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

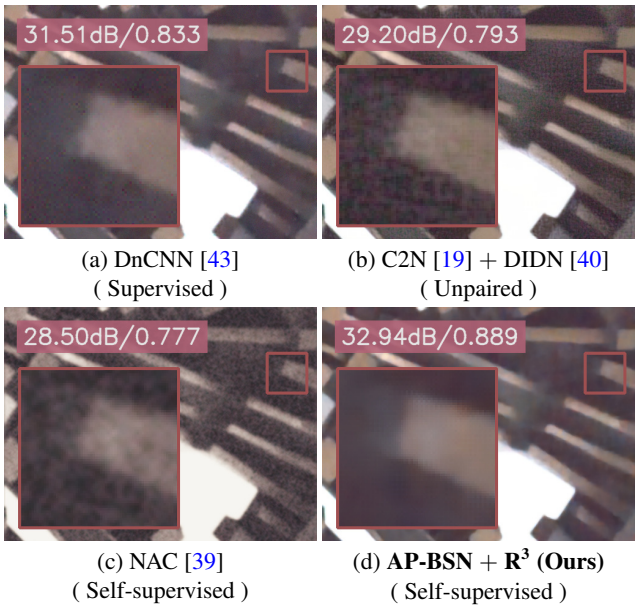
Wooseok Lee[1] Sanghyun Son[1] Kyoung Mu Lee[1,2]
[1]Dept. of ECE & ASRI, [2]IPAI, Seoul National University
adntjr4@gmail.com, {thstkdgus35, kyoungmu}@snu.ac.kr

Abstract

Blind-spot network (BSN) and its variants have made significant advances in self-supervised denoising. Nevertheless, they are still bound to synthetic noisy inputs due to less practical assumptions like pixel-wise independent noise. Hence, it is challenging to deal with spatially correlated real-world noise using self-supervised BSN. Recently, pixel-shuffle downsampling (PD) has been proposed to remove the spatial correlation of real-world noise. However, it is not trivial to integrate PD and BSN directly, which prevents the fully self-supervised denoising model on real-world images. We propose an Asymmetric PD (AP) to address this issue, which introduces different PD stride factors for training and inference. We systematically demonstrate that the proposed AP can resolve inherent trade-offs caused by specific PD stride factors and make BSN applicable to practical scenarios. To this end, we develop AP-BSN, a state-of-the-art self-supervised denoising method for real-world sRGB images. We further propose random-replacing refinement, which significantly improves the performance of our AP-BSN without any additional parameters. Extensive studies demonstrate that our method outperforms the other self-supervised and even unpaired denoising methods by a large margin, without using any additional knowledge, e.g., noise level, regarding the underlying unknown noise.

1. Introduction

Image denoising is one of the essential topics in the computer vision area, which aims to recover a clean image from the noisy signal. Due to its practical usage in several vision-related applications, several learning-based denoising algorithms [28, 36, 43, 44] have been proposed with the advent of convolutional neural networks (CNNs). Conventional methods usually adopt additive white Gaussian noise (AWGN) to acquire large-scale training data by synthesizing clean-noisy image pairs for supervised learning. Nevertheless, models learned on the synthetic noise do not generalize well in practice since the characteristics of real-world noise differ much from AWGN. To overcome the limitation, several attempts have been made to construct pairs of real-world datasets like SIDD [1] and NIND [4]. Using the real-world training pairs, supervised denoising methods [8, 16, 21, 41, 42] can be trained to restore clean images from the noisy real-world input. However, constructing the real-world dataset requires massive human labor, strictly controlled environments, and complicated post-processing. In addition, it is difficult to generalize the learned model toward diverse practical scenarios as the characteristic of noise varies much for the different capturing devices.



(a) DnCNN [43] (Supervised)
(b) C2N [19] + DIDN [40] (Unpaired)
(c) NAC [39] (Self-supervised)
(d) **AP-BSN + R^3 (Ours)** (Self-supervised)

Figure 1. **Visual comparison between different denoising methods on the DND benchmark [34].** (a) DnCNN is trained on real-world noisy-clean pairs from the SIDD [1] dataset. (b) C2N uses clean SIDD [1] and noisy DND [34] samples to simulate real-world noise distribution in an unsupervised manner. (c–d) Self-supervised methods can be trained on the DND [34] noisy images directly. We mark PSNR(dB) and SSIM with respect to the ground-truth clean image for the quantitative comparison.

Code is available at: https://github.com/wooseoklee4/AP-BSN

Recently, several self-supervised approaches [3, 17, 23, 24, 31, 38, 39] have been introduced, which do not rely on paired training data. Such methods require noisy images only for training instead of clean-noisy pairs. Among them, Blind-Spot Network (BSN) [23] is one of the representative methods motivated by Noise2Noise [25]. Under the assumption that noise signals are pixel-wise independent and zero-mean, BSN reconstructs a clean pixel from the neighboring noisy pixels without referring to the corresponding input pixel. Based on BSN, several approaches [15, 24, 37] have achieved better performance on synthetic noise while ensuring strict blindness w.r.t. the center pixel. However, real-world noises are known to be spatially-correlated [6, 20, 32], which does not meet the basic assumption of BSN: noise is pixel-wise independent.

To break spatial correlation of real-world noise, Zhou *et al*. [45] utilize pixel-shuffle downsampling (PD). PD creates a mosaic by subsampling a noisy image with a fixed stride factor, and thereby increases an actual distance between noise signals. Nevertheless, integrating PD to BSN is nontrivial when handling real-world noise in a fully self-supervised manner, where it cannot stand alone without knowledge from additional noisy-clean synthetic pairs [37]. We identify that the principal reason for such limitation is the trade-off between the pixel-wise independent assumption and reconstruction quality. For example, a large PD stride factor (> 3) ensures the strict pixel-wise independent noise assumption and benefits BSN during training. However, it also destructs detailed structures and textures from the noisy image. In contrast, a small PD stride factor (≤ 3) preserves image structures but cannot satisfy the pixel-wise independent assumption when training BSN.

