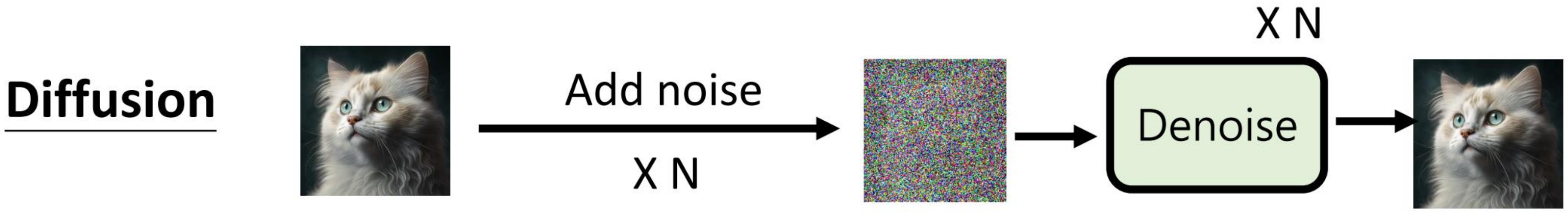
Resfusion: Denoising Diffusion Probabilistic Models for Image Restoration Based on Prior Residual Noise

Reporter: Shi, Zhenning

DDPM



$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim N(0, I_{n\times n})$$

$$x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \epsilon \sim N(0, I_{n\times n})$$

$$\mathbb{E}_{x_0,\epsilon,t}[||\epsilon - \epsilon_\theta(x_t, t)||^2]$$

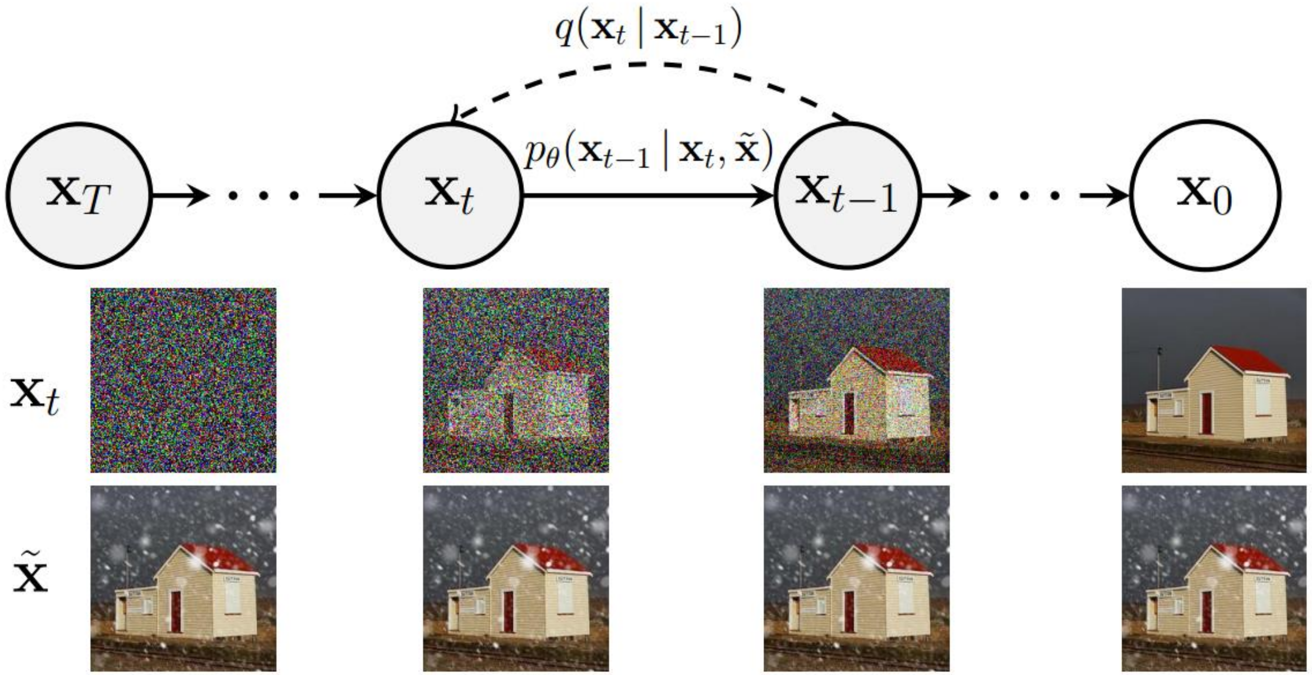
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Diffusion for Image Restoration

