Unsupervised Boosting of Supervised Denoising Networks for Realistic Noise Removal

Yuan Huang, Jun Xu, Li Liu, Fan Zhu, Ling Shao, Xingsong Hou, Wangmeng Zuo

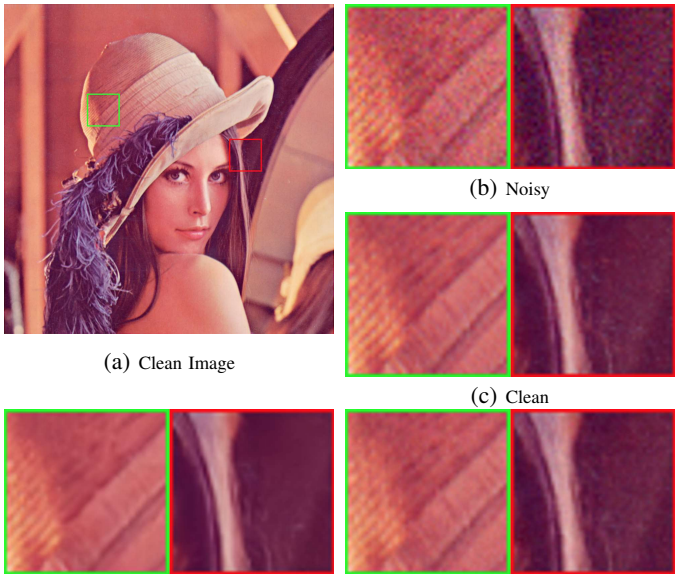
***Abstract*—Recent deep image denoising networks highly rely on pairs of clean and noisy images to learn image priors and noise statistics. However, the paired clean and noisy images cannot be easily obtained in real-world scenarios. Supervised image denoising networks deploying synthetic clean-noisy pairs suffer from a domain shift problem between real-world and synthesized noisy images, and fail to achieve satisfactory performance on realistic noise removal. In this paper, to facilitate the supervised image denoising networks for real-world scenarios, we introduce a "Noisy-As-Clean" (NAC) strategy to train supervised denoising networks in an unsupervised manner, i.e., using unpaired noisy images. To boost the denoising performance, we propose a Prediction Augmentation (PA) strategy in the inference stage. Extensive experiments demonstrate that, with our NAC and PA strategies, our unsupervised networks achieve better performance than previous leading unsupervised networks, and is competitive with state-of-the-art supervised ones, on synthetic and real-world noisy image denoising. Comprehensive ablation studies also validate the effectiveness of our unsupervised NAC and PA strategies. The code will be publicly released.**

***Index Terms*—Image denoising, unsupervised learning, deep convolutional neural network.**

I. Introduction

IMAGE denoising is a fundamental image processing problem and plays a pre-requisite role in diverse multimedia tasks such as image registration [25], surveillance [52], [71], image compression [19], and other image restoration problems [48]–[50], *etc.* It recovers the clean image $\mathbf{x}$ from its noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{n}$ corrupted by the noise $\mathbf{n}$. One assumption on noise corruption $\mathbf{n}$ is the additive white Gaussian noise (AWGN), which is widely studied by previous image denoising methods [10]–[13], [15], [17], [18], [20]–[22], [29]–[31], [34], [35], [37], [39], [40], [42], [45], [47], [51], [53]–[57], [60], [63]–[67], [69], [73]. With plenty of paired noisy and clean images, deep networks [10], [12], [34], [35], [40], [42], [47], [53], [60], [69] are achieving significant progress on image denoising.

Most of previous denoising networks are trained with noisy-clean image pairs and noisy-noisy image pairs, or unpaired noisy images. With large amounts of noisy-clean image pairs, supervised denoising networks are learned with promising performance on synthetic additive white Gaussian noise (AWGN) [53], [69]. Several unsupervised denoising networks [36] trained with noisy-noisy image pairs, evading the need on clean ground truth images. However, these networks still require two independently sampled images on the same scene [36], which cannot be easily obtained in most real-world scenarios. Recently, several unsupervised denoising networks are trained with unpaired noisy images [9], [33], [44], [58], [61], assuming that either image-specific network structures can resonate different degradation [58] or the noise are signal independent in real photographs [9], [33], [61]. But training these networks often requires fine-tuning a set of hyper-parameters for every image [58] or the independence of captured image pixels [9], [33]. Different from [58], the work of [61] is a more stable solution for unsupervised image denoising, but not efficient enough in practice. Despite their promising performance, existing denoising networks [9], [33], [34], [36], [40], [69] are mainly designed for signal independent noise (e.g., AWGN), and cannot guarantee to perform well on realistic noise, which is signal-dependent. To alleviate this problem, several supervised denoising networks [10], [26] simulate the realistic noise either in mixed Gaussian and Poisson distribution [23], or by data-driven generative models [5],



(a) Clean Image (b) Noisy (c) Clean (d) CDnCNN [69]: 35.19dB/0.8881 (e) CDnCNN [69]+NACPA: **36.43**dB/**0.9151**

Fig. 1: Denoising results of CDnCNN [69] (c) and CDnCNN network trained with our unsupervised NACPA scheme (d) on the image *Lena* corrupted by AWGN noise ($\sigma = 10$). Best results of PSNR and SSIM [59] are highlighted in **bold**.

Yuan Huang and Xingsong Hou are with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. Jun Xu is with College of Computer Science, Nankai University, Tianjin, China. Li Liu, Fan Zhu and Ling Shao are with Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE. Wangmeng Zuo is with School of Computer Science and Technology, Harbin Institute of Technology. Jun Xu (nankaimathxujun@gmail.com) and Xingsong Hou are the corresponding authors. The first two authors contribute equally.

[14]. The training of these networks usually need ground-truth clean images [1], [6]. However, the ground-truth images of real-world noisy images can hardly be accurately collected in outdoor environments, since they need hundreds of captures on the same scene under controlled environment [46], [62] or exhaustive manipulations of the captured image pairs [1], [6].

In this work, to utilize the supervised networks for realistic noise removal, we introduce an unsupervised "Noisy-As-Clean" (NAC) strategy to train these supervised networks in an unsupervised manner. Our NAC strategy evades the requirements of fine-grained collected ground-truth clean images in the training and inference stages of supervised image denoising networks. Specifically, in our NAC strategy, we directly take the real-world noisy images as the targets, which previously are usually clean ground truth images, while synthesizing the corresponding noisy image by adding to the noisy images similar noise again. To boost our unsupervised denoising networks, we propose a Prediction Augmentation (PA) strategy in the inference stage, while previous methods perform data augmentation in the training stage. Combining our NAC and PA strategies, we propose an unsupervised training scheme (we call it NACPA) to boost previous supervised image denoising networks on realistic noise removal. To illustrate the effectiveness of our NACPA scheme, we compare the denoised images of CDnCNN [69] and the CDnCNN trained with our NACPA scheme on the image "Lena" in Figure 4. We observe that with our unsupervised NACPA scheme boosts the performance of CDnCNN on the results of visual quality, PSNR, and SSIM [59]. Extensive experiments on benchmark datasets demonstrate that several representative supervised denoising networks trained by our NACPA scheme achieve similar or even better performance than the original supervised networks, on removing "weak" synthetic noise and realistic noise. Comprehensive ablation studies validate the effectiveness of our strategies.

In summary, our contributions are mainly three-fold:

- **We introduce a "Noisy-As-Clean" (NAC) strategy to evade the requirements on ground truth clean images in training supervised denoising networks**. In the NAC strategy, we replace the training targets from ground truth clean images to the synthetic/captured noisy images (synthetic/real-world), while synthesizing the corresponding noisy images by adding to the noisy images similar noise degradation. Our NAC strategy transforms the supervised denoising networks into unsupervised ones by only changing their training inputs and targets.
- **We propose a Prediction Augmentation (PA) strategy to boost the performance of unsupervised denoising networks trained by our NAC**. Since most supervised denoising networks employ data augmentation techniques during training, our PA strategy can be seamlessly embedded into these networks to improve their performance on image denoising.
- **With our NACPA scheme, previous supervised denoising networks achieve better performance by being trained in an unsupervised manner**. Experiments on benchmark datasets show that the unsupervised networks trained with our NACPA scheme achieve competitive or even better performance on synthetic and realistic noise removal over state-of-the-art unsupervised and supervised denoising networks.

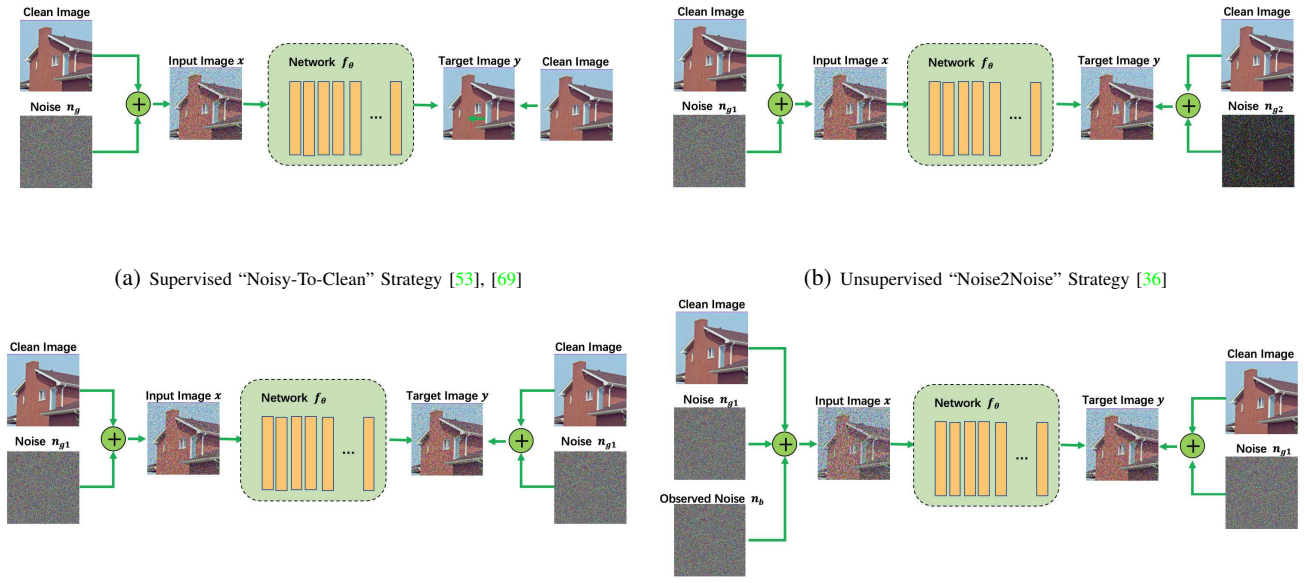
The rests of this paper are organized as follows. In §II, we introduce the related work. In §III, we present our unsupervised NACPA training scheme for boosting supervised networks on image denoising. Extensive experiments are conducted in §IV to evaluate the effectiveness of our strategies on synthetic and real photograph denoising datasets. The conclusion is given in §V.