Inspired by these observations, we propose Asymmetric PD (**AP**), which uses different stride factors for training and inference. For real-world noise, we systematically validate that a specific combination of training and inference strides can compensate shortcomings of each other. Then, we integrate AP to BSN (**AP-BSN**), which can learn to denoise noisy real-world inputs in a fully self-supervised manner, without requiring any prior knowledge of underlying noise. Furthermore, we propose random-replacing refinement ($\mathbf{R}^3$), a novel post-processing method that improves the performance of our AP-BSN without any additional training. To the best of our knowledge, our AP-BSN is the first attempt to introduce self-supervised BSN for real-world sRGB noisy images. Extensive studies demonstrate that our method outperforms not only the state-of-the-art self-supervised denoising methods but also several unsupervised/unpaired approaches by a large margin. We summarize our contributions as follows:

- To handle spatially correlated real-world noise in a blind fashion, we propose a novel self-supervised **AP-BSN**. Our framework employs asymmetric PD stride factors for training and inference in conjunction with BSN.
- We propose random-replacing refinement ($\mathbf{R}^3$), a novel post-processing method that further improves our AP-BSN without any additional parameters.
- Our AP-BSN is the first self-supervised BSN that covers real-world sRGB noisy inputs and outperforms the other self-supervised and even several unpaired solutions by large margins.

2. Related Work

Deep image denoising for synthetic noise. Beyond the classical non-learning based approaches [2, 9, 12, 18], DnCNN [43] has introduced a CNN-based architecture to remove AWGN from a given image. Following DnCNN, several learning-based approaches have been proposed such as FFDNet [44], RED30 [28], and MemNet [36], with advanced network architectures. Nevertheless, the methods trained on AWGN suffer from generalization toward the real-world denoising due to domain discrepancy between real and synthetic noises. Specifically, Guo *et al*. [13] have demonstrated that AWGN-based denoisers do not perform well when input noise signals are signal-dependent [10] or spatially-correlated [6, 20, 32].

Real-world image denoising. To reduce the gap between synthetic and real-world denoising, CBDNet [13] simulates in-camera ISP with gamma correction and demosaicking process. Then, synthetic heteroscedastic Gaussian noise can be transformed into realistic noise signals, which can be used to generate training pairs for supervised learning. Zhou *et al*. [45] have proposed pixel-shuffle downsampling (PD) to cover spatially-correlated real-world noise with conventional AWGN denoisers. In contrast, there have been a few attempts to capture the noisy-clean training pairs from real-world [1, 4]. Using the real-world pairs, it is straightforward to train supervised denoising methods [8, 16, 21, 41, 42], which generalize well on the corresponding real-world inputs. However, constructing real-world pairs require huge labor and is not always available.

Unpaired image denoising. When sets of unpaired clean and real-world noisy images are available, several methods leverage generative approaches [11] to synthesize realistic noise from the clean samples [5, 7, 14, 19]. Among them, GCBD [7] selectively uses plain regions from noisy images for stable learning. Recently, C2N [19] explicitly considers various noise characteristics to simulate real-world noise more accurately. Using the generated noisy-clean pairs, the following supervised denoising model [40, 43] can be trained to deal with real-world noise. On the other hand, Wu *et al*. [37] distill knowledge from a self-supervised denoising model while adopting synthetic noisy-clean pairs. Still, it is important to match the scene statistics of clean and noisy datasets even in the unpaired configuration [19], which can be difficult in practice.

Self-supervised denoising. A major bottleneck for real-world denoising is the absence of appropriate training data. Therefore, several approaches have been proposed to train their model using noisy images *only*. Motivated by Noise2Noise [25], Noise2Void [23] and Noise2Self [3] have introduced novel self-supervised learning frameworks by masking a portion of noisy pixels from the input image. Notably, the concept of BSN [23] has been later extended to more efficient architectures in the form of four halved receptive fields [24] or dilated and masked convolutions [37]. While Noise2Same [38] does not use BSN, a novel loss term is used to satisfy $\mathcal{J}$-invariant property [3] in the denoising network. Neighbor2Neighbor [17], on the other hand, acquires the noisy-noisy pair for self-supervision by subsampling the given input. Nevertheless, the above self-supervised methods heavily rely on assumptions that noise signals are pixel-wise independent. Therefore, they usually end up learning *identity* mappings when applied to real-world sRGB images as noise signals are spatially-correlated [6, 20, 32].

Recent Noisier2Noise [29], NAC [39], and R2R [31] add different synthetic noise signals to the given input to make auxiliary training pairs. However, Noisier2Noise requires prior knowledge regarding the underlying noise distribution, and Noisy-As-Clean relies on weak noise assumptions. R2R also requires several prior information such as noise level and ISP function, which may not be available in real-world scenarios.

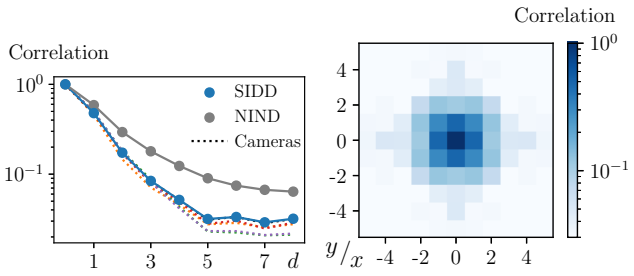
3. BSN and PD

Blind-spot network. BSN [23] is a variant of the conventional CNN that does not *see* the center pixel in the receptive field to predict the corresponding output pixel. Several studies [3, 23] have demonstrated that BSN $B(\cdot)$ can learn to denoise a noisy image $\mathrm{I_N} \in \mathbb{R}^{H \times W}$ in a self-supervised manner. We note that the image has a resolution of $H \times W$, and color channels are omitted for simplicity. To train BSN, the following two assumptions must be satisfied: noise is spatially, *i.e.*, pixel-wise, independent and zero-mean. Under such assumptions, it is known [3, 38] that minimizing the self-supervised loss $\mathcal{L}_{\text{self}}$ w.r.t. BSN is equivalent to conventional supervised learning as follows:

$$\begin{aligned}\mathcal{L}_{\text{self}} &= \mathbb{E}_{\mathrm{I_N}} \left\| B\left(\mathrm{I_N}\right) - \mathrm{I_N} \right\|_2^2 \\ &= \mathbb{E}_{\mathrm{I_N},\mathrm{I_C}} \left\| B\left(\mathrm{I_N}\right) - \mathrm{I_C} \right\|_2^2 + c = \mathcal{L}_{\text{super}} + c,\end{aligned} \tag{1}$$

where $\mathrm{I_C} \in \mathbb{R}^{H \times W}$ is a clean ground-truth for the noisy input $\mathrm{I_N}$, $\mathcal{L}_{\text{super}}$ is a supervised denoising loss function, and c is a constant, respectively.

