

# Dual Adversarial Network: Toward Real-world Noise Removal and Noise Generation

Zongsheng Yue<sup>1,2</sup>, Qian Zhao<sup>1</sup>, Lei Zhang<sup>2,3</sup>, and Deyu Meng<sup>1,4,✉</sup>

<sup>1</sup> Xi'an Jiaotong University, Shaanxi, China

{zsy Zam, timmy.zhaqian}@gmail.com, dymeng@mail.xjtu.edu.cn

<sup>2</sup> Hong Kong Polytechnic University, Hong Kong, China

cslzhang@comp.polyu.edu.hk

<sup>3</sup> DAMO Academy, Alibaba Group, Shenzhen, China

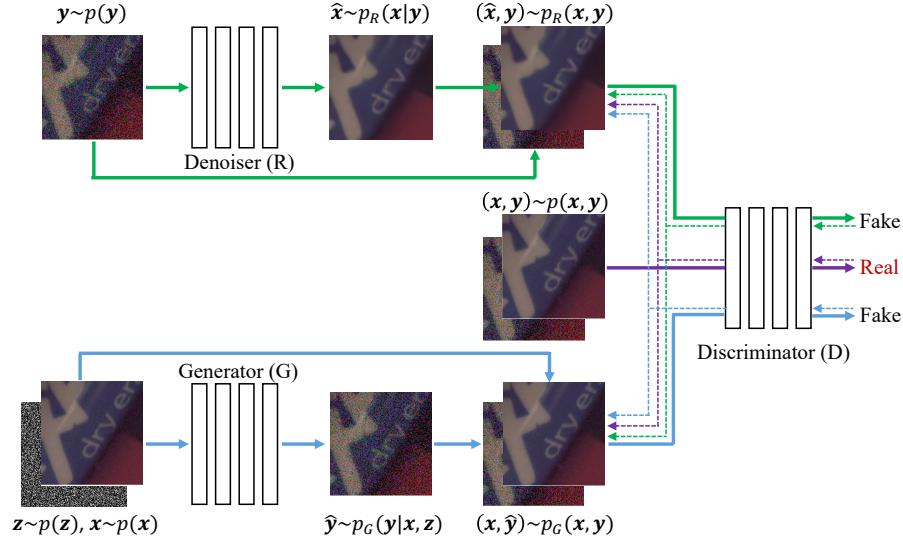
<sup>4</sup> The Macau University of Science and Technology, Macau, China

**Abstract.** Real-world image noise removal is a long-standing yet very challenging task in computer vision. The success of deep neural network in denoising stimulates the research of noise generation, aiming at synthesizing more clean-noisy image pairs to facilitate the training of deep denoisers. In this work, we propose a novel unified framework to simultaneously deal with the noise removal and noise generation tasks. Instead of only inferring the posteriori distribution of the latent clean image conditioned on the observed noisy image in traditional MAP framework, our proposed method learns the joint distribution of the clean-noisy image pairs. Specifically, we approximate the joint distribution with two different factorized forms, which can be formulated as a denoiser mapping the noisy image to the clean one and a generator mapping the clean image to the noisy one. The learned joint distribution implicitly contains all the information between the noisy and clean images, avoiding the necessity of manually designing the image priors and noise assumptions as traditional. Besides, the performance of our denoiser can be further improved by augmenting the original training dataset with the learned generator. Moreover, we propose two metrics to assess the quality of the generated noisy image, for which, to the best of our knowledge, such metrics are firstly proposed along this research line. Extensive experiments have been conducted to demonstrate the superiority of our method over the state-of-the-arts both in the real noise removal and generation tasks. The training and testing code is available at <https://github.com/zsy0AOA/DANet>.

**Keywords:** real-world, denoising, generation, metric

## 1 Introduction

Image denoising is an important research problem in low-level vision, aiming at recovering the **latent clean image  $\mathbf{x}$**  from its **noisy observation  $\mathbf{y}$** . Despite the significant advances in the past decades [8, 14, 56, 57], real image denoising still remains a challenging task, due to the complicated processing steps within the camera system, such as demosaicing, Gamma correction and compression [46].



**Fig. 1.** Illustration of our proposed dual adversarial framework. The solid lines denote the forward process, and the dotted lines mark the gradient interaction between the denoiser and generator during the backward.

From the Bayesian perspective, most of the traditional denoising methods can be interpreted within the Maximum A Posteriori (MAP) framework, i.e.,  $\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ , which involves one likelihood term  $p(\mathbf{y}|\mathbf{x})$  and one prior term  $p(\mathbf{x})$ . Under this framework, there are two methodologies that have been considered. The first attempts to model the likelihood term with proper distributions, e.g., Gaussian, Laplacian, MoG [33, 55, 59] and MoEP [10], which represents different understandings for the noise generation mechanism, while the second mainly focuses on exploiting better image priors, such as total variation [40], non-local similarity [8], low-rankness [15, 17, 47, 53] and sparsity [31, 52, 58]. Despite better interpretability led by Bayesian framework, these MAP-based methods are still limited by the manual assumptions on the noise and image priors, which may largely deviate from the real images.

In recent years, deep learning (DL)-based methods have achieved impressive success in image denoising task [4, 56, 57]. However, as is well known, training a deep denoiser requires large amount of clean-noisy image pairs, which are time-consuming and expensive to collect. To address this issue, several noise generation<sup>5</sup> approaches were proposed to simulate more clean-noisy image pairs to facilitate the training of deep denoisers. The main idea behind them is to unfold the in-camera processing pipelines [7, 19], or directly learn the distribution  $p(\mathbf{y})$  as in [11, 25] using generative adversarial network (GAN) [16]. However, the former methods involve many hyper-parameters needed to be carefully tuned for

<sup>5</sup> The phrase “noise generation” indicates the generation process of noisy image from clean image throughout this paper.

specific cameras, and the latter ones suffer from simulating very realistic noisy image with high-dimensional signal-dependent noise distributions. Besides, to the best of our knowledge, there is still no metric to quantitatively assess the quality of the generated noisy images w.r.t. the real ones.

Against these issues, we propose a new framework to model the joint distribution  $p(\mathbf{x}, \mathbf{y})$  instead of only inferring the conditional posteriori  $p(\mathbf{x}|\mathbf{y})$  as in conventional MAP framework. Specifically, we firstly factorize the joint distribution  $p(\mathbf{x}, \mathbf{y})$  from two opposite directions, i.e.,  $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$  and  $\int_{\mathbf{z}} p(\mathbf{y}|\mathbf{x}, \mathbf{z})p(\mathbf{x})p(\mathbf{z})d\mathbf{z}$ , which can be well approximated by a image denoiser and a noise generator. Then we simultaneously train the denoiser and generator in a dual adversarial manner as illustrated in Fig. 1. After that, the learned denoiser can either be directly used for the real noise removal task, or further enhanced with new clean-noisy image pairs simulated by the learned generator. In summary, the contributions of this work can be mainly summarized as:

- Different from the traditional MAP framework, our method approximates the joint distribution  $p(\mathbf{x}, \mathbf{y})$  from two different factorized forms in a dual adversarial manner, which subtly avoids the manual design on image priors and noise distribution. What’s more, the joint distribution theoretically contains more complete information underlying the data set comparing with the conditional posteriori  $p(\mathbf{x}|\mathbf{y})$ .
- Our proposed method can simultaneously deal with both the noise removal and noise generation tasks in one unified Bayesian framework, and achieves superior performance than the state-of-the-arts in both these two tasks. What’s more, the performance of our denoiser can be further improved after retraining on the augmented training data set with additional clean-noisy image pairs simulated by our learned generator.
- In order to assess the quality of the simulated noisy images by a noise generation method, we design two metrics, which, to the best of our knowledge, are the first metrics to this aim.

