# Learning A Spiking Neural Network for Efficient Image Deraining

**Tianyu Song**[1] , **Guiyue Jin**[1] , **Pengpeng Li**[2] , **Kui Jiang**[3] , **Xiang Chen**[2] and **Jiyu Jin**[1,*]
[1]Dalian Polytechnic University
[2]Nanjing University of Science and Technology
[3]Harbin Institute of Technology
songtienyu@163.com, guiyue.jin@dlpu.edu.cn, pengpengli@njust.edu.cn, kuijiang_1994@163.com,
chenxiang@njust.edu.cn, jiyu.jin@dlpu.edu.cn

## Abstract

Recently, spiking neural networks (SNNs) have demonstrated substantial potential in computer vision tasks. In this paper, we present an **E**fficient **S**piking **D**eraining **Net**work, called ESDNet. Our work is motivated by the observation that rain pixel values will lead to a more pronounced intensity of spike signals in SNNs. However, directly applying deep SNNs to image deraining task still remains a significant challenge. This is attributed to the information loss and training difficulties that arise from discrete binary activation and complex spatio-temporal dynamics. To this end, we develop a spiking residual block to convert the input into spike signals, then adaptively optimize the membrane potential by introducing attention weights to adjust spike responses in a data-driven manner, alleviating information loss caused by discrete binary activation. By this way, our ESDNet can effectively detect and analyze the characteristics of rain streaks by learning their fluctuations. This also enables better guidance for the deraining process and facilitates high-quality image reconstruction. Instead of relying on the ANN-SNN conversion strategy, we introduce a gradient proxy strategy to directly train the model for overcoming the challenge of training. Experimental results show that our approach gains comparable performance against ANN-based methods while reducing energy consumption by 54%. The code source is available at https://github.com/MingTian99/ESDNet.

## 1 Introduction

Single image deraining aims to reconstruct high-quality rain-free images from rain-affected ones. This challenging problem has witnessed significant advances due to the development of various effective image priors [Luo *et al.*, 2015; Li *et al.*, 2016] and deep learning models [Chen *et al.*, 2023b]. However, early prior-based methods typically necessitate numerous iterative optimization steps to ascertain the optimal solution, also constraining their practical applicability.

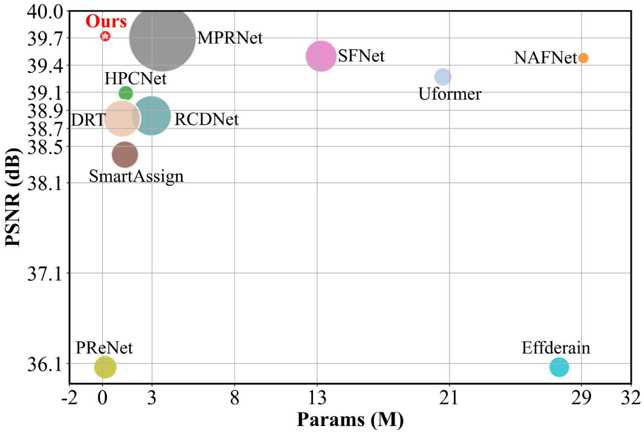


Figure 1: Model parameters and performance comparison between our proposed ESDNet and other deraining methods on the Rain200L dataset. The circle sizes represent the FLOPs of the various methods. The results show that the proposed method obtains a superior balance between model complexity and deraining performance.

In the deep learning era, numerous convolutional neural networks (CNNs) based models [Li *et al.*, 2018; Fu *et al.*, 2019; Ren *et al.*, 2019; Chen *et al.*, 2021] have been developed as better solutions for image deraining. The success of these methods greatly stems from their elaborate network architectures, such as multi-scale [Jiang *et al.*, 2020] and multi-stage [Zamir *et al.*, 2021]. However, these networks heavily rely on increasing the depth or complexity of models in order to expand the receptive fields and achieve better performance.

Recently, to facilitate global information aggregation, Transformers [Xiao *et al.*, 2022; Chen *et al.*, 2024b] have been applied to image deraining field and achieved significant advancements as they can model the non-local information for better image reconstruction. However, the computation of the scaled dot-product self-attention in Transformers results in quadratic space and time complexity as the number of tokens increases. To alleviate the computational and memory burden, some efforts have been proposed to reduce model parameters or complexity (FLOPs), such as pruning [Zou *et al.*, 2022], feature reuse [Fu *et al.*, 2019], pixel-wise dilation filtering [Guo *et al.*, 2021], token sparsification [Chen *et al.*, 2023a] and low-rank matrices [Xiong *et al.*, 2021].

Although these approaches have made initial explorations, achieving a better performance-efficient trade-off remains a

[*]Corresponding author.

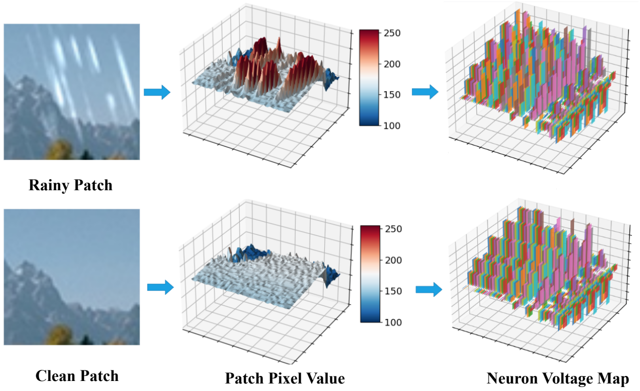


Figure 2: Visualization of image patch. From the pixel value map, it can be seen that the rain patch is generally higher than the surrounding areas without rain. The neuron voltage map represents the voltage heatmap learned by the LIF model. For the rain patches, the voltage changes of neurons are more drastic. This indicates that neurons are in an active state with a higher spike rate.

major challenge. Spiking neural networks (SNNs) [Zhou *et al.*, 2022; Su *et al.*, 2023; Qiu *et al.*, 2023], as the third generation of neural networks, which aim to simulate the way signals are transmitted in the brain. Unlike the artificial neural networks (ANNs) that directly learn embedding features from the input image, information in SNNs is represented as binary spike sequences and transmitted by neurons. Due to strong reflections caused by rain, the values of rain pixels in a rainy image are typically higher than those of their neighboring non-rain pixels. As shown in Figure 2, we observe that higher pixel values may result in a more pronounced intensity of spike signals in SNNs because they cause more neurons to fire spikes. This naturally arise a question: *whether we can harmonize the characteristics of rain perturbation and SNN for high-quality and low-energy-consuming rain removal?*

