Not Just Streaks: Towards Ground Truth for Single Image Deraining

Yunhao Ba[1][*], Howard Zhang[1][*], Ethan Yang[1], Akira Suzuki[1], Arnold Pfahnl[1], Chethan Chinder Chandrappa[1], Celso M. de Melo[2], Suya You[2], Stefano Soatto[1], Alex Wong[3], and Achuta Kadambi[1]

[1] University of California, Los Angeles
{yhba,hwdz15508,eyang657,asuzuki100,ajpfahnl,chinderc}@ucla.edu
soatto@cs.ucla.edu, achuta@ee.ucla.edu
[2] DEVCOM Army Research Laboratory
{celso.m.demelo.civ,suya.you.civ}@army.mil
[3] Yale University
alex.wong@yale.edu

Abstract. We propose a large-scale dataset of real-world rainy and clean image pairs and a method to remove degradations, induced by rain streaks and rain accumulation, from the image. As there exists no real-world dataset for deraining, current state-of-the-art methods rely on synthetic data and thus are limited by the sim2real domain gap; moreover, rigorous evaluation remains a challenge due to the absence of a real paired dataset. We fill this gap by collecting a real paired deraining dataset through meticulous control of non-rain variations. Our dataset enables paired training and quantitative evaluation for diverse real-world rain phenomena (e.g. rain streaks and rain accumulation). To learn a representation robust to rain phenomena, we propose a deep neural network that reconstructs the underlying scene by minimizing a rain-robust loss between rainy and clean images. Extensive experiments demonstrate that our model outperforms the state-of-the-art deraining methods on real rainy images under various conditions. Project website: https://visual.ee.ucla.edu/gt_rain.htm/.

Keywords: Single-image rain removal, Real deraining dataset

1 Introduction

Single-image deraining aims to remove degradations induced by rain from images. Restoring rainy images not only improves their aesthetic properties, but also supports reuse of abundant publicly available pretrained models across computer vision tasks. Top performing methods use deep networks, but suffer from a common issue: it is not possible to obtain ideal real ground-truth pairs of rain and clean images. The same scene, in the same space and time, cannot be observed both with and without rain. To overcome this, deep learning based rain removal relies on synthetic data.
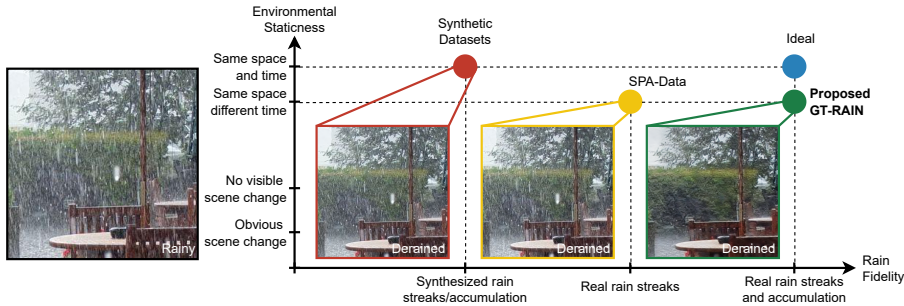
---

[*] Equal contribution.

Fig. 1. The points above depict datasets and their corresponding outputs from models trained on them. These outputs come from a real rain image from the Internet. Our opinion* is that GT-RAIN can be the right dataset for the deraining community to use because it has a smaller domain gap to the ideal ground truth. * Why an asterisk? The asterisk emphasizes that this is an "opinion". It is impossible to quantify the domain gap because collecting true real data is infeasible. To date, deraining is largely a viewer's imagination of what the derained scene should look like. Therefore, we present the derained images above and leave it to the viewer to judge the gap. Additionally, GT-RAIN can be used in complement with the litany of synthetic datasets [12,19,27,29,50,56,57], as illustrated in Table 4.

The use of synthetic data in deraining is prevalent [12,19,27,29,50,56,57]. However, current rain simulators cannot model all the complex effects of rain, which leads to unwanted artifacts when applying models trained on them to real-world rainy scenes. For instance, a number of synthetic methods add *rain streaks* to clean images to generate the pair [12,29,50,56,57], but rain does not only manifest as streaks: If raindrops are further away, the streaks meld together, creating *rain accumulation*, or *veiling* effects, which are exceedingly difficult to simulate. A further challenge with synthetic data is that results on real test data can only be evaluated qualitatively, for no real paired ground truth exists.

Realizing these limitations of synthetic data, we tackle the problem from another angle by relaxing the concept of ideal ground truth to a sufficiently short time window (see Fig. 1). We decide to conduct the experiment of obtaining short time interval paired data, particularly in light of the timely growth and diversity of landscape YouTube live streams. We strictly filter such videos with objective criteria on illumination shifts, camera motions, and motion artifacts. Further correction algorithms are applied for subtle variations, such as slight movements of foliage. We call this dataset GT-RAIN, as it is a first attempt to provide real paired data for deraining. Although our dataset relies on streamers, YouTube's fair use policy allows its release to the academic community.

Defining "real, paired ground truth": Clearly, obtaining real, paired ground truth data by capturing a rain and rain-free image pair at the exact same space and time is not feasible. However, the dehazing community has accepted several

test sets [1,2,3,4] following these guidelines as a satisfactory replacement for evaluation purposes:

– A pair of degraded and clean images is captured as real photos at two different timestamps;
– Illumination shifts are limited by capturing data on cloudy days;
– The camera configuration remains identical while capturing the degraded and clean images.

We produce the static pairs in GT-RAIN by following the above criterion set forth by the dehazing community while enforcing a stricter set of rules on sky and local motion. More importantly, as a step closer towards obtaining real ground truth pairs, we capture natural weather effects instead, which address problems of scale and variability that inherently come with simulating weather through man-made methods. In the results of the proposed method, we not only see quantitative and qualitative improvements, but also showcase a unique ability to handle diverse rain physics that was not previously handled by synthetic data.

Contributions: In summary, we make the following contributions:

– We propose a real-world paired dataset: GT-RAIN. The dataset captures *real* rain phenomena, from rain streaks to accumulation under various rain fall conditions, to bridge the domain gap that is too complex to be modeled by synthetic [12,19,27,29,50,56,57] and semi-real [44] datasets.
– We introduce an avenue for the deraining community to now have standardized quantitative and qualitative evaluations. Previous evaluations were quantifiable only wrt. simulations.
– We propose a framework to reconstruct the underlying scene by learning representations robust to the rain phenomena via a rain-robust loss function. Our approach outperforms the state of the art [55] by 12.1% PSNR on average for deraining real images.

2 Related Work

Rain physics: Raindrops exhibit diverse physical properties while falling, and many experimental studies have been conducted to investigate them, i.e. equilibrium shape [5], size [35], terminal velocity [10,14], spatial distribution [34], and temporal distribution [58]. A mixture of these distinct properties transforms the photometry of a raindrop into a complex mapping of the environmental radiance which considers refraction, specular reflection, and internal reflection [13]:

$$L(\hat{n}) = L_r(\hat{n}) + L_s(\hat{n}) + L_p(\hat{n}), \tag{1}$$

where $L(\hat{n})$ is the radiance at a point on the raindrop surface with normal $\hat{n}$, $L_r(\cdot)$ is the radiance of the refracted ray, $L_s(\cdot)$ is the radiance of the specularly reflected ray, and $L_p(\cdot)$ is the radiance of the internally reflected ray. In real images,

Table 1. Our proposed large-scale dataset enables paired training and quantitative evaluation for real-world deraining. We consider SPA-Data [44] as a semi-real dataset since it only contains real rainy images, where the pseudo ground-truth images are synthesized from a rain streak removal algorithm.