## 2 Related Work

### 2.1 Noise Removal

Image denoising is an active research topic in computer vision. Under the MAP framework, rational priors are necessary to be pre-assumed to enforce some desired properties of the recovered image. Total variation [40] was firstly introduced to deal with the denoising task. Later, the non-local similarity prior, meaning that the small patches in a large non-local area may share some similar patterns, was considered in NLM [8] and followed by many other denoising methods [14, 15, 28, 30]. Low-rankness [15, 17, 53, 54] and sparsity [30, 31, 50, 52, 58] are another two well-known image priors, which are often used together within the dictionary learning methods. Besides, discriminative learning methods also represent another research line, mainly including Markov random field (MRF) methods [6, 41, 44], cascade of shrinkage fields (CSF) methods [42, 43] and the trainable

nonlinear reaction diffusion (TNRD) [12] method. Different from above priors-based methods, noise modeling approaches focus on the other important component of MAP, i.e., likelihood or fidelity term. E.g., Meng and De La Torre [33] proposed to model the noise distribution as mixture of Gaussians (MoG), while Zhu *et al.* [59] and Yue *et al.* [55] both introduced the non-parametric Dirichlet Process to MoG to expand its flexibility. Furthermore, Cao *et al.* [10] proposed the mixture of exponential power (MoEP) distributions to fit more complex noise.

In recent years, DL-based methods achieved significant advances in the image denoising task. Jain and Seung [23] firstly adopted a five-layer network to deal with the denoising task. Then Burger *et al.* [9] obtained the comparable performance with BM3D using one plain multi-layer perceptron (MLP). Later, some auto-encoder based methods [2, 49] were also immediately proposed. It is worthy mentioning that Zhang *et al.* [57] proposed the convolutional denoising network DnCNN and achieved the state-of-the-art performance on Gaussian denoising. Following DnCNN, many different network architectures were designed to deal with the denoising task, including RED [32], MemNet [45], NLRN [29], N3Net [37], RIDNet [4] and VDN [56].

## 2.2 Noise Generation

As is well known, the expensive cost of collecting pairs of training data is a critical limitation for deep learning based denoising methods. Therefore, several methods were proposed to explore the generation mechanism of image noise to facilitate an easy simulation of more training data pairs. One common idea was to generate image pairs by “unprocessing” and “processing” each step of the in-camera processing pipelines, e.g., [7, 19, 24]. However, these methods involve many hyper-parameters to be tuned for specific camera. Another simpler way was to learn the real noise distribution directly using GAN [16] as demonstrated in [11] and [25]. Due to the complexity of real noise and the instability of training GAN, it is very difficult to train a good generator for simulating realistic noise.

## 3 Proposed Method

Like most of the supervised deep learning denoising methods, our approach is built on the given training data set containing pairs of real noisy image  $\mathbf{y}$  and clean image  $\mathbf{x}$ , which are accessible thanks to the contributions of [1, 3, 51]. Instead of forcibly learning a mapping from  $\mathbf{y}$  to  $\mathbf{x}$ , we attempt to approximate the underlying joint distribution  $p(\mathbf{x}, \mathbf{y})$  of the clean-noisy image pairs. In the following, we present our method from the Bayesian perspective.

### 3.1 Two Factorizations of Joint Distribution

In this part, we factorize the joint distribution  $p(\mathbf{x}, \mathbf{y})$  from two different perspectives, and discuss their insights respectively related to the noise removal and noise generation tasks.

**Noise removal perspective:** The noise removal task can be considered as inferring the conditional distribution  $p(\mathbf{x}|\mathbf{y})$  under the Bayesian framework. The learned denoiser  $R$  in this task represents an implicit distribution  $p_R(\mathbf{x}|\mathbf{y})$  to approximate the true distribution  $p(\mathbf{x}|\mathbf{y})$ . The output of  $R$  can be seen as an image sampled from this implicit distribution  $p_R(\mathbf{x}|\mathbf{y})$ . Based on such understanding, we can obtain a pseudo clean image pair  $(\hat{\mathbf{x}}, \mathbf{y})$  as follows<sup>6</sup>, i.e.,

$$\mathbf{y} \sim p(\mathbf{y}), \hat{\mathbf{x}} = R(\mathbf{y}) \implies (\hat{\mathbf{x}}, \mathbf{y}), \quad (1)$$

which can be seen as one example sampled from the following pseudo joint distribution:

$$p_R(\mathbf{x}, \mathbf{y}) = p_R(\mathbf{x}|\mathbf{y})p(\mathbf{y}). \quad (2)$$

Obviously, the better denoiser  $R$  is, the more accurately that the pseudo joint distribution  $p_R(\mathbf{x}, \mathbf{y})$  can approximate the true joint distribution  $p(\mathbf{x}, \mathbf{y})$ .

**Noise generation perspective:** In real camera system, image noise is derived from multiple hardware-related random noises (e.g., short noise, thermal noise), and further affected by in-camera processing pipelines (e.g., demosaicing, compression). After introducing an additional latent variable  $\mathbf{z}$ , representing the fundamental elements conducting the hardware-related random noises, the generation process from  $\mathbf{x}$  to  $\mathbf{y}$  can be depicted by the conditional distribution  $p(\mathbf{y}|\mathbf{x}, \mathbf{z})$ . The generator  $G$  in this task expresses an implicit distribution  $p_G(\mathbf{y}|\mathbf{x}, \mathbf{z})$  to approximate the true distribution  $p(\mathbf{y}|\mathbf{x}, \mathbf{z})$ . The output of  $G$  can be seen as an example sampled from  $p_G(\mathbf{y}|\mathbf{x}, \mathbf{z})$ , i.e.,  $G(\mathbf{x}, \mathbf{z}) \sim p_G(\mathbf{y}|\mathbf{x}, \mathbf{z})$ . Similar as Eq. (1), a pseudo noisy image pair  $(\mathbf{x}, \hat{\mathbf{y}})$  is easily obtained:

$$\mathbf{z} \sim p(\mathbf{z}), \mathbf{x} \sim p(\mathbf{x}), \hat{\mathbf{y}} = G(\mathbf{x}, \mathbf{z}) \implies (\mathbf{x}, \hat{\mathbf{y}}), \quad (3)$$

where  $p(\mathbf{z})$  denotes the distribution of the latent variable  $\mathbf{z}$ , which can be easily set as an isotropic Gaussian distribution  $\mathcal{N}(0, \mathbf{I})$ .

