DiffIR: Efficient Diffusion Model for Image Restoration

ICCV 2023 Poster

Bin Xia[1], Yulun Zhang[2], Shiyin Wang[3], Yitong Wang[3], Xinglong Wu[3], Yapeng Tian[4], Wenming Yang[1], and Luc Van Gool[2]

[1]Tsinghua University, [2]ETH, [3]ByteDance Inc., [4]University of Texas at Dallas

马逸扬
2024/03/10

Content

- Authors
- Background
- Method
- Experiments

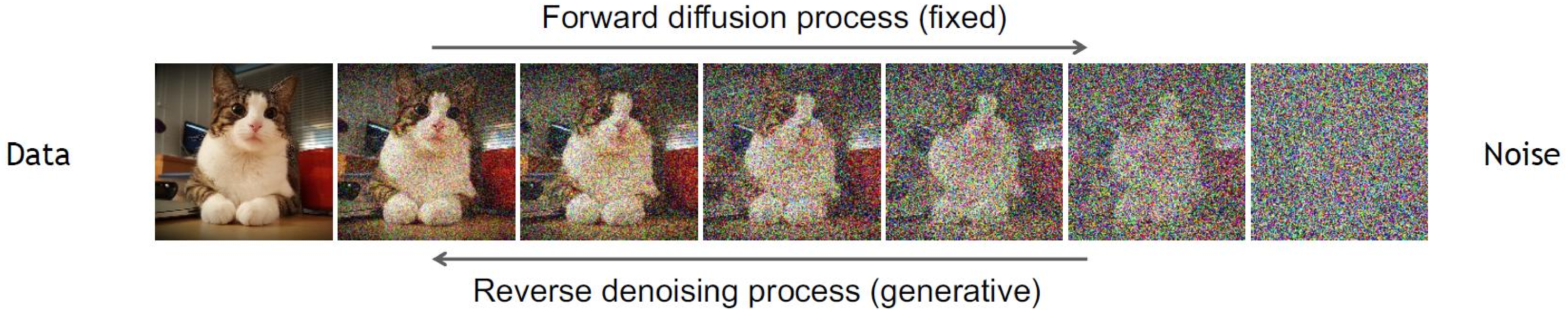
Content

- Authors
- Background
- Method
- Experiments

Background

DDPMs & Score-based models: two perspectives.

Denoising diffusion probabilistic models (DDPMs), NIPS 20':



$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})) \quad (1)$$

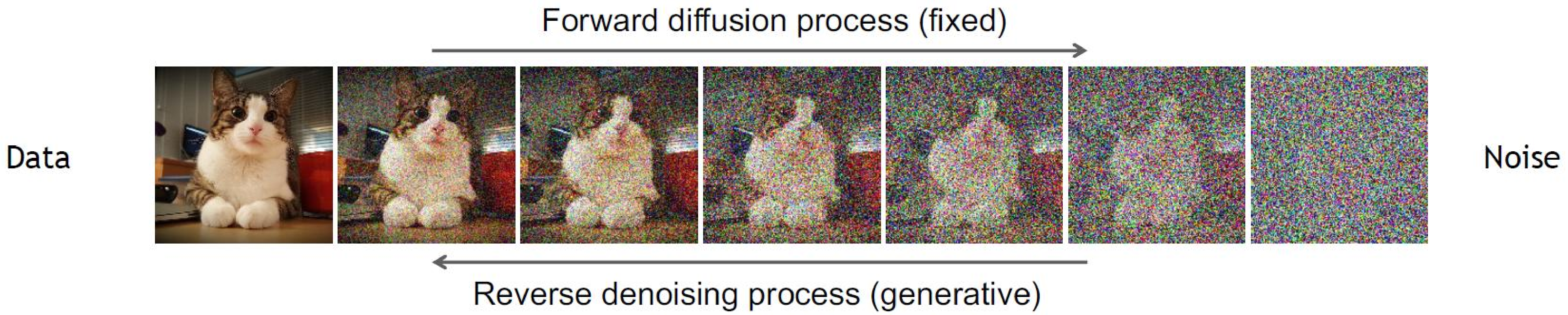
$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \quad (2)$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \quad (3)$$

Background

DDPMs & Score-based models: two perspectives.

Denoising diffusion probabilistic models (DDPMs), NIPS 20':



$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \left[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, t)||^2 \right] \quad (4)$$

Background

DDPMs & Score-based models: two perspectives.

