LAN: Learning to Adapt Noise for Image Denoising

Supplementary Material

Model	Dataset	Self-loss	PSNR$^{\uparrow}$ (dB) Full-train	LAN (Ours)
Restormer	PolyU	ZS-N2N	39.06	**39.20**
		Nbr2Nbr	38.77	**39.11**
	Nam	ZS-N2N	38.40	**38.84**
		Nbr2Nbr	37.83	**38.49**
Uformer	PolyU	ZS-N2N	38.38	**39.05**
		Nbr2Nbr	38.48	**39.09**
	Nam	ZS-N2N	37.80	**38.37**
		Nbr2Nbr	37.23	**38.19**

Table S1. PSNR comparison for Restormer and Uformer on the PolyU and Nam dataset with 40 iterations.

Model	Dataset	Self-loss	SSIM$^{\uparrow}$ Full-train	LAN (Ours)
Restormer	PolyU	ZS-N2N	0.963	**0.968**
		Nbr2Nbr	0.960	**0.969**
	Nam	ZS-N2N	0.950	**0.965**
		Nbr2Nbr	0.942	**0.962**
Uformer	PolyU	ZS-N2N	0.953	**0.966**
		Nbr2Nbr	0.957	**0.968**
	Nam	ZS-N2N	0.941	**0.965**
		Nbr2Nbr	0.931	**0.960**

Table S2. SSIM comparison for Restormer and Uformer on the PolyU and Nam dataset with 40 iterations.

S1. Additional quantitative results

To evaluate the effectiveness of our proposed framework LAN, we compare it to alternative adaptation methods. To demonstrate the applicability of our framework, we also conducted experiments with different network backbones, namely DnCNN [48], Restormer [47], Uformer [41], with different self-supervised losses such as ZS-N2N [34] and Nbr2Nbr [17]. In particular for Restormer and Uformer, in order to verify the performance differences when the main compared method, Full-trainable, and our LAN reach their respective peaks, we performed a performance comparison at 40 iterations for both models instead of the original 20 iterations as Figure S1 and Figure S2. Comparing the peak PSNR and SSIM performance in each experimental case as shown in Table S1 and Table S2, our method shows higher performance.

S2. Additional qualitative results

We show additional qualitative results of the pre-train networks, the 'full-trainable' adaptation, and our LAN framework in Figure S3. These images were obtained with Restormer fine-tuned via N2N with 20 iterations on the Nam dataset and fine-tuned via ZS-N2N with 20 iterations on the PolyU dataset. Similar to Uformer's results presented in the previous qualitative results, our method significantly removes unseen noise compared to the pre-trained network and 'full-trainable' adaptation.

S3. Efficiency differences as input size changes

As we discussed in Section 4.2, our proposed method, LAN, assumes a 256 x 256 image size, and we verified its time and space efficiency using Restormer and Uformer. In this section, we discuss the changes in time and space efficiency as the input image size changes. In total, we conducted experiments for image sizes ranging from 64 to 512. The comparison between 64 and 256 image sizes is performed using the Restormer. This is because the minimum image size for the Uformer model with the subsampling-based self-loss method is 256. The comparison between 256 and 512 image sizes is performed using the Uformer. This is because Restormer had a problem with exceeding memory when used with ZS-N2N at 512 image size.

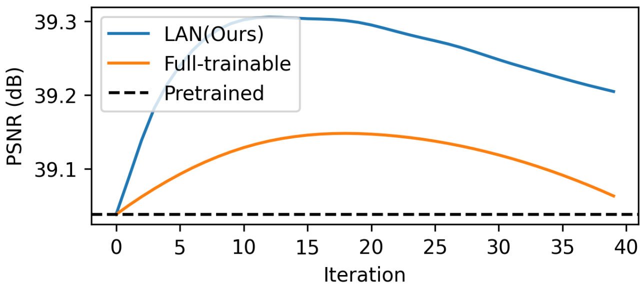
S4. LAN-black-box (LAN-BB)

Our LAN framework also works for scenarios where a denoising network is not directly available to us, where we only have access to the prediction by a given denoising network. Thus, we take a denoising network as a black box when performing noise adaptation. We call this version LAN-black-box (LAN-BB). In this case, we perform the adaptation using only the prediction results, without using the gradient from the denoising network. To achieve this, our implementation uses the stop-gradient operation. Thus, noise adaptation is performed as follows:

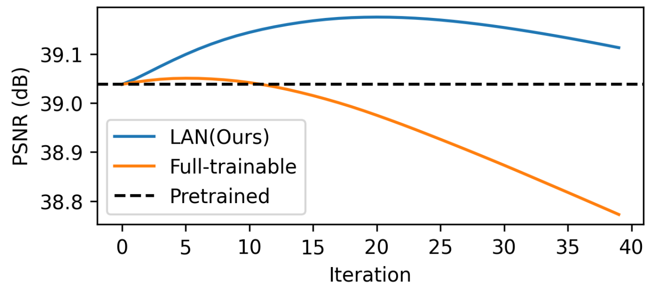
$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \left\| \mathrm{sg}[f_{\boldsymbol{\theta}^*}(D_1(\boldsymbol{y}^u + \boldsymbol{\phi}))] - D_2(\boldsymbol{y}^u + \boldsymbol{\phi}) \right\|_2^2, \tag{S1}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient operation.

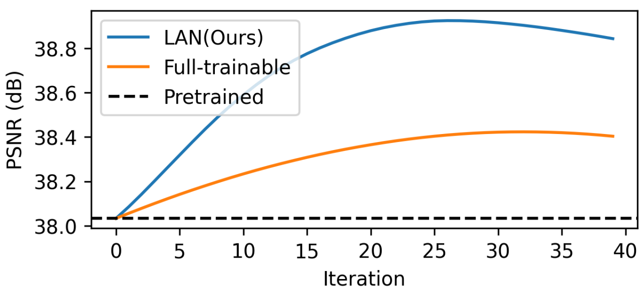
This approach allows us to effectively adapt to unseen noise without backpropagation through a denoising network. Removing the backpropagation process for the denoising network results in the significant improvement in efficiency, as shown in Table S5. Meanwhile, in contrast to expectation that the denoising performance may be degraded, Table S3 shows that there is not much performance degradation. One possible explanation would be that the network prediction (an estimated clean image) reveals the misalignment between the unseen noise from input image and noise a network expects. Furthermore, the training of our learnable offset $\boldsymbol{\phi}$ can be still achieved with gradients computed from the term $D_2(\boldsymbol{y}^u + \boldsymbol{\phi})$ in the self-supervised loss function (Equation S1).



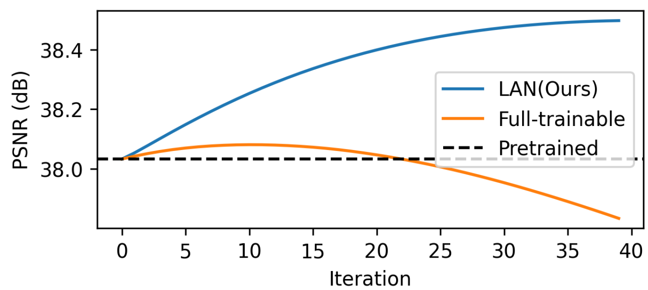
PSNR with Restormer and ZS-N2N on PolyU



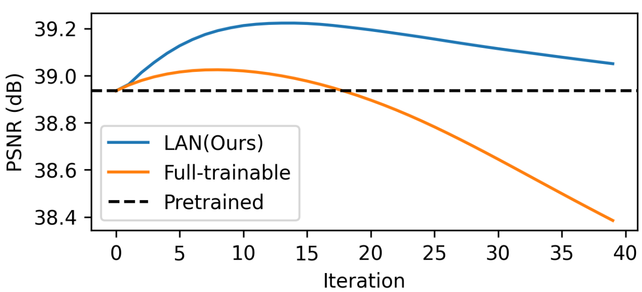
PSNR with Restormer and Nbr2Nbr on PolyU