Theoretically, we can marginalize the latent variable  $\mathbf{z}$  to obtain the following pseudo joint distribution  $p_G(\mathbf{x}, \mathbf{y})$  as an approximation to  $p(\mathbf{x}, \mathbf{y})$ :

$$p_G(\mathbf{x}, \mathbf{y}) = \int_{\mathbf{z}} p_G(\mathbf{y}|\mathbf{x}, \mathbf{z})p(\mathbf{x})p(\mathbf{z})d\mathbf{z} \approx \frac{1}{L} \sum_i^L p_G(\mathbf{y}|\mathbf{x}, \mathbf{z}_i)p(\mathbf{x}), \quad (4)$$

where  $\mathbf{z}_i \sim p(\mathbf{z})$ . As suggested in [27], the number of samples  $L$  can be set as 1 as long as the minibatch size is large enough. Under such setting, the pseudo noisy image pair  $(\mathbf{x}, \hat{\mathbf{y}})$  obtained from the generation process in Eq. (3) can be roughly regarded as an sampled example from  $p_G(\mathbf{x}, \mathbf{y})$ .

### 3.2 Dual Adversarial Model

In the previous subsection, we have derived two pseudo joint distributions from the perspectives of noise removal and noise generation, i.e.,  $p_R(\mathbf{x}, \mathbf{y})$  and  $p_G(\mathbf{x}, \mathbf{y})$ .

---

<sup>6</sup> We mildly assume that  $\mathbf{y} \sim p(\mathbf{y})$  is easily implemented by sampling  $\mathbf{y}$  from the empirical distribution  $p(\mathbf{y})$  of the training data set, and so does as  $\mathbf{x} \sim p(\mathbf{x})$ .

Now the problem becomes how to effectively train the denoiser  $R$  and the generator  $G$ , in order to well approximate the joint distribution  $p(\mathbf{x}, \mathbf{y})$ . Fortunately, the tractability of sampling process defined in Eqs. (1) and (3) makes such training possible in an adversarial manner as GAN [16], which gradually pushes  $p_R(\mathbf{x}, \mathbf{y})$  and  $p_G(\mathbf{x}, \mathbf{y})$  toward the true distribution  $p(\mathbf{x}, \mathbf{y})$ . Specifically, we formulate this idea as the following dual adversarial problem inspired by Triple-GAN [13],

$$\min_{R, G} \max_D \mathcal{L}_{gan}(R, G, D) = E_{(\mathbf{x}, \mathbf{y})}[D(\mathbf{x}, \mathbf{y})] - \alpha E_{(\hat{\mathbf{x}}, \mathbf{y})}[D(\hat{\mathbf{x}}, \mathbf{y})] \\ - (1 - \alpha)E_{(\mathbf{x}, \hat{\mathbf{y}})}[D(\mathbf{x}, \hat{\mathbf{y}})], \quad (5)$$

where  $\hat{\mathbf{x}} = R(\mathbf{y})$ ,  $\hat{\mathbf{y}} = G(\mathbf{x}, \mathbf{z})$ , and  $D$  denotes the discriminator, which tries to distinguish the real clean-noisy image pair  $(\mathbf{x}, \mathbf{y})$  from the fake ones  $(\hat{\mathbf{x}}, \mathbf{y})$  and  $(\mathbf{x}, \hat{\mathbf{y}})$ . The hyper-parameter  $\alpha$  controls the relative importance between the denoiser  $R$  and generator  $G$ . As in [5], we use the Wasserstein-1 distance to measure the difference between two distributions in Eq. (5).

The working mechanism of our dual adversarial network can be intuitively explained in Fig. 1. On one hand, the denoiser  $R$ , delivering the knowledge of  $p_R(\mathbf{x}|\mathbf{y})$ , is expected to conduct the joint distribution  $p_R(\mathbf{x}, \mathbf{y})$  of Eq. (2), while the noise generator  $G$ , conveying the information of  $p_G(\mathbf{y}|\mathbf{x}, \mathbf{z})$ , is expected to derive the joint distribution  $p_G(\mathbf{x}, \mathbf{y})$  of Eq. (4). Through the adversarial effect of discriminator  $D$ , the denoiser  $R$  and generator  $G$  are both gradually optimized so as to pull  $p_R(\mathbf{x}, \mathbf{y})$  and  $p_G(\mathbf{x}, \mathbf{y})$  toward the true joint distribution  $p(\mathbf{x}, \mathbf{y})$  during training. On the other hand, the capabilities of  $R$  and  $G$  are mutually enhanced by their dual regularization between each other. Given any real image pair  $(\mathbf{x}, \mathbf{y})$  and one pseudo image pair  $(\mathbf{x}, \hat{\mathbf{y}})$  from generator  $G$  or  $(\hat{\mathbf{x}}, \mathbf{y})$  from denoiser  $R$ , the discriminator  $D$  will be updated according to the adversarial loss. Then  $D$  is fixed as a criterion to update both  $R$  and  $G$  simultaneously as illustrated by the dotted lines in Fig. 1, which means  $R$  and  $G$  are keeping interactive and guided by each other in each iteration.

Previous researches [22, 60] have shown that it is beneficial to mix the adversarial objective with traditional losses, which would speed up and stabilize the training of GAN. For noise removal task, we adopt the  $L_1$  loss, i.e.,  $\|\hat{\mathbf{x}} - \mathbf{x}\|_1$ , which enforces the output of denoiser  $R$  to be close to the groundtruth. For the generator  $G$ , however, the direct  $L_1$  loss would not be beneficial because of the randomness of noise. Therefore, we propose to apply the  $L_1$  constrain on the statistical features of noise distribution:

$$\|\mathcal{GF}(\hat{\mathbf{y}} - \mathbf{x}) - \mathcal{GF}(\mathbf{y} - \mathbf{x})\|_1, \quad (6)$$

where  $\mathcal{GF}(\cdot)$  represents the Gaussian filter used to extract the first-order statistical information of noise. Intergrating these two regularizers into the adversarial loss of Eq. (5), we obtain the final objective:

$$\min_{R, G} \max_D \mathcal{L}_{gan}(R, G, D) + \tau_1 \|\hat{\mathbf{x}} - \mathbf{x}\|_1 + \tau_2 \|\mathcal{GF}(\hat{\mathbf{y}} - \mathbf{x}) - \mathcal{GF}(\mathbf{y} - \mathbf{x})\|_1, \quad (7)$$

where  $\tau_1$  and  $\tau_2$  are hyper-parameters to balance different losses. More sense-tiveness analysis on them are provided in Sec. 5.2.

