Unsupervised Blur Kernel Estimation and Correction for Blind Super-Resolution

YOUNGSOO KIM¹, (Member, IEEE), JEONGHYO HA², (Member, IEEE), YOOSHIN JO¹, (Member, IEEE), AND JUNMO KIM¹², (Member, IEEE)

¹School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
²Division of Future Vehicle, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Junmo Kim (e-mail: junmo.kim@kaist.ac.kr).

ABSTRACT Blind super-resolution (blind-SR) is an important task in the field of computer vision and has various applications in real-world. Blur kernel estimation is the main element of blind-SR along with the adaptive SR networks and a more accurately estimated kernel guarantees a better performance. Recently, generative adversarial networks (GANs), comparing recurrence patches across scales, have been the most successful unsupervised kernel estimation methods. However, they still involve several problems. The sharpness discrimination ability has been noted as being too weak, causing them to focus more on pattern shapes than sharpness. In some cases, kernel correction processes were omitted; however, these are essential because the optimally generated kernel may be narrower than a point spread function (PSF) except when the PSF is ideal low-pass filter. Previous studies also did not consider that GANs are affected by the thickness of edges as well as PSF. Thus, in this paper, 1) we propose a degradation and ranking comparison process designed to induce GAN models to become sensitive to image sharpness, and 2) propose a scale-free kernel correction technique using Gaussian kernel approximation including a thickness parameter. To improve the kernel accuracy further, we 3) propose a combination model of the proposed GAN and DIP (deep image prior) for more supervision, and designed a kernel correction network to propagate gradients through developed correction method. Several experiments demonstrate that our methods enhanced the $l_2$ error and the shape of the kernel significantly. In addition, by combining with ordinary blind-SR algorithms, the best reconstruction accuracy was achieved among unsupervised blur kernel estimation methods.

INDEX TERMS Blind super-resolution, kernel estimation, kernel correction, generative adversarial networks, deep image prior

I. INTRODUCTION

Super-Resolution (SR) is a fundamental low-level vision problem that is being actively studied and used in various practical applications such as image and video enhancement, surveillance, and medical imaging. The SR task enhances the quality of images and videos by recovering natural and detailed high-resolution patterns, as close to their original shape as possible, from a degraded low-resolution (LR) image $I_{LR}$ or video. With the rapid advancement of deep learning methods, research on image SR study also shifted toward supervised learning models. Since [27] had applied deep convolutional networks to SR, many studies have proposed convolutional SR networks [9], [15], [17]–[26], [28], and have achieved state-of-the-art performances on most reconstruction benchmarks. In image SR research, $I_{LR}$ is generally degraded by a blur kernel $k_{blur}$ and additive noise, and it can be expressed as shown in Eq. (1).

$$I_{LR} = (I_{HR} * k_{blur}) \downarrow_s + n$$ (1)

Many previous image SR studies were performed under the assumption that the blur kernel is known, and usually a bi-cubic kernel is used as the blur kernel (SISR). However, in practical applications, degradation information such as blur kernel is generally unknown, which significantly increases the difficulty of SR. For example, a network trained with low- and high-resolution (LR and HR, respectively) pairs created by bi-cubic degradation encountered a problem in that the blur of $I_{LR}$ propagated blurred the output as well. This blur kernel mismatch in the training and evaluation stages results in artifacts such as exaggerated textures or blurry SR output.
The goal of blind-SR is to recover $I_{HR}$ from $I_{LR}$ of which blur kernel is unknown. Generally, blind-SR can be divided into two parts: blur kernel estimation in $I_{LR}$ and adaptive SR network according to the blur kernel input. The fundamental solution to the unknown blur kernel problem is to predict kernel information exactly and apply it effectively. Usually, a more accurately estimated kernel information ensures better performance. However, accurately predicting the blur kernel of an image is very difficult because the pattern varies slightly over a given image according to blur kernels, and similar patterns appear with varying sharpness in a given image or in different images. Recently, GAN methods mining kernel information from inherent recurrence patches have been successful in estimating blur kernel information in an unsupervised manner [1], [6], [10], [29]. KernelGAN [6] implicitly compares arbitrary patches across scales to find recurrence patches and generate an optimal blur kernel that equals the sharpness domain of patches in an image.

The inherent recurrence property provides a strong cue for extracting a blur kernel from an image itself [1], [6], [10]. It utilizes recurrence patches across scales in $I_{LR}$ and $I_{LR2}$, degraded and downscaled from $I_{LR}$ by $k$. [1] showed that if we can find $k$, which places recurrence patches in the same sharpness domain, then $k$ is the kernel by which $I_{LR}$ is degraded from $I_{HR}$. When PSF is an ideal low-pass filter, $k$ represents PSF, and when it is not, $k$ is narrower than PSF, but otherwise resembles it. Utilizing this information, the GAN can find the optimal $k$ by implicitly matching and optimizing the recurrence pairs [6], [10]. However, previous GAN methods have a problem in that they did not adopt the proper kernel correction method, which is essential because PSF is not ideal as a low-pass filter. KernelGANs [6], [10] post-processed with a dilated convolution to transform $2 \times 2$ scale kernel to $2 \times 4$ scale or discarded negligible values near zero. In addition, KernelGANs are affected by edge thickness because edges are thinner in $I_{LR2}$ and $k$ may compensate for the edge shape too, as shown in Fig. 3. In the ablation study (Fig. 9), we showed that the GAN focuses on edges more than the other regions. To address the narrow kernel problem, we propose a simple and effective kernel correction method that resizes the kernel and removes unintended isotropic kernel caused by edge thickness from the estimated kernel by Gaussian kernel approximation in Section. III-A. The proposed correction method adheres to the GAN methods and can be applied to various SR scales without training.