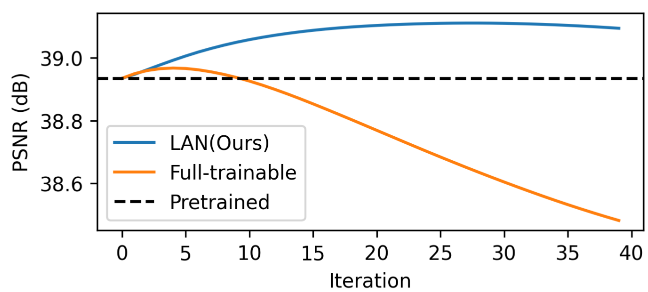
PSNR with Restormer and ZS-N2N on Nam



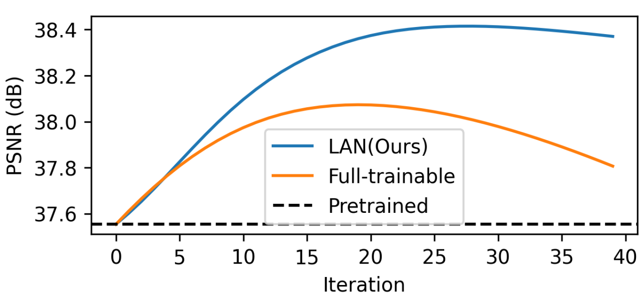
PSNR with Restormer and Nbr2Nbr on Nam



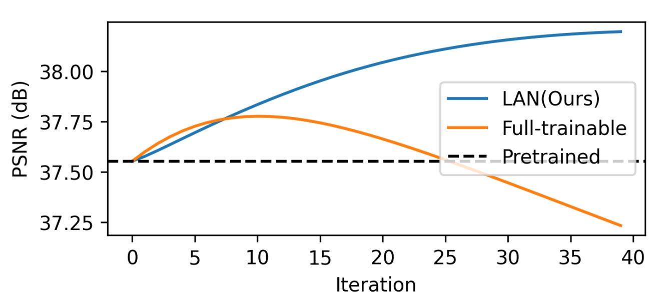
PSNR with Uformer and ZS-N2N on PolyU



PSNR with Uformer and Nbr2Nbr on PolyU

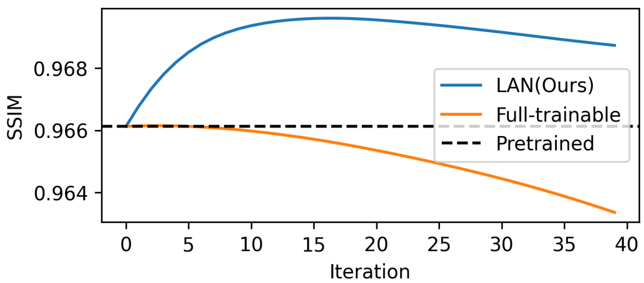


PSNR with Uformer and ZS-N2N on Nam

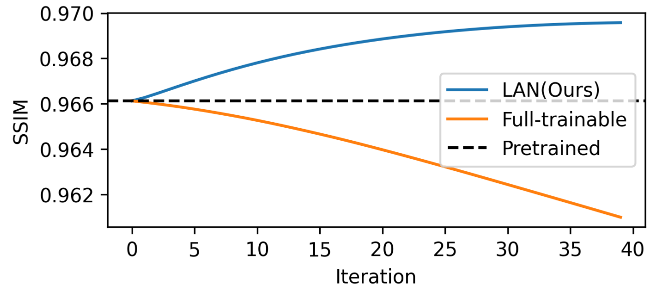


PSNR with Uformer and Nbr2Nbr on Nam

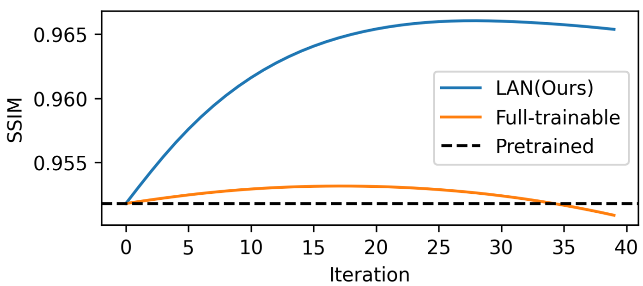
Figure S1. PSNR comparison for Restormer and Uformer on the PolyU and Nam dataset. Iteration increased from 20 to 40