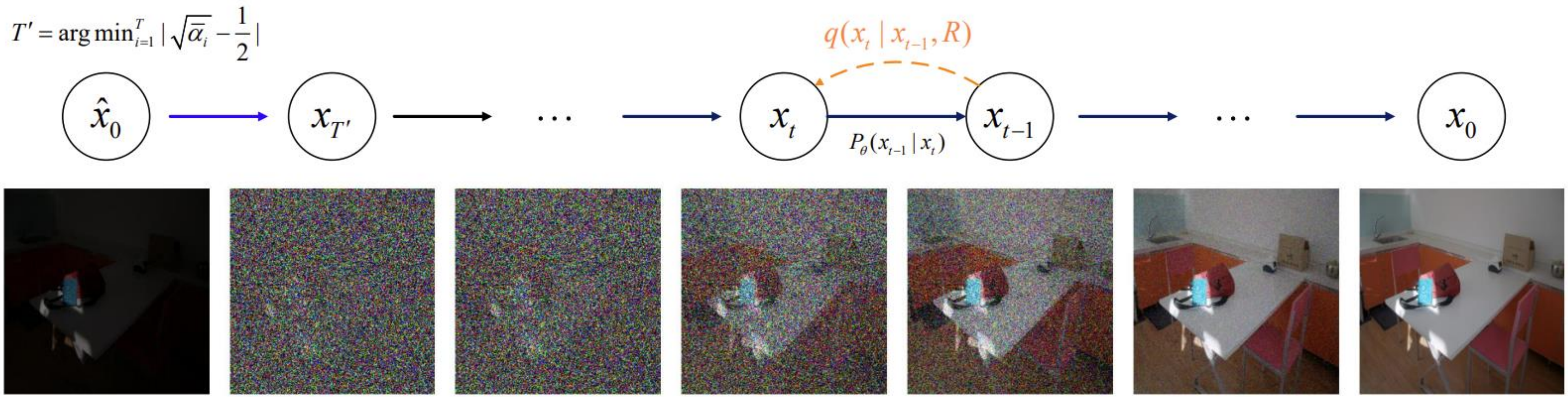


$$p_\theta(\mathbf{x}_{0:T} \mid \tilde{\mathbf{x}}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \tilde{\mathbf{x}}).$$

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \tilde{\mathbf{x}}, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \tilde{\mathbf{x}}, t),$$