A deep network is less sensitive to fine variations such as noise than the shape of patterns [14]; likewise comparing the fine sharpness variations on a pattern is typically difficult for discriminators. In previous GAN models, no additional mechanisms focus on sharpness. To address this problem, we add another supervision for the discriminator to learn sharpness of patches in Section. III-D. With “less real” and “more fake” images that are degraded from “real” and “fake” by blur processes, we trained the discriminator to learn to distinguish variations between “less real/more fake” and “real/fake” in a given pixel position. The proposed methods make GAN to focus on the sharpness and to extract more accurate blur kernel (Fig. 8, Table. 1).

In addition, to enhance the accuracy further, we combined the proposed method with DIP in Section. V. We expected two methods to have synergy in that, for kernel prediction, DIP compares a predicted LR image and a given LR image on same pixel position, while GAN implicitly compares the fake patches with learned real patches on arbitrary positions. With combining two algorithm, we could improve the kernel accuracy further (Fig. 12). The main contributions of this paper can be summarized as follows:

- We proposed a simple and effective kernel correction method using Gaussian kernel approximation. The correction method can be applied to various SR scales even when it is a real number without changing model or additional training. Kernel correction method includes edge thickness parameter so that it can remove unintended isotropic component in optimal kernel.
- We proposed a degradation and ranking comparison process to induce GAN models to become sensitive to image sharpness. With ablation study that varying kernel angle, we showed that proposed method (E-KernelGAN) enhanced the discrimination ability significantly and investigated where the GAN focuses.
- With combining the proposed methods and ordinary blind-SR methods, we achieved the best blind-SR reconstruction accuracy on DIV2KRK dataset comparing to previous kernel estimation methods.
- We executed further research that combining DIP and proposed GAN (E-KernelGAN-DIP). To propagate the gradient from DIP to generator, we trained simple linear networks with random kernel and corrected kernel pairs, and replaced the proposed kernel correction method. By combination, the kernel accuracy were enhanced especially in the failure cases of proposed GAN (E-KernelGAN).
II. BACKGROUND
In many studies, blind-SR consist of kernel extraction and a degradation-aware SR network [2]–[8], [10], [11], [4], [5] trained a specific SR network to a specific blur kernel so that the SR network didn’t need any mechanism for blur kernel adaptation. To train SR networks adaptively to various blur kernels, [7], [8] appended degradation information to \( I_{LR} \) input as a short code and [2], [3] connect the blur kernel information to middle layers of SR networks. As many previous researches shown, kernel adaptive SR networks is very important for blind-SR, likewise blur kernel estimation also is an indispensable element for blind-SR. It is well known that the blur kernel mismatch significantly decrease SR performance [2], [12] and more accurately estimated blur kernel guarantees better SR performance.

Many approaches have been proposed in recent years to estimate the blur kernel based on the learning method [2], [3], [7] and on the recurrence patches across \( I_{LR} \) and \( I_{LR2} \) [1], [6], [10]. [7] trained kernel discriminator network to predict the artifacts map due to an incorrect blur kernel and extracted a blur kernel that lowers the artifacts map. However, there was stability problem that in sometimes miss predicted artifacts map cause too sharp Gaussian blur kernel. [2] directly trained blur kernel to kernel extraction network with correction mechanism, and [3] also trained network directly in spatially variant degradation scheme. However, learning based methods have common problem that, for each resolution scale, network should be trained with each training dataset of that scale. Recently, in several researches, prediction methods that extract blur kernel precisely with one evaluation image by inherent recurrence property were studied [1], [6], [10].

As illustrated in upper of Fig. 1, [1] showed that if \( l[n] \) is generated from continuous scene and point spreading function(PSF) \( b_1 \), and \( b_2 \) is a scaled-down version of \( b_1 \), by optic zoom-in, i.e., \( b_2(y) = \alpha b_1(\alpha y) \), and \( \alpha \) and \( \alpha \) are related by \( k \), \( l = (h * k) \downarrow \alpha \), and \( \alpha \) there are a pattern \( q[n] \) and a \( \alpha \) scale larger resemblance pattern \( r[n] \) related by a kernel \( \hat{k} \) in \( l[n] \), i.e., \( q[n] = (r[n] * \hat{k}[n]) \downarrow \alpha \), then the kernel \( \hat{k} \) is an optimal kernel representing \( k[n] \) of \( l[n] = (h[n] * k[n]) \downarrow \alpha \). Especially, when PSF is an ideal low-pass filter, estimated kernel \( \hat{k} \) is equal to the PSF \( b_1 \). However, in most of the case, \( \hat{k} \) is narrower than PSF (in Eq. (6) of [1]). Simply put, when \( \alpha = 2 \), if \( r[n] \) is generated from step edge and Gaussian PSF with \( \alpha \), then \( \hat{k} \) is also Gaussian with \( \sqrt{\alpha} / 2 \). The detailed explanation and proof are in Section 2, 3 of [1].

To extract blur kernel with inherent recurrence patches, [1] searched recurrence patch pair candidates explicitly, \( q[n] \) and \( r^\alpha[n] \), and optimized with MAP to find \( \hat{k} \). However, it was fragile to image noise and execution time was exponentially increased for a large image. With GAN, [6], [10] searched recurrence patches implicitly and optimized kernel generator, and showed that GAN architecture is efficient and effective in extracting blur kernels.

If a kernel is extracted strictly based on above property, then the kernel correction method is essential to find PSF from extracted kernel. However, in previous GAN studies, kernel correction processes were omitted. Moreover, prior GAN studies did not consider that GAN models are affected by the thickness of edges as well as the PSF. Thus, in Section III-A, to address the problem, we developed a simple scale-free kernel correction method by resizing the kernel and removing the unintended isotropic kernel component caused by the edge thickness from the estimated kernel (Section III-A and III-B).

