arXiv:2311.14900v4 [cs.CV] 24 Oct 2024

Resfusion: Denoising Diffusion Probabilistic Models for Image Restoration Based on Prior Residual Noise

Zhenning Shi [1] **Haoshuai Zheng**[1] **Chen Xu**[1] **Changsheng Dong**[1] **Bin Pan**[2]
Xueshuo Xie[3] **Along He**[1]* **Tao Li**[1,3]* **Huazhu Fu**[4]
[1]College of Computer Science, Nankai University
[2]School of Statistics and Data Science, Nankai University
[3]Haihe Lab of ITAI
[4]Institute of High Performance Computing, A*STAR

Abstract

Recently, research on denoising diffusion models has expanded its application to the field of image restoration. Traditional diffusion-based image restoration methods utilize degraded images as conditional input to effectively guide the reverse generation process, without modifying the original denoising diffusion process. However, since the degraded images already include low-frequency information, starting from Gaussian white noise will result in increased sampling steps. We propose Resfusion, a general framework that incorporates the residual term into the diffusion forward process, starting the reverse process directly from the noisy degraded images. The form of our inference process is consistent with the DDPM. We introduced a weighted residual noise, named **resnoise**, as the prediction target and explicitly provide the quantitative relationship between the residual term and the noise term in resnoise. By leveraging a smooth equivalence transformation, Resfusion determine the optimal acceleration step and maintains the integrity of existing noise schedules, unifying the training and inference processes. The experimental results demonstrate that Resfusion exhibits competitive performance on ISTD dataset, LOL dataset and Raindrop dataset with only **five** sampling steps. Furthermore, Resfusion can be easily applied to image generation and emerges with strong versatility. Our code and model are available at https://github.com/nkicsl/Resfusion.

1 Introduction

Denoising diffusion models [1, 2] have emerged as powerful and effective conditional generative models, demonstrating remarkable success in synthesizing high-fidelity data for image generation. Saharia et al. [3] proved that these generative processes can be applied to image restoration by feeding degraded images as conditional input into the score network. SNIPS [4] combines annealed Langevin dynamics and Newton's method to arrive at a posterior sampling algorithm, exploring the generative diffusion processes to solve the general linear inverse problems. Based on these, many diffusion-based models were adapted for downstream image restoration tasks [5, 6, 7, 8, 9, 10, 11]. For traditional diffusion-based models, the reverse process begins with Gaussian white noise, considering only the degraded images as the conditional input. This results in an increased number of sampling steps. Image restoration tasks often focus on restoring and editing specific high-frequency details while preserving crucial low-frequency information, such as the image structure. The degraded images used as conditional input inherently contain the low-frequency information. Therefore, initiating the reverse process from Gaussian white noise for image restoration tasks appears unnecessary and inefficient.

*Corresponding authors: litao@nankai.edu.cn, healong2020@163.com

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

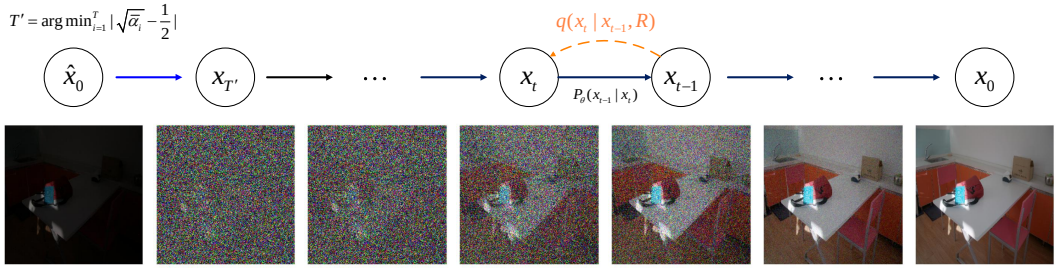

Figure 1: The proposed Resfusion is a general framework for image restoration and can be easily expand to image generation (setting $\hat{x}_0 = 0$). We introduce the residual term ($R = \hat{x}_0 - x_0$) into the forward process, redefine $q(x_t|x_{t-1})$ to $q(x_t|x_{t-1}, R)$ (as shown by the *orange* arrow), and name this diffusion process as resnoise diffusion. Through employing a novel technique called "smooth equivalence transformation", we can directly use the degraded image $\hat{x}_0$ to obtain $x_{T'}$ (as shown by the *blue* arrow). We bridge the gap between the input image and ground truth, unifying the training and inference processes.

Consequently, some works have proposed to generate clean images directly from degraded images or noisy degraded images. InDI [12] restores clean images through the reverse process of direct iteration to degraded images; DDRM [13] reformulate the image restoration tasks as inverse problems when the mapping between clean and degraded images is available; IR-SDE [14] directly models the image degradation process using mean-reverting SDE (Stochastic Differential Equations); I^2SB [15] constructs a Schrödinger bridge between clean and degraded data distributions; Resshift [16] shifts the residual term from degraded low-resolution images to high-resolution images, performing the recovery in the latent space. Liu et al. [17] introduced the Residual Denoising Diffusion Models (RDDM), generalizing the diffusion process of InDI and I^2SB. RDDM points out that co-learning the residual term and the noise term can effectively improve the model performance. However, RDDM has some limitations. Firstly, RDDM predicts the residual term and the noise term separately, without explicitly specifying their quantitative relationship. Secondly, due to its forward process adopting an accumulation strategy for the residual term and the noise term, its forward and reverse processes are inconsistent with the DDPM [1], which results in poor generalization and interpretability. Thirdly, RDDM requires the design of a complex noise schedule, as utilizing existing noise schedules would result in performance loss.

To solve the problems mentioned above, we propose **Resfusion**, a general framework for image restoration and can be easily expand to image generation. By introducing the residual term into the diffusion forward process, we bridge the gap between the input image and the ground truth, starting the reverse process directly from the noisy degraded images. We calculate the quantitative relationship between the residual term and the noise term, naming their weighted sum as the resnoise. Through the smooth equivalence transformation, we determine the optimal acceleration step and unify the training and inference processes. As a versatile methodology for image restoration, Resfusion does not require any physical prior knowledge. Resfusion allows for the direct use of existing noise schedules, and the image restoration process can be completed in just five sampling steps.

Our contributions can be summarized as follows:

- First, by introducing the residual term into the diffusion forward process, our **resnoise-diffusion** process starts the reverse process directly from the noisy degraded images, closing the gap between the degraded input and the ground truth.
- Second, we explicitly provide the quantitative relationship between the residual term and the noise term in the loss function, and name the weighted residual noise as **resnoise**. Through transforming the learning of the noise term into the resnoise term, the form of our reverse inference process is consistent with the DDPM.
- Third, through the **smooth equivalence transformation** in resnoise-diffusion process, we determine the optimal acceleration step and unify the training and inference processes. The optimal acceleration step is non-trivial where the posterior probability distribution is

equivalent to the prior probability distribution at this step. Moreover, we can directly use the existing noise schedule instead of redesigning the noise schedule.

2 Methodology

2.1 Learning the resnoise

First, we denote the input degraded image and the ground truth as $\hat{x}_0$ and x_0. In order to extend the diffusion process of Denoising Diffusion Probabilistic Models (DDPM) [1] to image restoration, we define the residual term as Eq. (1).

$$R = \hat{x}_0 - x_0 \tag{1}$$

Then the forward process can be defined as Eq. (2) and Eq. (3). We provide a detailed explanation for why we introduce the residual term R to Eq. (3) in this way in Sec. 2.2 and Figure 2. Consistent with the definition in DDPM, we use the notation $\beta_t = 1 - \alpha_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

$$q(x_{1:T}|x_0, R) = \prod_{t=1}^{T} q(x_t|x_{t-1}, R) \tag{2}$$

$$q(x_t|x_{t-1}, R) = N(x_t; \sqrt{\alpha_t}x_{t-1} + (1 - \sqrt{\alpha_t})R, (1 - \alpha_t)I) \tag{3}$$

Then the redefined forward process can be formalized as Eq. (4), where ϵ represents the Gaussian white noise.

$$x_t = \sqrt{\alpha_t}x_{t-1} + (1 - \sqrt{\alpha_t})R + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim N(0, I) \tag{4}$$

According to Eq. (4), x_t can be reparameterized as Eq. (5).

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + (1 - \sqrt{\overline{\alpha}_t})R + \sqrt{1 - \overline{\alpha}_t}\epsilon, \quad \epsilon \sim N(0, I) \tag{5}$$

We can easily incorporate this forward process into the vanilla DDPM. We introduce a residual noise, named as **resnoise** (symbolized as $res\epsilon$), to describe the gap between the current estimate x_t and the ground truth x_0, and the term to be minimized can be formulated as Eq. (6). Detailed proof can be found in Appendix A.1.

$$res\epsilon = \epsilon + \frac{(1 - \sqrt{\alpha_t})\sqrt{1 - \overline{\alpha}_t}}{\beta_t}R, \quad \mathbb{E}_{x_0,\epsilon,t}[||res\epsilon - res\epsilon_\theta(x_t, t)||^2] \tag{6}$$

Through this process, we transform the learning of $\epsilon_\theta(x_t, t)$ into $res\epsilon_\theta(x_t, t)$. $\epsilon_\theta(x_t, t)$ represents the noise of the noisy ground truth, while $res\epsilon_\theta(x_t, t)$ represents the residual noise between the input degraded images and the ground truth. We name this process **resnoise-diffusion**.

