Technical Report

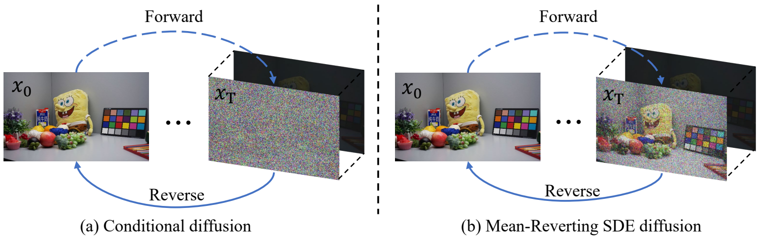
Fei Wan[1,2], Bingxin Xu[1,2*], Jian Cheng[1,2], Hongzhe Liu[1,2], Cheng Xu[1,2], Yuxiang Zou[1,2], Weiguo Pan[1,2], Songyin Dai[1,2]

[1*]Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, 100101, China.
[2]College of Robotics, Beijing Union University, Beijing, 100101, China.

*Corresponding author(s). E-mail(s): xubingxin@buu.edu.cn;
Contributing authors: 20221083510904@buu.edu.cn;

1 Motivation

Diffusion models are increasingly applied in low-light image enhancement tasks due to their exceptional capability to model data distributions, but an inherent drawback of diffusion models in image restoration tasks is that starting the reverse process from pure Gaussian noise can lead to artifacts [1, 2]. Therefore, as illustrated in Fig. 1, we adopt the Mean-Reverting Stochastic Differential Equation (SDE) [3] as the base diffusion framework, directly implementing the mapping from low-quality to high quality images.



(a) Conditional diffusion

(b) Mean-Reverting SDE diffusion

Fig. 1 (a) Conditional diffusion (b) Mean-Reverting SDE diffusion

The fundamental idea of diffusion models is to gradually corrupt images by injecting noise, and then learn how to progressively remove this noise to reconstruct the original image. U-Net plays a crucial role in this denoising process. It is trained to predict the noise injected at each step, thereby methodically eliminating the noise and restoring the image. The U-Net used in diffusion models typically consists of residual blocks, upsampling and downsampling operations, and attention mechanisms. While the stacking of multiple residual blocks is beneficial for feature extraction, it increases the computational load, and the extensive convolutional operations are not friendly to low pixel values in low-light images.

Our motivation is to reduce multiplication operations in U-Net, protect low pixel values, and lighten the computational load. The simplified U-Net designed in this paper, as illustrated in Fig. 2(a), is only constructed from the feature extraction module SimpleGate [4] and Parameter-free attention [5] (SimPF) block, and includes upsampling and downsampling operations, making it suitable for both processing low-light images and reducing the resource consumption of the diffusion model for faster sampling.

The code is available at https://github.com/MrWan001/SFDiff.

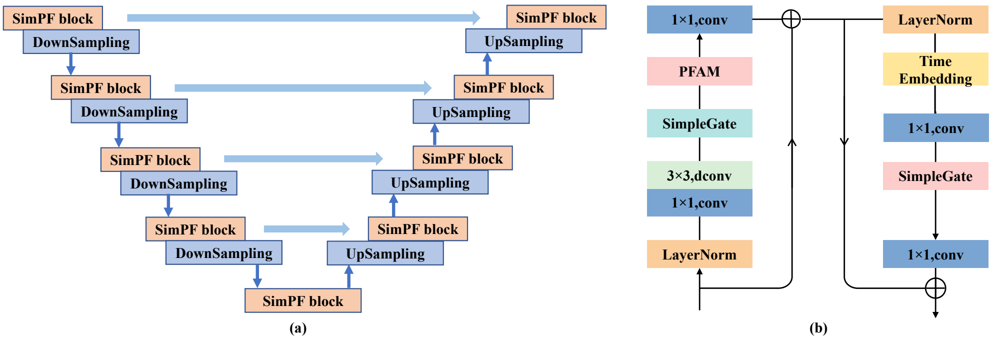


Fig. 2 (a) U-Net with SimPF block. It is composed of SimPF blocks, upsampling and downsampling operations, along with skip connections; (b) SimPF block. It retains necessary convolution and normalization layers, incorporates SimpleGate and PFAM to minimize multiplication operations, and utilizes Time Embedding to align with diffusion models.

