Dreaming to Prune Image Deraining Networks

Weiqi Zou[1,*] Yang Wang[1,*] Xueyang Fu[1] Yang Cao[1,2,†]
[1] University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
artisan@mail.ustc.edu.cn, {ywang120, xyfu, forrest}@ustc.edu.cn

Abstract

Convolutional image deraining networks have achieved great success while suffering from tremendous computational and memory costs. Most model compression methods require original data for iterative fine-tuning, which is limited in real-world applications due to storage, privacy, and transmission constraints. We note that it is overstretched to fine-tune the compressed model using self-collected data, as it exhibits poor generalization over images with different degradation characteristics. To address this problem, we propose a novel data-free compression framework for deraining networks. It is based on our observation that deep degradation representations can be clustered by degradation characteristics (types of rain) while independent of image content. Therefore, in our framework, we "dream" diverse in-distribution degraded images using a deep inversion paradigm, thus leveraging them to distill the pruned model. Specifically, we preserve the performance of the pruned model in a dual-branch way. In one branch, we invert the pre-trained model (teacher) to reconstruct the degraded inputs that resemble the original distribution and employ the orthogonal regularization for deep features to yield degradation diversity. In the other branch, the pruned model (student) is distilled to fit the teacher's original statistical modeling on these dreamed inputs. Further, an adaptive pruning scheme is proposed to determine the hierarchical sparsity, which alleviates the regression drift of the initial pruned model. Experiments on various deraining datasets demonstrate that our method can reduce about 40% *FLOPs of the state-of-the-art models while maintaining comparable performance without original data.*

1. Introduction

Convolutional Neural Networks (CNNs) based approaches have achieved remarkable progress on single image deraining [5, 10, 16, 28, 31]. However, due to the inherent properties of dense prediction tasks, coupled with the requirements to handle various degradation characteristics, these CNN models typically suffer from tremendous computational costs and bulky memory usage. This limits their applications in real scenarios, especially on devices with constrained computing capacity.

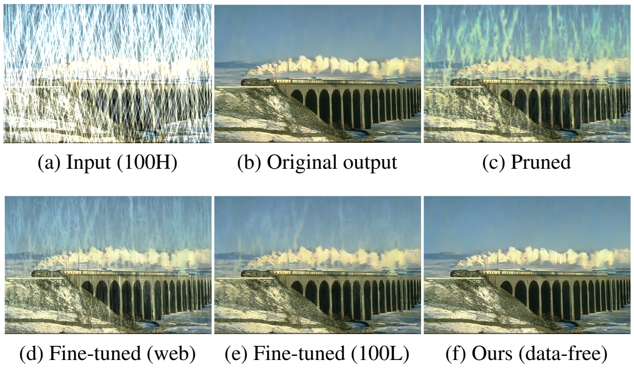
(a) Input (100H) (b) Original output (c) Pruned
(d) Fine-tuned (web) (e) Fine-tuned (100L) (f) Ours (data-free)

Figure 1. Pre-trained HINet [5] drops significantly after pruning 30% weights (1b → 1c). Fine-tuning the pruned model with images from website or Rain100L [29] exhibits poor generalization to images in Rain100H [29]. We preserve the performance of the pruned model without the original data.

In practice, various attempts have been made to compress the heavy CNN models, including quantization [11, 12, 23], pruning [8, 13, 15, 21], distillation [3, 7, 14], and so on. These approaches require original data for interactive training to preserve the performance of compressed models. However, the original training data is often unaccessible due to storage, privacy, or transmission constraints. To alleviate this problem, one may naturally acquire paired data by collecting degraded (rainy) images and exporting their pseudo labels output by the pre-trained model. However, our study suggests this way exhibits poor generalization. For example, state-of-the-art deraining network HINet [5] gets a significant performance drop after pruning 30% weights, as shown in Fig. 1c. Fine-tuning the pruned model with data collected from the website or Rain100L [29] exhibits poor generalization to images in Rain100H [29], as shown in

[*] Equal contribution. [†] Corresponding author.

Fig. 1d and Fig. 1e. Those heavy deraining models typically achieve promising performance over images with various degradation characteristics, such as rain steaks or drops in different orientations and densities. It is impractical to acquire all types of in-distribution data on which the model has ever been trained.

Recently, some feasible methods have been proposed to perform data-free model compression. They mainly attempt to reconstruct the original data for data-free knowledge distillation, either through a model inversion paradigm [9, 19, 22, 30] similar to DeepDream [1], or by retraining a generative network [4]. However, these studies focus only on image recognition tasks, in which *argmax* of the class conditional probabilities determine decisions. Moreover, their synthetic images are visually unnatural and difficult to yield diversity since constrained mainly by category content. Thus, these existing data-free approaches for model compression cannot be employed on a direct basis.

In this paper, we propose a novel data-free deraining model compression framework by exploring the statistical priors learned from pre-trained networks. This method is based on our statistical observations that image deraining networks can learn deep representations for degradation characteristics (rain types) independent of image content (detailed in Sec. 3.1). This motivates us to reconstruct diverse and in-distribution degraded images and thus provide sufficient supervision to compensate for the statistical drift of the compressed model without original data.