In fact, two challenges still need to be addressed when applying the SNN to the image deraining task. Firstly, to effectively learn complex and diverse rain streaks, the network needs to possess powerful information representation capabilities. Unfortunately, most existing SNN-based methods suffer from information loss caused by discrete binary activation, which may significantly interfere with the subsequent clear image reconstruction. To enhance the representation capability of SNNs, researchers have explored effective network structures, such as MS-ResNet [Hu *et al.*, 2021b] and SEW-ResNet [Fang *et al.*, 2021]. Nevertheless, these networks are not elaborately designed for image deraining task. Secondly, the non-differentiability of spike signals poses a challenge in using traditional backpropagation training strategies, causing difficulties in training the network. To address this issue, previous studies have employed the ANN-SNN conversion strategy. However, this strategy has limitations, as it heavily relies on the performance of the original ANNs and the converted SNN models fail to surpass the performance of the original artificial neural network (ANNs). A promising approach is to directly train SNNs using surrogate gradients, which can gain better performance with shorter time steps.

To this end, we propose an efficient and effective SNN-based method, called ESDNet, to solve the image deraining problem. Specifically, to tackle the weak representation capabilities and scale problems of SNN, we design a spiking residual block (SRB), which consists of a spiking convolution unit (SCU) and a mixed attention unit (MAU). SRB first converts the input image into spike signals, then adaptively optimizes the membrane potential by introducing attention weights to adjust spike responses, alleviating information loss caused by discrete binary activation. Furthermore, to overcome the difficulty of training SNNs, we apply the gradient surrogate strategy to directly train the model rather than the ANN-SNN conversion strategy. Compared to other ANN-based deraining models, our method exhibits lower energy consumption and superior performance (see Figure 1).

The main contributions can be summarized as follows:

- We propose a deep spike neural network (ESDNet) for single image deraining, and provide a new perspective on developing efficient models for resource-limited devices in real-world applications.
- We design a spiking residual block, which can effectively alleviate the problem of information loss caused by discrete binary activation and learn multi-scale rain streaks, facilitating high-quality image reconstruction.
- Extensive experimental results on single image deraining datasets display that our proposed model can obtain comparable rain removal performance against ANNs-based methods while reducing energy cost by 54%.

## 2 Related Work

### 2.1 Single Image Deraining

Traditional methods for image deraining explored various handcrafted prior knowledge to provide additional constraints for this ill-posed inverse problem. These methods involved complex optimization problems and only learned the characteristics of rain streaks without considering the underlying clear information [Ding *et al.*, 2016; Luo *et al.*, 2015]. Recently, existing most methods used CNN-based frameworks, treating image deraining as a pixel regression task and achieving significant performance gains. [Jiang *et al.*, 2020], [Zamir *et al.*, 2021] and [Chen *et al.*, 2024a] decomposed the deraining task into multiple subspaces, further acquiring multi-scale information to enhance image reconstruction quality.

With the successful application of Vision Transformer (ViT) in various advanced visual tasks, the ViT has also been applied to image rain removal, achieving better performance than CNN-based models. For example, Restormer [Zamir *et al.*, 2022] enhanced representation ability by calculating the similarity of tokens at the channel dimension. [Xiao *et al.*, 2022] used a hierarchical architecture of window-based attention to capture global correlations. Despite these methods achieving impressive reconstruction performance, their heavy computation burden made it difficult to deploy them in practical applications on resource-constrained devices.

To improve efficiency, many efforts have been made. [Fu *et al.*, 2019] constructed a lightweight network based on the Laplace pyramid architecture, which significantly improves the model's efficiency. [Guo *et al.*, 2021] proposed pixel-wise dilated filtering to predict multi-scale information cor-

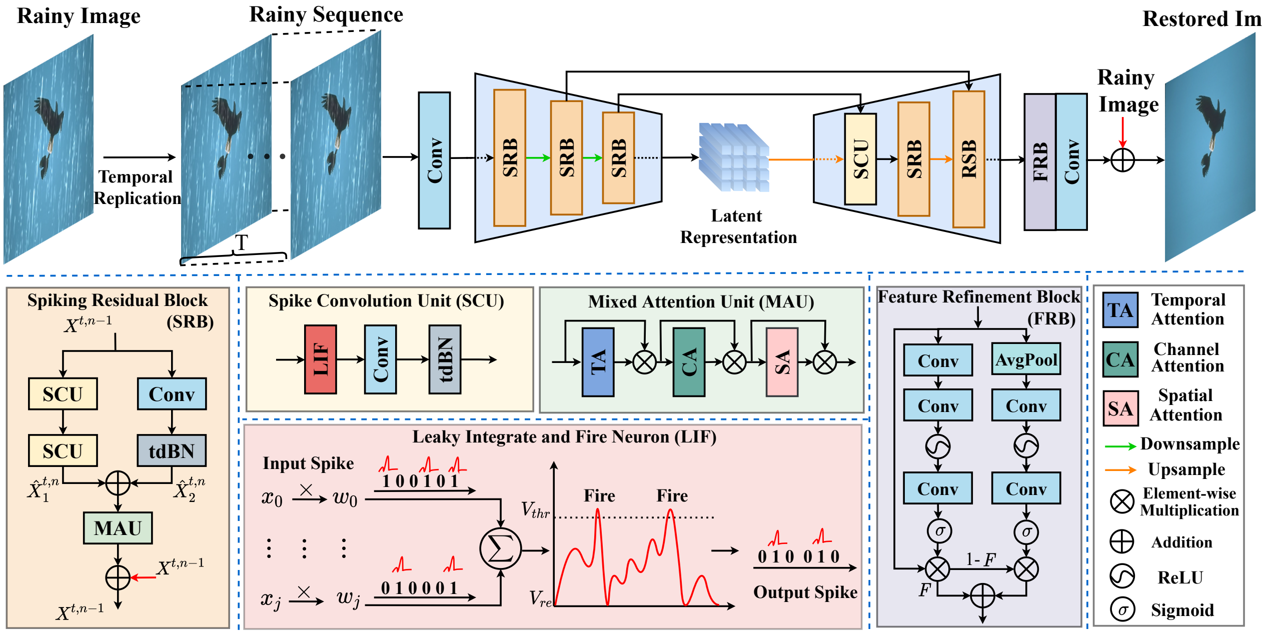


Figure 3: The architecture of the proposed Efficient Spiking Deraining Network (ESDNet), which takes input rainy images and generates output derained images. It mainly contains (1) the Spiking Residual Block (SRB) with Spike Convolution Unit (SCU) and Mixed Attention Unit (MAU), (2) Feature Refinement Block (FRB). tdBN refers to threshold-dependent batch normalization.

