Thanks for your valuable comments. Due to limited space, we can only respond to major concerns. And for other suggestions like typos, will be carefully revised in the revision.

I would like to respond to some common questions first.

Accuracy of predicted kernels. DAN does not predict kernels directly. The kernels are calculated in the reduced space transformed by PCA. Consequently, they cannot be intuitively visualized. Instead, we calculate the L1 error in the reduced space, and the results on Urban100 are shown in Figure 1. As one can see that the L1 error of reduced kernels predicted by DAN are much lower than that of IKC. It suggests that the overall improvements of DAN may partially come from more accurate retrieved kernels.

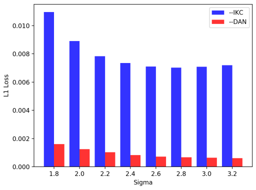

Figure 1: L1 error of reduced kernels.

Test with GT kernels. If GT kernels are provided, the iterating processing becomes meaningless. Thus we test the *Restorer* with just once forward propagation. The tested results for setting 1 is shown in Table 1. The result almost keeps unchanged and sometimes even gets worser when GT kernels are provided. It indicates that *Predictor* may have already satisfied the requirements of *Restorer*, and the superiority of DAN also partially comes from this good cooperation between its *Predictor* and *Restorer*.

Methods	Set5	Set14	B100	Urban100	Manga109
DAN	31.89	28.43	27.51	25.86	30.50
DAN(GT)	31.85	28.42	27.51	25.87	30.51

Table 1: PSNR results when GT kernel is provided.

Above discussions will be detailed reported in the revision.

To Reviewer 1:

Additional comparisons. We test three more methods on DIV2KRK for setting 2, *i.e.* NITRE'20 leading method, USRNet [33] and method of Cornillere *et al*. The results are shown in Table 2. As USRNet needs extra kernel-estimation, the predictions of KernelGAN are used. The failure of USRNet suggests that it is difficult for this two separate models to cooperate with each other. These results will be added in the revision. As for setting 1, the use of *Gaussian8* is following that of IKC.

Table 2: Additional comparisons on DIV2KRK.

Methods	x2		x4	
	PSNR	SSIM	PSNR	SSIM
Cornillere *et al.*	29.46	0.8474	-	-
Ji *et al.* (NITRE'20)	-	-	25.43	0.6907
KernalGAN+USRNet.	-	-	20.06	0.5359
DAN	32.56	0.8997	27.27	0.7462

Comparison with ZSSR. For setting 1, ZSSR is tested only with bicubic kernel, because it roughly takes 60 GPU days to test KernelGAN + ZSSR on the five datasets. We will remove this comparison in the revision.

Ablation. The basic network (SFTMD) in IKC is much larger than our *Restorer* in DAN (Table 3). We infer that the results will be improved if we substitute *Restorer* by SFTMD, but this experiment is limited by our computing devices.

To Reviewer 2:

Performance over different kernels. The performance over different kernels is shown in Figure 2. The results of IKC are also provided as a reference. The average PSNR decays when sigma increases. But compared with IKC, DAN behaves slightly more stable.

When it fails. Following the setting of previous methods, we do not consider white noises now. It may fail on noisy cases. But as the experiment on real world images suggests, if DAN is trained with noisy images, the model can also learn to denoise.

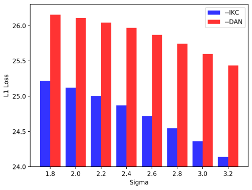

Figure 2: PSNR results under diferen kernels on Urban100.

To Reviewer 3:

Use of estimated kernels. The estimated kernels are not used to remove blur, but input to the *Restorer* to restore the HR image like IKC[13] dose. It is carefully illustrated in Sec 3.

Retrainning other network. In blind SR, different methods require very different training setting. We can only use the pretrained models provided by authors. Otherwise, the comparison may be unfair.

Model size. The model size comparison with IKC is shown in Table 3. DAN has less parameters and much fewer FLOPs (calculated when HR is of 720P) than IKC. Model size comparison with KernelGAN + ZSSR is meaningless, because its pipeline does not allow large model. In fact, blind SR methods have various pipelines, and the comparison on model size is unfair.

Methods	Params (M)	GFLOPs	Speed (s)
IKC	5.29	2178.72	3.93
DAN	4.33	926.72	0.75

Table 3: Model size comparison. FLOPs are calculated with HR being 720P

To Reviewer 4:

Denoise. Most nowadays blind-SR methods do not explicitly consider noisy cases, and it is true that denoising will damage the performance of following kernel-estimating. It is more preferred to integrate this two progress together.

Prior term. With some predefined assumptions, the prior term surely can be analytically expressed. But in more general cases, it is unknown. We will clarify this point more clearly in the revision.

About Eq. (4). Although our network is only supervised in the end, but the parameters are shared between different iterations. It should be correct to say that the two modules solve the two equations in Eq (4) alternately.