Dataset	Type	Rain Effects	Size
Rain12 [29]	Simulated	Synth. streaks only	12
Rain100L [50]	Simulated	Synth. streaks only	300
Rain800 [57]	Simulated	Synth. streaks only	800
Rain100H [50]	Simulated	Synth. streaks only	1.9K
Outdoor-Rain [27]	Simulated	Synth. streaks & Synth. accumulation	10.5K
RainCityscapes [19]	Simulated	Synth. streaks & Synth. accumulation	10.62K
Rain12000 [56]	Simulated	Synth. streaks only	13.2K
Rain14000 [12]	Simulated	Synth. streaks only	14K
NYU-Rain [27]	Simulated	Synth. streaks & Synth. accumulation	16.2K
SPA-Data [44]	Semi-real	Real streaks only	29.5K
Proposed	Real	Real streaks & Real accumulation	31.5K

the appearance of rain streaks is also affected by motion blur and background intensities. Moreover, the dense rain accumulation results in sophisticated veiling effects. Interactions of these complex phenomena make it challenging to simulate realistic rain effects. Until GT-RAIN, previous works [15,20,22,27,42,44,55] have relied heavily on simulated rain and are limited by the sim2real gap.

Deraining datasets: Most data-driven deraining models require paired rainy and clean, rain-free ground-truth images for training. Due to the difficulty of collecting real paired samples, previous works focus on synthetic datasets, such as Rain12 [29], Rain100L [50], Rain100H [50], Rain800 [57], Rain12000 [56], Rain14000 [12], NYU-Rain [27], Outdoor-Rain [27], and RainCityscapes [19]. Even though synthetic images from these datasets incorporate some physical characteristics of real rain, significant gaps still exist between synthetic and real data [51]. More recently, a "paired" dataset with real rainy images (SPA-Data) was proposed in [44]. However, their "ground-truth" images are in fact a product of a video-based deraining method – synthesized based on the temporal motions of raindrops which may introduce artifacts and blurriness; moreover, the associated rain accumulation and veiling effects are not considered. In contrast, we collect pairs of real-world rainy and clean ground-truth images by enforcing rigorous selection criteria to minimize the environmental variations. To the best of our knowledge, our dataset is the first large-scale dataset with real paired data. Please refer to Table 1 for a detailed comparison of the deraining datasets.

Single-image deraining: Previous methods used model-based solutions to derain [7,23,29,33]. More recently, deep-learning based methods have seen increasing popularity and progress [11,15,20,22,27,38,39,42,44,50,55,56]. The multi-scale
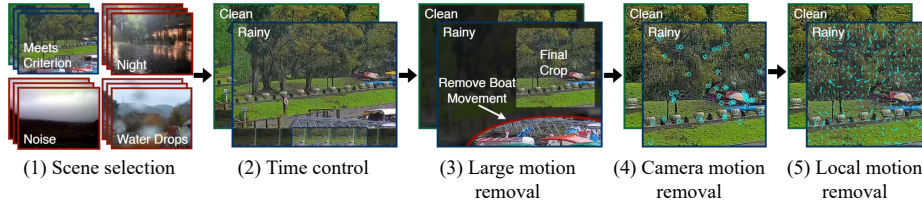
(1) Scene selection   (2) Time control   (3) Large motion removal   (4) Camera motion removal   (5) Local motion removal

Fig. 2. We collect the a real paired deraining dataset by rigorously controlling the environmental variations. First, we remove heavily degraded videos such as scenes without proper exposure, noise, or water droplets on the lens. Next, we carefully choose the rainy and clean frames as close as possible in time to mitigate illumination shifts before cropping to remove large movement. Lastly, we correct for small camera motion (due to strong wind) using SIFT [31] and RANSAC [9] and perform elastic image registration [40,41] by estimating the displacement field when necessary.

progressive fusion network (MSPFN) [22] characterizes and reconstructs rain streaks at multiple scales. The rain convolutional dictionary network (RCD-Net) [42] encodes the rain shape using the intrinsic convolutional dictionary learning mechanism. The multi-stage progressive image restoration network (MPR-Net) [55] splits the image into different sections in various stages to learn contextualized features at different scales. The spatial attentive network (SPANet) [44] learns physical properties of rain streaks in a local neighborhood and reconstructs the clean background using non-local information. EfficientDeRain (EDR) [15] aims to derain efficiently in real time by using pixel-wise dilation filtering. Other than rain streak removal, the heavy rain restorer (HRR) [27] and the depth-guided non-local network (DGNL-Net) [20] have also attempted to address rain accumulation effects. All of these prior methods use synthetic or semi-real datasets, and show limited generalizability to real images. In contrast, we propose a derainer that learns a rain-robust representation directly.

3   Dataset

We now describe our method to control variations in a real dataset of paired images taken at two different timestamps, as illustrated in Fig. 2.

Data collection: We collect rain and clean ground-truth videos using a Python program based on FFmpeg to download videos from YouTube live streams across the world. For each live stream, we record the location in order to determine whether there is rain according to the OpenWeatherMap API [32]. We also determine the time of day to filter out nighttime videos. After the rain stops, we continue downloading in order to collect clean ground-truth frames. Note: while our dataset is formatted for single-image deraining, it can be re-purposed for video deraining as well by considering the timestamps of the frames collected.
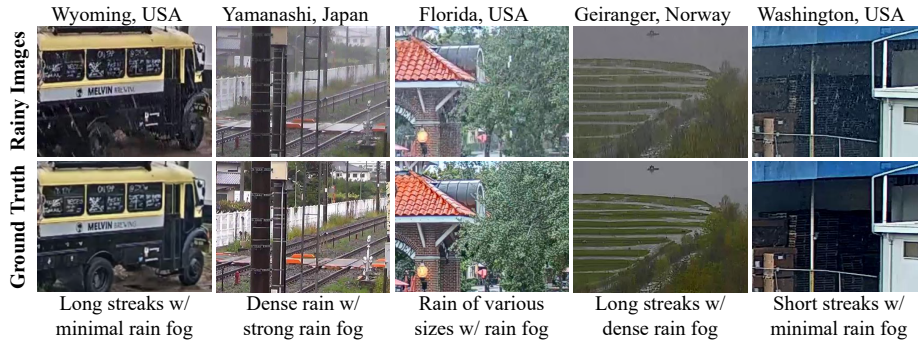
| Wyoming, USA | Yamanashi, Japan | Florida, USA | Geiranger, Norway | Washington, USA |

Rainy Images / **Ground Truth**

| Long streaks w/ minimal rain fog | Dense rain w/ strong rain fog | Rain of various sizes w/ rain fog | Long streaks w/ dense rain fog | Short streaks w/ minimal rain fog |

Fig. 3. Our proposed dataset contains diverse rainy images collected across the world. We illustrate several representative image pairs with various rain streak appearances and rain accumulation strengths at different geographic locations.

Collection criteria: To minimize variations between rainy and clean frames, videos are filtered based on a strict set of collection criteria. Note that we perform realignment for camera and local motion only when necessary – with manual oversight to filter out cases where motion still exists after realignment. Please see examples of motion correction and alignment in the supplement.

- **Heavily degraded scenes** that contain excessive noise, webcam artifacts, poor resolution, or poor camera exposure are filtered out as the underlying scene cannot be inferred from the images.
- **Water droplets** on the surface of the lens occlude large portions of the scene and also distort the image. Images containing this type of degradation are filtered out as it is out of the scope of this work – we focus on rain streak and rain accumulation phenomena.
- **Illumination shifts** are mitigated by minimizing the time difference between rainy and clean frames. Our dataset has an average time difference of 25 minutes, which drastically limits large changes in global illumination due to sun position, clouds, etc.
- **Background changes** containing large discrepancies (e.g cars, people, swaying foliage, water surfaces) are cropped from the frame to ensure that clean and rainy images are aligned. By limiting the average time difference between scenes, we also minimize these discrepancies before filtering. All sky regions are cropped out as well to ensure proper background texture.
- **Camera motion.** Adverse weather conditions, i.e. heavy wind, can cause camera movements between the rainy and clean frames. To address this, we use the Scale Invariant Feature Transform (SIFT) [31] and Random Sample Consensus (RANSAC) [9] to compute the homography to realign the frames.
- **Local motion.** Despite controlling for motion whenever possible, certain scenes still contain small local movements that are unavoidable, especially in areas of foliage. To correct for this, we perform elastic image registration when necessary by estimating the displacement field [40,41].
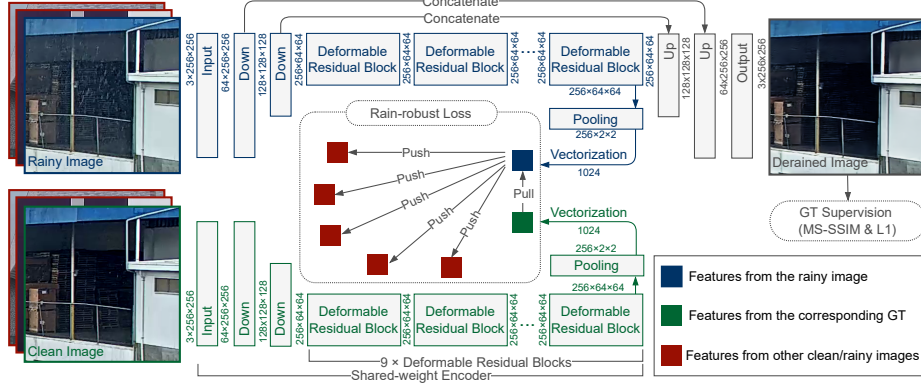
Fig. 4. By minimizing a rain-robust objective, our model learns robust features for reconstruction. When training, a shared-weight encoder is used to extract features from rainy and ground-truth images. These features are then evaluated with the rain-robust loss, where features from a rainy image and its ground-truth are encouraged to be similar. Learned features from the rainy images are also fed into a decoder to reconstruct the ground-truth images with MS-SSIM and $\ell 1$ loss functions.

