Generative Diffusion Prior for Unified Image Restoration and Enhancement

Ben Fei , Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo ,
Bo Zhang , Bo Dai

CVPR2023

STRUCT Group Seminar
Presenter: Yifan Li
2024.1.21

Outline

- Background
- Method
- Experiments
- Conclusion

Background

Unified image restoration

Background

Restoration methods

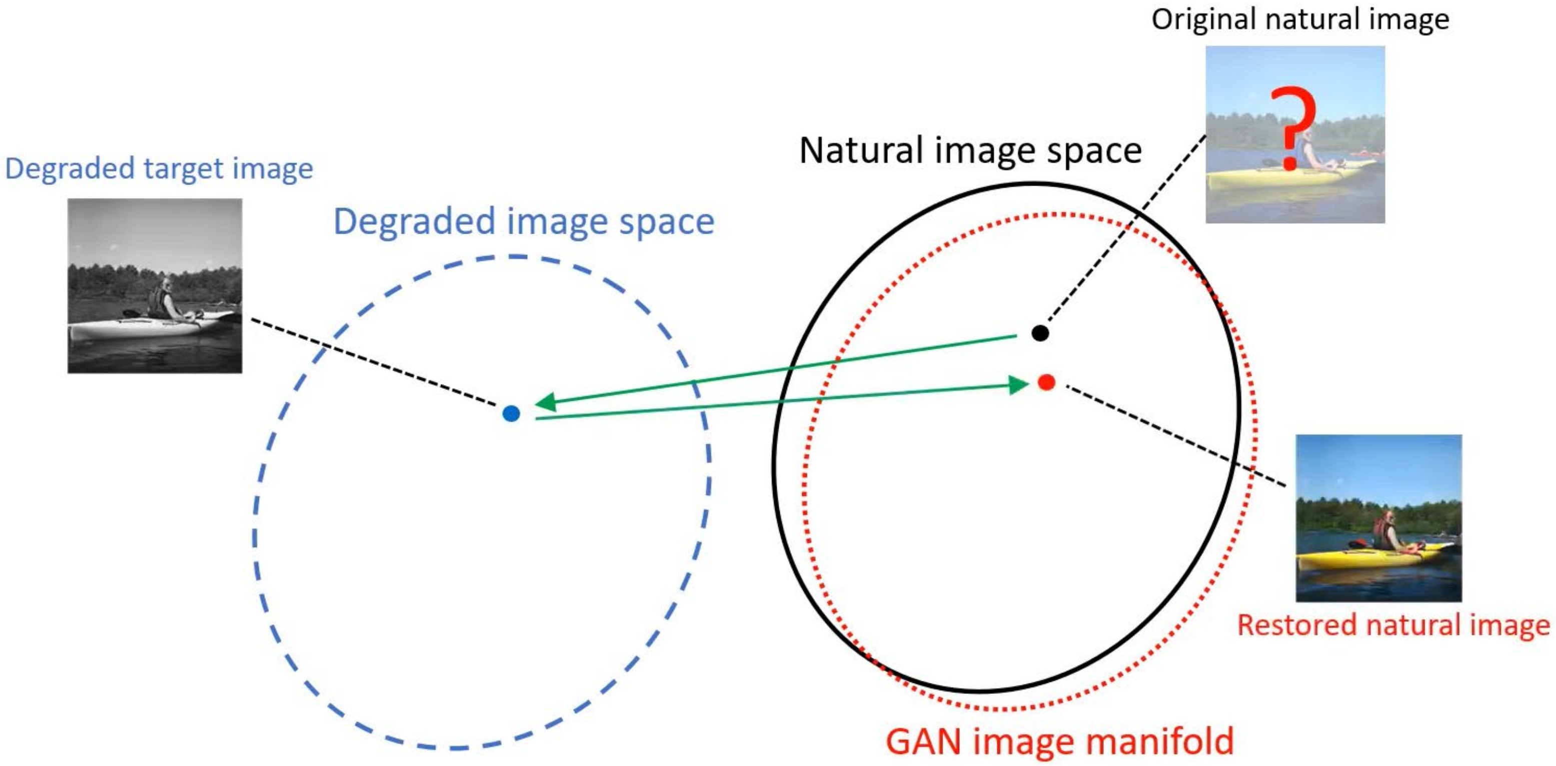
- Supervised training of neural networks
  - suffer to generalize with multiple complex degradation