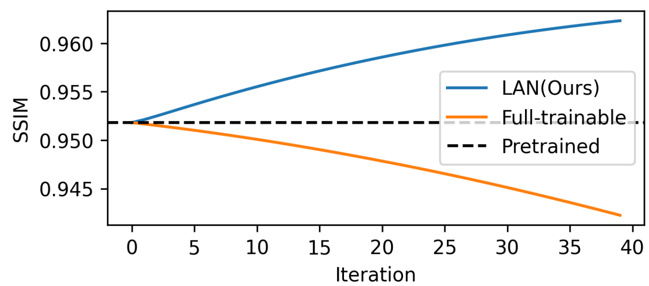
SSIM with Restormer and ZS-N2N on PolyU



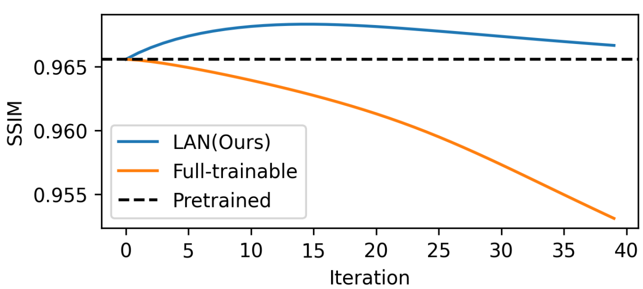
SSIM with Restormer and Nbr2Nbr on PolyU



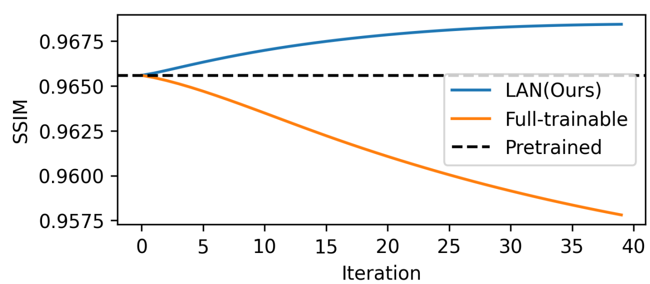
SSIM with Restormer and ZS-N2N on Nam



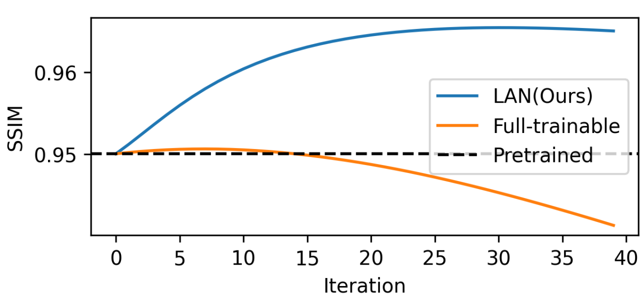
SSIM with Restormer and Nbr2Nbr on Nam



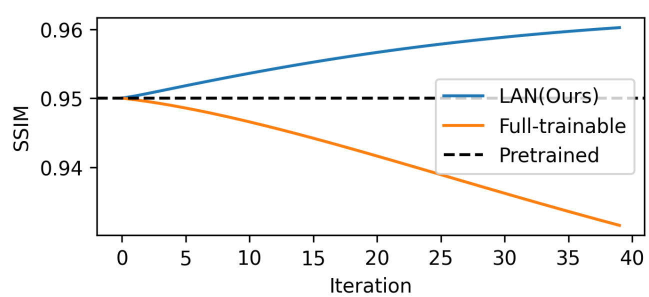
SSIM with Uformer and ZS-N2N on PolyU



SSIM with Uformer and Nbr2Nbr on PolyU



SSIM with Uformer and ZS-N2N on Nam



SSIM with Uformer and Nbr2Nbr on Nam

Figure S2. SSIM comparison for Restormer and Uformer on the PolyU and Nam dataset. Iteration increased from 20 to 40