Specifically, given a pre-trained image restoration model (teacher), we utilize two branches in one stage to optimize both the *random noise input* and the *pruned model* (student). In one branch, the random noise forward through the fixed teacher model, then the output is forced to be close to the collected clean images (target) under the constraint of a *dream loss*. Meanwhile, we employ the orthogonality regularization of the deep features along the batch dimension to yield diverse degradation characteristics. In another branch, these input and output pairs are employed to distill the pruned model supervised by a *knowledge distillation loss*. Further, to alleviate the statistical drift caused by pruning before distillation, an adaptive pruning scheme is proposed to determine the hierarchical sparsity by constructing an explicit metric for pruning sensitivities of different layers combined with our reconstruction. The contributions of this paper are summarized as follows:

1) We find that pre-trained deraining networks can learn deep degradation representations that are independent of image content. Thus, we propose a novel data-free compression framework for deraining networks, in which diverse degraded images are reconstructed and utilized to distill the pruned model.

2) We further propose an adaptive pruning scheme to determine the hierarchical sparsities and moderate the statistical drift of the pruned model before fine-tuning.

3) Experiments on various deraining datasets demonstrate that our method can compress about 40% FLOPs of the state-of-the-art models while maintaining comparable performance without original data.

2. Related Work

Single Image Deraining. Image deraining is an ill-posed problem and therefore challenging, traditional methods explore degradation priors to obtain solutions, including orientation histogram [2], spatio-temporally correlation [6], structural similarity [26], discriminative sparse coding [20], etc. These handcrafted priors tend to rely on empirical observations and thus are not generalizable. Recently, CNNs have made significant achievements in image deraining. A deep detail network [10] is first proposed to remove rain from single images. Yang et al. [29] jointly detect and remove rain streaks using a multi-stream network. More complicated CNN-based models are designed for better performance, such as [5, 17, 25, 31, 32]. In addition, Qian et al. [24] use visual attention with a generative adversarial network to address a different problem of removing raindrops from single images. However, with the requirements to handle various degradation characteristics, such as different rain patterns in various densities and orientations, those performance-designed CNN models [5, 31] typically suffer from tremendous computational costs.

Data-Free Model Compression. Many data-free model compression methods have been proposed in recent years to alleviate the requirement for source training data. For example, Lopes et al. [19] first propose to use meta data to reconstruct the original training samples for knowledge distillation, and Nayak et al. [22] explore the information of pre-trained models to synthesize useful training samples. DeepInversion [30] achieves better reconstruction by introducing the statistics of BatchNorm layers based on DeepDream. Chen et al. [4] retrain a generator based on the pretrained network for synthesizing training samples to provide distillation supervision. However, these methods focus only on recognition tasks, and generally, adopt one-hot constraints thus cannot be applied to dense regression tasks like image deraining. In addition, their reconstructed images appear unnatural since the reconstruction process is supervised by semantics merely. Zhang et al. [36] retrain a generator for super-resolution tasks to synthesize training samples for distilling a smaller student network. However, they utilize down-sampling to supervise the reconstruction, which does not conform to the degradation process of rain.

Our proposed method differs from them in two main ways. Firstly, we reconstruct degraded images with both

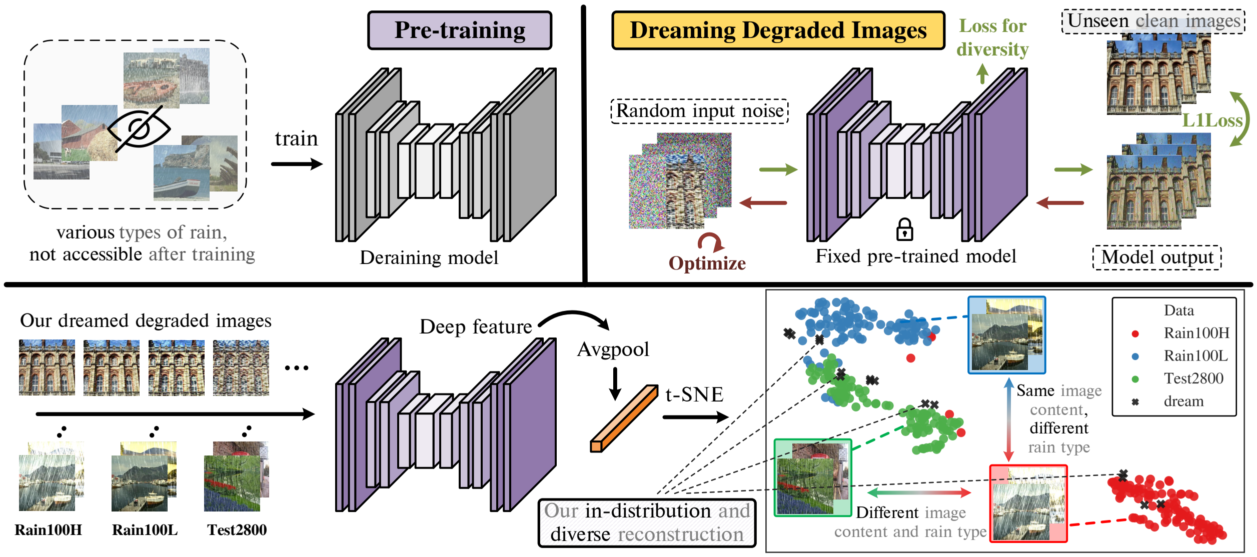

Figure 2. We observed that deep representations of pre-trained deraining networks can be clustered by degradation characteristics (types of rain) while independent of image content. This motivates us to invert the pre-trained deraining models and constrain the orthogonality of their degradation representations to "dream" in-distribution and diverse samples for data-free distillation.

natural appearance and diverse degradation types, which provides sufficient supervision for model compression to maintain the performance on various degradation characteristics. Secondly, we optimize the reconstruction and distillation in one stage with computational efficiency.