2.2 Smooth equivalence transformation

According to Eq. (5) and Eq. (1), we can derive Eq. (7). It is worth mentioning that x_T is uncomputable because the ground truth x_0 is unavailable in Eq. (7) during the reverse process, so we can not initialize x_T directly.

$$x_t = (2\sqrt{\overline{\alpha}_t} - 1)x_0 + (1 - \sqrt{\overline{\alpha}_t})\hat{x}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon, \epsilon \sim N(0, I) \tag{7}$$

Fortunately, the weighted coefficient of x_0, which is $(2\sqrt{\overline{\alpha}_t} - 1)$ in Eq. (7), can be very close to zero. Since the input degraded image $\hat{x}_0$ is available, we can find a time step T' where $x_{T'}$ is computable. When T is sufficiently large, the variation of $\sqrt{\overline{\alpha}_t}(t \leq T)$ with respect to time t is smooth. We call this technique **smooth equivalence transformation**. Therefore, we can derive T' as Eq. (8) and obtain $x_{T'}$ in Eq. (9) with a small bias. This bias can also be eliminated through the Truncated Schedule technique that we propose next.

$$T' = \arg\min_{i=1}^{T} |\sqrt{\overline{\alpha}_i} - \frac{1}{2}| \tag{8}$$

$$x_{T'} \approx (1 - \sqrt{\overline{\alpha}_{T'}})\hat{x}_0 + \sqrt{1 - \overline{\alpha}_{T'}}\epsilon \approx \sqrt{\overline{\alpha}_{T'}}\hat{x}_0 + \sqrt{1 - \overline{\alpha}_{T'}}\epsilon \tag{9}$$

Thus we only need to minimize Eq. (6) when $t \leq T'$ since $x_{T'}$ is available, as shown in Eq. (10).

$$P_\theta(x_0) = \int_{x_1:x_{T'}} P_{data}(x_{T'}) \prod_{t=1}^{T'} P_\theta(x_{t-1}|x_t)dx_1 : x_{T'} \tag{10}$$

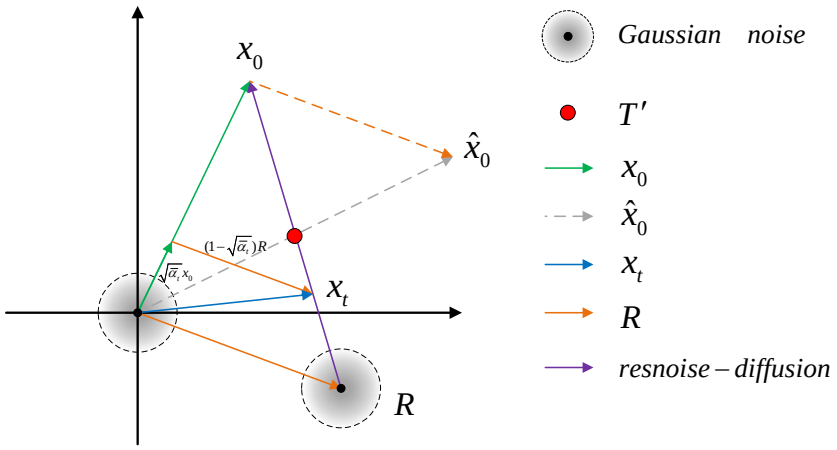

Figure 2: The working principle of Resfusion. x_0 represents the distribution of the ground truth, while $\hat{x}_0$ represents the distribution of the degraded images. $\hat{x}_0 - x_0$ represents the gap between them, defined as the residual term R in Eq. (1). Resfusion does not explicitly guide $\hat{x}_0$ to x_0. Instead, it implicitly learns the distribution of R by doing resnoise-diffusion reverse process from x_t to x_0. The resnoise-diffusion reverse process can be imagined as doing diffusion reverse process from $R + \epsilon$ to x_0 (as shown by the *violet* arrow), guiding x_t gradually towards x_0 along this direction. Following the principles of similar triangles, the coefficient of R at step t is computed as $1 - \sqrt{\overline{\alpha}_t}$. At any step t during the training process, x_t can be calculated based on x_0 and R through Eq. (4).

The resnoise-diffusion reverse process can be formulated as Eq. (11) and Eq. (12). Consistent with the definition in DDPM, the Σ_θ is taken fixed as $\widetilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$.

$$P_\theta(x_{0:T'-1}|x_{T'}) = \prod_{t=1}^{T'} P_\theta(x_{t-1}|x_t) \tag{11}$$

$$P_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad P_\theta(x_0|x_1) = N(x_0; \mu_\theta(x_1, 1)) \tag{12}$$

The mean $\mu_\theta(x_t, t)$ of resnoise-diffusion reverse process can be formalized as Eq. (13), which is demonstrated in Appendix A.1 from Eq. (19) to Eq. (23).

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} res\epsilon_\theta) \tag{13}$$

Similar to previous work [5, 17], our method enhances the diffusion model by incorporating a conditioning function. This function integrates latent representation from both the current estimate x_t and the input degraded image $\hat{x}_0$. Then Eq. (6) can be modified to Eq. (14).

$$\mathbb{E}_{x_0,\epsilon,t}[||res\epsilon - res\epsilon_\theta((x_t, \hat{x}_0, t)||^2] \tag{14}$$

Just like the vanilla DDPM, Resfusion gradually fit the current estimate x_t to the ground truth x_0, implicitly reducing the residual term R between $\hat{x}_0$ and x_0 with the resnoise. When we define $\hat{x}_t$ as Eq. (15) with the same Gaussian noise ϵ in x_t, $\hat{x}_{T'}$ can be seen as an intermediate result in an implicit DDPM process with the input degraded image $\hat{x}_0$ as the target distribution. We essentially quantitatively computed the accelerated step T', on which step $x_{T'}$ is closed to $\hat{x}_{T'}$. When T is large enough, the approximate equal sign will become an equal sign. The implicit DDPM reverse process (ϵ to $\hat{x}_0$) is deterministic because $\hat{x}_0$ is available during the inference. The determination of step T' corresponds to the point where the posterior probability distribution (resnoise-diffusion reverse process) becomes consistent with the prior probability distribution (implicit DDPM reverse process).

$$\hat{x}_t = \sqrt{\overline{\alpha}_t}\hat{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \quad \sqrt{\overline{\alpha}_{T'}} \approx 1 - \sqrt{\overline{\alpha}_{T'}} \approx \frac{1}{2}, \quad x_{T'} \approx \hat{x}_{T'} \tag{15}$$

As shown in Fig. 2, the resnoise-diffusion reverse process ($R + \epsilon$ to x_0) intersects with implicit DDPM reverse process (ϵ to $\hat{x}_0$), this intersection corresponds to step T'. The intersection of two

diagonals of a parallelogram is the midpoint of them (corresponding to 0.5 in Eq. (11)), but due to the discrete nature of the diffusion process, the acceleration point actually falls on the point closest to the intersection, which is step T' as Eq. (8). Meanwhile, since $x_{T'}$ is available, resnoise-diffusion steps after step T' are not necessary according to Eq. (10). Therefore, Resfusion can directly start from step T' for both inference and training process. Because the α coefficient of resnoise-diffusion is exactly the same as vanilla DDPM, Resfusion can directly use any existing noise schedule. In practical implementation, we utilized a technique called **Truncated Schedule** to control the offset between $x_{T'}$ and $\hat{x}_{T'}$ when T is small. Further details can be found in Appendix A.7.

3 Experiments

Table 1: Quantitative comparisons with other shadow removal methods. We report PSNR, SSIM [18] and MAE in the shadow region (S), the non-shadow region (NS) and all image (ALL). The best and second-best results are highlighted in **bold** and underlined. "↑" (resp. "↓") means the larger (resp. smaller), the better. We use the symbol "-" to indicate models or results that are unavailable.

	ISTD [19]										
	Method	Params	Shadow Region (S)			Non-Shadow Region (NS)			All Image (ALL)		
			PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	MAE ↓
256 × 256	Input Image	-	22.40	0.936	32.11	27.32	0.976	6.83	20.56	0.893	10.97
	ST-CGAN [20]	31.8M	33.74	0.981	9.99	29.51	0.958	6.05	27.44	0.929	6.65
	DSC [21]	22.3M	34.64	0.984	8.72	31.26	0.969	5.04	29.00	0.944	5.59
	DHAN [22]	21.8M	35.53	0.988	7.49	31.05	0.971	5.30	29.11	0.954	5.66
	FusionNet [23]	186.5M	34.71	0.975	7.91	28.61	0.880	5.51	27.19	0.945	5.88
	UnfoldingNet [24]	10.1M	36.95	0.987	8.29	31.54	0.978	4.55	29.85	0.960	5.09
	DMTN [25]	22.8M	35.83	0.990	7.00	33.01	0.979	4.28	30.42	0.965	4.72
	RDDM (SM-Res-N) [17]	15.5M	36.74	0.988	6.67	33.18	**0.979**	**4.27**	30.91	0.962	**4.67**
	Resfusion (ours)	**7.7M**	**37.51**	**0.990**	**6.49**	**34.26**	0.978	4.48	**31.81**	**0.965**	4.81
Original	Input Image	-	22.34	0.935	33.23	26.45	0.947	7.25	20.33	0.874	11.35
	ARGAN [26]	-	-	-	9.21	-	-	6.27	-	-	6.63
	DHAN [22]	21.8M	34.79	0.983	8.13	29.54	0.941	5.94	27.88	0.921	6.29
	CANet [27]	358.2M	-	-	8.86	-	-	6.07	-	-	6.15
	Resfusion (ours)	**7.7M**	**36.45**	**0.985**	**7.08**	**32.08**	**0.950**	**5.02**	**30.09**	**0.932**	**5.34**

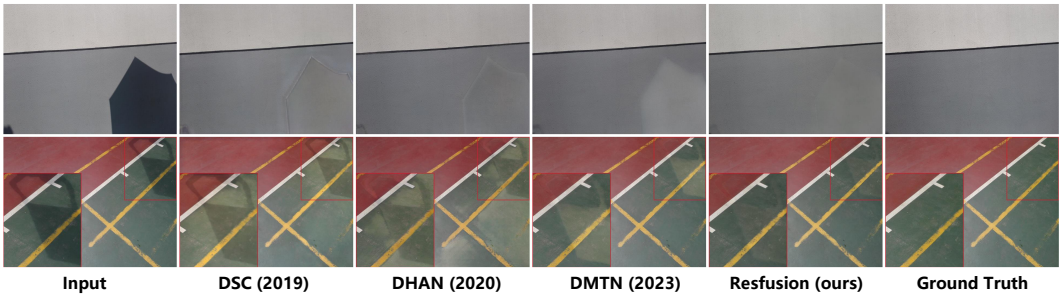

Figure 3: Visual comparisons of the restored results by different shadow-removal methods on the ISTD dataset.