Therefore, several types of BSN [24, 37] are constructed under the pixel-wise independent noise assumption. However, real-world noise is spatially correlated due to the image signal processors (ISP). Specifically, demosaicking on Bayer filter [6, 20, 32] involves interpolation between noisy subpixels. Fig. 2 demonstrates that in real-world, noise intensities between neighboring pixels show non-negligible correlation based on their relative distance. Since the neighboring noise signals can be clues for inferring the unseen center pixel, we have identified that BSN operates as an approximately identity mapping on real-world sRGB images.



(a) By relative distances d (b) By relative locations

Figure 2. **Analysis of spatial correlation on real-world noise.** (a) As the relative distance d between two noise signals increases, their correlation decreases. We note that different camera devices, *e.g.*, **Motorola Nexus 6 (N6)** or **LG G4**, in the SIDD [1] dataset show similar noise behaviors in terms of spatial correlation, as illustrated with dotted lines. (b) x and y axis represent a relative distance along with horizontal and vertical directions, respectively.

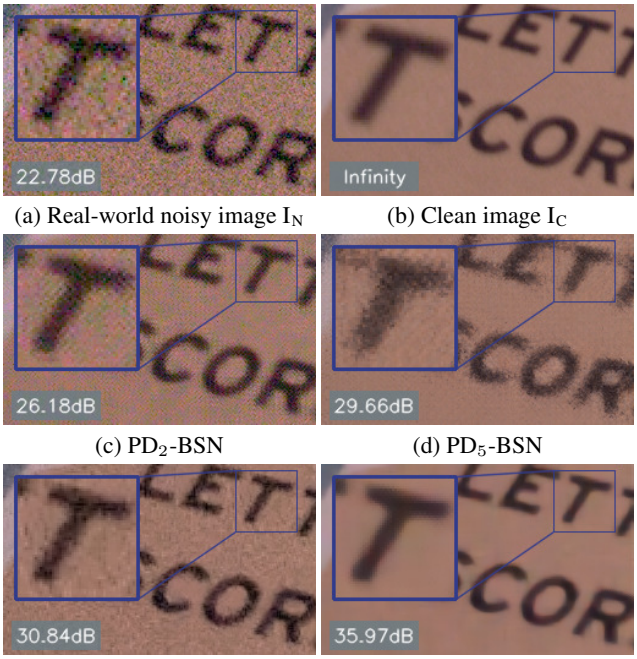
Pixel-shuffle downsampling. Zhou *et al.* [45] have introduced a novel concept of PD to break down the spatial correlation in the real-world noise. Specifically, PD_s can be regarded as an inverse operation of the pixel-shuffling [35] with a stride factor of s. Since real-world noise signals are correlated with few neighboring pixels, subsampling in PD process may break the dependency between them. Then, conventional denoising algorithms can be applied to the downsampled images, where the PD-inverse operation PD_s^{-1} follows to reconstruct a full-sized output. To preserve image textures and details, Zhou *et al.* [45] set the stride factor to 2, *i.e.* PD_2, for the best performance.

4. Method

Our goal is to generalize BSN on real-world sRGB images in a self-supervised manner. To this end, we adopt PD and minimize the following loss $\mathcal{L}_{\text{BSN}}$ to train BSN:

$$\begin{aligned}\mathcal{L}_{\text{BSN}} &= \left\| \mathrm{PD}_s^{-1}\left(B\left(\mathrm{PD}_s\left(\mathrm{I_N}\right)\right)\right) - \mathrm{I_N} \right\|_1 \\ &= \left\| \mathrm{I}_{\text{BSN}}^s - \mathrm{I_N} \right\|_1,\end{aligned} \tag{2}$$

where $\mathrm{I}_{\text{BSN}}^s$ is an output from PD_s and BSN pipeline, namely **PD_s-BSN**. Instead of widely-used L^2 loss, we use L^1 norm for better generalization [26]. In brief, we first decompose the given noisy image $\mathrm{I_N}$ into s^2 sub-images. We note that $\mathrm{PD}_s\left(\mathrm{I_N}\right)$ is a tiling of those sub-images [45] $\mathrm{I}_{\text{sub}}^s \in \mathbb{R}^{H/s \times W/s}$, as shown in Fig. 4. Then, we apply BSN to the sub-images and reconstruct the output $\mathrm{I}_{\text{BSN}}^s$ using the PD-inverse operation PD_s^{-1}.



(a) Real-world noisy image I_N (b) Clean image I_C

(c) PD_2-BSN (d) PD_5-BSN

(e) Zhou *et al*. [45] (f) **AP-BSN + R^3 (Ours)**

Figure 3. **Issues on PD$_s$-BSN when handling real-world noise.** (c) With a small stride factor, PD-BSN cannot remove noise from the input I_N. (d) With a large stride factor, PD-BSN destructs edge structures. (e) When AWGN denoiser meets PD [45], the model cannot completely remove real-world noise. (f) Our self-supervised approach delivers an accurate denoising result by overcoming the limitation of combining PD and BSN.

However, it is not straightforward to apply PD-BSN directly on real-world sRGB images. While Wu *et al*. [37] have also tried to integrate PD and BSN, they resort to knowledge distillation combined with additional synthetic noisy-clean pairs. We have also observed that PD-BSN is not applicable to real-world noisy images when trained with the self-supervised loss in Eq. (2). Figs. 3c and 3d demonstrate that PD_2-BSN and PD_5-BSN cannot restore a clean and sharp image from the given noisy input, regardless of the PD stride factor s.

4.1. Trade-offs in PD-BSN

When applying the AWGN-based denoiser on real-world images, Zhou *et al*. [45] use PD_2. However, we have observed that PD exhibits different behaviors as the stride factor s varies. Therefore, we first describe two important aspects of PD-BSN regarding the stride factor s.

