Rebuttal of LAPAR (Paper ID: 653)

We highly appreciate all of your constructive comments as well as your recognition of this work's interesting perspective of rethinking and designing a lightweight and practical SISR method. We'll first answer some questions in common.

1. Comparison with EDSR, RCAN, ESRGAN and ProSR. As motivated in the paper, we aim to design a lightweight and real-time SISR method by exploring a novel idea of linear coefficient regression over a dictionary of filter bases. Thus, the models we compared are also small models (< 1 M params.). By contrast, EDSR, RCAN and ESRGAN and ProSR have params./PSNR as 43M/27.71dB, 16M/27.77dB, 17M/27.76dB and 16M/27.79dB under $\times 4$ setting on B100 dataset, which are **20 – 400**$\times$ larger than LAPAR-A (**0.66M/27.56dB**) with only 0.2dB gain. We'll add comparisons.

2. Results trained only on DIV2K. We followed exactly SRFBN, USRNet, DRN, UDVD to use both DIV2K and Flickr2K datasets. Here we also show the results of our LAPAR trained **only on DIV2K**. LAPAR-A *still* achieves SOTA performance among lightweight SISR methods. Red/blue: PSNR(dB) results trained on DIV2K/DIV2K+Flickr2K.

Method	Scale	Params	MultiAdds	Set5	Set14	B100	Urban100	Manga109
LAPAR-A	$\times 2$	0.548M	171G	37.95/38.01	33.58/33.62	32.17/32.19	32.01/32.10	38.41/38.67
	$\times 4$	0.659M	94G	32.10/32.15	28.53/28.61	27.56/27.61	26.01/26.14	30.22/30.42

3. There are not enough results for the task of image denoising and JPEG de-blocking. Like RAISR [5] and BLADE [39], we briefly studied the usefulness of LAPAR in image denoising and JEPG deblocking tasks, with the purpose to suggest the extensible applications of the proposed framework. Actually, it is *not* our intention to claim the SOTA performance of LAPAR. In view of the limited space, we will shrink this part to focus more on SISR discussions.

4. Hand-crafted filters v.s. learned filters. In our experiments, we found the learned filters may sometimes generate artifacts along edges due to overfitting. By contrast, the predefined meaningful Gaussian and DoG filters perform stably, also allowing for flexible control of the dictionary size and the consequent computational complexity in practice. Nevertheless, as also admitted in paper conclusions, it's indeed an interesting direction to study coupled optimization.

[*Reviewer* #1]
1. Whether the proposed approach also lends itself to GAN training. Thanks. LAPAR is a plug and play method and can lend itself easily to GAN training by adopting a discriminator. However, most GAN-based methods require generators with large capacity, which deviates from our current design objective. We'll explore this idea in the future.

2. Whether the predefined kernels inherently make it difficult to make more complex hallucination. As shown in L99-102, previous work demonstrated the strong representation ability of Gaussians and DoGs. Also, aided by the estimated combination coefficients, LAPAR can aggregate neighboring pixels to reconstruct arbitrary target values, hence allowing it to make complex hallucination in both fine-grained and flat areas. We'll elaborate more on this point.

3. Performance in the real world. As shown in Figure 7, our LAPAR performs well on the real-world images. Thanks, we will provide more visual examples with various processing beforehand in the supplementary material.

[*Reviewer* #2]
1. The $\times 3$ results are missing. We'll add the results of $\times 3$ setting. We saw similar conclusions as $\times 2/\times 4$ settings.

[*Reviewer* #3]
1. Linear v.s. non-linear combination in terms of performance and optimization speed. For simplicity and speed, we adopt the linear combination, which facilitates the optimization as verified in SMPL [31] and LSM [Tang *et al.* , CVPR 2020]. Thanks for this nice suggestion. Exploring a non-linear constraint will be an interesting future direction.

2. The predefined dictionary is over-complete or not. As stated in L98-99, our dictionary is redundant but not intended to be over-complete (a good future study topic), just like RAISR [5]. However, RAISR only designates fixed filters, we propose to linearly assemble filter bases to further improve the representation ability of dictionary. Actually, based on the linear combination weights regressed for every given pixel, LAPAR can reconstruct any target value for it.

3. Experiments on Set5 is limited in data size and generalization ability. We also obtained the same conclusions on B100, Urban100 and Manga109 datasets. Thanks for useful suggestions. We will add these results into the paper.

4. How does the cheap upsampling method (bicubic in the paper) influence the result? What is the limitation of upscaling factor? In our experiments, it has little impact on the recovery result by replacing the bicubic with bilinear. Under the $\times 8$ setting, in our offline experiments, LAPAR can still work well. Thank you, and we will add the results.

5. LAPAR-A is used for *all* visual comparisons. We'll add more comparisons with RAISR (2dB lower than LAPAR-A).

[*Reviewer* #4]
1. Input size and batch size settings. Thanks. Our settings just followed previous work e.g. CARN, DRN, UDVD, USRNet, SAN. Even after reducing the batch size from 32 to 16, LAPAR still yields approximate results (± 0.02dB).