Dataset statistics: Our large-scale dataset includes a total of 31,524 rainy and clean frame pairs, which is split into 26,124 training frames, 3,300 validation frames, and 2,100 testing frames. These frames are taken from 101 videos, covering a large variety of background scenes from urban locations (e.g. buildings, streets, cityscapes) to natural scenery (e.g. forests, plains, hills). We span a wide range of geographic locations (e.g. North America, Europe, Oceania, Asia) to ensure that we capture diverse scenes and rain fall conditions. The scenes also include varying degrees of illumination from different times of day and rain of varying densities, streak lengths, shapes, and sizes. The webcams cover a wide array of resolutions, noise levels, intrinsic parameters (focal length, distortion), etc. As a result, our dataset captures diverse rain effects that cannot be accurately reproduced by SPA-Data [44] or synthetic datasets [12,19,27,29,50,56,57]. See Fig. 3 for representative image pairs in GT-RAIN.

4   Learning to Derain Real Images

To handle greater diversity of rain streak appearance, we propose to learn a representation (illustrated in Fig. 4) that is robust to rain for real image deraining.

Problem formulation: Most prior works emphasize on the rain streak removal and rely on the following equation to model rain [8,12,26,29,42,44,52,56,61]:

$$\mathbf{I} = \mathbf{J} + \sum_{i}^{n} \mathbf{S}_i, \tag{2}$$

where $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$ is the observed rainy image, $\mathbf{J} \in \mathbb{R}^{3 \times H \times W}$ is the rain-free or "clean" image, and $\mathbf{S}_i$ is the i-th rain layer. However, real-world rain can be more complicated due to the dense rain accumulation and the rain veiling effect [27,28,49]. These additional effects, which are visually similar to fog and mist, may cause severe degradation, and thus their removal should also be considered for single-image deraining. With GT-RAIN, it now becomes possible to study and conduct optically challenging, real-world rainy image restoration.

Given an image $\mathbf{I}$ of a scene captured during rain, we propose to learn a function $\mathcal{F}(\cdot, \theta)$ parameterized by θ to remove degradation induced by the rain phenonmena. This function is realized as a neural network (see Fig. 4) that takes as input a rainy image $\mathbf{I}$ and outputs a "clean" image $\hat{\mathbf{J}} = \mathcal{F}(\mathbf{I}, \theta) \in \mathbb{R}^{3 \times H \times W}$, where undesirable characteristics, i.e. rain streaks and rain accumulation, are removed from the image to reconstruct the underlying scene $\mathbf{J}$.

Rain-robust loss: To derain an image $\mathbf{I}$, one may directly learn a map from $\mathbf{I}$ to $\hat{\mathbf{J}}$ simply by minimizing the discrepancies between $\hat{\mathbf{J}}$ and the ground truth $\mathbf{J}$, i.e. an image reconstruction loss – such is the case for existing methods. Under this formulation, the model must explore a large hypothesis space, e.g. any region obfuscated by rain streaks is inherently ambiguous, making learning difficult.

Unlike previous works, we constrain the learned representation such that it is robust to rain phenomena. To "learn away" the rain, we propose to map both the rainy and clean images of the same scene to an embedding space where they are close to each other by optimizing a similarity metric. Additionally, we minimize a reconstruction objective to ensure that the learned representation is sufficient to recover the underlying scene. Our approach is inspired by the recent advances in contrastive learning [6], and we aim to distill rain-robust representations of real-world scenes by directly comparing the rainy and clean images in the feature space. But unlike [6], we do not define a positive pair as augmentation to the same image, but rather any rainy image and its corresponding clean image from the same scene.

When training, we first randomly sample a mini-batch of N rainy images with the associated clean images to form an augmented batch $\{(\mathbf{I}_i, \mathbf{J}_i)\}_{i=1}^{N}$, where $\mathbf{I}_i$ is the i-th rainy image, and $\mathbf{J}_i$ is its corresponding ground-truth image. This augmented batch is fed into a shared-weight feature extractor $\mathcal{F}_E(\cdot, \theta_E)$ with weights θ_E to obtain a feature set $\{(\mathbf{z}_{\mathbf{I}_i}, \mathbf{z}_{\mathbf{J}_i})\}_{i=1}^{N}$, where $\mathbf{z}_{\mathbf{I}_i} = \mathcal{F}_E(\mathbf{I}_i, \theta_E)$ and $\mathbf{z}_{\mathbf{J}_i} = \mathcal{F}_E(\mathbf{J}_i, \theta_E)$. We consider every $(\mathbf{z}_{\mathbf{I}_i}, \mathbf{z}_{\mathbf{J}_i})$ as the positive pairs. This is so that the learned features from the same scene should be close to each other regardless of the rainy conditions. We treat the other $2(N-1)$ samples from the same batch as negative samples. Based on the noise-contrastive estimation (NCE) [16], we adopt the following InfoNCE [37] criterion to measure the rain-robust loss for a positive pair $(\mathbf{z}_{\mathbf{J}_i}, \mathbf{z}_{\mathbf{I}_i})$:

$$\ell_{\mathbf{z}_{\mathbf{J}_i}, \mathbf{z}_{\mathbf{I}_i}} = -\log \frac{\exp\left(\text{sim}_{\cos}(\mathbf{z}_{\mathbf{I}_i}, \mathbf{z}_{\mathbf{J}_i})/\tau\right)}{\sum_{\mathbf{k} \in \mathcal{K}} \exp\left(\text{sim}_{\cos}(\mathbf{z}_{\mathbf{J}_i}, \mathbf{k})/\tau\right)}, \tag{3}$$

where $\mathcal{K} = \{\mathbf{z}_{\mathbf{I}_j}, \mathbf{z}_{\mathbf{J}_j}\}_{j=1, j\neq i}^{N}$ is a set that contains the features extracted from other rainy and ground-truth images in the selected mini-batch, $\text{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathsf{T}}\mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$ is the cosine similarity between two feature vectors $\mathbf{u}$ and $\mathbf{v}$, and τ is the temperature parameter [48]. We set τ as 0.25, and this loss is calculated across all positive pairs within the mini-batch for both $(\mathbf{z}_{\mathbf{I}_i}, \mathbf{z}_{\mathbf{J}_i})$ and $(\mathbf{z}_{\mathbf{J}_i}, \mathbf{z}_{\mathbf{I}_i})$.