II. Related Work

A. Supervised denoising networks

The work of [12], [34], [40], [42], [47], [53], [60], [69], [70] employ noisy and clean image pairs as the training data, as plotted in Figure 2a. The differences among these networks are mainly the network architectures they employed. Early work [12], [60] applied multi-layer perception (MLP) for deep learning based image denoising. The layers of MLP are usually less than 10, and the famous "gradient vanishing problem" [27] largely restrict their depth. In order to obtain deeper convolution networks for better denoising performance, [42] utilizes skip connections between multiple convolution layers and deconvolution layers. The authors of [53] introduced a memory block containing a recursive unit and a gate unit to build consistent and adaptive connection between different layers, achieving better performance. Inspired by the residual network [27], the work of [69] learned to predict the noise map instead of the clean images. Later, Zhang *et al.* [70] introduced a more flexible and faster solution by using an estimated noise map and the down-sampling of the noisy image as input. [34] further exploited the non-local self similarity priors of natural images for training supervised denoising networks. This is implemented by k-Nearest Neighbors (kNN) for feature matching, which is non-differentiable during training. To alleviate this problem, [47] introduced a neural nearest neighbors block, a continuous extension of discrete kNN, to perform differentiable feature matching during training. Different from [34], [47], [40] implemented the non-local matching by using a recurrent neural network, allowing efficient propagating of the correlated features though the recurrent states.

Recently, supervised denoising networks tackle the realistic noise removal with the release of benchmark datasets [1], [6], [62] on real-world noisy images and corresponding clean ground-truth images. The CBDNet [26] trains a noise estimation sub-network to obtain the realistic noise map, and a denoising network accordingly for realistic noise removal. In [5], a generative flow network is employed to synthesis realistic noise. In [7], a feature attention module was proposed to tackle the synthesis and realistic noise. These supervised denoising networks [5], [7], [26] are effective upon the availability of the ground truth clean images to the corresponding real-world noisy images. But these clean images are difficult to collect in most real-world scenes. In summary, the training of supervised denoising networks requires plenty of noisy-clean image pairs, but collecting the ground-truth clean images are difficult in many real-world scenarios.



(a) Supervised "Noisy-To-Clean" Strategy [53], [69]

(b) Unsupervised "Noise2Noise" Strategy [36]

(c) Unsupervised "Noise2Self" Strategy [9]

(d) **Our Unsupervised "Noisy-As-Clean" Strategy**

Fig. 2: Comparison of different strategies of training deep image denoising networks. The plus sign in a green circle indicates the element-wise addition. (a) In "Noisy-To-Clean" strategy, noisy and clean image pairs are used for training supervised denoising networks; (b) The "Noise2Noise" strategy uses the noisy image pairs (the same scene but different noise) for training unsupervised denoising networks; (c) The "Noise2Self" strategy uses the single noisy image itself for training unsupervised denoising networks; (d) Our "Noisy-As-Clean" strategy uses pairs of more noisy (add a observed noise on the original noisy image) and the original noisy image for training unsupervised denoising networks.

B. Unsupervised denoising networks

Due to the aforementioned limitations, researchers have began to train unsupervised denoising networks with noisy image pairs, as plotted in Figure 2b. In [36], under the assumption of AWGN, pairs of noisy images on the same scene were generated independently as the training data. By being trained with $300,000$ iterations on BSD300 [8], the Noise2Noise network is capable of denoising diverse synthesis noise. However, capturing two independent images of the same scene is difficult for real-world outdoor scenes and medical scenes. The work of Deep Image Prior [58] learned an image-specific denoising network using multi-channels of random noise as input and the test noisy image as target. But this method perform the inference by early-stopping the training. Besides, it should assign suitable network architectures and training settings for different test images. The work of [44] trains the denoising network using unpaired noisy images. This work requires to know the noise statistics, e.g., additive white Gaussian noise or multiplicative Bernoulli noise, from the degraded image. However, this is impractical in complex real-world scenarios since realistic noise is hard to be modeled [1], [6], [46], [64]. As illustrated in Figure 2c, [9], [33] are two self-supervised denoising network trained by using the noisy images as both inputs and targets. While self-supervision is a promising way for unsupervised denoising, both [9] and [33] can only achieve comparable performance with [36].

In unsupervised denoising networks for realistic noise removal, [14] restore real-world noisy images by a two-stage framework: firstly training a generative adversarial network [24] to generate noise samples similar to the test realistic noise, and then train the denoising network with the clean image to form the noisy and clean image pairs. But [14] is effective on the additive white noise, which is not the case in most real photograph.

III. Proposed Unsupervised Boosting Strategy

In this section, we introduce the proposed "Noisy-As-Clean" (NAC) and Prediction Augmentation (PA) strategies for training unsupervised image denoising networks. The combination of NAC and PA strategies (we call it the NACPA scheme) are employed in the training stage and inference stage, respectively. We first present the NAC strategy in §III-A. Then we present the PA strategy in §III-B. Finally, in §III-C, we describe the application of our unsupervised NACPA scheme onto boosting representative supervised denoising networks by been trained with only noisy images.

A. Proposed "Noisy-As-Clean" Strategy

Due to the unavailability of ground truth clean images in many real-world imaging scenarios, we propose to train unsupervised denoising networks by regarding the "Noisy" images directly "As" the "Clean" targets. The proposed "Noisy-As-Clean" strategy trains denoising networks by taking noisy images as targets, and generating more noisy images, based on the noisy images, as inputs. The more noisy images are generated by adding to the "target" noisy images the simulated noise similar to the corrupted noise on that noisy image. As illustrated in Figure 2, our NAC strategy can be easily utilized

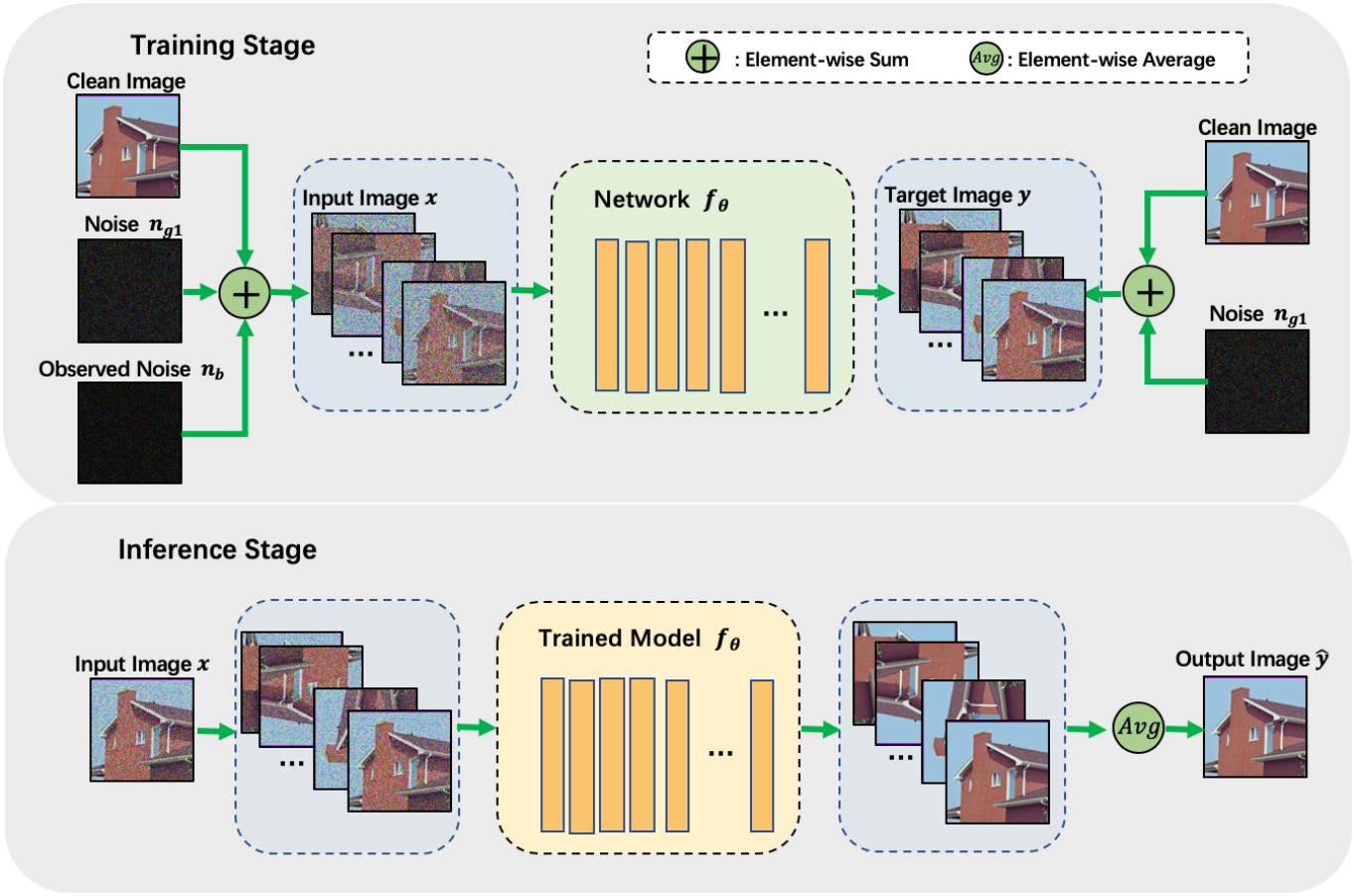


Fig. 3: **Pipeline of the proposed NACPA scheme** for supervised image denoising networks. The NACPA scheme contains two parts: the "Noisy-As-Clean" (NAC) strategy is applied in the training stage (upper), while the Prediction Augmentation (PA) strategy is performed in the inference stage (lower) of the unsupervised denoising networks.

by mangy supervised denoising networks to evade the ground truth clean images. As will be shown in the experimental section §IV, transforming these supervised denoising networks into unsupervised ones by our NAC degrades little their performance on "weak" noise. However, by avoiding ground truth images, these unsupervised denoising networks can be more adaptive to real-world scenarios.