- Unsupervised generative prior
  - GAN
  - DDPM

  } Learn rich knowledge of real-world scenes

Background: DGP

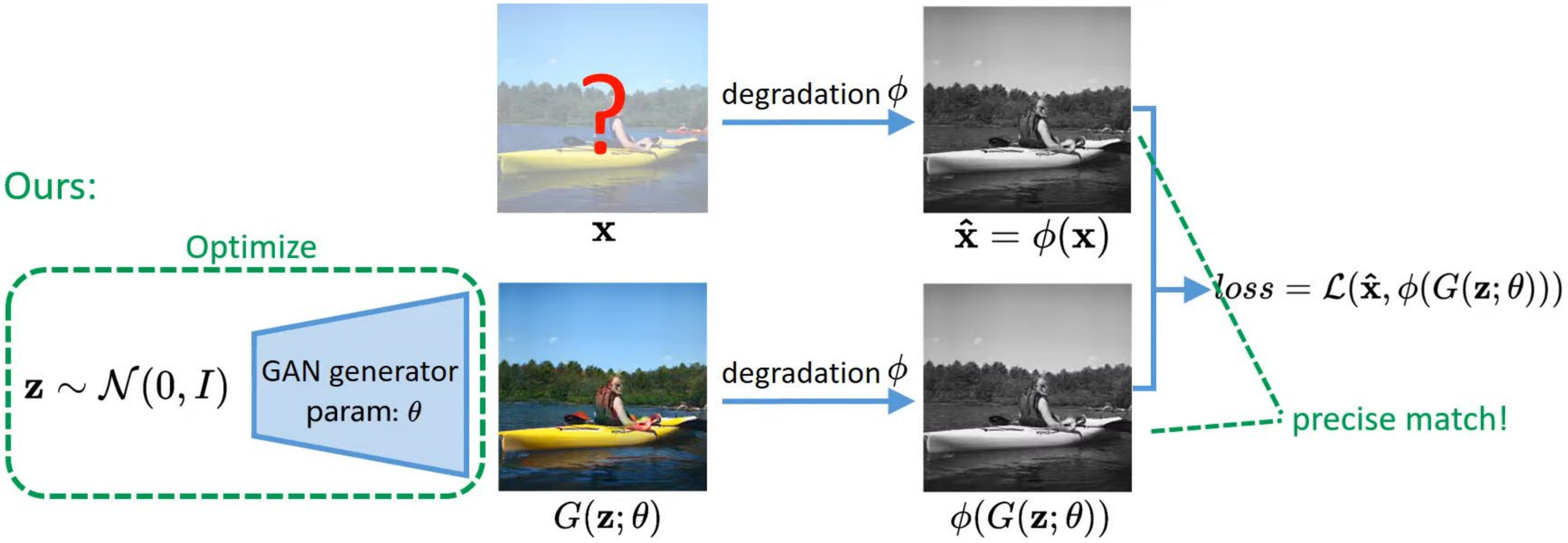
Motivation: exploit generic image prior of GAN



Exploiting deep generative prior for versatile image restoration and manipulation, Xingang Pan et al., ECCV2020 oral

