Blind Image Super-resolution with Elaborate Degradation Modeling on Noise and Kernel

Zongsheng Yue¹, Qian Zhao², Jianwen Xie³, Lei Zhang⁴, Deyu Meng²,⁵, Kwan-Yee K. Wong¹

¹The University of Hong Kong, Hong Kong, China  ²Xi’an Jiaotong University, Xi’an, China  ³Cognitive Computing Lab, Baidu Research, Bellevue, USA  ⁴The Hong Kong Polytechnic University, Hong Kong, China  ⁵Peng Cheng Laboratory, Shenzhen, China

Abstract

While researches on model-based blind single image super-resolution (SISR) have achieved tremendous successes recently, most of them do not consider the image degradation sufficiently. Firstly, they always assume image noise obeys an independent and identically distributed (i.i.d.) Gaussian or Laplacian distribution, which largely underestimates the complexity of real noise. Secondly, previous commonly-used kernel priors (e.g., normalization, sparsity) are not effective enough to guarantee a rational kernel solution, and thus degenerates the performance of subsequent SISR task. To address the above issues, this paper proposes a model-based blind SISR method under the probabilistic framework, which elaborately models image degradation from the perspectives of noise and blur kernel. Specifically, instead of the traditional i.i.d. noise assumption, a patch-based non-i.i.d. noise model is proposed to tackle the complicated real noise, expecting to increase the degrees of freedom of the model for noise representation. As for the blur kernel, we novelly construct a concise yet effective kernel generator, and plug it into the proposed blind SISR method as an explicit kernel prior (EKP). To solve the proposed model, a theoretically grounded Monte Carlo EM algorithm is specifically designed. Comprehensive experiments demonstrate the superiority of our method over current state-of-the-arts on synthetic and real datasets. The source code is available at https://github.com/zsyOAOA/BSRDM.

1. Introduction

Single image super-resolution (SISR) is a fundamental problem in computer vision. It aims to recover the sharp detailed high-resolution (HR) counterpart from an observed low-resolution (LR) image. Image degradation, the functional opposite of image super-resolution, is the process of generating a LR image from the HR one. Unfortunately, the degradation model is always unknown while complicated, making the blind SISR problem extremely challenging. How to rationally and practically model the degradation is therefore of great significance in blind SISR.

Early methods [14, 24, 44, 45] simply regard SISR as an interpolation problem. They have fast processing speed but always blur high frequency details. Later methods begin to consider the image degradation, and can be roughly divided into two categories, namely model-based methods and learning-based methods. From the Bayesian perspective, model-based methods [11, 20, 25, 36, 40, 42] firstly build a generative model based on the image degradation and then estimate the blur kernel and the HR image under the maximum a posteriori (MAP) framework. Such MAP estimation is implemented for each LR image individually, and thus tends to achieve better generalization for unknown degradations. Learning-based methods [10, 25, 56, 65], on the other hand, aim to learn a unified super-resolver based on a large amount of LR/HR image pairs synthesized according to the pre-assumed degradation model. Recently, to improve their generalization, some works [8, 16, 28, 51, 54] attempt to learn the degradation model from unpaired real image data. However, these learning-based methods rely heavily on the collected training data, and may suffer from a severe performance drop when unseen degradations show up in testing. In this paper, we follow the model-based methodology for its better generalization capability.

Most of the model-based blind SISR methods can be generally formulated as the following MAP problem:

\[
\max_{x, k} \log p(y|x, k) + \log p(k) + \log p(x),
\]

(1)

where \(y\), \(x\), and \(k\) denote the observed LR image, the underlying HR image, and the blur kernel, respectively. The last term represents the image prior, while the first and second terms deliver our knowledges on the degradation model (i.e., noise distribution and kernel prior). Most of the existing methods focus on designing more rational image priors,
such as gradient profile [42], sparsity [7, 20], DIP [46] and so on [9, 11, 22, 32, 34, 38]. However, they often do not sufficiently consider the degradation model:

- As for noise modeling, most of current method adopt the independent and identically distributed (i.i.d.) Gaussian or Laplacian distribution to model the noise. Such a simple noise assumption, however, usually underestimates the complexity of real image noise and shows limited robustness in practical applications. For example, the most common camera sensor noise affected by the in-camera pipeline is signal-dependent, and thus exhibits evident non-i.i.d. property in statistics.

