Supplementary Material *for*

AP-BSN: Self-Supervised Denoising for Real-World Images via Asymmetric PD and Blind-Spot Network

Wooseok Lee[1] Sanghyun Son[1] Kyoung Mu Lee[1,2]
[1]Dept. of ECE & ASRI, [2]IPAI, Seoul National University
adntjr4@gmail.com, {thstkdgus35, kyoungmu}@snu.ac.kr

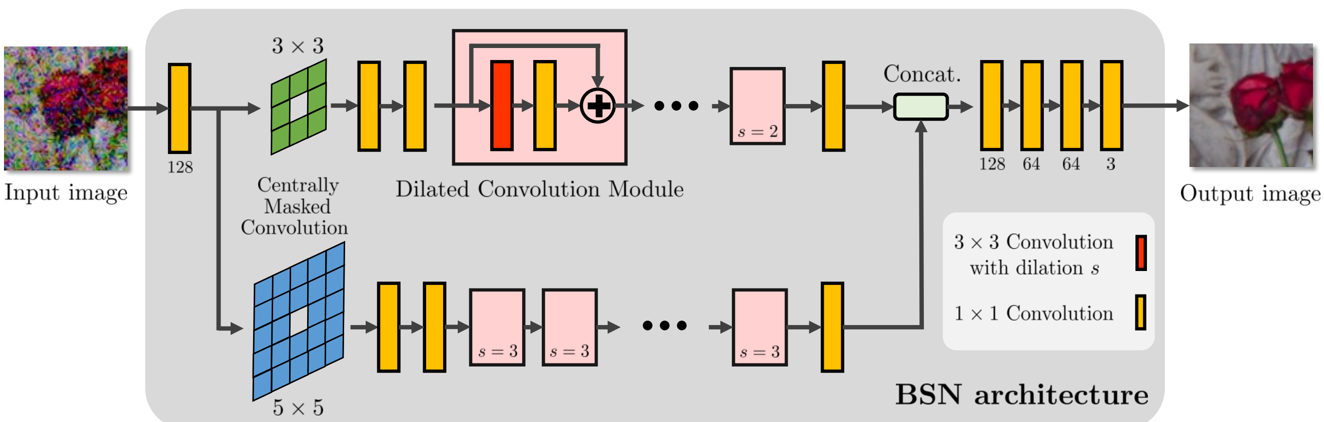


Figure S1. **Visualization of our BSN architecture.** We adopt 3×3 and 5×5 Centrally Masked Convolutions [7] to implement the blind-spot network. Each Dilated Convolution module (DC) contains one 3×3 dilated convolution with a stride s, where $s = 2$ and $s = 3$ are used for the upper and lower path of the network, respectively. For each path, we stack 9 DC modules. The number of output channels is denoted below each convolutional layer, where 128 is used by default.

S1. Optimization

To train our AP-BSN, we randomly crop 120×120 noisy patches from the SIDD and DND datasets, respectively. We note that 24,542, 24,784, and 24,320 patches are used in one epoch for SIDD-Medium, DND, and SIDD benchmark datasets, respectively. Each sample is augmented with random 90° rotation and horizontal/vertical flips. Our mini-batch contains the 8 augmented samples. The proposed AP-BSN is optimized for 20 epochs, where the learning rate is decayed by a factor of 10 for every 8 epochs.

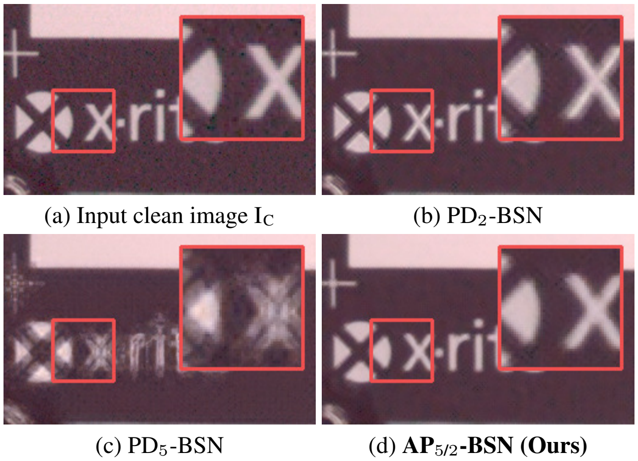
S2. Network architecture

Our BSN architecture is based on Wu *et al.* [7], while several changes are made for simplification. Instead of the MDC modules with multiple branches of the dilated convolutions, we use a sequence of dilated convolution modules (DC) that have a single branch only. Fig. S1 visualizes a detailed architecture of the BSN used to construct our AP-BSN framework.