Background: DGP

Degradation Alignment

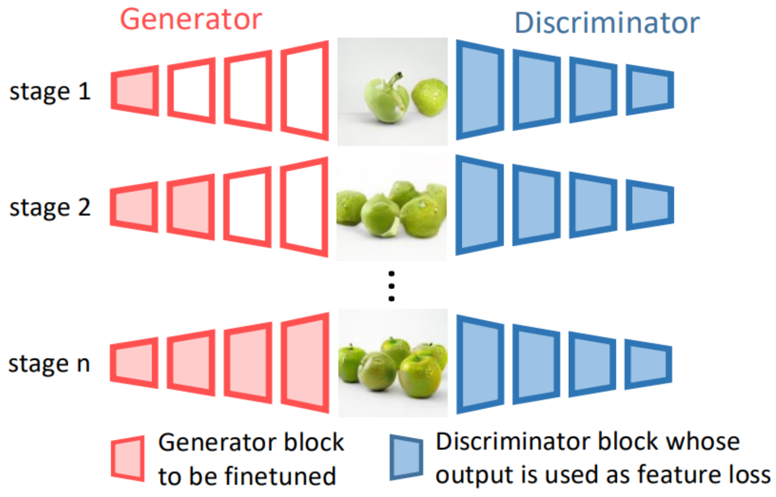


$$\theta^*, \mathbf{z}^* = argmin_{\theta,\mathbf{z}} \mathcal{L}(\hat{\mathbf{x}}, \phi(G(\mathbf{z};\theta)))$$ (Relaxed GAN-inversion)

Exploiting deep generative prior for versatile image restoration and manipulation, Xingang Pan et al., ECCV2020 oral