- As for kernel modeling, traditional methods often ignore it or adopt some heuristic priors, e.g., normalization (i.e., the kernel elements sum to 1) [17] and sparsity [3], which usually cannot guarantee a rational kernel solution. Recently, Liang et al. [25] trained an implicit mapping parameterized as a convolutional neural network (CNN) from the latent noises to anisotropic Gaussian kernels, and then embedded it into the blind SISR as a kernel prior. Albeit achieving evident performance improvement, this method depends on a time-consuming and labor-cumbersome pre-training phase. Moreover, the fitting error, which is inevitable during training, may be enlarged in the alternate iterations between the kernel estimation and super-resolution tasks. The performance of blind SISR can be therefore further improved by designing an explicit yet effective kernel prior.

To address the above issues, this paper proposes a probabilistic blind SISR method that elaborately considers the noise and kernel modeling (see Fig. 1). To better model the complicated real noise, a patch-based non-i.i.d. Gaussian noise assumption is adopted instead of the conventional i.i.d. one. Under such setting, each $p \times p$ image patch has its own noise parameter, which complies better with the configurations of real noise. As for blur kernel, we observe that it can be formulated as an explicit and differentiable function in terms of the covariance matrix. This inspire us to construct an explicit kernel prior (EKP) for the generally used anisotropic Gaussian kernel, which can be easily embedded into current deep learning (DL)-based blind SISR methods. In summary, the contributions of this work is three-fold:

1. Different from the commonly-used i.i.d. Gaussian or Laplacian distribution, a patch-based non-i.i.d. noise distribution is employed in the proposed method, making it able to handle complicated real noise.

2. A generative kernel prior named EKP is newly constructed for the blind SISR task. It is with explicit and concise form, and substantiated to be able to attain a more stable kernel estimation for SISR.

3. A theoretically grounded Monte Carlo EM algorithm (see Fig. 1) is designed to solve our proposed model.

2. Related Work

In this section, we briefly review the literatures on image degradation models and blind SISR.

2.1. Image Degradation Model

Image degradation model is a long-standing and open research topic in SISR. The most common and also simplest degradation model is bicubic downsampling, which is widely used to synthesize training and testing data in many SISR works [6, 11, 19, 23, 64]. More general degradation models, consisting of a sequence of blurring, downsampling and noise addition, are also widely adopted by previous works [37, 41, 58, 61, 65]. Recently, Zhang et al. [59] propose a more practical degradation model by introducing a random shuffle strategy among the blurring, downsampling and noise addition, and more practical camera sensor

Figure 1. An overview of the proposed SISR method and the corresponding EM algorithm. A probabilistic model is constructed to depict the generation process of the observed LR image, which mainly involves two groups of parameters, including the latent variable $z$ and the model parameters $\{\alpha, L, \lambda\}$. A Monte Carlo EM algorithm is designed to alternately update them in the E-Step and M-Step, respectively.
and JPEG compression noises in the noise addition procedure. Furthermore, Wang et al. [47] consider the common ringing and overshoot artifacts, and propose a high-order degradation model to cover a larger degradation space.

2.2. Blind SISR Methods

As mentioned in the introduction, other than the heuristic interpolation-oriented methods [14, 18, 24, 44], most of the existing methods can be loosely divided into two categories, namely model-based and learning-based methods. Even though this paper focuses on model-based methods, we also briefly review learning-based methods for completeness.

Model-based Methods. Model-based methods mainly focus on designing the three terms in Eq. (1), i.e., the likelihood, kernel prior, and image prior. The image prior has received more attentions in the past decades. Typical traditional image priors include total variation (TV) [38], hyper-Laplacian [22], gradient profile [42], sparsity [20], and non-local similarity [7]. With the prevalence of deep learning, more DL-base image priors have been proposed. A representative work is proposed by Ulyanov et al. [46], namely deep image prior (DIP), to capture the low-level image statistics. Shocher et al. [41] attempt to recover the HR image using the prior of patch recurrence across scales. More related works can be found in [2, 34].