Therefore, our network has 3.7M parameters, which are fewer than 6.6M parameters from the original BSN proposed by Wu *et al.* [7]. We also note that recent unsupervised/unpaired methods adopt larger denoising networks than the proposed AP-BSN. Specifically, *e.g.* DIDN [9] and C2N [3], MWCNN [4] has ∼16.2M in Wu *et al.* [7]). Our AP-BSN w/o R^3 shows comparable results with much smaller denoising networks even our AP-BSN only uses noisy images.

S3. Effects of aliasing artifacts

To examine the effect of aliasing artifact during the training and inference, we train our AP-BSN using clean SIDD images *only*. Specifically, BSN is trained to reconstruct the same image from given a clean input while not seeing the center pixel in the receptive field. We suppose that the clean images contain zero-intensity noise, which follows the two basic assumptions of BSN: noise signals are spatially uncorrelated and zero-mean. Thus, PD-BSN should learn an



(a) Input clean image I_C (b) PD_2-BSN (c) PD_5-BSN (d) **$AP_{5/2}$-BSN (Ours)**

Figure S2. **Effects of aliasing artifacts in BSN.** To validate that the advantage of AP-BSN comes from the existence of aliasing artifacts, we conduct a clean-to-clean experiment. We sample a clean image from the SIDD validation dataset for visualization.

identity mapping if sub-images from PD do not contain any noise. However, as shown in Figs. S2b and S2c, PD_5-BSN removes high-frequency information from the given input clean image in Fig. S2a and does not operate an identity function even on the clean image while PD_2-BSN does not. From this observation, we can assume that PD_5-BSN learns to remove some information during the training that does not exist in PD_2 sub-images. When we apply the proposed $AP_{5/2}$ strategy, BSN does not remove high-frequency components and preserves the image structure well, as shown in Fig. S2d. Therefore, we conclude that the aliasing artifacts prevent PD_5-BSN from being a feasible denoising model since removing the artifacts during inference can significantly degrade the performance of PD-BSN.

S4. AP-BSN on the NIND dataset

In Fig. 2a of our main manuscript, we have demonstrated that noise signals in the NIND [2] dataset show gradually decreasing correlations between them as their relative distance d increases. Such observation implies that the proposed AP-BSN may perform better with $a = 6$ or larger, as the spatial correlations between noise can be further reduced. Therefore, we analyze the trade-offs of $AP_{a/b}$ on the NIND dataset similar to Section 5.2 in our main manuscript. To investigate the trade-off under diverse scenes, we conduct a per-sample analysis rather than calculating the performance on the entire dataset. Fig. S3a shows several noisy images in the NIND dataset. In Fig. S3b, we also visualize the denoising results of our $AP_{5/2}$-BSN + R^3 trained on the NIND dataset. Since the noise property of the NIND dataset differs from SIDD, $AP_{6/2}$ may perform slightly better on some specific samples as shown in Fig. S3c. However, we note that the performance gaps are marginal, and $AP_{5/2}$ generalizes well on various real-world datasets on average.

S5. Qualitative results

S5.1. Additional qualitative results

Since several existing methods do not provide qualitative results on specific datasets, we could not perform extensive qualitative comparisons in our main manuscript. For example, Figs. 10d (upper figure in the 3rd column) and 10e (lower figure in the 3rd column) in our main manuscript represent results of NAC [8] on the DND benchmark and R2R [5] on the SIDD benchmark, respectively, because R2R does not provide results on the DND dataset. Fig. S4 shows additional qualitative comparison between different denoising methods on the DND [6] benchmark and SIDD [1] validation dataset.