From Guassian noise to noisy images

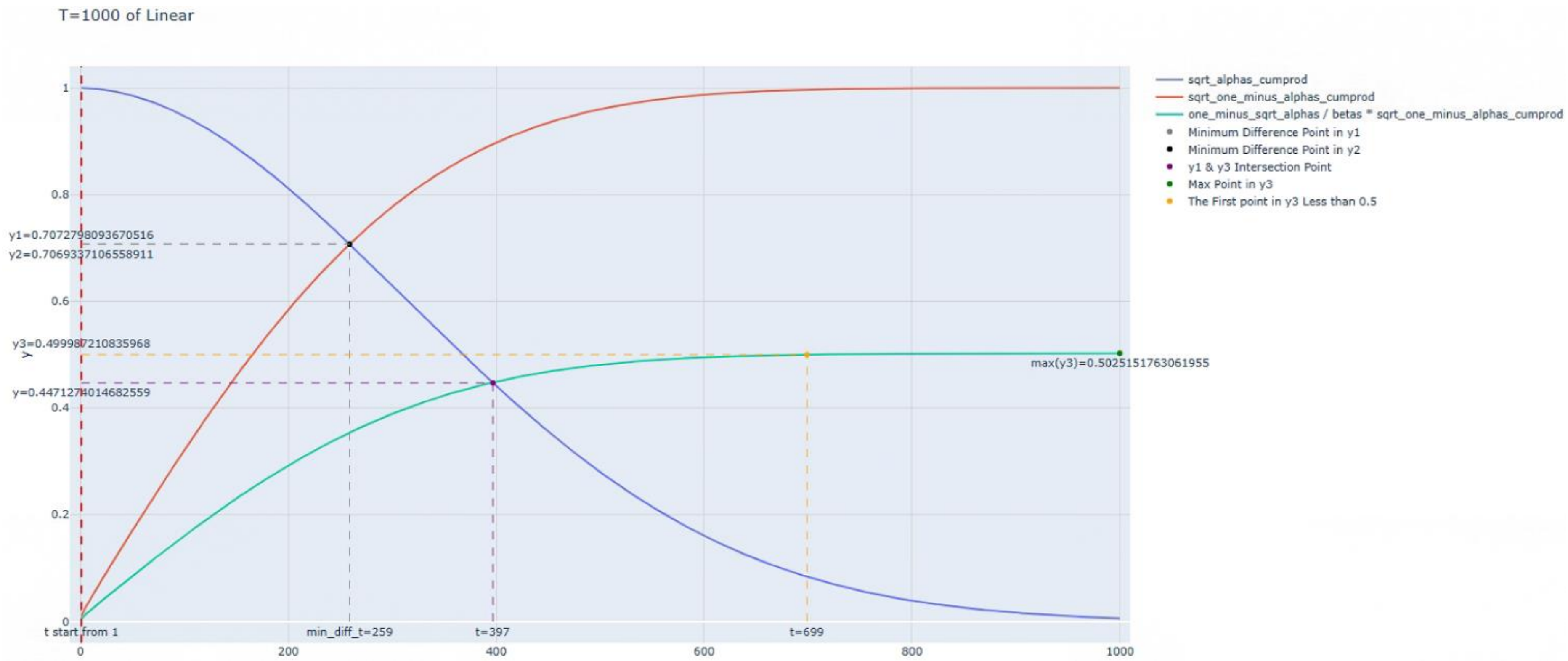
The degraded image already contains low-frequency (overall) information, so we only need to recover the high-frequency (detail) and the residual term (i.e., denoising and subtracting the residual term).



Why residual modeling work?

Diffusion models are better at modeling high-frequency (detailed) information!

The degraded image already contains low-frequency (overall) information, so we only need to recover the high-frequency (detail) and the residual term (i.e., denoising and subtracting the residual term).



Limitations of existing residual diffusion models

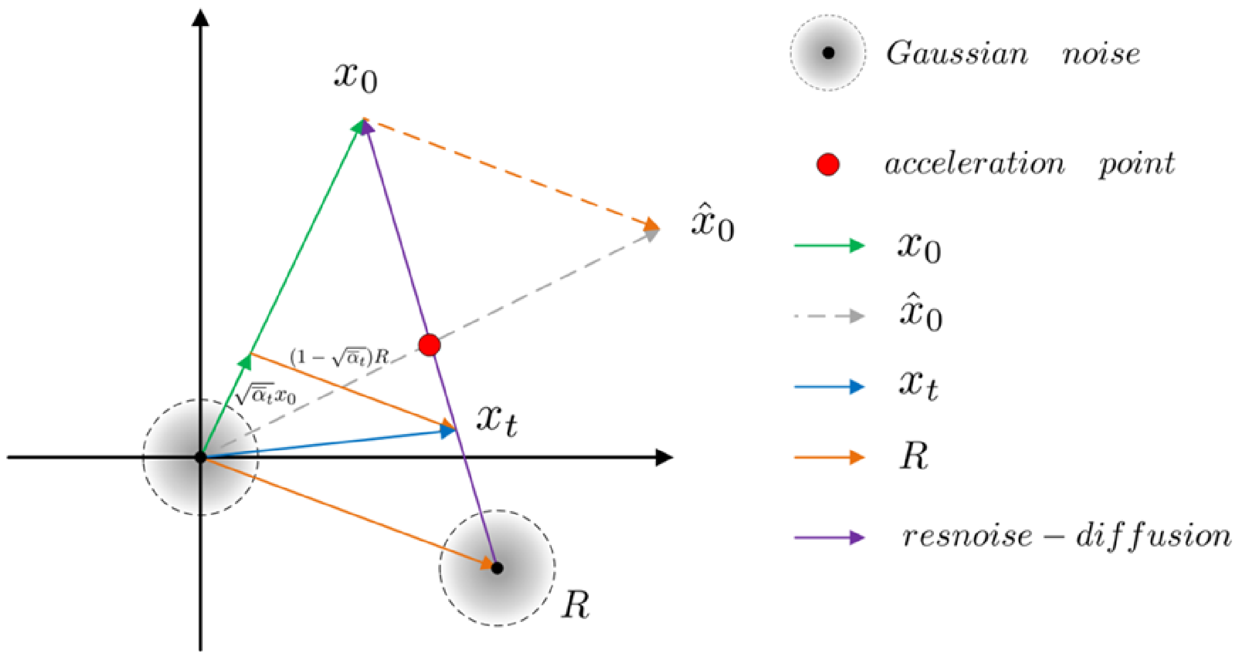
- predicting the residual term and the noise term separately, without explicitly specifying their quantitative relationship