Full objective: While minimizing Eq. (3) maps features of clean and rainy images to the same subspace, we also need to ensure that the representation is sufficient to reconstruct the scene. Hence, we additionally minimize a Multi-Scale Structural Similarity Index (MS-SSIM) [46] loss and a $\ell 1$ image reconstruction loss to prevent the model from discarding useful information for the reconstruction task. Our full objective $\mathcal{L}_{\text{full}}$ is as follows:

$$\mathcal{L}_{\text{full}}(\hat{\mathbf{J}}, \mathbf{J}) = \mathcal{L}_{\text{MS-SSIM}}(\hat{\mathbf{J}}, \mathbf{J}) + \lambda_{\ell 1}\mathcal{L}_{\ell 1}(\hat{\mathbf{J}}, \mathbf{J}) + \lambda_{\text{robust}}\mathcal{L}_{\text{robust}}(\mathbf{z}_{\mathbf{J}}, \mathbf{z}_{\mathbf{I}}), \quad (4)$$

where $\mathcal{L}_{\text{MS-SSIM}}(\cdot)$ is the MS-SSIM loss that is commonly used for image restoration [59], $\mathcal{L}_{\ell 1}(\cdot)$ is the $\ell 1$ distance between the estimated clean images $\hat{\mathbf{J}}$ and the ground-truth images $\mathbf{J}$, $\mathcal{L}_{\text{robust}}(\cdot)$ is the rain-robust loss in Eq. (3), and $\lambda_{\ell 1}$ and λ_{robust} are two hyperparameters to control the relative importance of different loss terms. In our experiments, we set both $\lambda_{\ell 1}$ and λ_{robust} as 0.1.

Network architecture & implementation details: We design our model based on the architecture introduced in [24,60]. As illustrated in Fig. 4, our network includes an encoder of one input convolutional block, two downsampling blocks, and nine residual blocks [18] to yield latent features $\mathbf{z}$. This is followed by a decoder of two upsampling blocks and one output layer to map the features to $\mathbf{J}$. We fuse skip connections into the decoder using 3×3 up-convolution blocks to retain information lost in the bottleneck. Note: normal convolution layers are replaced by deformable convolution [62] in our residual blocks – in doing so, we enable our model to propagate non-local spatial information to reconstruct local degradations caused by rain effects. Latent features $\mathbf{z}$ are used for the rain-robust loss described in Eq. (3). Since these features are high dimensional ($256 \times 64 \times 64$), we use an average pooling layer to condense the feature map of each channel to 2×2. The condensed features are flattened into a vector of length 1024 for the rain-robust loss. It is worth noting that our rain-robust loss does not require additional modifications on the model architectures.

Our deraining model is trained on 256×256 patches and a mini-batch size $N = 8$ for 20 epochs. We use the Adam optimizer [25] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is 2×10^{-4}, and it is steadily modified to 1×10^{-6} based on a cosine annealing schedule [30]. We also use a linear warm-up policy for the first 4 epochs. For data augmentation, we use random cropping, random rotation, random horizontal and vertical flips, and RainMix augmentation [15]. More details can be found in the supplementary material.

Table 2. Quantitative comparison on GT-RAIN. Our method outperforms the existing state-of-the-art derainers. The preferred results are marked in **bold**.

Data Split	Metrics	Rainy Images	SPANet [44] (CVPR'19)	HRR [27] (CVPR'19)	MSPFN [22] (CVPR'20)	RCDNet [42] (CVPR'20)	DGNL-Net [20] (IEEE TIP'21)	EDR [15] (AAAI'21)	MPRNet [55] (CVPR'21)	Ours
Dense Rain	PSNR↑	18.46	18.87	17.86	19.58	19.50	17.33	18.86	19.12	**20.84**
Streaks	SSIM↑	0.6284	0.6314	0.5872	0.6342	0.6218	0.5947	0.6296	0.6375	**0.6573**
Dense Rain	PSNR↑	20.87	21.42	14.82	21.13	21.27	20.75	21.07	21.38	**24.78**
Accumulation	SSIM↑	0.7706	0.7696	0.4675	0.7735	0.7765	0.7429	0.7766	0.7808	**0.8279**
Overall	PSNR↑	19.49	19.96	16.55	20.24	20.26	18.80	19.81	20.09	**22.53**
	SSIM↑	0.6893	0.6906	0.5359	0.6939	0.6881	0.6582	0.6926	0.6989	**0.7304**

5   Experiments

We compare to state-of-the-art methods both quantitatively and qualitatively on GT-RAIN, and qualitatively Internet rainy images [47]. To quantify the difference between the derained results and ground-truth, we adopt peak signal-to-noise ratio (PSNR) [21] and structure similarity (SSIM) [45].

Quantitative evaluation on GT-RAIN: To quantify the sim2real gap of the existing datasets, we test seven representative existing state-of-the-art methods [15,20,22,27,42,44,55] on our GT-RAIN test set.[4] Since there exist numerous synthetic datasets proposed by previous works [12,19,27,29,50,56,57], we found it intractable to train our method on each one; whereas, it is more feasible to take the best derainers for each respective dataset and test on our proposed dataset as a proxy (Table 2). This follows the conventions of previous deraining dataset papers [11,20,29,44,51,56,57] to compare with top performing methods from each existing dataset.

SPANet [44] is trained on SPA-Data [44]. HRR [27] utilizes both NYU-Rain [27] and Outdoor-Rain [27]. MSPFN [22] and MPRNet [55] are trained on a combination of multiple synthetic datasets [12,29,50,57]. DGNL-Net [20] is trained on RainCityscapes [19]. For RCDNet [42] and EDR [15], multiple weights from different training sets are provided. We choose RCDNet trained on SPA-Data and EDR V4 trained on Rain14000 [12] due to superior performance.

Compared to training on GT-RAIN (ours), methods trained on other data perform worse, with the largest domain gap being in NYU-Rain and Outdoor-Rain (HRR) and RainCityscapes (DGNL). Two trends do hold: training on (1) more synthetic data gives better results (MSPFN, MPRNet) and (2) semi-real data also helps (SPANet). However, even when multiple synthetic [12,29,50,57] or semi-real [44] datasets are used, their performance on real data is still around 2dB lower than training on GT-RAIN (ours).

Fig. 5 illustrates some representative derained images across scenarios with various rain appearance and rain accumulation densities. Training on GT-RAIN enables the network to remove most rain streaks and rain accumulation; whereas, training on synthetic/semi-real data tends to leave visible rain streaks. We note

---

[4] We use the original code and network weights from the authors for comparison. Code links for all comparison methods are provided in the supplementary material.

that HRR [27] and DGNL [20] may seem like they remove rain accumulation, but they in fact introduce undesirable artifacts, e.g. dark spots on the back of the traffic sign, tree, and sky. The strength of having ground-truth paired data is demonstrated by our 2.44 dB gain compared to the state of the art [55]. On test images with dense rain accumulation, the boost improves to 3.40 dB.

Rain	SPANet [44]	HRR [27]	MSPFN [22]	RCDNet [42]
(23.64/0.8561)	(23.56/0.8474)	(19.78/0.7508)	(25.57/0.8659)	(24.71/0.8654)
DGNL [20]	EDR V4 [15]	MPRNet [55]	**Ours**	Ground Truth
(17.26/0.7516)	(23.93/0.8539)	(24.33/0.8657)	**(26.31/0.8763)**	(PSNR/SSIM)
Rain	SPANet [44]	HRR [27]	MSPFN [22]	RCDNet [42]
(19.81/0.7541)	(20.03/0.7244)	(15.03/0.4944)	(19.64/0.7491)	(20.58/0.7164)
DGNL [20]	EDR V4 [15]	MPRNet [55]	**Ours**	Ground Truth
(15.51/0.6508)	(19.96/0.7461)	(19.88/0.7551)	**(23.89/0.7906)**	(PSNR/SSIM)

Fig. 5. Our model simultaneously removes rain streaks and rain accumulation, while the existing models fail to generalize to real-world data. The red arrows highlight the difference between the proposed and existing methods on the GT-RAIN test set (zoom for details, PSNR and SSIM scores are listed below the images).

