Authors' Response to "A Blind Pixel-level Non-local Method for Real-world Image Denoising"

Thanks to the reviewers for their thoughtful feedback and enthusiasm for our paper. We regrettably found that most of R1's concerns can be explained by our paper itself.

R1: "row-wise matching is not well motivated". In this paper, we introduced the pixel-level NSS property to search similar pixels within similar patches, since "finding closely similar pixels is more feasible than similar patches in natural images". As in Lines 93-125, the "pixel-wise distance (PWD) is the distance apportioned to each pixel". Hence, it is not influenced by the number of pixels in a row or column. To make it more clear, in Figure 1, we search 64 patches for each 8×8 patch (now the number of pixels in each row and column are equal), the average PWDs of similar patches and pixels in noisy "House" are 0.0047 and 0.0036, respectively. The overall histogram still shifts leftwards clearly.

R1: "unspecified parameters and technique concerns". **Questions 2)-a, 2)-b, 2)-d**: The parameters such as q, λ, K are all specified in Sec. 4.1., i.e., "Implementation Details". The answer to **2)-b)** can be found at Lines 381-384: "According to the wavelet theory [65], the coefficients in the last two rows...should largely be noise." **Question 2)-c**: This is a trivial technique used in BM3D and several methods [5-15], and has been explained in Lines 430-431, 439. **Question 2)-d**: the idea of "adding a fraction of the noisy image back" is a trivial "iterative regularization" technique introduced in "S. Osher *et al*. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 2005". We will cite this paper for clearer explaination. This technique is commonly used in denoising methods [5-15]. As explained in Lines 401-404, using more iterations is to "obtain more accurate estimation of local signal intensities". **Question 2)-e**: Most of existing works have their backbones, e.g., ResNet, without which many computer vision methods would suffer lower accuracies. In this work, our backbone, or "tricks" by R1, is closely similar to the famous BM3D method. The ablation studies in Table 6 of the main paper already "imply the effectiveness of the proposed pixel-level NSS prior" over the backbone.

R1: "The noise in raw camera images is dominantly Poisson, and treating it as Gaussian is unsound". We agree wholeheartedly that the noise in **raw** images is dominantly Poisson. However, it is the **RGB** (not **raw**) image we processed in the paper. The noise in **RGB** images can be safely modeled as Gaussian (please refer to [36,41,61]).

R1: "σ_l is not robust due to a few very different pixels". The robustness of σ_l can be double guaranteed: 1) using the m most similar (instead of all) patches, 2) using the q most similar (instead of all) pixel groups. After the two filtering steps, few pixels very different from the rest will be used for noise level estimation. Even if a few pixels will be very diverse, they have little influence of the overall robustness of σ_l since the majority groups of pixels are very similar.

R1: "the paper raises more questions than it answers, though idea is interesting". Many technical details can be explained by the paper itself. We are very appreciated if R1 could pay more attention to its insights and better performance over the other methods on real-world denoising.

R2: "perform experiments on BSD68". We perform more experiments on the **BSD68** dataset. The results are listed in Tables 1 and 2. One can see that NLH achieves comparable performance with other methods on both experiments.

Noise std σ	5	15	25	35	50	75	100
Zoran *et al.*	4.74	14.42	-	-	49.23	74.33	-
Liu *et al.*	**5.23**	**15.18**	**25.13**	**34.83**	49.54	74.36	98.95
Chen *et al.*	8.66	16.78	26.26	36.00	50.82	75.75	101.62
Ours	5.91	15.88	25.64	35.50	**50.45**	**75.40**	**100.97**

Table 1: Estimated noise levels of different methods on the **BSD68** dataset corrupted by AWGN noise with std σ. "-" indicates that the results cannot be obtained due to the internal errors of the code.

σ	**BM3D**	**EPLL**	**WNNM**	**TNRD**	**DnCNN**	**NLH** (Blind)
15	31.07	31.21	31.37	31.42	31.73	31.24
25	28.57	28.68	28.83	28.92	29.23	28.82
50	25.62	25.67	25.87	25.97	26.23	26.05

Table 2: Average PSNR(dB) results of different methods on **BSD68**.

R2: "visualization comparison". We had compared the denoised images by different methods in Figures 3-4 in the main paper and Figures 3-12 in the supplementary file.

R2: "Haar or DCT transforms?". As described in Lines 362-364, we chose Haar transform "due to its flexible operation, faster speed, and lightweight memory". The reasons we do not employ DCT transform are: 1) it is slower than Haar; 2) it requires the pixel matrix to be square, which largely hinders the flexibility. On **CC** dataset, by replacing Haar transform with DCT, NLH achieves nearly the same PSNR/SSIM results (38.48dB/0.9647) as its original Haar version (38.49dB/0.9647). **Hence, we employ Haar transform for speed and flexibility considerations**. Motivated by the design philosophy of famous BM3D, we combining Haar transform and Wiener filtering in order to take advantages of hard thresholding based signal estimation (Haar) and soft thresholding based noise removal (Wiener).

R2: "hyper-parameters for Gaussian noise removal?" The influence of parameters on Gaussian noise removal are very similar with real-world denoising. NLH can perform well once these parameters are set according to the previous patch level NSS based methods [5-15], except the novel parameter q: number of similar pixels. All these parameters are slightly tuned for better PSNR/SSIM performance.

R2: "reproducible codes". We will definitely released the codes publicly, and add the link to the paper if accepted.

R3: "Results on PolyU dataset". Thanks for the suggestion. We performed experiments on PolyU dataset. The results are listed in Table 3. It can be seen that the proposed NLH still outperforms other methods on PSNR and SSIM.

Metric	**CBM3D**	**NI**	**NC**	**TWSC**	**DnCNN+**	**NLH**
PSNR↑	37.40	37.92	37.77	38.60	37.82	**38.84**
SSIM↑	0.9526	0.9549	0.9570	0.9685	0.9583	**0.9705**

Table 3: Average PSNR(dB) and SSIM results of different methods on the 100 cropped real-world noisy images in the **PolyU** dataset.