From Rain Generation to Rain Removal

Hong Wang[1,2,*], Zongsheng Yue[1,*], Qi Xie[1], Qian Zhao[1], Yefeng Zheng[2], Deyu Meng[3,1,†]

[1]Xi'an Jiaotong University, Xi'an, China
[2]Tencent Jarvis Lab, Shenzhen, China
[3]Macau University of Science and Technology, Macau, China

hongwang01@stu.xjtu.edu.cn   zsyzam@gmail.com   xie.qi@mail.xjtu.edu.cn

timmy.zhaoqian@gmail.com   yefengzheng@tencent.com   dymeng@mail.xjtu.edu.cn

Abstract

For the single image rain removal (SIRR) task, the performance of deep learning (DL)-based methods is mainly affected by the designed deraining models and training datasets. Most of current state-of-the-art focus on constructing powerful deep models to obtain better deraining results. In this paper, to further improve the deraining performance, we novelly attempt to handle the SIRR task from the perspective of training datasets by exploring a more efficient way to synthesize rainy images. Specifically, we build a full Bayesian generative model for rainy image where the rain layer is parameterized as a generator with the input as some latent variables representing the physical structural rain factors, e.g., direction, scale, and thickness. To solve this model, we employ the variational inference framework to approximate the expected statistical distribution of rainy image in a data-driven manner. With the learned generator, we can automatically and sufficiently generate diverse and non-repetitive training pairs so as to efficiently enrich and augment the existing benchmark datasets. User study qualitatively and quantitatively evaluates the realism of generated rainy images. Comprehensive experiments substantiate that the proposed model can faithfully extract the complex rain distribution that not only helps significantly improve the deraining performance of current deep single image derainers, but also largely loosens the requirement of large training sample pre-collection for the SIRR task. Code is available in https://github.com/hongwang01/VRGNet.

1. Introduction

Recently, single image rain removal (SIRR) has attracted considerable attention, which is usually regarded as a necessary pre-processing step of outdoor image processing tasks,

(a) The flowchart of rain generation
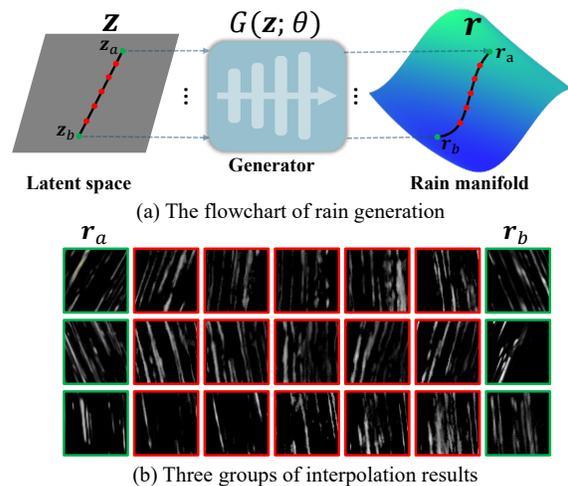

(b) Three groups of interpolation results

Figure 1. Interpolation results in latent space z representing rain factors. (a) The rain distribution is implicitly modeled as a generator G; (b) Three groups of generated rain layers through interpolations in latent space. For each group, r_a and r_b (marked as green) represent the rain layers in the original training dataset, while the ones (marked as red) between them are generated from latent codes (marked as red points) in z space. These codes are obtained by linearly interpolating between z_a and z_b which are the latent codes of r_a and r_b, respectively.

e.g., autonomous driving [14], scene segmentation [5], and object tracking [6]. Due to the complex and diverse rain structures in real scenes, SIRR is still a typical challenging task in computer vision [33, 45, 54].

Driven by massive training data (rainy images) and the powerful fitting capability of deep convolutional neural network (CNN), deep learning (DL) represents the current research trend in the SIRR task. Clearly, the performance of DL-based methods is mainly affected by two key factors, i.e., the rationality and capacity of deraining models and the quality of training datasets. Most of current works focus on the former and aim to improve the deraining results mainly by building more sophisticated networks [8, 10, 22, 29, 35, 39, 41, 43, 44, 49, 53, 55, 58] and designing

---

better learning manners [23, 31, 37, 46, 48, 50, 56]. Albeit achieving satisfied performance in some scenarios, they put less emphasis on the impact of training data and largely rely on the off-the-shelf datasets to train their deraining models. Curiously, are the existing datasets sufficiently good? Is it possible to further improve the performance of current DL-based derainers directly by ameliorating the quality of these datasets? This paper mainly concentrates on these issues.

Currently, for the SIRR task, the existing datasets are mainly obtained by the following manners: 1) The common one is to synthesize rain streaks with the photo-realistic rendering technique [12] and then add them on clear images [10, 33, 36, 53, 58, 59]. 2) Instead of such simple addition operation, inspired by [13], some works [18, 20] explored better fusion mechanisms between clear images and rain streaks. However, the exploited rains are still synthesized by manually setting some oscillation parameters of raindrops [12]. 3) The unpaired image translation strategy is another new generation manner, which attempts to learn a mapping from a clean image to rainy one with adversarial learning so as to generate paired rainy-clean images, such as [38, 51, 62]. 4) There is one real rain dataset proposed by [49], which is semi-automatically generated through rain videos shot in real rain scenes by manually adjusting camera parameters, including exposure duration and ISO.

Although these existing datasets can be used to train deep derainers to some extent, their generation manners still possess some evident limitations. Specifically, for 1) and 2), rains are synthesized by empirically setting some parameters through human subjective assumptions, which would restrict the generated rain types. Besides, the acquisition process of training samples needs human supervision and physical simulators. This is time-consuming and labor-cumbersome. As for 3), the intrinsic mechanisms of rains are more or less ignored and thus it has less physical interpretability. While for 4), it is always hard to shoot enough rain scenes for sufficiently representing the complicated rain shapes in real world. All these deficiencies tend to adversely affect the quality and the diversity of training datasets and limit the performance improvement of current deep SIRR derainers. Thus, it is critical to build a proper model representing the rain statistical distribution in order to automatically and faithfully generate diverse rains.