Kernel prior is another important part in the MAP framework. Traditional blind SISR methods only consider some heuristic kernel priors, such as normalization [17] and sparsity [3]. Recently, some works begin to implicitly model the kernel by DNN. For example, Ren et al. [36] propose to model the kernel prior using a multilayer perceptron (MLP), while Liang et al. [25] train a flow-based kernel prior named FKP for blind SISR. Instead of such an implicitly modeling manner, this paper attempts to design an explicit and concise kernel prior, hoping to induce a more stable kernel estimation in blind SISR task.

As for the likelihood, most of the existing methods adopt an i.i.d. Gaussian or Laplacian distribution, which often fails to comply with the configurations of real noise and causes a performance drop in real scenarios. To address this issue, this work employs a non-i.i.d. noise modeling method to better deliver the real noise configurations and thus improve its generalized capability.

Learning-based Methods. The main idea of learning-based methods is to learn a super-resolver from large amount of pre-simulated LR/HR image pairs. Dong et al. [6] firstly propose to learn an end-to-end CNN mapping from LR to HR images. Later, plenty of CNN architectures are designed for SISR [13, 19, 26, 31, 43, 48, 64]. Recently, a flurry of unpaired SISR methods [8, 16, 28, 51, 54] have been proposed, due to the fact that real LR images rarely come with the corresponding HR images in practice.

3. The Proposed Method

3.1. Degradation Assumption

Various degradation models have been proposed in previous works. Most of them can be written as a downsampling with a subsequent noise addition process, i.e.,

\[
y = D(x; k, \downarrow_s) + n, \tag{2}
\]

where \(y\) and \(x\) denote the LR and HR images, respectively, \(D(x; k, \downarrow_s)\) represents the downsampling process with a blur kernel \(k\) and \(s\)-fold downsampler \(\downarrow_s\), and \(n\) is the noise. In fact, the real LR image may be also obtained by firstly adding noise and then downsampling the HR image [59], which makes the noise more complicated. This process can also be formulated in the same format as Eq. (2), i.e.,

\[
y = D(x + n; k, \downarrow_s) = D(x; k, \downarrow_s) + \hat{n}, \tag{3}
\]

where \(\hat{n} = D(n; k, \downarrow_s)\). Hence, we only need to consider the degradation sequence in Eq. (2).

For the blur kernel \(k\), we assume it to be the general anisotropic Gaussian kernel which is sufficient for SISR as pointed out in [37, 58]. Furthermore, considering different settings for the downsampler \(\downarrow_s\) (e.g., bicubic [19] and direct\(^1\) [58]) and the imposed order between the blurring and downsampling procedures (i.e., \((x \ast k) \downarrow_s\) [58] and \((x \downarrow_s) \ast k\) [62], where \(\ast\) is the convolution operator), we can obtain multiple different degradation assumptions based on Eq. (2). This paper aims to propose a blind SISR method with elaborate considerations on noise and kernel modeling, which does not depend on the format of the downsampler and the specific imposed order between the blurring and downsampling procedures. For the ease of presentation, we adopt the most widely used degradation assumption to construct our SISR model in the next subsection, i.e.,

\[
y = (x \ast k) \downarrow^d_s + n, \tag{4}
\]

where \(\downarrow^d_s\) is the direct downsampler with a scale factor \(s\).

3.2. Probabilistic SISR Model

In this subsection, we are going to build our blind SISR method based on the degradation model in Eq. (4).

