WDFSR: Normalizing Flow based on Wavelet-Domain for Super-Resolution

Chao Song, Shaobang Li, Bailin Yang
Zhejiang Gongshang University
Hangzhou, 310018, CN
ybl@zjsu.edu.cn

Frederick W.B. Li
Durham University
Durham, UK
rederick.li@durham.ac.uk

Abstract

We propose a Normalizing flow based on the Wavelet framework for super-resolution called WDFSR. The framework learns the conditional distribution mapping between low-resolution images in the RGB domain and high-resolution images in the wavelet domain to generate high-resolution images of different styles simultaneously. To address the problem that some flow-based models are sensitive to datasets and weak in generalization, we design a method that combines T-distribution and QR decomposition layers to mitigate the problem while maintaining the performance of the model. We also propose the Refinement layer combined with attention mechanism to refine the extracted condition features for performance improvement. Extensive experiments on many super-resolution datasets show that WDFSR outperforms most general CNN models and flow-based models in terms of PSNR and Perception quality. We also demonstrate that our framework works well for other low-level vision tasks, such as low-light enhancement.

Keywords: ***Normalizing flow, super-resolution, wavelet domain, attention mechanism, generative model***

1 Introduction

Super-resolution(SR) aims at recovering a high-resolution image from one or many low-resolution input images, which has a wide range of applications in many domains, such as film and television, art, and cultural relics protection, etc. A great deal of excellent research has been done in this field. Recently, based on traditional vision methods and machine learning methods, neural networks have made great progress in solving super-resolution problems. In particular, a class of generative methods represented by GAN has achieved good results, such as [28][46]. However, for its ill-posed properties, super-resolution is still a challenging computer vision problem, leaving room for performance and image quality improvement.

Normalizing flow is a reversible probabilistic generative model that has gained an increasing attention in the image field due to its powerful generative capabilities. Unlike GAN, Normalizing flow can explicitly compute probabilistic likelihoods. Compared to VAE, it can sample and compute distributions more accurately and has a faster computational speed than autoregressive models. In recent years, Normalizing flow has been applied to many generation tasks and achieved good results, such as point cloud [25], audio [44], and image generation tasks [21]. However, some flow-based models have out-of-distribution (OOD) generalization problems, being sensitive to datasets, and may lead to possible training instability. It is necessary to improve training methods and alleviate these problems.

For traditional image processing tasks, methods based on frequency can make weak signals more prominent and easier to be distinguished and processed. With the development of deep neural networks, CNN combined with frequency domain or wavelet domain is often used in image processing tasks. Some studies have gradually noticed

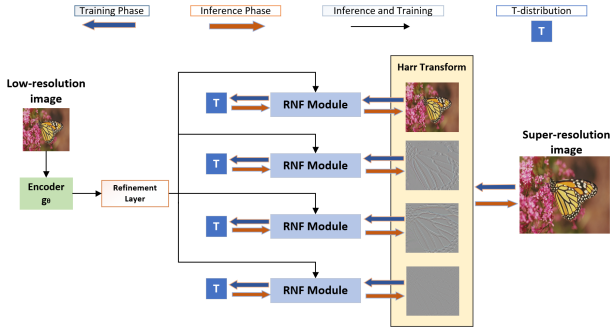


Figure 1: An overall framework of WDFSR. T represents the T-distribution.

that processing image information in the wavelet domain can achieve more satisfactory results in tasks, such as image classification [24], object detection[53], and instance segmentation[29]

The above techniques inspire us to combine Normalizing flow with the wavelet domain for enhancing super-resolution task. Our proposed network, namely WDFSR, transforms the local feature signals in high-resolution images into the wavelet domain, combining with Normalizing flow to better deal with image processing tasks. After wavelet transformation, the distribution of image data will become more regular and the details can be more highlighted than in the RGB domain, so that Normalizing flow can better learn the features of super-resolution image. Our contributions are as follows:

- We are the first to propose combining wavelet domain with Normalizing flow to deal with image super-resolution task. Our model is capable of generating both PSNR-oriented and Perception-oriented images. Our experiments show that WDFSR performs better than most existing CNN-based models and flow-based models.
- We propose a solution to combine QR decomposition layer and T-distribution, helping stabilize the network training process and enhance their generalization ability.
- We adopt the Refinement layer with an attention mechanism to help refine the extracted conditional features for performance improvement.
- Our model can handle other related image generation tasks, $e.g.$ Low-light enhancement applications, and obtain satisfactory results.

2 Related work

We review representative works from Image Super-Resolution, Normalizing Flow, and Wavelet-based Methods as they are closely related works.

2.1 Image Super-Resolution

There are different directions of exploration, such as single image SR, referenced-based image SR [35] and blind SR [27][12]. Our paper mainly focuses on the single image super-resolution (SISR). In recent years, many CNN-based super-resolution methods have been proposed for SISR. At first, some works[28][49][15] use either L1 or L2 as the loss function to train, yet output images were too smooth. Then, some methods [49] [48] [55] [46] use GAN to generate a real image stream and introduce perceptual loss to increase the texture of the image, making the generated images have better visual perception.

While some recent work has taken different approaches to super-resolution tasks, such as using implicit functions[22], transformers[36], and Multi-scale[23] networks, and other neural networks still employ GAN[38] or VAE[34] to produce good visuals, they can only produce high-resolution images from low-resolution images with deterministic one-to-one mapping, or some results are disappointing. Recently, researchers realize this problem, leading to methods, such as PULSE [40] (only for face), which can generate many different high-resolution images from one low-resolution image by using GAN. FxSR-PD [43] can generate different styles, including PSNR-Oriented and Perception-Oriented, by combining different train loss functions, yet the combination of losses is complex. Normalizing flow is also applied to image super-resolution task. However, such methods still need further research, including stability and effect improvement.

2.2 Normalizing Flow

Normalizing flow [8] [7] can learn mappings between arbitrary distributions and is a very powerful generative model. In recent years, Normalizing flow has not only been applied to speech generation and point cloud [25] tasks, but also has gradually been applied to image generation tasks and achieved good results, such as denoising and simulating noise [50][11], deblurring [27], SR [37] [26] and other low-level visual tasks. Glow [21] provides us with ideas on image generation tasks.

Normalizing flow is a reversible network model, meaning that it may suffer from many constraints that reduce its expressiveness, so researchers have offered many methods [16][45][4] to improve its performance. These methods mainly improve the model structure, or apply data preprocessing methods and other measures to improve model expression ability, stability, generalization. We also propose some measures to further increase the training stability of the model and enhance the generalization ability of the model to alleviate the OOD problem.

2.3 Wavelet-based Methods

In traditional image processing tasks, using frequency domain augmentation can bring some good results. Compared with Fourier transform and discrete cosine transform, wavelet transform considers both spatial domain information and frequency domain information. Recently wavelet-based methods have been explored in several computer vision tasks, including classification [29] [53] [41], face aging [33], network compression [9], super-resolution [31] [52], style transfer [54] and demoire [30].

Gal [10] proposes the SWAGAN which implements progressive generation in frequency domain. They verify that content generation in wavelet domain results in higher quality images with more realistic high frequency content, and frequency-aware methods also induce low-level visual quality improvements. Liu [31] proposes MWCNN for SR, which implements multi-level Wavelet methods. However, this method is less effective than existing methods because its network architecture may not be very expressive. Xiao [52] proposes a method for image rescaling in wavelet domain using a reversible network structure but not a flow-based model.