In this work, we attempt to explore the intrinsic generative mechanism underlying rain streaks and propose a better generation process. As seen in Fig. 1, high quality of rain streaks, with diverse and non-repetitive shapes, can be easily obtained through the learned generator G by our method that represents the implicit distribution of rains. It is worth mentioning that the generated rain streaks r (marked as red in Fig. 1(b)) exhibit more unseen patterns in the original training dataset r_a and r_b (marked as green). Especially, such generator with an explicit mapping form tends to

provide intrinsic clues for understanding the generation of rains, which is meaningful for general tasks on rainy images. In summary, our contributions are mainly three-fold:

Firstly, this work specifically proposes a generative model to depict the generation process of a rainy image. Specifically, different from hand-crafted priors for rains [16,50,57] or physics-based imaging analysis about rains [13], the proposed model makes effort to explore an implicit distribution of rain layer in statistics. A deep variational inference algorithm is specifically designed to to approximate the expected distribution of rainy images.

Secondly, an interpretable rain generator can be obtained, capable of delivering the intrinsic manifold projection from latent factors, such as direction and thickness, to rain streaks as shown in Fig. 1. This makes it possible to efficiently generate diverse and non-repetitive rain streaks without subjective human intervention and empirical parameter settings. Disentanglement and interpolation experiments substantiate the rationality of the proposed generator, and a user study evaluates the realism of generated rainy images. Moreover, the small sample experiment exhibits the potentials of the proposed model in real applications.

Thirdly, the proposed generator facilitates an easy augmentation of diverse rains for current DL-based SIRR derainers. Comprehensive experiments on synthetic and real datasets validate that the performance of these DL-based derainers can be significantly improved by retraining them on augmented datasets. This coincides with our motivation that improving the quality of datasets is rational and helpful.

2. Related Work

Rain Dataset Synthesizing. Previously, Garg and Nayar analyzed the appearance and imaging process of rain [13] and synthesized a rain streak database with the photo-realistic rendering technique [12]. Similarly, researchers synthesize different rain streaks and then add them on clear images to construct paired samples such as Rain100H [53], Rain1400 [10], Rain800 [59], and DID-MDN [58]. Besides, there are some works exploring how to merge rains with background images, for example, RainCityscapes [20], NYU-Rain [31], and MPID [33]. Recently, the unpaired image translation idea is widely adopted to generate weather-corrupted images [38, 51, 62]. These generation methods often require setting model parameters, which would limit the diversity of synthesized rains.

Instead of synthesizing rains, Wang *et al.* [49] proposed a large-scale real rain dataset, called SPA-Data, which was semi-automatically generated from real rain videos shot in real rain scenes or collected from Internet. To construct paired samples, the clean images are roughly estimated based on successive several frames. The main limitation of this dataset is that the expensive cost of shooting rain scenes makes it difficult to capture large number of rain types.

Rain Removal. Very recently, for the SIRR task, researchers have designed various network structures, from simple CNN [9, 10] to complicated recurrent and multi-stage learning [35, 41, 52, 53]. Besides, some works incorporate multi-scale learning to exploit the self-similarity both within the same scale or across different scales [11, 22, 55, 60]. There are also some other network frameworks, for example, adversarial learning [31, 32, 51, 58, 59], encoder-decoder [8, 20, 29, 44, 47], and semi-/un-supervised learning [23, 50, 51, 56, 62]. Now there is another novel research line that prior knowledge is embedded into deep networks to improve the interpretability, such as [37, 46, 48, 50].

Although these DL-based techniques have achieved remarkable success, they mainly utilize the aforementioned off-the-shelf datasets as training data. In this work, we aim to explore an automatic generative mechanism with the capability to simulate possibly variant rain types, for ameliorating the quality of the existing datasets and thus expectantly improving the deraining results of current deep derainers.

Generative Models. As an active research topic in computer vision and machine learning, deep generative models have been widely studied recently, such as variational autoencoder (VAE) [27, 42], generative adversarial network (GAN) [15, 40], and flow-based generative model [7]. Especially, as prominent models, VAE and GAN have achieved remarkable success in many image generation tasks, including face modeling [28, 34], style transfer [63], image noise generation [3, 24] and so on. To the best of our knowledge, there is still little work completely focusing on the rain generation task. Therefore, inspired by these deep generative models, we take a step forward to explore the intrinsic generative mechanisms of rainy images as well as rain streaks.

3. The Proposed Method

Given a training set $\mathcal{D} = \{o_n, x_n\}_{n=1}^{N}$, where o_n is the n-th rainy image and x_n is the background, we aim to explore the physical mechanism of the rainy image and learn its underlying distribution. To this aim, we construct a generative model for rainy image under the Bayesian framework by implicitly modeling rain layer as a generator. With our specifically designed inference algorithm, the model can extract the general statistical distribution of rainy image as well as rain streak based on the training dataset in a data-driven manner. This enables the free generation of rains with diverse shapes. The details are given below.

3.1. Generative Model

Similar to [41, 48, 49], given any single rainy image $o \in \mathbb{R}^d$ with size d as height × width, the generation process is:

$$o = r + b, \tag{1}$$

where r and b denote the rain layer and latent clean background underlying o, respectively. Therefore, the generation of rainy image decomposes into two parts as follows:

Rain Modeling. For rain layer r, it is difficult to depict it by using an accurate distribution in statistics. But it is very intuitive that the appearance of rain can be represented by some evident latent factors, such as direction, scale, and thickness [30, 48, 58]. Motivated by this observation, we encode such physical structural factors underlying rains as latent variable $z \in \mathbb{R}^t$ and generally model the rain layer r as a deep generator conditioned on z, i.e.,

$$r = G(z; \theta), \tag{2}$$

where θ denotes the parameters of the generator G.

As suggested in [2, 25], the isotropic Gaussian prior distribution is imposed on z as:

$$z \sim \mathcal{N}(z|\mathbf{0}, \mathbf{I}_t), \tag{3}$$

where $\mathbf{I}_t \in \mathbb{R}^{t \times t}$ is the unit matrix. Such a prior has the potential to disentangle the physical rain factors in z. This is visually validated in Section 5.1.

Background Modeling. In the given training pairs, the rain-free image x is usually simulated or estimated based on multiple rainy images taken on the same condition like real SPA-Data [49], and it is not the exact latent clean background b. We thus embed x into the following Gaussian prior distribution to constrain b as:

$$b \sim \mathcal{N}(b|x, \varepsilon_0^2 \mathbf{I}_d), \tag{4}$$

where ε_0^2 is a hyper-parameter measuring the similarity between x and b, and can be easily set as a small value. Note that for synthetic data where rainy images are obtained by adding synthesized rains on the pre-collected images [10, 36, 53], x can be regarded as the true groundtruth b. In this case, Dirac prior on b is a proper choice which can be well approximated by setting ε_0^2 close to 0 in Eq. (4).