responding to each pixel and developed an efficient rain removal framework. [Ren *et al.*, 2019] designed a deraining network using a recursive computing architecture to accelerate model inference. In contrast to CNN-based methods that aimed to improve efficiency through carefully designed architectures, Transformer-based approaches focused more on optimizing attention to enhance computational efficiency. Sparse representation [Chen *et al.*, 2023a; Song *et al.*, 2023b] and low-rank matrix approximation calculation [Xiong *et al.*, 2021] were employed to compress and accelerate the inference speed of the model, improving efficiency. Although these methods have successfully reduced computational costs, they sacrifice the network's representational ability and cannot effectively explore spatial information, which may affect deraining performances. In contrast to these ANN-based rain removal methods, we propose using SNN with lower energy efficiency as a framework to achieve efficient image rain removal.

## 2.2 Spiking Neural Networks

Currently, there are two main routes to implementing deep SNNs. One approach is the ANN-SNN conversion strategy, which involves converting a well-trained and high-performance ANN into a SNN [Diehl *et al.*, 2015]. In this conversion strategy, the basic idea is to replace the ReLU activation functions in the ANN with spiking neurons to achieve the conversion to an SNN. However, this strategy has some inherent limitations. Firstly, achieving a conversion close to the performance of the ANN typically requires longer time steps, which leads to higher latency and extra computational consumption [Yao *et al.*, 2023b]. Secondly, the performance of the converted SNN is highly dependent on the original ANN and cannot surpass it. The other promising route is to utilize surrogate gradient functions to unfold the SNN over simulation time steps and directly train the SNN using backpropagation or spike timing dependent plasticity (STDP) [Wang *et al.*, 2023c]. Compared to the ANN-SNN conversion strategy, directly trained SNNs can significantly reduce the simulation time step, making them more attractive in terms of energy efficiency [Kim *et al.*, 2020]. For example, [Su *et al.*, 2023] achieved performance equivalent to an ANN with the same architecture using only 4 time steps. [Zheng *et al.*, 2021] proposed temporal delay batch normalization (tdBN), which greatly improved the depth of the SNN model. [Hu *et al.*, 2021b; Yao *et al.*, 2023a] applied SNNs to classification tasks and achieved good performance. Although some works have applied directly trained deep SNNs to regression tasks such as object detection, there has been little exploration in pixel-level regression tasks such as image deraining.

# 3 Proposed Method

In this section, we first offer an overview of the proposed ESDNet. Then, we introduce the spiking residual block (SRB) and the spike convolution unit (SCU). Next, we describe the feature refinement block (FRB). The training strategy and model optimization are described in the final section.

## 3.1 Data Preprocessing

Encoding to generate spike signals globally is a typical approach in SNNs for simulating pixel intensity signals in images. Considering the spatiotemporal properties of SNNs, we first perform direct encoding on the input rainy image $\mathcal{X}_t$ to generate a sequence $\mathbf{X} = \{\mathcal{X}_t\}_{t=1}^{T}$, *i.e.*, copying the single degraded image $\mathcal{X}_t$ as the input for each time step $t$.

### 3.2 Network Architecture

The proposed network architecture is shown in Figure 3. This network takes a single rainy image as input and adopts an encoder-decoder framework to learn hierarchical representations for capturing information in the network and achieving high-quality deraining output. First, for the rainy input, the network employs the encoding layer to generate the rainy sequence $\mathbf{X}$ and extract shallow features from the $\mathbf{X}$ by $3 \times 3$ convolutional layers. Then, the captured shallow features will be fed into N stacked SRBs to conduct spike transformation, feature extraction, and fusion of rain spike information through SCU and MAU. For SCU, the Leaky Integrate-and-Fire neurons (LIF) [Fang *et al.*, 2023; Deng *et al.*, 2022] convert and weight the inputs to generate output spike sequences, which consist of 0 and 1. The value of the output spike from LIF is 1 when the membrane potential exceeds the threshold, otherwise 0. Next, MAU can adaptively optimize the membrane potential by applying attention weights, leading to adjusting the spike response in a data-driven manner and learning the distribution of rain streaks.

After the decoder stage, the network adopts a FRB to convert the features extracted from the spike sequence into accurate continuous value representations, while refining the potential representations which are used for restoring clear images [Zheng *et al.*, 2023]. Finally, the network generates high-quality and clear image output from the obtained representations by applying a $3 \times 3$ convolutional layer. Furthermore, skip connections are employed to assist model training during the feature extraction and image reconstruction stages.

### 3.3 Spiking Residual Block

As previously mentioned, regions degraded with rain perturbation have higher pixel values than those of no rain regions, exhibiting noticeable fluctuations. Correspondingly, this may lead to pronounced firing spike rates of neurons in SNNs. Hence, we develop the spiking residual block (SRB) to learn the spike information from the converted rain streaks, which is displayed in Figure 3. We first adopt SCU to convert the inputs into a discrete spike sequence and mine the rain streak clues included in the spike sequence. Then, an additional branch comprising solely a convolutional layer and a tdBN layer [Zheng *et al.*, 2021] is constructed to mitigate the significant decline in performance resulting from the inadequate information conveyed by SNNs. Furthermore, inspired by the previous work [Yao *et al.*, 2023b], we introduce MAU to augment the spike activity rate of neurons and regulate spike responses, better guiding the process of eliminating rain streaks. Finally, residual learning is applied to SRB to alleviate the issues of gradient vanishing and explosion. Assuming that the output of the $n$-th SRB at the $t$-th time step is $\mathbf{X}^{t,n}$, the calculated process of SRB can be depicted as:

$$\begin{aligned}
\hat{\mathbf{X}}_1^{t,n} &= \mathrm{SCU}\left(\mathrm{SCU}\left(\mathbf{X}^{t,n-1}\right)\right), \\
\hat{\mathbf{X}}_2^{t,n} &= \mathrm{tdBN}\left(\mathrm{Conv}\left(\mathbf{X}^{t,n-1}\right)\right), \\
\mathbf{X}^{t,n} &= \mathrm{MAU}\left(\hat{\mathbf{X}}_1^{t,n} + \hat{\mathbf{X}}_2^{t,n}\right) + \mathbf{X}^{t,n-1},
\end{aligned} \tag{1}$$

tdBN refers to threshold-dependent batch normalization. It considers both temporal and spatial dimensions during the normalization process to relieve the gradient explosion. Profiting from the above designs, SRB can enable adaptively learning the spike fluctuations induced by rain streaks, thereby effectively simulating degradation in diverse regions.