Breaking spatial correlation. Originally, PD has been proposed to reduce spatial correlation between neighboring noise signals in real-world images. While Zhou *et al*. [45] resort to the stride factor of 2, our analysis in Fig. 2a demonstrates that the stride factor should be at least 5 to minimize the dependency in the given noise signal. In other words, noise signals in the sub-images I^2_{sub} are still spatially correlated, where the pixel-wise independent noise assumption for BSN does not hold.

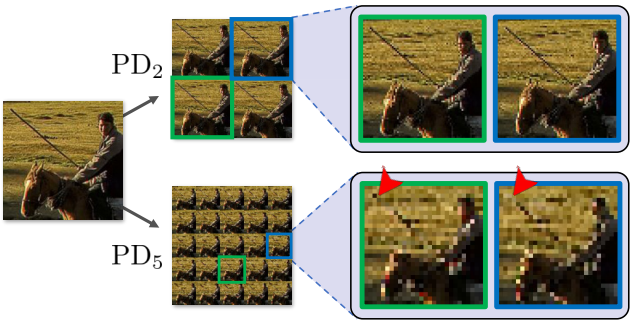


Figure 4. **Comparison between PD$_2$ and PD$_5$.** Each operation decomposes the given image into 4 and 25 sub-images, respectively. In sub-images from PD_5, we mark the aliasing artifact, *i.e*. a black dot, with **red**, which can be interpreted as noise for BSN. We note that the artifact does not appear in the **blue** sub-image.

Aliasing artifacts. Nevertheless, the sub-images I^s_{sub} from PD_s suffer stronger degree of aliasing as the stride factor s becomes larger. From the perspective of signal processing, it is well-known that a downsampled image suffers aliasing when the original signal is not properly bandlimited [30]. Since the PD process does not leverage a low-pass filter before subsampling, we have identified that aliasing occurs as a form of noise when applying large-stride PD, *e.g*., $s = 5$, as shown in Fig. 4.

4.2. Effective training stride factor for PD-BSN

We next establish a strategy to *train* PD_s-BSN. For such purpose, the correlation between noise signals in the training input images I_N has to be minimized [23]. However, as discussed in Section 4.1, PD_2 is not enough to break spatial correlation of real-world noise. Since the underlying assumption of BSN is not satisfied, the model *cannot* learn to denoise with PD_2. By setting $s = 5$ to suppress the spatial correlation between noise signals in training samples, we *can* train BSN on the smaller sub-images I^5_{sub}.

We note that BSN also learns to remove the aliasing artifacts induced by the large PD stride factor. The aliasing happens when high-frequency signals are not removed before subsampling [30]. As the high-frequency components change rapidly in the original noisy image I_N, we can *ignore* the spatial correlation of aliasing artifacts in the sub-images I^5_{sub}. The artifacts also satisfy the *zero-mean* constraint, *i.e*., their statistical mean is approximately the same as that of the noisy image I_N, since they are random samples of the observed signal. As the aliasing artifacts satisfy two preconditions of BSN, our PD-BSN also learns to remove them.

4.3. Asymmetric PD for BSN

Several studies [7,19] have already identified that matching data distribution between training and test samples play a critical role in accurate image denoising. Therefore, it is

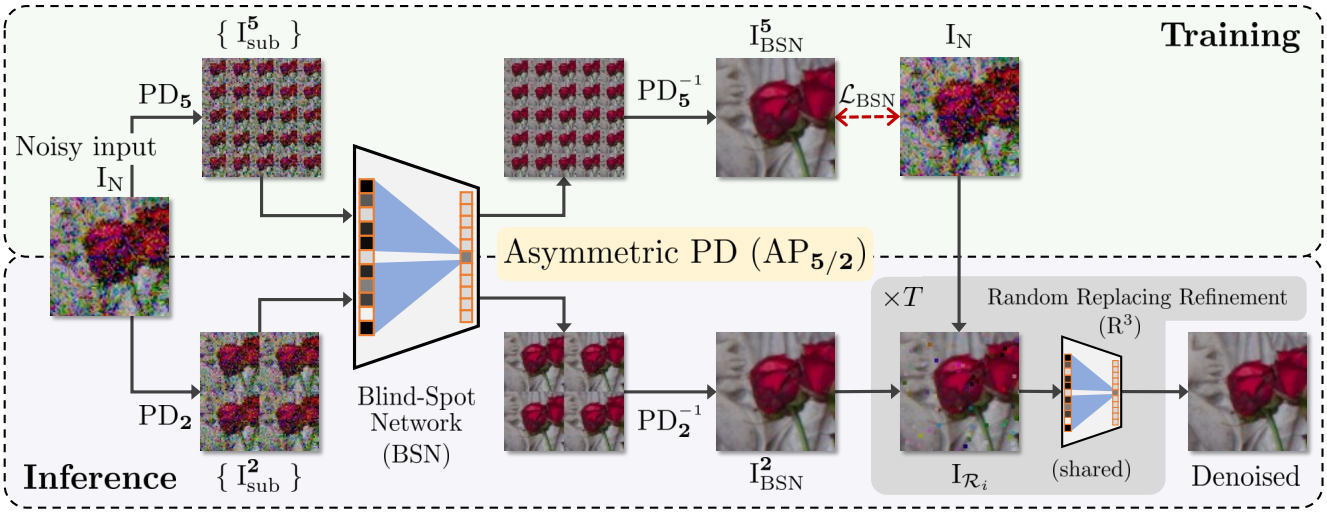


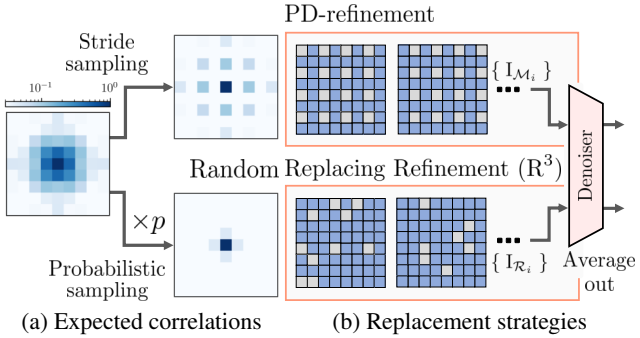
Figure 5. **Overview of the proposed AP-BSN and R^3 post-processing.** We visualize the proposed $AP_{5/2}$-BSN. To apply BSN on real-world sRGB images, we introduce $AP_{a/b}$ to maximize synergies of using different stride factors for training and inference. We use a large stride factor, *e.g.*, $a = 5$, to ensure pixel-wise independence between noise signals for training. During the inference, we use a minimum stride factor of $b = 2$ to avoid aliasing artifacts while breaking down the spatial correlation of noise to some extent. Our random-replacing refinement (R^3) further improves the performance of AP-BSN without any additional parameters.