Previous supervised denoising networks [34], [40], [69] learn a mapping function f_θ with a set of parameters θ by taking the noisy images $\{\mathbf{y}_i\}_{i=1}^N$ as inputs, and targeting at corresponding clean images $\{\mathbf{x}_i\}_{i=1}^N$, with a loss function $\mathcal{L}$:

$$\arg\min_\theta \sum_{i=1}^{N} \mathcal{L}(f_\theta(\mathbf{y}_i), \mathbf{x}_i). \tag{1}$$

However, the "noise-free" ground truth images are usually difficult to collect in real-world scenarios. To ease this problem, it is more feasible to train an unsupervised denoising network without resorting to $\mathbf{x}_i$. As studied in [9], [33], [36], [61], noisy images can be directly taken as targets for network training. Noise2Noise [36] trains the denoising network with pairs of noisy images corrupted by independent noise:

$$\mathbf{y}_{i1} = \mathbf{x}_i + \mathbf{n}_1,\ \mathbf{y}_{i2} = \mathbf{x}_i + \mathbf{n}_2, \tag{2}$$

where $\mathbf{x}_i$ is the i-th clean image, $\mathbf{n}_1$ and $\mathbf{n}_2$ are two independent AWGN noise, $\mathbf{y}_{i1}$ and $\mathbf{y}_{i2}$ are the noisy image pairs used as training target and input, respectively. But it is hard to guarantee that the two independent samples share the same ground truth $\mathbf{x}_i$, especially for scenes that have outdoor moving objects. A more practical alternative is to capture the scene only once for real photographs.

In our NAC, the noisy images $\mathbf{y}_{i1}$ are taken as the training inputs, while a more noisy image $\mathbf{y}_{i2}$ is taken as the input. This is formulated as follows:

$$\mathbf{y}_{i1} = \mathbf{x}_i + \mathbf{n}_1,\ \mathbf{y}_{i2} = \mathbf{y}_{i1} + \mathbf{n}_2, \tag{3}$$

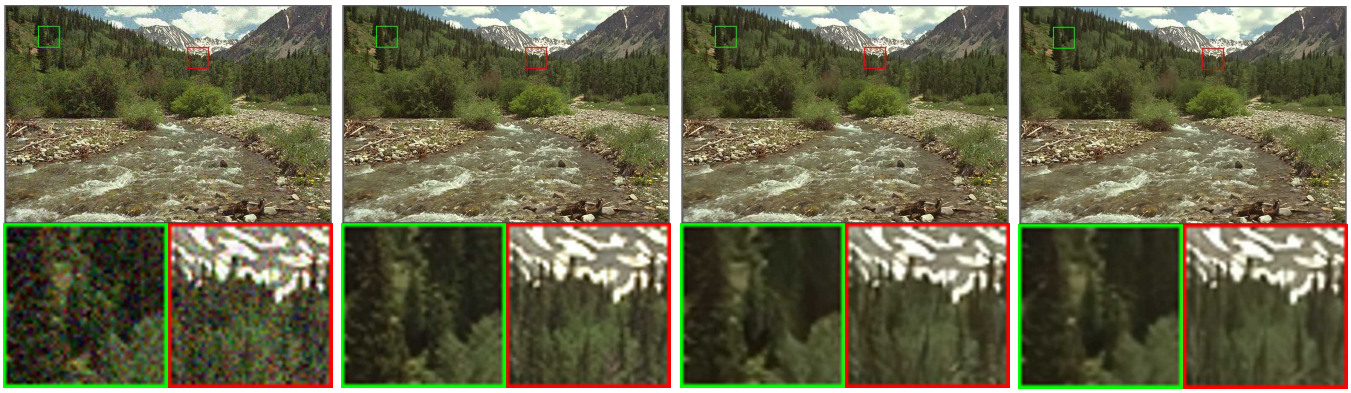
where the noise $\mathbf{n}_2$ is very close to $\mathbf{n}_1$, $\mathbf{n}_2 \approx \mathbf{n}_1$. Here, we do not require that the noise $\mathbf{n}_1, \mathbf{n}_2$ to be zero mean. Since the training target $\mathbf{y}_{i1}$ is itself noisy images, our NAC strategy is more suitable for "weak" noise. The original supervised denoising network f_θ can be transferred into an unsupervised one by our NAC as follows:

$$\mathbf{y}_{i1} = f_\theta(\mathbf{y}_{i2}). \tag{4}$$

With (3), (4) can be written as:

$$\mathbf{y}_{i1} = f_\theta(\mathbf{y}_{i1} + \mathbf{n}_2). \tag{5}$$

This indicates that after training, the network f_θ has the capacity to remove the noise $\mathbf{n}_2$ from $\mathbf{y}_{i1}$. By assuming that



(a) Noisy Image (b) Clean Image (c) CDnCNN [69]: 29.50dB/0.8912 (d) CDnCNN [69]+NACPA: **32.61**dB/**0.9365**

Fig. 4: Denoising results of CDnCNN [69] network and CDnCNN with our NACPA scheme on the image "*kodim13*" (from the *Koda24* [2] dataset) corrupted by AWGN noise ($\sigma = 20$). Best results of PSNR and SSIM [59] are highlighted in **bold**.

$\mathbf{n}_2$ and $\mathbf{n}_1$ are weak and similar, the network f_θ is feasible to remove the noise $\mathbf{n}_1$ from $\mathbf{y}_{i1}$, i.e., $\hat{\mathbf{x}}_i = f_\theta(\mathbf{x}_i + \mathbf{n}_1) \approx \mathbf{x}_i$. However, in most situation, such assumption can be hardly satisfied. In §IV, we will verify the effectiveness of our NAC strategy on synthetic and realistic noise removal.

With applying our NAC at training stage, previous supervised denoising networks can be directly trained with real-world noisy images [34], [40], [69]. Though the realistic noise $\mathbf{n}_1$ is complex [23], [41], it can be modeled by mixed Gaussian and Poisson noise [23], [41], or learned by variational generative models [68]. The former solution acquires to process the RAW image by in-camera processing pipeline (ISP) [26], as well as corresponding inverse operators to simulate the process that realistic noise are generated in sensors. With the real-world noisy images and synthetic more noisy images corrupted again by similar noise, the unsupervised denoising networks trained with our NAC strategy are very effective on real-world image des, as will be demonstrated in §IV. Note that previous unsupervised denoising networks [9], [33], [36], [58] require the noise to be zero-mean, while our NAC strategy can learn to remove non-zero mean noise.

B. Proposed Prediction Augmentation Strategy

To further boost the performance of unsupervised networks, we propose a Prediction Augmentation (PA) strategy, which augments and fuses the inference predictions for better denoising performance. Note that our PA strategy is different from the widely used data augmentation (DA) techniques [28]. DA aims to boost training performance by providing more training data to evade over-fitting and reduce the domain gap between the training and test data. Built upon DA, our PA provides more inference predictions for each augmented training input, and is thus able to fuse the complementary denoising results for better performance.

Similar strategy is also studied by the famous Deep Image Prior (DIP) [58] in an online *learning-while-inference* manner, in which the *inference* is performed via a weighted average of inferences in thousands *learning* iterations. This online inference strategy indeed improves the final prediction of DIP, but is time-consuming and infeasible for offline training networks. With DA, our PA strategy naturally obtains different inference predictions of the denoised images. During the training, each training input is rotated or flipped for data augmentation. The same noisy pixels (i.e., pixels in the same positions of original image) of different augmented images are restored using different receptive fields in all augmented images, and are thus averaged in a complementary manner to produce better results.

For the PA strategy in inference stage, two ways of augmentation can be explored. The Strategy 1 performs data augmentation during model training, to augmented the noisy input image with rotation and flip into 8 images. Then the augmented images go through the denoising model and are averaged to output the mean image. This strategy is also used in image super-resolution community [38], [72] in an ensemble manner. Different from the Strategy 1, the Strategy 2 is to complement the clean image or an estimated clean image by 8 augmented images, and then add them with the same observed noise. We evaluate the above two trategies based on the CDnCNN [69] and CDnCNN with our NAC strategy in the same training settings, the results are provided in Table I. From Table I, both strategies show clear improvements over the networks only with the original inference results. The Strategy 2 demonstrates higher improvements, since the noise is able to sample different pixels by adding to the augmentation of the clean images. While CDnCNN [69] with NAC and the first inference strategy can still achieve better results than the original supervised trained CDnCNN [69] at $\sigma = 10, 15, 20$. Since the Strategy 2 achieves better performance, we take the Strategy 2 as our PA strategy studied in the experimental section.

As illustrated in Figure 5, built upon general data augmentation techniques [28], our PA strategy augments the inference predictions by rotating 90°, 180°, 270° and/or flipping vertically/horizontally. With the noisy image $\mathbf{y}_i = \mathbf{x}_i^a + \mathbf{n}_2$ ($\mathbf{n}_2$ is the observed noise, $\mathbf{x}_i^a$ is an approximation of $\mathbf{x}_i$), we augment the $\mathbf{x}_i^a$ to $\{\mathbf{x}_{i1}^a, ..., \mathbf{x}_{i8}^a\}$. Then we generate the augmentation of the inference images for input as (6):

$$\mathbf{y}_{i\_aug\_1}^a = \mathbf{x}_{i1}^a + \mathbf{n}_2, ..., \mathbf{y}_{i\_aug\_8}^a = \mathbf{x}_{i8}^a + \mathbf{n}_2. \quad (6)$$

The outputs of these augmented inference images are rotated

TABLE I: The average PSNR (dB) and SSIM [59] results of blind denoising network CDnCNN [69] and CDnCNN [69] with NAC applied with two different inference strategies for color image denoising, tested on *Koda24* [2] dataset under noise levels $\sigma = \{5, 10, 15, 20\}$. Higher results are highlighted in **bold**. The Strategy 1 directly augments the noisy image for complementary inferences. The Strategy 2 firstly separates the noisy image into a denoised image and the roughly removed noise, and then augmented the denoised image for complementary inferences.