Qualitative evaluation on other real images: Other than the models described in the above section, we also include EDR V4 [15] trained on SPA-

| Rainy Image | SPANet [44] | HRR [27] | MSPFN [22] | RCDNet [42] |

| DGNL-Net [20] | EDR V4 (S) [15] | EDR V4 (R) [15] | MPRNet [55] | Ours |

| Rainy Image | SPANet [44] | HRR [27] | MSPFN [22] | RCDNet [42] |

| DGNL-Net [20] | EDR V4 (S) [15] | EDR V4 (R) [15] | MPRNet [55] | Ours |

Fig. 6. Our model can generalize across real rainy images with robust performance. We select representative real rainy images with various rain patterns and backgrounds for comparison (zoom for details). EDR V4 (S) [15] denotes EDR trained on SPA-Data [44], and EDR V4 (R) [15] denotes EDR trained on Rain14000 [12].

Data [44] for the qualitative comparison, since it shows more robust rain streak removal results as compared the version trained on Rain14000 [12]. The derained results on Internet rainy images are illustrated in Fig. 6. The model trained on the proposed GT-RAIN (i.e. ours) deals with large rain streaks of various shapes and sizes as well as the associated rain accumulation effects, while preserving the features present in the scene. In contrast, we observe that models [20,27] trained on data with synthetic rain accumulation introduce unwanted color shifts and residual rain streaks in their results. Moreover, the state-of-the-art methods [22,42,55] are unable to remove the majority of rain streaks in general as highlighted in the red zoom boxes. This demonstrates the gap between top methods on synthetic versus one that can be applied to real data.

Retraining other methods on GT-RAIN: We additionally train several state-of-the-art derainers [15,42,55] on the GT-RAIN training set to demonstrate that our real dataset leads to more robust real-world deraining and benefits all models. We have selected the most recent derainers for this retraining

Table 3. Retraining comparison methods on GT-RAIN. The improvement of these derainers further demonstrates the effectiveness of real paired data.

Data Split	Metrics	Rainy Images	RCDNet [42] (Original)	RCDNet [42] (GT-RAIN)	EDR [15] (Original)	EDR [15] (GT-RAIN)	MPRNet [55] (Original)	MPRNet [55] (GT-RAIN)	Ours
Dense Rain Streaks	PSNR↑	18.46	19.50	19.60	18.86	19.95	19.12	20.19	**20.84**
	SSIM↑	0.6284	0.6218	0.6492	0.6296	0.6436	0.6375	0.6542	**0.6573**
Dense Rain Accumulation	PSNR↑	20.87	21.27	22.74	21.07	23.42	21.38	23.38	**24.78**
	SSIM↑	0.7706	0.7765	0.7891	0.7766	0.7994	0.7808	0.8009	**0.8279**
Overall	PSNR↑	19.49	20.26	20.94	19.81	21.44	20.09	21.56	**22.53**
	SSIM↑	0.6893	0.6881	0.7091	0.6926	0.7104	0.6989	0.7171	**0.7304**

Table 4. Fine-tuning comparison methods on GT-RAIN. (F) denotes the fine-tuned models, and (O) denotes the original models trained on synthetic/real data.

Data Split	Metrics	Rainy Images	RCDNet [42] (O)	RCDNet [42] (F)	EDR [15] (O)	EDR [15] (F)	MPRNet [55] (O)	MPRNet [55] (F)	Ours (O)	Ours (F)
Dense Rain Streaks	PSNR↑	18.46	19.50	19.33	18.86	20.03	19.12	20.65	**20.84**	20.79
	SSIM↑	0.6284	0.6218	0.6463	0.6296	0.6433	0.6375	0.6561	0.6573	**0.6655**
Dense Rain Accumulation	PSNR↑	20.87	21.27	22.50	21.07	23.57	21.38	24.37	24.78	**25.20**
	SSIM↑	0.7706	0.7765	0.7893	0.7766	0.8016	0.7808	0.8250	0.8279	**0.8318**
Overall	PSNR↑	19.49	20.26	20.69	19.81	21.55	20.09	22.24	22.53	**22.68**
	SSIM↑	0.6893	0.6881	0.7076	0.6926	0.7111	0.6989	0.7285	0.7304	**0.7368**

study.[5] All the models are trained from scratch, and the corresponding PSNR and SSIM scores on the GT-RAIN test set are provided in Table 3. For all the retrained models, we can observe a PSNR and SSIM gain by using the proposed GT-RAIN dataset. In addition, with all models trained on the same dataset, our model still outperforms others in all categories.

Fine-tuning other methods on GT-RAIN: To demonstrate of the effectiveness of combining real and synthetic datasets, we also fine-tune several more recent derainers [15,42,55] that are previously trained on synthetic datasets with the proposed GT-RAIN dataset. We fine-tune from the official weights as described in the above quantitative evaluation section, and the fine-tuning learning rate is 20% of the original learning rate for each method. For the proposed method, we pretrain the model on the synthetic dataset used by MSPFN [22] and MPRNet [55]. The corresponding PSNR and SSIM scores on the GT-RAIN test set are listed in Table 4. In the table, we can observe a further boost as compared with training the models from scratch with just real or synthetic data.

Table 5. Ablation study. Our rain-robust loss improves both PSNR and SSIM.

Metrics	Rainy Images	Ours w/o $\mathcal{L}_{\text{robust}}$	Ours w/ $\mathcal{L}_{\text{robust}}$
PSNR↑	19.49	21.82	**22.53**
SSIM↑	0.6893	0.7148	**0.7304**

---

[5] Both DGNL-Net [20] and HRR [27] cannot be retrained on our real dataset, as both require additional supervision, such as transmission maps and depth maps.
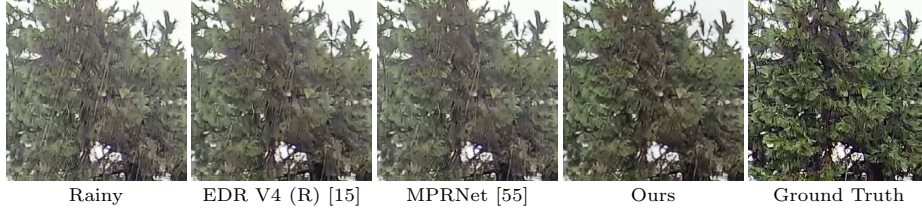
| Rainy | EDR V4 (R) [15] | MPRNet [55] | Ours | Ground Truth |

Fig. 7. Deraining is still an open problem. Both the proposed method and the existing work have difficulty in generalizing the performance to some challenging scenes.

Ablation study: We validate the effectiveness of the rain-robust loss with two variants of the proposed method: (1) the proposed network with the full objective as describe in Sec. 4; and (2) the proposed network with just MS-SSIM loss and ℓ_1 loss. The rest of the training configurations and hyperparameters remain identical. The quantitative metrics for these two variants on the proposed GT-RAIN test set are listed in Table 5. Our model trained with the proposed rain-robust loss produces a normalized correlation between rainy and clean latent vectors of $.95 \pm .03$; whereas it is $.85 \pm .10$ for the one without. These rain-robust features help the model to show improved performance in both PSNR and SSIM.

Failure cases: Apart from the successful cases illustrated in Fig. 5, we also provide some of the failure cases in the GT-RAIN test set in Fig. 7. Deraining is still an open problem, and we hope future work can take advantages of both real and synthetic samples to make derainers more robust in diverse environments.

6   Conclusions

Many of us in the deraining community probably wish for the existence of parallel universes, where we could capture the exact same scene with and without weather effects at the exact same time. Unfortunately, however, we are stuck with our singular universe, in which we are left with two choices: (1) synthetic data at the same timestamp with simulated weather effects or (2) real data at different timestamps with real weather effects. Though it is up to opinion, it is our belief that the results of our method in Fig. 6 reduce the visual domain gap more than those trained with synthetic datasets. Additionally, we hope the introduction of a real dataset opens up exciting new pathways for future work, such as the blending of synthetic and real data or setting goalposts to guide the continued development of existing rain simulators [17,36,43,53,54].