### 3.4 Spike Convolution Unit

Compared to ANNs, which employ continuous decimal values for information propagation and calculation, SNNs exploit discrete binary spike values for communication. Spiking neurons can convert received continuous values into discrete spike sequences, encoding information with temporal precision. In this paper, we select LIF neurons to convert input signals into spike sequences, as they better balance biological characteristics and computational complexity [Guo *et al.*, 2023b]. The explicit dynamic equation for LIF neurons can be expressed mathematically as:

$$\begin{cases}
\boldsymbol{H}^{t,n} = \boldsymbol{U}^{t-1,n} + \frac{1}{\tau}\left(\boldsymbol{X}^{t,n} - \left(\boldsymbol{U}^{t-1,n} - V_{\text{reset}}\right)\right) \\
\boldsymbol{S}^{t,n} = \Theta\left(\boldsymbol{H}^{t,n} - V_{thr}\right) \\
\boldsymbol{U}^{t,n} = \left(\beta \boldsymbol{H}^{t,n}\right) \odot \left(\mathbf{1} - \boldsymbol{S}^{t,n}\right) + V_{\text{reset}} \boldsymbol{S}^{t,n},
\end{cases} \tag{2}$$

where $t$ and $n$ represent the $t$-th time step and the $n$-th layer. $\boldsymbol{H}^{t,n}$ represents membrane potential, which is generated by a combination of time input $\boldsymbol{U}^{t-1,n}$ and spatial input $\boldsymbol{X}^{t,n}$. $\tau$ is the membrane time constant. Among them, $V_{thr}$ represents the activation threshold, used to determine whether to output $\boldsymbol{S}$ or maintain it at 0. $\Theta(\cdot)$ denotes the step function. When $x \geq 0$, $\Theta(x) = 1$, otherwise $\Theta(x) = 0$. $V_{\text{reset}}$ denotes the potential reset after $\boldsymbol{S}$ activates the output. $\beta$ represents the decay factor and $\odot$ indicates the element-wise multiplication.

In Eq. 2, the spatial representation $\boldsymbol{X}^{t,n+1}$ of rain streaks is captured from the output spike sequence $\boldsymbol{U}^{t,n}$ by convolution operators. This calculation process can be defined as:

$$\boldsymbol{X}^{t,n+1} = \mathrm{tdBN}\left(\mathrm{Conv}\left(\boldsymbol{S}^{t,n}\right)\right), \tag{3}$$

where $\boldsymbol{S}^{t,n}$ denotes a spike tensor, which only included 0 and 1. tdBN refers to threshold-dependent batch normalization. Due to space limitations, specific descriptions of MAU are included in the supplementary materials.

### 3.5 Feature Refinement Block

To transform discrete pulse sequences into continuous pixel values, a typical approach is to apply mean sampling in the time dimension and then scale it to the input size. Nevertheless, this sampling process may result in the loss of some crucial structural elements, which could affect the final image restoration quality. To address this issue, we introduce a FRB that can adaptively aggregate sampled features and mitigate information loss. Given the input spike sequence $\boldsymbol{Y}$, the calculation process of FRB can be presented as:

$$\begin{aligned}
\hat{\boldsymbol{F}} &= \sigma\left(\mathrm{Conv}\left(\phi\left(\mathrm{Conv}(\mathrm{GAP}(\boldsymbol{Y}))\right)\right)\right), \\
\boldsymbol{F} &= \sigma\left(\mathrm{Conv}\left(\phi\left(\mathrm{Conv}(\mathrm{Conv}(\boldsymbol{Y}))\right)\right)\right), \\
\tilde{\boldsymbol{F}} &= \boldsymbol{Y} \odot \boldsymbol{F} + (1 - \boldsymbol{F}) \odot \hat{\boldsymbol{F}},
\end{aligned} \tag{4}$$

where $\tilde{\boldsymbol{F}}$ refers to the final output feature and $\mathrm{GAP}(\cdot)$ denotes the global average pooling operation. $\sigma(\cdot)$ and $\phi(\cdot)$ indicate

| Methods | | Rain12 | | Rain200L | | Rain200H | | Rain1200 | | Average | | Params(M)↓ | FLOPs(G)↓ | Energy(uJ)↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | | | |
| C | RESCAN | 35.48 | 0.9499 | 33.82 | 0.9547 | 26.22 | 0.8219 | 32.60 | 0.9271 | 32.03 | 0.9134 | **0.149** | 32.119 | $4.014\times10^5$ |
| | PReNet | 36.32 | 0.9593 | 36.06 | 0.9735 | 27.18 | 0.8698 | 33.39 | 0.9280 | 33.24 | 0.9327 | 0.168 | 66.249 | $8.281\times10^5$ |
| | RCDNet | 37.67 | 0.9612 | 38.84 | 0.9854 | 28.98 | 0.8889 | 32.68 | 0.9189 | 34.54 | 0.9386 | 2.958 | 194.501 | $2.431\times10^6$ |
| | MPRNet | 37.55 | 0.9665 | 39.82 | 0.9863 | 29.94 | 0.8999 | 34.50 | 0.9369 | 35.46 | 0.9474 | 3.637 | 548.651 | $6.858\times10^6$ |
| | Effderain | 36.12 | 0.9588 | 36.06 | 0.9731 | 26.11 | 0.8341 | 32.85 | 0.9147 | 32.78 | 0.9202 | 27.654 | 52.915 | $6.614\times10^5$ |
| | NAFNet | 37.53 | 0.9639 | 39.48 | 0.9847 | 29.19 | 0.8880 | 34.69 | 0.9367 | 35.22 | 0.9433 | 29.102 | 16.064 | $2.008\times10^5$ |
| | SFNet | 36.11 | 0.9503 | 39.50 | 0.9850 | 29.75 | 0.9008 | 34.51 | 0.9383 | 34.97 | 0.9437 | 13.234 | 124.439 | $1.555\times10^6$ |
| T | DRT | 37.74 | 0.9674 | 38.81 | 0.9830 | 28.67 | 0.8796 | 33.88 | 0.9283 | 34.78 | 0.9395 | 1.176 | 166.045 | $2.075\times10^6$ |
| | ELFformer | 35.06 | 0.9420 | 38.85 | 0.9800 | 28.93 | 0.8852 | 33.54 | 0.9360 | 34.09 | 0.9358 | 1.221 | 23.667 | $2.958\times10^5$ |
| | HPCNet | 36.84 | 0.9639 | 39.14 | 0.9847 | 29.17 | 0.8962 | 34.46 | 0.9378 | 34.90 | 0.9378 | 1.411 | 29.540 | $3.692\times10^5$ |
| | SmartAssign | 36.87 | 0.9618 | 38.41 | 0.9814 | 27.71 | 0.8536 | 33.11 | 0.9154 | 34.03 | 0.9281 | 1.359 | 90.386 | $1.129\times10^6$ |
| S | Ours | **37.82** | **0.9681** | **39.85** | **0.9869** | **30.01** | **0.9132** | **34.52** | **0.9388** | **35.55** | **0.9518** | 0.165 | **7.320** | $\mathbf{9.150}\times10^4$ |