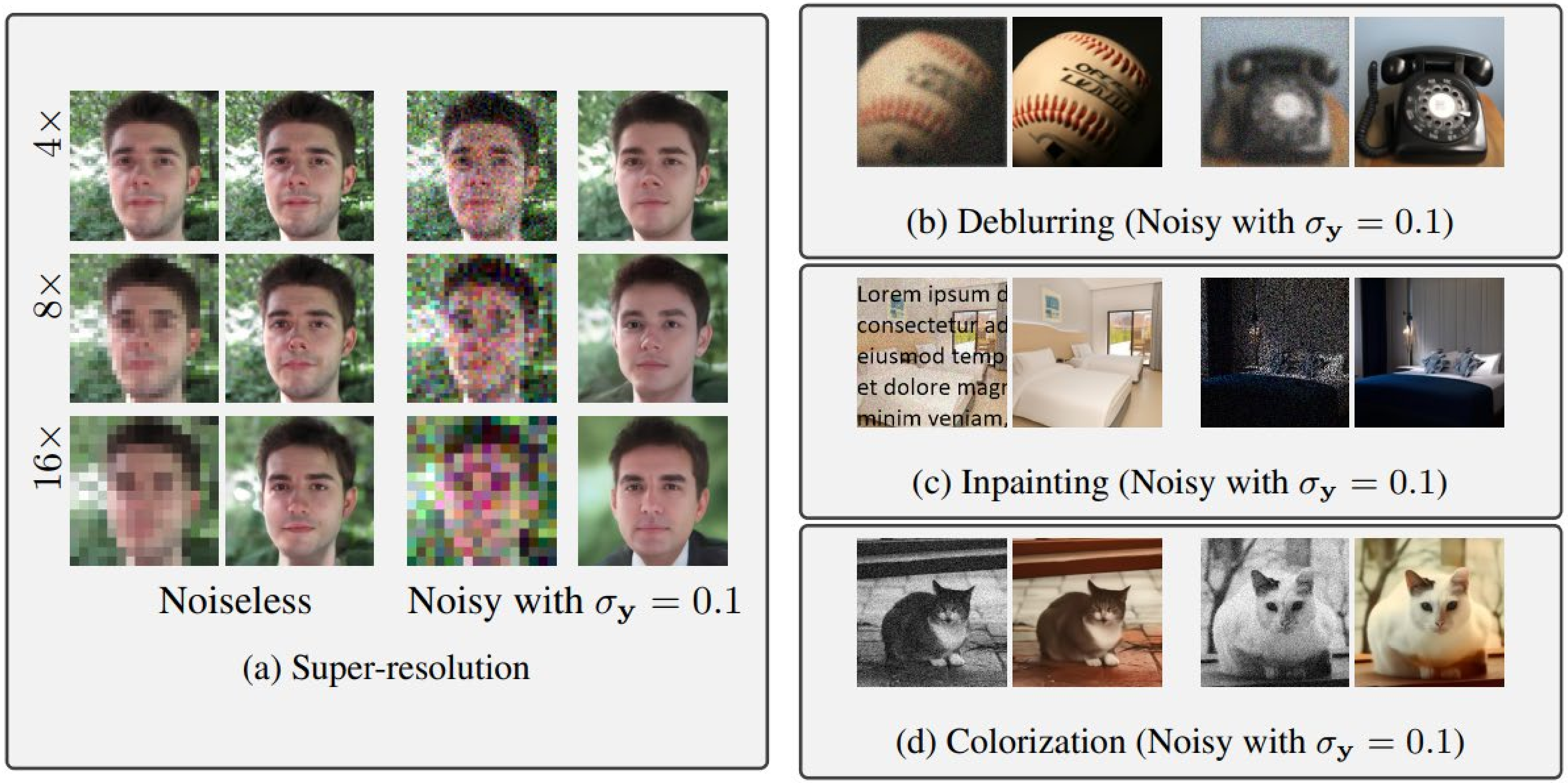
Background: DGP

- Iteratively optimization: time consuming
- Degradation model should be derivative
- GAN is not the best generative model currently



Exploiting deep generative prior for versatile image restoration and manipulation, Xingang Pan et al., ECCV2020 oral

Background: DDRM



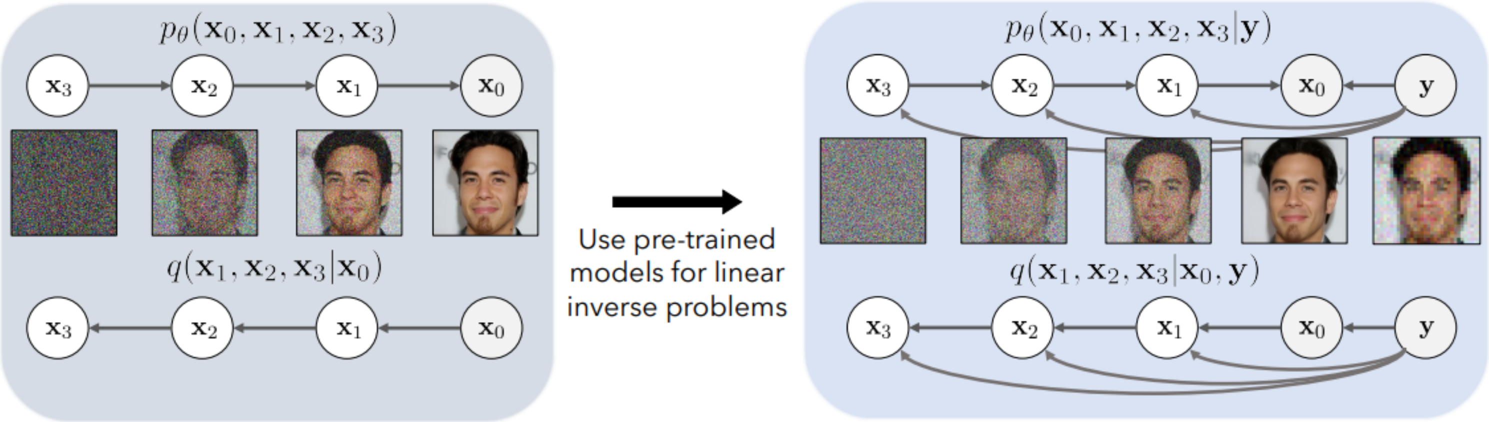
Denoising Diffusion Restoration Models, Bahjat Kawar, et al., NIPS22