- forward and reverse processes are inconsistent with the DDPM, which results in poor generalization and interpretability

- requires the design of a complex noise schedule

Resfusion

Motivation: If we want to introduce R into the forward process, then the starting point of its reverse process is unknown, which is a posterior distribution. We need to find an equivalent prior distribution for it. In a visual sense, this is like finding an intersection point.



Forward process:

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

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

Reverse process:

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

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

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

Algorithm

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 $x_0 = \frac{1}{\sqrt{\alpha_1}}(x_1 - \frac{\beta_1}{\sqrt{1-\overline{\alpha}_1}} res\epsilon_\theta(x_1, \hat{x}_0, 1))$

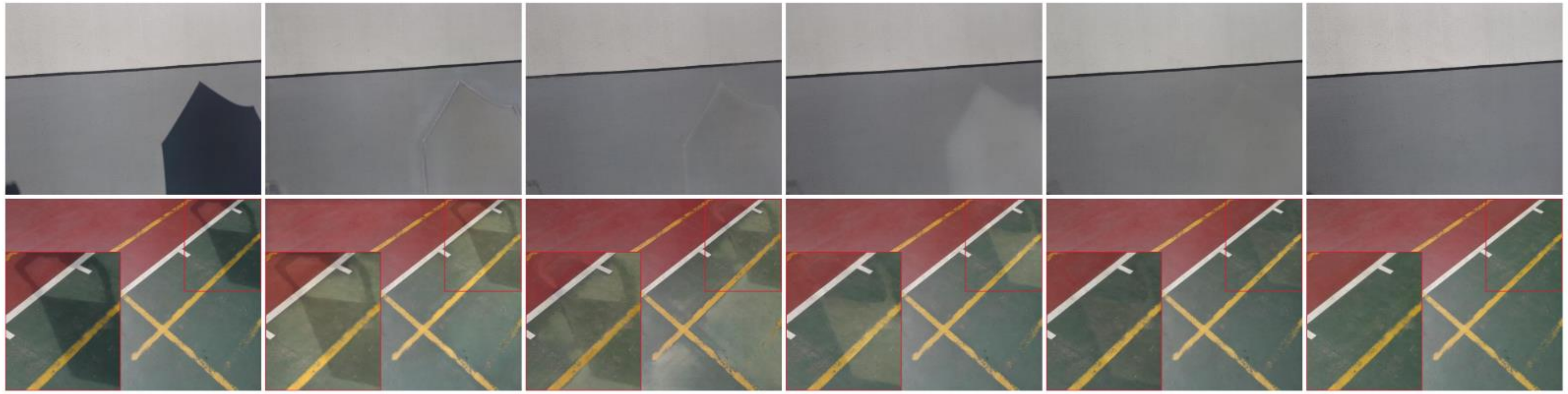
2、Experiments

$$\epsilon_{resnoise} = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} R$$