Test Set	Noise Level / Metric	$\sigma = 5$ PSNR↑	$\sigma = 5$ SSIM↑	$\sigma = 10$ PSNR↑	$\sigma = 10$ SSIM↑	$\sigma = 15$ PSNR↑	$\sigma = 15$ SSIM↑	$\sigma = 20$ PSNR↑	$\sigma = 20$ SSIM↑
Koda24	**CDnCNN**	40.60	0.9729	36.78	0.9469	34.68	0.9223	33.26	0.9013
	CDnCNN+Strategy 1	40.70	0.9734	36.98	0.9489	34.85	0.9250	33.42	0.9040
	CDnCNN+Strategy 2	**43.16**	**0.9812**	**39.32**	**0.9633**	**37.06**	**0.9441**	**35.71**	**0.9311**
	CDnCNN+NAC	40.43	0.9725	36.75	0.9467	34.58	0.9193	33.10	0.8971
	CDnCNN+NAC+Strategy 1	40.53	0.9729	36.88	0.9480	34.77	0.9223	33.31	0.9016
	CDnCNN+NAC+Strategy 2	**42.96**	**0.9812**	**39.24**	**0.9632**	**36.72**	**0.9390**	**35.66**	**0.9306**

and/or flipped back to be averaged as the final inference prediction, as follows:

$$\hat{\mathbf{y}}_{final} = \frac{1}{N}\sum_{n=1}^{N} f_\theta(\mathbf{y}^a_{i\_aug\_n}). \tag{7}$$

Our PA strategy improves the prediction performance for image denoising in the NACPA scheme and can also be applied to many other supervised trained image denoising networks, as will be shown in the experimental section. Please refer to Figure VII for a direct comparisons of the networks with or without our PA inference strategy.

C. Training Unsupervised Denoising Networks

Our NACPA scheme can be seamlessly embedded into existing supervised denoising networks. During the inference stage, we only change the input-output of these networks from noisy-clean image pairs to noisier-noisy ones. During training, the ℓ_2 loss function is used to optimize the unsupervised networks:

$$\mathcal{L} = \sum_{i=1}^{N} ||f_\theta(\mathbf{y}_{i2}) - \mathbf{y}_{i1}||_2^2. \tag{8}$$

More implementation details are described in the experimental section to deal with synthetic and realistic noise removal.

Algorithm 1 Unsupervised boosting of supervised denoising networks by our NACPA scheme

Generate noisy image pairs for training: with noisy image $\mathbf{y}_i$ and the observed noise $\mathbf{n}_2$, $\mathbf{y}_{i2} = \mathbf{y}_{i1} + \mathbf{n}_2$, to form noisy pairs as $\{\mathbf{y}_{i2}, \mathbf{y}_{i1}\}_{i=1}^{N}$, N indicates the number of the training images
Learning: noisy image pairs as inputs and targets $\{\mathbf{y}_{i2}, \mathbf{y}_{i1}\}_{i=1}^{N}$
$\theta \leftarrow$ Initialize network $\mathbf{f}_\theta$ parameters
repeat
  Network training: using noisy image pairs for training, optimized the parameters θ in the denoising models $\mathbf{f}_\theta$ according to the loss function
until Convergence
return $\mathbf{f}_\theta$

Inference: Input noisy image $\mathbf{y}_i = \mathbf{x}_i^a + \mathbf{n}_2^a$, where $\mathbf{n}_2$ is a observed noise of the noisy input image
Generate multiple augmentation: rotate and flip $\mathbf{x}^a$ to generate$(\mathbf{x}^a_{i1}, ..., \mathbf{x}^a_{i8})$, then add with $\mathbf{n}_2$:
$\mathbf{y}^a_{i\_aug\_1} = \mathbf{x}^a_{i1} + \mathbf{n}_2, ..., \mathbf{y}^a_{i\_aug\_8} = \mathbf{x}^a_{i8} + \mathbf{n}_2$
Testing thought the denoising model: take $(\mathbf{y}^a_{i\_aug\_1}, ..., \mathbf{y}^a_{i\_aug\_8})$ though the trained network $\mathbf{f}_\theta$ and get the correspondence output $(\hat{\mathbf{y}}^a_{i1}, ..., \hat{\mathbf{y}}^a_{i8})$
De-augmentation: inverse the rotation and flipped for $(\hat{\mathbf{y}}_{i1}, ..., \hat{\mathbf{y}}_{i8})$
Final output: compute the average of the $(\hat{\mathbf{y}}_{i1}, ..., \hat{\mathbf{y}}_{i8})$ and get the final inference result $\hat{\mathbf{y}}_{final}$

IV. Experiments

In this section, extensive experiments on diverse benchmark datasets are performed to evaluate the effectiveness of the proposed NACPA scheme on both synthetic and realistic noise removal. By our unsupervised NACPA scheme, we boost several supervised denoising networks, and compare them with the original supervised denoising networks. We also conduct comprehensive ablation studies to gain more insights of our NACPA scheme.

A. Synthetic Noise Removal

Implementation details. In training stage, we apply our NAC strategy to several representative supervised denoising networks, to evaluate its effectiveness on these networks, and compare their performance with the original supervised trained denoising models. For the color image denoising, the employed networks are CDnCNN [69], MemNet [53], RedNet [42], and a simplified version of RCAN [72] (originally used in image super resolution) with 10 resblocks in the network and represents as RCAN*. While for gray-scale image denoising, we use CDnCNN [69], MemNet [53] and RCAN* [72] in our experiments. For fair comparison, the network architectures and training settings of these networks are kept unchanged.

In inference stage, our PA strategy produces 8 augmentations of rotating 90°, 180°, 270° and/or flipping vertically/horizontally of the original noisy image, which are added with the observed noise as the input test images. Two widely used evaluation metrics are the Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index (SSIM) [59]. All the networks are implemented in PyTorch [3] and trained on a single NVIDIA V100 GPU.

Datasets. The training is performed individually on the CBSD400 [43] (400 RGB images) for color image denoising,

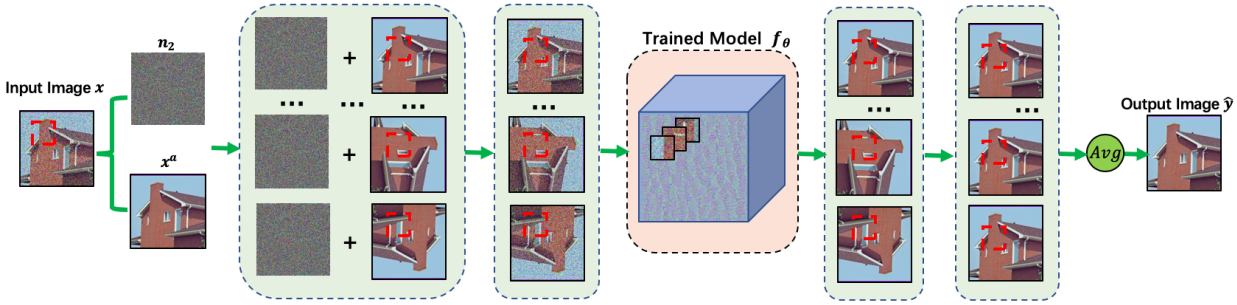


Fig. 5: **Proposed Prediction Augmentation (PA) strategy** augments the inference prediction images by rotating 90°, 180°, 270° and/or flipping vertically/horizontally.

and on BSD400 [43] (400 gray-scale images) for gray-scale image denoising. Each image is used to generate the synthetic noisier-noisy image pairs for training by our NAC strategy. Here we test additive white Gaussian noise (AWGN) with different standard deviations (stds). The testing is performed on CBSD68 [43], Kodak24 [2] and Urban100 [32] datasets for color image denoising. The denoising model for gray-scale were tested on BSD68 [43], gray-scale version of Kodak24 [2] and Urban100 [32] datasets.

Results. We test on AWGN noise with stds $\sigma = 5, 10, 15, 20$ to evaluate the effectiveness of our NACPA scheme applied on different denoising networks. For denoising RGB images, CBSD400 as the training set and tested the denoising model on CBSD68 [43], Koda24 [2] and Urban100 [32] datasets. The original denoising models were trained supervised using the synthesis noise, while the denoising model applied with NACPA scheme trained unsupervised. The results are listed in Table II. We observe that with our NACPA scheme, the unsupervised trained networks exceed the original supervised trained models in the cases of low noise levels $\sigma = 5, 10, 15$. However, the improvements decline as the noise becomes stronger. The reason is that with the increasing of the noise levels, the input noisy images are doubly degraded with the noise as the training inputs, making the latent clean image is difficult to be recovered from the noisy image.

For gray-scale image denoising, the BSD400 dataset [43] is utilized to train the denoising networks. The test results on BSD68 [43] and the gray-scale version of Urban100 [32] and Koda24 [2] datasets are listed in Table III. We observe that the supervised denoising networks are clearly boosted by our unsupervised NACPA strategies. Similar phenomenon appears in gray-scale image denoising, that the denoising networks with the NACPA scheme achieved a superior performance at the most noise level compared with the original supervised trained denoising models. The performance becomes comparable when σ=20. More visual comparison and qualitative evaluation comparison of the denoising results are presented in the supplementary file.

Speed. Our NAC strategy does not affect the training speed. But in testing, our PA strategy requires multiple inferences, and hence the inference time of our unsupervised boosted networks will be larger than the original supervised denoising ones. The running time of the comparison methods are listed in Table IV. One can see that the unsupervised networks require more time to obtain better denoising performance.