To verify the performance of Resfusion, we conducted experiments on three image restoration tasks, including shadow removal, low-light enhancement and deraining. For fair comparisons, we use an U-net [28] structure which is the **same** as RDDM as the backbone. We simply concatenate x_t and $\hat{x}_0$ in the channel dimension and feed them into the network. For all the tasks, we only employ one U-net to predict resnoise. Furthermore, we simply utilize a truncated linear schedule [29] and perform only **five** sampling steps for all datasets. The experimental setting details are provided in the Appendix A.4.

ISTD dataset [19] is a dataset designed for shadow removal, comprising 1870 sets of image triplets consisting of shadow image, shadow mask, and shadow-free image. It consists of 1330 image triplets for training and 540 image triplets for quantitative evaluations. We compare the proposed method with the popular shadow removal methods, i.e., ST-CGAN [20], DSC [21], ARGAN [26], DHAN [22],

Table 2: Quantitative comparisons with other low-light enhancement methods. We report PSNR, SSIM and LPIPS [38]. The best and second-best results are highlighted in **bold** and underlined. "↑" (resp. "↓") means the larger (resp. smaller), the better.

LOL [30]				
Method	Params	PSNR ↑	SSIM ↑	LPIPS ↓
YCbCr space, 256 × 256				
Input Image	-	9.30	0.377	0.513
RDDM (SM-Res-N) [17]	15.5M	23.90	0.931	-
RDDM (SM-Res) [17]	7.7M	25.39	0.937	0.116
Resfusion (ours)	**7.7M**	**30.02**	**0.954**	**0.070**
RGB space, Original				
Input Image	-	7.77	0.191	0.560
RetinexNet [30]	0.6M	16.77	0.560	0.474
KinD [31]	8.0M	20.87	0.790	0.170
KinD++ [32]	9.6M	21.30	0.820	0.160
Zero-DCE [33]	**0.3M**	14.86	0.562	0.335
EnlightenGAN [34]	8.6M	17.48	0.652	0.322
Restormer [35]	-	22.37	0.816	0.141
LLFormer [19]	24.6M	23.65	0.816	0.169
Resfusion (ours)	7.7M	**24.63**	**0.860**	**0.107**

Table 3: Quantitative comparisons with other deraining methods. We report PSNR and SSIM. The best and second-best results are highlighted in **bold** and underlined. "↑" means the larger, the better.

Raindrop [39]			
Method	Params	PSNR ↑	SSIM ↑
YCbCr space, 256 × 256			
Input Image	-	25.81	0.887
RDDM (SM-Res-N) [17]	15.5M	32.51	0.956
Resfusion (ours)	**7.7M**	**34.40**	**0.975**
YCbCr space, Original			
Input Image	-	25.40	0.882
pix2pix [40]	-	28.02	0.855
AttentiveGAN [39]	**6.2M**	31.59	0.917
DuRN [41]	10.2M	31.24	0.926
RaindropAttn [42]	-	31.44	0.926
All-in-One [43]	-	31.12	0.927
IDT [44]	16.4M	31.87	0.931
WeatherDiff_{64} [5]	82.9M	30.71	0.931
$\text{RainDropDiff}_{128}$ [5]	109.7M	32.43	0.933
Resfusion (ours)	7.7M	**32.61**	**0.938**

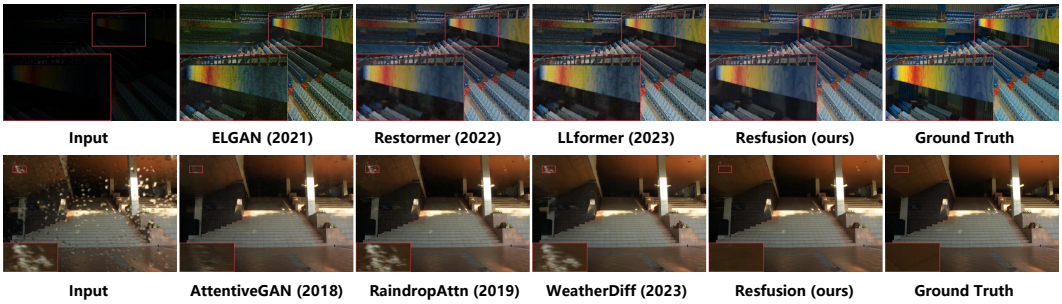

Figure 4: Visual comparisons of the restored results by different image restoration methods on the LOL dataset and the Raindrop dataset.

FusionNet [23], CANet [27], UnfoldingNet [24], DMTN [25], and RDDM(SM-Res-N) [17]. In order to ensure a fair comparison, we conducted experiments on two settings for the ISTD dataset, following the methods used in DMTN [25] and DHAN [22]: (1) The results are evaluated at a resolution of 256 × 256 after being resized. (2) The original image resolutions (640 × 480) are maintained for evaluation.

LOL dataset [30] comprises 500 pairs of images, consisting of both low-light and normal-light versions, which are further divided into 485 training pairs and 15 evaluation pairs. The low-light images contain noise produced during the photo capture process. We compare the proposed method with the popular low-light enhancement methods, i.e., RetinexNet [30], KinD [31], KinD++ [32], Zero-DCE [33], EnlightenGAN [34], Restormer [35], LLFormer [19], RDDM (SM-Res-N) [17], and RDDM (SM-Res) [17]. Some existing methods [8, 36, 37] calculate metrics by adjusting the overall brightness based on reference images (called as using GT-mean). However, this approach can introduce biases and potential unfairness. In accordance with LLFormer [19], we compute metrics without utilizing any reference information. To ensure a fair comparison, we conducted experiments on two settings for the LOL dataset, following the methods employed in RDDM [17] and LLFormer [19]: (1) The results are evaluated at a resolution of 256 × 256 after being resized. PSNR and SSIM are evaluated based in YCbCr color space. (2) The original image resolutions (600 × 400) are maintained for evaluation. PSNR and SSIM are evaluated in RGB color space.

Raindrop dataset [39] is a dataset designed for deraining, comprising 861 training image pairs for training, and 58 image pairs dedicated for quantitative evaluations, denoted in [39] and [5] as Raindrop-A. We compare the proposed method with the popular deraining methods, i.e., pix2pix [40],

AttentiveGAN [39], DuRN [41], RaindropAttn [42], All-in-One [43], IDT [44], WeatherDiff [5], RainDropDiff [5], and RDDM(SM-Res-N)[17]. We conduct experiments on two settings for the Raindrop dataset for fairness, following the methods employed in RDDM [17] and WeatherDiff [5]: (1) The results are evaluated at a resolution of 256×256 after being resized. (2) The original image resolutions are maintained for evaluation.

Quantitative comparison. As shown in the Tables 1, 2 & 3, We provide the quantitative evaluation results on ISTD dataset, LOL dataset and Raindrop dataset. Our methods clearly outperform all competing methods by significant margins in terms of PSNR, SSIM, MAE and LPIPS across all three datasets. The current experimental results demonstrate that Resfusion achieves highly competitive results under these conditions: (1) utilizing only one U-net to predict resnoise. (2) simply concatenating x_t and $\hat{x}_0$ in the channel dimension. (3) employing a simple truncated linear schedule. (4) conducting only **five** sampling steps for all datasets. In contrast, alternative methods often rely on intricate network architectures, including multi-stage [19, 23, 24], multi-scale [45], multi-branch [22] and prior knowledge of physics [9, 30, 33, 42], complex noise schedules [16, 17] and patch-overlapping strategy [5]. For the ISTD dataset and Raindrop dataset, we only employ one U-net to predict resnoise, outperforming RDDM with two U-nets to predict the residual term and the noise separately in terms of PSNR and SSIM. Resfusion use **half** the number of parameters of RDDM and achieved better quantitative evaluation metrics. For the LOL dataset, under the **same** parameters, Resfusion outperforms RDDM in terms of PSNR (+18%) and LPIPS (-40%) significantly. Furthermore, for all datasets, we employed a simple truncated linear schedule, while RDDM utilized a complex custom noise schedule.

4 Ablation Study

4.1 The analysis of the residual term and the noise term

Figure 5: The analysis of the residual term and the noise term on the LOL dataset. Only removing noise will reconstruct the details of the degraded image without causing any semantic shift. Only removing residual can only accomplish the semantic shift (from low-light to normal-light) without reconstructing the details. Removing resnoise can achieve both the semantic shift and the detail reconstruction.

Resfusion and DDPM share the consist form of the reverse inference process. The DDPM reverse process restores the original image distribution by gradually removing the noise (ϵ_θ) from Gaussian white noise, while resnoise-diffusion reverse process restores clean image by gradually removing the resnoise ($res\epsilon_\theta$) from the noisy degraded images. Mathematically, the resnoise term ($res\epsilon_\theta$) and the noise term (ϵ_θ) only differ in the form of a weighted residual term ($\frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t}R_\theta$). In other words, Resfusion subtracts an extra weighted residual term while removing the noise term during each step of the reverse process.