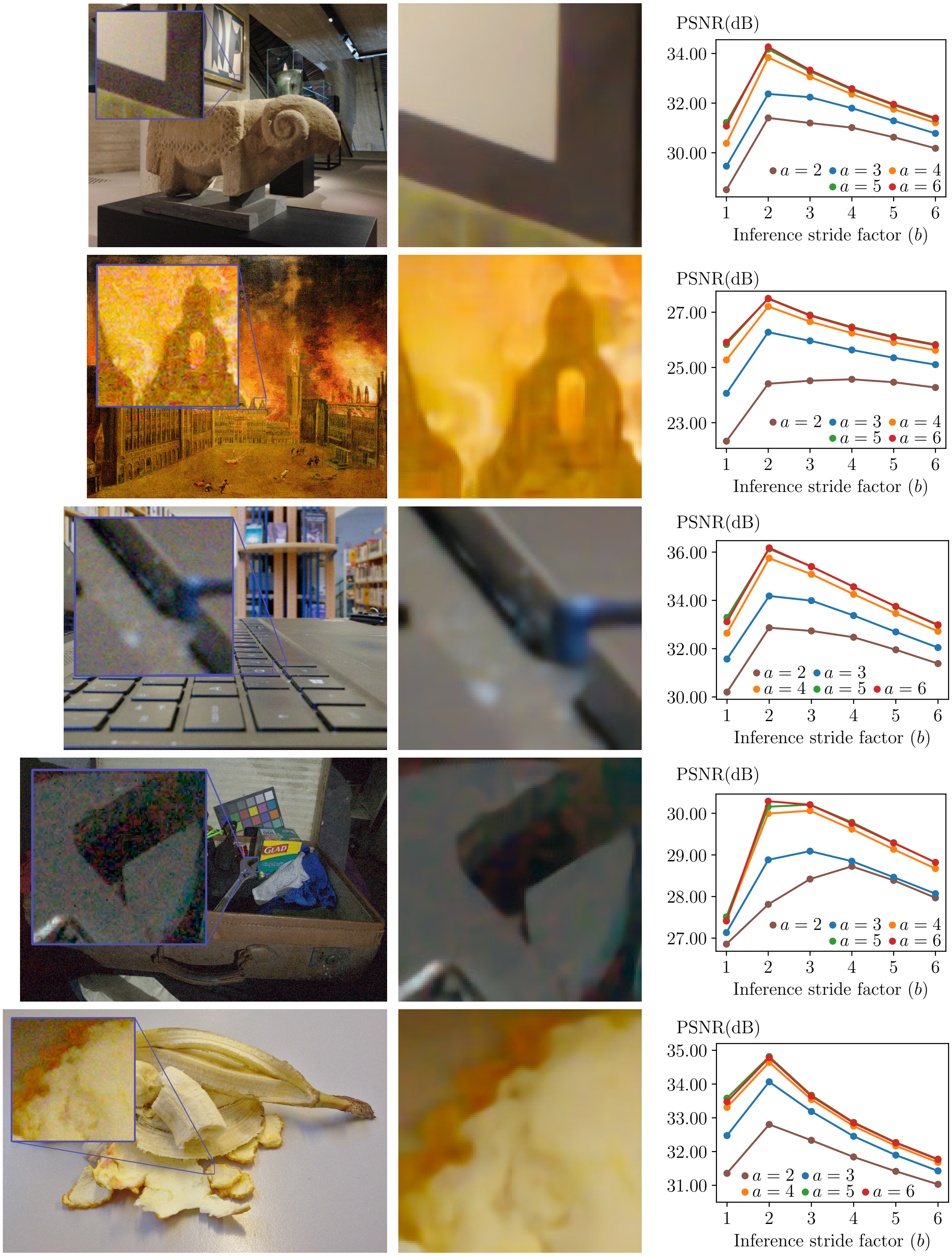
S5.2. Results on real-world inputs

Our AP-BSN is designed to handle real-world sRGB images, where appropriate training examples, *i.e.*, noisy-real pairs for supervised, a set of clean images for unpaired learning, may not exist. One of the major advantages of the proposed fully self-supervised framework is that we can apply our model on a *single* noisy test image directly without any pre-trained knowledge. To this end, we capture real-world noisy images under a high ISO condition using the recent Samsung Galaxy smartphone. Modern smartphone cameras usually incorporate software-based denoising algorithms to remove unpleasing noise from the captured scene. Therefore, we first acquire RAW data and leverage the simulated camera pipeline without explicit denoising stage [1] to get the corresponding sRGB images.