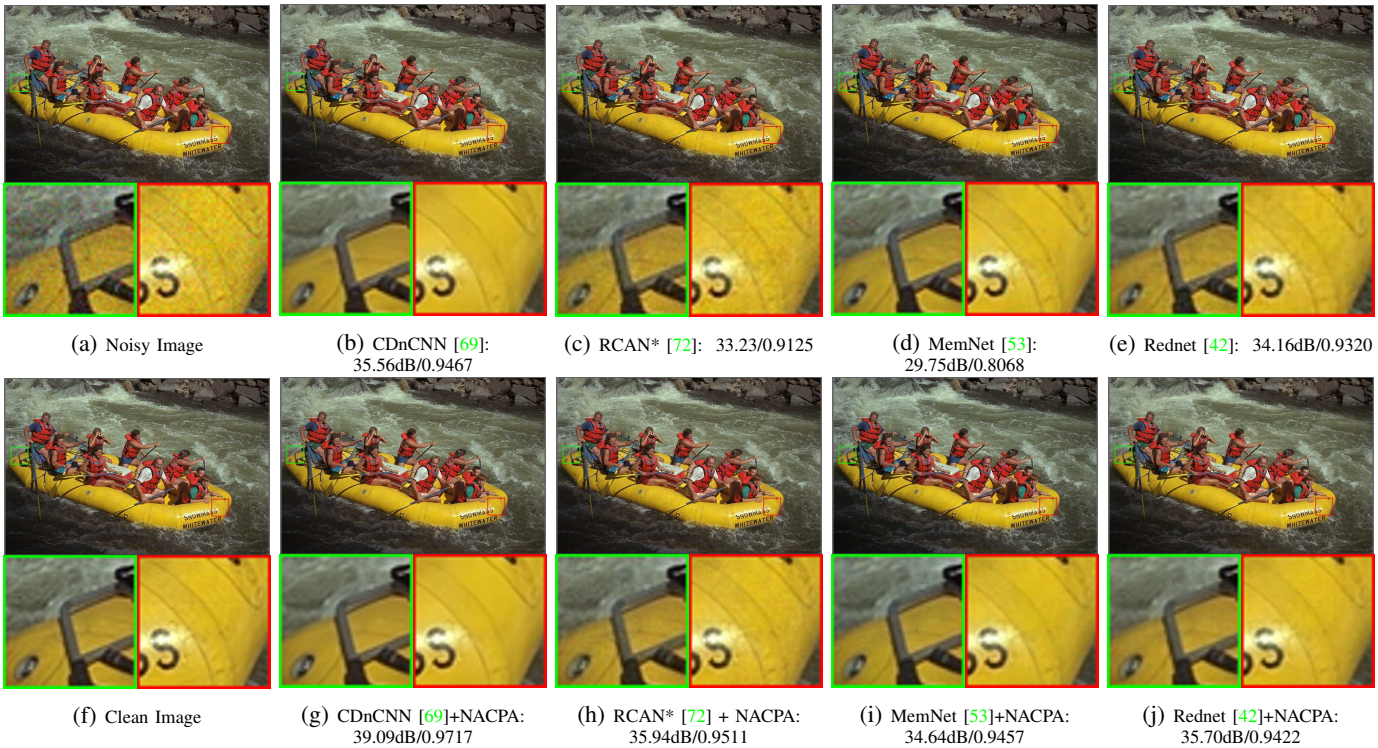
B. Realistic Noise Removal

Implementation details. For real photograph denoising, our NACPA scheme is applied on CDnCNN [69] and RCAN* [72] for realistic noise removal. To implied our NACPA scheme, a similar observed noise is needed to build the noisy image pairs for training. Here we employ the work of [68] to estimate the noise, and generate an observed noise for the real-world noisy image. For fair comparison, the network architectures and training settings of these networks are kept unchanged.

Datasets. For real photograph denoising, the training set including 5000 selected images from the SIDD small dataset [6]. The test sets including the DND [1] (1000 images), SIDD validation set [6] (1280 images) and the SIDD Benchmark test set [4] (1280 images).

Results Comparison. Here, we compare the performance of RCAN* [72] applied with NACPA scheme based unsupervised denoising with the traditional method CBM3D [16], unsupervised denoising networks [14], [36], and supervised ones such as CDnCNN [69] and RCAN* [72] (both supervised trained with SIDD small dataset [6]) for realistic noise removal. The comparisons of these methods on PSNR (dB) and SSIM [59] are listed in Table V.

In Figures 7-8, we demonstrate visual and quantitative denoised results of these methods on test images of DND [1] and SIDD [6] datasets. One can see that in the task of realistic noise removal, for DND [1] dataset, which do not provide similar training data, no "noisy-clean" image pairs like SIDD dataset [6], applied with NACPA scheme indeed boosted the original supervised trained model and acquired a 0.2dB of improvement and surpass the other state-of-the-art unsupervised denoising methods. While for testing results on SIDD validation set [6] and SIDD Benchmark set [4] the supervised trained with SIDD training set denoising model shows a higher performance. Both the CDnCNN [69] with NACPA and RCAN* [72] with NACPA scheme achieve a highest performance among other unsupervised denoising methods. Figure 8 shows a visual comparison of the denoised results from SIDD Validation set, we observe that supervised trained models have higher PSNR (dB)/SSIM, but both the CDnCNN [69] with NACPA and RCAN* [72] with NACPA deliver a more clear vision of the text in the image. This



(a) Noisy Image (b) CDnCNN [69]: 35.56dB/0.9467 (c) RCAN* [72]: 33.23/0.9125 (d) MemNet [53]: 29.75dB/0.8068 (e) Rednet [42]: 34.16dB/0.9320

(f) Clean Image (g) CDnCNN [69]+NACPA: 39.09dB/0.9717 (h) RCAN* [72] + NACPA: 35.94dB/0.9511 (i) MemNet [53]+NACPA: 34.64dB/0.9457 (j) Rednet [42]+NACPA: 35.70dB/0.9422

Fig. 6: Denoising results of a typical images ("*kodim14*") of different denoising networks with the proposed NACPA scheme on *Koda24* [2] test set corrupted by AWGN noise ($\sigma = 10$).

demonstrate the effectiveness of our unsupervised NAC and PA strategies for real photograph denoising.

C. Validation of the Proposed Method

Here, we conduct more detailed examinations and analysis of our proposed NACPA scheme on image denoising to gain more insights of our unsupervised training and inference scheme. Specifically, we assess: 1) the effectiveness of our NAC strategy; 2) applied PA strategy to other image denoising networks; 3) NACPA for blind denoising; 4) how inaccurate observed noise affect the denoising performance. All the experiments are performed by training the networks on the CBSD400 dataset and testing them on the Kodak24 [2] and CBSD68 [43] datasets.

1) Effectiveness of our NAC Strategy on image denoising. The proposed NAC strategy works as an important role to transform the supervised networks into unsupervised ones. With our NAC, previous denoising networks [69] are trained with the noisy-noisy image pairs, without the requirements of clean ground truth images. To evaluate the effectiveness of our NAC strategy, we applied it to several [69] (without the PA strategy) under the same settings. The noise levels are $\sigma = \{5, 10, 15, 20\}$. As shown in Table VI, the performance of the unsupervised denoising networks with our NAC are comparable to the original supervised denoising networks, bearing little sacrifice on denoising performance. However, our NAC strategy evades the requirements on clean ground truth images, which are hardly available in real scenarios.

2) Improvements of our PA strategy on image denoising networks. While our NAC strategy provides an unsupervised training method for image denoising networks. Our PA strategy is proposed to further improve the unsupervised denoising performance. While the Table II in §III already demonstrates that the proposed NACPA scheme can improve the qualitative performance of unsupervised image denoising networks, here we further study the contribution of our PA strategy applied on the other supervised image denoising networks. As shown in Figure VII, by applying our PA strategy to the supervised networks such as CDnCNN [69], RCAN* [72] and MemNet [53], all their PSNR and SSIM results are clearly improved, without additional training strategies. This demonstrates that our PA strategy indeed helps improving the denoising performance of the supervised denoising networks.

3) How is the effects of our NACPA scheme for blind image denoising? The above synthetic denoising experiments were performed on specific noise levels, i.e., the training and test noise levels (σ) are known and exactly the same. However, in real photograph denoising scenarios we can hardly know the noise levels. To further exploit the potential of our NACPA scheme, we trained a blind denoising network in unsupervised manner, based on CDnCNN-B [69]. In the training stage, the AWGN noise is randomly generated with $\sigma \in [0, 55]$. Then at inference stage, the network is tested on the Koda24 [2], CBSD68 [43] and Urban100 [32] datasets. As shown in Table VIII, the blindly trained network can still achieves satisfactory results on specific noise levels, even comparable with the original supervised trained denoising model.

4) How inaccurate observed noise affects the denoising performance? In the above presentation of the proposed framework for synthesis noise denoising, the observed noise were described as the noise with the same statistics, while for

TABLE II: Average PSNR (dB) and SSIM [59] results of different denoising networks applied with and without our NACPA scheme for color image denoising, tested on *Koda24* [2], *CBSD68* [43], and *Urban100* [32] datasets under AWGN with $\sigma = \{5, 10, 15, 20\}$.

Test Set	Noise Level / Metric	$\sigma = 5$ PSNR↑	$\sigma = 5$ SSIM↑	$\sigma = 10$ PSNR↑	$\sigma = 10$ SSIM↑	$\sigma = 15$ PSNR↑	$\sigma = 15$ SSIM↑	$\sigma = 20$ PSNR↑	$\sigma = 20$ SSIM↑
Koda24	**CDnCNN**	40.60	0.9729	36.78	0.9469	34.68	0.9223	33.26	0.9013
	CDnCNN+NACPA	42.96	0.9812	39.24	0.9632	36.72	0.9390	35.66	0.9306
	RCAN*	37.21	0.9420	33.72	0.8812	31.02	0.8217	29.65	0.7754
	RCAN*+NACPA	41.23	0.9778	36.56	0.9341	33.93	0.8701	31.22	0.7927
	MemNet	37.90	0.9420	32.87	0.9047	30.80	0.8840	28.95	0.8578
	MemNet+NACPA	39.65	0.9730	35.00	0.9342	32.03	0.8682	28.69	0.7995
	RED	38.74	0.9566	35.10	0.9243	33.03	0.8844	31.08	0.8346
	RED+NACPA	41.57	0.9775	36.56	0.9321	33.43	0.8754	31.51	0.8193
CBSD68	**CDnCNN**	40.19	0.9800	35.99	0.9545	33.89	0.9303	32.19	0.9060
	CDnCNN+NACPA	43.09	0.9878	38.67	0.9709	35.97	0.9472	34.78	0.9376
	RCAN*	36.97	0.9519	33.37	0.9017	30.66	0.8507	29.10	0.7943
	RCAN*+NACPA	41.91	0.9831	36.92	0.9467	33.79	0.8922	31.16	0.8270
	MemNet	38.15	0.9680	33.30	0.9062	30.88	0.8520	29.27	0.7873
	MemNet+NACPA	38.96	0.9771	35.01	0.9539	32.27	0.9154	29.18	0.8732
	RED	38.51	0.9650	34.61	0.9363	32.38	0.8980	30.46	0.8529
	RED+NACPA	41.29	0.9812	36.10	0.9414	33.00	0.8921	31.04	0.8436
Urban100	**CDnCNN**	39.43	0.9780	35.85	0.9553	33.54	0.9376	32.16	0.9223
	CDnCNN+NACPA	40.97	0.9832	37.41	0.9669	35.21	0.9479	34.20	0.9416
	RCAN*	36.85	0.9605	32.42	0.8931	29.72	0.8495	28.66	0.8063
	RCAN*+NACPA	39.33	0.9786	35.39	0.9428	32.79	0.8986	30.64	0.8491
	MemNet	37.32	0.9246	32.15	0.8991	30.57	0.8226	28.05	0.7725
	MemNet+NACPA	39.60	0.9788	34.98	0.9450	31.96	0.8966	28.86	0.8412
	RED	37.21	0.9608	33.50	0.9335	31.50	0.9008	29.59	0.8587
	RED+NACPA	40.29	0.9805	35.73	0.9444	32.46	0.8965	30.44	0.8492