ISTD Dateset：

Table 1: Quantitative comparisons with other shadow removal methods. We report PSNR, SSIM [15] 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 [16]										
	Method	Params	Shadow Region (S) PSNR ↑	Shadow Region (S) SSIM ↑	Shadow Region (S) MAE ↓	Non-Shadow Region (NS) PSNR ↑	Non-Shadow Region (NS) SSIM ↑	Non-Shadow Region (NS) MAE ↓	All Image (ALL) PSNR ↑	All Image (ALL) SSIM ↑	All Image (ALL) 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 [17]	31.8M	33.74	0.981	9.99	29.51	0.958	6.05	27.44	0.929	6.65
	DSC [18]	22.3M	34.64	0.984	8.72	31.26	0.969	5.04	29.00	0.944	5.59
	DHAN [19]	21.8M	35.53	0.988	7.49	31.05	0.971	5.30	29.11	0.954	5.66
	FusionNet [20]	186.5M	34.71	0.975	7.91	28.61	0.880	5.51	27.19	0.945	5.88
	UnfoldingNet [21]	10.1M	36.95	0.987	8.29	31.54	0.978	4.55	29.85	0.960	5.09
	DMTN [22]	22.8M	35.83	0.990	7.00	33.01	0.979	4.28	30.42	0.965	4.72
	RDDM (SM-Res-N) [14]	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 [23]	-	-	-	9.21	-	-	6.27	-	-	6.63
	DHAN [19]	21.8M	34.79	0.983	8.13	29.54	0.941	5.94	27.88	0.921	6.29
	CANet [24]	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**



Input **DSC (2019)** **DHAN (2020)** **DMTN (2023)** **Resfusion (ours)** **Ground Truth**

LOL Dateset：

$$\epsilon_{resnoise} = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} R$$

Table 2: Quantitative comparisons with other low-light enhancement methods. We report PSNR, SSIM and LPIPS [34]. The best and second-best results are highlighted in **bold** and underlined. “↑" (resp. “↓") means the larger (resp. smaller), the better.

LOL [27]				
Method	Params	PSNR ↑	SSIM ↑	LPIPS ↓
YCbCr space, 256 × 256				
Input Image	-	9.30	0.377	0.513
RDDM (SM-Res-N) [14]	15.5M	23.90	0.931	-
RDDM (SM-Res) [14]	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 [27]	0.6M	16.77	0.560	0.474
KinD [28]	8.0M	20.87	0.790	0.170
KinD++ [29]	9.6M	21.30	0.820	0.160
Zero-DCE [30]	**0.3M**	14.86	0.562	0.335
EnlightenGAN [31]	8.6M	17.48	0.652	0.322
Restormer [32]	-	22.37	0.816	0.141
LLFormer [16]	24.6M	23.65	0.816	0.169
Resfusion (ours)	7.7M	**24.63**	**0.860**	**0.107**