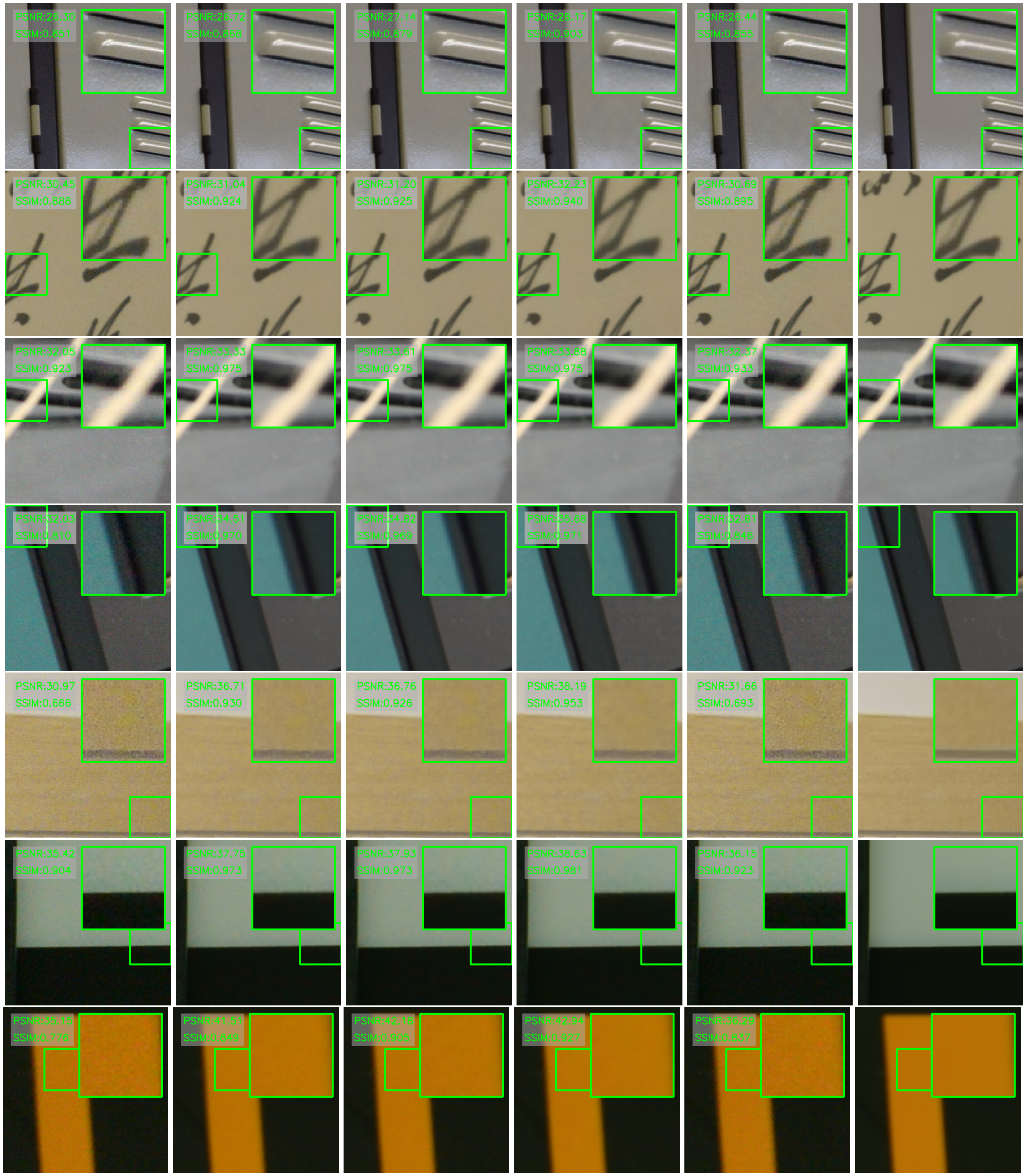


Figure S3. Qualitative comparisons among different adaptation methods. Images are obtained with SIDD-pretrained Restormer. Full-trainable and LAN (Ours) finetuned the pretrained network for 20 iterations on PolyU via ZS-N2N (first five rows) and on Nam via Nbr2Nbr (last two rows).