3. Data-Free Pruning for Image Deraining

To maintain the original deraining performance while compressing a pre-trained network without any assumptions or dependencies on the source training data, this paper proposes a novel data-free compression framework.

3.1. Motivation

Convolutional image deraining models can learn strong statistical priors to map from degraded (rainy) image distribution to clean image distribution. While model compression inevitably impairs the original statistical modeling, which is difficult to be compensated without fine-tuning. In the absence of source training data, a natural idea is to employ knowledge distillation that leverages the statistical priors learned by the original bulky model (teacher) to fine-tune the compressed model (student).

In the case that the reconstructed samples obey the original degraded image distribution that can be mapped to clean images by the teacher, thus providing satisfactory distillation supervision. However, this distribution is typically difficult to be formulated, leading to the questions: *Given only a pre-trained model, how to reconstruct the in-distribution degraded images?* We analyze that those images are required to satisfy two main conditions. First, these images should appear with natural textures and features. More importantly, these degradation characteristics should be as diverse as possible within the preferences of the teacher model. Hence, it seems that we attempt to solve:

$$I_d = \phi^{-1}(I_c), \tag{1}$$

where ϕ denotes the learned deraining mapping from the degraded image distribution p_d to the clean image distribution p_c. In practice, as the clean image I_c can be easily sampled, the degraded image I_d ($\sim p_d$) ought to be optimized by the model inversion paradigm similar to DeepDream [1], which can be formulated as:

$$I_d = \arg\min_{x} \mathcal{L}(\phi(x), I_c), \tag{2}$$

where x is optimized from random noise to image under the similarity constraint $\mathcal{L}$.

Although this is an ill-posed problem with the inherent nature of many-to-one, the optimal inverse (I_d) tends to lack diversity if no prior constraints are imposed. Our intuition is that the pre-trained deraining network should learn the degradation representations independent of the image content. Our observations confirm this and motivate us to obtain the regularization of degradation diversity. As shown in Fig. 2, we adopt 3 different rain datasets with 100 images selected separately in each, where Rain100H [29] and Rain100L [29] are identical in image content but different in rain types, and Test2800 [10] is different from both of them. We find that deep representations of the pre-trained deraining network (layer `cat12` with 64 dimensions in HINet [5]) can be clustered by rain types while independent of image

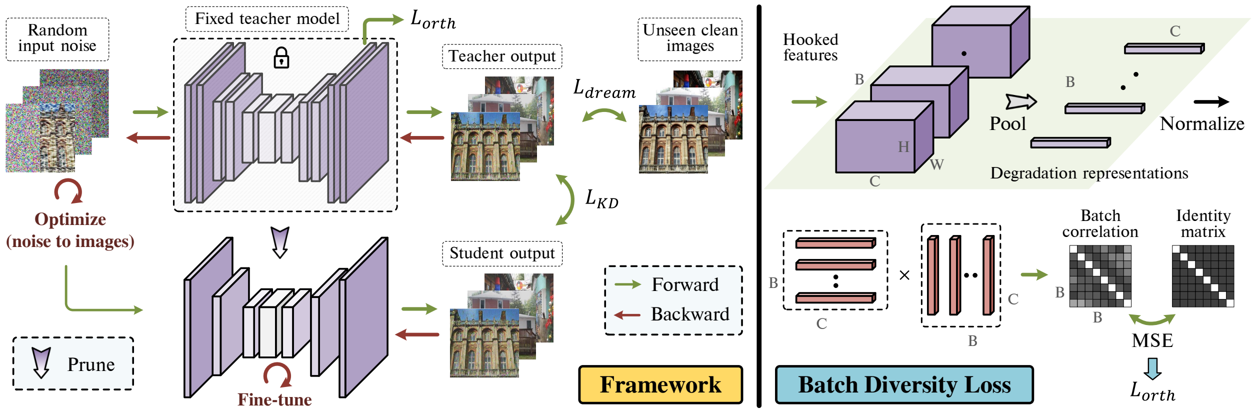

Figure 3. Overview of our data-free pruning framework. We jointly "dream" diverse in-distribution rainy images and distill the pruned model, which preserve the deraining performance without the original data.

content. Therefore, over Eq. (2), we employ the orthogonal constraint (detailed in Sec. 3.2) on a batch to yield a diversity of degradation representations. As shown by the black cross in the t-SNE [27] results of Fig. 2, our "dreamed" degraded images exhibit a variety of degradation characteristics and bridge the gap with the original distribution.

So far, these diverse and in-distribution data can provide sufficient supervision for distilling the compressed model. Further, we jointly perform the model inversion and the distillation in one-stage learning, and our framework is specified as follows.