Fig. S5 visualizes denoising results of our method on the real-world sRGB images. Compared to the hardware-specific in-camera denoising algorithm in Fig. S5b, our approach reconstructs much sharper edges while suppressing unwanted noise signals effectively, as shown in Fig. S5d. The proposed method also outperforms DnCNN [10] trained on SIDD [1] noisy-clean pairs, while our formulation utilizes a *single* noisy image only for training.

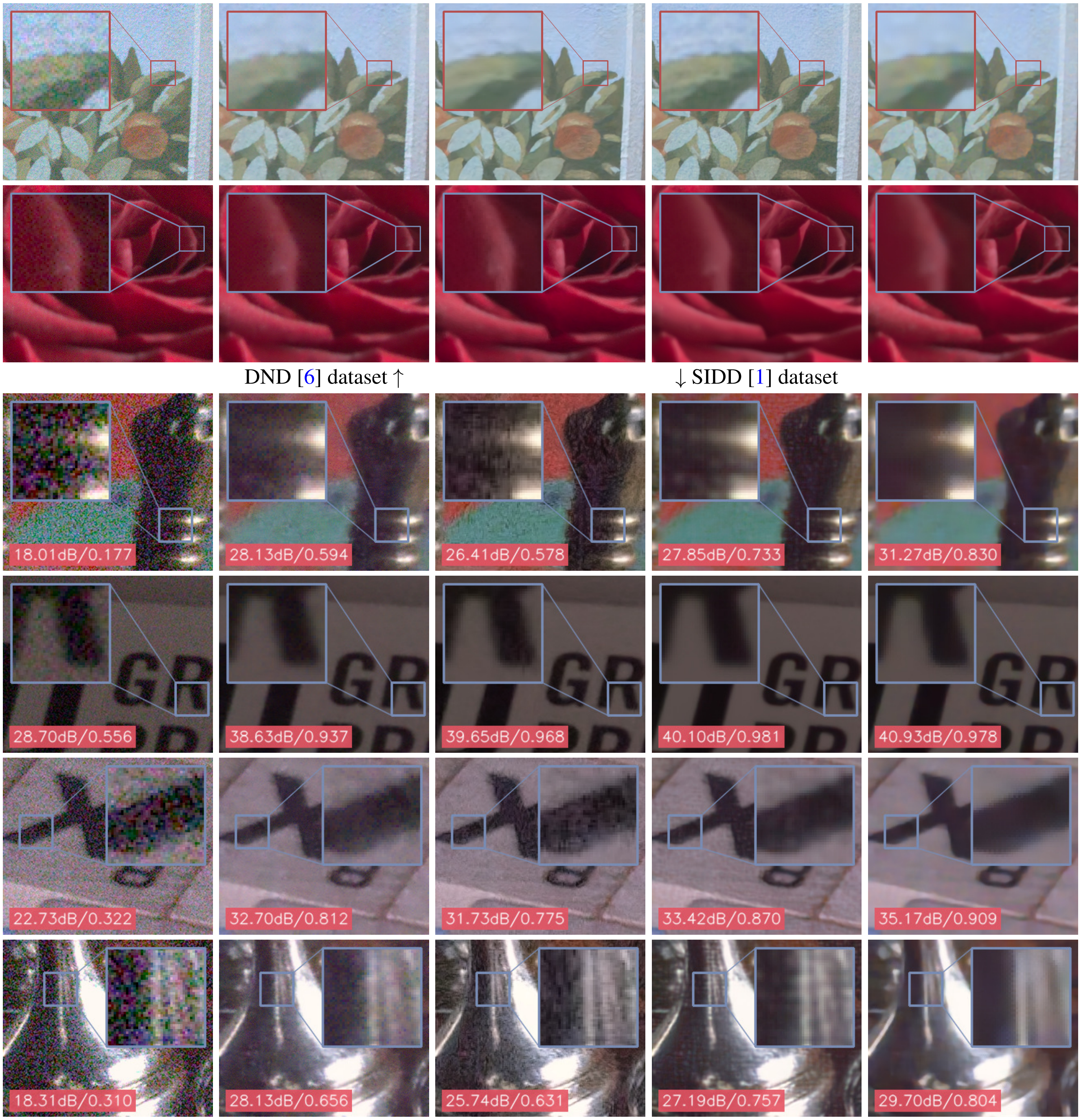
S5.3. Qualitative improvement by R^3

Our R^3 post-processing strategy significantly improves the performance of the proposed denoising method. Fig. S6 provides qualitative comparisons between AP-BSN *without* R^3 and AP-BSN + R^3. Without R^3, our AP-BSN tends to generate unpleasing blocky artifacts as shown in Fig. S6b. By using the proposed R^3, our AP-BSN can reconstruct smooth and natural image structures without requiring any additional parameters and training.