Model	Method	Iter.	SIDD → PolyU PSNR↑ (dB) / SSIM↑ ZS-N2N	SIDD → PolyU PSNR↑ (dB) / SSIM↑ Nbr2Nbr	SIDD → Nam PSNR↑ (dB) / SSIM↑ ZS-N2N	SIDD → Nam PSNR↑ (dB) / SSIM↑ Nbr2Nbr
DnCNN	pretrained	-	38.10 / 0.952		36.60 / 0.930	
	full-trainable	5	38.07 / 0.951	38.08 / 0.951	36.60 / 0.929	36.60 / 0.929
		10	38.04 / 0.950	38.06 / 0.951	36.59 / 0.928	36.60 / 0.928
		20	37.99 / 0.949	38.02 / 0.949	36.56 / 0.925	36.56 / 0.925
	LAN	5	38.22 / 0.954	38.16 / 0.953	36.73 / 0.934	36.66 / 0.932
		10	**38.29** / **0.955**	38.22 / 0.954	**36.79** / 0.936	36.71 / 0.933
		20	**38.29** / **0.955**	38.31 / **0.956**	36.78 / **0.938**	**36.80** / 0.935
	LAN-BB	5	38.23 / 0.954	38.17 / 0.953	36.70 / 0.934	36.66 / 0.932
		10	38.27 / **0.955**	38.25 / 0.955	36.73 / 0.936	36.71 / 0.934
		20	38.14 / 0.953	**38.34** / **0.956**	36.59 / 0.937	36.79 / **0.937**
Restormer	pretrained	-	39.03 / 0.966		38.03 / 0.951	
	full-trainable	5	39.09 / 0.966	39.04 / 0.965	38.14 / 0.952	38.07 / 0.951
		10	39.12 / 0.965	39.04 / 0.965	38.23 / 0.952	38.08 / 0.950
		20	39.14 / 0.965	38.98 / 0.964	38.35 / 0.953	38.05 / 0.948
	LAN	5	39.23 / 0.968	39.09 / 0.967	38.31 / 0.957	38.14 / 0.953
		10	39.30 / **0.969**	39.14 / 0.967	38.58 / 0.961	38.25 / 0.955
		20	39.28 / **0.969**	39.17 / 0.968	**38.86** / 0.965	38.38 / 0.958
	LAN-BB	5	39.24 / 0.968	39.13 / 0.967	38.36 / 0.957	38.16 / 0.954
		10	**39.33** / **0.969**	39.22 / 0.968	38.55 / 0.961	38.30 / 0.956
		20	39.19 / 0.968	**39.33** / **0.969**	38.53 / **0.966**	**38.50** / **0.961**
Uformer	pretrained	-	38.93 / 0.965		37.55 / 0.950	
	full-trainable	5	39.01 / 0.964	38.96 / 0.964	37.80 / 0.950	37.72 / 0.948
		10	39.01 / 0.963	38.92 / 0.963	37.97 / 0.950	37.77 / 0.946
		20	38.91 / 0.961	38.77 / 0.961	38.07 / 0.948	37.67 / 0.942
	LAN	5	39.12 / 0.967	39.00 / 0.966	37.82 / 0.955	37.69 / 0.951
		10	**39.21** / **0.968**	39.05 / 0.966	38.09 / 0.960	37.83 / 0.953
		20	39.20 / **0.968**	39.10 / 0.967	**38.36** / **0.964**	38.02 / 0.956
	LAN-BB	5	39.13 / 0.967	39.01 / 0.966	37.89 / 0.955	37.68 / 0.952
		10	39.20 / **0.968**	39.09 / 0.967	38.13 / 0.959	37.83 / 0.954
		20	38.95 / 0.966	**39.16** / **0.968**	38.21 / **0.964**	**38.07** / **0.958**

Table S3. **Quantitative comparison of denoising performance** for each combination of denoising network backbone, adaptation method of full-train, LAN and LAN-BB, and self-supervised loss on real-world noise datasets (PolyU and Nam) after pre-training on another real noise dataset, SIDD.

Model	Input size	Self-loss	LAN / Full-trainable Time	LAN / Full-trainable Memory
Restormer	64	ZS-N2N	87.17 %	87.82 %
		Nbr2Nbr	88.79 %	79.06 %
	128	ZS-N2N	87.69 %	91.85 %
		Nbr2Nbr	87.86 %	86.98 %
	256	ZS-N2N	79.88 %	93.27 %
		Nbr2Nbr	93.04 %	92.22 %
Uformer	256	ZS-N2N	74.10 %	73.75 %
		Nbr2Nbr	85.24 %	74.21 %
	512	ZS-N2N	74.72 %	75.06 %
		Nbr2Nbr	90.98 %	80.17 %

Table S4. The runtime and memory efficiency ratio of LAN (ours) to full-trainable on various input image size.

Model	Self-loss	LAN-BB/Full-trainable Time	LAN-BB/Full-trainable Memory
Restormer	ZS-N2N	26.77 %	3.81 %
	Nbr2Nbr	75.34 %	18.75 %
Uformer	ZS-N2N	42.51 %	14.00 %
	Nbr2Nbr	74.29 %	49.80 %

Model	Self-loss	LAN-BB/LAN Time	LAN-BB/LAN Memory
Restormer	ZS-N2N	33.51 %	4.09 %
	Nbr2Nbr	75.86 %	20.36 %
Uformer	ZS-N2N	57.37 %	18.98 %
	Nbr2Nbr	78.50 %	67.44 %

Table S5. The runtime and memory efficiency ratio of LAN-BB to full-trainable. And the runtime and memory efficiency ratio of LAN-BB to LAN. Each comparison is based on an image size of 256x256.