2 Network Architecture

As shown in Figure 2(b), we designed the SimPF block with the idea of retaining the necessary convolution and normalisation layers and using less computationally intensive components to reduce multiplication operations across feature maps. We use 1×1 convolutions and 3×3 depth-wise separable convolutions for feature extraction, both convolution types have been applied and proven effective in a variety of image restoration tasks. Specifically, the feature map first undergoes a 1×1 convolution to expand the number of channels while preserving spatial information. Subsequently, a 3×3 depth-wise separable convolution is employed to encode features from spatially adjacent pixel positions, facilitating the learning of local image structures.

Since the activation function requires multiple multiplication operations, we use SimpleGate to replace complex nonlinear activation functions. SimpleGate can achieve the effect of nonlinear mapping through a single multiplication operation, which is particularly beneficial for preserving information in low pixel values, as complex functions like the cubic operations required in the GELU activation function can be detrimental to such information. The computation of SimpleGate is illustrated in Equation (1):

$$\text{SimpleGate}(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X} \odot \boldsymbol{Y} \tag{1}$$

$\boldsymbol{X}$ and $\boldsymbol{Y}$ represent the division of a feature map with channels C, height H, and width W along the channel dimension into two parts of $(\frac{C}{2}, H, W)$. The essence of this multiplication operation is a type of nonlinear mapping that can substitute for an activation function.

After the feature matrix has been given weights through Parameter-Free Attention Mechanism (PFAM), a 1×1 convolution is used to aggregate pixel-level cross-channel context information. The subsequent two 1×1 convolutions serve to facilitate interaction and combination among features across different channels, creating more complex and effective feature representations. In order to apply to the diffusion model, we have incorporated a time embedding block, which takes the current diffusion time step t as input and encodes t into the feature matrix, enabling the model to perceive noise at different time steps t. Overall, the design of SimPF block, while minimizing multiplication operations, maintains robust feature extraction capabilities.

3 Training Strategy

Our method is implemented using the PyTorch framework. The diffusion time step T is established at 100. A cosine scheduling scheme is utilized for noise scheduling. The optimization is carried out using the LION optimizer. The batch size is set to 6. The initial learning rate is set at 4×10^{-5}, and the Cosine Annealing strategy is employed for learning rate scheduling. The model is trained on a single NVIDIA GeForce RTX 3090 GPU and converged after 300,000 iterations.

During the training phase, we first attempted to crop or randomly crop the center of the training set to 256 x 256, but did not achieve good results. Finally, we resized the training set to 256 x 256 and achieved good results. During testing, due to the large size of 6720 x 4480, which exceeded the maximum range that the model could handle, we first attempted to crop the image into 2240 x 2240 and merge it, but the effect was not good. Finally, we resized the image to 480 x 320, and after using model enhancement, we resized it to 6720 x 4480, achieving good results. In addition, due to the unknown GT, we used the lpips metric to preliminarily evaluate the enhancement results of the model. We found that our proposed SimPF block performed better on the three images 1162, 496, 735, while the model trained on the original NAF block performed better on the other test images. Therefore, we combined the results of the two models to obtain the final version for submission.

References

[1] Yin, Y., Xu, D., Tan, C., Liu, P., Zhao, Y., Wei, Y.: Cle diffusion: Controllable light enhancement diffusion model. In: Proceedings of the 31st ACM International

Conference on Multimedia, pp. 8145–8156 (2023)

[2] Yang, Z., Liu, B., Xxiong, Y., Yi, L., Wu, G., Tang, X., Liu, Z., Zhou, J., Zhang, X.: Docdiff: Document enhancement via residual diffusion models. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 2795–2806 (2023)

[3] Luo, Z., Gustafsson, F.K., Zhao, Z., Sjölund, J., Schön, T.B.: Image restoration with mean-reverting stochastic differential equations. In: Proceedings of the 40th International Conference on Machine Learning, pp. 23045–23066 (2023)

[4] Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: European Conference on Computer Vision, pp. 17–33 (2022). Springer

[5] Yang, L., Zhang, R.-Y., Li, L., Xie, X.: Simam: A simple, parameter-free attention module for convolutional neural networks. In: International Conference on Machine Learning, pp. 11863–11874 (2021). PMLR