3.2. Framework

Let $\mathcal{T}$ be a pre-trained image deraining model, typically with a huge amount of parameters. Let $\mathcal{S}$ be a pruned student network, which is more compact than $\mathcal{T}$. Overall, we employ two branches within the one-stage learning framework respectively inverting $\mathcal{T}$ to reconstruct the source degraded images and utilizing these data to fine-tune $\mathcal{S}$. Our overall method is illustrated in Fig. 3.

Dreaming Degraded Images. In this branch, we invert the pre-trained $\mathcal{T}$ by optimizing random input noise to images, which is analogous to resample from the original degraded image distribution.

Specifically, we collect a few clean natural images to construct the target set $\mathcal{Y}$. Given an arbitrary target image ($y \in \mathbb{R}^{H \times W \times C}$, H, W, C being the height, width, and color channels), the degraded image $\hat{x}$ is reconstructed by optimizing:

$$\min_{x} \mathcal{L}_{inv}(\mathcal{T}, x, y) + \lambda_{orth}\mathcal{L}_{orth}(x), \tag{3}$$

where $x \in \mathbb{R}^{H \times W \times C}$ and is initialized randomly. We can implement $\mathcal{L}_{inv}$ by calculating $\mathcal{L}\left(\mathcal{T}(x), y\right)$, where $\mathcal{T}(\cdot)$ denotes the output of the teacher model $\mathcal{T}$, and $\mathcal{L}(\cdot)$ is a similarity criterion loss (e.g., $\ell 1$ loss). And $L_{orth}(x)$ denotes the orthogonality constraint in the deep feature space of $\mathcal{T}$, which can be expressed as:

$$L_{orth}(x) = \left\| F \cdot F^{T} - I \right\|_{2}, \tag{4}$$

where $F \in \mathbb{R}^{B \times C}$ represents the deep feature after global average pooling and then normalizing, and $I \in \mathbb{R}^{B \times B}$ denotes the identical matrix. Based on our previous observations that F can represent the degradation type, we constrain its orthogonality to yield batch diversity, as shown in the right side of Fig. 3.

Then, the loss function of this dreaming branch can be formulated as:

$$\mathcal{L}_{Dream} = \mathbb{E}_{(x,y)\in\mathcal{P}_{xy}}\ \mathcal{L}\left(\mathcal{T}(x),\, y\right) + \lambda_{orth}\mathcal{L}_{orth}(x), \tag{5}$$

where $\mathcal{P}_{xy}$ denotes the pairs combined with *learnable* degraded images and *fixed* clean images.

Knowledge Distillation. The knowledge distillation loss $\mathcal{L}_{KD}$ can be formulated as:

$$\mathcal{L}_{KD} = \mathbb{E}_{(x,y)\in\mathcal{P}_{xy}}\ \mathcal{L}\left(\mathcal{T}(x),\, \mathcal{S}(x)\right), \tag{6}$$

where $\mathcal{S}(x)$ denotes the output of the pruned student model $\mathcal{S}$. When naive pruning is performed, there is a subtle gap in statistical modeling between the pruned model $\mathcal{S}$ and the original model $\mathcal{T}$. Under the supervision of $\mathcal{L}_{\mathcal{KD}}$, we force $\mathcal{S}$ to approximate the original statistical modeling of $\mathcal{T}$.

3.3. Adaptive Pruning Scheme

Further, we observed that different component modules of $\mathcal{T}$ exhibit different pruning sensitivities. For example, as shown in Fig. 4, given a pre-trained derain model (HINet [5]) with a test rainy image, we perform individual weights pruning based on $\ell 1$ regularization [13], for *each module* (x-axis) of HINet with *varying sparsities* (y-axis),

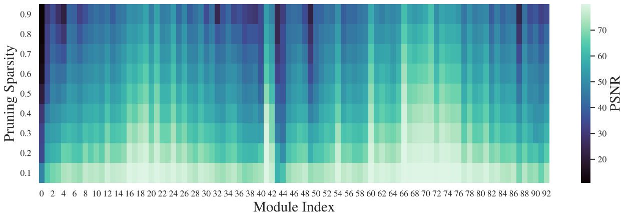

Figure 4. *PSNR results* (heatmap) of pruning *each module* (x-axis) separately with different *pruning sparsities* (y-axis).

which yields quite different *PSNR results* (heatmap). However, most of previous methods implement implicit measures of the parameter importance, which are difficult to be applied in image deraining, especially without source training data. To reduce the statistical drift of the pruned model $\mathcal{S}$ and alleviate the pressure of knowledge distillation, we propose an adaptive pruning scheme. As shown in Algorithm 1, our algorithm consists of two main steps as follows.

First, we explicitly measure the performance drop of the pruned model in the absence of the original data. Specifically, we invert the model $\mathcal{T}$ to reconstruct (dream) a batch of degraded images $\mathcal{X}$ using the collection $\mathcal{Y}$. Then, we can evaluate that a pruning sparsity for a given module is *acceptable* if and only if the average PSNR is greater than our threshold, where this result is calculated by the output pairs from $\mathcal{T}$ and $\mathcal{S}$ on $\mathcal{X}$. Second, based on the above measurement, we hierarchically search the optimal pruning rate for all sub modules of the pre-trained model. For each given sparsity rate, we can perform pruning using a common regularization, such as L1 [13]. To speed up the search process, we introduce dichotomous approach by starting from the midpoint of the sparsity interval in each iteration and then searching up by half if *acceptable* otherwise down.