2.4 Attention mechanisms

Attention mechanism can be understood as having a computer vision system to quickly and effectively focus on the characteristics of key areas, which simulates human vision system. For humans, when facing with a complex scene, we can quickly focus on the key areas and process them. For deep learning, attention mechanism can make the model pay more attention to important features.

Attention mechanisms [18][42][51] have recently attracted extensive attention and have been applied in various fields, such as object classification, object detection and image restoration. Zhang [57] propose RCAN using residual channel attention mechanism for SR, which inspires us to apply the attention mechanism.

3 Our Method

3.1 Overview

As shown in Figure 1, our WDFSR network comprises Normalizing Flow with Refinement layer (RNF) module, Harr Transform, and T-distribution module. It is important to note that the three seemingly gray pictures from top to bottom represent the horizontal details, vertical details and diagonal details of the original image, respectively. Their details are significant, yet they are hard to see by naked eyes unless binarization is applied. Due to the special design of the RNF module and its reversibility, the training process is different from the general CNN-based model.

For training, we first transform a high-resolution image from the RGB domain into four different kinds of wavelet information, which are relatively regular distribution, by using Harr transform. Each RNF module branch will independently learn the mapping relationship between the T-distribution and the four different spectral information, such as diagonal information distribution, in the wavelet domain. To better fit the mapping relationship, during our training, low-resolution images are put into the encoder to obtain conditional features, feeding into the RNF module. We combine negative log-maximum likelihood with other training losses to optimize our model.

For generating a super-resolution image, different from training, in order to reversibly reconstruct the super-resolution image, we require to sample from the T-

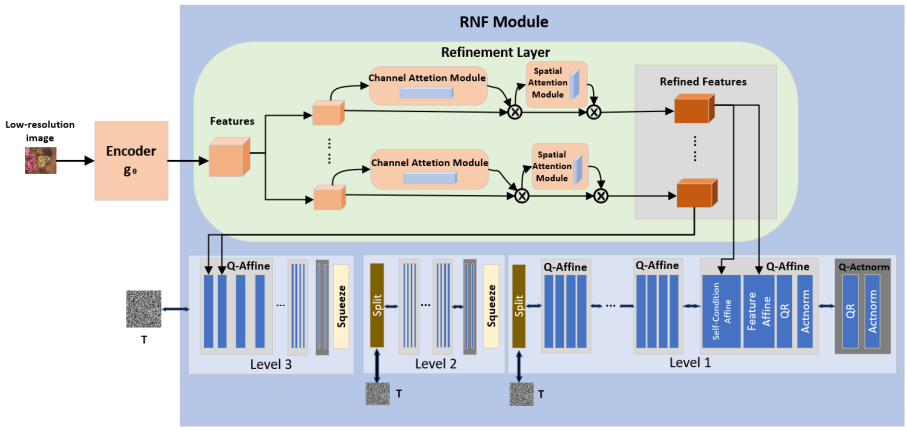


Figure 2: The structure of our RNF module for 4×super-resolution model, which is applied to all four RNF module branches of our WDFSR network. The structure of RNF modules for 8×super-resolution task and Low-light enhancement task are similar. Note that we have omitted the processing of wavelet transform in this figure.

distribution (Note that the sampled data of different branches of the T-distribution are also independent), and inject the conditional feature information of the low resolution image into the four trained RNF module branches. Subsequently, the four trained RNF modules will obtain four kinds of spectral information in the wavelet domain, which can be used to restore a high-quality super-resolution image through the Harr transform. By changing different sampling conditions (temperate τ), WDFSR can generate different styles of images.

We now introduce the technical details of the three main modules of our WDFSR network.

3.2 RNF Module

The RNF module is used to learn the exact mapping relationship between the fitting T-distribution and the four different information distributions obtained through wavelet transform. This module mainly contains a Normalizing flow module and a Refinement layer, where the Normalizing flow module is a multi-level architecture inspired by RealNVP [8]. The overall structure of the RNF module is depicted in Figure 2. Note that all four RNF module branches in our WDFSR network, as shown in Figure 1, use exactly the same structure.

Specifically, the architecture of Normalizing flow mainly contains L scales (levels). For each of the different levels of the architecture, it has a reversible same structure including one Squeeze layer, one split layer, a Q-Actnorm, and K Q-Affines. In addition, each Q-Affine contains an Actnorm layer, one QR layer, and two different conditional mapping layers, while a Q-Actnorm contains an Actnorm layer and one QR layer.

At the end of each level, the split layer divides half of the features by dimension to obey T-distribution for calculating the negative logarithmic maximum likelihood (instead of a Gaussian distribution), allowing half of the features to continue flowing in the architecture. It is worth noting that the first level architecture has no Squeeze layer, and the last level has no split layer. We will now elaborate the important details in the following subsections. Other module improvements are presented in Appendix A.

3.2.1 QR layer

In order to further strengthen the mapping ability of RNF modules to better fit the relationship between different distributions, we utilize the QR decomposition characteristics to build a reversible QR layer for exchanging information on channel dimensions. We have done some experiments and found that T-distribution and QR layer can better improve the generalization ability of the model.

Glow [21] also proposed 1×1 convolution and PLU,

which can exchange information in dimensions. In actual training, 1×1 convolution will cause training loss to fluctuate largely and may easily be detrimental to convergence. Although PLU is stable, it is not flexible, which may lead to damaging the expressiveness of the flow-based model. QR layer as proved by Hoogeboom [17] is more stable while maintaining flexibility. In a similar fashion to the PLU parametrization, we stabilize the decomposition by choosing $W = \mathbf{Q}(R + diag(s))$, where $\mathbf{Q}$ is orthogonal, R is strictly triangular, and elements in s are nonzero. Because $\mathbf{Q}$ is an orthogonal matrix, $\mathbf{Q}$ can be constructed from at most n Householder reflections through $\mathbf{Q} = \mathbf{Q}_1 * ... * \mathbf{Q}_n$ to ensure its flexibility.

$$\mathbf{Q}_i = \mathbf{I} - 2\frac{\mathbf{k}_i \mathbf{k}_i^T}{\mathbf{k}_i^T \mathbf{k}_i}, \tag{1}$$

where $\{\mathbf{k}_i\}_{i=1}^n$ are learnable parameters.

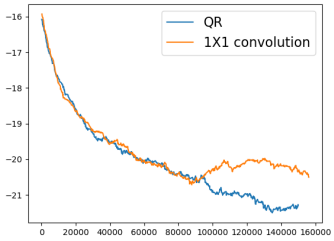


Figure 3: Visually comparison the effects of QR and 1x1 convolution on training loss.

Visual comparison of training loss and quantitative comparison of two layers is shown in Figure 3 and Table 6, respectively. We can see that using QR layer will make the training more stable. When the temperate τ of the variable sampled from the target distribution is 0, the network using QR layers achieves similar results as a network using 1×1 convolutions. However, when the sampled temperate τ is 0.8, using 1×1 convolution achieves poor results compared to the network using QR layers. This phenomenon shows that the QR layer can guarantee stable training while maintaining good performance compared to 1×1 convolution. Therefore, we provide a training suggestion that combining T-distribution and QR layer makes the flow-based model more stable and maintains a good performance.