natural to use the same stride factor for training and inference when applying PD-BSN. However, we have found that the learned BSN recognizes aliasing artifacts from PD_5 as noise signals to be removed during *inference*. Since those artifacts contain necessary information to reconstruct high-frequency details, PD_5-BSN destructs image structures during inference while removing noise as shown in Fig. 3d.

Instead, we propose an asymmetric stride factor during the *inference* of PD-BSN, which we refer to as Asymmetric PD ($AP_{a/b}$). We note that a and b are stride factors for training and inference, respectively. Specifically, we set $b = 2$ so that the sub-images I^2_{sub} contain minimum aliasing artifacts during inference, while the correlation between neighboring noise signals can be decreased. In Section 5, we demonstrate how each trade-off, *i.e.*, spatial correlation and aliasing artifacts, affects the denoising performance of our method. Our BSN with the proposed $AP_{5/2}$ (AP-BSN) can learn to remove real-world noise in a self-supervised manner, while preserving image structures as shown in Fig. 3f. We also note that our AP-BSN does not require any clean samples for training and is directly applicable to sRGB noisy images in practical scenarios. Fig. 5 illustrates our asymmetric training and inference schemes for AP-BSN.

4.4. Random-replacing refinement

Even with the smallest stride factor, PD and the following denoising step may remove some informative high-frequency components from the input, resulting in visual artifacts [45]. Therefore, Zhou *et al.* [45] propose PD-refinement to suppress artifacts from the PD process and enhance details of the denoising result. In PD-refinement,



(a) Expected correlations (b) Replacement strategies

Figure 6. **Comparison between PD-refinement and our R^3.** While PD-refinement adopts regular binary masks $\mathcal{M}_i$ with a stride of 2, our R^3 uses randomized masks $\mathcal{R}_i$. (a) We compare the expected spatial correlation of noise signals in the replaced image $I_{\mathcal{M}_i}$ and $I_{\mathcal{R}_i}$. (b) Each gray box represents a pixel from the original noisy image I_N, which replaces the denoised pixel in I^s_{BSN}.

an i-th replaced image $I_{\mathcal{M}_i}$ is formulated as follows:

$$I_{\mathcal{M}_i} = \mathcal{M}_i \odot I_N + (\mathbf{1} - \mathcal{M}_i) \odot I^s_{BSN}, \tag{3}$$

where $\mathcal{M}_i \in \{0, 1\}^{H \times W}$ is a binary mask indicating pixels to be replaced and $\odot$ denotes element-wise multiplication. Here, $\mathcal{M}_i$ is a structured binary matrix where ones are placed with a fixed stride of 2 and $\sum_i \mathcal{M}_i = \mathbf{1}$. After the replacement, each image $I_{\mathcal{M}_i}$ is denoised again and averaged to reconstruct the final result I_{DN} as follows:

$$I_{DN} = \frac{1}{T} \sum_{i=1}^{T} D\left(I_{\mathcal{M}_i}\right), \tag{4}$$

where D is the denoising model targeting pixel-wise independent noise and T is the number of masks, *i.e.*, $2^2 = 4$, for the original PD-refinement.

However, the deterministic strategy in PD-refinement leaves a non-negligible correlation between the replaced noise signals. Specifically, a replaced noisy pixel in $\mathrm{I}_{\mathcal{M}_i}$ is *always* correlated with some of its neighbors, as visualized in Fig. 6a. Such correlation negatively affects the performance of the following denoising method D, which assumes spatially uncorrelated noise. Therefore, we propose an advanced **random-replacing refinement** ($\mathbf{R}^3$) strategy to mitigate the limitation of PD-refinement.

In our R^3, we adopt T randomized binary masks $\mathcal{R}_i$ instead, which are defined as follows:

$$\mathcal{R}_i\left(x, y\right) = \begin{cases} 1, & \text{with a probability of } p, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where (x, y) denotes an index of the element in a $H \times W$ matrix. For Eq. (3) and Eq. (4), we adopt the randomized mask $\mathcal{R}_i$ rather than the fixed one $\mathcal{M}_i$ to acquire the final output. Since noisy pixels are randomly placed in the i-th replaced image $\mathrm{I}_{\mathcal{R}_i}$, an expected correlation between two noise signals is multiplied by p, as shown in Fig. 6a. Hence, our R^3 significantly reduces the expected correlation compared to the previous PD-refinement. When we combine R^3 with AP-BSN, we do not perform PD and feed the replaced image $\mathrm{I}_{\mathcal{R}_i}$ to BSN directly because spatial correlation of noise in the input is almost negligible. Fig. 6 highlights major differences between PD-refinement and our R^3.

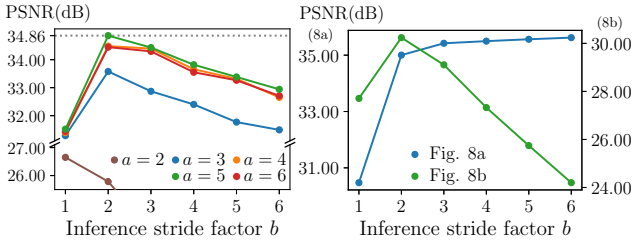
5. Experiment

5.1. Experimental configurations

Dataset. To train and evaluate our AP-BSN, we adopt widely-used real-world image denoising datasets: SIDD [1] and DND [34]. SIDD-Medium consists of 320 real-world noisy and clean image pairs for training. For validation and performance evaluation, we adopt SIDD validation and benchmark datasets, respectively. Both contain 1,280 noisy patches with a size of 256×256, where the corresponding clean images are also provided for the validation set.