From Eqs. (1)-(4), it is easy to derive a full Bayesian model. Specifically, for rainy image o, the likelihood is:

$$o \sim p_\theta(o|z, b). \tag{5}$$

Note that $p_\theta(\cdot)$ means that this implicit distribution also relies on the parameters θ of generator G defined in Eq. (2).

Finally, the task of generating rainy image turns to learn the general statistical distribution $p(o)$, expressed as:[1]

$$p(o) = \int \int p_\theta(o|z, b) \, p(z) \, p(b) \, dz db, \tag{6}$$

where $p(z)$ and $p(b)$ are the prior distributions of z and b, corresponding to Eq. (3) and Eq. (4), respectively.

Since the integral in Eq. (6) is intractable, next we adopt the variational Bayesian framework to learn the $p(o)$.

3.2. Variational Objective

To learn $p(o)$, we can decompose its logarithm as [1]:[2]

$$\log p(o) = \mathcal{L}(z, b; o) + D_{KL}[q(z, b|o) \| p(z, b|o)], \tag{7}$$

---

[1]Here we assume that z and b are mutually independent.
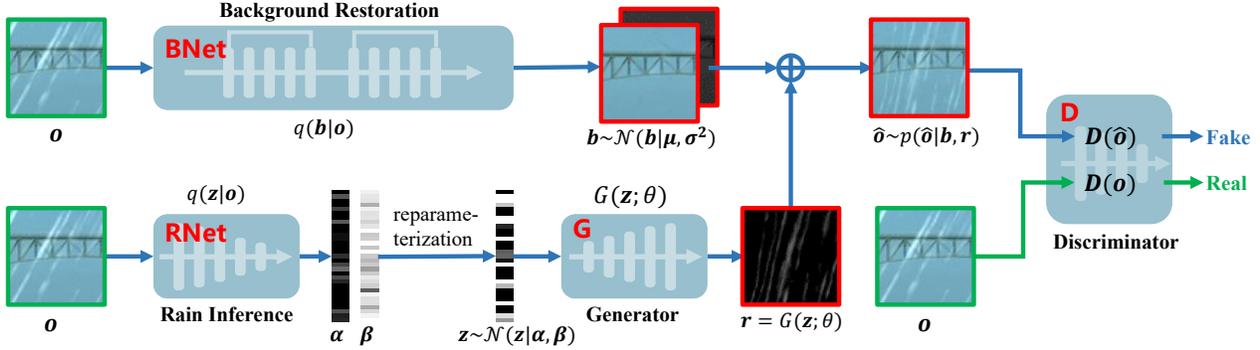[2]More derivations are included in supplementary material (SM).

Figure 2. The flowchart of the proposed variational rain generation network (VRGNet). It contains four sub-networks, which are correspondingly constructed based on the estimation of $\log p(\boldsymbol{o})$, i.e., $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$ in Eq. (11).

where the first term in Eq. (7) is expressed as:

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o}) = E_{q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})}[\log p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b})p(\boldsymbol{z})p(\boldsymbol{b}) - \log q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})] .$$

(8)

Here $E_{p(a)}[f(a)]$ is the expectation of function $f(a)$ about the stochastic variable a with the probability density function $p(a)$. $q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$ is the variational approximate posterior of the true posterior $p(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$ about the latent variables $\boldsymbol{z}$ and $\boldsymbol{b}$. The second term in Eq. (7) is the KL divergence measuring the difference between $q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$ and $p(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$. The non-negative property of KL divergence can thus lead to the following inequality, i.e.,

$$\log p(\boldsymbol{o}) \geq \mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o}) .$$

(9)

Thus the variational lower bound $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$ can be viewed as an estimation of $\log p(\boldsymbol{o})$ with error as the KL divergence. The learning of $p(\boldsymbol{o})$ can be achieved through approaching $\log p(\boldsymbol{o})$ by maximizing its estimation $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$.

Based on the analysis above, once θ is optimized by maximizing $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$, the obtained explicit mapping $G(\boldsymbol{z}; \theta)$ can be directly used to synthesize rainy image, i.e., $\boldsymbol{o} = G(\boldsymbol{z}; \theta) + \boldsymbol{b}$, where $\boldsymbol{z}$ and $\boldsymbol{b}$ are sampled from $p(\boldsymbol{z})$ and $p(\boldsymbol{b})$, respectively. Therefore, for this generation task, the key problem is how to maximize the $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$ in Eq. (8).

3.3. Optimization

Now we give the optimization algorithm for maximizing the $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$. From Eq. (8), the key is to deal with the variational posterior $q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$ and the implicit $p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b})$.

As for $q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})$, like the commonly-used factorized hypothesis in the mean-field variational inference [27], we introduce the conditional independence assumption as:

$$q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o}) = q(\boldsymbol{z}|\boldsymbol{o}) q(\boldsymbol{b}|\boldsymbol{o}) .$$

(10)

Then, the $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$ in Eq. (8) can be equally rewritten as:

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o}) = E_{q(\boldsymbol{z}, \boldsymbol{b}|\boldsymbol{o})}[\log p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b})]$$
$$- D_{KL}[q(\boldsymbol{z}|\boldsymbol{o}) \| p(\boldsymbol{z})] - D_{KL}[q(\boldsymbol{b}|\boldsymbol{o}) \| p(\boldsymbol{b})] .$$

(11)

To maximize $\mathcal{L}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o})$ in Eq. (11), we impose Gaussian distribution on the posteriors $q(\boldsymbol{z}|\boldsymbol{o})$ and $q(\boldsymbol{b}|\boldsymbol{o})$ to approach the Gaussian priors $p(\boldsymbol{z})$ and $p(\boldsymbol{b})$, respectively, i.e.,

$$q(\boldsymbol{z}|\boldsymbol{o}) = \prod_{i=1}^{t} \mathcal{N}(z_i | \alpha_i(\boldsymbol{o}; W_R), \beta_i(\boldsymbol{o}; W_R)), \quad (12)$$

$$q(\boldsymbol{b}|\boldsymbol{o}) = \prod_{j=1}^{d} \mathcal{N}(b_j | \mu_j(\boldsymbol{o}; W_B), \sigma_j^2(\boldsymbol{o}; W_B)), \quad (13)$$

where $\alpha_i(\boldsymbol{o}; W_R)$ and $\beta_i(\boldsymbol{o}; W_R)$ are functions for inferring the posterior parameters (i.e., mean and variance, respectively) of latent variable $\boldsymbol{z}$, and they are integrally parameterized as one rain inference network, called *RNet* with parameter W_R. $\mu_j(\boldsymbol{o}; W_B)$ and $\sigma_j^2(\boldsymbol{o}; W_B)$ are functions from $\boldsymbol{o}$ to variational posterior parameters of latent variable $\boldsymbol{b}$. They are jointly parameterized as another network, called *BNet* with parameter W_B for restoring clean background.

