We thank reviewers for the insightful comments. Overall, all reviewers noted the novelty and convincing results of IGCN. Due to space limit, we only provide answers to main concerns. We shall fix minor issues and typos in the final version. We will release our code once this work is accepted.

R1: Comparison of same-scale aggregation and cross-scale aggregation. Table 2 in the submission shows that cross-scale aggregation (GraphAgg) performs better than fully-connected same-scale aggregation (Non-local block). In Table 1, we further report a baseline that finds and aggregates k neighbors within the same scale. Cross-scale aggregation still outperforms same-scale aggregation by a considerable margin. We believe these results are adequate to show the effectiveness of our GraphAgg, i.e., aggregation across scales indeed obtain useful HR information. We will revise the statement of "hardly improve" in L45-49 since same-scale aggregation also improves the baseline, despite being marginal compared to cross-scale aggregation.

Table 1: Comparison of GraphAgg with same-scale and cross-scale aggregation on Urban100 ($\times 2$).

	Baseline (EDSR)	same-scale	cross-scale
PSNR	32.93	33.01	**33.23**
SSIM	0.9351	0.9364	**0.9383**

R1: Difference from [34]. There are two main differences: 1) Different from [34, 23, 28, 41] that exploit and aggregate recurrent patches within LR input image, our method aggregates cross-scale internal HR cues and obtains an HR feature $F_{L\uparrow s}$ directly by GraphAgg. Table 2 in the manuscript and Table 1 demonstrate the effectiveness of cross-scale aggregation. 2) We introduce AdaPN that reduces the color discrepancy between query patch and k neighbor patches, keeping the high-frequency texture information unchanged. As shown in Table 6 in the manuscript and Figure 2 in the suppl., AdaPN allows more robust patch aggregation and benefits the subsequent image restoration.

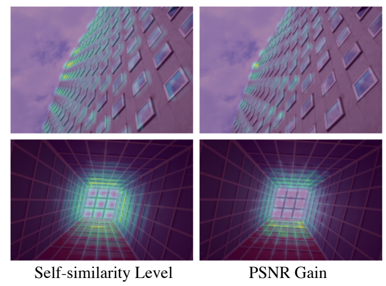

Figure 1: Examples to show the relationship between self-similarity level and PSNR gain (over EDSR). The brighter regions indicate larger values.

R1: The relationship between performance gain and self-similarity level. As shown in Figure 1, our method performs better in regions with self-similarity, especially in regions where texture patterns are extremely small. Besides, the performance can also be well maintained to that of EDSR in regions with few self-similar patches. More analysis will be provided in our final version.

R1+R4: Is IGCN dependent to the downsampling kernels? Does it work for blind SR? The patch matching for graph construction is performed in the VGG feature domain, which is relatively robust for different degradation kernels. Figure 2 shows an example of blind SR with an unknown blur kernel. IGCN recovers sharper result than ZSSR and EDSR. Our result is better because IGCN obtains and aggregates k image-specific HR exemplars, which form helpful internal complements when the blur kernel is unseen in the training dataset.

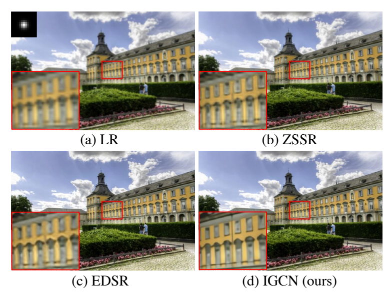

Figure 2: Results of blind SR ($\times 4$).

R2: How about perceptual quality? We compare our method with other SOTA methods in terms of LPIPS, a perceptual quality metric for images, (AlexNet version, Richard Zhang et al., CVPR'18). As shown in Table 2, IGCN achieves the best LPIPS scores for all scale factors. Besides, the visual results provided in the manuscript and suppl. also suggest the capability of IGCN in generating sharp and visually-pleasant images.

R2+R4: Running time. We provide a runtime comparison in Table 5 in the suppl. Benefit from the design of the searching window, IGCN runs about two times faster than the SOTA method SAN.

Table 2: Comparisons on Urban100 in terms of LPIPS. (Lower scores indicate better.)

	RDN	RNAN	OISR	SAN	EDSR	IGCN
$\times 2$	0.0552	0.0579	0.0531	0.0541	0.0553	**0.0520**
$\times 3$	0.1421	0.1440	0.1381	0.1392	0.1413	**0.1375**
$\times 4$	0.2055	0.2037	0.2027	0.2031	0.2039	**0.2006**

R3: What if similar patches are inexistent? Glasner *et al.* in [9] (their Figure 2) report that above 80% image patches exist 5 or more similar patches across different scales. Even if there are discrepancies between found neighbors and query, AdaPN can align neighbors to the query and reduce the low-frequency discrepancies. Moreover, ECN weights k HR patches adaptively for aggregation in accordance with difference between neighbor and query. ECN tends to output very small aggregation weights for dissimilar neighbors. As such, errors caused by dissimilar neighbors are well suppressed in our network.

R3: Do optimal values of d and k depend on the input resolution? Due to the design of the searching window, the input resolution will not affect the selection of optimal values. Regardless of input resolution, we search for k neighbors in a $d \times d$ window for aggregation. In addition, we select the optimal values of d and k on Urban100, which contains images with different resolutions. Thus, the selected d and k work well for different resolutions.

R4: Performance improvements are minor. Our IGCN shows performance gain of 0.2$\sim$0.3dB over baseline EDSR (which IGCN built upon) on large resolution benchmarks, i.e., Urban100 and Manga109. Although our performance does not exceed the SOTA method by a large margin, we believe the PSNR gain over baseline and ablation results (shown in Table 2 in the manuscript and Table 1) are adequate to show the effectiveness of our method. IGCN could perform better accordingly if a better base model is employed.

R4. Compare with winners of SR challenges AIM 2019 and NTIRE 2019. The comparisons will be unfair. All winners (i.e., ADCSR, IMDN, Efficient SR Network, ASSR) of AIM 2019 in different tracks adopt Flickr2K (2,650 images) as an additional training dataset to DIV2K (800 images). Differently, we follow the standard setting of main conference papers, using only DIV2K for training. The two SR challenges (Real SR and video SR) in NTIRE 2019 are different tasks with ours. Our comparisons already covered recent SOTA in CVPR'19 [5, 13, 21] and ICLR'19 [41].