As a further research, we considered combining GAN and DIP in Section. V. [13], [14] showed image restoration can be performed by the prior of convolutional network without large dataset or pretraining. [10] applied DIP to predict kernel and SR image concurrently with flow-based kernel prior showed that DIP can support kernel prediction. DIP compares images in a same pixel position, while GAN compares in arbitrary positions implicitly, and we expected the difference of two methods may lead synergy and showed proposed methods are very effective in blur kernel prediction, especially for failure cases in Fig. 12.

III. ENHANCED-KERNELGAN
A. KERNAL CORRECTION
In this section, we developed simple and practical kernel correction method with several assumptions and simplifications. As seen in middle of Fig. 1, we have shifted our perspective to the way GAN solves the problem. In this view, \( I_{HR} \) is very clean image by ideal PSF and \( I_{LR} \) is downsampled from \( I_{HR} \) by \( k_1 \), target blur kernel. Given \( I_{LR} \), to find \( k_1 \) is the goal of E-KernelGAN (Enhanced-KernelGAN). GAN downscales \( I_{LR} \) to \( I_{LR2} \) with \( k_2 \) and optimize \( k_2 \) so that the sharpness of \( I_{LR2} \) equal to that of \( I_{LR} \) (in Fig. 4).

That optimal \( k_2 \) is resemblance of \( k_1 \) except width was shown in [1], and that \( k_2 \) from GAN contains a lot of blur kernel information was shown in [6], [10]. Thus, we will start from GAN’s view just to derive the correction method that transform \( k_2 \) to \( k_1 \).

In bottom of Fig. 1, we simplified the problem by limiting the search and comparison area to the center of a step edge. \( I_{HR} \), \( I_{LR} \) and \( I_{LR2} \) are one-dimensional edges in equal position. We will assume \( I_{HR} \) generated from convolution of a step edge \( E(x) \) and kernel \( k_{\sigma_1}(x) \) that determines edge thickness in figure. If \( \sigma_1 \) is larger, then edge of \( I_{HR} \) will be thicker. To simplify the problem more, all kernels are considered as Gaussian blur kernel in one-dimensional(1D) space Eq. (2). Our first goal is to find the relation between \( \sigma_1 \) and \( \sigma_2 \).

We start from a premise. \( I_{LR}(x) = I_{LR2}(x) \), that GAN finds the optimal kernel \( k_{\sigma_2}(x) \).

\[
k_{\sigma}(x) = \frac{1}{Z} \exp(-x^2/2\sigma^2) \quad (2)
\]

\[
(I(x) \ast k_{\sigma}(x)) \downarrow \alpha = I(s \cdot x) \ast k_{\sigma/\alpha}(x) \quad (3)
\]

\[
E(s \cdot x) = E(x) \quad (4)
\]
From $I_{LR}(x) = I_{LR2}(x)$, Eq. (5) is derived, through direct calculation by Eq. (3), (4) and convolution of Gaussian. (details are in Appendix.) Ep. (5) is the relation between $\sigma_1$ and $\sigma_2$ according to $\sigma_\alpha$. When the scale factor $s$ is 2 and $\sigma_\alpha$ is 0, $\sigma_2$ is equal to $\sqrt{\frac{3}{2}}\sigma_1$ as seen in Section II.

$$\sigma_1^2 = \frac{s^2}{3}\sigma_2^2 - \sigma_\alpha^2$$ (5)

With the relationship between $\sigma_1$ and $\sigma_2$, the kernel resizing parameter $\gamma$, Eq. (8) that corrects $\sigma_2$ to $\sigma_1$, is derived through Eq. (6) and (7).

$$k_{\sigma_1}(x) = k_{\sigma_2}(x)_{\text{resized with } \gamma}$$ (6)

$$\frac{x^2}{\sigma_1^2} = \frac{x^2/\gamma^2}{\sigma_2^2}$$ (7)

$$\gamma(s, \sigma_\alpha^2) = \frac{s}{\sqrt{3}} \sqrt{1 - \frac{\sigma_2^2}{\sigma_1^2}}$$ (8)

**Variance Calculation:** The resizing variable $\gamma$, in Eq. (8) changes according to the ratio $\sigma_\alpha/\sigma_2$. optimal $\sigma_\alpha$ varies spatially depending on each edge. However, a kernel will be extracted as $k_2$, i.e., $k_{\sigma_2}(x)$, per a image, thus we will get $\sigma_\alpha$ as a scalar for a model from validation set. Because $\sigma_\alpha$ is fixed, $\gamma$ changes according to the value of $\sigma_2$. As $\sigma_2$ increases, $\gamma$ increases and decreases as $\sigma_2$ decreases. We will simply determine 1-Dimensional discrete Gaussian distribution as Eq. (9) with the interval $d$ meaning pixel size, and the ratio between $k[n]$s are shown in Eq. (10). Finally, to measure $\sigma_2$, we got $\eta$ as Eq. (11) through a simple process and derived Eq. (12) for resizing parameter for kernel correction. For $\eta_\alpha$, we got 0.5, i.e., 0.1 for $\sigma_\alpha$, from validation set and we used it for whole data and scale factors.

$$k[n] = \frac{1}{2}\exp\left(-\frac{\eta \cdot n \cdot d^2}{2\sigma_2^2}\right) \cdot d$$ (9)

$$\frac{k[0]}{k[1]} = \frac{\exp\left(\frac{d^2}{2\sigma_\alpha^2}\right)}{\exp\left(\frac{3d^2}{2\sigma_\alpha^2}\right)}$$ (10)

$$\eta = \frac{\sigma^2}{2d^2} = \frac{1}{\ln(k[0]/k[1]) + \ln(k[1]/k[2])}$$ (11)

$$\gamma(s, \eta_\alpha) = \frac{s}{\sqrt{3}} \sqrt{1 - \frac{3}{s^2}\eta_\alpha}$$ (12)