From Eqs. (3), (4), (12), and (13), it is easy to compute the last two terms in Eq. (11) as:

$$D_{KL}[q(\boldsymbol{z}|\boldsymbol{o}) \| p(\boldsymbol{z})] = \sum_{i=1}^{t} \left\{ \frac{\alpha_i^2}{2} + \frac{1}{2}(\beta_i - \log \beta_i - 1) \right\},$$

$$D_{KL}[q(\boldsymbol{b}|\boldsymbol{o}) \| p(\boldsymbol{b})] = \sum_{j=1}^{d} \left\{ \frac{(\mu_j - x_j)^2}{2\varepsilon_0^2} + \frac{1}{2}\left(\frac{\sigma_j^2}{\varepsilon_0^2} - \log \frac{\sigma_j^2}{\varepsilon_0^2} - 1 \right) \right\}.$$

(14)

where we simplify $\alpha_i(\boldsymbol{o}; W_R)$, $\beta_i(\boldsymbol{o}; W_R)$, $\mu_j(\boldsymbol{o}; W_B)$, and $\sigma_j^2(\boldsymbol{o}; W_B)$, as α_i, β_i, μ_j, and σ_j^2, respectively.

However, we cannot directly calculate the first term in Eq. (11) due to the implicity of $p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b})$. Fortunately, the generator G enables the sampling from $p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b})$, i.e.,

$$\boldsymbol{o} \sim p_\theta(\boldsymbol{o}|\boldsymbol{z}, \boldsymbol{b}) \iff \boldsymbol{o} = G(\boldsymbol{z}; \theta) + \boldsymbol{b}, \quad (15)$$

which motivates us to introduce a *discriminator* D with parameter W_D to approximate the first term in Eq. (11) by the following two-player game [15]:

$$\min_G \max_D \mathcal{L}_{adv}(\boldsymbol{z}, \boldsymbol{b}) = E_{\boldsymbol{o} \sim p_{\text{data}}}[D(\boldsymbol{o})]$$
$$- E_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{o}), \boldsymbol{b} \sim q(\boldsymbol{b}|\boldsymbol{o})}[D(G(\boldsymbol{z}; \theta) + \boldsymbol{b})].$$

(16)

Thus, from Eqs. (14) and (16), we can reformulate the negative lower bound in Eq. (11) as follows:

$$\widehat{\mathcal{L}}(\boldsymbol{z}, \boldsymbol{b}; \boldsymbol{o}) = \gamma \mathcal{L}_{adv}(\boldsymbol{z}, \boldsymbol{b}) + D_{KL}[q(\boldsymbol{z}|\boldsymbol{o}) \| p(\boldsymbol{z})]$$
$$+ D_{KL}[q(\boldsymbol{b}|\boldsymbol{o}) \| p(\boldsymbol{b})],$$

(17)

where γ is a hyper-parameter controlling the importance between the adversarial loss and KL divergence. The value is set empirically and will be explained in experiment section.

Algorithm 1 Variational Inference for Rain Generation
___
Input: Training data $\mathcal{D}=\{\boldsymbol{o}_n, \boldsymbol{x}_n\}_{n=1}^N$, batch size n_b, n_{critic} times updating of D for every updating $BNet$, $RNet$, and G
Output: Network parameters $W = \{W_B, W_R, \theta, W_D\}$
1: **while** The loss in Eq. (18) is not convergent **do**
2:     **for** $m = 1$ to n_{critic} **do**
3:         $\{\boldsymbol{o}, \boldsymbol{x}\} \leftarrow$ SampleMiniBatch$(\mathcal{D}, n_b)$.
4:         $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\} \leftarrow RNet(\boldsymbol{o}; W_R)$.
5:         $\boldsymbol{z} \leftarrow$ Reparameterization$(\boldsymbol{\alpha}, \boldsymbol{\beta})$.
6:         $\{\boldsymbol{\mu}, \boldsymbol{\sigma}^2\} \leftarrow BNet(\boldsymbol{o}; W_B)$.
7:         $\boldsymbol{b} \leftarrow$ Reparameterization$(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.
8:         $\widehat{\boldsymbol{o}} \leftarrow G(\boldsymbol{z}; \theta) + \boldsymbol{b}$.
9:         Update D with fixed $BNet$, $RNet$, and G.
10:     **end for**
11:     Update $BNet$ with fixed $RNet$, D, and G.
12:     Update $RNet$ and G with fixed $BNet$ and D.
13: **end while**
___

From the analysis above, learning the generative process of rainy image is closely related to the minimization of $\widehat{\mathcal{L}}(\boldsymbol{b}, \boldsymbol{z}; \boldsymbol{o})$. For optimizing the involved network parameters W_B, W_R, θ, and W_D, the total objective function on the entire training dataset, can be formulated as:

$$\sum_{n=1}^N \widehat{\mathcal{L}}(\boldsymbol{z}_n, \boldsymbol{b}_n; \boldsymbol{o}_n). \quad (18)$$

Note that during training, W_B, W_R, θ and W_D are shared across the entire training data, leading to a general statistical distribution modelling for rainy image as well as rain layer.

Based on Eqs. (12), (13), (16), and (17), we can easily construct the inference framework as shown in Fig. 2, called variational rain generation network (VRGNet).[3]

4. Implementation Details

Training Strategy. The entire framework in Fig. 2 is first jointly trained based on the loss function in Eq. (18). The whole training procedure is summarized as Algorithm 1, where we adopt the gradient penalty strategy for D to stabilize the adversarial learning [17].

After obtaining the rain generator G, we can use it to automatically generate sufficient rain streaks, by taking $\boldsymbol{z}$ sampled from normal distribution as the input of G. Based on the augmented training dataset, including original and generated pairs, we retrain current representative DL-based derainers so as to further improve their performance (see Section 6). It is noteworthy that the augmentation operation is implemented on the original dataset, without introducing extra training pairs requiring pre-collecting groundtruth.