Non-i.i.d. Noise Modeling. Different from the traditional i.i.d. Gaussian or Laplacian noise assumption on the whole image, a patch-based non-i.i.d. noise model is proposed in this work. Given any observed LR image \(y \in \mathbb{R}^{h \times w}\), where \(h\) and \(w\) denote the image height and width, respectively, we regard \(y\) as \(N(N = hw)\) highly overlapped \(p \times p\) patches. Furthermore, we assume that the noises contained in each patch obey a different zero-mean Gaussian distribution with

\(^1\)Direct downsampler with a scale factor \(s\) means keeping the upper-left pixel for each distinct \(s \times s\) patch and discarding the rest.
its own variance parameter. Specifically, considering the i-th image patch centered at \( y_i \), we have

\[
y_i \sim \mathcal{N} \left( y_i \left( \left( x + k \right)_{dx}^d, \lambda_i \right), \lambda_i \right), \quad i = 1, 2, \ldots, N,
\]

where \( \lambda_i \) is the noise variance for the i-th image patch.

In previous researches, they often assume the noise as additive white Gaussian noise (AWGN), which is indeed a special case of our non-i.i.d. noise distribution. By regarding the whole image as one large patch with size \( h \times w \), our noise model then naturally degenerates to AWGN, but with noise variance parameter being automatically updated during learning (see Sec. 4) instead of manually adjusted.

**Kernel Prior**. Based on the anisotropic Gaussian assumption on the blur kernel, we construct a concise yet effective kernel prior. For any blur kernel \( k \) with size \( (2r+1) \times (2r+1) \), it is defined as follows:

\[
k_{ij} = \frac{1}{2\pi} \sqrt{\lambda_i} \exp \left\{ -\frac{1}{2} S^T \Lambda S \right\}, \quad i, j \in \{-r, \cdots, r\},
\]

where \( \Lambda \) is the precision matrix, \( S = \begin{bmatrix} j \end{bmatrix} \) is the spatial coordinate. From Eq. (6), it can be observed that the blur kernel is completely determined by the precision matrix \( \Lambda \) after fixing the kernel size. Note that Eq. (6) is differentiable w.r.t. \( \lambda_i \). This implies that it can be regarded as a kernel generator, in which \( \Lambda \) can be easily optimized with stochastic gradient descent (SGD) under the DL framework.

Another tricky issue is how to guarantee the positive-definiteness of the precision matrix \( \Lambda \) during optimization. Inspired by the Cholesky decomposition, we reparameterize \( \Lambda \) as follows:

\[
\Lambda = LL^T,
\]

where \( L \in \mathcal{R}^{2 \times 2} \) is a lower triangular matrix. By substituting Eq. (7) into Eq. (6), we obtain the following explicit kernel prior termed EKP,

\[
k_{ij} = h(L) = \frac{1}{2\pi} |L| \exp \left\{ -\frac{1}{2} S^T LL^T S \right\}.
\]

In practice, to make \( L \) be triangular during optimization, we rewrite \( L \) as \( L = Q \odot M \), where \( M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \) and \( \odot \) is the Hadamard product, and turn to optimize \( Q \).

To our best knowledge, the most effective kernel prior for SISR is FKP [25]. The main idea of FKP is to firstly train a deep generator that maps the latent noises to anisotropic Gaussian kernels, and then use the pre-trained generator to estimate the blur kernel by only adjusting the latent noises. The inevitable fitting error of this generator may be enlarged when applying it in blind SISR, and thus limits the final performance. Comparing with FKP, the advantages of our proposed EKP is three-fold: 1) EKP is an explicit kernel generator that does not rely on pre-training, making it more convenient to be used in SISR. 2) The generated kernel by EKP is always an exact anisotropic Gaussian kernel, which naturally avoids the issue of fitting error in FKP. 3) In EKP, the kernel \( k \) is completely controlled by \( L \), which contains much fewer parameters than that of the latent noise vector in FKP (3 vs. \( 11^2/15^2/19^2 \) for scale 2/3/4, respectively). This makes EKP more easier to be optimized after plugging into the blind SISR as a kernel prior.