3.2.2 Refinement layer

Our model is a Normalizing flow based on conditional features, i.e. when training and inferring, we are required to input low resolution images as conditional features to various Affine layers in the RNF module to enhance the mapping ability of the model. Therefore, we need a good encoder to extract image features accurately, so that the model can focus on important and useful information. We selected a part of RRDB [49] model as our encoder.

Although RRDB [49] network architecture as an encoder for SR task can extract features from images very well, the output dimension is large and some dimensions are not very significant for our model. Hence, we propose the Refinement layer to further focus on the features of channel dimension and spatial dimension to promote better mapping capability. Through experiments, we found that the Refinement layer can incorporate different attention mechanism modules[18][42] and they all improve performance. However, some of their performance improvement is not obvious or their solutions use more parameters or increasing computational complexity.

Consequently, we choose the relatively superior CAMB [51] as our attention mechanism module, which is a combination of channel attention mechanism and spatial attention mechanism, to produce better results while using fewer parameters.

Note that we do not insert the network into the encoder but as a separate small module. Meanwhile, we use independent attention modules in condition affine layers instead of sharing attention modules to learn for different scale layers. Because the features extracted from the pre-trained RRDB are actually relatively good, we fix the RRDB when training our model. We only optimize the independent Refinement layer, without needing to retrain a modified RRDB model. This approach can both reduce the GPU memory utilization for training and speeding up the training process. Also, our model accuracy is even higher than if we modified the RRDB model.

3.3 Harr Transform

There are many kinds of wavelet transforms such as Morlet, Mexican hat, Gaussian, etc. They can transform feature information in RGB domain into the spectral information in the wavelet domain. We use the simplest Harr

transform which has been proved to be simple and efficient in some previous work.

Due to the reversible network structure and special training method, we do not use the wavelet transform multiple times in our network, like some methods [30][31][10]. Similar to Xiao [52], we convert high-quality images to wavelet domain to learn from the beginning and we only use the wavelet transform once. We believe that the four different information distributions obtained after a wavelet transform can make the RNF module better learn the mapping relationship. We will discuss the role of the Harr transform in Section 5. What the wavelet transform does is shown as below.

$$A, H, V, D = Harr(X) \tag{2}$$

where X, A, H, V, D are high-resolution images, horizontal detail information, vertical detail information, diagonal detail information and areas of low-frequency information, respectively. Harr Transform is invertible, which means that $Harr^{-1}(A, H, V, D) = X$ and Harr Transform can restore the super-resolution picture very well. Their channel dimension is one-fourth of X and the length and width are one-half of X.

3.4 T-distribution

In theory, the flow-based model can learn a mapping from a very complex distribution to a very simple distribution ($e.g.$ Gaussian distribution). The model is optimized by minimizing the negative log maximum likelihood value as in Equation 7. However, Maximum Likelihood Estimation (MLE) is very sensitive to test set and train set that do not meet the model assumptions. Meanwhile, as shown in Figure 4 and as mentioned by Alexanderson [2], different distributions have some different features (Figure 4(a)(b)), which will influence the generalization ability of the model and training process. For abnormal data points, the corresponding Gaussian probability will be very low, which will cause problems such as log(0) resulting in a loss of null or huge loss volatility, which makes the training process unstable.

The general solution is to modify the learning rate or use gradient clipping. However, choosing an appropriate learning rate is very difficult. Furthermore, as proposed in [2], using gradient clipping may pull it to a different optimal solution, and the accuracy of the model may not be very good. Replacing the multivariate Gaussian distribution with the T-distribution, we can increase the generalization and training stability of the network without modifying the learning rate or using gradient clipping. For data that does not meet the model assumptions ($i.e.$ both ends of the curve as shown in Figure 4(c)), T-distribution is less affected and less punished by outliers than the Gaussian distribution. In general, our flow-based model is more stable and can generalize better than other flow models using T-distribution. We have depicted a generalization comparison of flow-based Super-Resolution models and ablation experiments about it in Section 5. The probability density function of T-distribution for computing loss in D dimensions is:

$$\begin{aligned} p_t(x; \mu, \Sigma, v) =& \Gamma\left(\frac{v+D}{2}\right)\left(\Gamma\left(\frac{v}{2}\right)\right)^{-1}|v\pi\Sigma|^{-\frac{1}{2}} \\ & \cdot\left(1+\frac{1}{v}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)^{-\frac{v+D}{2}}, \end{aligned} \tag{3}$$

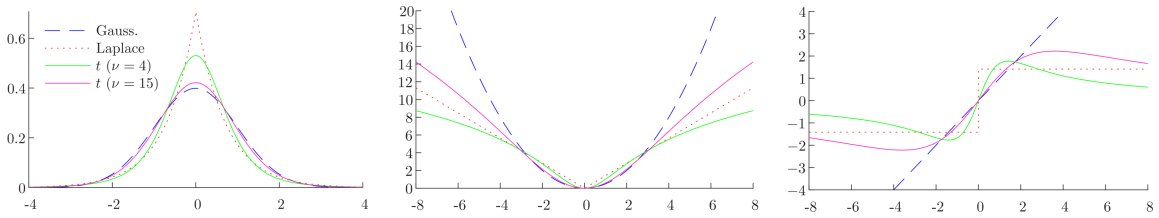
where the scalar $v > 0$ is called the degrees of freedom, and $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$. Meanwhile, μ, Σ and D represent the mean, covariance, and the number of channel dimensions of sample data x, respectively. When $v > 0$ tends to infinity, T-distribution becomes Normal distribution. We set it to 20 in our flow-based model.

3.5 Training Objectives

3.5.1 Preliminaries

Normalizing flow is an invertible model, which can learn the mapping between an observed distribution and simple distributions ($e.g.$ a multivariate Gaussian z): z = $f^{-1}(x)$, where f represents the flow-based model and x represents the observed distribution. Since the network needs to be able to compute the Jacobian matrix, each layer of it has to be carefully designed, making Jacobian matrix very easily be computed. In addition, the performance of single flow model is limited due to the reversible reason. In order to ensure good network performance, multi-layer flow stacking is required, leading to $f = f_1 * f_2 \cdots * f_N$.

$$x \overset{f_1}{\longleftrightarrow} \mathbf{h}_1 \overset{f_2}{\longleftrightarrow} \mathbf{h}_2 \longleftrightarrow \cdots \longleftrightarrow \mathbf{h}_{N-1} \overset{f_N}{\longleftrightarrow} \boldsymbol{z}, \tag{4}$$



(a) Probability density (b) Penalty functions (c) Influence functions

Figure 4: The analysis of different distributions.

where $\mathbf{h_i}$ represents the intermediate results produced by the flow model and their process is reversible. Here $\boldsymbol{z}$ can be any simple distribution. However, through extensive experiments, we found that fitting T-distribution, called Student-distribution, could get better results, which we will discuss later in Section 5. According to the change of variable formula and the chain rule, for a sample x, the log-likelihood can be calculated as:

$$\log p(x;\boldsymbol{\theta}) = \log p_{\boldsymbol{z}}(\boldsymbol{z}) + \sum_{i=1}^{N} \log \left| \det \frac{\partial f_i}{\partial f_{i-1}} \right| \tag{5}$$

In general, we train flow-based models by optimizing the negative log-maximum likelihood value $-\log p(x)$. For the conditional flow model, the initial formula will become the following.

$$p_{x|\mathbf{e}}(x \mid \mathbf{e}, \boldsymbol{\theta}) = p_{\mathbf{z}}\left(f_{\boldsymbol{\theta}}^{-1}(x;\mathbf{e})\right) \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial x}(x;\mathbf{e}) \right|, \tag{6}$$

where $\mathbf{e}$ represents the latent feature of low resolution picture and $\mathbf{z} = f_{\boldsymbol{\theta}}^{-1}(x;\mathbf{e})$.