Training Details. During the joint training, the entire network in Fig. 2 is optimized by the Adam algorithm [26]. The initial learning rates for $BNet$, $RNet$, G, and D are 2×10^{-4}, 1×10^{-4}, 1×10^{-4}, and 4×10^{-4}, respectively, and divided by 2 at epochs [400, 600, 650, 675, 690, 700]. The

___
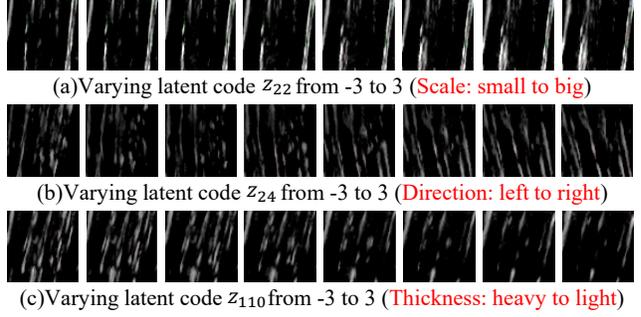[3]More details can be found in supplementary material.

(a)Varying latent code z_{22} from -3 to 3 (Scale: small to big)

(b)Varying latent code z_{24} from -3 to 3 (Direction: left to right)

(c)Varying latent code z_{110} from -3 to 3 (Thickness: heavy to light)

Figure 3. Manipulating latent code $\boldsymbol{z} \in \mathbb{R}^{128}$. Taking subfigure (a) as an example, we sample a random vector (latent code $\boldsymbol{z}$) from the normal distribution, and then only vary the latent element at the 22-th dimension of $\boldsymbol{z}$ from -3 to 3 with the interval as 0.8. Taking each varied vector $\boldsymbol{z}$ as the input of the generator G, the corresponding output $\boldsymbol{r}$ is each rain layer shown in (a), which demonstrates the scale property of rain. (a)-(c) denote varying different latent elements and the learned latent variables physically represent scale, direction, and thickness, respectively.

different initialized learning rate settings for G and D are inspired by [19]. The prior hyper-parameter ε_0^2 is set as 1×10^{-6} and the dimension t of latent variable $\boldsymbol{z}$ is 128. In each epoch, the batch size n_b is set as 18, and we randomly crop 18×3000 patches with size 64×64 pixels from the rainy image $\boldsymbol{o}$ in $\mathcal{D}$ for training. As suggested in [17], the penalty coefficient in WGAN-GP is 10, and n_{critic} is 5, meaning that we update D 5 times for each updating of $BNet$, $RNet$, and G. The coefficient γ in Eq. (17) is empirically set as 1 for synthetic datasets and 0.01 for SPA-Data.

5. Rain Generation Experiments

We first conduct disentanglement and interpolation analysis to verify the potential of the VRGNet in extracting physical structural rain factors, and then evaluate the perceptual realism of our synthetic rain. Besides, with small sample experiments, we finely substantiate the effectiveness of our model in compactly capturing the manifold of rain.

5.1. Disentanglement and Interpolation

Similar to [2, 4, 25], we manipulate the latent code $\boldsymbol{z}$ and the disentanglement results are displayed in Fig. 3, where the proposed VRGNet is trained on Rain100L [53]. From it, we can easily observe that these latent variables well represent interpretable physical properties in characterizing rain, including scale, direction, and thickness. Clearly, the proposed VRGNet has the capability of discovering meaningful latent rain factors, which finely complies with our latent variable modelling for rain layer in Eq. (2).

Besides, we also conduct interpolation operations in the latent space as shown in Fig. 1 (b). The results validate that our rain generator possesses the manifold continuity in the latent space for changing the direction and thickness of rains, and thus it can generate diverse and non-repetitive
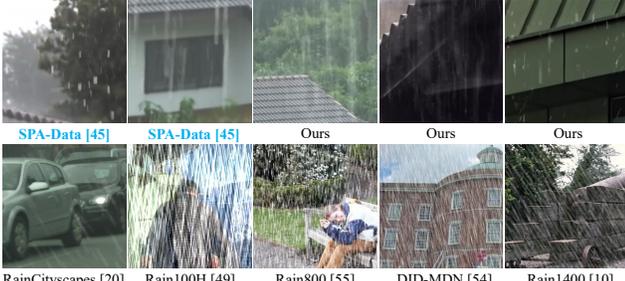
Figure 4. Rainy images randomly selected from seven different datasets. Only SPA-Data [49] is captured in real rain scenes.
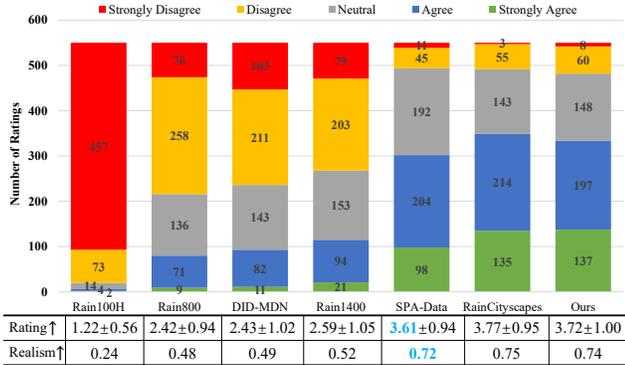

	Rain100H	Rain800	DID-MDN	Rain1400	SPA-Data	RainCityscapes	Ours
Rating↑	1.22±0.56	2.42±0.94	2.43±1.02	2.59±1.05	3.61±0.94	3.77±0.95	3.72±1.00
Realism↑	0.24	0.48	0.49	0.52	0.72	0.75	0.74

Figure 5. User study results. Upper figure: the ratings given by all participants on various datasets. Lower table: (1st row) the mean and standard deviation of the ratings; (2nd row) the realism computed by converting the mean rating to the [0,1] interval.

rain types instead of simply memorizing the patterns in input images. More results as video clips are provided in SM.

5.2. User Study

Fig. 4 displays the visual comparisons of rainy images randomly selected from 7 different datasets, including SPA-Data [49] captured in real rain scenes by controlling camera parameters, the samples randomly generated by the proposed VRGNet trained on SPA-Data, RainCityscapes [20] generated based on a rain streak database [12], and the other 4 synthetic datasets for rain removal, i.e., Rain1400 [10], DID-MDN [58], Rain800 [59], and Rain100H [53]. As seen, our synthetic rains have better diversity and their appearances look closer to the real SPA-Data.