**Algorithm 1** Dual adversarial network.

---

**Input:** hyper-parameters:  $\tau_1, \tau_2, \alpha, n_{critic}$

- 1: **while**  $\theta$  had not converged **do**
- 2:   **for**  $i = 1, 2, \dots, n_{critic}$  **do**
- 3:     Sample a batch of pairs  $(\mathbf{x}, \mathbf{y})$  from  $p(\mathbf{x}, \mathbf{y})$
- 4:     Sample a batch of pairs  $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$  from  $p_R(\mathbf{x}, \mathbf{y})$  and  $(\mathbf{x}, \hat{\mathbf{y}})$  from  $p_G(\mathbf{x}, \mathbf{y})$
- 5:     Update discriminator  $D$  with fixed  $R$  and  $G$
- 6:   **end for**
- 7:   Update denoiser  $R$  with fixed  $G$  and  $D$
- 8:   Update generator  $G$  with fixed  $R$  and  $D$
- 9: **end while**

---

### 3.3 Training Strategy

In the dual adversarial model of Eq. (7), we have three objects to be optimized, i.e., the denoiser  $R$ , generator  $G$  and discriminator  $D$ . As in most of the GAN-related papers [5, 13, 16], we jointly train  $R$ ,  $G$  and  $D$  but update them in an alternating manner as shown in Algorithm 1. In order to stabilize the training, we adopt the gradient penalty technology in WGAN-GP [18], enforcing the discriminator to satisfy 1-Lipschitz constraint by an extra gradient penalty term.

After training, the generator  $G$  is able to simulate more noisy images given any clean images, which are easily obtained from the original training data set or by downloading from internet. Then we can retrain the denoiser  $R$  by adding more synthetic clean-noisy image pairs generated by  $G$  to the training data set. As shown in Sec. 5, this strategy can further improve the denoising performance.

### 3.4 Network Architecture

The denoiser  $R$ , generator  $G$  and discriminator  $D$  in our framework are all parameterized as deep neural networks due to their powerful fitting capability. As shown in Fig. 1, the denoiser  $R$  takes noisy image  $\mathbf{y}$  as input and outputs denoised image  $\hat{\mathbf{x}}$ , while the generator  $G$  takes the concatenated clean image  $\mathbf{x}$  and latent variable  $\mathbf{z}$  as input and outputs the simulated noisy image  $\hat{\mathbf{y}}$ . For both  $R$  and  $G$ , we use the UNet [39] architecture as backbones. Besides, the residual learning strategy [57] is adopted in both of them. The discriminator  $D$  contains five stride convolutional layers to reduce the image size and one fully connected layer to fuse all the information. More details about the network architectures are provided in the supplementary material due to page limitation. It should be noted that our proposed method is a general framework that does not depend on the specific architecture, therefore most of the commonly used networks architectures [4, 32, 57] in low-level vision tasks can be substituted.

## 4 Evaluation Metrics

For the noise removal task, PSNR and SSIM [48] can be readily adopted to compare the denoising performance of different methods. However, to the best of

our knowledge, there is still no any quantitative metric having been designed for noise generation task. To address this issue, we propose two metrics to compare the similarity between the generated and the real noisy images as follows:

- **PGap** (PSNR Gap): The main idea of PGap is to compare the synthetic and real noisy images indirectly by the performance of the denoisers trained on them. Let  $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$ ,  $\mathcal{T} = \{(\tilde{\mathbf{x}}^j, \tilde{\mathbf{y}}^j)\}_{j=1}^S$  denote the available training and testing sets, whose noise distributions are same or similar. Given any one noisy image generator  $G$ , we can synthesize another training set:

$$\mathcal{D}_G = \{(\mathbf{x}^i, \tilde{\mathbf{y}}^i) | \tilde{\mathbf{y}}^i = G(\mathbf{x}^i, \mathbf{z}^i), \mathbf{z}^i \sim p(\mathbf{z})\}_{i=1}^N. \quad (8)$$

After training two denoisers  $R_1$  on the original data set  $\mathcal{D}$  and  $R_2$  on the generated data set  $\mathcal{D}_G$  under the **same** conditions, we can define PGap as

$$\text{PGap} = \text{PSNR}(R_1(\mathcal{T})) - \text{PSNR}(R_2(\mathcal{T})), \quad (9)$$

where  $\text{PSNR}(R_i(\mathcal{T}))(i = 1, 2)$  represents the PSNR result of denoiser  $R_i$  on testing data set  $\mathcal{T}$ . It is obvious that, if the generated noisy images in  $\mathcal{D}_G$  are close to the real noisy ones in  $\mathcal{D}$ , the performance of  $R_2$  would be close to  $R_1$ , and thus the PGap would be small.

- **AKLD** (Average KL Divergence): The noise generation task aims at synthesizing fake noisy image  $\mathbf{y}^f$  from the real clean image  $\mathbf{x}^r$  to match the real noisy image  $\mathbf{y}^r$  in distribution. Therefore, the KL divergence between the conditional distributions  $p_{\mathbf{y}^f}(\mathbf{y}|\mathbf{x})$  on the fake image pair  $(\mathbf{x}^r, \mathbf{y}^f)$  and  $p_{\mathbf{y}^r}(\mathbf{y}|\mathbf{x})$  on the real image pair  $(\mathbf{x}^r, \mathbf{y}^r)$  can serve as a metric. To make this conditional distribution tractable, we utilize the pixel-wisely Gaussian assumption for real noise in recent work VDN [56], i.e.,

$$p_{\mathbf{y}^c}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|[\mathbf{x}^r], \text{diag}([\mathbf{V}^c])), \quad c \in \{f, r\}, \quad (10)$$

where

$$\mathbf{V}^c = \mathcal{GF}((\mathbf{y}^c - \mathbf{x}^r)^2), \quad c \in \{f, r\}, \quad (11)$$

$[\cdot]$  denotes the reshape operation from matrix to vector,  $\mathcal{GF}(\cdot)$  denotes the Gaussian filter, and the square of  $(\mathbf{y}^c - \mathbf{x}^r)^2$  is pixel-wise operation. Based on such explicit distribution assumption, the KL divergence between  $p_{\mathbf{y}^f}(\mathbf{y}|\mathbf{x})$  and  $p_{\mathbf{y}^r}(\mathbf{y}|\mathbf{x})$  can be regarded as an intuitive metric. To reduce the influence of randomness, we randomly generate  $L$  synthetic fake noisy images:

$$\mathbf{y}^{f_j} = G(\mathbf{x}^r, \mathbf{z}^j), \quad \mathbf{z}^j \sim p(\mathbf{z}), \quad j = 1, 2, \dots, L, \quad (12)$$

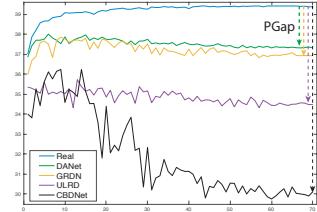
for any real clean image  $\mathbf{x}^r$ , and define the following average KL divergence as our metric, i.e.,

$$\text{AKLD} = \frac{1}{L} \sum_{j=1}^L \text{KL}[p_{\mathbf{y}^{f_j}} p(\mathbf{y}|\mathbf{x}) || p_{\mathbf{y}^r} p(\mathbf{y}|\mathbf{x})]. \quad (13)$$

Evidently, the smaller AKLD is, the better the generator  $G$  is. In the following experiments, we set  $L = 50$ .

**Table 1.** The PGap and AKLD performances of different compared methods on the SIDD validation data set. And the best results are highlighted in bold.