**Image Prior**. We employ a CNN-based generator \( G \) to generate the HR image from the latent space, i.e.,

\[
x = G(z; \alpha),
\]

where \( z \) and \( \alpha \) denote the latent variable and network parameters, respectively. As demonstrated in [46], \( G \) is very easy to overfit onto the image noise due to the powerful fitting capability of CNN. Therefore, we introduce the conventional hyper-Laplacian prior to constrain the statistical regularity of the generated HR image through the following joint distribution of \( \alpha \) and \( z \):

\[
(\alpha, z) \sim p(\alpha, z) = p(\alpha|z)p(z),
\]

\[
p(\alpha|z) \propto \exp \left( -\rho \sum_{k=1}^{2} |f_k + G(z; \alpha)|^\gamma \right),
\]

\[
p(z) = \mathcal{N}(z|0, I),
\]

where \( \{f_k\}_{k=1}^{2} \) are the gradient filters along the horizontal and vertical directions, \( \rho \) and \( \gamma \) are both hyper-parameters.

As for the generator \( G \), we follow the “hourglass” architecture in DIP [46] but use a tiny version that contains much fewer parameters. The detailed network architecture can be found in supplementary material (SM).

### 3.3. MAP Estimation

According to Eqs. (5)-(12), a full probabilistic model is constructed. Under the MAP framework, our goal turns to maximize the following posterior:

\[
p(\alpha, L, \lambda|y) \propto \int p(y|\alpha, L, \lambda, z)p(\alpha|z)p(z)dz.
\]

Note that we have omitted the prior terms \( p(L) \) and \( p(\Lambda) \), since they are set as non-informative priors in our model. Taking the logarithm of both sides of Eq. (13), we have the following maximization problem:

\[
\max_{\alpha, L, \lambda} \log p(\alpha, L, \lambda|y) = \log \int p(y|\alpha, L, \lambda, z)p(\alpha|z)p(z)dz + \text{const}.
\]

### 4. Inference Algorithm

Inspired by [52, 53], we design a Monte Carlo expectation-maximization (EM) algorithm [5] to solve
Eq. (14), which alternately samples the latent variable \( z \) from its posterior \( p(z|y) \) in E-Step and updates the model parameters \( \{ \alpha, L, \lambda \} \) in M-Step. The whole inference framework is illustrated in Fig. 1.

**E-Step.** Given current model parameters \( \{ \alpha_{\text{old}}, L_{\text{old}}, \lambda_{\text{old}} \} \), we denote the posterior of \( z \) under them as \( p_{\text{old}}(z|y) \). In E-Step, our goal is to sample \( z \) from \( p_{\text{old}}(z|y) \) using Langevin dynamics [50]:

\[
z^{(r+1)} = z^{(r)} + \frac{\delta^2}{2} \frac{\partial}{\partial z} \log p_{\text{old}}(z|y) \bigg|_{z=z^{(r)}} + \delta \zeta^{(r)}, \tag{15}
\]

where \( r \) indexes the time step for Langevin dynamics, \( \delta \) denotes the step size, \( \zeta \) is the Gaussian white noise used to prevent trapping into local modes. A key note to calculate Eq. (15) is \( \frac{\partial}{\partial z} \log p_{\text{old}}(z|y) = \frac{\partial}{\partial z} \log p_{\text{old}}(z, y) \), and the detailed calculation can be found in SM.

In practice, a small trick to accelerate the convergence speed of Monte Carlo sampling in Eq. (15) is to start from the previous updated \( z \) in each learning iteration. We empirically found that it performs very stably and well by simply sampling 10 times according to Eq. (15).

**M-Step.** Let’s denote the sampled latent variable in E-step as \( \tilde{z} \). M-Step aims to maximize the approximate lower bound of Eq. (14) w.r.t. the model parameters \( \{ \alpha, L, \lambda \} \):

\[
\max_{\alpha, L, \lambda} Q(\alpha, L, \lambda) = \int p_{\text{old}}(z|y) \log p(y|\alpha, L, \lambda, z) p(\alpha|z)p(z) dz
\approx \log p(y|\alpha, L, \lambda, \tilde{z}) p(\alpha|\tilde{z}) p(\tilde{z}). \tag{16}
\]

Equivalently, Eq. (16) can be reformulated into a minimization problem as follows:

\[
\min_{\alpha, L, \lambda} E(\alpha, L, \lambda) = \frac{1}{2} \left\| \frac{1}{\lambda} \odot \left\{ y - \left[ G(\tilde{z}; \alpha) * h(L) \right] \downarrow d \right\} \right\|_2^2
+ \rho \sum_{k=1}^3 \left\| f_k * G(\tilde{z}; \alpha) \right\|_2^2. \tag{17}
\]