We further conduct a user study to quantitatively evaluate the quality (i.e., how realistic) of the generated rain streaks. Specifically, we prepare for 70 rainy images randomly selected from these 7 datasets with 10 samples from each dataset. Then, we recruit 55 participants with 14 females and 41 males. For each participant, we present him/her the 70 rainy images in a random order. Then they are asked to rate how real every image is, using a 5-point Likert scale. Finally, we get 550 ratings for each category.

Results are reported in Fig. 5, showing that our synthetic rain is judged to be significantly more realistic than most of SOTA datasets. Besides, there are three points to clari-

fy: 1) Owning to the good diversity of generated rains (see Fig. 3 and Fig. 4), the ratings of our synthesized rainy images even outperform SPA-Data. 2) The realism of the synthetic RainCityscapes is slightly better than ours. However, RainCityscapes focus on modelling the fusion process of the pre-collected background layer and rain streaks that are synthesized by manually setting some model parameters [12], and our method is for learning an interpretable generator to synthesize diverse rain streaks. From the perspective of rain layer, our method has a better capability to synthesize more diverse rain streaks than RainCityscapes (see Fig. 4). 3) As compared with SPA-Data and RainCityscapes, our method is able to automatically generate more sufficient and diverse rain patterns without any human intervention and empirical parameter settings, which is helpful for improving the deraining performance (see Section 6).

As seen, our method is mainly limited by directly adopting the commonly-used addition operation in Eq. (1) between rain and background layer to generate rainy image. To synthesize more realistic rainy images, it is worth further exploring how to combine our rain generator and the fusion mechanism of RainCityscapes in the future.

5.3. Small Sample Experiments on Real SPA-Data

To further verify that our generator is able to efficiently generate more non-repetitive and diverse rain patterns, we conduct a small sample experiment on real SPA-Data with $\sim$600K training pairs and 1K test pairs. Specifically, we randomly select 1K pairs from the training set and augment them with ratio N_f (i.e., generate N_fK fake pairs) for training. Meanwhile, we also randomly choose **the same number** (i.e., 1K+N_fK) of real pairs all from the original SPA-Data and take this case as a baseline. Due to its simplicity and fast training speed, we adopt the latest PReNet [41] as the deep derainer to implement this experiment.

Table 1 reports the PSNR averaged over 5 repetitions for different augmentation ratios.[4] From it, we can observe that with the increase of ratio N_f from 0 to 6, the average PSNR under augmented training is superior (N_f = 2, 3, 4, 5, 6) or at least comparable (N_f = 0, 1) to the performance (40.16 dB) under original training based on the $\sim$600K real pairs. This is mainly attributed to two points: 1) In SPA-Data, the rain scenes are not sufficiently collected to cover complicated shapes of rain streaks and many pairs are obtained by cropping one rain video shot in the same scene, which both lead to the repeatability of rain patterns. 2) The proposed generator learns the rain distribution in SPA-Data and thus can efficiently generate possible non-repetitive and diverse rain types that more compactly scatter on the manifold of such rain distribution. This also tells that the learned generator can loosen the requirement on pre-collected training

---

[4]More peak-signal-to-noise raito (PSNR) [21] and structure similarity (SSIM) [61] results are listed in SM.

Table 1. Average PSNR of PReNet on the SPA-Data test set. **Baseline** denotes that training samples are all from SPA-Data (∼600K), and **GNet** means the augmented training where training samples consist of 1K real pairs randomly selected from ∼600K and different number of fake pairs generated by our generator that is jointly trained on ∼600K. **In each scene, the training pairs between Baseline and GNet keep the same**, and the result is computed over 5 random attempts. The case that **Baseline** with ∼600K has no randomness about samples.

# Real samples	1K	1.5K	2K	3K	4K	5K	6K	7K	∼600K
Baseline (PSNR), mean±std	39.41±0.24	39.70±0.21	**39.86**±0.20	39.96±0.20	40.05±0.19	40.04±0.18	40.00±0.18	40.06±0.15	40.16
# Samples (real+fake)	1K+0K	1K+0.5K	1K+1K	1K+2K	1K+3K	1K+4K	1K+5K	1K+6K	-
GNet (PSNR), mean±std	39.41±0.24	**39.71**±0.26	39.83±0.20	**40.25**±0.21	**40.24**±0.17	**40.53**±0.20	**40.68**±0.17	**40.70**±0.11	-

Table 2. PSNR and SSIM comparisons on synthetic datasets. "+" denotes the augmented training. △↑ represents the performance gain brought by the augmented rains generated by our rain generator. **Note that the baseline of one method "A+" is "A"**.

Methods		Input	DSC	JCAS	DDN	DDN+	△↑	SPANet	SPANet+	△↑	PReNet	PReNet+	△↑	JORDER_E	JORDER_E+	△↑
Rain100L	PSNR	26.90	27.34	28.54	32.38	35.56	3.18	35.33	35.83	0.50	37.42	37.74	0.32	37.68	38.01	0.33
	SSIM	0.838	0.849	0.852	0.926	0.966	0.040	0.969	0.972	0.003	0.979	0.980	0.001	0.979	0.980	0.001
Rain100H	PSNR	13.56	13.77	14.62	22.85	25.99	3.14	25.11	27.24	2.13	30.11	30.42	0.31	30.50	32.16	1.66
	SSIM	0.371	0.312	0.451	0.725	0.797	0.072	0.833	0.883	0.050	0.905	0.910	0.005	0.897	0.921	0.024
Rain1400	PSNR	25.24	27.88	26.20	28.45	30.07	1.62	29.85	30.14	0.29	32.21	32.51	0.30	32.00	32.55	0.55
	SSIM	0.810	0.839	0.847	0.889	0.917	0.028	0.915	0.927	0.012	0.943	0.945	0.002	0.935	0.946	0.011

Input 14.72 / 0.352   DSC 17.56 / 0.390   DDN 25.49 / 0.782   SPANet 23.50 / 0.809   PReNet 28.96 / 0.901   JORDER_E 29.13 / 0.884

Groundtruth Inf / 1   JCAS 16.67 / 0.514   DDN+ 26.36 / 0.769   SPANet+ 27.87 / 0.869   PReNet+ 29.29 / 0.911   JORDER_E+ 32.10 / 0.934

Figure 6. **Vertical contrast**. Performance comparison on a test image from Rain100H, including rainy image/groundtruth, derained results from DSC/JCAS, and deep SOTAs trained on the original (1st row) / augmented (2nd row) Rain100H training set.

samples, which should be meaningful for real applications.