Generative Modeling by Estimating Gradients of the Data Distribution (NCSN), ICLR 19':

Score matching:

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})||_2^2] \tag{5}$$

Why?

Considering a parameterized distribution:

$$p(x;\theta) = \frac{1}{Z(\theta)} q(x;\theta) \tag{6}$$

$$Z(\theta) = \int_{x\in\mathbb{R}^n} q(x;\theta)dx \tag{7}$$

The MLE target is:

$$\theta_{mle} = \arg\max_\theta \sum_{t=1}^{T} \log p(x_t;\theta) \tag{8}$$

Background

DDPMs & Score-based models: two perspectives.

Generative Modeling by Estimating Gradients of the Data Distribution, ICLR 19':

$$\theta_{mle} = \arg\max_\theta \sum_{t=1}^{T} \log p(x_t;\theta) \tag{8}$$

However, such a target is difficult to learn due to $Z(\theta)$. We notice that:

$$\nabla_x \log p(x;\theta) = \nabla_x [\log q(x;\theta) - logZ(\theta)] \tag{9}$$

Thus, the training target becomes:

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})||_2^2] \tag{5}$$

The sampling process can be:

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_{\mathbf{x}} \log p(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\,\mathbf{z}_t \tag{10}$$

How to compute $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})||_2^2$?

Background

DDPMs & Score-based models: two perspectives.

Generative Modeling by Estimating Gradients of the Data Distribution, ICLR 19':

How to compute $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2$?

We perturb the sample by:

$$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I) \tag{11}$$

And we approximate the perturbed distribution:

$$q_\sigma(\tilde{\mathbf{x}}) \triangleq \int q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) p_{\text{data}}(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{12}$$

Through:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\|_2^2] \tag{13}$$

Background

DDPMs & Score-based models: two perspectives.

Generative Modeling by Estimating Gradients of the Data Distribution, ICLR 19':

After the training the $q_\sigma(\tilde{\mathbf{x}})$ on a set of different σ, we sample by:

- Sample $\mathbf{x}_0$ from $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I)$ with a large σ_0
- Repeat:
  - Start from $\mathbf{x}_{t-1}$ within $q_{\sigma_{t-1}}$, repeat (10), until get $\mathbf{x}_t$ from q_{σ_t}
- Until σ_T is 0, where $\tilde{\mathbf{x}}$ is equal to $\mathbf{x}$.

What's the loss? Recall that $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I)$, we have:

$$\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -(\tilde{\mathbf{x}}-\mathbf{x})/\sigma^2 \tag{11}$$

Thus:

$$\ell(\boldsymbol{\theta}; \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}, \sigma^2 I)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2}\right\|_2^2\right] \tag{12}$$

Background

Two perspectives are both **vital**.

DDPM is more simple and intuitive.

NCSN is more fundamental and easy to extend:

Classifier guidance:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}(t) \mid \mathbf{y}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}(t)) + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x}(t)) \tag{13}$$

Classifier-free guidance:

$$\nabla_{\mathbf{z}_\lambda} \log p^i(\mathbf{c}|\mathbf{z}_\lambda) = -\frac{1}{\sigma_\lambda}[\boldsymbol{\epsilon}^*(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon}^*(\mathbf{z}_\lambda)] \tag{14}$$

(and my previous diffusion-based compression paper)

are both derived from NCSN.