Table 1: Comparison of quantitative results on four synthetic benchmarks. **C**, **T** and **S** refer to the CNN-based, Transformer-based and SNN-based baselines, respectively. **Blod** and underline indicate the best and second-best results, respectively.

the Sigmoid and ReLU functions. Subsequently, the output feature of the FRB will be fed into a $3 \times 3$ convolutional layer to map it back to the original input resolution. By incorporating FRB into our proposed model, the network can better learn and refine the latent representation of clear images for restoring fine structural and textural details in images.

### 3.6 Training Strategies

In recent years, backpropagation-based training schemes have achieved significant progress in facilitating the training of SNNs, showing considerable superiority over transformation-based strategies. Since the non-differentiable nature of binary activations in SNN neurons, most existing methods use gradient-proxy functions to implement backpropagation training. Inspired by the previous wisdom, the Sigmoid function serves as the gradient-surrogate function learning to train SNNs. It can effectively handle binary neuron outputs and enable gradient propagation during backpropagation. The gradient surrogate function can be defined as:

$$\begin{aligned} \sigma(x) =& \frac{1}{1+e^{-\alpha x}}, \\ \sigma'(x) =& \alpha{\cdot}\sigma(x) \cdot (1-\sigma(x)), \end{aligned} \tag{5}$$

where $\alpha$ refers to a hyper-parameter, adopted to adjust the gradient of the surrogate function. The larger the value of $\alpha$, the greater the gradient of the function.

### 3.7 Model Optimization

To better recover more similar structure details, we apply SSIM loss as the reconstruction loss and train the model by minimizing it. The loss function can be depicted as:

$$\mathcal{L}_{ssim} = 1 - \mathrm{SSIM}(\mathcal{S}(\mathcal{X}), \mathcal{Y}), \tag{6}$$

where $\mathcal{S}(\cdot)$ is the proposed ESDNet, $\mathcal{X}$ and $\mathcal{Y}$ are the input rainy image and its corresponding ground-truth, respectively.

## 4 Experiments

In this section, we conduct comprehensive experiments on commonly used benchmark datasets to evaluate the effectiveness of the proposed method. The experiments involve rigorous testing and analysis, aiming to provide a thorough assessment of the performance. More results are included in the supplemental material. The training code and test models will be available to the public.

### 4.1 Experimental Setup

**Datasets**. Since the original comparison methods are trained on different datasets, we retrained all models on the four publicly available datasets (Rain12 [Li *et al.*, 2016], Rain200L [Yang *et al.*, 2017], Rain200H [Yang *et al.*, 2017], Rain1200 [Zhang and Patel, 2018]) to ensure a fair comparison of all methods. The Rain12 dataset consists of 12 synthetic rain images that were used for testing. The Rain200L and Rain200H datasets contain 1800 synthetic rain images for training, along with 200 images designated for testing. The Rain1200 dataset includes 12000 training images and 1200 test images, which have varying rain directions and density levels. Furthermore, we exploit the RW-Data [Zhang *et al.*, 2019], a real-world rain benchmark, consisting of 185 real-world rainy images.

**Evaluation Metrics**. In light of practical considerations, we select two common evaluation metrics, *e.g.,* Peak Signal to Noise Ratio (PSNR) [Hore and Ziou, 2010] and Structural Similarity (SSIM) [Wang *et al.*, 2004], to perform a quantitative comparison of the synthesized rainy dataset. Following the previous approaches [Chen *et al.*, 2023a], we compute these metrics in the Y channel of the YCbCr space. For the real-world rain dataset, we adopt non-reference image quality evaluation metrics such as Perception-based Image Quality Evaluator (PIQE) [Venkatanath *et al.*, 2015] and Meta-learning-based Image Quality Assessment (MetaIQA) [Zhu *et al.*, 2020] for assessing performance.

**Comparison Methods**. We compare our proposed method with various image rain removal baselines, including seven CNN-based methods (*i.e.*, RESCAN [Li *et al.*, 2018], PreNet [Ren *et al.*, 2019], RCDNet [Wang *et al.*, 2020], MPRNet [Zamir *et al.*, 2021], Effderain [Guo *et al.*, 2021], NAFNet [Chen *et al.*, 2022], and SFNet [Cui *et al.*, 2022]) and four transformer-based networks (*i.e.*, DRT [Liang *et al.*, 2022], ELFormer [Jiang *et al.*, 2022], HPCNet [Wang *et al.*, 2023a], and SmartAssign [Wang *et al.*, 2023b]). For the recent representative method SmartAssign, we encounter a lack of available code. Hence, we recreate the corresponding network according to the provided information in their paper and retrain it. For other competitive methods, if the authors provide pre-trained models available, we utilize their online code to evaluate performance, otherwise, we will retrain them using the same settings as ESDNet to ensure a fair comparison.

| Methods | RESCAN | PReNet | RCDNet | MPRNet | Effderain | NAFNet | SFNet | DRT | ELFformer | HPCNet | SmartAssign | Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIQE↓ | 18.364 | 18.309 | 19.692 | 18.061 | 18.400 | 18.106 | 18.124 | 18.392 | <u>18.037</u> | 18.287 | 18.316 | **17.959** |
| MetaIQA↑ | 0.419 | 0.420 | 0.421 | <u>0.426</u> | 0.421 | 0.420 | 0.423 | 0.419 | 0.422 | **0.426** | 0.417 | 0.423 |

Table 2: Comparison of PIQE/MetaIQA scores on the RW-Data. **Blod** and <u>underline</u> indicate the best and second-best results, respectively.