**B. ANALYSIS OF THE RELATION BETWEEN EDGE THICKNESS AND PARAMETER $\sigma_\alpha$**

From GAN process, we could expect that generator may compensate reduced edge thickness caused by downsampling as well as reduced kernel variance to make fake edge close to real on equal position across scale. To analyze that, we designed ablation study with synthesized image (1600x1200 size). Image consist of eight large (radius=200 pixel) bright circles on black background. We prepared eight $I_{HR}$ of different edge thickness by convolving isotropic Gaussian. Variances of each isotropic Gaussian are arranged from 0.2 to 0.9 with step size 0.1. Each $I_{LR}$ is degraded and downscaled from each $I_{HR}$ by pre-defined random anisotropic blur kernel $k_1$. Then we extracted blur kernel $k_2$ with proposed GAN and got parameter $\sigma_\alpha$ that results best kernel similarity. In Fig. 3, parameter $\sigma_\alpha$, is correlated to edge thickness, and it means unintended component from edge could affect GAN and could be removed by parameter $\sigma_\alpha$.

**C. ARCHITECTURE**

In this section, we introduce the proposed E-KernelGAN method. The architecture of the proposed method is illustrated in Fig. 4. The objective is to find $k_1$ with given $I_{LR}$, $I_{HR}$ and $k_1$ is unknown high-resolution image and unknown blur kernel. $I_{LR}$ and $I_{LR2}$ are degraded and downscaled from $I_{HR}$ and $I_{LR}$ by blur kernel $k_1$ and $k_2$ with scale $\times s$ and $\times 2$ each, Eq. (13).

$$I_{LR} = (I_{HR} * k_1) \downarrow_2, \quad I_{LR2} = (I_{LR} * k_2) \downarrow_2$$ (13)

In proposed method, to find $k_1$, we utilize the relation between cross scale recurrence patches and blur kernel [1]. Cross-scale recurrence patches of $I_{LR}$ and $I_{LR2}$ were obtained implicitly by the GAN process [6], [10]. The GAN...
Y. Kim et al.: Blur Kernel Estimation and Correction for Blind Super-Resolution

VOLUME 4, 2022

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

FIGURE 4: Architecture of E-KernelGAN. Top is GAN architecture that extract $\hat{k}_2$ from given image $I_{LR}$. By ranking comparison between less real/more fake and real/fake, discriminator became sensitive to sharpness on patches. Bottom is kernel correction part that resize $\hat{k}_2$ close to $k_1$.

FIGURE 5: Architecture of generator and discriminator. Generator is composed of six convolutional layers and average pooling for downscaling. Discriminator adopts combination of parallel convolution paths.

FIGURE 6: Illustration of degradation and ranking comparison process. $I_{LR\_blur}$ is degraded image from $I_{LR}$ by $k_{real}$. Pairwise ranking loss compares the discrimination results so that $D(I_{LR})_{m,n}$ is larger than $D(I_{LR\_blur})_{m,n}$ in each pixel position $m,n$. $I_{LR\_blur}$ is degraded and downscaled image from $I_{LR}$ by $k_{fake}$. Likewise, discriminator learns so that $D(I_{LR2})_{m,n}$ is larger than $D(I_{LR2\_blur})_{m,n}$. We include $\hat{k}_2$ of general size to compare to $k_{real}$ and $k_{fake}$.

architecture can be divided into two parts, including an ordinary GAN and two blur generators, labeled 1 and 2, the second being used to distinguish “less real/more fake”. The purpose of an ordinary GAN is to equalize the sharpness domain of $I_{LR}$ and $I_{LR2}$ by optimizing the kernel $k_2$. The generator convolves patches of $I_{LR}$ with $k_2$ and downscales to half-scale. The generator is composed of six linear layers and average pooling, and the discriminator adopts 3x3, 5x5 and 7x7 convolution modules to achieve better discrimination capability than previous studies Fig. 5. To enhance the sharpness discrimination ability of the discriminator, we adopt a degradation process with blur generators and pairwise ranking loss, which will be explained in Section. III-D. In the bottom of Fig. 4, an estimated kernel $\hat{k}_2$ is corrected with the Eq. (8) derived by the Gaussian kernel approximation and a simple trick. The kernel correction method modulates the variance of $\hat{k}_2$ to be close to the variance of $k_1$ by resizing, and is explained in Section. III-A.

D. DEGRADATION AND RANKING COMPARISON PROCESS

As shown in Fig. 4, blur generators are used to degrade real and fake images. Blur generators consist of convolution by the prepared blurry kernels $k_{LR\_blur}$ and $k_{LR2\_blur}$ with downscaling, and there is no learning parameter. Blur generator1 degrades $I_{LR}$ by $k_{real}$, and blur generator2 degrades and downscales $I_{LR}$ by $k_{fake}$ in Fig. 6. With a relatively blurry image $I_{LR2\_blur}$, we compared it to $I_{LR2}$ to allow the...
discriminator to learn the rank between the images. This is performed for blur generator1 in the same way. To compare sharpness in the same position, the inputs of blur corruption and generator are provided with the same image. That is, the discriminator is trained so that $I_{LR2}$ has a larger discrimination output than the blurry image $I_{blur}$. The pairwise ranking loss, $L_r(I, I_b)$, is expressed as Eq. (14). Pairwise ranking loss only represents the difference in sharpness images and does not induce the output of discrimination to an absolute value such as real/fake, so it won’t interfere with GAN loss too much.