References

1. Ancuti, C.O., Ancuti, C., Sbert, M., Timofte, R.: Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In: 2019 IEEE international conference on image processing (ICIP). pp. 1014–1018. IEEE (2019)
2. Ancuti, C.O., Ancuti, C., Timofte, R.: Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 444–445 (2020)
3. Ancuti, C.O., Ancuti, C., Timofte, R., De Vleeschouwer, C.: O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 754–762 (2018)
4. Ancuti, C., Ancuti, C.O., Timofte, R., Vleeschouwer, C.D.: I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In: International Conference on Advanced Concepts for Intelligent Vision Systems. pp. 620–631. Springer (2018)
5. Beard, K.V., Chuang, C.: A new model for the equilibrium shape of raindrops. Journal of Atmospheric Sciences **44**(11), 1509–1524 (1987)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
7. Chen, Y.L., Hsu, C.T.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1968–1975 (2013)
8. Deng, L.J., Huang, T.Z., Zhao, X.L., Jiang, T.X.: A directional global sparse model for single image rain removal. Applied Mathematical Modelling **59**, 662–679 (2018)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
10. Foote, G.B., Du Toit, P.S.: Terminal velocity of raindrops aloft. Journal of Applied Meteorology **8**(2), 249–253 (1969)
11. Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: A deep network architecture for single-image rain removal. IEEE Transactions on Image Processing **26**(6), 2944–2956 (2017)
12. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3855–3863 (2017)
13. Garg, K., Nayar, S.K.: Vision and rain. International Journal of Computer Vision **75**(1), 3–27 (2007)
14. Gunn, R., Kinzer, G.D.: The terminal velocity of fall for water droplets in stagnant air. Journal of Atmospheric Sciences **6**(4), 243–248 (1949)
15. Guo, Q., Sun, J., Juefei-Xu, F., Ma, L., Xie, X., Feng, W., Liu, Y., Zhao, J.: Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1487–1495 (2021)
16. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 297–304. JMLR Workshop and Conference Proceedings (2010)

17. Halder, S.S., Lalonde, J.F., Charette, R.d.: Physics-based rendering for improving robustness to rain. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10203–10212 (2019)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
19. Hu, X., Fu, C.W., Zhu, L., Heng, P.A.: Depth-attentional features for single-image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8022–8031 (2019)
20. Hu, X., Zhu, L., Wang, T., Fu, C.W., Heng, P.A.: Single-image real-time rain removal based on depth-guided non-local features. IEEE Transactions on Image Processing **30**, 1759–1770 (2021)
21. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters **44**(13), 800–801 (2008)
22. Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8346–8355 (2020)
23. Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., Wang, Y.: Fastderain: A novel video rain streak removal method using directional gradient priors. IEEE Transactions on Image Processing **28**(4), 2089–2102 (2018)
24. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European Conference on Computer Vision. pp. 694–711. Springer (2016)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Li, G., He, X., Zhang, W., Chang, H., Dong, L., Lin, L.: Non-locally enhanced encoder-decoder network for single image de-raining. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1056–1064 (2018)
27. Li, R., Cheong, L.F., Tan, R.T.: Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1633–1642 (2019)
28. Li, R., Tan, R.T., Cheong, L.F.: All in one bad weather removal using architectural search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3175–3185 (2020)
29. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2736–2744 (2016)
30. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), `https://openreview.net/forum?id=Skq89Scxx`
31. Lowe, D.: Sift-the scale invariant feature transform. Int. J **2**(91-110), 2 (2004)
32. Ltd., O.: OpenWeatherMap API. `https://openweathermap.org/`, accessed: 2021-11-05
33. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3397–3405 (2015)
34. Manning, R.M.: Stochastic Electromagnetic Image Propagation. McGraw-Hill Companies (1993)

35. Marshall, J., Palmer, W.M.: The distribution of raindrops with size. Journal of Meteorology **5**(4), 165–166 (1948)

36. Ni, S., Cao, X., Yue, T., Hu, X.: Controlling the rain: From removal to rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6328–6337 (2021)

37. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

38. Pan, J., Liu, S., Sun, D., Zhang, J., Liu, Y., Ren, J., Li, Z., Tang, J., Lu, H., Tai, Y.W., et al.: Learning dual convolutional neural networks for low-level vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3070–3079 (2018)

39. Ren, D., Shang, W., Zhu, P., Hu, Q., Meng, D., Zuo, W.: Single image deraining using bilateral recurrent network. IEEE Transactions on Image Processing **29**, 6852–6863 (2020)

40. Thirion, J.P.: Image matching as a diffusion process: an analogy with maxwell's demons. Medical image analysis **2**(3), 243–260 (1998)

41. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Diffeomorphic demons: Efficient non-parametric image registration. NeuroImage **45**(1), S61–S72 (2009)

42. Wang, H., Xie, Q., Zhao, Q., Meng, D.: A model-driven deep neural network for single image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2020)

43. Wang, H., Yue, Z., Xie, Q., Zhao, Q., Zheng, Y., Meng, D.: From rain generation to rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14791–14801 (2021)

44. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12270–12279 (2019)

45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

46. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 2, pp. 1398–1402. Ieee (2003)

47. Wei, W., Meng, D., Zhao, Q., Xu, Z., Wu, Y.: Semi-supervised transfer learning for image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3877–3886 (2019)

48. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3733–3742 (2018)

49. Yang, W., Tan, R.T., Feng, J., Guo, Z., Yan, S., Liu, J.: Joint rain detection and removal from a single image with contextualized deep networks. IEEE transactions on pattern analysis and machine intelligence **42**(6), 1377–1393 (2019)

50. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1357–1366 (2017)

51. Yang, W., Tan, R.T., Wang, S., Fang, Y., Liu, J.: Single image deraining: From model-based to data-driven and beyond. IEEE Transactions on pattern analysis and machine intelligence (2020)

52. Yasarla, R., Patel, V.M.: Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8405–8414 (2019)
53. Ye, Y., Chang, Y., Zhou, H., Yan, L.: Closing the loop: Joint rain generation and removal via disentangled image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2053–2062 (2021)
54. Yue, Z., Xie, J., Zhao, Q., Meng, D.: Semi-supervised video deraining with dynamical rain generator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 642–652 (2021)
55. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14821–14831 (2021)
56. Zhang, H., Patel, V.M.: Density-aware single image de-raining using a multi-stream dense network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 695–704 (2018)
57. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE transactions on circuits and systems for video technology **30**(11), 3943–3956 (2019)
58. Zhang, X., Li, H., Qi, Y., Leow, W.K., Ng, T.K.: Rain removal in video by combining temporal and chromatic properties. In: 2006 IEEE international conference on multimedia and expo. pp. 461–464. IEEE (2006)
59. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging **3**(1), 47–57 (2016)
60. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232 (2017)
61. Zhu, L., Fu, C.W., Lischinski, D., Heng, P.A.: Joint bi-layer optimization for single-image rain streak removal. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2526–2534 (2017)
62. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9308–9316 (2019)

Not Just Streaks: Towards Ground Truth for Single Image Deraining
(Supplementary Material)

Yunhao Ba[1★], Howard Zhang[1★], Ethan Yang[1], Akira Suzuki[1], Arnold Pfahnl[1],
Chethan Chinder Chandrappa[1], Celso M. de Melo[2], Suya You[2], Stefano
Soatto[1], Alex Wong[3], and Achuta Kadambi[1]

[1] University of California, Los Angeles
{yhba,hwdz15508,eyang657,asuzuki100,ajpfahnl,chinderc}@ucla.edu
soatto@cs.ucla.edu, achuta@ee.ucla.edu
[2] DEVCOM Army Research Laboratory
{celso.m.demelo.civ,suya.you.civ}@army.mil
[3] Yale University
alex.wong@yale.edu

A   Visualization of Previous Deraining Datasets

We illustrate some typical image pairs from various deraining datasets in Fig. A.
Synthetic datasets in the community are usually generated by adding synthetic
rain effects on real images taken under sunny illumination conditions, and the
semi-real SPA-Data [34] only considers rain streaks. As a result, the domain gap
between these existing datasets and real rainy scenarios are relatively larger as
compared with the proposed GT-RAIN dataset.

B   More Results from GT-RAIN

As an additional supplement to Fig. 5 in the main paper, we provide some
more quantitative and qualitative results from our test set in Fig. B. Note that
these comparison models are using the weights provided by the authors which
are trained on synthetic or semi-real datasets. We see that our proposed model
trained on GT-RAIN continues to outperform other competing models.