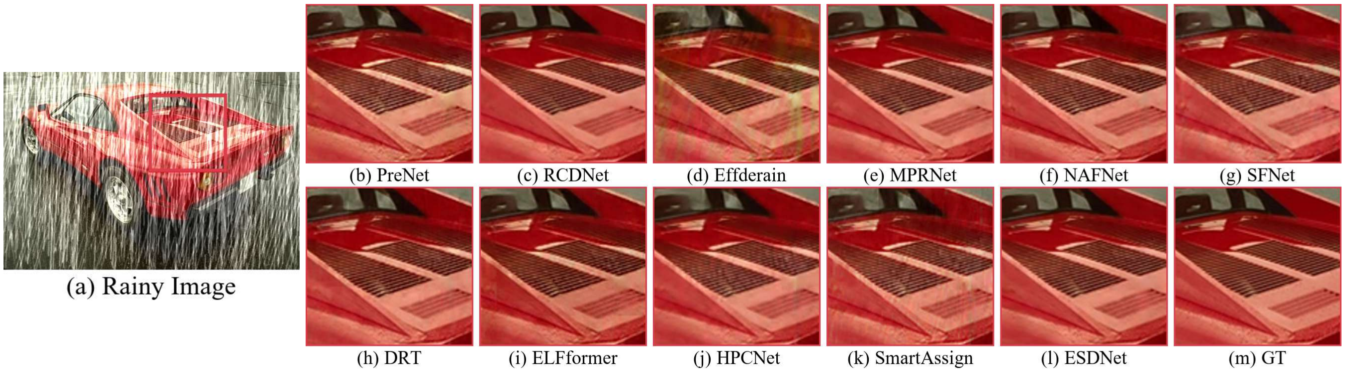

(a) Rainy Image
(b) PreNet
(c) RCDNet
(d) Effderain
(e) MPRNet
(f) NAFNet
(g) SFNet
(h) DRT
(i) ELFformer
(j) HPCNet
(k) SmartAssign
(l) ESDNet
(m) GT

Figure 4: Visual comparison of the Rain200H dataset. Best viewed by zooming in on the figures on high-resolution displays.

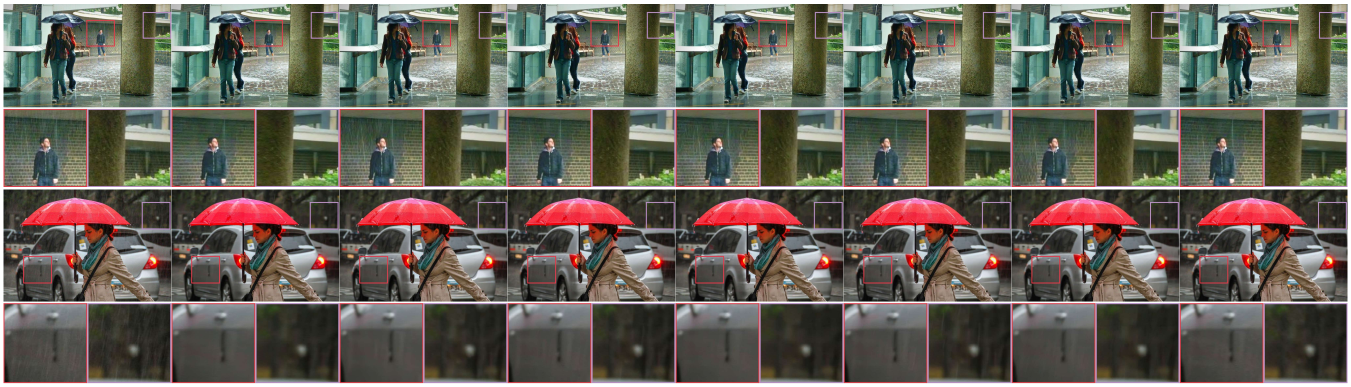
**Implementation Details.** During the training process, we conduct the proposed network in the PyTorch framework with an Adam optimizer and a batch size of 12. We set the learning rate to $1\times10^{-3}$ and apply the cosine annealing strategy [Song *et al.*, 2023a] to steadily decrease the final learning rate to $1\times 10^{-7}$. For Rain200L, Rain200H, and Rain1200 datasets, we train the model by 1000 epochs. We set the stacking numbers of SRB to [4,4,8] in the encoder stage and [2,2] in the decoder stage. For the $\alpha$ of the gradient proxy function, it is set to 4 according to [Su *et al.*, 2023]. All experiments are executed on an NVIDIA GeForce RTX 3080Ti GPU (12G). To ensure a fair comparison, for all retrained methods, we uniformly cut the data into 64 patch sizes for training, and apply the sliding window slicing strategy for testing.

## 4.2 Experimental Results

**Synthetic Datasets.** Table 1 provides a comprehensive comparison between our proposed method and 12 representative and competitive training methods. It is evident that the incorporation of spike convolution units in our ESDNet significantly improves the performance in terms of PSNR and SSIM values compared to all other baselines. Notably, our approach achieves more appealing results on the Rain200H benchmark, surpassing the recent CNN-based method SFNet by 0.26 dB in PSNR. The performance improvement, when compared to existing ANNs-based deraining methods, shows that our framework offers a fresh perspective for deep image training architectures. Furthermore, Figure 3 presents the qualitative evaluation results on the Rain200H benchmark. The visual comparison indicates that our method exhibits enhanced contrast and reduced color distortion in comparison to other approaches, corroborating the quantitative findings. Specifically, Effderain and DRT exhibit noticeable rain artifacts in Figure 3, while also tending to produce overly smooth results. In contrast, our model preserves more details and achieves superior perceptual quality.

**Real-world Datasets.** We conduct additional experiments using the RW-Data benchmark dataset. The obtained quantitative results are presented in Table 2. Analysis of these results reveals that our network achieves lower PIQE values compared to other comparative models. This indicates that in real rain scenarios, our network exhibits superior output quality with clearer content and enhanced perceptual quality. To further evaluate the qualitative performance, we implement a quality comparison analysis as shown in Figure 4. The results demonstrate that most models exhibit sensitivity to spatial long rain streaks, which often leads to noticeable rain effects in their outputs. In contrast, our network effectively eliminates a significant portion of rain disturbances, resulting in visually pleasing restoration effects. Overall, the results suggest that our network outperforms existing models in terms of output quality, perceptual clarity, and the ability to handle rain disturbances in real scenarios.