Content

- Authors
- Background
- Method
- Experiments

Method

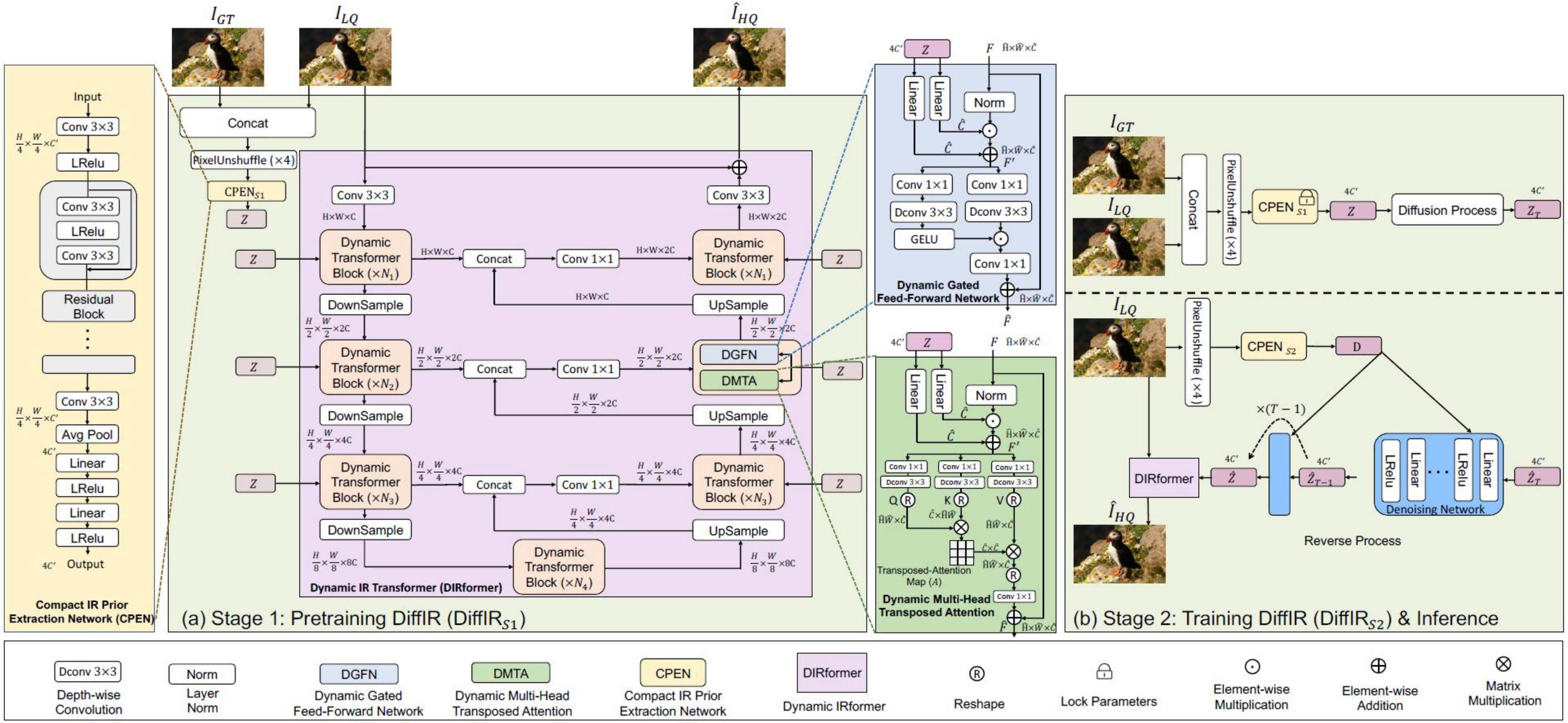
Employ diffusion models in image super-resolution.

Challenges of generating SR images directly via diffusion models:

- Training the models
- Sampling from the models

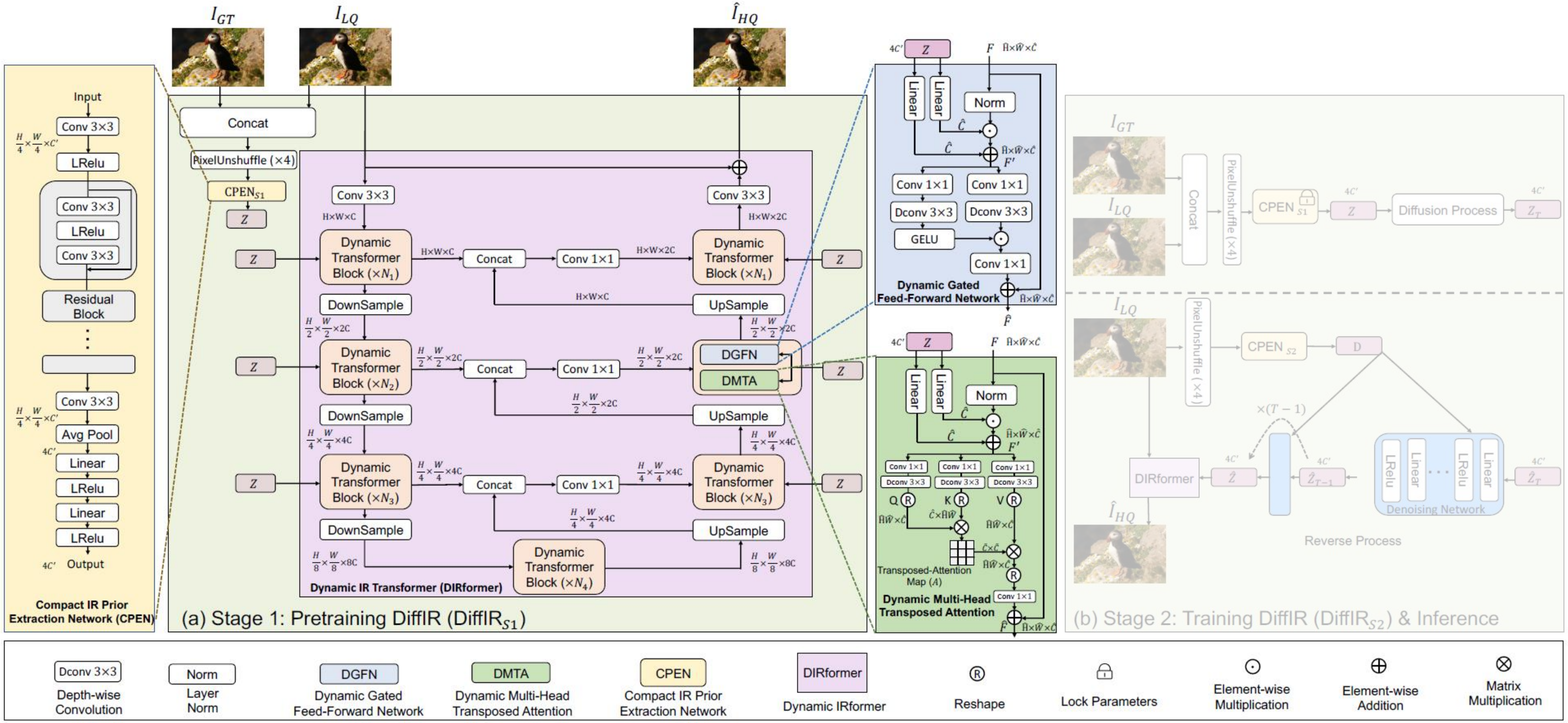
Approach: only employ diffusion models as a guidance to create details.

Method



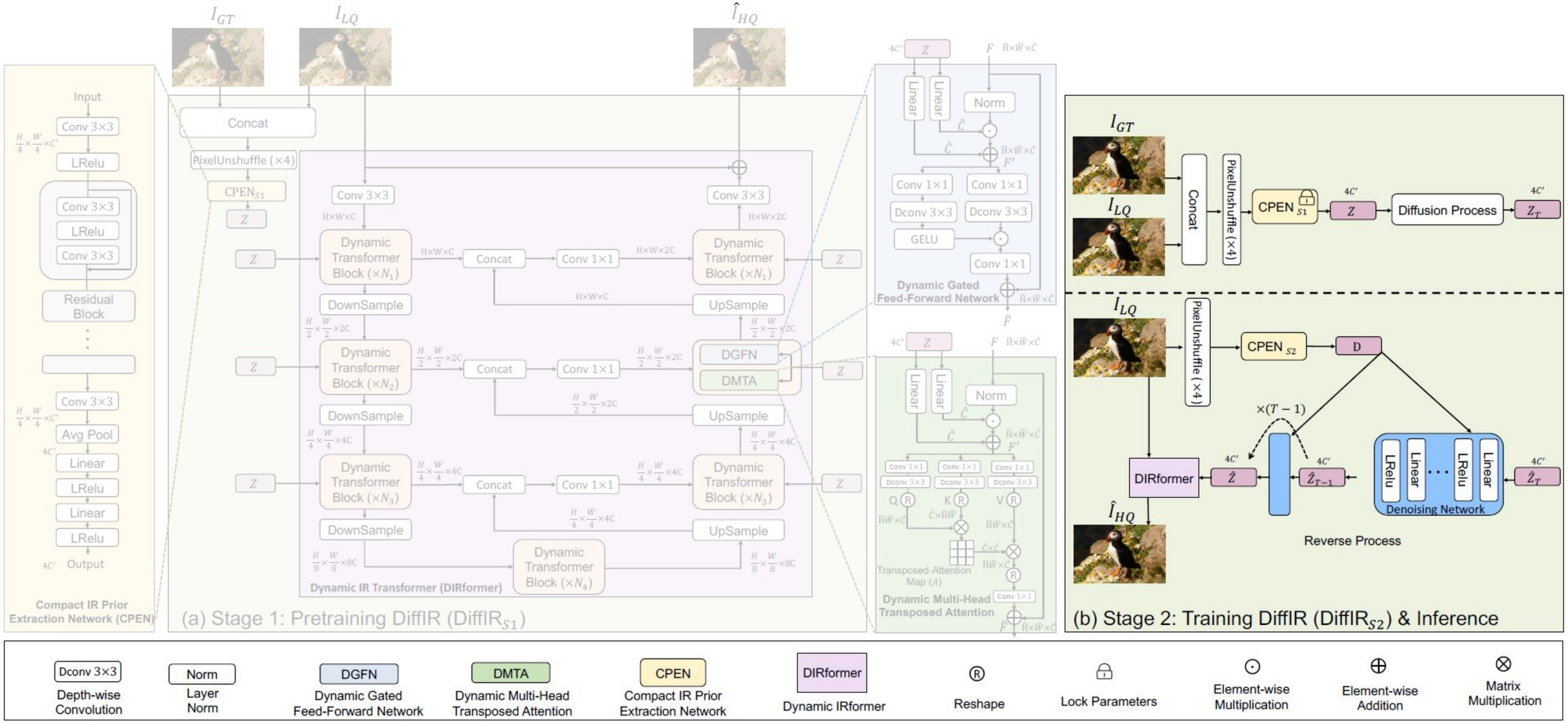
Two stages.