To solve Eq. (17), we alternately update the model parameters \( \{ \alpha, L \} \) and \( \lambda \). Specifically, for \( \alpha \) and \( L \), they can be directly optimized by SGD based on the back-propagation (BP) algorithm [39]:

\[
W_{\text{new}} = W_{\text{old}} - \eta \frac{\partial}{\partial W} E(\alpha, L, \lambda), \quad W \in \{ \alpha, L \}. \tag{18}
\]

where \( \eta \) is the learning rate. Actually, we adopt the more advanced Adam [21] algorithm to update \( \alpha \) and \( L \) instead of the SGD strategy of Eq. (18), which empirically makes it converge much faster.

For the noise variance \( \lambda \), we consider \( \lambda_i \) in the \( p \times p \) patch centered at the \( i \)-th pixel. Fortunately, based on the i.i.d. Gaussian assumption within this image patch, we have the following closed-form solution for \( \lambda_i \):

\[
\lambda_i = \frac{1}{p^2} \sum_{j \in N(i)} \left\{ y_j - \left[ G(\tilde{z}; \alpha_{\text{old}}) * h(L_{\text{old}}) \right] \downarrow d \right\}_j \right\}_j^2, \tag{19}
\]

where \( N(i) \) is the index set of the pixels in the \( p \times p \) patch centered at \( i \).

It should be noted that the first term of (17) can be regarded as a re-weighted \( L_2 \) loss with weight \( \frac{1}{\lambda} \), which is automatically updated through Eq. (19) during optimization.

Detailed description of the proposed EM algorithm is presented in Algorithm 1.

### 5. Experimental Results

We conducted extensive experiments to verify the effectiveness of the proposed method in this section. For ease of presentation, we briefly denote our blind super-resolution method with elaborate degradation modeling on noise and kernel as BS-RDM in the rest of this paper.

#### 5.1. Experimental Setup

**Model Settings.** Throughout the experiments, we empirically set the hyper-parameters \( \rho \) and \( \gamma \) to be 0.2 and \( 2/3 \), respectively. The setting on \( \gamma \) lies on the fact that the hyper-Laplacian with exponent \( \gamma = 2/3 \) is a better model of image gradient than a Laplacian or Gaussian [22]. To update the model parameters \( \alpha \) and \( L \) in M-Step, the Adam [21] algorithm with default settings in Pytorch [35] is used. The learning rates for \( \alpha \) and \( L \) are set as \( 2e^{-3} \) and \( 5e^{-3} \), respectively. As for the patch size \( p \) of the noise model, we provide two different settings. For the synthetic Gaussian noise in Sec. 5.2, we regard the whole image as one special image patch. While for synthetic camera sensor noise in Sec. 5.2 and real image noise in Sec. 5.3, we set \( p = 15 \). For fair comparison, the quantitative results of our method are averaged by running it five times with different random seeds.

**Comparison Methods.** To evaluate BS-RDM, we compare it against five methods, including one learning-based method RCAN [64], and four model-based methods, namely CSC [11], ZSSR [41], DoubleDIP [36], and
DIPFKP [25]. Specifically, RCAN is a blind SISR method trained under the bicubic degradation; CSC attempts to recover high frequency image details using convolutional sparse coding; ZSSR is a zero-shot method that exploits the patch recurrence across scales in a single image; DoubleDIP and DIPFKP are both blind SISR methods but with different kernel priors. In the synthetic experiments of Sec. 5.2, we consider both blind and non-blind settings for ZSSR, and denote them as “ZSSR-B” and “ZSSR-NB”, respectively. For ZSSR-B, we use the default setting of its official code, in which the degradation model is assumed to be a bicubic downsampler followed by AWGN noise. While for ZSSR-NB, the ground truth blur kernels are pre-provided by us. Noted that ZSSR, DoubleDIP, DIPFKP, and BSRDM all employ deep CNN to generate HR image. Thus the comparison with them can better verify the marginal effects brought up by the noise and kernel modeling in BSRDM.