3.4. Overall Optimization

Equipped with the adaptive pruning scheme, our framework exhibits enhanced performance, and the overall optimization is summarized as follows.

First, we prune the pre-trained image deraining model $\mathcal{T}$ utilizing our adaptive pruning scheme, as shown in Algorithm 1, and obtain a naive pruned model $\mathcal{S}$ which still suffers from moderate performance drop before fine-tuning.

Then, we employ the proposed framework to distill the pruned model S without source training data, as shown in Sec. 3.2. This one-stage loss function can be expressed as:

$$\mathcal{L}_{total} = \underbrace{\mathcal{L}_{inv} + \lambda_{orth}\mathcal{L}_{orth}}_{\mathcal{L}_{Dream}} + \lambda_{KD}\mathcal{L}_{KD}. \tag{7}$$

When this total loss converges after optimization, $\mathcal{S}$ can be distilled by $\mathcal{T}$ using diverse in-distribution images.

Algorithm 1: Adaptive Pruning Scheme

```
Input  : A pre-trained image deraining model $\mathcal{T}$
Output: Hierarchical adaptive sparsities
Collect natural clean images $\mathcal{Y}$ ;
Dream $\mathcal{X} \leftarrow \left\{\arg\min_{x} \mathcal{L}\left(\mathcal{T}(x), y\right) \mid y \in \mathcal{Y}\right\}$ ;
Initialize PSNR threshold $tp$;
Initialize sparsity precision $\epsilon \leftarrow 1 \cdot e^{-3}$ ;
Initialize student model $\mathcal{S} \leftarrow \mathcal{T}$ ;
foreach module in $\mathcal{S}$.modules do
    Initialize $[l, r] \leftarrow [0, 1]$ ;      /* interval */
    while $r - l \geq \epsilon$ do
        $spa \leftarrow (l + r) / 2$ ;        /* sparsity */
        TryPrune(module, spa) ;
        $psnr \leftarrow$ AveragePSNR $(\mathcal{S}(\mathcal{X}), \mathcal{T}(\mathcal{X}))$ ;
        if $psnr \geq tp$ then
            $l \leftarrow spa$ ;           /* search up */
        else
            $r \leftarrow spa$ ;       /* search down */
        end
    end
    AppendSparsity(spa) ;
end
```

4. Experiments

In this section, we evaluate our data-free pruning based on state-of-the-art methods across various image deraining datasets, and analyze the effectiveness of our method.

4.1. Implementation

Datasets. In order to emulate the complexity and diversity of rain types in real-world scenarios, and explore a more generalizable approach, we attempt to evaluate on datasets with as diverse rain types as possible. First, we conduct experiments on five validation rain datasets, respectively **Test2800** [10], **Test1200** [32], **Test100** [33], **Rain100H** [29], and **Rain100L** [29]. State-of-the-art deraining methods [5, 31] are proposed to handle these five datasets (for simplicity, denoted as **Rain13k** in the following) simultaneously. In addition, considering that real-world camera sensors or glass windows may be obscured by raindrops, we evaluate on **RainDrop** [24] which is captured with various background scenes and raindrops. Following [5, 24, 31], we adopt *PSNR* and *SSIM* as the evaluation metrics for rain removal performance, where the *PSNR* is calculated on the Y channel in the YCbCr color space.

Details. We implement our approach with PyTorch. Adam optimizer is used for training, where the learning rate for image reconstruction is set to $5 \cdot 10^{-2}$, and the learn-

			Test2800 [10]		Test1200 [32]		Test100 [33]		Rain100H [29]		Rain100L [29]		Average	
Model	Method	FLOPs	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MPRNet [31]	$\ell 1$ [13]	87.9 G	27.81	0.844	26.32	0.781	24.41	0.782	16.07	0.459	27.72	0.831	24.47	0.739
	erk [8]		32.84	0.931	31.82	0.888	28.44	0.868	27.76	0.844	34.32	0.946	31.04	0.895
	lamp [15]		33.07	0.934	32.38	0.899	29.09	0.879	28.82	0.864	35.59	0.957	31.79	0.907
	Ours		**33.40**	**0.938**	**32.70**	**0.912**	**30.07**	**0.894**	**29.19**	**0.874**	**36.56**	**0.965**	**32.38**	**0.917**
	original	141.0 G	33.64	0.938	32.91	0.916	30.27	0.897	30.41	0.890	36.40	0.965	32.73	0.921
HINet [5]	$\ell 1$ [13]	100.0 G	29.34	0.887	27.49	0.831	24.92	0.816	18.66	0.599	29.49	0.888	25.98	0.804
	erk [8]		32.37	0.929	31.09	0.895	25.30	0.835	23.46	0.783	28.74	0.880	28.19	0.864
	lamp [15]		33.23	0.936	32.52	0.912	27.58	0.872	27.21	0.862	30.98	0.919	30.30	0.900
	Ours		**33.79**	**0.940**	**32.95**	**0.919**	**30.12**	**0.906**	**29.54**	**0.890**	**36.94**	**0.969**	**32.67**	**0.925**
	original	170.5 G	33.91	0.941	33.05	0.919	30.29	0.906	30.65	0.894	37.28	0.970	33.03	0.926