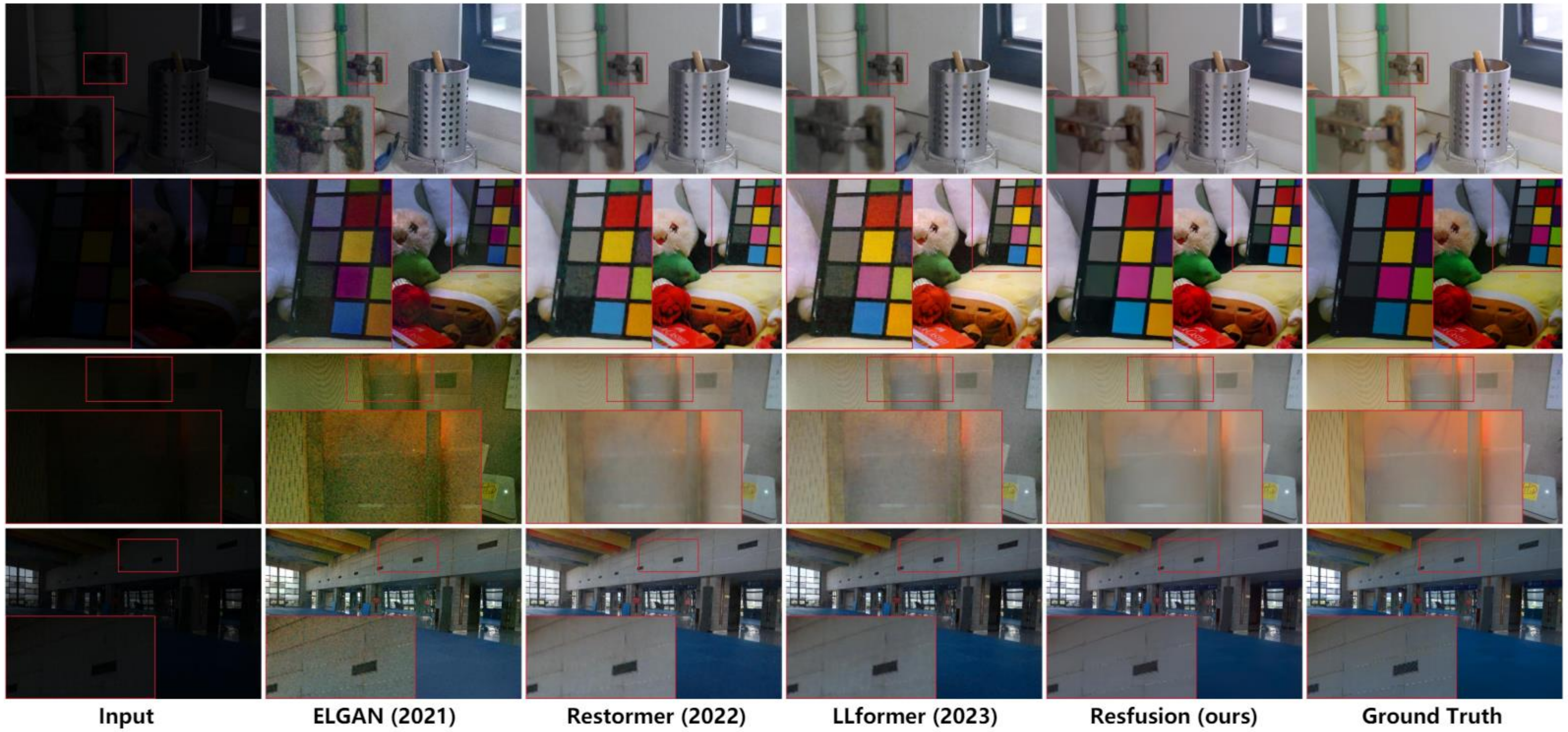


Figure 11: More visual comparisons of the restored results by different low-light enhancement methods on the LOL dataset.

Raindrop Dateset：

$$\epsilon_{resnoise} = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\bar{\alpha}_t}}{\beta_t} R$$

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 [35]			
Method	Params	PSNR ↑	SSIM ↑
YCbCr space, 256 × 256			
Input Image	-	25.81	0.887
RDDM (SM-Res-N) [14]	15.5M	32.51	0.956
Resfusion (ours)	**7.7M**	**34.40**	**0.975**
YCbCr space, Original			
Input Image	-	25.40	0.882
pix2pix [36]	-	28.02	0.855
AttentiveGAN [35]	**6.2M**	31.59	0.917
DuRN [37]	10.2M	31.24	0.926
RaindropAttn [38]	-	31.44	0.926
All-in-One [39]	-	31.12	0.927
IDT [40]	16.4M	31.87	0.931
$WeatherDiff_{64}$ [4]	82.9M	30.71	0.931
$RainDropDiff_{128}$ [4]	109.7M	32.43	0.933
Resfusion (ours)	7.7M	**32.61**	**0.938**

Figure 12: More visual comparisons of the restored results by different deraining methods on the Raindrop dataset.

resfusion can learn some pattern

The analysis of the residual term and the noise term

$$\epsilon_{resnoise} = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} R$$

The residual is responsible for the shift/generation of low-frequency semantic information, while the noise is responsible for the generation of high-frequency detail information!!!