$$L_r(I, I_b) = \mathbb{E}_x \left[ \max \left( D(I_b) - D(I) + \xi_1, \frac{D(I_b)}{D(I)} - \xi_2, 0 \right) \right]$$

(14)

$$L_{\text{ranking}} = \frac{1}{2} \left( L_r(I_{LR}, I_{LRw}) + L_r(I_{LR2}, I_{LR2w}) \right)$$

(15)

The first term in $\max$ in Eq. (14) is a ranking loss with margin, and the second term prevents the discrimination output of $I_b$ from exceeding the discrimination output of $I$ at near zero. The rank loss improves the ability of the model to classify sharpness. It is natural that a discriminator with improved sensitivity to sharpness distinguishes sharpness in a clean edge region better than on regions of smooth or complex patterns (Fig. 9). The total loss $L_{\text{Total}}$ is expressed by Eq. (16), consists of GAN loss $L_{\text{GAN}}$, pairwise ranking loss $L_{\text{ranking}}$, reconstruction loss $L_{\text{Recon}}$, and constraints $C$. We adopted LSGAN-L1 as GAN loss as like KernelGAN [6], [16]. $L_{\text{Recon}}$ prevents failure in generator training by comparing $I_{LR2}$ to the downscaled image from $I_{LR}$ by the bi-cubic kernel. $\lambda_2$ gradually decreases during training from 5.0 and $\lambda_1$ is set to 0.5 in this work. Constraints are summed to 1 constraint $C_{\text{one}} = \left[ 1 - \sum_{i,j} k_{i,j} \right]$, non-negative constraint $C_{\text{non-neg}} = \sum_{i,j} \max(-k_{i,j}, 0)$, boundary constraint $C_b = \sum_{i,j} |k_{i,j} \cdot b_{i,j}|$ and unimodal constraint $C_u$. In unimodal constraint, $r$ is radius of ring, $\mathbb{R}_{out}(r)$ is outer pixels from ring and $\mathbb{R}_{in}(r)$ is pixels on ring.

$$L_{\text{Total}} = L_{\text{GAN}} + \lambda_1 L_{\text{ranking}} + \lambda_2 L_{\text{Recon}} + C$$

(16)

$$L_{\text{GAN}}(D) = \frac{1}{2} \mathbb{E}[||D(x) - 1| + |D(G(x))||]$$

$$L_{\text{GAN}}(G) = \frac{1}{2} \mathbb{E}[||D(G(x)) - 1||]$$

$$C_u = \sum_r \mathbb{E}_{i,j \sim \mathbb{R}_{out}(r)} \left[ \max \left( k_{i,j} - \sum_{m,n} \mathbb{E}_{i,j \sim \mathbb{R}_{in}(r)} [k_{m,n}, 0] \right) \right]$$

(17)

E. EFFECTIVENESS OF DEGRADATION AND RANKING COMPARISON

To investigate the effect of degradation and ranking comparison processes on discriminating ability, we performed an ablation study (Fig. 7). We trained the GAN with an image downscaled by a 40° Gaussian blur kernel and monitored the discrimination values for images downscaled by 0° to 160°. Because the GAN learned a 40° blurred image as a real image, if the GAN was trained well, the output of the discriminator may be higher on 40° than the others, 0° to 160°. Fig. 8 shows discrimination values (y-axis) at each angles (x-axis). The left graph is a result of the proposed GAN with degradation and ranking comparison, and the right graph shows the base GAN without degradation and ranking comparison. The 29th image of DIV2KRK [6] was used to perform training and evaluation. The y-axis represents the average discrimination output of the 26x26 sized 8000 patches. We illustrate the standard deviation of the discrimination output at 40° by red lines bounding above and below. The graphs show the effectiveness of the degradation and ranking comparison for the discriminator. Thus, the results show that the proposed method significantly enhanced the sharpness discrimination ability of the model. The similarity results are presented in Table. 1. Where the GAN focuses: In the above mentioned ablation study, we investigated where the GAN focused during the process. We illustrate the discrimination value differences between 40° and the others in Fig. 9. For ease of viewing, we showed the discrimination values from the highest to the 10,000th; we found that the difference values were higher on edges than in other regions. Higher discrimination differences would propagate higher gradients to the generator. From this simple observation, we were able to determine where the GAN focused during the experiment.

IV. EXPERIMENTS ON E-KERNELGAN

In our experiments, kernel similarities are measured with $\ell^2_{\text{conv}}$ and $\ell^2_{\text{co}}$, Eq. (18). For parameter $\sigma_{\alpha}^2$, we prepared

FIGURE 7: Illustration of ablation study for degradation and ranking comparison.

FIGURE 8: Discrimination outputs at each rotated kernel. Left graph is result of proposed GAN with degradation and ranking comparison. Right is result of base GAN without degradation and ranking comparison. x-axis is angle of anisotropic Gaussian blur kernel and y-axis is average of discrimination output.
validation set with DIV2K training dataset of 800 images. We degraded 800 images by independent random Gaussian kernels of uniform random variance $\mathcal{U}(0.6, 5.0)$ and estimated blur kernel with our method. After kernel correction, we got $\sigma_\alpha^2 \approx 0.1$ that is best in validation set. When the similarity is from $l_2^p$, we add $\sigma_\alpha^2 \approx 0.1$ next to model’s name at each Table. Because we used a correction technique, there may be some doubts about the fairness. Thus, we prepared another kernel correction method in Eq. (19). It could scale the variance of estimated kernel and remove isotropic kernel elements with parameter $\sigma_2^2$ similarly with Eq. (8). And there is not scale parameter $s$. We will get best value $\sigma_\beta^2$ and $a^2$ for parameter $\sigma_2^2$ and $a^2$ from evaluation dataset, that is, DIV2KRK and Flickr50. With $l_2^p$, we will measure how high the ability of a model is potentially. $l_2^p$ is potential ability of each model and $\sigma_2^2$, $a^2$ will be added next to the each value. When parameters are fixed to $(\sigma_\beta^2, a^2)$, $\gamma$ is strictly monotonic and invertable function of $\sigma_2^2$ as like $\gamma(s, \sigma_\alpha^2)$. $\gamma(l_2^p)$ does not represent the performance on evaluation dataset. It only implies the upper-bound of a model’s performance given a kernel correction method and dataset.