6. Rain Removal Experiments

Similar to the augmented strategy in [18], we now utilize the generator to augment the existing datasets with ratio 0.5 so as to further improve the deraining performance of current deep derainers on synthetic and real rain datasets. More experiments as well as ablation studies are included in SM.

6.1. Evaluation on Synthetic Data

Representative Methods and Datasets. We evaluate the effectiveness of the augmentation strategy benefitted from VRGNet through latest DL-based SIRR methods, including DDN [10], PReNet [41], SPANet [49], and JORDER_E [53], based on common synthetic datasets, including Rain100L [53], Rain100H [53], and Rain1400 [10]. In the followings, we use notation "A+" to denote the results of the method A after being retrained on the augmented dataset. We also list the performance of model-based DSC [57] and JCAS [16] for comprehensive comparisons.

Deraining Results. Table 2 lists the quantitative performance of all competing methods. As seen, the deraining performance of every deep derainer after augmented training is significantly improved on all datasets and the gain △↑ far outperforms the sensitivity value of human visual system (about 0.1 dB). This strongly confirms that the generated

rains indeed ameliorate original training sets and thus further improve the performance of DL-based methods. Naturally, the gain △↑ varies among different deep derainers, which is mainly caused by their different model capacities.

Fig. 6 illustrates the visual deraining results on one hard sample from Rain100H. For every DL-based method, when trained on augmented dataset generated by VRGNet, its reconstructed background (2nd row) has better visual quality, especially in texture preservation, than the corresponding one (1st row) trained on original Rain100H. Clearly, the VRGNet has the potential to generate rains with higher quality and better diversity.

6.2. Generalization Evaluation on Real Data

We further verify the role of the generated rains in helping improve the robustness of all these deep derainers to rainy images in real-world, based on two real datasets both from [49], i.e., SPA-Data and Internet-Data (no label).

Comparisons on SPA-Data. Table 3 quantitatively compares the generalization performance on SPA-Data where all deep derainers are trained on Rain100L. In original training, we can find that the generalization performance of all deep methods is not optimistic since the domain gap between Rain100L and SPA-Data is extremely large. Even under such a challenging scenario, after augmented training, the performance of all these methods has

Table 3. Generalization performance on the test data of SPA-Data. All the DL-based methods are trained on Rain100L. The rain patterns between Rain100L and SPA-Data are quite different, which makes the generalization task hard.

Methods	Input	DSC	JCAS	DDN	DDN+	△↑	SPANet	SPANet+	△↑	PReNet	PReNet+	△↑	JORDER_E	JORDER_E+	△↑
PSNR	34.15	34.83	34.95	34.66	35.01	0.35	35.13	35.42	0.29	34.91	35.13	0.22	35.04	35.15	0.11
SSIM	0.927	0.941	0.945	0.935	0.943	0.008	0.944	0.948	0.004	0.940	0.942	0.002	0.941	0.942	0.001

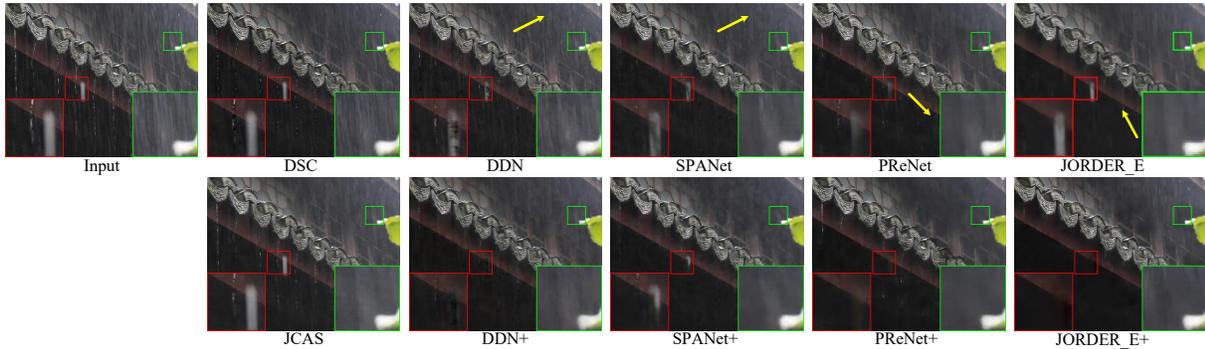

Figure 7. **Vertical contrast**. Generalization results on a real image from Internet-Data. All DL-based methods are trained on SPA-Data.

(a) Original samples (b) Discrete points covering small part of the unknown manifold (c) Resampling on learned manifold (d) Sampled rain layers
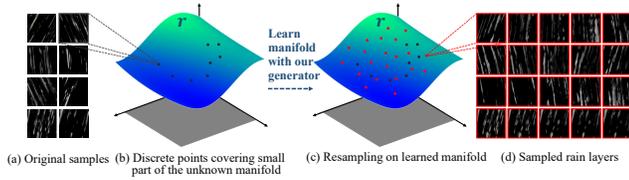
Figure 8. The diagram about the insights of the proposed method.

been improved to some extent. Note that due to larger network capacity (parameters), JORDER_E is easier to fall into the overfitting issue and thus the performance gain △↑ is lower. Besides, owing to the dual influence of network structure and the quality of training set, the improvement room △↑ for every method is different.

Comparisons on Internet-Data. Fig. 7 shows the generalization performance on a real rainy image from Internet-Data. Under such a complex rain scene not seen in SPA-Data, these DL-based methods with augmented training evidently remove heavy rains (see the amplified red boxes). This can be rationally attributed to the diversity of generated rain types. Note that since the Internet-Data has no groundtruth, here we only provide the visual comparisons.

7. More Discussions about VRGNet

Under the VAE framework, we have approached the complicated rain distribution by our model. Fig. 8 shows the underlying insights. Specifically, the original rains can be regarded as samples generated from the high dimensional distribution, only covering a small discrete part of the entire manifold surface underlying this distribution. With the learned generator that reflects such distribution implicitly, we can easily sample sufficient and diverse rain patterns to obtain an augmented training dataset, being able to more compactly cover this manifold. Using such augmented dataset, the derainers are expected to more completely capture the rain information so as to achieve better results.

However, there are still some limitations. First, due to

the inherent nature of VAE, we can only implicitly control the rain intensity, but cannot perfectly disentangle completely independent and interpretable latent codes strictly corresponding to physical factors. Admittedly, the ideal case is that we can explicitly control the rain and generate the types we want. Second, although the proposed generator can generate diverse rain samples, it does not contain any more information than that of the original training set. The effectiveness of the proposed method can be further substantiated through some downstream tasks. Third, when the domain gap between training set and testing set is very large, our method might possibly fail to help current deep single image derainers achieve evident improvement.