### 5.2. Evaluation on Synthetic Data

In this part, we quantitatively evaluate different methods on two commonly-used datasets, i.e., Set14 [57] and DIV2K100 [1]. DIV2K100 contains 100 high resolution images of the validation set of DIV2K, and we crop a $1024 \times 1024$ patch around the center from each image in our experiments due to GPU memory limitation. The LR images are synthesized via Eq. (4). To conduct a thorough comparison, we consider diverse degradations combined with different blur kernels and noise types. For blur kernels, two isotropic Gaussian kernels with different widths (i.e., 1.2 and 2.0) and four anisotropic Gaussian kernels are chosen as shown in Fig. 2. Furthermore, we consider two noise types as follows:

- **Case 1**: Gaussian noise with noise level 2.55, which is widely used in current SISR literatures [55, 58].
- **Case 2**: Camera sensor noise simulated by [4, 12], one typical example is shown in Fig. 3.

Especially, the noise in Case 2 is very close to real camera noise, and is thus suitable for evaluating different methods under the degradations with complicated real noise. As for the quantitative metrics, except for the commonly-used PSNR and SSIM [49], we also adopted LPIPS [63] to compare the perceptual similarity between the recovered HR image and ground-truth. Note that PSNR and SSIM are calculated in the luminance channel like most of the SISR literatures, while LPIPS is directly calculated in RGB channels.

### Comparison with SotA Methods

Table 1 lists the PSNR, SSIM, and LPIPS results of different methods under diverse degradations on Set14. The comparison on DIV2K100 can be found in SM. From Table 1, we can see that the proposed BSRDM achieves the best or at least the second best results for all degradations. For the degradation with scale factor 2 and Gaussian noise, ZSSR-NB achieves the best performance. While for the degradation with scale factor 2 and camera sensor noise, BSRDM outperforms ZSSR-NB, indicating that BSRDM is able to handle more complicated noise due to its non-i.i.d. noise modeling. Comparing with current state-of-the-arts (SotA) method DIPFKP, the evi-
Figure 4. Super-resolution results of different methods for two degradations with Gaussian noise (top row) and camera sensor noise (bottom row) under scale factor 3 on Set14. The blur kernel is shown on the upper-right corner of the zoomed LR image.

Figure 5. Three super-resolution results of different methods on real LR images with scale factor 4. Please zoom in for best view.

Ablation Studies. The core contributions of this paper mainly include the non-i.i.d. noise modeling manner and the constructed kernel prior EKP. To justify their effectiveness, we design two baseline methods. In the first baseline (denoted as Baseline1), we replace the non-i.i.d. noise assumption with the conventional i.i.d. one. Similarly, in the second baseline (denoted as Baseline2), the proposed EKP kernel prior is replaced with FKP [25], which is the current most effective kernel prior to our best knowledge.

We compare BSRDM with these two baselines on different degradations that combine the six blur kernels in Fig. 2 and camera sensor noise under scale factor 2 on Set14. The detailed results are listed in Table 2. Firstly, comparing with Baseline1, the performance gain of BSRDM is mainly brought up by the non-i.i.d. noise assumption, which makes

doubtedly superiority of BSRDM demonstrates the importance of the noise and kernel modeling in SISR, since they both use the same network architecture to generate HR image even though BSRDM has fewer parameters (see Sec. 5.4).

Two visual results on Gaussian noise (top row) and camera sensor noise (bottom row) are shown in Fig. 4. Note that we only display the five best methods due to page limitation, and the complete results can be found in SM. We can easily observe that: 1) In the case of Gaussian noise, all the comparison methods can remove such simple AWGN noise. Due to the better kernel modeling, the proposed BSRDM evidently achieves sharper results. 2) Under camera sensor noise, the recovered images of the four comparison methods still contain some obvious noises or artifacts, mainly because their i.i.d. Gaussian noise assumption largely deviates from the true noise distribution. On the contrary, BSRDM is able to remove most of the noises and preserves clear image details. This demonstrates the effectiveness of the proposed non-i.i.d. noise assumption under the complicated noise.
it be able to better deal with such signal-dependent camera sensor noise. Secondly, the superiority of BSRDM over Baseline2 indicates that our proposed kernel prior EKP is more effective than FKP as analysed in Sec. 3.2.