\[ l_2^{aw} = \frac{1}{N} \sum_n \sum_{i,j} |k_{GT,i,j,n} - \hat{k}_{i,j,n}|^2 \]
\[ l_2^{co} = \frac{1}{N} \sum_n \sum_{i,j} |k_{GT,i,j,n} - C(\hat{k}, \sigma_\beta^2, a^2)_{i,j,n}|^2 \]
\[ l_2^{po} = \frac{1}{N} \sum_n \sum_{i,j} |k_{GT,i,j,n} - C(\hat{k}, \sigma_\beta^2, a^2)_{i,j,n}|^2 \]
\[ \sigma_2^2 = \frac{\sigma_\beta^2}{a^2} - \sigma_\beta^2 \]
\[ \gamma(\sigma_2^2, a^2) = \sqrt{\frac{1}{a^2} - \frac{\sigma_2^2}{\sigma_\beta^2}} \]

| Table 2: Kernel similarity on DIV2KRK × 2 (less is better). |
|-----------------|-----------------|-----------------|
| Method          | $l_2^{aw}$ or $l_2^{co}$ | $l_2^{po}$ ($\sigma_2^2$, $a^2$) |
| SVDBSR [7]      | 0.0081           |                               |
| KOALAnet [3]    | 0.0044           |                               |
| KernelGAN [6]   | 0.0046           | 0.0042 (0.02, 1.14)          |
| DIP-FKP [10]    | 0.0050           | 0.0036 (0.00, 0.79)          |
| KernelGAN-FKP [10] | 0.0031       | 0.0029 (0.00, 1.20)          |
| E-KernelGAN, $\sigma_\alpha^2 = 0.1$ (ours) | 0.0022 | 0.0019 (0.00, 0.70) |

| Table 3: Kernel similarity on DIV2KRK × 4 (less is better). |
|-----------------|-----------------|-----------------|
| Method          | $l_2^{aw}$ or $l_2^{co}$ | $l_2^{po}$ ($\sigma_2^2$, $a^2$) |
| KernelGAN [6]   | 0.00079         | 0.00071 (0.00, 0.80) |
| DIP-FKP [10]    | 0.00552         | 0.00087 (0.00, 0.39) |
| KernelGAN-FKP [10] | 0.00062       | 0.00057 (0.00, 1.07) |
| E-KernelGAN, $\sigma_\alpha^2 = 0.1$ (ours) | 0.00041 | 0.00029 (0.34, 0.74) |

In implementation detail, we set the batch size and initial learning rate to 16 and $2 \times 10^{-4}$, and trained for 3000 iterations with ADAM optimizer. We implement our code with PyTorch library on NVIDIA TITAN X(pascal) GPU. To compare proposed method to previous works, we utilize the DIV2KRK (DIV2K random kernel) dataset for kernel similarity and blind-SR performance test. DIV2KRK was generated from DIV2K validation set with blur kernels [6]. For DIV2KRK, DIV2K validation set was degraded and downscaled by anisotropic Gaussian random kernel in scale factor $\times 2$ and $\times 4$. Random Gaussian kernels were generated from uniform random variance $\mathcal{U}(0.6, 5.0)$, random angle and random multiplicative noise. For scale factor $\times 4$, image was degraded two times by blur kernel. Kernel similarity in scale factor $\times 2$ is in Table 2 and shape comparison is in Fig. 10. SVDBSR and KOALAnet learning based blind-SR algorithms and they train kernel estimator concurrently with SR network. Results of SVDBSR is Gaussian and the others are non-Gaussian in that SVDBSR outputs variance and angle of Gaussian blur kernel in our experiments. SVDBSR estimates optimal kernel that minimizing artifact map, thus, if it failed to predict artifact map correctly, then it failed to predict blur kernel too as seen in Fig. 10. As will be explained in Section V, learning based kernel estimation methods have difficulty.
in estimating blur kernel when sharpness domains are different in evaluation and training stage. Similarly, KOALAnet [3] also suffered estimating kernels in DIV2KKR, even though it was very successful when the sharpness domain in evaluation and training stage were equal. KernelGAN [6] and KernelGAN-FKP [10] are same series as our method. They estimate blur kernels by recurrence patches comparison. DIP-KernelGAN-FKP [10] are same series as our method. They evaluation and training stage were equal. KernelGAN [6] and it was very successful when the sharpness domain in evaluation and training stage. Similarly, KOALAnet [3] also suffered estimating kernels in DIV2KRK, even though errors on these images exceed 0.02 that is about 10 times of KernelGAN series failed similarly for these images. The 2nd and 82nd images of DIV2KRK in Fig. 12. Especially KernelGAN series failed similarly for these images. The $l_2$ errors on these images exceed 0.02 that is about 10 times of the other images. In our knowledge, this may be caused by many similar patterns of different sharpness or edge thickness thus generator may imitate wrong real. In Section V, we tried to solve this problem with DIP.