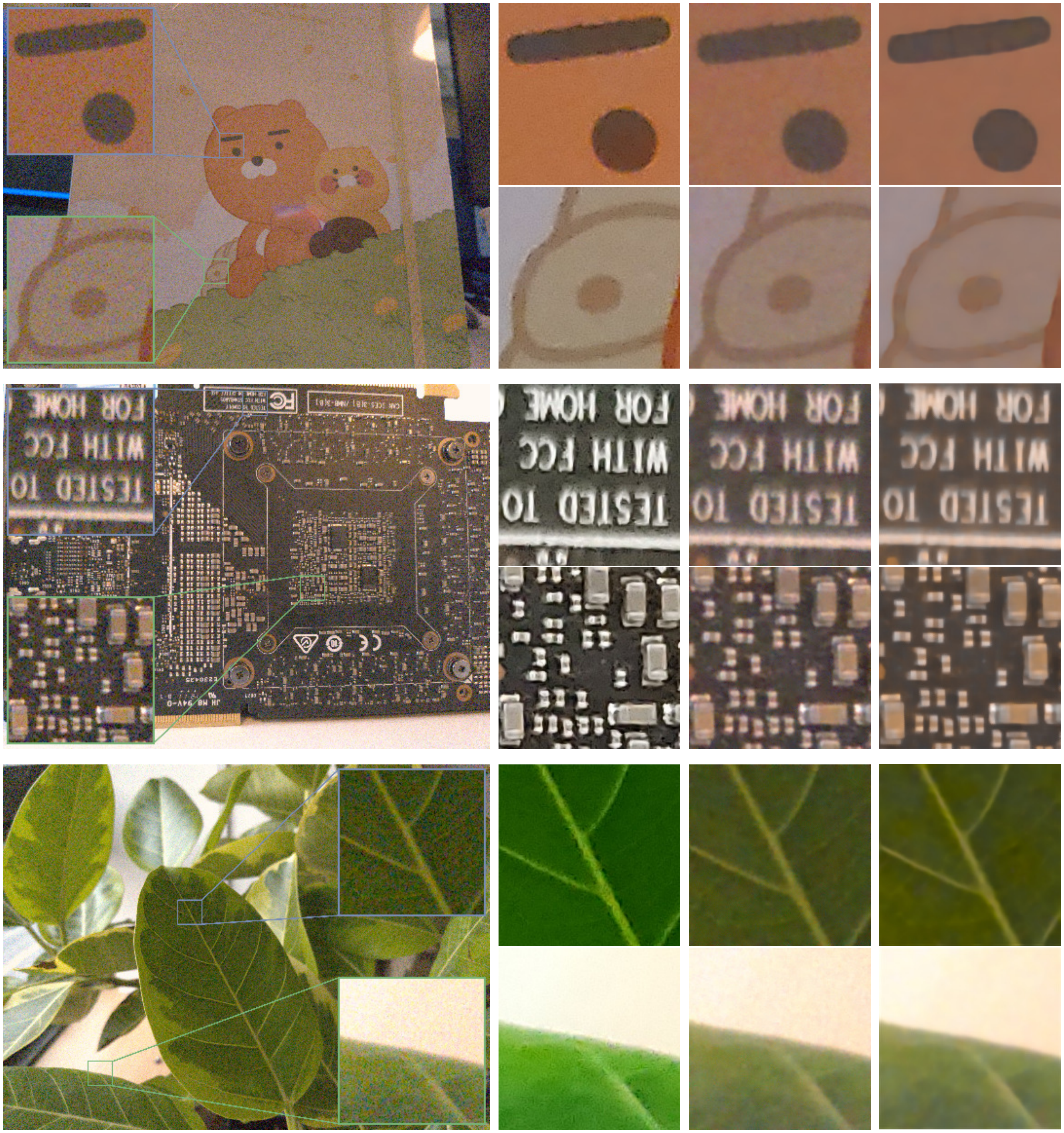
(a) Noisy images from the NIND [2] dataset (b) **AP-BSN + R^3 (Ours)** (c) Effects of AP$_{a/b}$