According to the Green's theorem [46], when the neural network is sufficiently robust, the components of the resnoise should be path independent. Based on this belief, we trained two separate neural networks on the LOL dataset. One network predicts only ϵ_θ and removes only the noise term during the resnoise-diffusion reverse process. The other network predicts only $\frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t}R_\theta$ and removes only the weighted residual term during the resnoise-diffusion reverse process. As shown

in Fig 5, we qualitatively determine the functionalities of each component in the loss function of Resfusion. The weighted residual term are responsible for semantic shift, while the noise term handles detail reconstruction. Predicting resnoise $res\epsilon_\theta$ achieves both semantic shifts and detail reconstruction, bridging the gap between the input degraded image and the ground truth, enabling effective image restoration.

4.2 Equivalent representations of the loss function

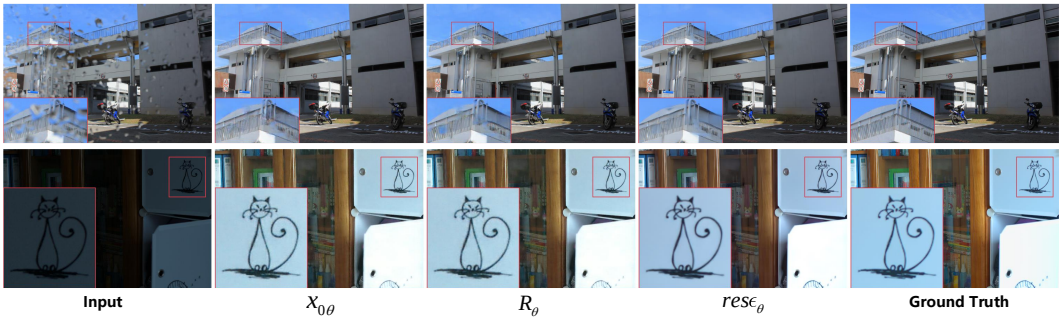

Figure 6: Visual comparisons of the restored results generated by other equivalent loss functions on Raindrop dataset and LOL dataset. Using $res\epsilon_\theta$ as the loss function allows for better restoration of details while completing semantic shifting.

Based on Eq. (1), since $\hat{x}_0$ is available, once we acquire either x_0 or R, we can determine the other. Moreover, according to Eq. (18), since x_t is obtainable and considering the equivalence between x_0 and R, we can also obtain the equivalent reverse process of resnoise-diffusion by predicting either $x_{0\theta}$ or R_θ. Similar to DDPM, the fundamental purpose of Resfusion is also to predict x_0 (which is equivalent to predicting R). In contrast, DDPM reconstructs the distribution of the original image x_0 by incrementally reducing noise from the Gaussian white noise, whereas Resfusion skips the generation of low-frequency information by utilizing $\hat{x}_0$ for initialization. Therefore, in addition to the noise removal, Resfusion also involves removing a weighted residual term to accomplish the semantic shifting. It is worth mentioning that, unlike RDDM, predicting the noise term ϵ_θ is **not** an equivalent loss function. Due to the truncated schedule we adopt, the starting point $x_{T'}$ will only consist of weighted $\hat{x}_0$ and ϵ. As $\hat{x}_0$ is obtainable and serves as a conditional input to the neural network, directly predicting the noise term would lead to the neural network learning a simple pattern, resulting in training failure.

We compare the quantitative performance obtained by using different equivalent prediction targets as loss functions on ISTD dataset, LOL dataset and Raindrop dataset. We use the same backbone as described in Sec. 3 and employ the same truncated linear schedule, performing five sampling steps for all datasets. The original image resolutions are maintained for evaluation. As shown in Table 4, predicting $res\epsilon_\theta$ outperformed predicting $x_{0\theta}$ and R_θ in terms of PSNR and SSIM on all three datasets. Predicting $res\epsilon_\theta$ resulted in better reconstruction of the fine details, as illustrated in Fig 6.

Table 4: Quantitative comparisons with other equivalent loss functions on ISTD dataset, LOL dataset and Raindrop dataset. We report PSNR, SSIM, MAE and LPIPS. The best and second-best results are highlighted in **bold** and <u>underlined</u>. "↑" (resp. "↓") means the larger (resp. smaller), the better.

Prediction Targets	ISTD [19]			LOL [30]			RainDrop [39]	
	PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
$x_{0\theta}$	29.67	0.927	5.35	<u>23.10</u>	<u>0.813</u>	0.150	32.51	0.935
R_θ	<u>29.75</u>	<u>0.930</u>	**5.26**	22.87	0.807	<u>0.143</u>	<u>32.57</u>	<u>0.935</u>
$res\epsilon_\theta$	**30.09**	**0.932**	<u>5.34</u>	**24.63**	**0.860**	**0.107**	**32.61**	**0.938**

Figure 7: Visual comparisons between DDPM and Resfusion on the CIFAR10 (32×32) dataset. We do not cherry-pick any results. With the same sampling steps, Resfusion outperforms DDPM in semantic generation and detail reconstruction.

5 Discussion

Resfusion is not limited to image restoration. In fact, It is a versatile framework which can be applied to any general image generation domain. For image generation tasks, as it is impossible to obtain any additional information, we redefine $\hat{x}_0$ as a zero matrix. Thus we can obtain a new definition of the residual term $R = -x_0$ for image generation. Therefore, $res\epsilon$ is redefined as $res\epsilon = \epsilon - \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} x_0$. Applied with the redefined residual term and the resnoise, Resfusion's forward and reverse process for image generation are completely consistent with the original resnoise-diffusion process for image retoration.

The redefined $res\epsilon$ further explains why Resfusion can complete the reverse inference process in fewer sampling steps than DDPM, under the same noise schedule. Due to consistency between the reverse processes, both of Resfusion and DDPM require the removal of the noise term. However, compared to DDPM, Resfusion additionally add a weighted x_0 instead of simply removing the noise. This is the key factor that allows Resfusion to diffuse faster.

We train DDPM, Resfusion, and DDIM [2] on the CIFAR10 (32×32) dataset [47] with the same backbone. The experimental details are provided in the Appendix A.4. A truncated linear schedule is used for Resfusion, and a linear schedule is used for DDPM and DDIM. We employ the Frechet Inception Distance (FID) [48] as the quantitative metric. As shown in Table 5, Resfusion significantly outperforms DDPM with the same sampling steps. At nearly half of the sampling steps, Resfusion achieves a similar FID with DDPM. Interestingly, when using a truncated linear schedule, the value of T'/T is also closed to 0.5, further validating Resfusion's accelerated sampling property. Consistent with DDPM, Resfusion also performs stochastic steps during the reverse process. Due to its stochastic nature, similar to DDPM, Resfusion's performance is lower than the deterministic DDIM.

Table 5: Quantitative comparisons with DDPM and DDIM on CIFAR10 (32×32) dataset. We report FID under different sampling steps. "↓" means the smaller, the better.

CIFAR10 (FID ↓)	DDPM	Resfusion(ours)	DDIM
10 steps	43.11	28.81	18.37
20 steps	24.88	15.46	10.93
50 steps	14.02	7.96	7.39
100 steps	9.79	6.31	6.21

Figure 8: Visualization of the five sampling steps, where the *blue* arrow represents the smooth equivalence transformation, and the *red* box represents the resnoise-diffusion reverse process. We select the LOL web Test dataset (which **does not** have ground truth) and the Raindrop-B test dataset (which is much more challenging than Raindrop-A) to showcase the effects of low-light enhancement and deraining. We directly use the pretrained models on LOL dataset and Raindrop dataset, demonstrating the strong robustness of Resfusion.

6 Conclusion

We propose the Resfusion, a general framework for image restoration. We explicitly provide the quantitative relationship between the residual term and the noise term, named as resnoise. Through employing the smooth equivalence transformation, we unify the training and inference process. Resfusion does not require any prior physical knowledge and can directly utilize existing noise schedules. Experimental results shows that Resfusion exhibits competitive performance for shadow removal, low-light enhancement, and deraining tasks, with only five sampling steps. It is important to note that Resfusion is not limited to image restoration and can be applied to any image generation domain. The versatility of our framework lies in its ability to simultaneously model the residual term and the noise term. Our subsequent experiments have demonstrated that Resfusion can be easily applied to various image generation tasks and exhibits strong competitiveness.

7 Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (62272248), the Natural Science Foundation of Tianjin (23JCZDJC01010, 23JCQNJC00010) and the Key Program of Science and Technology Foundation of Tianjin (24HHXCSS00004).

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[2] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[3] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

[4] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

[5] Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[6] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16293–16303, 2022.

[7] Tao Wang, Kaihao Zhang, Ziqian Shao, Wenhan Luo, Bjorn Stenger, Tae-Kyun Kim, Wei Liu, and Hongdong Li. Lldiffusion: Learning degradation representations in diffusion models for low-light image enhancement. *arXiv preprint arXiv:2307.14659*, 2023.

[8] Jinhui Hou, Zhiyu Zhu, Junhui Hou, Hui Liu, Huanqiang Zeng, and Hui Yuan. Global structure-aware diffusion process for low-light image enhancement. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] Lanqing Guo, Chong Wang, Wenhan Yang, Siyu Huang, Yufei Wang, Hanspeter Pfister, and Bihan Wen. Shadowdiffusion: When degradation prior meets diffusion model for shadow removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14049–14058, 2023.

[10] Chunming He, Chengyu Fang, Yulun Zhang, Kai Li, Longxiang Tang, Chenyu You, Fengyang Xiao, Zhenhua Guo, and Xiu Li. Reti-diff: Illumination degradation image restoration with retinex-based latent diffusion model. *arXiv preprint arXiv:2311.11638*, 2023.