Metrics	Methods			
	CBDNet	ULRD	GRDN	DANet
PGap $\downarrow$	8.30	4.90	2.28	<b>2.06</b>
AKLD $\downarrow$	0.728	0.545	0.443	<b>0.212</b>



**Fig. 2.** PSNR results of different methods during training.

## 5 Experimental Results

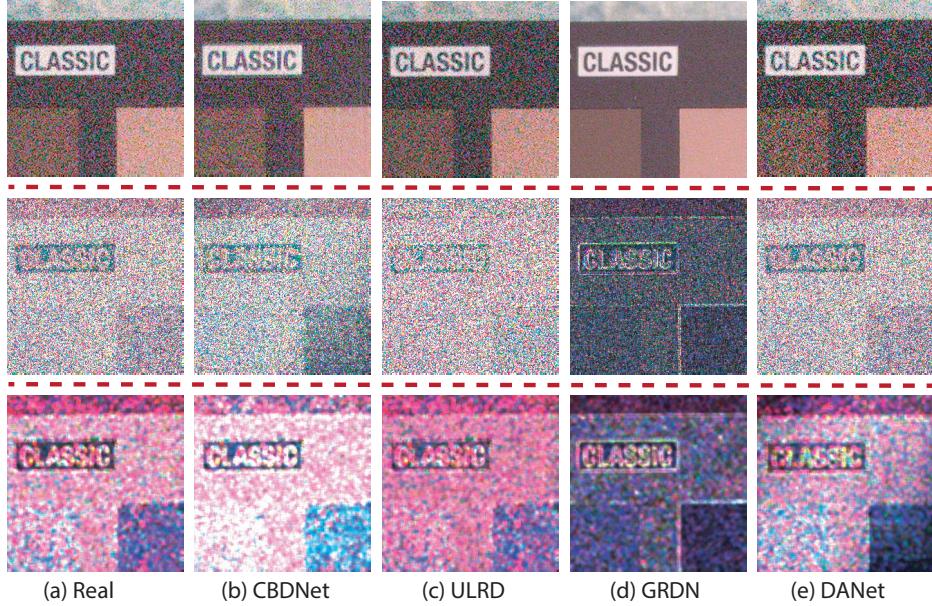
In this section, we conducted a series of experiments on several real-world denoising benchmarks. In specific, we considered two groups of experiments: the first group (Sec. 5.2) is designed for evaluating the effectiveness of our method on both of the noise removal and noise generation tasks, which is implemented on one specific real benchmark containing training, validation and testing sets; while the second group (Sec. 5.3) is conducted on two real benchmarks that only consist of some noisy images as testing set, aiming at evaluating its performance on general real-world denoising tasks. Due to the page limitation, the running time comparisons are listed in the supplementary material.

In brief, we denote the jointly trained **Dual Adversarial Network** following Algorithm 1 as DANet. As discussed in Sec. 3.3, the learned generator  $G$  in DANet is able to augment the original training set by generating more synthetic clean-noisy image pairs, and the retrained denoiser  $R$  on this augmented training data set under  $L_1$  loss is denoted as DANet<sub>+</sub>.

### 5.1 Experimental Settings

**Parameter settings and network training:** In the training stage of DANet, the weights of  $R$  and  $G$  were both initialized according to [20], and the weights of  $D$  were initialized from a zero-centered Normal distribution with standard deviation 0.02 as [38]. All the three networks were trained by Adam optimizer [26] with momentum terms (0.9, 0.999) for  $R$  and (0.5, 0.9) for both  $G$  and  $D$ . The learning rates were set as 1e-4, 1e-4 and 2e-4 for  $R$ ,  $G$  and  $D$ , respectively, and linearly decayed in half every 10 epochs.

In each epoch, we randomly cropped  $16 \times 5000$  patches with size  $128 \times 128$  from the images for training. During training, we updated  $D$  three times for each update of  $R$  and  $G$ . We set  $\tau_1 = 1000$ ,  $\tau_2 = 10$  throughout the experiments, and the sensetiveness analysis about them can be found in Sec. 5.2. As for  $\alpha$ , we set it as 0.5, meaning the denoiser  $R$  and generator  $G$  contribute equally in our model. The penalty coefficient in WGAN-GP [18] is set as 10 following its default settings. As for DANet<sub>+</sub>, the denoiser  $R$  was retrained with the same settings as that in DANet. All the models were trained using PyTorch [35].



**Fig. 3.** Illustration of one typical generated noisy images (1st row) by different methods and their corresponding noise (2nd row) and variance map (3rd row) estimated by Eq. (11). The first column represents the real ones in SIDD validation set.

## 5.2 Results on SIDD Benchmark

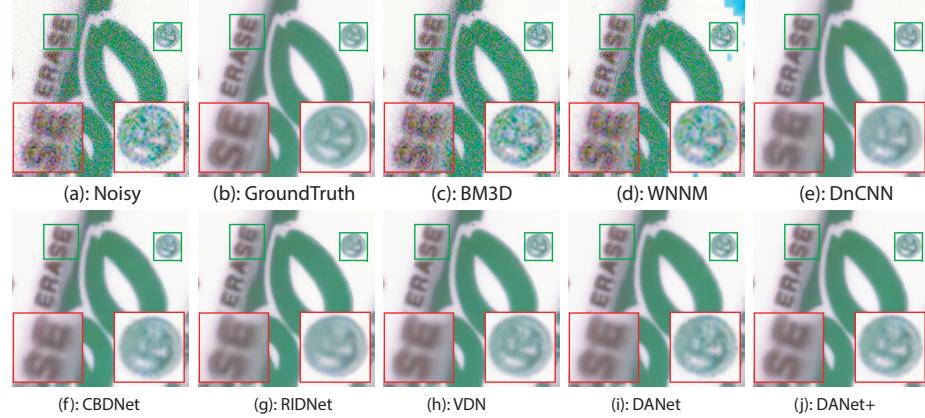
In this part, SIDD [1] benchmark is employed to evaluate the denoising performance and generation quality of our proposed method. The full SIDD data set contains about 24000 clean-noisy image pairs as training data, and the rest 6000 image pairs are held as the benchmark for testing. For fast training and evaluation, one medium training set (320 image pairs) and validation set (40 image pairs) are also provided, but the testing results can only be obtained by submission. We trained DANet and DANet<sub>+</sub> on the medium version training set, and evaluated on the validation and testing sets.

**Noise Generation:** The generator  $G$  in DANet is mainly used to synthesize the corresponding noisy image given any clean one. As introduced in Sec. 4, two metrics PGap and AKLD are designed to assess the generated noisy image. Based on these two metrics, we compared DANet with three recent methods, including CBDNet [19], ULRD [7] and GRDN [25]. CBDNet and ULRD both attempted to generate noisy images by simulating the in-camera processing pipelines, while GRDN directly learned the noise distribution using GAN [16]. It should be noted that ULRD [7] and GRDN [25] both make use of the metadata of the images.