Table 1. Data-free pruning results on Test2800 [10], Test1200 [32] Test100 [33], Rain100H [29], and Rain100L [29]. We compressed the pre-trained state-of-the-art deraining models by the same FLOPs, for a fair comparison to the most classical ($\ell 1$ [13]) and modern (erk [8], lamp [15]) pruning methods. Best scores of pruned and original model are **highlighted** and underlined, respectively.

ing rate for fine-tuning models is set to $1 \cdot 10^{-4}$. We collect 20 auxiliary clean images for each batch, and random crop these images to 256×256. For hyper-parameters, we set $\{\lambda_{orth}, \lambda_{KD}\} = \{0.05, 1.0\}$ in our framework. We set PSNR threshold in Algorithm 1 to 50 for a better trade-off of FLOPs and performance. It takes only 3 epochs with 600 iterations per epoch on a NVIDIA GTX 3090 GPU.

4.2. Data-free Compression Results

Considering that the original data is not available, we mainly compare with the alternative magnitude-based pruning methods, including the most classical $\ell\mathbf{1}$ regularization [13], and the most modern methods of **erk** [8] and **lamp** [15]. Different from them, we design explicit metrics for weight importance and jointly perform data-free distillation while pruning the deraining models. It is worth noting that smaller models sometimes yield larger computations due to different designs, such as progressive architectures [5, 31] or attention operations [34, 35]. Therefore, we mainly evaluate FLOPs of all methods, which are more reflective of practical computational cost than model size. And to make the comparison fair, we ensure all the pruned models to hold the same FLOPs, calculated with the input size of $1 \times 3 \times 256 \times 256$ using `nni` [1] toolkit.

Rain13k. We adopt **HINet** [5] and **MPRNet** [31], which achieve state-of-the-art results and outperform other methods by a margin. As shown in Table 1, our approach compresses the original HINet by 41.3% FLOPs , with drop of only 0.36 dB PSNR and 0.001 SSIM. We also reduce MPRNet by 37.7% flops, with drop of only 0.35 dB PSNR and 0.004 SSIM. And we note that, with the same FLOPs, the compressed model size are also approximate for all methods: 77% − 80% of HINet, and 38% − 41% of MPRNet.

			TestSet A		TestSet B	
Model	Method	FLOPs	PSNR	SSIM	PSNR	SSIM
DuRN [18]	$\ell 1$ [13]	34.7 G	24.52	0.849	21.81	0.758
	erk [8]		28.63	0.906	24.07	0.803
	lamp [15]		29.54	0.908	24.69	0.807
	Ours		**30.42**	**0.918**	**25.00**	**0.813**
	original	55.9 G	31.24	0.926	25.32	0.817
AGAN [24]	$\ell 1$ [13]	41.8 G	14.61	0.725	13.77	0.639
	erk [8]		17.83	0.824	16.68	0.732
	lamp [15]		19.63	0.822	18.18	0.731
	Ours		**31.18**	**0.921**	**24.82**	**0.808**
	original	89.4 G	31.51	0.921	24.92	0.809

Table 2. Data-free pruning results on RainDrop dataset [24]. Best scores of pruned and original model are **highlighted** and underlined, respectively.

It can be seen that, with the same compressed computational cost, other pruning methods inevitably result in significant performance drop after pruning. In contrast, we achieve comparable performance with the original models on handling various types and scenarios of rain, while pruning without original data.

RainDrop. We adopt representative methods, including **AGAN** [24] which using adversarial training, and **DuRN** [18] designed with the "dual residual connection" style. As shown in Table 2, our approach reduce AGAN by 53.2% FLOPs, with drop of only 0.215 dB PSNR and 0.0005 SSIM. We also outperform other methods 0.595 dB PSNR and 0.007 SSIM for DuRNs. It can be noted that other pruning methods result in drastic performance drop for AGAN, while our method still maintain the original performance. We conjecture that this is because these methods may not fit for generative networks. On the contrary, we ex-