**Model Effciency**. In ANNs, each operation involves the multiply-accumulate operations (MAC) and floating-point operations (FLOPs) are used to estimate computational burden. SNNs exhibit energy-saving properties in neuromorphic hardware, as neurons only participate in accumulated operations (AC) during spikes and can complete AC through approximately the same number of synaptic operations (SOPs). In this work, we exploit 4 time steps to estimate energy costs. It is observed that the first convolutional encoding layer, RB block, and last convolutional encoding layer require floating-point operations due to they enjoy the matrix multiplication. The other layers are calculated through SOPs (synaptic operations). Following the previous works [Guo *et al.*, 2023a], SOPs can be computed through $s \times T \times A$, where s represents the average sparsity, $T$ is the time step, and $A$ is the additive number in the artificial neural network. For the binary SNNs, the sparsity is 16.42%. It is noted that SOPs can be triggered once when the sign function output is 1 due to the membrane potential exceeding the threshold. One sign (using energy per spike to calculate) needs 3.7pJ. Furthermore, the sign function only exists in LIF, and the number of LIF is limited compared to convolution operations. According to [Hu *et al.*, 2021a], we compute the energy as follows: a FLOP requires 12.5pJ, an SOP requires 77fJ, and a Sign (calculated using energy per spike) requires 3.7pJ.



(a) Rainy Image (b) Effderain (c) MPRNet (d) NAFNet (e) DRT (f) HPCNet (g) SmartAssign (h) ESDNet

Figure 5: Visual comparison on the RW-Data. Best viewed by zooming in the figures on high-resolution displays.

| Model | FLOPs (G) | Energy (uJ) | PSNR | SSIM |
|---|---|---|---|---|
| ANN-ResNet | 83.453 | $1.043\times10^6$ | 28.79 | 0.8942 |
| MS-ResNet | 6.970 | $8.730\times10^4$ | 25.20 | 0.8230 |
| SEW-ResNet | 6.998 | $8.765\times10^4$ | 26.61 | 0.8613 |
| SRB | 7.320 | $9.150\times10^4$ | 29.31 | 0.9033 |

Table 3: Ablation studies for different residual blocks in the model.

| Time steps | FLOPs (G) | Energy (uJ) | PSNR | SSIM |
|---|---|---|---|---|
| 1 | 7.023 | $8.779\times10^4$ | 28.69 | 0.8645 |
| 2 | 7.150 | $8.938\times10^4$ | 28.81 | 0.8963 |
| 4 | 7.320 | $9.150\times10^4$ | 29.31 | 0.9033 |
| 6 | 7.490 | $9.366\times10^4$ | 29.40 | 0.9040 |

Table 4: Impact of different time steps on Rain200H dataset.

| Model | BN | tdBN | MAU | FRB | PSNR | SSIM |
|---|---|---|---|---|---|---|
| (a) | ✓ | | | | 28.26 | 0.8779 |
| (b) | | ✓ | | | 28.49 | 0.8724 |
| (c) | ✓ | | ✓ | | 29.07 | 0.8984 |
| (d) | | ✓ | | ✓ | 28.59 | 0.8931 |
| (e) | | ✓ | ✓ | | 28.78 | 0.8968 |
| (f) | | ✓ | ✓ | ✓ | 29.31 | 0.9033 |

Table 5: Ablation studies for different designs in the model.

The results are displayed in Table 1. It can be observed that our network has achieved performance comparable to the recent ANNs-based deraining approaches, and has lower energy costs. In addition, we compared the computational complexity of various deep derivation transformer-based methods, including the number of trainable parameters and FLOPs on 256 × 256 images, as reported in Figure 1.

### 4.3 Ablation Studies

We conduct the ablation experiments to better understand the impact of different residual blocks, time steps, and model components on performance. Unless otherwise specified, all ablation experiments are conducted on the Rain200H dataset using the experimental setup described in Section 4.1.

**Effectiveness of Different Residual Blocks**. To investigate the impact of different residual modules on model performance, Table 3 presents the PSNR/SSIM values of the corresponding models. From the experimental results, we note that our SRB obtains a more attractive performance compared to the other two spiking residual blocks, thanks to the additional branches in SRB and the information compensation brought by MAU. By introducing a lower computational burden alone, the model performance is significantly improved.

**Effectiveness of Different Time Steps**. The performance results of the model with different time steps are reported in Table 4. According to the experimental results, it can be found that the longer the time step, the higher the quality of image restoration. This indicates that higher time steps can enhance the model to better obtain feature representations. Similarly, this also brings about a significant increase in energy costs. To better balance performance and efficiency, we chose a time step of 4 as the default time step in this work.

**Effectiveness of Different Designs**. To evaluate the effectiveness of the model designs, we conduct experiments based on different model variables in Table 5. Compared to the baseline model (a), model (b) provides additional performance advantages due to the consideration of temporal correlation. In addition, we note a significant decrease in performance when MAU is not applied, indicating that MAU can effectively enhance the spike activity rate of neurons and regulate spike responses.

## 5 Concluding Remarks

This paper presents an image deraining approach using spiking neural networks. The proposed network learns rain streak information by leveraging the spike convolution unit. Initially, continuous pixel values are transformed into spike sequences by LIF neurons. Convolution operations are then applied to learn the spike information associated with rain streaks. To enhance the accuracy of rain streak removal, a mixed attention unit is introduced, which adjusts spike responses based on the input data, thereby facilitating the extraction of rain streak clues. In addition, feature refinement blocks are incorporated to further improve the model representation for gaining high-quality images. The efficacy of the proposed method is demonstrated through extensive experiments conducted on synthetic and real rain datasets.

## References

[Chen *et al.*, 2021] X. Chen, J. Pan, K. Jiang, Y. Li, Y. Huang, C. Kong, L. Dai, and Z. Fan. Unpaired deep image deraining using dual contrastive learning. In *CVPR*, pages 2017–2026, 2021.

[Chen *et al.*, 2022] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pages 17–33, 2022.

[Chen *et al.*, 2023a] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *CVPR*, pages 5896–5905, 2023.

[Chen *et al.*, 2023b] Xiang Chen, Jinshan Pan, Jiangxin Dong, and Jinhui Tang. Towards unified deep image deraining: A survey and a new benchmark. *arXiv preprint arXiv:2310.03535*, 2023.

[Chen *et al.*, 2024a] Hongming Chen, Xiang Chen, Jiyang Lu, and Yufeng Li. Rethinking multi-scale representations in deep deraining transformer. In *AAAI*, volume 38, pages 1046–1053, 2024.