[11] Chunming He, Yuqi Shen, Chengyu Fang, Fengyang Xiao, Longxiang Tang, Yulun Zhang, Wangmeng Zuo, Zhenhua Guo, and Xiu Li. Diffusion models in low-level vision: A survey. *arXiv preprint arXiv:2406.11138*, 2024.

[12] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023.

[13] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

[14] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *arXiv preprint arXiv:2301.11699*, 2023.

[15] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

[16] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36, 2024.

[17] Jiawei Liu, Qiang Wang, Huijie Fan, Yinong Wang, Yandong Tang, and Liangqiong Qu. Residual denoising diffusion models. *arXiv preprint arXiv:2308.13712*, 2023.

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

[19] Tao Wang, Kaihao Zhang, Tianrun Shen, Wenhan Luo, Bjorn Stenger, and Tong Lu. Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2654–2662, 2023.

[20] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1788–1797, 2018.

[21] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection and removal. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2795–2808, 2019.

[22] Xiaodong Cun, Chi-Man Pun, and Cheng Shi. Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10680–10687, 2020.

[23] Lan Fu, Changqing Zhou, Qing Guo, Felix Juefei-Xu, Hongkai Yu, Wei Feng, Yang Liu, and Song Wang. Auto-exposure fusion for single-image shadow removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10571–10580, 2021.

[24] Yurui Zhu, Zeyu Xiao, Yanchi Fang, Xueyang Fu, Zhiwei Xiong, and Zheng-Jun Zha. Efficient model-driven network for shadow removal. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 3635–3643, 2022.

[25] Jiawei Liu, Qiang Wang, Huijie Fan, Wentao Li, Liangqiong Qu, and Yandong Tang. A decoupled multi-task network for shadow removal. *IEEE Transactions on Multimedia*, 2023.

[26] Bin Ding, Chengjiang Long, Ling Zhang, and Chunxia Xiao. Argan: Attentive recurrent generative adversarial network for shadow detection and removal. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10213–10222, 2019.

[27] Zipei Chen, Chengjiang Long, Ling Zhang, and Chunxia Xiao. Canet: A context-aware network for shadow removal. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4743–4752, 2021.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[30] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018.

[31] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1632–1640, 2019.

[32] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129:1013–1037, 2021.

[33] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin 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.

[34] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE transactions on image processing*, 30:2340–2349, 2021.

[35] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022.

[36] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex Kot. Low-light image enhancement with normalizing flow. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2604–2612, 2022.

[37] Dewei Zhou, Zongxin Yang, and Yi Yang. Pyramid diffusion models for low-light image enhancement. *arXiv preprint arXiv:2305.10028*, 2023.

[38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[39] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[40] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

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

[42] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2463–2471, 2019.

[43] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3175–3185, 2020.

[44] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[45] Lanqing Guo, Siyu Huang, Ding Liu, Hao Cheng, and Bihan Wen. Shadowformer: global context helps shadow removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 710–718, 2023.

[46] Bernhard Riemann. *Grundlagen fur eine allgemeine Theorie der Functionen einer veränderlichen complexen Grösse*. Adalbert Rente, 1867.

[47] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[48] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[49] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[50] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[51] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36, 2024.

[52] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *arXiv preprint arXiv:2307.12348*, 2023.

[53] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[54] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[55] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[58] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8578–8587, 2019.

[59] Yurui Zhu, Jie Huang, Xueyang Fu, Feng Zhao, Qibin Sun, and Zheng-Jun Zha. Bijective mapping network for shadow removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5627–5636, 2022.

[60] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[61] Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021.

[62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[63] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1680–1691, 2023.

[64] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

A Appendix Section

A.1 Detailed proof

According to Eq. (4) and Eq. (5), the forward process can be written as Eq. (16) and Eq. (17).

$$q(x_t|x_{t-1}, R) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1} + (1-\sqrt{\alpha_t})R, (1-\alpha_t)I) \tag{16}$$

$$q(x_t|x_0, R) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0 + (1-\sqrt{\overline{\alpha}_t})R, (1-\overline{\alpha}_t)I) \tag{17}$$

By simply performing a change of variable ($x_0 \to x_0 - R$, $x_t \to x_t - R$, $x_{t-1} \to x_{t-1} - R$), the derivation of Eq. (18) is identical in form to (71) - (84) in the reference [49], where line 6 - 7 of Eq. (18) corresponds to (73).

$$\begin{aligned}
&q(x_{t-1}|x_t, x_0, R) \\
&= \frac{q(x_t|x_{t-1}, x_0, R)q(x_{t-1}|x_0, R)}{q(x_t|x_0, R)} \\
&= \frac{\mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1} + (1-\sqrt{\alpha_t})R, (1-\alpha_t)I)\mathcal{N}(x_{t-1}; \sqrt{\overline{\alpha}_{t-1}}x_0 + (1-\sqrt{\overline{\alpha}_{t-1}})R, (1-\overline{\alpha}_{t-1})I)}{\mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0 + (1-\sqrt{\overline{\alpha}_t})R, (1-\overline{\alpha}_t)I)} \\
&\propto exp\{-[\frac{[x_t - (\sqrt{\alpha_t}x_{t-1} + (1-\sqrt{\alpha_t})R)]^2}{2(1-\alpha_t)} + \frac{[x_{t-1} - (\sqrt{\overline{\alpha}_{t-1}}x_0 + (1-\sqrt{\overline{\alpha}_{t-1}})R)]^2}{2(1-\overline{\alpha}_{t-1})} \\
&- \frac{[x_t - (\sqrt{\overline{\alpha}_t}x_0 + (1-\sqrt{\overline{\alpha}_t})R)]^2}{2(1-\overline{\alpha}_t)}]\} \\
&= exp\{-[\frac{[(x_t - R) - \sqrt{\alpha_t}(x_{t-1} - R)]^2}{2(1-\alpha_t)} + \frac{[(x_{t-1} - R) - \sqrt{\overline{\alpha}_{t-1}}(x_0 - R)]^2}{2(1-\overline{\alpha}_{t-1})} \\
&- \frac{[(x_t - R) - \sqrt{\overline{\alpha}_t}(x_0 - R)]^2}{2(1-\overline{\alpha}_t)}]\} \\
&\propto \mathcal{N}(x_{t-1} - R; \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})(x_t - R) + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)(x_0 - R)}{1-\overline{\alpha}_t}, \widetilde{\beta}_t I) \\
&\propto \mathcal{N}(x_{t-1}; \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})(x_t - R) + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)(x_0 - R)}{1-\overline{\alpha}_t} + R, \widetilde{\beta}_t I)
\end{aligned} \tag{18}$$

Then we can derive $\widetilde{\mu}(x_t, x_0, R)$ as Eq. (19).

$$\widetilde{\mu}(x_t, x_0, R) = \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}(x_0 - R) + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}(x_t - R) + R \tag{19}$$

We can derive Eq. (20) through Eq. (5).

$$x_t - R = \sqrt{\overline{\alpha}_t}(x_0 - R) + \sqrt{1-\overline{\alpha}_t}\epsilon, \quad \epsilon \sim N(0, I) \tag{20}$$

Thus we can derive Eq. (21) through Eq. (19) and Eq. (20). By simply performing a change of variables ($x_0 \to x_0 - R$, $x_t \to x_t - R$), the derivation process becomes exactly identical in form to the derivation of equations (115) - (124) in the reference [49], where Eq. (20) corresponds to (115) and Eq. (19) corresponds to (116).

$$\begin{aligned}
L_{t-1} - C &= \mathbb{E}_{x_0,\epsilon,t}[\frac{1}{2\sigma_t^2}||\widetilde{\mu}(x_t, x_0, R) - \mu_\theta(x_t(x_0, \epsilon, R), t)||^2] \\
&= \mathbb{E}_{x_0,\epsilon,t}[\frac{1}{2\sigma_t^2}||\{\frac{1}{\sqrt{\alpha_t}}[(x_t(x_0, \epsilon, R) - R) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon] + R\} - \mu_\theta(x_t(x_0, \epsilon, R), t)||^2]
\end{aligned} \tag{21}$$

According to Eq. (22), we can modify Eq. (21) as Eq. (23).

$$\begin{aligned}
\frac{1}{\sqrt{\alpha_t}}(x_t - R - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon) + R &= \frac{1}{\sqrt{\alpha_t}}(x_t - R + \sqrt{\alpha_t}R - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon) \\
&= \frac{1}{\sqrt{\alpha_t}}(x_t - (1-\sqrt{\alpha_t})R - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon) \\
&= \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t}\frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}R - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon) \\
&= \frac{1}{\sqrt{\alpha_t}}[x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}(\epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t}R)]
\end{aligned} \tag{22}$$

$$
\begin{aligned}
&L_{t-1} - C \\
&= \mathbb{E}_{x_0,\epsilon,t}[\frac{1}{2\sigma_t^2}||\frac{1}{\sqrt{\alpha_t}}[x_t(x_0,\epsilon,R) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}(\epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t}R)] - \mu_\theta(x_t(x_0,\epsilon,R),t)||^2]
\end{aligned} \tag{23}
$$

According to [1], the minimize term become Eq. (6).

A.2 Comparison with other methods

The main difference is in how the denoising diffusion, score, flow, or Schrödinger's bridge are adapted to image restoration. Different methods select various elements as prediction targets: the noise term (Shadow Diffusion [9], SR3 [3], WeatherDiffusion [5]), the residual term (DvSR [6], Rectified Flow [50]), the target image (ColdDiffusion [51], InDI [12]), or its linear transformation term (I^2SB [15]). Similar to RDDM [17], Resfusion simultaneously predicts both the residual term and the noise term, and provides the quantitive relationship between them.