Figure 6 displays the estimated blur kernels by our method in different iterations during optimization. Note that the blur kernel is initialized as an isotropic Gaussian kernel with width \( s \) (i.e., the 1st iteration), where \( s \) is the scale factor. From this figure, we can see that the kernel is gradually adjusted toward the ground truth. After 300 iterations, the estimated kernel is very close to the ground truth, which facilitates a good super-resolution result.

### 5.3. Evaluation on Real Data

To further justify the effectiveness of BSRDM in real SISR task, we evaluate it on RealSRSet [59], which contains 20 real images from internet or the existing testing datasets [15, 29, 30, 60]. Figure 5 shows three typical examples that include different scenarios in SISR, i.e., natural image (top row), cartoon image (middle row), and text image (bottom row). It can be easily seen that the proposed BSRDM achieves evidently better visual results than the other comparison methods. In the first and second examples (bottom and middle row of Fig. 5), the LR images contain some obvious camera sensor noises or artifacts. Most of the comparison methods cannot finely deal with these cases, and tend to enlarge the noises or artifacts after super-resolution. The proposed BSRDM is able to remove most of these noises or artifacts and preserve clearer image structures due to its more robust non-i.i.d. noise modeling. For the commonly-used “chip” example (bottom row of Fig. 5) in SISR, the super-resolution results of the comparison methods are all very blurry, which may be caused by the fact that the estimated kernel does not match with the true one. On the contrary, BSRDM can obtain a relatively sharper and cleaner HR image because the proposed EKP makes it easier to estimate a rational blur kernel.

As pointed out by [59], we also find that current non-reference metrics (e.g., NIQE [33], NRQM [27], and PI [1]) are not consistent with our perceptual visual system in real SISR task. We put the detailed quantitative comparisons in terms of non-reference metrics and more visual results in SM due to page limitation.

### 5.4. Comparison on Model Size and Runtime

Table 3 lists the comparison results on model size (number of parameters) and runtime with existing model-based SISR methods. For fair comparison, we consider three typical methods (i.e., ZSSR, DoubleDIP, and DIPFKP) that are all accelerated by GPU, and the runtime results in Table 3 are tested on a GeForce RTX 2080 Ti GPU. Specifically, we fix the LR image size as \( 256 \times 256 \) and count the elapsed time of super-resolving it to size of \( 512 \times 512, 768 \times 768, \) and \( 1024 \times 1024 \) with scale factor 2, 3, and 4, respectively. From Table 3, it can be easily observed that: 1) Our BSRDM has a moderate number of parameters comparing with other methods. 2) Even though BSRDM contains more parameters than ZSSR, it still has the similar speed with ZSSR. What’s more, BSRDM is a little faster than ZSSR under scale factor 4. 3) Comparing with current SotA method DIPFKP, BSRDM is not only with faster speed but also much fewer parameters. Taking all of the comparisons on model size, runtime, and the performances on SISR into consideration, it should be rational to say that BSRDM is effective and potentially useful in real applications.

### 6. Conclusion

In this paper, we have proposed a new blind SISR method under the probabilistic framework, which elaborately considers the degradation modeling on noise and kernel. Specifically, to better fit the complicated real noise, a patch-based non-i.i.d. noise distribution is adopted in our method. As for the blur kernel, we construct an explicit yet effective kernel prior named EKP and apply it in the proposed method. Through extensive experiments, we have verified the effectiveness and superiority of the proposed method on synthetic and real datasets. We believe that this work can benefit the blind SISR research community.

**Acknowledgement.** This work was partially supported by the National Key R&D Program of China (2020YFA0713900), the Hong Kong RGC GRF grant (project # 17203119), the Macao Science and Technology Development Fund under Grant 061/2020/A2, the China NSFC projects under contracts 61721002 and U1811461.


References