Table 1 lists the PGap values of different compared methods on SIDD validation set. For the calculation of PGap, SIDD validation set is regarded as the testing set  $\mathcal{T}$  in Eq. (9). Obviously, our proposed DANet achieves the best performance. Figure 2 displays the PSNR curves of different denoisers trained on

**Table 2.** The PSNR and SSIM results of different methods on SIDD validation and testing sets. The best results are highlighted in bold.

Datasets	Metrics	Methods							
		CBM3D	WNNM	DnCNN	CBDNet	RIDNet	VDN	DANet	DANet <sub>+</sub>
Testing	PSNR↑	25.65	25.78	23.66	33.28	-	39.26	39.25	<b>39.43</b>
	SSIM↑	0.685	0.809	0.583	0.868	-	0.955	0.955	<b>0.956</b>
Validation	PSNR↑	25.29	26.31	38.56	38.68	38.71	39.29	39.30	<b>39.47</b>
	SSIM↑	0.412	0.524	0.910	0.909	0.913	0.911	0.916	<b>0.918</b>



**Fig. 4.** One typical denoising example in the SIDD validation dataset.

the real training set or only the synthetic training sets generated by different methods, which gives an intuitive illustration for our defined PGap. It can be seen that all the methods tend to gradually overfit to their own synthetic training set, especially for CBDNet. However, DANet performs not only more stably but also better than other methods.

The average AKLD results calculated on all the images of SIDD validation set are also listed in Table 1. The smallest AKLD of DANet indicates that it learns a better implicit distribution to approximate the true distribution  $p(\mathbf{y}|\mathbf{x})$ . Fig. 3 illustrates one typical example of the real and synthetic noisy images generated by different methods, which provides an intuitive visualization for the AKLD metric. In summary, DANet outperforms other methods both in quantization and visualization, even though some of them make use of additional metadata.

**Noise Removal:** To verify the effectiveness of our proposed method on real-world denoising task, we compared it with several state-of-the-art methods, including CBM3D [14], WNNM [17], DnCNN [57], CBDNet [19], RIDNet [4] and VDN [56]. Table 2 lists the PSNR and SSIM results of different methods on SIDD validation and testing sets. It should be noted that the results on testing sets are cited from official website<sup>7</sup>, but the results on validation set are calculated by ourselves. For fair comparison, we retrained DnCNN and CBDNet on SIDD training set. From Table 2, it is easily observed that: 1) deep learning methods obviously performs better than traditional methods CBM3D and WNNM due to

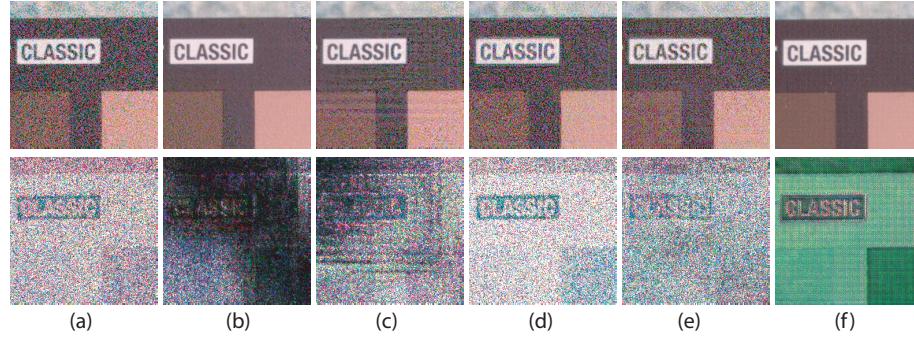
<sup>7</sup> <https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php>

**Table 3.** The PSNR and SSIM results of DANet under different  $\tau_1$  values on SIDD validation data set.

Metrics	$\tau_1$			
	1e+2	1e+3	1e+4	$+\infty$
PSNR↑	38.66	39.30	39.33	39.39
SSIM↑	0.901	0.916	0.916	0.917

**Table 4.** The PGap and AKLD results of DANet under different  $\tau_2$  values on SIDD validation data set.

Metrics	$\tau_2$				
	0	5	10	50	$+\infty$
PGap↓	5.33	3.10	2.06	4.17	15.14
AKLD↓	0.386	0.216	0.212	0.177	0.514



**Fig. 5.** This figure displays the real or generated noisy images (the 1st row) by DANet under different  $\tau_2$  value and the corresponding noise (the 2nd row). From left to right: (a) real case, (b)  $\tau_2 = 0$ , (c)  $\tau_2 = 5$ , (d)  $\tau_2 = 10$ , (e)  $\tau_2 = 50$ , (f)  $\tau_2 = +\infty$ .

the powerful fitting capability of DNN; 2) DANet and DANet<sub>+</sub> both outperform the state-of-the-art real-world denoising methods, substantiating their effectiveness; 3) DANet<sub>+</sub> surpasses DANet about 0.18dB PSNR, which indicates that the synthetic data by  $G$  facilitates the training of the denoiser  $R$ .

Fig. 4 illustrates the visual denoising results of different methods. It can be seen that CBM3D and WNNM both fail to remove the real-world noise. DnCNN tends to produce over-smooth edges and textures due to the  $L_2$  loss. CBDNet, RIDNet and VDN alleviate this phenomenon to some extent since they adopt more robust loss functions. DANet recovers sharper edges and more details owing to the adversarial loss. After retraining with more generated image pairs, DANet<sub>+</sub> obtains the closer denoising results to the groundtruth.

**Hyper-parameter Analysis:** Our proposed DANet involves two hyper-parameters  $\tau_1$  and  $\tau_2$  in Eq. (7). The parameter  $\tau_1$  mainly influences the performance of denoiser  $R$ , while  $\tau_2$  directly affects the generator  $G$ .

Table 3 lists the PSNR/SSIM results of DANet under different  $\tau_1$  settings, where  $\tau_1 = +\infty$  represents the results of the denoiser  $R$  trained only with  $L_1$  loss. As expected, small  $\tau_1$  value, meaning that the adversarial loss plays more important role, leads to the decrease of PSNR and SSIM performance to some extent. However, when  $\tau_1$  value is too large, the  $L_1$  regularizer will mainly dominates the performance of denoiser  $R$ . Therefore, we set  $\tau_1$  as a moderate value 1e+3 throughout all the experiments, which makes the denoising results more realistic as shown in Fig. 4 even sacrificing a little PSNR performance.