Comparison with traditional diffusion-based methods. Traditional diffusion-based image restoration methods [5, 6, 7, 8, 9] adapt the diffusion model for image restoration tasks with degraded images as conditional input to implicitly guide the reverse generation process, without altering the original denoising diffusion process [1, 2]. Starting the reverse process from Gaussian white noise, traditional diffusion-based models consider only the degraded images as conditional input, resulting in an increased number of sampling steps. Meanwhile, these models are often task-specific, requiring the design of different model structures based on different scenarios. By introducing the residual term into the diffusion forward process, Resfusion bridge the gap between the input degraded images and ground truth, starting the reverse process directly from the noisy degraded images. As a versatile methodology for image restoration, Resfusion does not require any physical prior knowledge, and the image restoration can be completed in just five sampling steps.

Comparison with RDDM [17]. RDDM can be seen as a diffusion process from the noisy input degraded image to the ground truth, while Resfusion represents a diffusion process from the noisy residual term to the ground truth. RDDM predicts the residual term and the noise term separately without specifying their weighted relationship, using a complex Automatic Objective Selection Algorithm (AOSA) to learn them. In contrast, Resfusion calculates the quantitative relationship between the residual term and the noise term, naming their weighted sum as resnoise. RDDM's forward process accumulates the residual term and the noise term, making its forward and backward processes inconsistent with DDPM, leading to poor generalization and interpretability. By transforming the learning of the noise term into the resnoise term, Resfusion's reverse inference process becomes consistent with DDPM, unifying the training and inference processes. Lastly, RDDM requires a customized noise schedule, as using existing noise schedules results in performance loss. Through the smooth equivalence transformation in resnoise-diffusion process, Resfusion can directly use the existing noise schedule.

Comparison with Resshift [52]. Similar to RDDM, Resshift's forward process also adopts an accumulation strategy for the residual term and the noise term. Therefore, Resshift also requires the design of a complex noise schedule, which is formulated as equation (10) in Resshift [52]. Resfusion can directly use the existing noise schedule instead of redesigning the noise schedule. The reverse process of Resshift is inconsistent with DDPM. The form of Resfusion's reverse inference process is consistent with the DDPM, leading to better generalization and interpretability. The prediction target of Resshift is x_0, while the prediction target of Resfusion is $res\epsilon$. Given that the essence of $res\epsilon$ is the noise term with an offset, and LDM models mainly predict the noise term, the loss function of Resfusion is extremely friendly to fine-tuning techniques such as Lora, which helps further scale up. Resshift diffuses in the latent space, utilizing the powerful encoding capability of models like VQ-GAN [53]. Resfusion, on the other hand, directly diffuses in the RGB space. Resshift only explores fixed degradations such as image super-resolution. Resfusion explores more complex scenarios, including shadow removal, low-light enhancement, and deraining.

Comparison with DvSR [6]. DvSR predicts clean images from input degraded images using a traditional (non-diffusion) network and calculates the residual term between the ground truth and the predicted clean images. DvSR employs denoising-based diffusion models to predict the noise term like DDPM, generating the residual term from Gaussian white noise. Unlike DvSR, Resfusion does not directly learn the residual term. Instead, it indirectly learns the distribution of the residual term through resnoise-diffusion process.

Comparison with ColdDiffusion [51]. ColdDiffusion aims to completely remove random noise from the diffusion model, replacing it with other transformations like blurring and masking. In contrast, Resfusion still incorporates noise diffusion. As shown in our ablation study, Resfusion requires the noise term for detail recovery. Since ColdDiffusion discards random noise, it needs additional degradation injection to enhance generation diversity. To simulate degradation processes for various restoration tasks, ColdDiffusion uses Gaussian blur for deblurring, snowification transform for snow removal, etc. These specific explorations might lose generality. Resfusion employs the residual term for directed diffusion from the ground truth to the noisy residual term, eliminating the need for task-specific degradation operators. Additionally, Resfusion provides solid theoretical derivation, whereas ColdDiffusion lacks a theoretical basis.

A.3 Image translation

Resfusion can also be implemented in image-to-image distribution transformation. By redefining $\hat{x}_0$ as the translated image and x_0 as the target image, we can easily transition Resfusion from image restoration to image translation. We train Resfusion with a truncated linear schedule on CelebA-HQ (64×64) dataset [54] and AFHQV2 (64×64) dataset [55] for image translation with 50 sampling steps. We selected the following image translation tasks: "Dog $\rightarrow$ Cat", "Male $\rightarrow$ Cat", "Male $\rightarrow$ Female", and "Female $\rightarrow$ Male". As shown in Fig. 9, Resfusion can effectively model the shift between image domains, making it a unified methodology for a wider range of image generation tasks.

Figure 9: Visual results for image translation on the CelebA-HQ dataset and AFHQV2 dataset. The images are presented in pairs, with the translated image on the left and the target image on the right. We showcase the visual results of Resfusion for image translation tasks “Dog $\rightarrow$ Cat”, “Male $\rightarrow$ Cat”, “Male $\rightarrow$ Female”, and “Female $\rightarrow$ Male”.

Table 6: Experimental settings for our Resfusion during the training stage.

Tasks	Image Restoration			Image Generation	Image Translation
	Shadow Removal	Low-light	Deraining		
Datasets	ISTD	LOL	Raindrop	CIFAR-10	CelebA-HQ AFHQ-V2
Batch size	32	32	32	128	128
Image/patch size	256	256	256	32	64
$\hat{x}_0$	I_{in}	I_{in}	I_{in}	0	I_{in}
Sampling steps	5	5	5	10 - 100	50
Learning rate	1.1e-4	1.1e-4	1.1e-4	2e-4	2e-4
Training epochs	5k	5k	5k	3k	3k

A.4 Experimental setting details

We use the PyTorch Lightning framework to train all the models, utilizing the AdamW [56] optimizer with the default settings of PyTorch [57]. Following Liu et al. [17], we utilize the THOP to compute the number of parameters (Params) and multiply-accumulate operations (MACs). We only employ one U-net to predict resnoise and simply utilize a truncated Linear schedule across all tasks. For all the tasks, we implement a regular MSE loss as the loss function. The detailed experimental settings are provided in Table 6. All experiments listed in Table 6 can be carried out with 8 NVIDIA RTX A6000 GPUs.

Image restoration. We evaluate our method on several image restoration tasks, including shadow removal, low-light enhancement, and image deraining on 3 different datasets. For fair comparisons, the results of other image restoration methods are referenced from previously published papers [5, 7, 17, 19, 45] whenever possible. For all image restoration tasks, we used an identical U-net as the backbone, which is the same as RDDM [17]. We take the shadow images and shadow masks together as the input condition (similar to [17, 58, 59]) for the ISTD dataset and only degraded images as the input condition for other datasets. We simply concatenate x_t and $\hat{x}_0$ (and shadow masks) together in the channel dimension and feed them into the network. For the LOL dataset, we **do not** use pre-processing and post-processing techniques like Histogram Equalization and GT-mean. For the Raindrop dataset, we evaluate PSNR and SSIM based on the luminance channel Y of the YCbCr color space in accordance with previous work [5, 17, 39]. We employ a patch size of 256×256 for all the datasets during the training stage. We use the peak signal-to-noise ratio (PSNR), structural similarity (SSIM), learned perceptual image patch similarity (LPIPS), and mean absolute error (MAE) as quantitative metrics.

Image generation and image translation. For image generation on the CIFAR10 (32×32) dataset, we utilize the same U-net structure as DDIM [2]. In contrast to DDIM which employs a linear schedule with $T = 1000$ and a quadratic selection procedure to select sub-sampling steps, we use a linear schedule with $T = 100$ and a linear selection procedure to select sub-sampling steps (for a fair comparison with truncated linear schedule) while training DDPM and DDIM. We use the Frechet Inception Distance (FID) as the quantitative metric. For image translation tasks, we employ a U-net structure with the same configuration as used in DDIM for CelebA [60] as the backbone. To increase the diversity of the generated images, the translated images are not fed into the network as conditional input.

A.5 Algorithm

Based on the derivations from the Sec. 2.1 and Sec. 2.2, the training and inference processes of Resfusion can be represented as Algorithm 1 and Algorithm 2. We highlight the modifications in our training and inference algorithms compared to DDPM in red. Just like the vanilla Denoising Diffusion Probabilistic Models (DDPM), Resfusion gradually fit x_t to x_0, implicitly reducing the residual term between $\hat{x}_0$ and x_0 with the resnoise during the reverse inference process. Through transforming the learning of the noise term into the resnoise term, the form of resnoise-diffusion reverse inference process is consist with DDPM, leading to excellent interpretability.