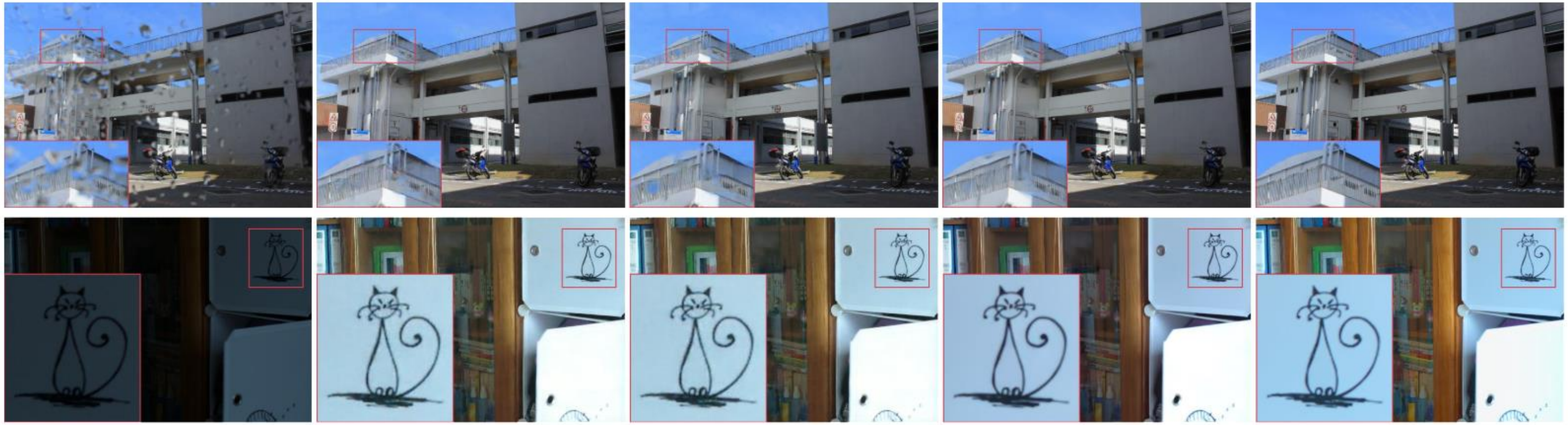
Input **Only Remove Noise** **Only Remove Residual** **Remove Resnoise** **Ground Truth**

Why resnoise?

$$\epsilon_{resnoise} = \epsilon + \frac{(1-\sqrt{\alpha_t})\sqrt{1-\overline{\alpha}_t}}{\beta_t} R$$

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 underlined. “↑" (resp. “↓") means the larger (resp. smaller), the better.

Prediction Targets	ISTD [16]			LOL [27]			RainDrop [35]	
	PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
$x_{0\theta}$	29.67	0.927	5.35	23.10	0.813	0.150	32.51	0.935
R_θ	29.75	0.930	**5.26**	22.87	0.807	0.143	32.57	0.935
$res\epsilon_\theta$	**30.09**	**0.932**	5.34	**24.63**	**0.860**	**0.107**	**32.61**	**0.938**



Input $x_{0\theta}$ R_θ $res\epsilon_\theta$ **Ground Truth**

Resource efficiency

10× fewer parameters, 10× fewer sampling steps, and 50× fewer MACs !!!

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 <u>underlined</u>. "↑" (resp. "↓") means the larger (resp. smaller), the better. We use the symbol "-" to indicate models or results that are unavailable.

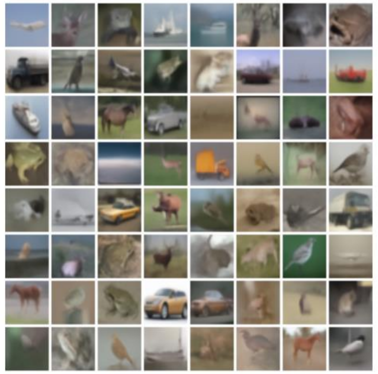
Methods	PSNR ↑	SSIM ↑	Params ↓	MACs(G) × Steps ↓	Inference Time (s) ↓
			ISTD Dataset		
Shadow Diffusion [9]	**32.33**	**0.969**	<u>55.5M</u>	<u>182.1×25 = 4552.5</u>	<u>0.024×25 = 0.600</u>
SR3 [3]	27.49	0.871	155.3M	155.3×100 = 15530.0	-×100 = -
Resfusion (ours)	<u>31.81</u>	<u>0.965</u>	**7.7M**	**33.3×5 = 167.5**	**0.027×5 = 0.135**
			LOL Dataset		
LLFormer [17]	23.65	0.816	<u>24.5M</u>	**22.0×1 = 22.0**	**0.092×1 = 0.092**
LLDiffusion [7]	**24.65**	<u>0.843</u>	-	-×30 = -	-×30 = -
Resfusion (ours)	<u>24.63</u>	**0.860**	**7.7M**	<u>32.9×5 = 164.5</u>	<u>0.027×5 = 0.135</u>
			Raindrop Dataset		
RainDiff_{64} [5]	32.29	**0.942**	-	-×10 = -	-×10 = -
RainDiff_{128} [5]	<u>32.43</u>	0.933	109.7M	248.4×50 = 12420.0	-×50 = -
WeatherDiff_{64} [5]	30.71	0.931	<u>82.9M</u>	<u>463.1×25 = 11577.5</u>	<u>0.328×25 = 8.20</u>
WeatherDiff_{128} [5]	29.66	0.923	85.6M	261.8×50 = 13090.0	0.439×50 = 21.95
Resfusion (ours)	**32.61**	<u>0.938</u>	**7.7M**	**32.9×5 = 164.5**	**0.027×5 = 0.135**