**Table 5.** The comparison results of BaseD and DANet on SIDD validation set.

Metrics	Methods	
	BaseD	DANet
PSNR↑	39.19	39.30
SSIM↑	0.907	0.916

**Table 6.** The comparison results of BaseG and DANet on SIDD validation set.

Metrics	Methods	
	BaseG	DANet
PGap↓	4.07	2.06
AKLD↓	0.223	0.212

**Table 7.** The PSNR and SSIM results of different methods on DND benchmark. The best results are highlighted as bold.

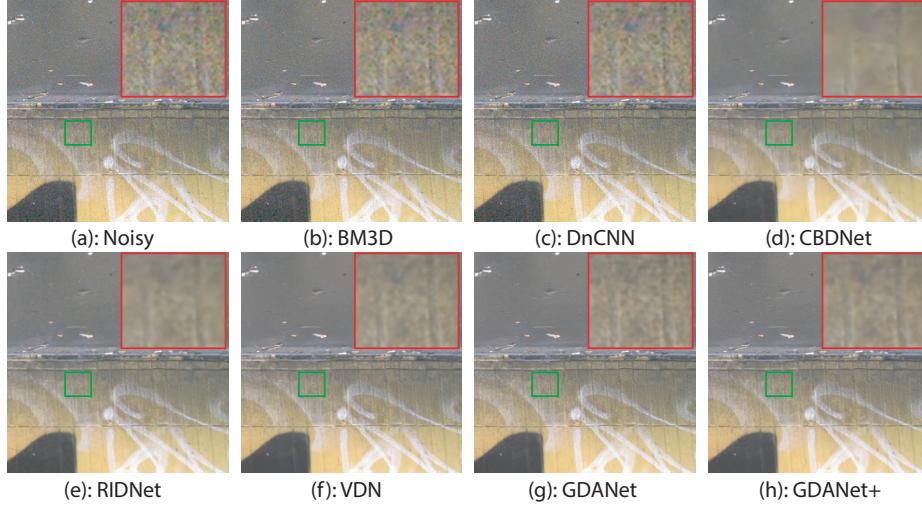
Metrics	Methods							
	CBM3D	WNNM	DnCNN	CBDNet	RIDNet	VDN	GDANet	GDANet <sub>+</sub>
PSNR↑	34.51	34.67	32.43	38.06	39.26	39.38	39.47	<b>39.58</b>
SSIM↑	0.8244	0.8646	0.7900	0.9421	0.9528	0.9518	<b>0.9548</b>	0.9545

The PGap and average AKLD results of DANet under different  $\tau_2$  values are shown in Table 4. Note that  $\tau_2 = +\infty$  represents the results of the generator  $G$  trained only with the regularizer of Eq. (6). Fig. 5 also shows the corresponding visual results of one typical example. As one can see,  $G$  fails to simulate the real noise with  $\tau_2 = 0$ , which demonstrates that the regularizer of Eq. (6) is able to stabilize the training of GAN. However, it is also difficult to train  $G$  only with the regularizer of Eq. (6) as shown in Fig. 5 (f). Taking both the quantitative and visual results into consideration,  $\tau_2$  is constantly set as 10 in our experiments.

**Ablation studies:** To verify the marginal benefits brought up by our dual adversarial loss, two groups of ablation experiments are designed in this part. In the first group, we train DANet without the generator and denote the trained model as BaseD. On the contrary, we train DANet without the denoiser and denote the trained model as BaseG. And the comparison results of these two baselines with DANet on noise removal and noise generation tasks are listed in Table 5 and Table 6, respectively. It can be easily seen that DANet achieves better performance than both the two baselines in noise removal and noise generation tasks, especially in the latter, which illustrates the mutual guidance and amelioration between the denoiser and the generator.

### 5.3 Results on DND and Nam Benchmarks

To evaluate the performance of our method in general real-world denoising tasks, we test on two real-world benchmarks, i.e., DND [36] and Nam [34]. These two benchmarks do not provide any training data, therefore they are suitable to test the generalization capability of any denoiser. Following the experimental setting in RIDNet [4], we trained another model using  $512 \times 512$  image patches from SIDD [1], Poly [51] and RENOIR [3] for fair comparison. To be distinguished from the model of Sec. 5.2, the trained models under this setting are denoted as GDANet and GDANet<sub>+</sub>, aiming at dealing with the general denoising task in real application. For the training of GDANet<sub>+</sub>, we employed the images of MIR Flickr [21] as clean images to synthesize more training pairs using  $G$ .

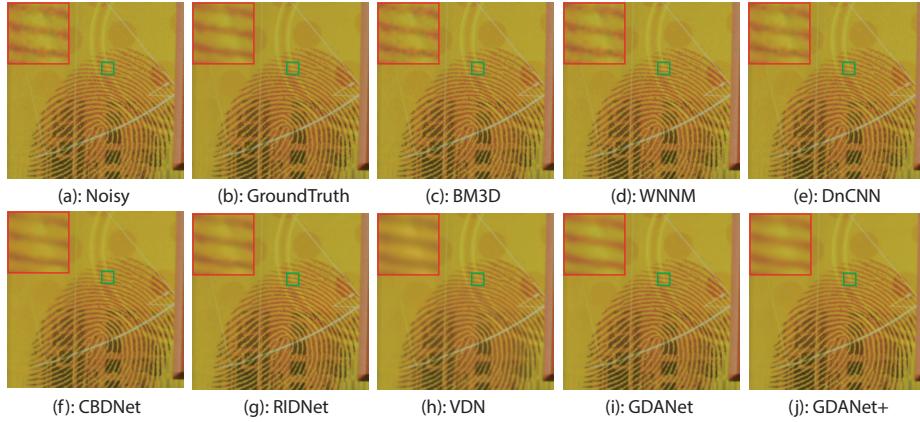
**Fig. 6.** Denoising results of different methods on DND benchmark.**Table 8.** The PSNR and SSIM results of different methods on the Nam benchmark. The best results are highlighted in bold.

Metrics	Methods							
	CBM3D	WNNM	DnCNN	CBDNet	RIDNet	VDN	GDANet	GDANet+
PSNR↑	35.36	35.33	35.68	39.20	39.33	38.66	<b>39.91</b>	39.79
SSIM↑	0.8708	0.8812	0.8811	0.9676	0.9623	0.9613	<b>0.9693</b>	0.9689

**DND Benchmark:** This benchmark contains 50 real noisy and almost noise-free image pairs. However, the almost noise-free images are not publicly released, thus the PSNR/SSIM results can only be obtained through online submission system. Table 7 lists the PSNR/SSIM results released on the official DND benchmark website<sup>8</sup>. From Table 7, we have the following observations: 1) GDANet<sub>+</sub> outperforms the state-of-the-art VDN about 0.2dB PSNR, which is a large improvement in the field of real-world denoising; 2) GDANet obtains the highest SSIM value, which means that it preserves more structural information than other methods as that can be visually observed in Fig. 6; 3) DnCNN cannot remove most of the real noise because it overfits to the Gaussian noise case; 4) the classical CBM3D and WNNM methods cannot handle the complex real noise.

**Nam Benchmark:** This benchmark contains 11 real static scenes and the corresponding noise-free images, which are obtained by averaging 500 noisy images of the same scenes. We cropped these images into  $512 \times 512$  patches, and randomly selected 100 of them for the purpose of evaluation. The quantitative PSNR and SSIM results are given in Table 8. It is easy to see that our proposed GDANet performs better than the other compared methods. Note that VDN does not achieve good performance since the noisy images in this benchmark are JPEG compressed, which is not considered in VDN. For easy comparison, we also dis-

<sup>8</sup> <https://noise.visinf.tu-darmstadt.de/benchmark/>



**Fig. 7.** One typical denoising example of Nam benchmark by different methods.

play one typical denoised example by different methods in Fig. 7, and the better visual performance of our methods can be observed.