Finally, we optimize the flow-based model by taking the negative log maximum likelihood. For our flow-based model WDFSR, the formula will look like this:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}; x, \mathbf{e}) &= -\log p_{x|\mathbf{e}}(x \mid \mathbf{e}, \boldsymbol{\theta}) \\ &= -\sum_{i=1}^{4} \log p_{\mathbf{z}}\left(f_{\boldsymbol{\theta}}^{-1}(\mathbf{y}_i;\mathbf{e})\right) - \sum_{i=1}^{4} \log \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{y}_i}(\mathbf{y}_i;\mathbf{e}) \right| \end{aligned} \tag{7}$$

where $\mathbf{z_i} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{y}_i;\mathbf{e})$ corresponds to the mapped T-distribution of the four branches and $\mathbf{y}_i$ represents the spectral information of x in wavelet domain.

3.5.2 Loss function

Flow-based models can be trained by optimizing only a single negative log-likelihood loss, like this,

$$\mathcal{L}_{nll} = -\sum_{i=1}^{4} \log p(\mathbf{y}_i \mid \mathbf{e}, \boldsymbol{\theta_i}). \tag{8}$$

where $\mathbf{y}_i$ represents the spectral information of x in wavelet domain.

Training with this single $\mathcal{L}_{nll}$ can make the network eventually convergent, but in the actual training process, we found that the network converges very slowly and may not reach the optimal value because of unsupervised reasons. Because our network is capable of generating both PSNR-oriented and Perception-oriented images, we can improve some aspects of performance again according to the loss design, such as the PSNR value. If we add L1 or L2 pixel losses to the original negative log-maximum -likelihood $\mathcal{L}_{nll}$, we can obtain a model that produces images with higher PSNR values. In our experiments, we have found that using L1 pixel loss is more stable and can achieve better results than using L2 pixel loss training. Therefore, the training loss function can become:

$$\mathcal{L}_{PSNR} = \lambda_1 \mathcal{L}_{nll} + \lambda_2 \mathcal{L}_{\text{pixel}}\left(\mathbf{x}, \mathbf{x}_{\tau=0}\right), \tag{9}$$

where $\mathbf{x}$ represents the ground-truth super-resolution image and $\mathbf{x}_{\tau=0}$ is the super-resolution image generated by the model by sampling the latent variable with temperate τ=0 from T-distribution.

Similarly, if we want to get a model, which can generate better Perception-oriented images with better visual quality, we add perceptual loss to its loss. Then, the for-

Table 1: Comparison with state-of-the-art SR methods on DIV2K test sets in 4× and 8× SR tasks, and Urban datasets in 4× SR task. The 1st and the 2nd best performances are highlighted in red and blue, respectively. ↑ (↓) denotes that, larger (smaller) values lead to better quality.

Datasets		DIV2K						Urban		
		4×			8×			4×		
Type	Methods	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
	Bicubic	26.81	0.772	0.412	23.83	0.632	0.584	21.80	0.662	0.473
PSNR-oritened	EDSR[28]	28.98	0.833	0.269	-	-	-	25.32	0.803	0.208
	DBPN[15]	29.18	0.838	0.266	-	-	-	24.86	0.789	0.223
	RRDB[49]	29.44	0.844	0.254	25.50	0.695	0.419	25.48	0.809	0.196
	RDN-LTE[22]	29.33	0.841	0.256	-	-	-	25.27	0.801	0.204
	ESRT[36]	28.85	0.830	0.285	-	-	-	24.46	0.776	0.238
Perception-oriented	ESRGAN[49]	26.63	0.764	0.124	22.18	0.583	0.277	22.78	0.721	0.123
	RankSRGAN[55]	26.55	0.750	0.128	-	-	-	23.05	0.721	0.135
	SFT-GAN[48]	26.50	0.757	0.133	-	-	-	22.74	0.710	0.134
	NatSR[46]	27.82	0.793	0.152	-	-	-	23.94	0.753	0.150
	SPSR[38]	26.70	0.761	0.109	-	-	-	23.24	0.737	0.119
PSNR-Perception	SRFlow,$\tau = 0$[37]	29.05	0.829	0.251	25.09	0.659	0.403	25.03	0.792	0.203
	SRFlow,$\tau = 0.9$[37]	27.09	0.756	0.120	23.04	0.573	0.272	23.65	0.732	0.133
	HCFlow+,$\tau = 0$[26]	29.25	0.83	0.212	-	-	-	25.03	0.780	0.210
	HCFlow++,$\tau = 0.9$[26]	26.61	0.743	0.111	-	-	-	23.03	0.711	0.124
	(Ours)WDFSR,$\tau = 0$	29.22	0.838	0.243	25.31	0.686	0.389	25.16	0.799	0.196
	(Ours)WDFSR,$\tau = 0.8$	28.36	0.809	0.134	24.76	0.647	0.380	24.43	0.769	0.136
	(Ours)WDFSR+,$\tau = 0$	29.39	0.846	0.254	25.52	0.697	0.419	25.36	0.813	0.199
	(Ours)WDFSR++,$\tau = 0.9$	27.72	0.789	0.110	24.13	0.642	0.268	24.19	0.769	0.117

mula will become:

$$\mathcal{L}_{Visual} = \lambda_1 \mathcal{L}_{nll} + \lambda_2 \mathcal{L}_{\text{pixel}}\left(\mathbf{x}, \mathbf{x}_{\tau=0}\right) + \lambda_3 \mathcal{L}_{\text{perceptual}}\left(\mathbf{x}, \mathbf{x}_{\tau=\tau_0}\right) \tag{10}$$

where $\mathbf{x}_{\tau=\tau_0}$ represents the image biased towards visual perception produced by sampling the latent variable with temperate τ=τ_0 from the T-distribution. We set τ_0 to 0.9, which can produce better Perception-oriented pictures.

4 Experiments

4.1 Settings

While our model can handle super-resolution tasks, it can also support other image processing tasks, such as low-light enhancement. We conduct extensive experiments on the general image SR dataset, as well as some experiments on the low-light enhancement dataset. For the SR task, our model has three combinations of model losses: $\mathcal{L}_{\text{nll}}$, $\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{pixel}}$ and $\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{pixel}} + \mathcal{L}_{\text{percep}}$. They are used for WDFSR, WDFSR+, and WDFSR++, respectively. However, we only use $\mathcal{L}_{\text{nll}}$ for low-light enhancement task. Because the main purpose of the low light enhancement task is to improve the overall brightness while maintaining good image quality, it is sufficient to use $\mathcal{L}_{\text{nll}}$ only.