Method



Stage1: Structural reconstruction module.

Method



Stage2: Diffusion-based guidance module

Content

- Authors
- Background
- Method
- Experiments

Experiments

The framework can be employed in different low-level vision tasks.

Image inpainting:

Method	#Params (M)	Places [88] (512×512)				CelebA-HQ [27] (256×256)			
		Narrow Masks		Wide Masks		Narrow Masks		Wide Masks	
		FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
EdgeConnect [46]	22	1.3421	0.1106	8.4866	0.1594	6.9566	0.0922	7.8346	0.1149
ICT [61]	150	-	-	-	-	8.4977	0.0982	9.8794	0.1196
LaMa [57]	27	0.6340	0.0898	2.2494	0.1339	5.3889	0.0806	5.7023	0.0951
LDM [50]	215	-	-	2.1500	0.1440	-	-	-	-
RePaint [40]	607	-	-	-	-	4.7395	0.0890	5.4881	0.1094
DiffIR$_{S2}$ (Ours)	26	**0.4913**	**0.0758**	**1.9788**	**0.1306**	**4.5967**	**0.0769**	**5.1440**	**0.0918**

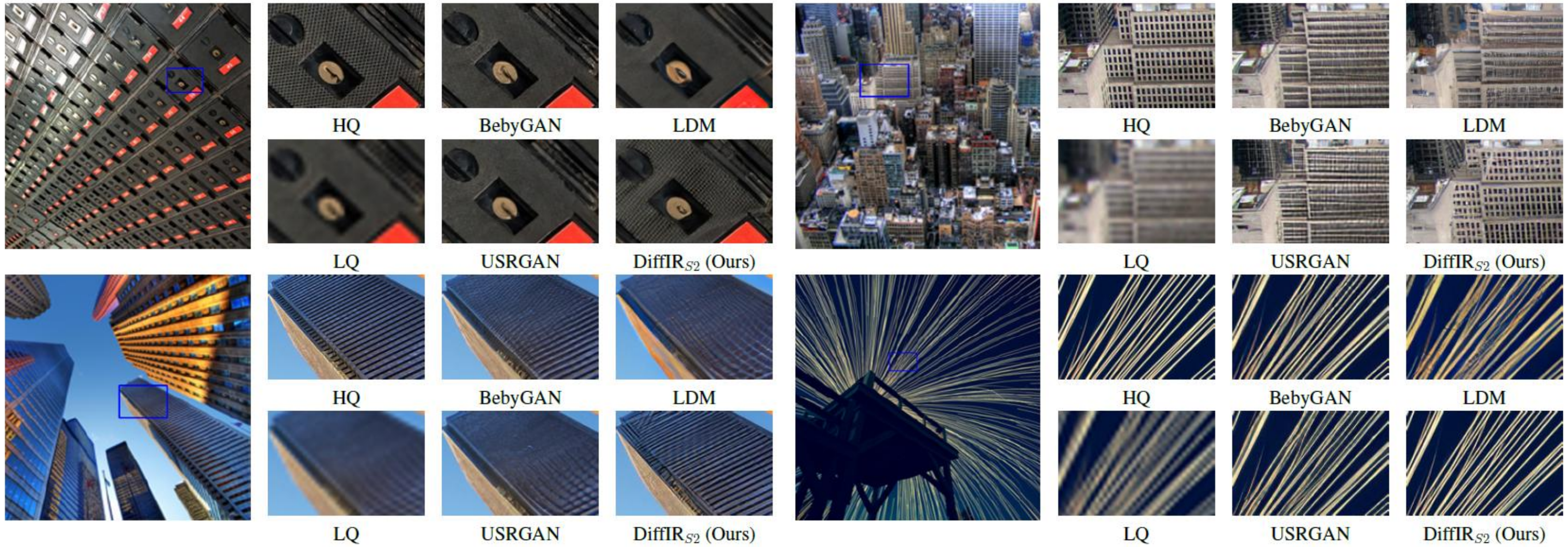
HQ | LQ | ICT [61] | LaMa [57] | RePaint [40] | DiffIR$_{S2}$ (Ours)