**Discussion:** Different from the results in Sec. 5.2, GDANet performs more stably than GDANet<sub>+</sub> as shown in Table 7 and 8, especially on SSIM metric. That's because the noise types simulated by the generator  $G$ , which are mainly determined by the training data set, does not match well with that contained in the testing set. Therefore, GDANet is suggested to be used in such general real-world denoising task with uncertain noise types, while DANet<sub>+</sub> is more suitable in the scenario that provides similar training and testing data sets.

## 6 Conclusion

We have proposed a new Bayesian framework, namely dual adversarial network (DANet), for real-world image denoising. Different from the traditional MAP framework relied on subjective pre-assumptions on the noise and image priors, our proposed method focuses on learning the joint distribution directly from data. To estimate the joint distribution, we attempt to approximate it by its two different factorized forms using an dual adversarial manner, which corresponds to two tasks, i.e., noise removal and noise generation. For assessing the quality of synthetic noisy image, we have designed two applicable metrics, to the best of our knowledge, for the first time. The proposed DANet intrinsically provides a general methodology to facilitate the study of other low-level vision tasks, such as super-resolution and deblurring. Comprehensive experiments have demonstrated the superiority of DANet as compared with state-of-the-art methods specifically designed for both the noise removal and noise generation tasks.

**Acknowledgements:** This research was supported by National Key R&D Program of China (2018YFB1004300) and the China NSFC project under contracts 11690011, 61721002, U1811461, and Hong Kong RGC RIF grant (R5001-18).

## References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
2. Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: Advances in Neural Information Processing Systems 26. pp. 1493–1501 (2013)
3. Anaya, J., Barbu, A.: Renoir - a benchmark dataset for real noise reduction evaluation. arXiv preprint arXiv:1409.8230 (2014), <https://academic.microsoft.com/paper/1514812871>
4. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3155–3164 (2019)
5. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
6. Barbu, A.: Training an active random field for real-time image denoising. IEEE Transactions on Image Processing **18**(11), 2451–2462 (2009)
7. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11036–11045 (2019)
8. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 60–65. IEEE (2005)
9. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2392–2399 (2012)
10. Cao, X., Chen, Y., Zhao, Q., Meng, D., Wang, Y., Wang, D., Xu, Z.: Low-rank matrix factorization under general mixture noise distributions. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
11. Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3155–3164 (2018)
12. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1256–1272 (2017)
13. Chongxuan, L., Xu, T., Zhu, J., Zhang, B.: Triple generative adversarial nets. In: Advances in neural information processing systems. pp. 4088–4098 (2017)
14. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007)
15. Dong, W., Shi, G., Li, X.: Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE transactions on image processing **22**(2), 700–711 (2012)
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
17. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2862–2869 (2014)

18. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
19. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind de-noising of real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1712–1722 (2019)
20. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
21. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In: Proceedings of the international conference on Multimedia information retrieval. pp. 527–536 (2010)
22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
23. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: Advances in Neural Information Processing Systems 21. pp. 769–776 (2008)
24. Jaroensri, R., Biscarrat, C., Aittala, M., Durand, F.: Generating training data for denoising real rgb images via camera pipeline simulation. arXiv preprint arXiv:1904.08825 (2019)
25. Kim, D.W., Ryun Chung, J., Jung, S.W.: Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
26. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: ICLR 2015 : International Conference on Learning Representations 2015 (2015), <https://academic.microsoft.com/paper/2964121744>
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR 2014 : International Conference on Learning Representations (ICLR) 2014 (2014)
28. Lebrun, M., Buades, A., Morel, J.M.: A nonlocal bayesian image denoising algorithm. Siam Journal on Imaging Sciences **6**(3), 1665–1688 (2013)
29. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.: Non-local recurrent network for image restoration. In: NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems. pp. 1673–1682 (2018)
30. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 2272–2279 (2009)
31. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE Transactions on image processing **17**(1), 53–69 (2007)
32. Mao, X.J., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 2810–2818 (2016)
33. Meng, D., De La Torre, F.: Robust matrix factorization with unknown noise. In: The IEEE International Conference on Computer Vision (ICCV) (December 2013)
34. Nam, S., Hwang, Y., Matsushita, Y., Joo Kim, S.: 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. pp. 1683–1691 (2016)

35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
36. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1586–1595 (2017)
37. Plotz, T., Roth, S.: Neural nearest neighbors networks. In: NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems. pp. 1087–1098 (2018)
38. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR 2016 : International Conference on Learning Representations 2016 (2016), <https://academic.microsoft.com/paper/2963684088>
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015), <https://academic.microsoft.com/paper/1901129140>
40. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992)
41. Samuel, K.G.G., Tappen, M.F.: Learning optimized map estimates in continuously-valued mrf models. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 477–484 (2009)
42. Schmidt, U.: Half-quadratic inference and learning for natural images (2017)
43. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2774–2781 (2014)
44. Sun, J., Tappen, M.F.: Learning non-local range markov random field for image restoration. In: CVPR 2011. pp. 2745–2752 (2011)
45. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 4549–4557 (2017)
46. Tsin, Y., Ramesh, V., Kanade, T.: Statistical calibration of ccd imaging process. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 1, pp. 480–487. IEEE (2001)
47. Wang, R., Chen, B., Meng, D., Wang, L.: Weakly supervised lesion detection from fundus images. IEEE transactions on medical imaging **38**(6), 1501–1512 (2018)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
49. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems 25. pp. 341–349 (2012)
50. Xie, Q., Zhao, Q., Meng, D., Xu, Z.: Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery. IEEE transactions on pattern analysis and machine intelligence **40**(8), 1888–1902 (2017)
51. Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L.: Real-world noisy image denoising: A new benchmark. arXiv preprint arXiv:1804.02603 (2018)
52. Xu, J., Zhang, L., Zhang, D.: A trilateral weighted sparse coding scheme for real-world image denoising. In: The European Conference on Computer Vision (ECCV) (September 2018)

53. Xu, J., Zhang, L., Zhang, D., Feng, X.: Multi-channel weighted nuclear norm minimization for real color image denoising. ICCV (2017)
54. Yong, H., Meng, D., Zuo, W., Zhang, L.: Robust online matrix factorization for dynamic background subtraction. IEEE transactions on pattern analysis and machine intelligence **40**(7), 1726–1740 (2017)
55. Yue, Z., Yong, H., Meng, D., Zhao, Q., Leung, Y., Zhang, L.: Robust multiview subspace learning with nonindependently and nonidentically distributed complex noise. IEEE transactions on neural networks and learning systems (2019)
56. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational denoising network: Toward blind noise modeling and removal. In: Advances in Neural Information Processing Systems. pp. 1688–1699 (2019)
57. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
58. Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., Paisley, J.W.: Non-parametric bayesian dictionary learning for sparse image representations. In: Advances in neural information processing systems. pp. 2295–2303 (2009)
59. Zhu, F., Chen, G., Hao, J., Heng, P.A.: Blind image denoising via dependent dirichlet process tree. IEEE transactions on pattern analysis and machine intelligence **39**(8), 1518–1531 (2016)
60. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)