[1] `https://github.com/microsoft/nni`

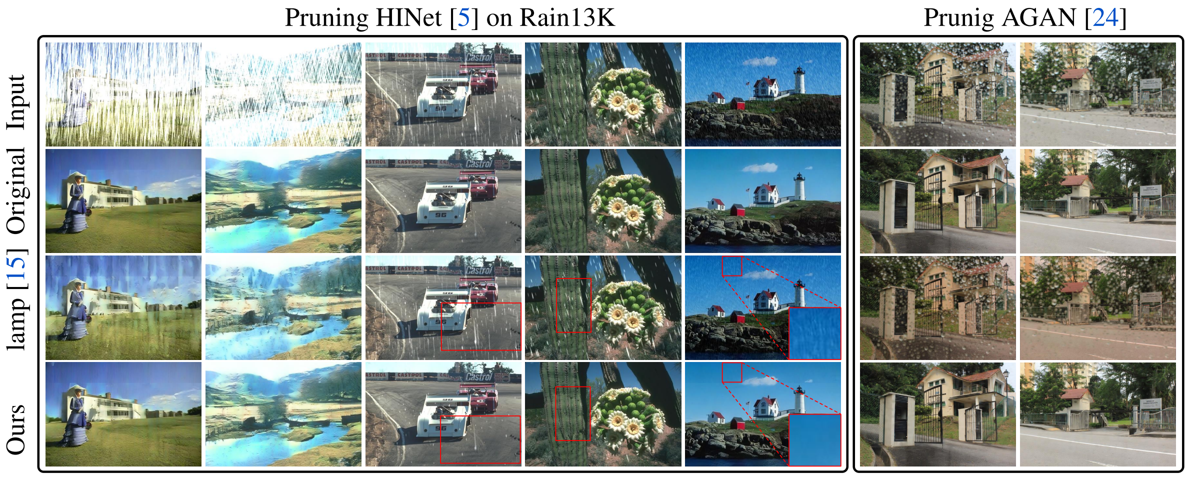

Figure 5. Qualitative results of pruning image deraining models. Our method preserve the performance of the pruned model on handling various degradation characteristics, and outperforms modern pruning method lamp [15] by a margin.

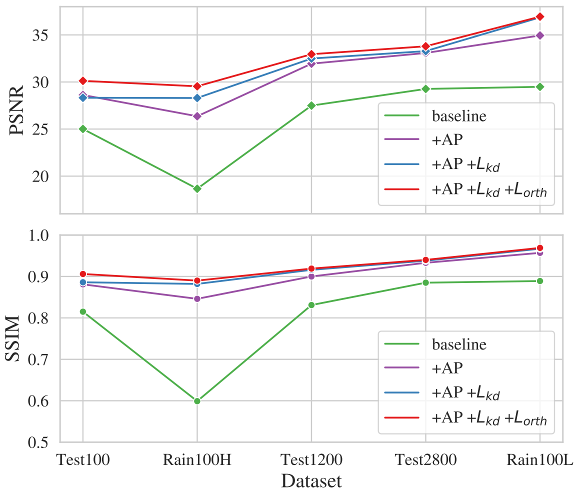

Figure 6. Ablation study on five test datasets. Our Adaptive Pruning scheme (AP) reduce the initial statistical drift of the pruned model without fine-tuning. And our dreaming for distillation (L_{kd}), and batch diversity loss (L_{orth}) are the keys to bridge the performance gap with the original model.

ploit strong priors learned by the pre-trained model to preserve the performance of the compressed model.

4.3. Ablation Study

To demonstrate the effectiveness of our approach, we conduct several ablation experiments on five validation datasets, including Test2800 [10], Test1200 [32], Test100 [33], Rain100H [29], and Rain100L [29]. In our approach, the performance of a pruned model is mainly attributed to two prominent components: the *adaptive pruning scheme* to moderate its initial statistical drift, and then *dreaming for distillation* to provide data-free compensation. Hence, we study the ablation of Adaptive Pruning (AP) and Knowledge Distillation ($\mathcal{L}_{kd}$). Moreover, to explore the impact of reconstruction diversity on distillation, we perform the ablation of batch diversity loss $\mathcal{L}_{orth}$.

Effectiveness of Adaptive Pruning Scheme. In this part, we explore the effectiveness of our adaptive pruning scheme. We adopt L1 regularization [13] as our baseline, since it is one of the most commonly used weights pruning methods. It sorts the weights of layers according to the $\ell 1$ norm and then removes the lowest given ratio among them. However this sparsity ratio often relies on handcrafted setting and search. To determine the appropriate sparsity ratios for different layers, we introduce our adaptive pruning scheme based on $\ell 1$ regularization. For a fair comparison, we adopt the same pre-trained HINet [5] with 170.5 G FLOPs, and ensure that all the models pruned with different methods achieve the same FLOPs (100.0 G). As shown in Fig. 6, adding AP outperforms the baseline average 5.01 dB PSNR and 0.09 SSIM, both without fine-tuning. This demonstrate that this scheme excels in estimating the different redundancy of different layers.

Effectiveness of Dreaming for Distillation. Although our adaptive pruning scheme can moderate the performance drop, the pruned model still struggles to maintain comparable performance with the original model. To address this issue, we proposed our framework to perform dreaming for distillation. As shown in Fig. 6, distillation alone (+$\mathcal{L}_{kd}$, blue line) in our framework brings an average of 0.9 dB PSNR and 0.015 SSIM improvement, compared to our initial pruning (without fine-tuning, purple line). This demonstrates that our data-free distillation enables the pruned model to re-fit the statistical modeling of the original model. And we notice that the improvement margin varies from dif-

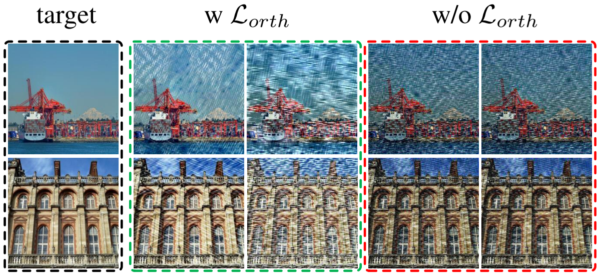

Figure 7. We can dream rainy images with diverse degradation characteristics using batch diversity loss L_{orth}, and thus provide generalizable distillation supervision.