Figure S3. **Per-sample analysis of AP$_{a/b}$ on the NIND dataset.** (a) Noisy images sampled from the NIND dataset. From top: 'NIND_MuseeL-ram_ISO6400.jpg,' 'NIND_MVB-Bombardement_ISOH1.jpg,' 'NIND_LaptopInLibrary_ISO2500.png,' 'NIND_Iain02_ISO3200.png,' 'NIND_partiallyeatenbanana_ISO2500.png.' (b) Results of our AP-BSN + R^3 on the NIND dataset. We show local patches for better visualization. (c) Per-image trade-off analysis. The proposed AP-BSN performs consistently well when $b = 2$, while the best performance can be achieved when the training stride factor a is set to 5 or 6. Please see Fig. 7 in our main manuscript for more details.



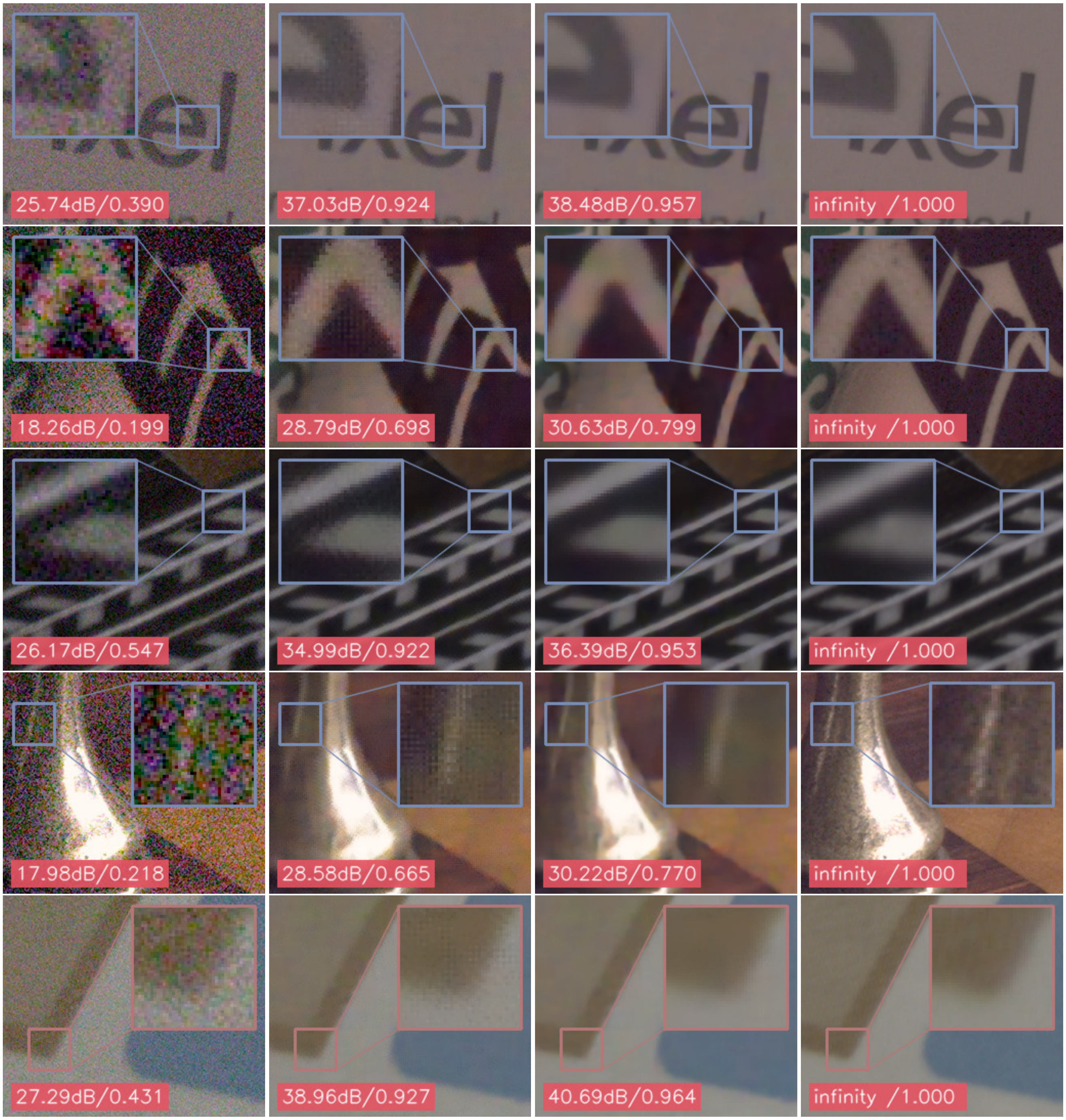
(a) Noisy | (b) DnCNN [10] Supervised - Real SIDD | (c) Zhou *et al.* [11] Supervised - Synthetic noise | (d) C2N [3] + DIDN [9] Unpaired | (e) **AP-BSN + R^3 (Ours)** Self-supervised