C   More Results on Internet Images

As a supplement to Fig. 6 in the main paper, we provide more qualitative results
on real Internet images in Fig. C. Note that all comparison models are using the
weights provided by the author, which are trained on synthetic or semi-real
datasets. All images are taken from the dataset of common real rainy images
provided by [36]. Our proposed model trained on GT-RAIN continues to remove
rain streaks of varying shapes and sizes as well as rain accumulation without
introducing the unwanted color shifts seen in HRR [20] and DGNL-Net [10].

★ Equal contribution.

Fig. A. GT-RAIN contains realistic rain effects (both rain streaks and rain accumulation), while the existing synthetic and semi-real datasets fail to cover the physical complexity and diversity of real-world rain. The synthetic image pair is from the commonly used Rain14000 dataset [6], and the pseudo ground-truth image of SPA-Data [34] in the figure is generated by running the official code from the authors on our collected rainy video.

Fig. B. More results on GT-RAIN test set. Similarly, the proposed method is capable of removing various rain streaks and rain accumulation effects.

D   Qualitative Results of Retrained Methods

As an additional supplement to Tab. 3 in the main paper, we provide some representative samples of the retrained models for some qualitative comparison in Fig. D. The visual improvements of these derainers in rain fog and streak removal further validate the effectiveness of the proposed dataset.

Fig. C. More qualitative results on Internet images. Our model continues to exhibit robust generalization to real rainy images, whereas existing derainers usually fail on removing rain streaks of diverse shapes and sizes. EDR V4 (S) [8] denotes the EDR model trained on SPA-Data [34], and EDR V4 (R) [8] denotes the EDR model trained on Rain14000 [6].

E   Comparison with Semi-supervised Methods

In addition to the models trained on synthetic and semi-real datasets, we also compare the proposed method with some recent semi-supervised methods, including SIRR [36] and MOSS [11], that are trained on real images as a complement to Tab. 2 in the main paper. The corresponding PSNR/SSIM scores on the entire GT-RAIN test set for these two semi-supervised methods are listed
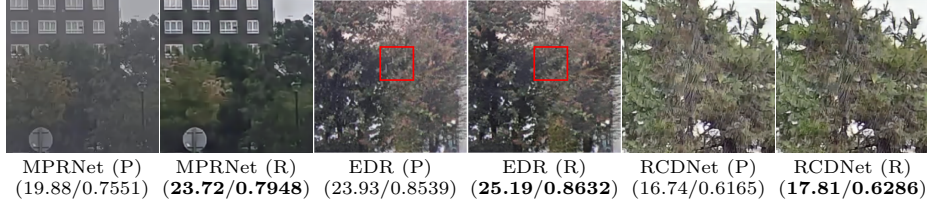
MPRNet (P)   MPRNet (R)   EDR (P)   EDR (R)   RCDNet (P)   RCDNet (R)
(19.88/0.7551) (**23.72/0.7948**) (23.93/0.8539) (**25.19/0.8632**) (16.74/0.6165) (**17.81/0.6286**)

Fig. D. Qualitative results of retrained SOTAs. (P) denote pretrained models provided by the original authors, and (R) denotes the retrained models on the proposed GT-RAIN dataset. The improvements further highlight the effectiveness of the proposed GT-RAIN dataset.

as follows: SIRR [36] (20.57/0.6448), and MOSS [11] (21.42/0.7073), where ours are (22.53/0.7304). Some qualitative results can be found in Fig. E.

Rainy   SSIR [36]   MOSS [11]   Ours

Fig. E. Qualitative comparison with semi-supervised SOTAs. As compared with semi-supervised models, the proposed method can remove the rain streaks more effectively.

F   Alignment of Small Motions

As a complement to Sec. 3 of the main paper, we first show, in Fig. F-(a), a ground-truth image overlaid on top of a rainy image to demonstrate representative samples that passed our data collection appearance criteria and also motion criterion, where we do not need to perform motion correction. We note that this is the case for the majority of our dataset. Additionally, we show an overlaid image pair that passed our appearance criteria, but failed the motion criterion. Fig. F-(b) shows the image pair before and after the motion correction. It should be noted that only a small portion of the data requires such correction, and our correction pipeline is designed to be robust to rain artifacts. It is because even though rain can influence local descriptors, the combinatorial matching stage is designed to be robust to a preponderance of outliers. For most

cases, the percentage of outliers affects the time it takes to converge, but not the quality. All samples that require our correction procedure were manually inspected after the alignment – any failure cases of the procedure, typically due to extreme weather conditions, were manually removed.

(a)   No correction needed   No correction needed

(b)   Before correction   After correction

Fig. F. The proposed method can correct for small motions under rain. We illustrate two types of scenes by overlaying the rainy images on top of their clean ground truths: (a) two scenes that do not need additional image processing for motion alignment; and (b) a scene with motion before and after running the correction algorithms. It should be noted that both types of scenes are aligned properly in GT-RAIN.

G   Runtime Comparison

We list the total number of parameters with the associated runtime for other state-of-the-art methods and our proposed model in Table A. The comparison is conducted on a single NVIDIA P100 GPU, and each derainer is asked to restore a colored rainy image of size 256×256. We note that the top three methods (DGNL-Net [10], EDR [8], and our proposed method) all operate at real-time deraining speeds. However, our method outperforms them by 3.73 dB and 2.72 dB PSNR respectively.

H   Limitations

Although we achieve the state of the art for deraining real images, our method is not perfect. Our PSNR and SSIM scores on GT-RAIN are 22.53 dB and 0.7304. This suggests that indeed, we still have ample room for improvement. For example, we leave a slight rain accumulation in the tree in Fig. B. While the recovered image is sharper and contains less rain artifacts than competing methods,

Table A. Runtime comparison. The average inference time is calculated on 256 × 256 color images.

Model	SPANet [34] (CVPR'19)	HRR [20] (CVPR'19)	MSPFN [13] (CVPR'20)	RCDNet [33] (CVPR'20)	DGNL-Net [10] (IEEE TIP'21)	EDR [8] (AAAI'21)	MPRNet [46] (CVPR'21)	Ours
Number of Parameters	284k	40.6M	15.8M	3.16M	4.03M	27.3M	3.63M	12.9M
Inference Time (ms)	86.65	35.35	145.5	189.6	4.230	4.617	36.91	12.79

boundaries in highly textured areas (e.g. leaves, bricks, and foliage) are blurred. In Fig. C, we observe a similar trend. However, this is a challenge that plagues all methods. We hope that further extensions of our approach and GT-RAIN will help mitigate these artifacts. We also do not consider occlusions from raindrops on the camera lens because the raindrops will likewise be present on the lens after the rain stops. Moreover, we do not consider specular reflections from water surfaces. This is because these reflections are nearly impossible to reconstruct as the water ripples in the puddles will destroy the visual patterns during raining. We hope that future works can address these limitations. While we have describe image restoration as the main task of deraining, we conjecture that our results may also be applicable towards the re-use of pretrained models on clean data for downstream tasks like: depth completion [9,21,24,25,38,39,40,43,45], stereo [2,3,4,41,44], optical flow [1,17,18,19,31,32], object detection [14,15,22,30], and monocular depth prediction [5,7,27,28,29,35,37,42].

I   Comparison Code Links

The code links for all the comparison methods in the main paper are listed in Table B.

Table B. Code links for the comparison methods.

Methods	Links
SPANet [34] (CVPR'19)	`https://github.com/stevewongv/SPANet`
HRR [20] (CVPR'19)	`https://github.com/liruoteng/HeavyRainRemoval`
MSPFN [13] (CVPR'20)	`https://github.com/kuijiang0802/MSPFN`
RCDNet [33] (CVPR'20)	`https://github.com/hongwang01/RCDNet`
DGNL-Net [10] (IEEE TIP'21)	`https://github.com/xw-hu/DGNL-Net`
Efficient Derain [8] (AAAI'21)	`https://github.com/tsingqguo/efficientderain`
MPRNet [46] (CVPR'21)	`https://github.com/swz30/MPRNet`

J   Network Architecture & Implementation

As an additional supplement of the network architecture & implementation section in the main paper, we provide more implementation details here. In our model, the input convolutional block contains two convolutional layers with kernel sizes of 7×7 and 3×3 respectively. The downsampling blocks are instantiated by 3×3 convolutional layers with a stride of 2, and each upsampling block consists of a bilinear interpolation layer and a 3×3 convolutional layer. Please refer to Table C for a more detailed illustration of the network architecture. We use batch normalization [12] and choose leaky ReLUs [23] with a negative slope of 0.1 as the activation function. Our model is implemented in PyTorch [26]. The MS-SSIM loss is implemented based on the PyTorch Image Quality (PIQ) library [16]. Experiments are conducted on an NVIDIA Tesla P100 GPU.