For 4× super-resolution task, we set L and K to 3 and 7, respectively. Similarly, for 8× super-resolution, we set L and K to 4 and 6, respectively. For the 8× SR task, we need 3 times of down-sampling, i.e. L=4. But at the same time, K is reduced to 6 in order to improve training efficiency. As L becomes larger, the mapping ability of the model does not decrease with K.

We use the part of the pretrained RRDB network as an encoder to extract features, but fix it to not being updated. For most image super-resolution models, they use Flickr2K and DIV2K [1] as training sets and test sets to compare model performance. Therefore, for fair comparison, we also use these two datasets for training for 150k iterators and testing. In order to prove that we still perform well on other datasets, we also tested on the Urban [19] dataset.

We set the size of the crop block and the number of mini-batches to 160×160 and 12, respectively. β_1, β_2 in Adam optimizer are set to 0.9, 0.99. For WDFSR, our initial learning rate lr is 1×10^{-4} and is decayed by

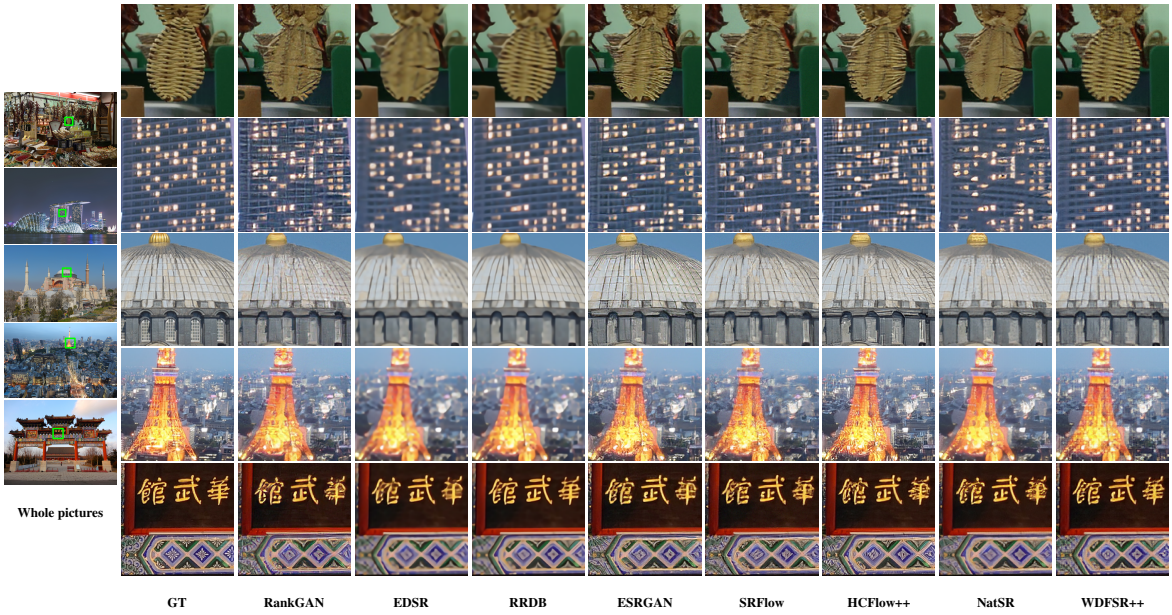


Figure 5: Visual results of general image SR (4×) on the DIV2K test set. The values of temperate τ of HCFlow++, SRFlow, and WDFSR++ are all 0.9.

half at [0.4, 0.6, 0.70, 0.75, 0.8, 0.85, 0.9, 0.95], respectively. On the basis of the pretrained WDFSR, we fine-tune WDFSR+ using $\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{pixel}}$ for about 30K iterations and decay by half at [0.3, 0.4, 0.60, 0.75, 0.8, 0.85, 0.9, 0.95]. We set the initial learning rate lr= 1×10^{-5}, $\lambda_1 = 3 \times 10^{-4}$ and λ_2=25, respectively. For WDFSR++ using $\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{pixel}} + \mathcal{L}_{\text{percep}}$, we also fine-tune for 20k iterators based on WDFSR and set $\lambda_1 = 3 \times 10^{-4}$, λ_2=15, λ_3=25 for 4× SR, respectively, but λ_2=25, λ_3=50 for 8× SR. All SR images are evaluated with PSNR, SSIM and Lpips in RGB space. LoL dataset [5] is often used for training and testing performance in Low-light enhancement tasks. Therefore, we also use it as our dataset for comparison in Low-light enhancement tasks.

4.2 Results on Image SR

For the DIV2K dataset, we test on 4× and 8× models and compare with CNN-based and flow-based models including EDSR, ESRGAN, RDN-LTE, ESRT, SPSR, SFT-GAN, NatSR, RRDB, RankSRGAN, HCFlow, and SR-Flow. These quantitative results are shown in Table 1 and some visual results are compared in Figure 5. There are three different types of models. One is the PSNR-based model that can only produce images with high PSNR values. One is the Perception-based model that can only produce more biased texture images. They are one-to-one deterministic mapping models that can generate high-resolution images from low-resolution images. The last one is an one-to-many mapping PSNR-Perception-based model that can generate both PSNR-oriented and Perception-oriented images.

WDFSR outperforms most PSNR-based models. Compared to the best results of the PSNR-based model, i.e. RRDB network, our model can achieve similar results to the PSNR value but our visual quality is better than theirs. Since RRDB is not a lightweight model, we decide to use only a small part of it as our encoder to enhance our training speed, yet the features extracted from our model is not as good as those from the complete RRDB model. Although the results from our model is slightly lower in PSNR than that of RRDB in a few cases, our results are

far superior to RRDB in terms of perception.

Table 2: Quantitative results comparison on the LOL dataset in terms of PSNR, SSIM, and LPIPS. ↑ (↓) denotes that, larger (smaller) values lead to better quality. We take an average of 5 tests.

methods	PSNR↑	SSIM↑	LPIPS↓
Zero-DCE[13]	14.86	0.56	0.335
LIME[14]	16.76	0.56	0.353
RetinexNet[6]	17.61	0.64	0.386
RUAS[32]	16.40	0.50	0.270
KinD[58]	20.87	0.83	0.159
KinD++[56]	21.80	0.83	0.164
(Ours)WDFSR$\tau = 0.65$	**22.01**	**0.84**	**0.156**

We also achieve good results compared with Perception-based models and performs better than other PSNR-Perception-based models. In the visual comparison, we found that our model could better reproduce the detailed textures more realistically. As shown in Table 1, our results are better than other flow-based models on the Urban dataset and also outperform most super-resolution methods, which shows that the model can obtain better results by learning Normalizing flow in the wavelet domain. There are more detailed discussions about generating different styles of an image by sampling different latent variate from target distribution in Appendix A.

4.3 Results on Low-light enhancement

Low-light enhancement task is to process images with insufficient lighting to improve visual quality. Our model is trained on LoL dataset and quantitatively and visually compare with some current methods [13][14][20][6][32][58][56]. As shown in Figure 6, our model achieves a good visual effect in Low-light enhancement application task. From Table 2, it can be analyzed that our model is the best in all quantitative comparisons, meaning that we have outperformed most low-light enhanced methods, proving that our model can handle such low-light enhanced image generation tasks. Due to space reasons, there will be more comparison of visual results and application of changing the light intensity of pictures by different sampling conditions in Appendix C.