Figure S4. **Additional qualitative comparison between different methods on DND [6] benchmark and SIDD [1] validation datasets.** The upper two rows are examples from the DND benchmark dataset, and the lower four rows are from the SIDD validation dataset. (a) Input noisy images. (b) Same as Fig. 10a in our main manuscript, DnCNN is trained on the paired SIDD-Medium dataset. (c) Zhou *et al.* train their method on synthetic AWGN and impulse noise. During the inference, PD_2 is used to break the spatial correlation of real-world noise. (d) C2N generates a realistic noisy image from the clean input, where the following denoising model, *i.e.*, DIDN, is trained on the generated pairs. (e) Our method is directly applicable to practical sRGB noisy images in a self-supervised manner, which does not require any additional data. For quantitative comparison, we mark per-sample PSNR/SSIM w.r.t. the ground-truth image at the bottom left of each patch. We also note that ground-truth images are not available for the DND dataset.



(a) Real-world sRGB images under the high ISO condition (b) In-camera processing (c) DnCNN [10] on SIDD [1] (d) **AP-BSN + R^3 (Ours)**

Figure S5. **AP-BSN + R^3 on noisy images captured by ourselves.** (a) To avoid the in-camera denoising pipeline, we first capture RAW images with ISO 3200 using a recent Samsung Galaxy smartphone. We note that no other pre/post-processing is done on the exported *.dng* files. Then, we render the sRGB images using the SIDD ISP pipeline [1], which does not include the denoising process. (b) The corresponding sRGB images processed by the smartphone. We note that the recent mobile devices have adopted software-based denoising algorithms, which suppress unwanted noise from the captured images. (c) Same as Fig. 10a in our main manuscript, DnCNN is trained on the real-world SIDD pairs. (d) Results of our AP-BSN + R^3 trained on a *single* noisy input without any external data. We note that there exist color shifts between (a) and (b) since the simulated ISP pipeline does not know the color mappings of the actual ISP.



(a) Noisy image $\mathrm{I_N}$ (b) AP-BSN (c) **AP-BSN + $\mathbf{R^3}$ (Ours)** (d) Clean image $\mathrm{I_C}$

Figure S6. **Visual comparison between denoising results of AP-BSN *without* $\mathbf{R^3}$ and *with* $\mathbf{R^3}$ on SIDD validation dataset.** (b) Even with the smallest inference stride factor ($b = 2$), BSN leaves unpleasing artifacts on the denoised results and cannot preserve the image structures well. (c) The proposed $\mathrm{R^3}$ removes artifacts from BSN and significantly improves the denoising performances. For quantitative comparison, we also provide per-sample PSNR/SSIM w.r.t. ground-truth images at the bottom left of each patch.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 2, 4, 5

[2] Benoit Brummer and Christophe De Vleeschouwer. Natural image noise dataset. In *CVPR Workshops*, 2019. 2, 3

[3] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2N: Practical generative noise modeling for real-world denoising. In *ICCV*, 2021. 1, 4

[4] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level Wavelet-CNN for image restoration. In *CVPR Workshops*, 2018. 1

[5] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising. In *CVPR*, 2021. 2

[6] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 2, 4

[7] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *ECCV*, 2020. 1

[8] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-As-Clean: Learning self-supervised denoising from corrupted image. *IEEE TIP*, 29:9316–9329, 2020. 2

[9] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up CNN for image denoising. In *CVPR Workshops*, 2019. 1, 4

[10] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 26(7):3142–3155, 2017. 2, 4, 5

[11] Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When AWGN-based denoiser meets real noises. In *AAAI*, 2020. 4