Table C. Illustration of our network architecture.

Network	Kernel		Channels		Resolution		Parameters	Input
	Size	Stride	In	Out	In	Out		
Encoder								
InputConv1	7	1	3	64	1	1	$\approx$ 9.5k	Rainy Image
InputConv2	3	1	64	64	1	1	$\approx$ 37.0k	InputConv1
DownConv1	3	2	64	128	1	1/2	$\approx$ 74.0k	InputConv2
DownConv2	3	2	128	256	1/2	1/4	$\approx$ 295.4k	DownConv1
DeformResBlock1								
DeformConv11	3	1	256	256	1/4	1/4	$\approx$ 652.6k	DownConv2
DeformConv12	3	1	256	256	1/4	1/4	$\approx$ 652.6k	DeformConv11
Sum1	-	-	256	256	1/4	1/4	DownConv2 + DeformConv12	
DeformResBlock2								
DeformConv21	3	1	256	256	1/4	1/4	$\approx$ 652.6k	Sum1
DeformConv22	3	1	256	256	1/4	1/4	$\approx$ 652.6k	DeformConv21
Sum2	-	-	256	256	1/4	1/4	Sum1 + DeformConv21	
$\vdots$								
DeformResBlock9								
DeformConv91	3	1	256	256	1/4	1/4	$\approx$ 652.6k	Sum8
DeformConv92	3	1	256	256	1/4	1/4	$\approx$ 652.6k	DeformConv91
Sum9	-	-	256	256	1/4	1/4	Sum8 + DeformConv92	
Decoder								
UpConvBlock1								
Bilinear1	-	-	256	256	1/4	1/2	-	Sum9
Conv11	3	1	256	128	1/2	1/2	$\approx$ 295.2k	Bilinear2
Concat1	-	-	128 + 128	256	1/2	1/2	DownConv1, Conv11	
Conv12	3	1	256	128	1/2	1/2	$\approx$ 295.2k	Concat1
UpConvBlock2								
Bilinear2	-	-	128	128	1/2	1	-	Conv12
Conv21	3	1	128	64	1	1	$\approx$ 73.9k	Bilinear2
Concat2	-	-	64 + 64	128	1	1	InputConv2, Conv21	
Conv22	3	1	128	64	1	1	$\approx$ 73.9k	Concat2
OutputConv	3	1	64	3	1	1	$\approx$ 1.7k	Conv22
Total Parameters $\approx$ 12.9M								

References

1. Aleotti, F., Poggi, M., Tosi, F., Mattoccia, S.: Learning end-to-end scene flow by distilling single tasks knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10435–10442 (2020)
2. Berger, Z., Agrawal, P., Liu, T.Y., Soatto, S., Wong, A.: Stereoscopic universal perturbations across different architectures and datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15180–15190 (2022)
3. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5410–5418 (2018)
4. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4384–4393 (2019)
5. Fei, X., Wong, A., Soatto, S.: Geo-supervised visual depth prediction. IEEE Robotics and Automation Letters **4**(2), 1661–1668 (2019)
6. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3855–3863 (2017)
7. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
8. Guo, Q., Sun, J., Juefei-Xu, F., Ma, L., Xie, X., Feng, W., Liu, Y., Zhao, J.: Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1487–1495 (2021)
9. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. arXiv preprint arXiv:2103.00783 (2021)
10. Hu, X., Zhu, L., Wang, T., Fu, C.W., Heng, P.A.: Single-image real-time rain removal based on depth-guided non-local features. IEEE Transactions on Image Processing **30**, 1759–1770 (2021)
11. Huang, H., Yu, A., He, R.: Memory oriented transfer learning for semi-supervised image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7732–7741 (2021)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
13. Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8346–8355 (2020)
14. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, Fang, J., imyhxy, Michael, K., Lorna, V, A., Montes, D., Nadar, J., Laughing, tkianai, yxNONG, Skalski, P., Wang, Z., Hogan, A., Fati, C., Mammana, L., AlexWang1900, Patel, D., Yiwei, D., You, F., Hajek, J., Diaconu, L., Minh, M.T.: ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference (Feb 2022). https://doi.org/10.5281/zenodo.6222936, https://doi.org/10.5281/zenodo.6222936

15. Kalra, A., Stoppi, G., Brown, B., Agarwal, R., Kadambi, A.: Towards rotation invariance in object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3530–3540 (2021)
16. Kastryulin, S., Zakirov, D., Prokopenko, D.: PyTorch Image Quality: Metrics and measure for image quality assessment (2019), `https://github.com/photosynthesis-team/piq`, open-source software available at https://github.com/photosynthesis-team/piq
17. Lao, D., Sundaramoorthi, G.: Minimum delay moving object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4250–4259 (2017)
18. Lao, D., Sundaramoorthi, G.: Extending layered models to 3d motion. In: Proceedings of the European conference on computer vision (ECCV). pp. 435–451 (2018)
19. Lao, D., Sundaramoorthi, G.: Minimum delay object detection from video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5097–5106 (2019)
20. Li, R., Cheong, L.F., Tan, R.T.: Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1633–1642 (2019)
21. Liu, T.Y., Agrawal, P., Chen, A., Hong, B.W., Wong, A.: Monitored distillation for positive congruent depth completion. arXiv preprint arXiv:2203.16034 (2022)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
23. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013)
24. Merrill, N., Geneva, P., Huang, G.: Robust monocular visual-inertial depth completion for embedded systems. In: International Conference on Robotics and Automation (ICRA). IEEE (2021)
25. Park, J., Joo, K., Hu, Z., Liu, C.K., Kweon, I.S.: Non-local spatial propagation network for depth completion. In: European Conference on Computer Vision, ECCV 2020. European Conference on Computer Vision (2020)
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`
27. Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S.: On the uncertainty of self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3227–3237 (2020)
28. Poggi, M., Tosi, F., Aleotti, F., Mattoccia, S.: Real-time self-supervised monocular depth estimation without gpu. IEEE Transactions on Intelligent Transportation Systems (2022)
29. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12179–12188 (2021)

30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
31. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
32. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
33. Wang, H., Xie, Q., Zhao, Q., Meng, D.: A model-driven deep neural network for single image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2020)
34. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12270–12279 (2019)
35. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2162–2171 (2019)
36. Wei, W., Meng, D., Zhao, Q., Xu, Z., Wu, Y.: Semi-supervised transfer learning for image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3877–3886 (2019)
37. Wong, A., Cicek, S., Soatto, S.: Targeted adversarial perturbations for monocular depth prediction. Advances in Neural Information Processing Systems **33** (2020)
38. Wong, A., Cicek, S., Soatto, S.: Learning topology from synthetic data for unsupervised depth completion. IEEE Robotics and Automation Letters **6**(2), 1495–1502 (2021)
39. Wong, A., Fei, X., Hong, B.W., Soatto, S.: An adaptive framework for learning unsupervised depth completion. IEEE Robotics and Automation Letters **6**(2), 3120–3127 (2021)
40. Wong, A., Fei, X., Tsuei, S., Soatto, S.: Unsupervised depth completion from visual inertial odometry. IEEE Robotics and Automation Letters (2020)
41. Wong, A., Mundhra, M., Soatto, S.: Stereopagnosia: Fooling stereo networks with adversarial perturbations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2879–2888 (2021)
42. Wong, A., Soatto, S.: Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5644–5653 (2019)
43. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12747–12756 (2021)
44. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1959–1968 (2020)
45. Yang, Y., Wong, A., Soatto, S.: Dense depth posterior (ddp) from single image and sparse range. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3353–3362 (2019)
46. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14821–14831 (2021)