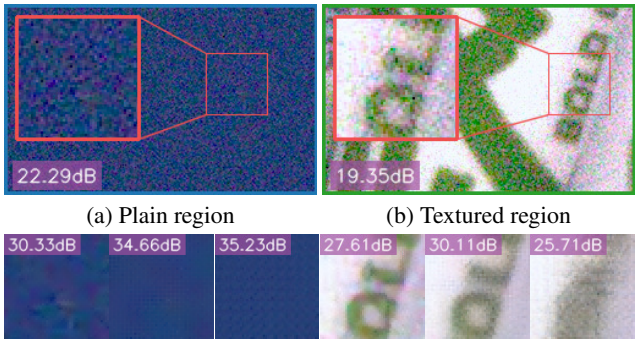
The DND dataset does not include training images and consists of 50 real-world noisy inputs only for evaluation. Rather than using the SIDD-Medium training dataset for this case, we enjoy the advantage of a fully self-supervised learning framework and use the same data for training and performance evaluation. In other words, we train our AP-BSN on 50 noisy DND images and reconstruct the final denoising results from the same inputs.

Metric. To evaluate our AP-BSN and compare it with the other denoising methods, we introduce widely-used peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics. For SIDD and DND benchmarks, we upload our results to the evaluation sites to calculate the metrics. On the SIDD validation dataset, we use the corresponding functions in `skimage.metrics` library and RGB color space for comparison.



(a) Effects of asymmetric a/b (b) Effects of aliasing artifacts

Figure 7. **Ablation study of $\mathrm{AP}_{a/b}$-BSN on the SIDD validation dataset.** We note that the proposed R^3 post-processing is not applied in these ablation studies. (a) Our $\mathrm{AP}_{a/b}$-BSN consistently achieves the best performance when $b = 2$. (b) We validate $\mathrm{AP}_{5/b}$-BSN on two representative images displayed in Figs. 8a and 8b.



(a) Plain region (b) Textured region

(c) $\mathrm{AP}_{5/1}$ (d) $\mathrm{AP}_{5/2}$ (e) $\mathrm{AP}_{5/5}$ (f) $\mathrm{AP}_{5/1}$ (g) $\mathrm{AP}_{5/2}$ (h) $\mathrm{AP}_{5/5}$

Figure 8. **Visual comparison of the trade-off in $\mathrm{AP}_{a/b}$-BSN.** (c–e) For a plain region in (a), performance of AP-BSN gradually increases as the inference stride factor b becomes larger. (f–h) For a textured region in (b), AP-BSN performs the best when $b = 2$ but shows decreased performance for larger b. Please refer to Fig. 7b for more details.

Implementation and optimization. We use PyTorch 1.9.0 [33] for implementation. By default, we adopt $\mathrm{AP}_{5/2}$ and set p and T to 0.16 and 8, respectively, for the proposed R^3. For BSN, we modify the architecture from Wu *et al.* [37] for efficiency. AP-BSN is trained using Adam [22] optimizer, and the initial learning rate starts from 10^{-4}. More details are described in our supplementary material.

5.2. Analyzing Asymmetric PD

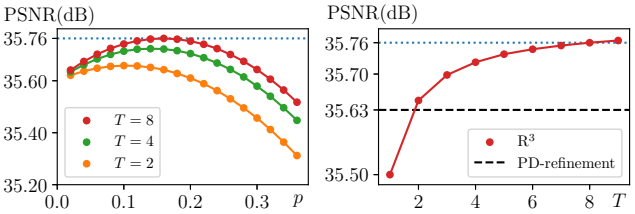
We first validate the effect of AP for real-world sRGB denoising. To this end, we conduct an extensive study on all possible combinations of feasible stride factors, *i.e.*, $a \in \{2, 3, 4, 5, 6\}$ and $b \in \{1, 2, 3, 4, 5, 6\}$, in Fig. 7a. We note that BSN cannot be trained when $a = 2$ due to the spatial correlation of real-world noise. With larger training stride factors a, the input noise of BSN follows pixel-wise independent assumption more strictly. Therefore, the model can learn the denoising function better, where the performances are maximized with $a = 5$. When $a = 6$ is used,

	Method	SIDD PSNR↑(dB)	SIDD SSIM↑	DND PSNR↑(dB)	DND SSIM↑
Non-learning based	BM3D [9]	25.65	0.685	34.51	0.851
	WNNM [12]	25.78	0.809	34.67	0.865
Supervised (Synthetic pairs)	DnCNN [43]	23.66	0.583	32.43	0.790
	CBDNet [13]	33.28	0.868	38.05	0.942
	Zhou *et al*. [45]	34.00$^{\diamond}$	0.898$^{\diamond}$	38.40	0.945
Supervised (Real pairs)	DnCNN [43]	35.13$^{\diamond}$	0.896$^{\diamond}$	37.89$^{\diamond}$	0.932$^{\diamond}$
	AINDNet (R)* [21]	38.84	0.951	39.34	0.952
	VDN [41]	39.26	0.955	39.38	0.952
	DANet [42]	39.43	0.956	39.58	0.955
Unsupervised (Unpaired)	GCBD [7]	-	-	35.58	0.922
	C2N [19] + DIDN* [40]	35.35	0.937	37.28	0.924
	D-BSN [37] + MWCNN [27]	-	-	37.93	0.937
Self-supervised	Noise2Void [23]	27.68$^{\mathbf{R}}$	0.668$^{\mathbf{R}}$	-	-
	Noise2Self [3]	29.56$^{\mathbf{R}}$	0.808$^{\mathbf{R}}$	-	-
	NAC [39]	-	-	36.20	0.925
	R2R [31]	34.78	0.898	-	-
	AP-BSN (Ours)	**34.90**	**0.900**	**37.46**	**0.924**
	AP-BSN + R^3 (Ours)	**35.97**	**0.925**	**38.09**	**0.937**
	AP-BSN† + R^3 (Ours)	**36.91**	**0.931**	-	-

Table 1. **Quantitative comparison of various denoising methods on the SIDD and DND benchmarks.** We note that several supervised methods leverage SIDD noisy-clean pairs for training and perform much better than our AP-BSN, while we use noisy sRGB images only for training. By default, we report official evaluation results from SIDD and DND benchmark websites. $\diamond$ and **R** indicate that the performances are evaluated by ourselves, or reported from R2R [31], respectively. We also mark methods with $*$ which adopt self-ensemble strategy [26]. † denotes that the model is trained on SIDD benchmark images in a fully self-supervised fashion.