[Chen *et al.*, 2024b] Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. *arXiv preprint arXiv:2404.01547*, 2024.

[Cui *et al.*, 2022] Yuning Cui, Yi Tao, Zhenshan Bing, Wenqi Ren, Xinwei Gao, Xiaochun Cao, Kai Huang, and Alois Knoll. Selective frequency network for image restoration. In *ICLR*, 2022.

[Deng *et al.*, 2022] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

[Diehl *et al.*, 2015] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *IJCNN*, pages 1–8, 2015.

[Ding *et al.*, 2016] Xinghao Ding, Liqin Chen, Xianhui Zheng, Yue Huang, and Delu Zeng. Single image rain and snow removal via guided l0 smoothing filter. *MTA*, 75(5):2697–2712, 2016.

[Fang *et al.*, 2021] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *NeurIPS*, 34:21056–21069, 2021.

[Fang *et al.*, 2023] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023.

[Fu *et al.*, 2019] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *IEEE TNNLS*, 31(6):1794–1807, 2019.

[Guo *et al.*, 2021] Qing Guo, Jingyang Sun, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Wei Feng, Yang Liu, and Jianjun Zhao. Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In *AAAI*, volume 35, pages 1487–1495, 2021.

[Guo *et al.*, 2023a] Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. *arXiv preprint arXiv:2312.06372*, 2023.

[Guo *et al.*, 2023b] Yufei Guo, Xiaode Liu, Yuanpei Chen, Liwen Zhang, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Rmp-loss: Regularizing membrane potential distribution for spiking neural networks. In *ICCV*, pages 17391–17401, 2023.

[Hore and Ziou, 2010] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *IEEE ICIP*, pages 2366–2369. IEEE, 2010.

[Hu *et al.*, 2021a] Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE TNNLS*, 2021.

[Hu *et al.*, 2021b] Yifan Hu, Yujie Wu, Lei Deng, and Guoqi Li. Advancing residual learning towards powerful deep spiking neural networks. *arXiv e-prints*, pages arXiv–2112, 2021.

[Jiang *et al.*, 2020] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, pages 8346–8355, 2020.

[Jiang *et al.*, 2022] Kui Jiang, Zhongyuan Wang, Chen Chen, Zheng Wang, Laizhong Cui, and Chia-Wen Lin. Magic elf: Image deraining meets association learning and transformer. In *ACM MM*, pages 827–836, 2022.

[Kim *et al.*, 2020] Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation-and timing-based learning rules for spiking neural networks. *NeurIPS*, 33:19534–19544, 2020.

[Li *et al.*, 2016] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *ICCV*, pages 2736–2744, 2016.

[Li *et al.*, 2018] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018.

[Liang *et al.*, 2022] Yuanchu Liang, Saeed Anwar, and Yang Liu. Drt: A lightweight single image deraining recursive transformer. In *CVPRw*, pages 589–598, 2022.

[Luo *et al.*, 2015] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015.

[Qiu *et al.*, 2023] Xuerui Qiu, Rui-Jie Zhu, Yuhong Chou, Zhaorui Wang, Liang-jian Deng, and Guoqi Li. Gated attention coding for training high-performance and efficient spiking neural networks. In *AAAI*, 2023.

[Ren *et al.*, 2019] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, pages 3937–3946, 2019.

[Song *et al.*, 2023a] Tianyu Song, Shumin Fan, Pengpeng Li, Jiyu Jin, Guiyue Jin, and Lei Fan. Learning an effective transformer for remote sensing satellite image dehazing. *IEEE GRSL*, 2023.

[Song *et al.*, 2023b] Tianyu Song, Pengpeng Li, Guiyue Jin, Jiyu Jin, Shumin Fan, and Xiang Chen. Image deraining transformer with sparsity and frequency guidance. In *ICME*, pages 1889–1894, 2023.

[Su *et al.*, 2023] Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *ICCV*, pages 6555–6565, 2023.

[Venkatanath *et al.*, 2015] N Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. In *NCC*, pages 1–6, 2015.

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[Wang *et al.*, 2020] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, pages 3103–3112, 2020.

[Wang *et al.*, 2023a] Qiong Wang, Kui Jiang, Jinyi Lai, Zheng Wang, and Jianhui Zhang. Hpcnet: A hybrid progressive coupled network for image deraining. In *ICME*, pages 2747–2752, 2023.

[Wang *et al.*, 2023b] Yinglong Wang, Chao Ma, and Jianzhuang Liu. Smartassign: Learning a smart knowledge assignment strategy for deraining and desnowing. In *CVPR*, pages 3677–3686, 2023.

[Wang *et al.*, 2023c] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *ICCV*, pages 1761–1771, 2023.

[Xiao *et al.*, 2022] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE TPAMI*, pages 1–18, 2022.

[Xiong *et al.*, 2021] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, volume 35, pages 14138–14148, 2021.

[Yang *et al.*, 2017] 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 *CVPR*, pages 1357–1366, 2017.

[Yao *et al.*, 2023a] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *arXiv preprint arXiv:2307.01694*, 2023.

[Yao *et al.*, 2023b] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE TPAMI*, 2023.

[Zamir *et al.*, 2021] 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*, pages 14821–14831, 2021.

[Zamir *et al.*, 2022] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022.

[Zhang and Patel, 2018] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multistream dense network. In *CVPR*, pages 695–704, 2018.

[Zhang *et al.*, 2019] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE TCSVT*, 30(11):3943–3956, 2019.

[Zheng *et al.*, 2021] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *AAAI*, volume 35, pages 11062–11070, 2021.

[Zheng *et al.*, 2023] Yu Zheng, Jiahui Zhan, Shengfeng He, Junyu Dong, and Yong Du. Curricular contrastive regularization for physics-aware single image dehazing. In *CVPR*, pages 5785–5794, 2023.

[Zhou *et al.*, 2022] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, YAN Shuicheng, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *ICLR*, 2022.

[Zhu *et al.*, 2020] Hancheng Zhu, Leida Li, Jinjian Wu, Weisheng Dong, and Guangming Shi. Metaiqa: Deep meta-learning for no-reference image quality assessment. In *CVPR*, pages 14143–14152, 2020.

[Zou *et al.*, 2022] Weiqi Zou, Yang Wang, Xueyang Fu, and Yang Cao. Dreaming to prune image deraining networks. In *CVPR*, pages 6023–6032, 2022.