Generation $R = -x_0$

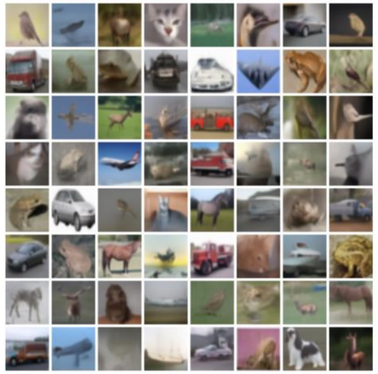
CIFAR10:

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

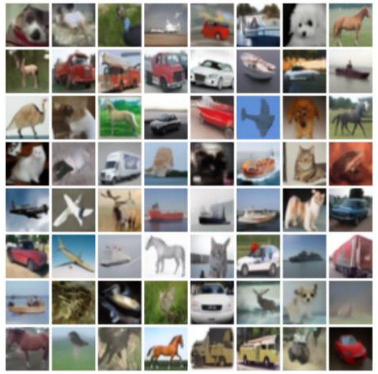
DDPM



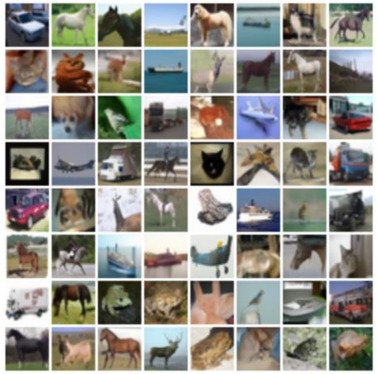
10 steps



20 steps

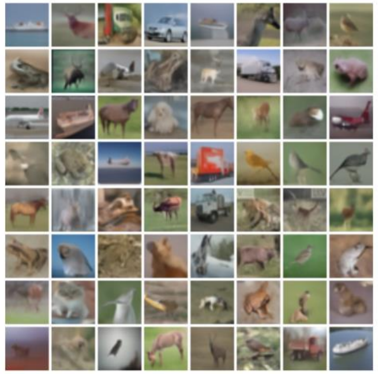


50 steps

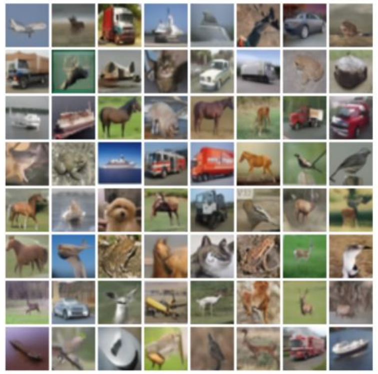


100 steps

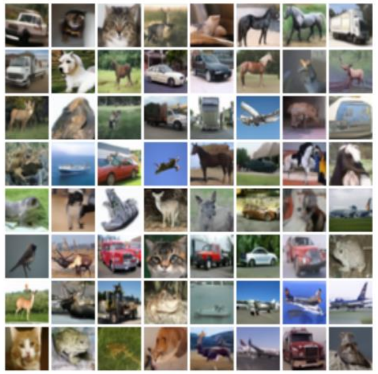
Resfusion



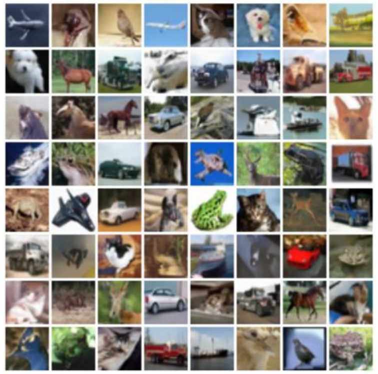
10 steps



20 steps



50 steps



100 steps

Translation

Figure 8: 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".

Visualization

Thank you.