V. E-KERNELGAN-DIP: COMBINATION MODEL WITH DIP

In this section, we proposed another kernel estimation method with E-KernelGAN and DIP with Fig. 13. In upper of Fig. 13, DIP part that consist of DIP network, $I_{SR,DIP}$, $\hat{k}$, and $I_{LR,DIP}$, has a general architecture to predict SR image and kernel at the same time [10], [13], [14]. DIP compares a predicted LR image and a given LR image on same pixel position by MSE loss. To generate sharp SR image is not easy without well defined prior, supervision or heavily trained SR model because sharp image has low-entropy than blurry image [14]. Thus, blur kernel information may be included in predicted SR image as well as in predicted kernel so that predicted kernel may be inaccurate. However, the mechanism that DIP predict blur kernel is quite different to KernelGAN series, thus we expected the differences may lead synergy. DIP compares in same pixel position, while GAN implicitly compares in arbitrary positions.

However, combining E-KernelGAN and DIP, there was an obstacle. Our proposed kernel correction method can’t propagate gradients so that two methods can’t cooperate. To solve this problem, we trained kernel correction network with synthesized kernel correction dataset generated by Eq. (5) as like bottom of Fig. 13. We trained each sub-network for each in/out synthesized kernel pairs, random anisotropic Gaussian kernel. The whole kernel correction network consists of 31 sub-networks according to $\sigma$, and weights were fixed for E-KernelGAN-DIP training stage. We trained E-KernelGAN-DIP for 2200 iterations from scratch. Weight parameter selecting sub-networks was trained after 600 iterations. And DIP network didn’t propagate gradients to E-KernelGAN for 500 iterations until DIP makes reasonable $I_{SR,DIP}$.

VI. EXPERIMENTS ON E-KERNELGAN-DIP

In this section, we investigated quantitative performance of E-KernelGAN-DIP by comparing to E-KernelGAN. In Table. 4 and 5, E-KernelGAN-DIP has two results for each model, ‘before KCnet’ and ‘after KCnet’. It means kernels before kernel correction network and after. As seen in table, E-KernelGAN-DIP outperformed E-KernelGAN on $l_2^{\text{co}}$ and $l_2^{\text{po}}$ at each scale.

### TABLE 4: Kernel similarity on DIV2KRK $\times 2$ (less is better).

| Method | $l_2^{\text{co}}$ | $l_2^{\text{po}}$ ($\sigma^{2+}$, $\sigma^{2-}$) |
|--------|------------------|-------------------------------------|
| E-KernelGAN, $\sigma^{2+} = 0.1$ (ours) | 0.002178 | 0.001931 (0.00, 0.70) |
| E-KernelGAN-DIP before KCNet, $\sigma^{2+} = 0.1$ (ours) | 0.001613 | 0.001566 (0.12, 0.73) |
| E-KernelGAN-DIP after KCNet (ours) | 0.001319 (0.52, 0.75) |

### TABLE 5: Kernel similarity on DIV2KRK $\times 4$ (less is better).

| Method | $l_2^{\text{co}}$ | $l_2^{\text{po}}$ ($\sigma_a^{2+}$, $\sigma_a^{2-}$) |
|--------|------------------|-------------------------------------|
| E-KernelGAN, $\sigma^{2+} = 0.1$ (ours) | 0.000405 | 0.000286 (0.34, 0.74) |
| E-KernelGAN-DIP before KCNet, $\sigma^{2+} = 0.1$ (ours) | 0.000310 | 0.000215 (0.58, 0.74) |
| E-KernelGAN-DIP after KCNet (ours) | 0.000191 (0.60, 0.74) |

To compare the quantitative and qualitative performance in blind-SR, we conducted several experiments by combining the estimated kernel of each algorithm with two super-resolution methods, ZSSR and SRMD [5], [8]. SRMD evaluate using pre-trained model which trained with external train data and kernel PCA code. ZSSR only use internal information of evaluation image for training data. In our experiments, ZSSR degrade $I_{LR}$ to $I_{LR2}$ or $I_{LR4}$ with provided kernel, thus it never see $I_{HR}$ during training. Blind-SR experiments are performed in scale $\times 2$ and $\times 4$ and the quantitative results are in Table. 7 as Y-channel PSNR and SSIM. We updated three results of previous works that showed better performance in our implementation and marked with *. With proposed method, the $\times 2$ SR performance has been greatly improved. As far as we know, our method achieve the best accuracy performance in the kernel estimation using internal-learning.

We presented qualitative performance comparison between kernel estimation methods in Fig. 11. Each method is combination of SR method and kernel estimation and quantitative results was obtained in the same experiments as quantitative results. We didn’t include DIP-FKP’s $\times 4$ results because it outputs too narrow kernel for $\times 4$. As several ablation studies and experiments, we showed our proposed methods are more effective in blur kernel prediction and supporting blind-SR.
FIGURE 10: Qualitative comparison to previous works. Evaluation dataset is DIV2KRK ×2 set. (a) is KernelGAN, (b) is SVDBSR, (c) is KernelGAN-FKP, (d) is DIP-FKP, (e) is E-KernelGAN and (f) is E-KernelGAN-DIP. Image numbers are 2, 3, 10, 40, 53, 64, 86 and 98 from left to right, top to bottom.

FIGURE 11: Qualitative comparison for images 10th and 16th in DIV2KRK. (a) is KernelGAN, (b) is DIP-FKP, (c) is KernelGAN-FKP, (d) is E-KernelGAN and (e) is E-KernelGAN-DIP. The first row of each image is the ×2 SR result and the second row is the ×4 result.