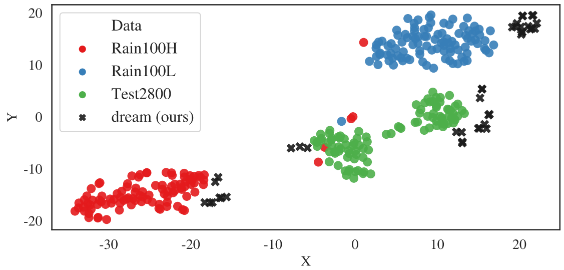

Figure 8. Our dreamed images using HINet [5] exhibit diversity in deep feature space of MPRNet [31] as well.

ferent datasets. For example, the improvement is smaller on Test2800 (0.29 dB), while larger on Rain100H (1.95 dB). The reason for this, in our analysis, is that rainy samples synthesized for distillation are not diverse enough.

Effectiveness of Dreaming Diversity. To increase the reconstruction diversity in our framework, we introduce a constraint ($\mathcal{L}_{orth}$) on the orthogonality of the deep features along the batch dimension. As can be seen in Fig. 6, our diversity loss (+$\mathcal{L}_{orth}$, redline) brings consistent improvements across five datasets. Even on Rain100H and Test100, challenging at previous stages, PSNR improvements of 1.3 and 1.8 dB are achieved, respectively. Combining with the table Table 1, we achieve a comparable performance to the original HINet after pruning, with an average drop of only 0.36 dB PSNR and 0.001 SSIM. We employ diverse dreamed images for knowledge distillation, and thus preserve the generalized performance of the pruned model on restoring images with various degradation characteristics similar to those ever trained.

4.4. Performance Analysis

Qualitative Results. As shown in Fig. 5, we display the performance comparison of pruning the pre-trained deraining models with our method and the modern weights pruning method (lamp [15]) on Rain13k and Raindrop datasets. The input rainy images (first row) appears various degradation characteristics. For a fair comparison, we ensure that the computational cost of the pruned models is compressed to that same level. We can see that, even without the original data, our pruning method can preserve the original capability on handling various degradation characteristics, including different rain patterns, directions, and densities.

Visualization of Dreaming. As shown in Fig. 7, we display the reconstructed images with dreaming approach. We collect several unseen clean images as the target (as shown in the black dashed box), and repeat them to form a batch for our inversion optimization. It can be seen that with the introduced $\mathcal{L}_{orth}$, these dreamed images appear various types of rain, including different orientations and densities. In contrast, without add $\mathcal{L}_{orth}$, those images optimized within a batch appear the similar styles. We conjecture that these local minima in the optimization space are related to some statistics of the entire source training dataset. And we employ the orthogonal regularization for the deep degradation representations and efficiently yield diverse rain types.

Analysis of Generalizability. To further verify the generalizability of our observed degradation priors, we obtain the degraded images dreamed by HINet and perform the t-SNE [27] clustering using their deep representations from MPRNet (the penultimate layer with 56 channels). As shown in Fig. 8, the deep representations of MPRNet can also be clustered by rain types while independent of the image content, which confirms the generalized degradation priors mentioned above. Those images dreamed by HINet exhibit diversity in deep feature space of MPRNet as well. It indicates that these reconstructions may share some common statistical properties with source training domain, thus can bridge the performance gap between the compressed model and the original model.

5. Conclusion

We propose a novel data-free deraining model compression framework. Firstly, based on our observations that deraining networks can learn the content-independent degradation representations, we invert the pre-trained model and constrain the orthogonality of their degradation representations to reconstruct diverse and in-distribution rainy data. Further, we jointly optimize the reconstruction and the distillation, thus preserving the performance of compressed models on handling various types of rain.

Acknowledgments. Supported by National Key R&D Program of China under Grant 2020AAA0105701, National Natural Science Foundation of China (NSFC) under Grants 61872327 and Major Special Science and Technology Project of Anhui (No. 012223665049).

References

[1] Mike Tyka Alexander Mordvintsev, Christopher Olah. Inceptionism: Going deeper into neural networks. https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html, 2015.

[2] Jérémie Bossu, Nicolas Hautiere, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011.

[3] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[4] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Dafl: Data-free learning of student networks. In *ICCV*, 2019.

[5] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–192, June 2021.

[6] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013.

[7] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019.

[8] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 13–18 Jul 2020.

[9] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive model inversion for data-free knowledge distillation. *arXiv preprint arXiv:2105.08584*, 2021.

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

[11] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32:1285–1296, 2019.

[12] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[13] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*, 2021.

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

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

[18] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 7007–7016, 2019.

[19] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks, 2017.

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

[21] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[22] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4743–4751. PMLR, 09–15 Jun 2019.

[23] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

[24] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019.

[26] Shao-Hua Sun, Shang-Pu Fan, and Yu-Chiang Frank Wang. Exploiting image structural similarity for single image rain removal. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4482–4486. IEEE, 2014.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

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

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

[30] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

[32] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018.

[33] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 30(11):3943–3956, 2019.

[34] Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020.

[35] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022.

[36] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7852–7861, June 2021.