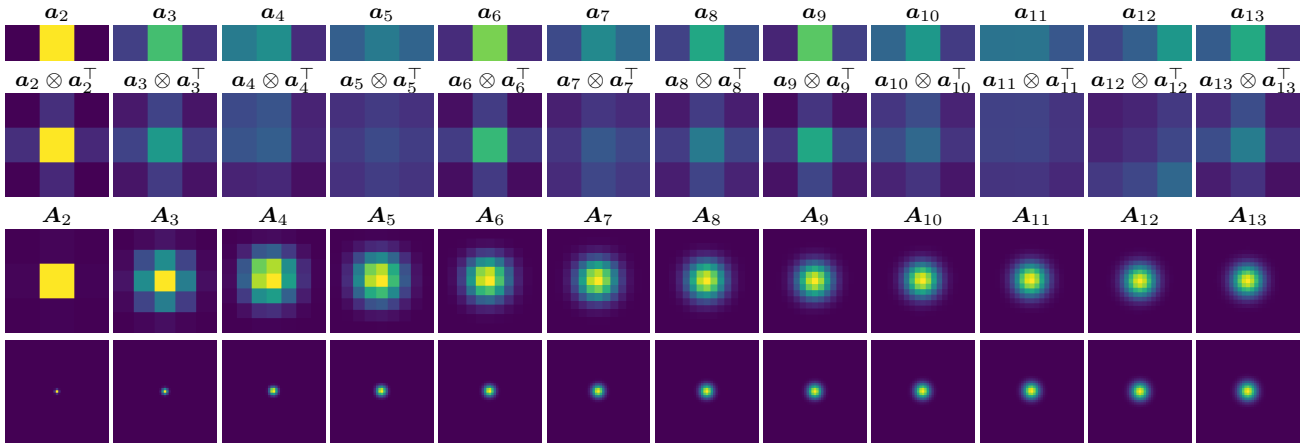

Figure 3. Visualization of learned adaptive kernels in our LFDOF-trained NRKNet. The kernels shown in the 4th row are the ones of the 3rd row padded with zeros to have the same size.

5. Visualization of Coefficient Maps

See Fig. 4 for the visualization of some coefficient maps at the original image scale. We can observe that the regions with larger blur amount tend to have larger coefficients on large-size kernels, and vice versa.

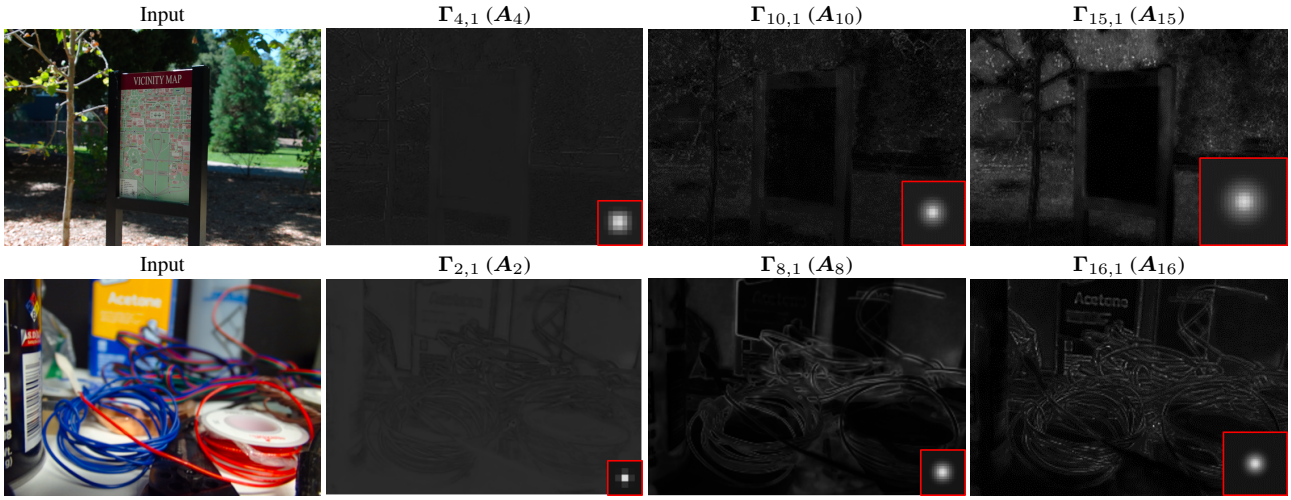

Figure 4. Visualization of coefficient maps and learned kernels.