Kernel similarity on Flickr2K50 dataset: to validate performance on the other dataset and broader range of blur kernel, we prepared Flickr2K50 dataset. Flickr2K50 dataset consist of 50 images from 1st to 50th among 2650 images in flickr2K dataset. Each image is degraded by independent random anisotropic Gaussian kernel of variance \( U(0.35, 5.0) \). It has broader range than DIV2KRK and relatively sharper. The results are in Table. 6. As seen in table, ours are even better than previous works and preserved the performance well.

VII. CONCLUSION

In this paper, we proposed E-KernelGAN and E-KernelGAN-DIP. E-KernelGAN consists of a degradation and ranking comparison process as well as kernel correction. The degradation and ranking comparison process address the problems of the ordinary KernelGAN. KernelGAN estimates a blur kernel by arbitrary implicit comparison on recurrence

### Table 6: Kernel similarity on Flickr2K50 ×2 dataset (less is better).

| Method                        | \( l_2^{\text{raw}} \) or \( l_2^{\text{co}} \) | \( l_2^m (\sigma_2^*, \sigma_2^*) \) |
|-------------------------------|---------------------------------|---------------------------------|
| KernelGAN [6]                 | 0.008412                        | 0.006426 (0.02, 1.00)           |
| DIP-FKP [10]                  | 0.005746                        | 0.004819 (0.02, 1.00)           |
| KernelGAN-FKP [10]            | 0.004701                        | 0.004350 (0.00, 1.15)           |
| E-KernelGAN, \( \sigma_2^* = 0.1 \) (ours) | 0.003112                        | 0.002785 (0.00, 0.66)           |
| E-KernelGAN-DIP before KCNet, \( \sigma_2^* = 0.1 \) (ours) | 0.002644                        | 0.002442 (0.54, 0.62)           |
| E-KernelGAN-DIP after KCNet (ours) | 0.002111                        | 0.002111 (0.34, 0.79)           |
patches across scales. When a GAN compares real and fake patches, the abilities of the network to discriminate pattern shapes and sharpness are both indispensable. However, as explained above, knowing the slight changes in a pattern is a very difficult task for convolutional networks without further specialized design. The proposed method compares “real” to “less real” and “fake” to “more fake” to lead the discriminator to focus more on sharpness. With several ablation studies, we showed that the proposed GAN is much more sensitive to sharpness changes caused by blur kernels.

experimentally that the GAN focuses more on edges than the other regions and that it is affected by edge thickness. To develop a kernel correction method that fits the KernelGAN process, we formulate the relation between the variance of the PSF and the optimally predicted kernel in the GAN’s view with Gaussian kernel approximation. When the thickness parameter $\sigma_\alpha$ is fixed, the correction function is a strictly monotonic invertible function.

Although we achieved better kernel similarity results than previous studies, there were several unresolved failure cases. To address the failure cases, we propose E-KernelGAN-DIP based on a combination of E-KernelGAN and DIP. The different comparison processes of the two methods create synergy so that the remaining problem of failure cases is greatly improved. The results of kernel similarity and blind-SR tests demonstrated the excellent performance of the proposed method.

**REFERENCES**

[1] Tomer Michaeli and Michal Irani, “Nonparametric blind super-resolution,” in Proceedings of the IEEE International Conference on Computer Vision, 2013.

[2] Jinjin Gu and Hannan Lu and Wangmeng Zuo and Chao Dong, “Blind super-resolution with iterative kernel correction,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[3] Kim Soo Ye and Hyeonjun Sim and Munchurl Kim, “KOALAnet: Blind Super-Resolution using Kernel-Oriented Adaptive Local Adjustment,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[4] Soh Jae Woong and Sunwoo Cho and Nam Ik Cho, “Meta-transfer learning for zero-shot super-resolution,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[5] Assaf Shocher and Nadav Cohen and Michal Irani, “‘zero-shot’ super-resolution using deep internal learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[6] Sefi Bell-Kligler and Assaf Shocher and Michal Irani, “Blind super-

---

**TABLE 7: Quantitative Performance of blind-SR on DIV2KRK. (PSNR/SSIM)**

| Kernel | Method   | $\times 2$ PSNR/SSIM | $\times 4$ PSNR/SSIM |
|--------|----------|-----------------------|-----------------------|
| GT     | SRMD     | 32.42/0.9026          | 27.87/0.7638*         |
|        | ZSSR     | 32.44/0.8992          | 27.53/0.7446          |
|        | Bicubic Interpolation | 28.73/0.8040 | 25.33/0.6795 |
|        | ZSSR     | 29.10/0.8215          | 25.61/0.6911          |
| NBSR [1] | SRMD     | 25.51/0.8083          | 23.34/0.6530          |
|        | ZSSR     | 29.37/0.8370          | 26.08/0.7138          |
| KernelGAN [6] | SRMD     | 29.57/0.8564          | 26.31/0.7378*         |
|        | ZSSR     | 30.36/0.8669          | 26.81/0.7316          |
| FKP-DIP [10] | SRMD     | 30.35/0.8633          | 26.70/0.7403          |
|        | ZSSR     | 30.65/0.8654          | 26.71/0.7403          |
| FKP-KernelGAN [10] | SRMD     | 29.55/0.8675          | 26.32/0.7358          |
|        | ZSSR     | 30.67/0.8721          | 26.81/0.7328          |
| E-KernelGAN (ours) | SRMD     | 30.73/0.8816          | 27.36/0.7522          |
|        | ZSSR     | 31.33/0.8825          | 27.21/0.7391          |
| E-KernelGAN-DIP (ours) | SRMD     | 31.16/0.8856          | 27.45/0.7530          |
|        | ZSSR     | 31.62/0.8874          | 27.28/0.7404          |
Y. Kim et al.: Blur Kernel Estimation and Correction for Blind Super-Resolution