From Rain Generation to Rain Removal

Hong Wang\textsuperscript{1,2,*}, Zongsheng Yue\textsuperscript{1,*}, Qi Xie\textsuperscript{1}, Qian Zhao\textsuperscript{1}, Yefeng Zheng\textsuperscript{2}, Deyu Meng\textsuperscript{3,1,†}
\textsuperscript{1}Xi’an Jiaotong University, Xi’an, China
\textsuperscript{2}Tencent Jarvis Lab, Shenzhen, China
\textsuperscript{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 images 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 images 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, 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

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{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.}
\end{figure}
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 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 networks (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 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 \( D = \{ o_n, x_n \}_{n=1}^{N} \), 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 \( \times \) 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 \( \theta \) denotes the parameters of the generator \( G \).

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

\[
z \sim N(z|0, I_t), \tag{3}
\]

where \( I_t \in \mathbb{R}^{t \times t} \) 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:

\[
b \sim N(b|x, \sigma_0^2 I_d), \tag{4}
\]

where \( \sigma_0^2 \) 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 \( \sigma_0^2 \) 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:

\[
o \sim p_h(o|z, b), \tag{5}
\]

Note that \( p_h(\cdot) \) means that this implicit distribution also relies on the parameters \( \theta \) 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:

\[
p(o) = \int \int p_h(o|z, b) p(z) p(b) dz db, \tag{6}
\]

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]:

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

\footnote{Here we assume that \( z \) and \( b \) are mutually independent.}

\footnote{More derivations are included in supplementary material (SM).}
where the first term in Eq. (7) is expressed as:

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

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

$$\log p(o) \geq L(z; b; o).$$

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

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

### 3.3. Optimization

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

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

$$q(z, b|o) = q(z|o)q(b|o).$$

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

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

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

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

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

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

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

$$D_{KL}[q(b|o)||p(b)] = \sum_{j=1}^{d} \left\{ \frac{(\mu_j - x_j)^2}{2\sigma_0^2} + \frac{1}{2} \left( \frac{\sigma_j^2}{\sigma_0^2} - \log \frac{\sigma_j^2}{\sigma_0^2} - 1 \right) \right\}.$$

where we simplify $\alpha_i(o; W_R)$, $\beta_i(o; W_R)$, $\mu_j(o; W_B)$, and $\sigma_j^2(o; W_B)$, as $\alpha_i$, $\beta_i$, $\mu_j$, and $\sigma_j^2$, respectively.

However, we cannot directly calculate the first term in Eq. (11) due to the implicity of $p_0(o|z, b)$, i.e.,

$$o \sim p_0(o|z, b) \iff o = G(z; θ) + b,$$

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]:

$$\min_D \max_G \mathcal{L}_{adv}(z, b) = E_{o \sim p_{data}}[D(o)] - E_{z \sim q(z|o), b \sim q(b|o)}[D(G(z; θ) + b)].$$

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

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

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

$$\sum_{n=1}^{N} \mathcal{L}(z_n, b_n; \alpha_n).$$

(18)

Note that during training, $W_B, W_R, \theta$ 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 $z$ 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 $2 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}$, and $4 \times 10^{-4}$, respectively, and divided by 2 at epochs [400, 600, 650, 675, 690, 700].

Algorithm 1 Variational Inference for Rain Generation

Input: Training data $D = \{o_n, x_n\}_{n=1}^{N}$, batch size $n_b$, $n_{\text{critic}}$ times updating of $D$ for every updating $BNet, RNet, G$.

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 $n_{\text{critic}}$ do
3:     $\{\alpha, x\} \leftarrow \text{SampleMiniBatch}(D, n_b)$.
4:     $\{\alpha, \beta\} \leftarrow \text{RNet}(\alpha; W_R)$.
5:     $z \leftarrow \text{Reparameterization}(\alpha, \beta)$.
6:     $\mu, \sigma^2 \leftarrow \text{BNet}(\alpha; W_B)$.
7:     $b \leftarrow \text{Reparameterization}(\mu, \sigma^2)$.
8:     $\hat{o} = G(z; \theta) + b$.
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

³More details can be found in supplementary material.

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 $z$ 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
Fig. 4. Rainy images randomly selected from seven different datasets. Only SPA-Data [49] is captured in real rain scenes.

Fig. 5. User study results. Upper figure: the ratings given by all participants on various datasets. Lower table: (1\textsuperscript{st} row) the mean and standard deviation of the ratings; (2\textsuperscript{nd} 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 clarify:

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 ~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. 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 ~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.

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>Input</th>
<th>DSC</th>
<th>JCAS</th>
<th>DDN</th>
<th>DDN+</th>
<th>SPA-Net</th>
<th>SPA-Net+</th>
<th>JCAS</th>
<th>JORDER</th>
<th>JORDER+</th>
<th>△↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>34.15</td>
<td>34.83</td>
<td>34.95</td>
<td>34.66</td>
<td>35.13</td>
<td>35.52</td>
<td>34.91</td>
<td>35.13</td>
<td>0.39</td>
<td>35.04</td>
<td>0.11</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.927</td>
<td>0.941</td>
<td>0.945</td>
<td>0.943</td>
<td>0.944</td>
<td>0.948</td>
<td>0.940</td>
<td>0.942</td>
<td>0.002</td>
<td>0.941</td>
<td>0.001</td>
</tr>
</tbody>
</table>

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

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, owning 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.

Acknowledgment. This research was supported by National Key R&D Program of China (2020YFA0713900) and the China NSFC projects 11690011, 61721002, U1811461.
References


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