8. Conclusion

In this paper, we have explored the rain generative mechanism and constructed a full Bayesian model for generating rains from latent factors representing physical structural rain factors, such as direction, scale, and thickness. To solve this model, we have proposed a variational rain generation network (VRGNet), which implicitly infers the general statistical distribution of rains in a data-driven manner. From the learned generator, rain patches can be automatically generated to simulate diverse training samples, which facilitates a beneficial augmentation and enrichment of the existing benchmark dataset. Comprehensive rain generation verifications have fully substantiated the rationality of our generative model and evaluated the realism of the generated rain both qualitatively and quantitatively. Moreover, rain removal experiments implemented on synthetic and real datasets have finely validated the effectiveness of our generated rains in helping significantly improve the robustness of current deep single image derainers to rains in real world.

References

[1] David M Blei and Michael I Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006. 3

[2] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018. 3, 5

[3] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming 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. 3

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016. 5

[5] Chang Cheng, Andreas Koschan, Chung-Hao Chen, David L Page, and Mongi A Abidi. Outdoor scene image segmentation based on background recognition and perceptual organization. *IEEE Transactions on Image Processing*, 21(3):1007–1019, 2011. 1

[6] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003. 1

[7] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 3

[8] Yingjun Du, Jun Xu, Qiang Qiu, Xiantong Zhen, and Lei Zhang. Variational image deraining. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2406–2415, 2020. 1, 3

[9] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 3

[10] Xueyang Fu, Jiabin Huang, Delu Zeng, Huang Yue, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017. 1, 2, 3, 6, 7

[11] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. 3

[12] Kshitiz Garg and Shree K Nayar. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics (TOG)*, 25(3):996–1002, 2006. 2, 6

[13] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 2

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. pages 3354–3361, 2012. 1

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3, 4

[16] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1708–1716, 2017. 2, 7

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 5

[18] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10203–10212, 2019. 2, 7

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 5

[20] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019. 2, 3, 6

[21] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6

[22] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. *arXiv preprint arXiv:2003.10985*, 2020. 1, 3

[23] Xin Jin, Zhibo Chen, Jianxin Lin, Zhikai Chen, and Wei Zhou. Unsupervised single image deraining with self-supervised constraints. In *IEEE International Conference on Image Processing*, pages 2761–2765. IEEE, 2019. 2, 3

[24] Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. 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*, pages 0–0, 2019. 3

[25] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018. 3, 5

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[27] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3, 4

[28] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2018. 3

[29] Guanbin Li, He Xiang, Zhang Wei, Huiyou Chang, and Lin Liang. Non-locally enhanced encoder-decoder network for single image de-raining. In *ACM Multimedia Conference*, 2018. 1, 3

[30] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018. 3

[31] Ruoteng Li, Loongfah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1633–1642, 2019. 2, 3

[32] Ruoteng Li, Robby T. Tan, and Loong Fah Cheong. All in one bad weather removal using architectural search. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[33] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2019. 1, 2

[34] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. pages 278–296, 2018. 3

[35] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision*, pages 254–269, 2018. 1, 3

[36] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2736–2744, 2016. 2, 3

[37] Pan Mu, Jian Chen, Risheng Liu, Xin Fan, and Zhongxuan Luo. Learning bilevel layer priors for single image rain streaks removal. *IEEE Signal Processing Letters*, 26(2):307–311, 2019. 2, 3

[38] Fabio Pizzati, Pietro Cerri, and Raoul de Charette. Model-based occlusion disentanglement for image-to-image translation. In *European Conference on Computer Vision*, 2020. 2

[39] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018. 1

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[41] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu 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, 3, 6, 7

[42] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014. 3

[43] Mingwen Shao, Le Li, Hong Wang, and Deyu Meng. Selective generative adversarial network for raindrop removal from a single image. *Neurocomputing*, 426:265–273, 2021. 1

[44] Guoqing Wang, Changming Sun, and Arcot Sowmya. Erlnet: Entangled representation learning for single image deraining. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5644–5652, 2019. 1, 3

[45] Hong Wang, Yichen Wu, Minghan Li, Qian Zhao, and Deyu Meng. A survey on rain removal from video and single image. *arXiv preprint arXiv:1909.08326*, 2019. 1

[46] Hong Wang, Yichen Wu, Qi Xie, Qian Zhao, Yong Liang, Shijun Zhang, and Deyu Meng. Structural residual learning for single image rain removal. *Knowledge-Based Systems*, page 106595, 2020. 2, 3

[47] Hong Wang, Qi Xie, Yichen Wu, Qian Zhao, and Deyu Meng. Single image rain streaks removal: a review and an exploration. *International Journal of Machine Learning and Cybernetics*, 11(4):853–872, 2020. 3

[48] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3103–3112, 2020. 2, 3

[49] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019. 1, 2, 3, 6, 7

[50] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2019. 2, 3

[51] Yanyan Wei, Zhao Zhang, Jicong Fan, Yang Wang, Shuicheng Yan, and Meng Wang. Deraincyclegan: An attention-guided unsupervised benchmark for single image deraining and rainmaking. *arXiv preprint arXiv:1912.07015*, 2019. 2, 3

[52] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 3

[53] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2019. 1, 2, 3, 5, 6, 7

[54] Wenhan Yang, Robby T. Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):1–1, 2020. 1

[55] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning CNN for single image de-raining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8405–8414, 2019. 1, 3

[56] Rajeev Yasarla, Vishwanath A Sindagi, and Vishal M Patel. Syn2real transfer learning for image deraining using Gaussian processes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2726–2736, 2020. 2, 3

[57] Luo Yu, Xu Yong, and Ji Hui. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015. 2, 7

[58] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 695–704, 2018. 1, 2, 3, 6

[59] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 2019. 2, 3, 6

[60] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. In *Conference on British Machine Vision Conference*, 2019. 3

[61] Wang Zhou, Bovik Alan Conrad, Sheikh Hamid Rahim, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[62] Hongyuan Zhu, Xi Peng, Joey Tianyi Zhou, Songfan Yang, Vijay Chanderasekh, Liyuan Li, and Joo-Hwee Lim. Single image rain removal with unpaired information: A differentiable programming perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9332–9339, 2019. 2, 3

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 3