AP$_{6/b}$-BSN performs slightly worse since the noise in the SIDD [1] dataset show increasing correlation as shown in Fig. 2a. Interestingly, $a = 6$ is slightly better than $a = 5$ on the NIND [4] dataset, as the correlation gradually decreases w.r.t. to the relative distance between pixels. More analysis on the NIND dataset is reported in our supplementary material. During the inference, BSN cannot remove real-world noise without PD, *i.e.*, $b = 1$, as it is learned on pixel-wise independent noise. The performances are maximized when $b = 2$, as the trade-off between spatial correlation and aliasing can be optimized. With larger inference stride factors, *i.e.*, $b > 2$, AP-BSN performs worse because more image details are removed in the form of aliasing artifacts.

In Fig. 7b, we justify that the existence of aliasing artifacts is a critical factor for our denoising framework. When applying AP$_{5/b}$-BSN to the plain region illustrated in Fig. 8a, the model performs better as the inference stride factor b becomes larger. Since the region does not contain high-frequency information, aliasing artifacts do not appear in Figs. 8c, 8d, and 8e. Rather, the spatial correlation of noise signals becomes smaller with a larger b, which results in better performance. For a general image in Fig. 8b, our AP$_{5/b}$-BSN shows a similar behavior to that of Fig. 7a, while the performance drop is much severe due to the stronger aliasing artifacts as shown in Fig. 8h.



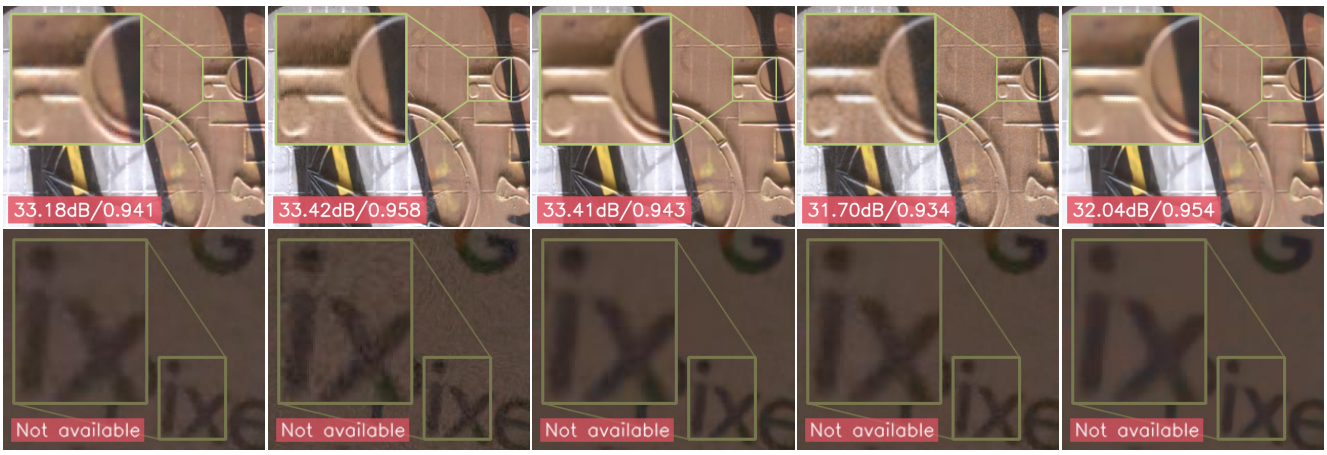
(a) Ablations on p (b) Ablations on T

Figure 9. **Ablation study of AP-BSN + R^3 on the SIDD validation dataset.** We note that AP-BSN without R^3 achieves 34.86dB on the same dataset. (a) We investigate the effect of different p for $T = 2, 4, 8$. (b) We fix $p = 0.16$ to see the effect of T in our R^3.

5.3. Analyzing Random-Replacing Refinement

Fig. 9 shows a detailed ablation study on hyperparameters for the proposed R^3. We first set $T = 2, 4, 8$ to find the optimal replacement probability p. As shown in Fig. 9a, our R^3 shows a consistent behavior where the maximum performance is achieved with $p \approx 0.16$. We note that a larger p increases the expected spatial correlation of noise signals which degrades the performance. Due to the stochastic behavior, the number of randomized masks T is not limited in our R^3, while PD-refinement can only use $T = 4$. Fig. 9b demonstrates that the proposed R^3 performs better than PD-refinement even with $T = 2$, and the performance increases as the number of randomized masks T goes higher. Since

(d) NAC [39]



(a) DnCNN [43] Supervised - Real SIDD | (b) Zhou *et al*. [45] Supervised - Synthetic noise | (c) C2N [19] + DIDN [40] Unpaired | (e) R2R [31] Self-supervised | (f) **AP-BSN + R^3 (Ours)** Self-supervised

Figure 10. **Qualitative comparison between different denoising methods on DND [34] and SIDD [1] benchmarks.** (a) DnCNN is trained on the paired SIDD-Medium dataset. (b) Zhou *et al*. train their method on synthetic AWGN and impulse noise. The learned denoising model is then combined with PD to handle real-world noise. (c) C2N generates a realistic noisy image from the clean input, where the following denoising model, *i.e*., DIDN, is trained on the generated pairs. (d–e) Recent self-supervised approaches are trained on noisy images only. (f) Our method is directly learnable on the practical sRGB images. We note that the DND benchmark (**Upper**) provides per-sample PSNR/SSIM, while SIDD benchmark (**Lower**) does not, *i.e*., Not available.

the computational complexity of R^3 is proportional to T, we set $T = 8$ to balance the performance and runtime.