Experiments

The framework can be employed in different low-level vision tasks.

Image super-resolution:

Method	Set14 [77]		Urban100 [25]		Manga109 [43]		General100 [16]		DIV2K100 [1]	
	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
SFTGAN [64]	26.74	0.1313	24.34	0.1343	28.17	0.0716	29.16	0.0947	28.09	0.1331
SRGAN [34]	26.84	0.1327	24.41	0.1439	28.11	0.0707	29.33	0.0964	28.17	0.1257
ESRGAN [65]	26.59	0.1241	24.37	0.1229	28.41	0.0649	29.43	0.0879	28.18	0.1154
USRGAN [80]	27.41	0.1347	24.89	0.1330	28.75	0.0630	30.00	0.0937	28.79	0.1325
SPSR [42]	26.86	0.1207	24.80	0.1184	28.56	0.0672	29.42	0.0862	28.18	0.1099
BebyGAN [37]	27.09	0.1157	25.23	0.1096	29.19	0.0529	29.95	0.0778	28.62	0.1022
LDM [50]	25.62	0.2034	23.36	0.1816	25.87	0.1321	27.17	0.1655	26.66	0.1939
SRdiff [35]	27.14	0.1450	25.12	0.1379	28.67	0.0665	29.83	0.1009	28.58	0.1293
DiffIR$_{S2}$ (Ours)	**27.73**	**0.1117**	**26.05**	**0.1007**	**30.32**	**0.0463**	**30.58**	**0.0717**	**29.13**	**0.0871**



Experiments

The framework can be employed in different low-level vision tasks.

Image de-motion-blur:

Method	GoPro [45]		HIDE [53]	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Xu *et al.* [70]	21.00	0.741	-	-
DeblurGAN [32]	28.70	0.858	24.51	0.871
Nah *et al.* [45]	29.08	0.914	25.73	0.874
Zhang *et al.* [79]	29.19	0.931	-	-
DeblurGAN-v2 [33]	29.55	0.934	26.61	0.875
SRN [58]	30.26	0.934	28.36	0.915
Gao *et al.* [20]	30.90	0.935	29.11	0.913
DBGAN [83]	31.10	0.942	28.94	0.915
MT-RNN [47]	31.15	0.945	29.15	0.918
DMPHN [78]	31.20	0.940	29.09	0.924
Suin *et al.* [56]	31.85	0.948	29.98	0.930
MIMO-Unet+ [9]	32.45	0.957	29.99	0.930
IPT [7]	32.52	-	-	-
MPRNet [75]	32.66	0.959	30.96	0.939
Restormer [74]	<u>32.92</u>	<u>0.961</u>	<u>31.22</u>	<u>0.942</u>
DiffIR$_{S2}$ (Ours)	**33.20**	**0.963**	**31.55**	**0.947**

Experiments

The core ablation: how does the diffusion module affect the performances?

Method	Mult-Adds (G)	GT	DM	Training Schemes: Traditional DM Optimization	Training Schemes: Joint Optimization	Inserting Noise	CelebA-HQ
DiffIR$_{S1}$	47.97	✓	✗	✗	✗	✗	4.8045
DiffIR$_{S2}$-V1	51.63	✗	✗	✗	✗	✗	5.6782
DiffIR$_{S2}$-V2	51.63	✗	✓	✓	✗	✗	5.9766
DiffIR$_{S2}$-V3 (Ours)	51.63	✗	✓	✗	✓	✗	5.1440
DiffIR$_{S2}$-V4	51.63	✗	✓	✗	✓	✓	5.1937

Diffusion-based methods can be less rigorous but well-performed.

The idea of the paper can be inspiring.

Thanks for watching.

马逸扬
2024/03/10