5 Ablation Study

Before we get into the discussion, it is worth mentioning that we average the results of five tests for each model during comparisons of ablation experiments.

5.1 Wavelet domain vs RGB domain

Since Harr transform can generate 4 different information in the frequency domain, we design 4 branches that can correspond one-to-one. To prove that using the wavelet domain can indeed improve the effect, we compare the model in the wavelet domain with the model in the RGB domain. We directly use the Squeeze layer, similar to wavelet transform, to obtain 4 spectral information in the RGB domain. As shown in Table 3, in PSNR, SSIM, and LPIPS, the result of processing in the wavelet domain exceeds that in the RGB domain.

As shown in Figure 7, using the information in the RGB domain cannot give full play to the power generation capability of the flow model. There is little difference between the two generated results for images with simple textures ($e.g.$ Figure 7(c)). But for complex texture modules, the images they produce will have uneven color patches ($e.g.$ Figure 7(a)(b)(d)). We reason that Normalizing flow cannot adequately learn its distribution. However, four different signal information in pictures in the wavelet domain can be highlighted, Normalizing flow can better learn relatively regular distributions. It is proved that in some cases, processing task in the wavelet domain is better than directly processing them in the RGB domain.

Table 3: Quantitative results comparison between Wavelet domain and RGB domain in DIV2K test set (4×).

WDFSRτ	domain	PSNR↑	SSIM↑	LPIPS↓
0	Wavelet	**29.22**	**0.838**	0.243
0.8	Wavelet	28.36	0.809	**0.134**
0	RGB	29.21	0.838	0.257
0.8	RGB	28.14	0.801	0.206

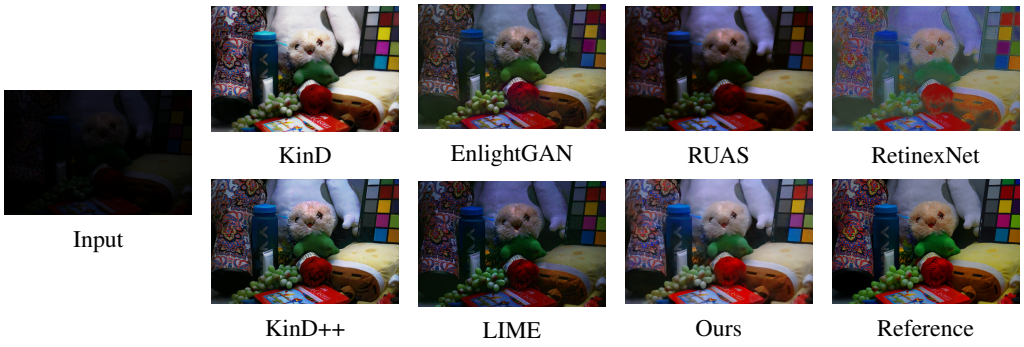


Figure 6: Visual results comparison with state-of-the-art low-light image enhancement methods on LOL dataset. The normally exposed image generated by our method has less noise and artifact, and better colorfulness.

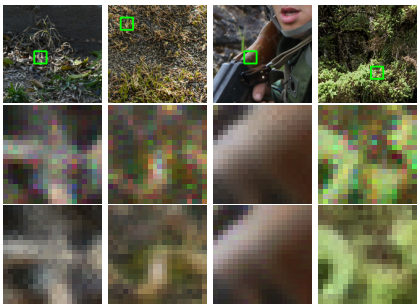


Figure 7: Visual results comparison of wavelet domain and RGB domain with τ=0.8. The first, second, and third rows are the original image, the important detail of part of images based on RGB domain and those based on wavelet domain, respectively.

5.2 Refinement layer

The Refinement layer can automatically learn to strengthen useful features and weaken useless features to improve the performance of our method. We take CBAM as the main attention module in the Refinement layer, which can improve the performance compared to not using CBAM. The results are shown in Table 4.

It is obvious that our model works better when the Refinement layer is included. Specifically, when we sample variate with temperate $\tau = 0$ from T-distribution, we obtain better results in terms of PSNR and SSIM than using our model without the Refinement layer. Similarly, when the temperate $\tau = 0.8$, our results obtain a better LPIPS than using our model without the Refinement layer.

Table 4: Quantitative comparison between with Refinement layer and without Refinement layer in DIV2K test set (4×).

WDFSR,τ	Refinement layer	PSNR↑	SSIM↑	LPIPS↓
0	✓	**29.22**	**0.838**	0.243
0.8	✓	28.36	0.809	**0.134**
0		29.10	0.819	0.248
0.8		28.24	0.789	0.144

5.3 T-distribution vs Gaussian distribution

We demonstrate through ablation experiments that when the model uses both T-distribution and QR layers, the re-

Table 5: Quantitative results comparison of test generalization of flow models on other datasets (4×). Evaluation indicators are PSNR, SSIM, and LPIPS in order. Red represents the best result.

-	HCFlow+,$\tau = 0$	SRFlow,$\tau = 0$	WDFSR+,$\tau = 0$(ours)
BSDS100	26.39/0.718/0.369	26.23/0.734/0.363	26.50/0.743/0.358
Set5	30.56/0.871/0.175	30.20/0.874/0.174	30.59/0.883/0.169
Set14	26.99/0.751/0.281	26.61/0.762/0.272	26.85/0.773/0.273
T91	29.36/0.824/0.210	29.11/0.843/0.204	29.35/0.845/0.194
General100	30.03/0.850/0.173	29.80/0.863/0.173	30.16/0.867/0.166

sults become better and the training process is more stable. As shown in Figure 8, when the Gaussian distribution is used as the target distribution without gradient clipping, the training loss fluctuates greatly and gets NULL values leading to stop training at about 58k iterators. To avoid this, we use gradient clipping.

Although their training loss is relatively stable after clipping whether using QR layer or 1×1 convolution, they cannot reach the training loss value of the model with T-distribution, which indicates that the model does not reach the optimal point. As shown in Table 6, the model with T-distribution and QR layer can achieve the best results at τ=0 and τ=0.8. Although the model using the Gaussian distribution performs slightly worse than the model combining T distribution at τ=0, the model using the Gaussian distribution performs extremely poorly at τ=0.8, which shows that the model using the Gaussian distribution is unstable and has poor generalization.

Despite Gaussian distribution used as the target distribution is unstable, as mentioned earlier in section 3, relatively good results can be obtained using a QR layer that is more stable than a model using a 1×1 convolution. As shown in Table 5, we quantitatively compare different flow-based models' capabilities in Set14 [19], Set5 [3], BSD100 [39], General100 [47], and T91 [47] datasets, respectively. Our model with T-distribution generalizes better than other flow-based models with Gaussian distribution.

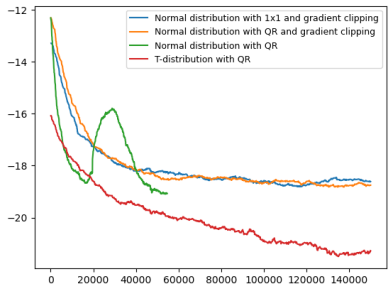


Figure 8: Visual results comparison of Gaussian distribution and T-distribution about training loss stability.