Algorithm 1 Training Algorithm for Resfusion

Require: total diffusion steps T, degraded image and ground truth dataset $D = (\hat{x}_0^n, x_0^n)_n^N$.
$T' = \arg\min_{i=1}^{T} |\sqrt{\overline{\alpha}_i} - \frac{1}{2}|$
repeat
  Sample $(\hat{x}_0^i, x_0^i) \sim D, \epsilon \sim N(0, I)$
  Sample $t \sim Uniform(1, ..., T')$
  $R = \hat{x}_0 - x_0$
  $x_t = \sqrt{\overline{\alpha}_t} x_0 + (1 - \sqrt{\overline{\alpha}_t})R + \sqrt{1 - \overline{\alpha}_t}\epsilon$
  $res\epsilon = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} R$
  take gradient step on
    $\nabla_\theta ||res\epsilon - res\epsilon_\theta(x_t, \hat{x}_0, t)||^2$
until convergence

Algorithm 2 Inference Algorithm for Resfusion

Require: total diffusion steps T, degraded image $\hat{x}_0$, pretrained Resfusion model $res\epsilon_\theta$.
$\widetilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$
$T' = \arg\min_{i=1}^{T} |\sqrt{\overline{\alpha}_i} - \frac{1}{2}|$
Sample $\epsilon \sim N(0, I)$
$x_{T'} = \sqrt{\overline{\alpha}_{T'}}\hat{x}_0 + \sqrt{1 - \overline{\alpha}_{T'}}\epsilon$
for $t = T', T' - 1, ..., 2$ **do**
  Sample $z \sim N(0, I)$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}(res\epsilon_\theta(x_t, \hat{x}_0, t)) + \sqrt{\widetilde{\beta}_t} z$
end for
return $x_0 = \frac{1}{\sqrt{\alpha_1}}(x_1 - \frac{\beta_1}{\sqrt{1-\overline{\alpha}_1}} res\epsilon_\theta(x_1, \hat{x}_0, 1))$

A.6 Resource efficiency

We compare the parameters, multiply-accumulate operation (MACs) and inference time with other image restoration methods on ISTD [20] dataset, LOL [30] dataset and Raindrop [39] dataset by THOP, using 256×256 images as the input. For ISTD dataset, PSNR and SSIM are evaluated at a resolution of 256×256 after being resized. For LOL dataset and Raindrop dataset, the original image resolutions are maintained for the evaluation of PSNR and SSIM. The experimental results are quoted from the results of previous papers as well as our implementation based on open source code.

As shown in Table 7, for the ISTD dataset, compared to Shadow Diffusion [9], Resfusion has 5× fewer parameters, 5× fewer sampling steps, and 20× fewer MACs. For the LOL dataset, compared to LLDiffusion [7], Resfusion has 6× fewer sampling steps. For the Raindrop dataset, compared to RainDiff_{128} [5], Resfusion has 10× fewer parameters, 10× fewer sampling steps, and 50× fewer MACs. Experiments in shadow removal, low-light enhancement, and deraining demonstrate the effectiveness of Resfusion, enabling computationally constrained researchers to utilize our model for image restoration tasks.

Table 7: Resource efficiency and performance analysis by THOP on ISTD dataset, LOL dataset and Raindrop dataset. "MAC" means multiply-accumulate operation. The best and second-best results are highlighted in **bold** and underlined. "↑" (resp. "↓") means the larger (resp. smaller), the better. We use the symbol "-" to indicate models or results that are unavailable.

Methods	PSNR ↑	SSIM ↑	Params ↓	MACs(G) × Steps ↓	Inference Time (s) ↓
ISTD Dataset					
Shadow Diffusion [9]	**32.33**	**0.969**	55.5M	182.1×25 = 4552.5	0.024×25 = 0.600
SR3 [3]	27.49	0.871	155.3M	155.3×100 = 15530.0	-×100 = -
Resfusion (ours)	31.81	0.965	**7.7M**	**33.3×5 = 167.5**	**0.027×5 = 0.135**
LOL Dataset					
LLFormer [19]	23.65	0.816	24.5M	**22.0×1 = 22.0**	**0.092×1 = 0.092**
LLDiffusion [7]	**24.65**	0.843	-	-×30 = -	-×30 = -
Resfusion (ours)	24.63	**0.860**	**7.7M**	32.9×5 = 164.5	0.027×5 = 0.135
Raindrop Dataset					
RainDiff_{64} [5]	32.29	**0.942**	-	-×10 = -	-×10 = -
RainDiff_{128} [5]	32.43	0.933	109.7M	248.4×50 = 12420.0	-×50 = -
WeatherDiff_{64} [5]	30.71	0.931	82.9M	463.1×25 = 11577.5	0.328×25 = 8.20
WeatherDiff_{128} [5]	29.66	0.923	85.6M	261.8×50 = 13090.0	0.439×50 = 21.95
Resfusion (ours)	**32.61**	0.938	**7.7M**	**32.9×5 = 164.5**	**0.027×5 = 0.135**

A.7 Truncated schedule

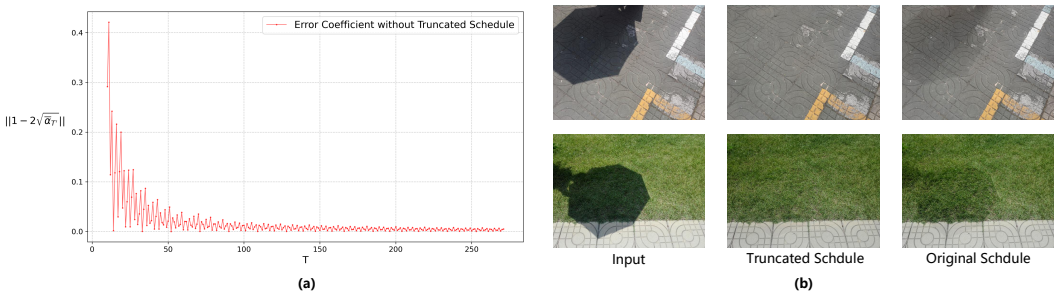

Figure 10: (a) Visualization of the relationship between the error coefficient and T. Technically, we can use the Truncated Schedule to eliminate this error when T is small. (b) Visual comparisons between Truncated Schedule and Original Schedule under five sampling steps ($T'/T = 5/12$). In terms of visual perception, the absence of Truncated Schedule will lead to residual shadows.

We observed that in the actual diffusion forward process, the noise addition steps are uniformly spaced and discrete. The discontinuity of diffusion steps implies that when we approximate the acceleration point using Eq. (8), the offset of this approximate acceleration point relative to the ideal acceleration point is unavoidable, because ensuring the existence of intersection point with no offset requires that the *gray* arrow and the *violet* arrow in Fig. 2 must be continuous. This offset actually quantifies the confidence level of the approximate equivalence $x_{T'} \approx \hat{x}_{T'}$ in Eq. (15). When T is small, the diffusion steps are divided sparsely, and the offset can be unacceptable. The absolute value of the offset can be derived as Eq. (24).

$$||x_{T'} - \hat{x}_{T'}|| = ||(2\sqrt{\bar{\alpha}_{T'}} - 1)x_0 + (1 - 2\sqrt{\bar{\alpha}_{T'}})\hat{x}_0|| = ||(1 - 2\sqrt{\bar{\alpha}_{T'}})R|| \tag{24}$$

As shown in Figure 10 (a), the absolute offset $||(1 - 2\sqrt{\bar{\alpha}_{T'}})R||$ exponentially decreases with the increase of T. When T is relatively small, this error is not negligible. However, this potential instability can be avoided in practical experiments, with a noise schedule named **Truncated Schedule** based on the existing noise schedules. In order to control the offset, we define an offset threshold h with a default value of 0.01. When decreasing $\sqrt{\bar{\alpha}_t}$, the first element less than 0.5 is denoted as $\sqrt{\bar{\alpha}_r}$. If the difference between 0.5 and $\sqrt{\bar{\alpha}_r}$ is greater than the offset threshold h, $\sqrt{\bar{\alpha}_r}$ will be reassigned to 0.5 and the following elements will be truncated from here. Since the diffusion steps after the acceleration point is not involved in the actual diffusion process, direct truncation can avoid potential risks. Taking the truncated linear schedule [29] and $T = 25$ as an example, Fig. 11 demonstrates how to achieve the acceleration point $T' = 10$. As shown in Figure 10 (b), Truncated Schedule can effectively eliminate "residual shadows".

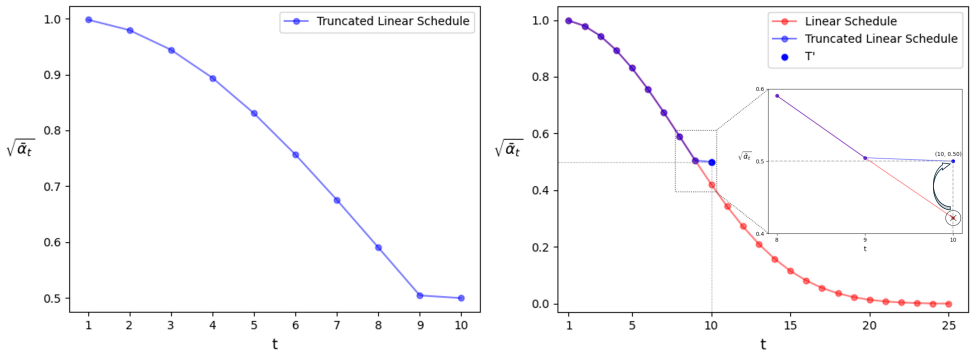

Figure 11: The schematic diagram of our truncated linear schedule. Taking $T = 25$, the left figure shows the 10 diffusion steps obtained by truncated linear schedule; the right figure shows the comparison of the truncated linear schedule and the linear schedule [29].

A.8 LOL-v2-real dataset

Table 8: Quantitative comparisons with other low-light enhancement methods on LOL-v2-real dataset. We report PSNR, SSIM and LPIPS. The best and second-best results are highlighted in **bold** and underlined. "↑" (resp. "↓") means the larger (resp. smaller), the better.

LOL-v2-real [61]			
Method	PSNR ↑	SSIM ↑	LPIPS ↓
Restormer [35]	18.69	0.834	0.232
LLFormer [19]	20.06	0.792	0.211
Resfusion (ours)	**22.06**	**0.839**	**0.175**

The LOL-v2-real dataset [61] includes visual degradations such as decreased visibility, intensive noise, and biased color. It contains 689 image pairs of both low-light and normal-light versions for training and 100 image pairs for evaluation. All experimental settings are exactly the same as the LOL dataset. As shown in Figure 12, compared to Histogram Equalization, Resfusion can significantly reduce noise, while also achieving a better color offset, demonstrating strong denoising capabilities. We provide results in terms of PSNR, SSIM, and LPIPS on LOL-v2-real dataset in Table 8.