TABLE III: Average PSNR (dB) and SSIM [59] results of different denoising networks applied with and without NACPA scheme for gray-scale image denoising, tested on gray-scale version of *Koda24* [2] , *BSD68* [43], gray-scale version of *Urban100* [32] datasets with AWGN levels $\sigma = \{5, 10, 15, 20\}$.

Test Set	Noise Level / Metric	$\sigma = 5$ PSNR↑	$\sigma = 5$ SSIM↑	$\sigma = 10$ PSNR↑	$\sigma = 10$ SSIM↑	$\sigma = 15$ PSNR↑	$\sigma = 15$ SSIM↑	$\sigma = 20$ PSNR↑	$\sigma = 20$ SSIM↑
Koda 24	**DnCNN**	38.73	0.9628	34.93	0.9255	32.86	0.8911	31.47	0.8610
	DnCNN+NACPA	41.01	0.9751	36.38	0.9387	33.89	0.9003	32.21	0.8950
	RCAN*	37.51	0.9504	31.78	0.8265	27.01	0.6245	24.24	0.5022
	RCAN*+NACPA	40.56	0.9724	36.15	0.9224	32.77	0.8445	30.23	0.7613
	MemNet	36.07	0.9193	32.91	0.8691	30.33	0.7923	28.71	0.7186
	MemNet+NACPA	40.12	0.9690	34.03	0.8781	31.16	0.7931	28.62	0.7007
	RED	37.91	0.9857	34.05	0.9117	31.64	0.8563	29.79	0.7940
	RED+NACPA	39.40	0.9664	35.02	0.9176	32.36	0.8597	30.90	0.8062
BSD68	**DnCNN**	38.00	0.9696	33.90	0.9316	31.72	0.8955	30.28	0.8629
	DnCNN+NACPA	39.68	0.9735	34.73	0.9271	32.01	0.8747	30.21	0.8200
	RCAN*	36.91	0.9588	31.28	0.8508	26.87	0.6782	24.22	0.5665
	RCAN*+NACPA	40.05	0.9767	35.66	0.9336	32.48	0.8682	30.07	0.7984
	MemNet	36.23	0.9583	32.35	0.9010	30.61	0.8877	28.57	0.8974
	MemNet+NACPA	37.72	0.9623	33.89	0.9058	30.95	0.8221	28.69	0.7431
	RED	37.41	0.9657	33.22	0.9201	30.76	0.8665	28.96	0.8082
	RED+NACPA	39.06	0.9713	34.37	0.9268	31.67	0.8749	30.31	0.8300
Urban100	**DnCNN**	38.53	0.9746	34.73	0.9500	32.57	0.9254	31.05	0.9029
	DnCNN+NACPA	40.33	0.9768	35.59	0.9381	32.63	0.8910	30.61	0.8413
	RCAN*	36.78	0.9626	31.24	0.8638	26.92	0.7058	24.31	0.6091
	RCAN*+NACPA	40.31	0.9800	35.62	0.9375	32.39	0.8754	30.05	0.8120
	MemNet	35.47	0.9244	32.52	0.8960	30.07	0.8442	28.45	0.7861
	MemNet+NACPA	39.84	0.9749	34.01	0.9023	30.96	0.8312	28.50	0.7616
	RED	37.48	0.9706	33.38	0.9362	30.81	0.8924	28.88	0.8411
	RED+NACPA	39.63	0.9765	34.66	0.9344	31.46	0.8835	29.82	0.8370

TABLE IV: Running time (in seconds) of different denoising networks and those trained with our NACPA scheme, on RGB images of size 256×256 and 512×512.

Image Size	CDnCNN	CDnCNN+NACPA	RCAN*	RCAN*+NACPA
256×256×3	0.0038	0.0405	0.3038	2.446
512×512×3	0.0171	0.1568	1.200	9.657

most cases it is hard to acquire the same statistics noise. To further discover the impact of inaccurate observed noise would affect the denoising performance. In Table IX and Table X, we applied the NACPA scheme to CDnCNN [69] and training for image denoising. The first column represent the specific σ that the model original trained for, while the first row represent the

TABLE V: Average PSNR (dB) and SSIM [59] results on realistic noise removal on the DND [1], SIDD Validation [6], and SIDD Benchmark [4] datasets. "S." indicates supervised models and "U." indicates unsupervised models.

Method	S./U.	**DND** [1]		**SIDD Val.** [6]		**SIDD Benchmark** [4]	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
CBM3D [16]	U.	34.51	0.8507	30.88	0.7025	25.65	0.6850
N2N [36]	U.	33.10	0.8110	24.27	0.3527	24.50	0.5050
GCBD [14]	U.	35.58	0.9217	-	-	-	-
CDnCNN [69]	S.	37.21	0.9297	36.36	0.8858	36.61	0.9270
CDnCNN [69]+NAC	U.	33.32	0.8159	27.81	0.5287	27.86	0.6660
CDnCNN [69]+NACPA	U.	36.16	0.9003	32.39	0.7273	32.42	0.8260
RCAN* [72]	S.	35.89	0.9128	34.91	0.8278	34.90	0.9000
RCAN* [72]+NAC	U.	33.62	0.8237	28.60	0.5431	28.55	0.6870
RCAN* [72]+NACPA	U.	36.10	0.9104	31.69	0.6945	31.51	0.8090

TABLE VI: Average PSNR (dB) and SSIM [59] results of color image denoising with our NAC strategy on *Koda24* [2] and *CBSD68* [43] datasets. The noise levels are $\sigma = \{5, 10, 15, 20\}$.

Test Set	Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Koda24	**CDnCNN**	40.60	0.9729	36.78	0.9469	34.68	0.9223	33.26	0.9013
	CDnCNN+NAC	40.43	0.9725	36.75	0.9467	34.58	0.9193	33.10	0.8971
CBSD68	**CDnCNN**	40.19	0.9800	35.99	0.9545	33.89	0.9292	32.19	0.9060
	CDnCNN+NAC	40.10	0.9797	35.94	0.9541	33.63	0.9136	32.10	0.9033

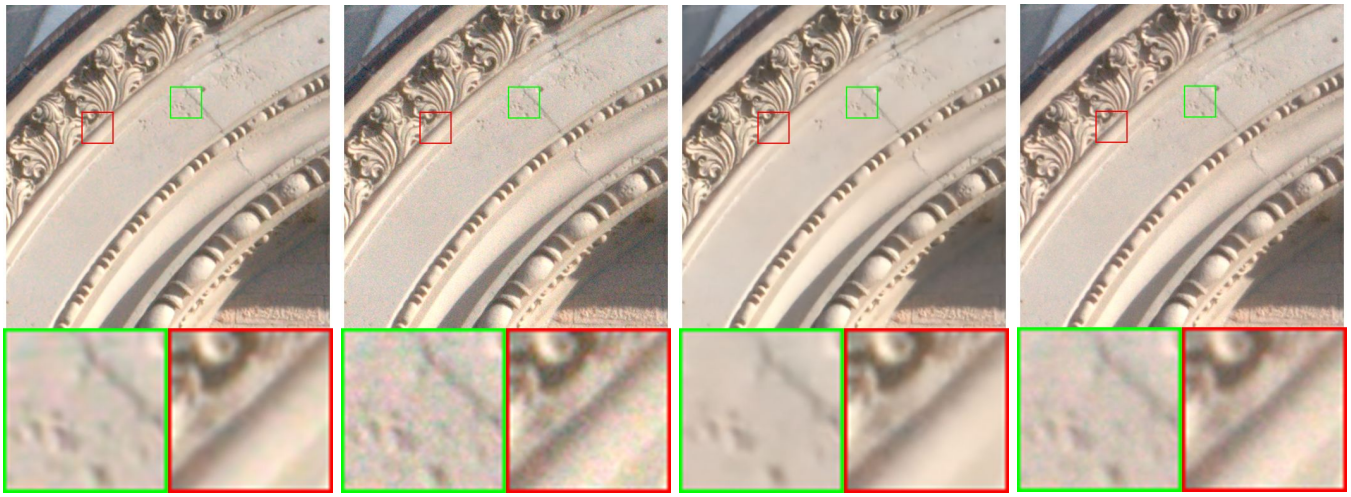
TABLE VII: Average PSNR (dB) and SSIM [59] of different supervised denoising networks with our PA strategy and tested on *Koda24* [2] and *CBSD68* [43] datasets. The noise levels are $\sigma = \{5, 10, 15, 20\}$. Higher results are highlighted in **bold**.