5.4. AP-BSN for real-world denoising

Our AP-BSN aims to denoise real-world sRGB images in a self-supervised manner. Table 1 compares various image denoising models on widely-used SIDD and DND benchmark datasets. Using noisy images *only* for training, the proposed AP-BSN + R^3 achieves the best performance among several unpaired [19, 37] and self-supervised approaches. Especially, we note that self-supervised NAC [39] and R2R [31] are constructed on less practical assumptions like noise level is weak or ISP function is known. On the other hand, our approach adopts BSN with several observations regarding the properties of PD and real-world noise. Therefore, we do not rely on specific assumptions and show better generalization on several real-world datasets. In addition, the proposed R^3 post-processing further improves the evaluation PSNR more than 1dB on the SIDD benchmark track without any additional parameters. Fig. 10 provides visual comparisons between several methods addressed in Table 1.

Furthermore, AP-BSN can be trained on noisy samples directly, without using any clean images. Since several un-/self-supervised methods are trained on auxiliary images [31] or generated noise [19], the discrepancy between training and test distributions may result in sub-optimal solutions. In contrast, our approach can use target sRGB noisy images directly during training phase. To validate the merit of our framework, we train AP-BSN on the SIDD benchmark and evaluate on the *same* dataset. The last row of Table 1 shows that the fully self-supervised strategy improves the denoising performance by about 1dB without making any modifications. Although SIDD-Medium contains about $\times 60$ more pixels than the benchmark split, such an improvement highlights that AP-BSN can also generalize well on practical cases where there exist noisy test samples only.

6. Conclusion

In this paper, we first identify several trade-offs regarding different PD stride factors in perspective of BSN. Rather than directly integrate PD and BSN, we propose asymmetric PD between training and inference to satisfy pixel-wise independent assumption while preserving image details. To this end, we propose AP-BSN, a fully self-supervised approaches for real-world denoising. We also propose random-replacing refinement R^3, which removes visual artifacts of AP-BSN without any additional parameters. The proposed AP-BSN + R^3 does not require any prior knowledge on real-world noise and outperforms recent self-supervised/unsupervised denoising methods.

Acknowledgment

This work was supported in part by IITP grant funded by the Korea government (MSIT) [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), and No.2021-0-02068, Artificial Intelligence Innovation Hub]

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 1, 2, 3, 6, 7, 8

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE TSP*, 54(11):4311–4322, 2006. 2

[3] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *ICML*, 2019. 2, 3, 7

[4] Benoit Brummer and Christophe De Vleeschouwer. Natural image noise dataset. In *CVPR Workshops*, 2019. 1, 2, 7

[5] Sungmin Cha, Taeeon Park, and Taesup Moon. GAN2GAN: Generative noise learning for blind image denoising with single noisy images. In *ICLR*, 2021. 2

[6] Priyam Chatterjee, Neel Joshi, Sing Bing Kang, and Yasuyuki Matsushita. Noise suppression in low-light images through joint denoising and demosaicing. In *CVPR*, 2011. 2, 3

[7] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, 2018. 2, 4, 7

[8] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. NBNet: Noise basis learning for image denoising with subspace projection. In *CVPR*, 2021. 1, 2

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007. 2, 7

[10] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE TIP*, 17(10):1737–1754, 2008. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[12] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 2, 7

[13] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 2, 7

[14] Zhiwei Hong, Xiaocheng Fan, Tao Jiang, and Jianxing Feng. End-to-end unpaired image denoising with conditional adversarial networks. In *AAAI*, 2020. 2

[15] David Honzátko, Siavash A Bigdeli, Engin Türetken, and L Andrea Dunbar. Efficient blind-spot neural network architecture for image denoising. In *SDS*, 2020. 2

[16] Xiaowan Hu, Ruijun Ma, Zhihong Liu, Yuanhao Cai, Xiaole Zhao, Yulun Zhang, and Haoqian Wang. Pseudo 3D auto-correlation network for real image denoising. In *CVPR*, 2021. 1, 2

[17] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2Neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021. 2, 3

[18] Samuel Hurault, Thibaud Ehret, and Pablo Arias. EPLL: an image denoising method using a Gaussian mixture model learned on a large set of patches. *IPOL*, 8:465–489, 2018. 2

[19] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2N: Practical generative noise modeling for real-world denoising. In *ICCV*, 2021. 1, 2, 4, 7, 8

[20] Qiyu Jin, Gabriele Facciolo, and Jean-Michel Morel. A review of an old dilemma: Demosaicking first, or denoising first? In *CVPR Workshops*, 2020. 2, 3

[21] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020. 1, 2, 7

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[23] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void-learning denoising from single noisy images. In *CVPR*, 2019. 2, 3, 4, 7

[24] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *NeurIPS*, 2019. 2, 3

[25] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018. 2, 3

[26] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, 2017. 3, 7

[27] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level Wavelet-CNN for image restoration. In *CVPR Workshops*, 2018. 7

[28] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, 2016. 1, 2

[29] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to denoise from unpaired noisy data. In *CVPR*, 2020. 3

[30] Alan V Oppenheim, Alan S Willsky, Syed Hamid Nawab, Gloria Mata Hernández, et al. *Signals & systems*. Pearson Educación, 1997. 4

[31] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising. In *CVPR*, 2021. 2, 3, 7, 8

[32] Sung Hee Park, Hyung Suk Kim, Steven Lansel, Manu Parmar, and Brian A Wandell. A case for denoising before demosaicking color filter array data. In *ACSSC*, 2009. 2, 3

[33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Workshops*, 2017. 6

[34] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 1, 6, 8

[35] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 3

[36] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. MemNet: A persistent memory network for image restoration. In *ICCV*, 2017. 1, 2

[37] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *ECCV*, 2020. 2, 3, 4, 6, 7, 8

[38] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2Same: Optimizing a self-supervised bound for image denoising. In *NeurIPS*, 2020. 2, 3

[39] 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. 1, 2, 3, 7, 8

[40] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up CNN for image denoising. In *CVPR Workshops*, 2019. 1, 2, 7, 8

[41] Zongsheng Yue, Hongwei Yong, Qian Zhao, Lei Zhang, and Deyu Meng. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019. 1, 2, 7

[42] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020. 1, 2, 7

[43] 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. 1, 2, 7, 8

[44] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE TIP*, 27(9):4608–4622, 2018. 1, 2

[45] 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. 2, 3, 4, 5, 7, 8