6 Conclusion

Combining Normalizing flow with wavelet domain is a promising solution to improve image generation tasks. We propose a super-resolution model based on Normalizing flow with the wavelet domain, which can generate different styles according to different sampling conditions and achieve better results than most previous networks. We are the first to propose combining the flow model with the wavelet domain for image generation tasks ($e.g.$ image super-resolution and image Low-light enhancement). In addition, we propose a training proposal to help some flow-based models stabilize training and increase generalization. We also propose a Refinement layer to help to refine features to aid training. Compared with the PSNR-based models and the Perception-based models, we can produce a variety of styles and better quality images than them. Compared with other flow-based super-resolution models, our model training is more stable, with stronger performance and better generalization. Meanwhile, we also demonstrate through experiments that our model can also achieve satisfactory performance compared to state-

Table 6: Quantitative results comparison between T-distribution and Gaussian distribution in DIV2K test set (4×). N-distribution represents the Normal Distribution and uses the gradient clipping to avoid the NULL problem.

WDFSR,τ	T-distribution	N-distribution	1×1 convolution	QR	PSNR↑	SSIM↑	LPIPS↓
0	✓			✓	**29.22**	**0.838**	0.243
0.8	✓			✓	28.36	0.809	**0.134**
0	✓		✓		29.21	0.836	0.246
0.8	✓		✓		27.92	0.781	0.149
0		✓		✓	29.14	0.836	0.251
0.8		✓		✓	20.93	0.548	0.410
0		✓	✓		29.13	0.836	0.258
0.8		✓	✓		17.64	0.435	0.539

of-the-art Low-light enhancement models.

The disadvantage of our method is that WDFSR is not lightweight compared to other models due to its four independent RNF modules. However, this design can indeed improve the expression ability of the model, yet we will still try to address the insufficient lightweight problem. In future work, we will explore applications of our network to other generative tasks such as rain removal, noise removal, and moiré removal.

A Some details about architectural design

Here we only introduce some important small modules, ignoring some similar modules in the flow-based model, such as Affine layer [7], Squeeze [21], Actnorm [37], etc.

A.1 Split layer

In our model, the split layer mainly deals with the channel dimension of features, allowing half of the feature dimensions to continue, allowing the model to learn and making the other half of the feature dimensions obey T-distribution, which cannot only reduce the training time but also increase the model performance to a certain extent.

$$X_{half}, M = SPLIT(X), M \sim T(\nu) \tag{11}$$

where X_{half} represents the features left behind and M represents the features that the other half needs to obey a T-distribution with degrees of freedom ν.

A.2 Condition affine layers

Through experiments, we found that the attention mechanism can improve the network ability very well. Therefore our condition layer is similar to SRFlow but finetunes the obtained features by taking advantage of the Refinement layer.

$$\begin{cases} h_A^n, h_B^n = S(h_n) \\ h_A^{n+1} = h_A^n \\ h_B^{n+1} = exp(Refined(f_{\theta,s}(h_A^n))) \cdot h_B^n \\ \qquad + Refined(f_{\theta,b}(h_A^n)) \\ h_{n+1} = Concat(h_A^{n+1}, h_B^{n+1}) \end{cases}$$

S represents a method for dividing the feature into even halves according to the channel dimension. $Refined$ stands for the Refinement layer and f represents a simple neural network. $Concat$ represents combining two features by channel dimension. h^{n+1} can be obtained through the same flow module from h^n described in Section 3.

B Different styles of super-resolution images

Since our model is flow-based, it can accurately learn from complex distributions (image data) to simple distributions (T-distribution). When we want to get super-resolution images, we need to randomly sample variate with temperate τ from T-distribution, which means that

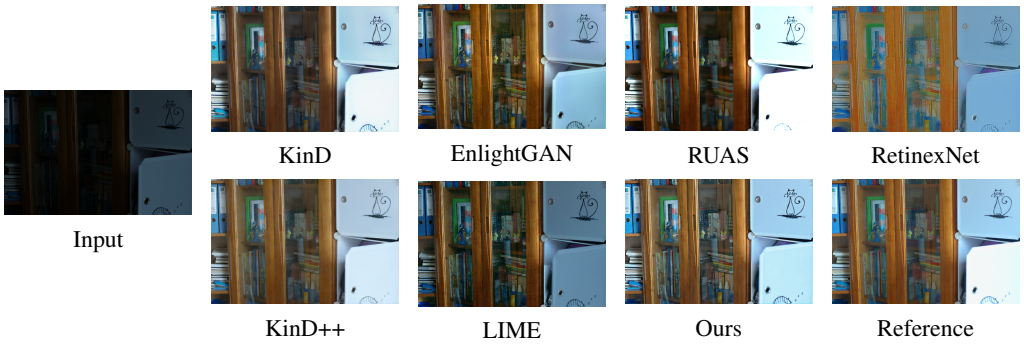


Figure 9: Visual results comparison with state-of-the-art low-light image enhancement methods on LOL dataset. The normally exposed image generated by our method has less noise and artifact, and better colorfulness.

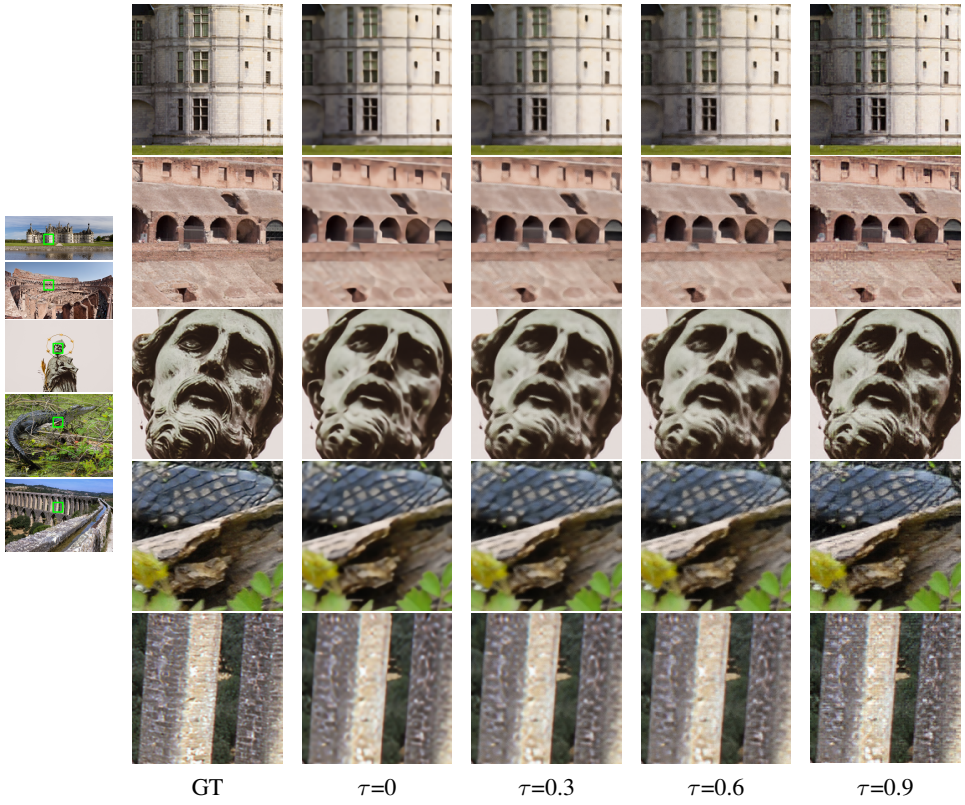


Figure 10: Changes in the result of WDFSR 4$\times$ SR by sampling latent variate t with different Temperate on DIV2K validation datasets.

our model has given one-to-many mappings from low-resolution images to super-resolution images. Although T-distribution is our target, in practice, we find that sampling from the normal distribution is slightly more effective than sampling from the T-distribution.