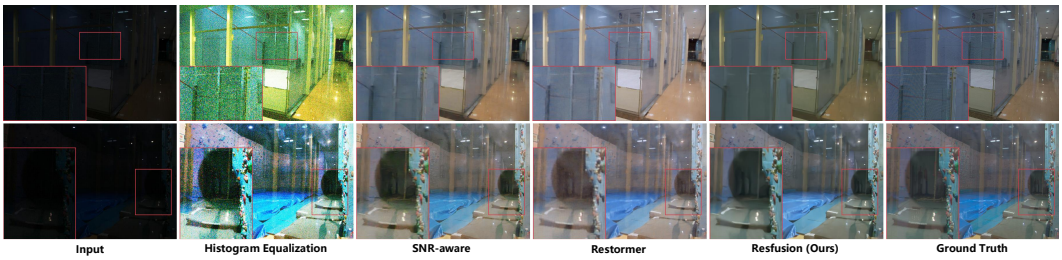

Figure 12: Visual comparisons of the restored results by different image restoration methods on the LOL-v2-real dataset.

A.9 Limitations and Future work

Task specific. Our main effort has been directed towards creating a general prototype model for image restoration and generation. This approach may lead to some performance limitations when compared to task-specific state-of-the-art methods [9, 10]. To enhance performance for particular tasks, potential strategies include employing task-specific backbones, incorporating physical prior knowledge, and utilizing customized noise schedules.

Feature fusion. In the reverse process, we simply concatenate the noisy image x_t at time step t with the conditional input $\hat{x}_0$ in the channel dimension. It is worth exploring more efficient feature fusion strategies, such as cross-attention, multi-stage, multi-scale, and multi-branch.

Latent space. The diffusion process in Resfusion is conducted in the original pixel space. Some studies [52, 62, 63] have shown that conducting diffusion process in the latent space [53, 64] can significantly reduce computational complexity while ensuring the quality of generated images, which is worth exploring in the future.

A.10 More results and failure cases

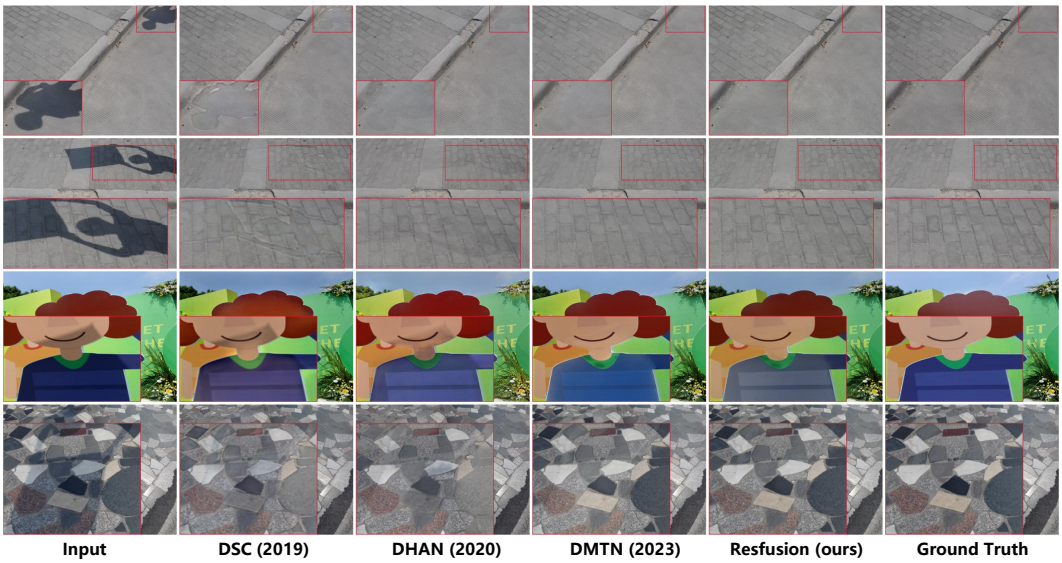

Figure 13: More visual comparisons of the restored results by different shadow-removal methods on the ISTD dataset.

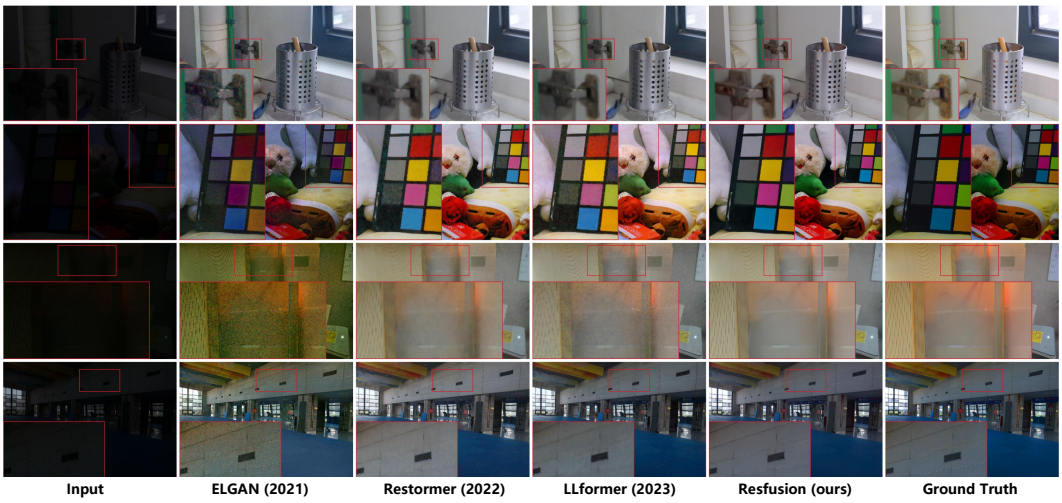

Figure 14: More visual comparisons of the restored results by different low-light enhancement methods on the LOL dataset.

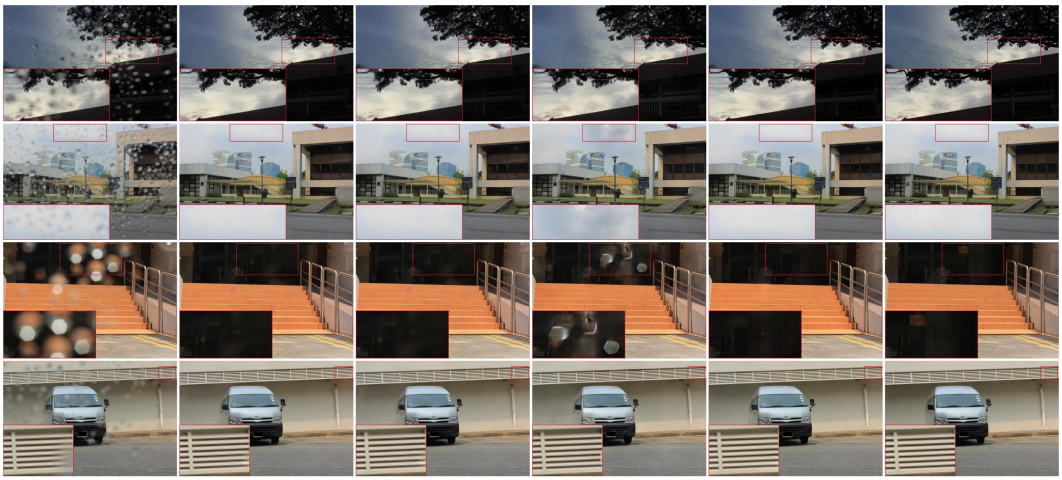

Figure 15: More visual comparisons of the restored results by different deraining methods on the Raindrop dataset.

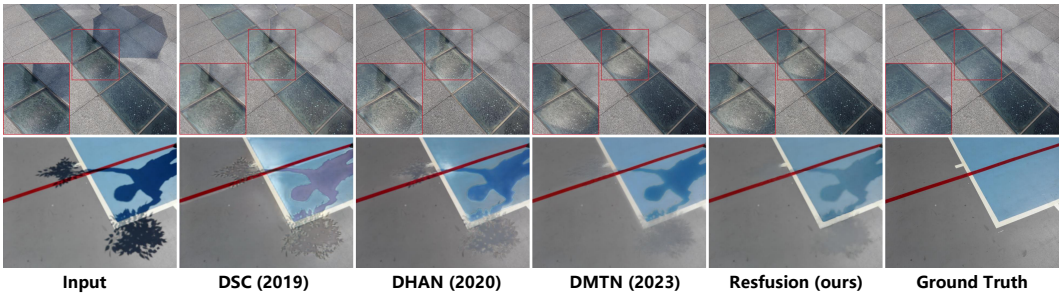

Figure 16: Visual comparisons of the restored results by different shadow-removal methods on the ISTD dataset. (failure cases)

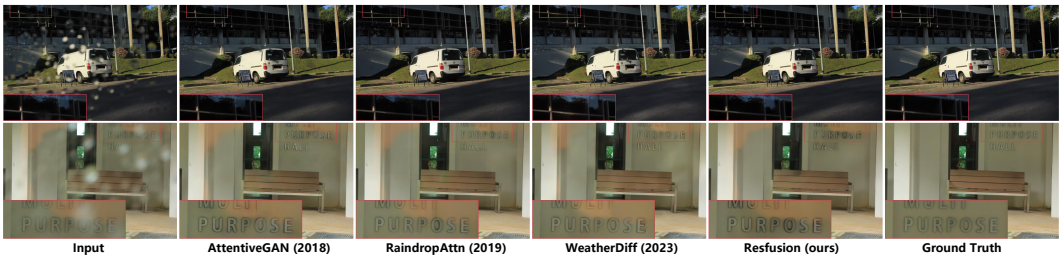

Figure 17: Visual comparisons of the restored results by different deraining methods on the Raindrop dataset. (failure cases)