Test Set	Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Koda24	**CDnCNN**	40.60	0.9729	36.78	0.9469	34.68	0.9223	33.26	0.9013
	CDnCNN+ PA.	**43.16**	**0.9812**	**39.32**	**0.9633**	**37.06**	**0.9441**	**35.71**	**0.9311**
	RCAN*	37.21	0.9420	33.72	0.8812	31.02	0.8217	29.65	0.7754
	RCAN*+ PA.	**41.23**	**0.9778**	**36.56**	**0.9341**	**33.93**	**0.8701**	**31.22**	**0.7927**
	MemNet	37.90	0.9420	32.87	0.9047	30.80	0.8840	28.95	0.8578
	MemNet+ PA.	**39.80**	**0.9687**	**35.10**	**0.9361**	**33.49**	**0.8731**	**29.74**	**0.8051**
CBSD68	**CDnCNN**	40.19	0.9800	35.99	0.9545	33.89	0.9292	32.19	0.9060
	CDnCNN+ PA.	**43.19**	**0.9879**	**38.83**	**0.9715**	**36.35**	**0.9528**	**34.78**	**0.9375**
	RCAN*	36.97	0.9519	33.37	0.9017	30.66	0.8507	29.10	0.7943
	RCAN*+ PA.	**41.91**	**0.9831**	**36.92**	**0.9467**	**33.79**	**0.8922**	**31.16**	**0.8270**
	MemNet	38.15	0.9680	33.30	0.9062	30.88	0.8520	29.27	0.7873
	MemNet+ PA.	**39.73**	**0.9730**	**35.70**	**0.9539**	**33.47**	**0.9200**	**30.75**	**0.8833**

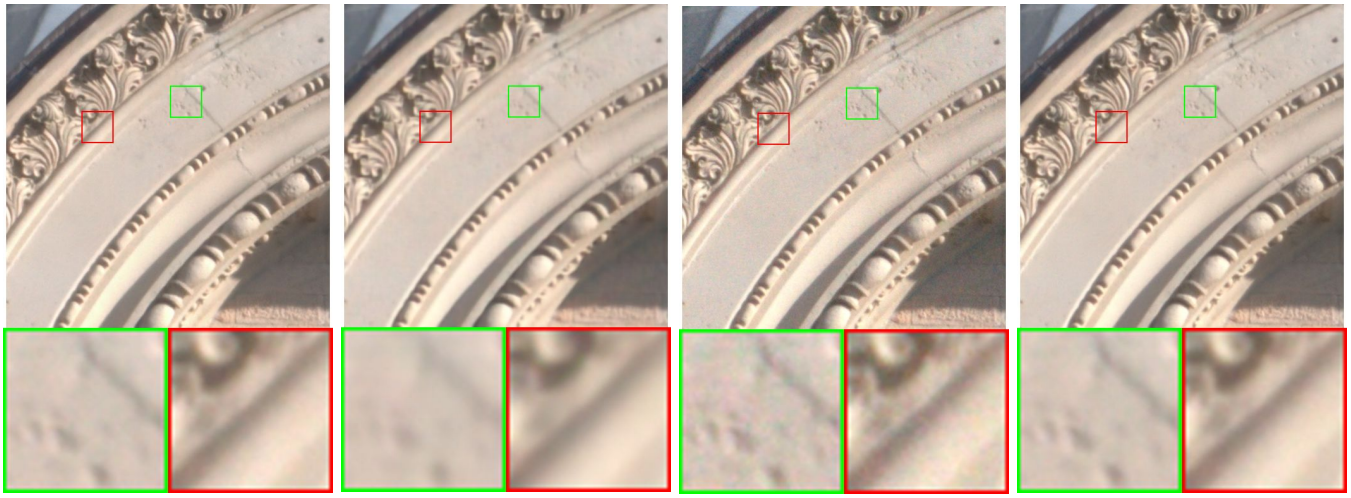
TABLE VIII: Average PSNR (dB) and SSIM [59] results of blind denoising network CDnCNN-B [69] and CDnCNN-B [69] with NACPA scheme for color image denoising, tested on *Koda24* [2], *CBSD68* [43] and *Urban100* [32] datasets. The noise levels are $\sigma = \{5, 10, 15, 20\}$. Higher results are highlighted in **bold**.

Test Set	Noise Level	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Koda24	**CDnCNN-B**	39.73	0.9686	36.38	**0.9428**	**34.41**	**0.9180**	**33.03**	**0.8951**
	CDnCNN-B+NACPA	**41.81**	**0.9746**	**36.47**	0.9203	33.34	0.8535	31.04	0.7817
CBSD68	**CDnCNN-B**	39.85	0.9779	35.91	**0.9528**	**33.71**	**0.9280**	**32.21**	**0.9044**
	CDnCNN-B+NACPA	**41.69**	**0.9786**	**36.34**	0.9290	33.20	0.8772	30.92	0.8166
Urban100	**CDnCNN-B**	37.00	0.9681	34.50	**0.9497**	32.88	**0.9237**	**31.67**	**0.9167**
	CDnCNN-B+NACPA	**40.89**	**0.9756**	**36.09**	0.9290	**33.12**	0.8769	30.92	0.8233

noisy image with different σ of noises that tested on. The best performance, evaluated by PSNR and SSIM, were highlighted.



(a) CBM3D [16]: 33.50dB/0.9461 (b) Noise2Noise [36]: 29.78dB/0.8563 (c) CDnCNN [69]: 35.82dB/0.9793 (d) CDnCNN [69]+NAC: 33.67dB/0.9393



(e) CDnCNN [69]+NACPA: **35.90**dB/**0.9736** (f) RCAN* [72]: 33.15dB/0.9678 (g) RCAN* [72]+NAC: 33.05dB/0.9641 (h) RCAN* [72]+NACPA: 35.81dB/0.9747

Fig. 7: Denoising results of image "*0044_1*" from DND [1] dataset of different methods. The ground-truth image is not released, but PSNR (dB)/SSIM results are publicly provided on the *DND* Benchmark. Best results of PSNR and SSIM [59] are highlighted in **bold**.

For σ=5, the best performance acquired by training with the same level of noise. While for σ=10 20, the best performance were acquired by training with lower level of noises.

TABLE IX: Average PSNR (dB) and SSIM [59] results of CDnCNN [69] with our NACPA scheme trained with noise levels $\sigma = \{5, 10, 15, 20\}$ and cross tested on *Koda24* [2] with noise levels $\sigma = \{5, 10, 15, 20\}$. The highest results are highlighted in **bold**.

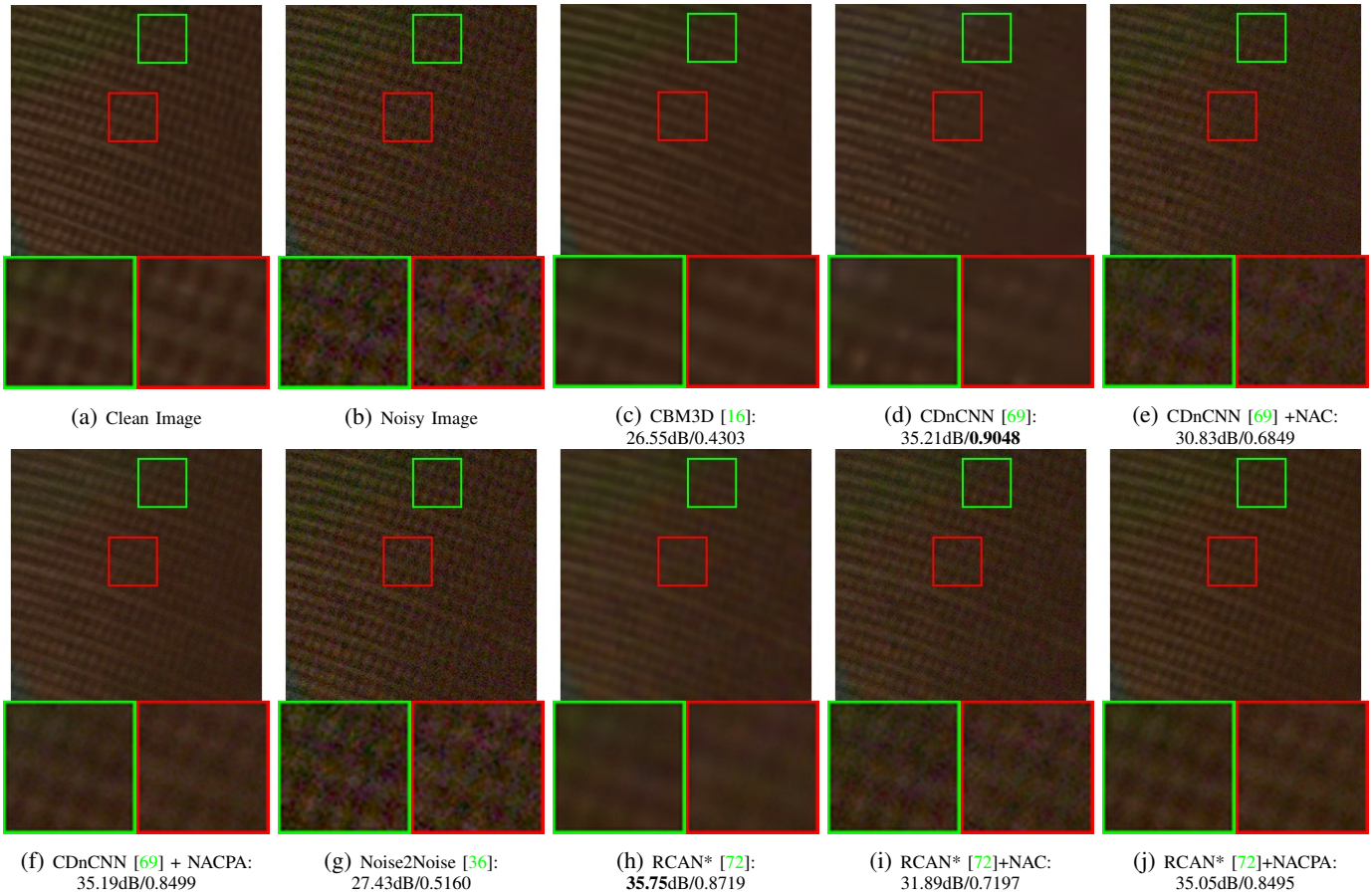
Noise Level	Metric	σ=5	σ=10	σ=15	σ=20
σ=5	PSNR	**42.96**	39.10	34.36	31.07
	SSIM	**0.9812**	0.9570	0.8788	0.7848
σ=10	PSNR	38.57	**39.24**	**37.48**	33.93
	SSIM	0.9520	**0.9632**	**0.9449**	0.8738
σ=15	PSNR	35.36	35.90	36.72	**36.28**
	SSIM	0.9108	0.9213	0.9396	**0.9376**
σ=20	PSNR	34.00	34.36	34.99	35.66
	SSIM	0.8877	0.8965	0.9117	0.9306

TABLE X: Average PSNR (dB) and SSIM [59] results of CDnCNN [69] with our NACPA scheme trained under noise levels $\sigma = \{5, 10, 15, 20\}$ and cross tested on *CBSD68* [43] under noise levels $\sigma = \{5, 10, 15, 20\}$. The highest results are highlighted in **bold**.