As shown in Figure 10, we can sample latent variables with different temperate τ called standard deviation to obtain different styles of images. When the standard deviation τ is close to 0, the image tends to be PSNR-oriented, having a similar blurring effect. When the standard deviation τ is close to 1, the image tends to be Perception-oriented in that the texture is clearer and the edges are sharper.

C Low-light Enhancement

For the training setting of our model, we set L, K, and batch size to 3, 8, and 10, respectively. We train for 100k iterators.Since the resolutions of our low-light pictures and high-light pictures are consistent, we modify the up-sampling module in the encoder to a down-sampling module, and other most settings are consistent with the Super-Resolution task. Here we show more visual results with other augmented models. Meanwhile, our model can modify the light intensity of images by modifying latent variables Z by sampling from T-distribution: Z=Z+v, where v is in the range of [-0.2,-0.1,0,0.1,0.2].

As shown in Figure 9, our model can produce better enhancement effects than other models in visual perception. Also, as shown in Figure 11, we demonstrate applying our model to modify the light intensity of three different images. It is obvious that our model can gradually increase image brightness by reducing the v value, while maintaining image visual quality.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] S. Alexanderson and G. E. Henter. Robust model training and generalisation with Studentising flows. In *Proceedings of the ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, volume 2 of *INNF+'20*, pages 25:1–25:9, 2020.

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012.

[4] J. Chen, C. Lu, B. Chenli, J. Zhu, and T. Tian. Vflow: More expressive generative flows with variational data augmentation. In *International Conference on Machine Learning*, 2020.

[5] W. Y. J. L. Chen Wei, Wenjing Wang. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018.

[6] W. Y. J. L. Chen Wei, Wenjing Wang. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*. British Machine Vision Association, 2018.

[7] L. Dinh, D. Krueger, and Y. Bengio. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[8] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[9] S. F. dos Santos and J. Almeida. Less is more: Accelerating faster neural networks straight from jpeg. In *Iberoamerican Congress on Pattern Recognition*, pages 237–247. Springer, 2021.

[10] R. Gal, D. C. Hochberg, A. Bermano, and D. Cohen-Or. Swagan: A style-based wavelet-driven generative model. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[11] A. Grover, C. Chute, R. Shu, Z. Cao, and S. Ermon. Alignflow: Cycle consistent learning from multiple domains via normalizing flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4028–4035, 2020.

[12] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1780–1789, 2020.

[14] X. Guo, L. Yu, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, PP(99):1–1, 2016.

Figure 11: Pictures of different light intensities by sampling and modifying latent variable from T-distribution.

[15] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[16] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019.

[17] E. Hoogeboom, R. Van Den Berg, and M. Welling. Emerging convolutions for generative normalizing flows. In *International Conference on Machine Learning*, pages 2771–2780. PMLR, 2019.

[18] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[19] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[20] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021.

[21] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[22] J. Lee and K. H. Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1929–1938, 2022.

[23] J. Li, F. Fang, J. Li, K. Mei, and G. Zhang. Mdcn: Multi-scale dense cross network for image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2547–2561, 2020.

[24] Q. Li, L. Shen, S. Guo, and Z. Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7245–7254, 2020.

[25] X. Li, H. He, X. Li, D. Li, G. Cheng, J. Shi, L. Weng, Y. Tong, and Z. Lin. Pointflow: Flowing semantics through points for aerial image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4217–4226, 2021.

[26] J. Liang, A. Lugmayr, K. Zhang, M. Danelljan, L. Van Gool, and R. Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4076–4085, 2021.

[27] J. Liang, K. Zhang, S. Gu, L. Van Gool, and R. Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10601–10610, 2021.

[28] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[29] Z. Lin, Y. Gao, and J. Sang. Investigating and explaining the frequency bias in image classification. *arXiv preprint arXiv:2205.03154*, 2022.

[30] L. Liu, J. Liu, S. Yuan, G. Slabaugh, A. Leonardis, W. Zhou, and Q. Tian. Wavelet-based dual-branch network for image demoiréing. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020.

[31] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018.

[32] R. Liu, L. Ma, J. Zhang, X. Fan, and Z. Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10561–10570, 2021.

[33] Y. Liu, Q. Li, and Z. Sun. Attribute-aware face aging with wavelet-based generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11877–11886, 2019.

[34] Z.-S. Liu, W.-C. Siu, and Y.-L. Chan. Photo-realistic image super-resolution via variational autoencoders. *IEEE Transactions on Circuits and Systems for video Technology*, 31(4):1351–1365, 2020.

[35] L. Lu, W. Li, X. Tao, J. Lu, and J. Jia. Masa-sr: Matching acceleration and spatial adaptation for reference-based image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6368–6377, 2021.

[36] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 457–466, 2022.

[37] A. Lugmayr, M. Danelljan, L. V. Gool, and R. Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *European conference on computer vision*, pages 715–732. Springer, 2020.

[38] C. Ma, Y. Rao, J. Lu, and J. Zhou. Structure-preserving image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[39] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[40] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

[41] E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *IEEE International Conference on Computer Vision*, 2017.

[42] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

[43] S. H. Park, Y. S. Moon, and N. I. Cho. Flexible style image super-resolution using conditional objective. *IEEE Access*, 10:9774–9792, 2022.

[44] W. Ping, K. Peng, K. Zhao, and Z. Song. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, pages 7706–7716. PMLR, 2020.

[45] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[46] J. W. Soh, G. Y. Park, J. Jo, and N. I. Cho. Natural and realistic single image super-resolution with explicit natural manifold discrimination. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8122–8131, 2019.

[47] X. Wang, L. Xie, K. Yu, K. C. Chan, C. C. Loy, and C. Dong. BasicSR: Open source image and video restoration toolbox. `https://github.com/XPixelGroup/BasicSR`, 2022.

[48] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[49] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[50] V. Wolf, A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 94–103, 2021.

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[52] M. Xiao, S. Zheng, C. Liu, Y. Wang, D. He, G. Ke, J. Bian, Z. Lin, and T.-Y. Liu. Invertible image rescaling. In *European Conference on Computer Vision*, pages 126–144. Springer, 2020.

[53] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1740–1749, 2020.

[54] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision, pages 9036–9045, 2019.

[55] W. Zhang, Y. Liu, C. Dong, and Y. Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3096–3105, 2019.

[56] Y. Zhang, X. Guo, J. Ma, W. Liu, and J. Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129(2), 2021.

[57] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[58] Y. Zhang, J. Zhang, and X. Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 1632–1640, New York, NY, USA, 2019. ACM.