REBUTTAL: Blind Super-Resolution Kernel Estimation using an Internal-GAN

We would like to thank the reviewers for their comments. Below are our answers to the main questions/concerns.

R1: How can runtime be independent of image size? The number of training iterations is predetermined and independent of image size. At each iteration a fixed-size image crop is selected (see below), and all patches within that crop are analyzed. Since the same downscaling kernel applies to the entire image, it is unnecessary to analyze all image patches (although this is likely to happen due to large number of iterations and crop size). We will add a clarification.

R3: How are crops selected? Thanks, we forgot to mention this. Fixed sized image crops (64×64 to G, and accordingly 32×32 to D) are randomly selected with probability proportional to their mean gradient. We will add this.

R1+R2: Why 2D map and not a single scalar output? D provides a scalar score for every patch it analyzes, representing the probability of it belonging to the learned patch distribution. Rendering a 2D map (D-map) of all these scalars *at once* (for all the patches in the crop), in a single convolutional pass, is computationally more efficient, as proposed by [15,28] (one scalar per patch, kept in the D-map at the center pixel location of each patch). In response to R1's question, this is in fact a pixelwise output.

R2: Performance of KernelGAN on Bicubicly downscaled images: To verify R2's hypothesis, we ran KernelGAN for SR$\times$4 on 100 bicubicly downscaled images (DIV2K). We applied ZSSR, once with our *estimated* kernel, and once with the GT (Ground-Truth) bicubic kernel. The resulting PSNR/SSIM were 28.65dB/0.795 vs. 28.73dB/0.796, respectively. See Fig.A to view a sample of recovered kernels. We Will add this to the paper or Supp-Material. It is important to distinguish between the *Blind-SR* task and the easier *non-blind SR* task, where the GT kernel is known. External networks further train exhaustively on a large dataset of images, all downscaled with this *single* GT kernel.

R2: Why not train SR networks on multiple degradations? In fact, [36] reported unsuccessful experiments with this exact approach (see Chapter 3.5 of [36]: "Why not learn a blind model"). There are combinatorially many possible kernels, and each image has its own unique kernel. If a network is trained on a certain collection of kernels (e.g., random Gaussian kernels), it will be restricted to those types, and is unlikely to generalize to kernels which significantly deviate from those. In contrast, since each image has hundreds of thousands of patches, all sharing one kernel, the *unsupervised* KernelGAN (which trains on the *image-specific* patch distribution and kernel) can handle new never-before-seen kernels.

R2: Integration with ZSSR's coarse-to-fine implementation: We perform SR$\times$2 in a single ZSSR step, and SR$\times$4 in 2 coarse-to-fine ZSSR steps (by supplying ZSSR with both the $\times2$ and $\times4$ kernels). Note that for fair comparison, we used the same coarse-to-fine configuration when incorporating [23] into ZSSR. We will add a clarification to paper.

R3: Add ablation study of kernel constrains: To address R3's request, we empirically evaluated the effect of omitting each kernel constraint on 100 images (the DIV2KRK dataset). See table in Fig.C (will be added to paper/Supp-Material).

R3: Is there a non negative constraint on the kernel? No, the network does not impose any non-negative constraint on the kernel. Note that some kernels may include negative values (as noted by [23]) – e.g., the bicubic kernel.

R1: Statistical significance: We report average PSNR/SSIM on the dataset, which is the common practice in all SR papers/challenges (e.g., see NTIRE challenge report [30]). A global std value per-se will not suffice, as in some images the PSNR variation among different SR methods is inherently very low, while in others it is very high. To address R1's valid concern, we provide in Fig.B the *percent* of images (out of 100) where our method (green) outperformed other methods (a very large percent). We believe this measure is a more statistically informative in the context of SR.

R1+R3: Missing code, data and implementation details: Code and data will be made publicly available (Github) upon acceptance. Fig.D contains implementation details. We will expand on these in the paper or Supp-Material.

R1: Clarifications too late in text (1x1 convolutions, overparameterized G, etc.): We'll edit and clarify accordingly.

Fig. A: Bicbuic Downscaling

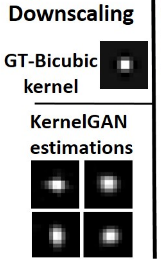

Fig. B: Statistical Significance (SRx4)

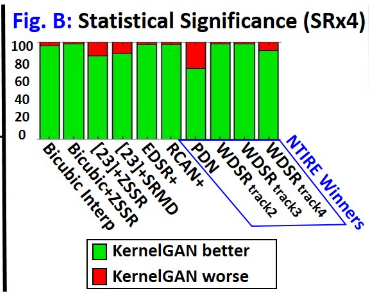

Fig. C: Constraints Ablation Study

Constraints a	SR X2	SR X4
No Sparseness	30.32/0.862	26.60/0.727
No Boundaries	30.37/0.863	26.66/0.729
No Sum to 1	29.82/0.851	26.58/0.724
No Centralization	30.43/0.865	26.78/0.731
w/o any constraints	29.62/0.850	26.32/0.721
All constraints (KernelGAN)	**<u>30.51/0.867</u>**	**<u>26.81/0.732</u>**

Fig. D: Implementation Details

Architecture - Fig.3,4 (in paper)
Crop size - Fig.3,4 (in paper)
Learning rate G = D = $2e^{-4}$
Learn rate update: $^{\times0.1}/_{750\ iters}$
# of iterations = 3,000
Iteration ratio: $^{G}/_{D} = {}^{1}/_{1}$
Adam Optimizer (β_1=0.5, β_2=0.999)
Kernel size: $s.f.\times 2 \rightarrow 13\times13$; $s.f.\times 4 \rightarrow 25\times25$
Constraints weights:
Sparseness=5 Boundaries=0.5
Sum to 1=0.5 Centralization=1

Fig. A. KernelGAN estimations for bicubicly downscaled images. Fig. B. Superiority percentage of our KernelGAN+ZSSR over each method mentioned in Table 1 of paper. Fig. C. Ablation study of the effect of different kernel constraints (PSNR(dB) / SSIM reported over 100 DIV2KRK images for SR$\times$2 and SR$\times$4). Fig. D. Implementation and hyper-parameters details.