Noise Level	Metric	σ=5	σ=10	σ=15	σ=20
σ=5	PSNR	**43.09**	**38.88**	34.31	31.06
	SSIM	**0.9878**	0.9647	0.8999	0.8211
σ=10	PSNR	38.19	38.67	**36.94**	33.71
	SSIM	0.9613	**0.9709**	**0.9534**	0.8947
σ=15	PSNR	34.77	35.26	35.97	**35.47**
	SSIM	0.9203	0.9306	0.9472	**0.9439**
σ=20	PSNR	33.33	33.67	34.22	34.78
	SSIM	0.8975	0.9061	0.9205	0.9376

V. Conclusion

In this paper, we proposed the "Noisy-As-Clean" (NAC) and Prediction Augmentation (PA) strategies to transform original



(a) Clean Image (b) Noisy Image (c) CBM3D [16]: 26.55dB/0.4303 (d) CDnCNN [69]: 35.21dB/**0.9048** (e) CDnCNN [69] +NAC: 30.83dB/0.6849

(f) CDnCNN [69] + NACPA: 35.19dB/0.8499 (g) Noise2Noise [36]: 27.43dB/0.5160 (h) RCAN* [72]: **35.75**dB/0.8719 (i) RCAN* [72]+NAC: 31.89dB/0.7197 (j) RCAN* [72]+NACPA: 35.05dB/0.8495

Fig. 8: Denoising results of different methods on image "*8_15*" from SIDD Validation set [6]. Best results of PSNR and SSIM [59] are highlighted in **bold**.

supervised denoising networks into unsupervised ones, to evade the cumbersome requirements on the clean ground truth images in real-world image denoising problem. We integrate our two strategies into a NACPA scheme. Through applying the proposed NACPA scheme to denoising networks, trained with only noisy images, provide an alternative solution for unsupervised real-world image denoising problem. Experiments also shows that our NACPA scheme boosts the supervised image denoising networks to achieve competitive or even superior performance when compared to their original supervised ones. In our NACPA scheme, the PA strategy offers a practical inference strategy to further improve their unsupervised denoising performance. It can also be easily applied to other image denoising models for performance improvement. Extensive experiments on both synthesis and realistic noise removal demonstrated that the effectiveness of the proposed framework. With our NACPA scheme, the unsupervised trained image denoising networks achieves better performance than previous state-of-the-art unsupervised image denoising networks, and comparable to the supervised ones. Comprehensive ablation studies validated our contributions.

Acknowledgment

This work was supported by the NSFC (61872286 and 61701391), the National Key RD Program of China (2017YFF0107700), and Natural Science Basic Research Plan in Shaanxi Province of China (2018JM6092).

References

[1] Darmstadt Noise Dataset Benchmark. https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb. 2, 3, 7, 10, 11

[2] Kodak lossless true color image suite. https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb. 5, 6, 7, 8, 9, 10, 11

[3] PyTorch. https://pytorch.org/. 6

[4] Smartphone Image Denoising Dataset Benchmark. http://r0k.us/graphics/kodak. 7, 10

[5] A. Abdelhamed, M. A. Brubaker, and M. S. Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 2

[6] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 7, 10, 12

[7] S. Anwar and N. Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 3

[9] J. Batson and L. Royer. Noise2Self: Blind denoising by self-supervision. *arXiv preprint arXiv:1901.11365*, 2019. 1, 3, 4, 5

[10] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2019. 1

[11] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005. 1

[12] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012. 1, 2

[13] P. Chatterjee and P. Milanfar. Clustering-based denoising with locally learned dictionaries. *IEEE Transactions on Image Processing*, 18(7):1438–1451, 2009. 1

[14] J. Chen, J. Chen, H. Chao, and M. Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018. 1, 3, 7, 10

[15] S. I. Cho and S. J. Kang. Geodesic path-based diffusion acceleration for image denoising. *IEEE Transactions on Multimedia*, 20(7):1738–1750, 2017. 1

[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *IEEE International Conference on Image Processing*, pages 313–316. IEEE, 2007. 7, 10, 11, 12

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *Structure and parsimony for the adaptive representation of signals*, 2009. 1

[19] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes. Postprocessing of compressed images via sequential denoising. *IEEE Transactions on Image Processing*, 25(7):3044–3058, 2016. 1

[20] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 1

[21] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao. Pltd: Patch-based low-rank tensor decomposition for hyperspectral images. *IEEE Transactions on Multimedia*, 19(1):67–79, 2016. 1

[22] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 1

[23] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 1, 5

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[25] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. Efficient subpixel image registration algorithms. *Optics letters*, 33(2):156–158, 2008. 1

[26] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[28] Z. He, L. Xie, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data. *arXiv e-prints*, page arXiv:1909.09148, Sep 2019. 5

[29] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu, and L. Shao. Nlh: A blind pixel-level non-local method for real-world image denoising, 2019. 1

[30] B. Hu, L. Li, H. Liu, W. Lin, and J. Qian. Pairwise-comparison-based rank learning for benchmarking image restoration algorithms. *IEEE Transactions on Multimedia*, 2019. 1

[31] D. Huang, L. Kang, Y. Frank Wang, and C. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2013. 1

[32] J. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 7, 8, 9, 10

[33] A. Krull, T. Buchholz, and F. Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019. 1, 3, 4, 5

[34] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3587–3596, 2017. 1, 2, 4, 5

[35] S. Lefkimmiatis. Universal denoising networks: A novel cnn architecture for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[36] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, pages 2971–2980, 2018. 1, 3, 4, 5, 7, 10, 11, 12

[37] C. Li, C. Guo, J. Guo, P. Han, H. Fu, and R. Cong. Pdr-net: Perception-inspired single image dehazing network with refinement. *IEEE Transactions on Multimedia*, 2019. 1

[38] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 5

[39] C. Liu, R. Szeliski, S. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):299–314, 2008. 1

[40] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682. 2018. 1, 2, 4, 5

[41] X. Liu, M. Tanaka, and M. Okutomi. Practical signal-dependent noise parameter estimation from a single noisy image. *IEEE Transactions on Image Processing*, 23(10):4361–4371, 2014. 5

[42] X. Mao, C. Shen, and Y. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. In *Advances in Neural Information Processing Systems*, 2016. 1, 2, 6, 8

[43] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001. 6, 7, 8, 9, 10, 11

[44] N. Moran, D. Schmidt, Y. Zhong, and P. Coady. Noisier2Noise: Learning to Denoise from Unpaired Noisy Data. *arXiv e-prints*, page arXiv:1910.11908, Oct 2019. 1, 3

[45] I. Mosseri, M. Zontak, and M. Irani. Combining the power of internal and external denoising. In *International Conference on Intelligent Computer Communication and Processing*, pages 1–9, 2013. 1

[46] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1683–1691, 2016. 2, 3

[47] T. Plötz and S. Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, 2018. 1, 2

[48] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo. Neural blind deconvolution using deep priors. *arXiv preprint arXiv:1908.02197*, 2019. 1

[49] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng. Progressive image deraining networks: a better and simpler baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019. 1

[50] D. Ren, W. Zuo, D. Zhang, J. Xu, and L. Zhang. Partial deconvolution with inaccurate blur kernel. *IEEE Transactions on Image Processing*, 27(1):511–524, 2018. 1

[51] D. Ren, W. Zuo, D. Zhang, L. Zhang, and M.-H. Yang. Simultaneous fidelity and regularization learning for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[52] S. K. Roy, N. Bhattacharya, B. Chanda, B. B. Chaudhuri, and S. Banerjee. Image denoising using fractal hierarchical classification. In *Annual Convention of the Computer Society of India*, pages 631–645. Springer, 2018. 1

[53] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 3, 6, 8

[54] H. Talebi and P. Milanfar. Global image denoising. *IEEE Transactions on Image Processing*, 23(2):755–768, 2014. 1

[55] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin. Deep learning on image denoising: An overview. *arXiv preprint arXiv:1912.13171*, 2019. 1

[56] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu. Attention-guided cnn for image denoising. *Neural Networks*, 2020. 1

[57] C. Tian, Y. Xu, and W. Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. 1

[58] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 1, 3, 5

[59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1, 2, 5, 6, 7, 9, 10, 11, 12

[60] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012. 1, 2

[61] J. Xu, Y. Huang, L. Liu, F. Zhu, X. Hou, and L. Shao. Noisy-As-Clean: Learning Unsupervised Denoising from the Corrupted Image. *arXiv e-prints*, page arXiv:1906.06878, Jun 2019. 1, 4

[62] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018.

2

[63] J. Xu, L. Zhang, and D. Zhang. External prior guided internal prior learning for real-world noisy image denoising. *IEEE Transactions on Image Processing*, 27(6):2996–3010, June 2018. 1

[64] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *European Conference on Computer Vision*, 2018. 1, 3

[65] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1

[66] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 244–252, 2015. 1

[67] J. L. Yin, B. H. Chen, and Y. Li. Highly accurate image reconstruction for multimodal noise suppression using semisupervised learning on big data. *IEEE Transactions on Multimedia*, PP(99):1–1, 2018. 1

[68] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang. Variational denoising network: Toward blind noise modeling and removal. In *Advances in Neural Information Processing Systems*, pages 1688–1699, 2019. 5, 7

[69] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 2017. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

[70] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 2018. 2

[71] L. Zhang, S. Vaddadi, H. Jin, and S. K. Nayar. Multiple view image denoising. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549. IEEE, 2009. 1

[72] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, 2018. 5, 6, 7, 8, 10, 11, 12

[73] F. Zhu, G. Chen, and P. A. Heng. From noise modeling to blind image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–429, 2016. 1