Background: DDRM

Linear inverse problems: $y = Hx + z$

H → Degradation model

z → noise



Denoising Diffusion Probabilistic Models (Independent of inverse problem)

Denoising Diffusion Restoration Models (Dependent on inverse problem)

Define a proper distribution of $\bar{q}(x_t | x_0, y)$ and $\mathrm{p}_\theta(x_t | x_{t+1}, y)$ to meet the requirements of origin DDPM

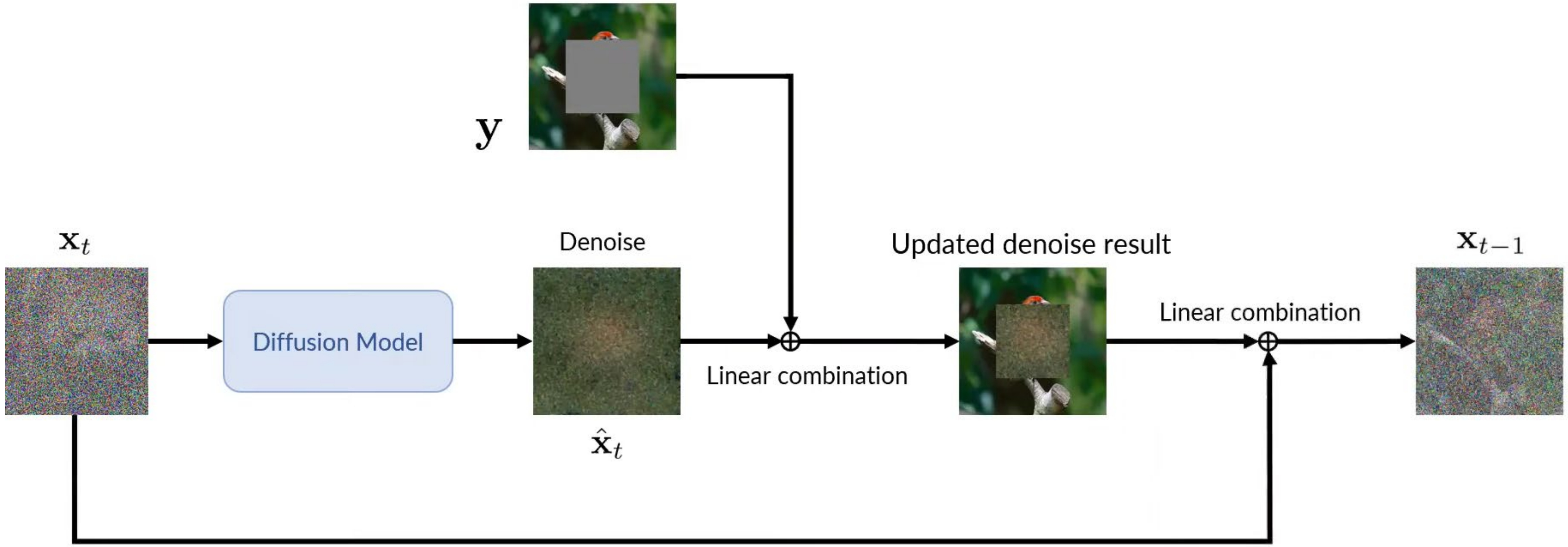
Denoising Diffusion Restoration Models, Bahjat Kawar, et al., NIPS22

Background: DDRM

Case for inpainting with no noise:

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

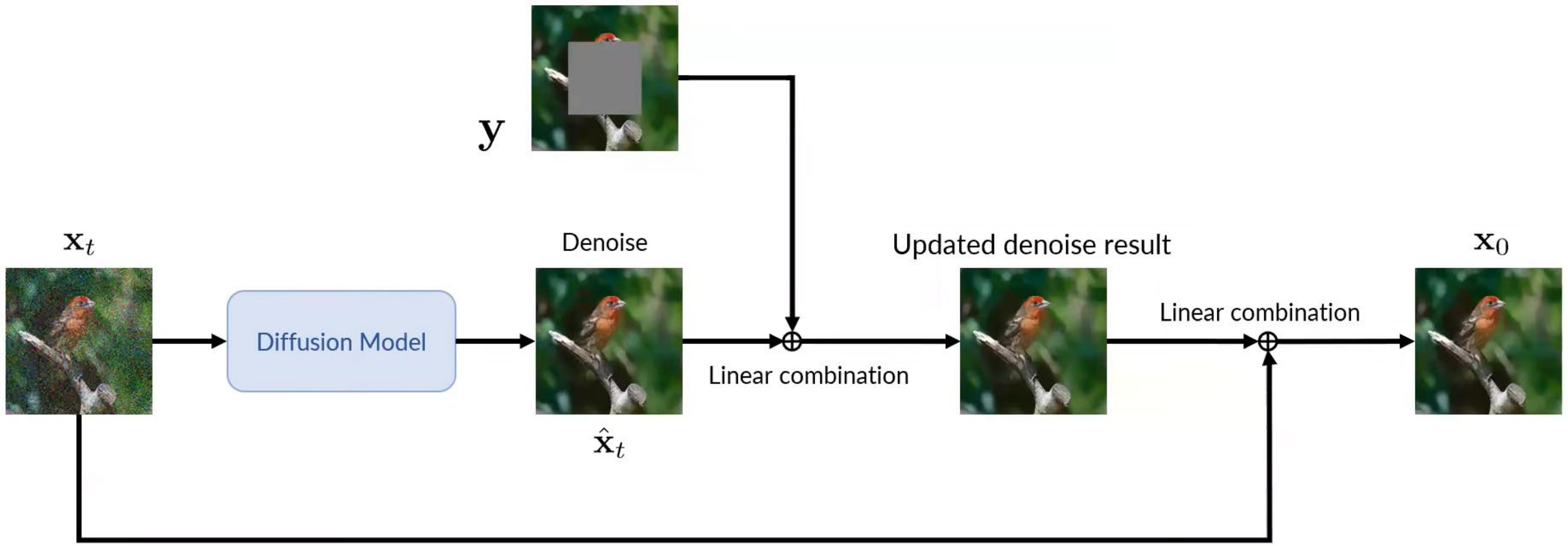


Denoising Diffusion Restoration Models, Bahjat Kawar, et al., NIPS22

Background: DDRM

Case for inpainting with no noise: [H = Diagonal with 0 and 1's]

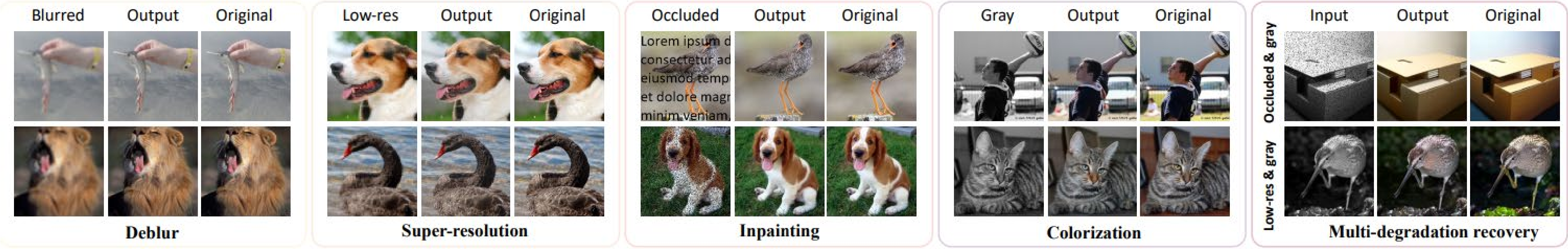
$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$



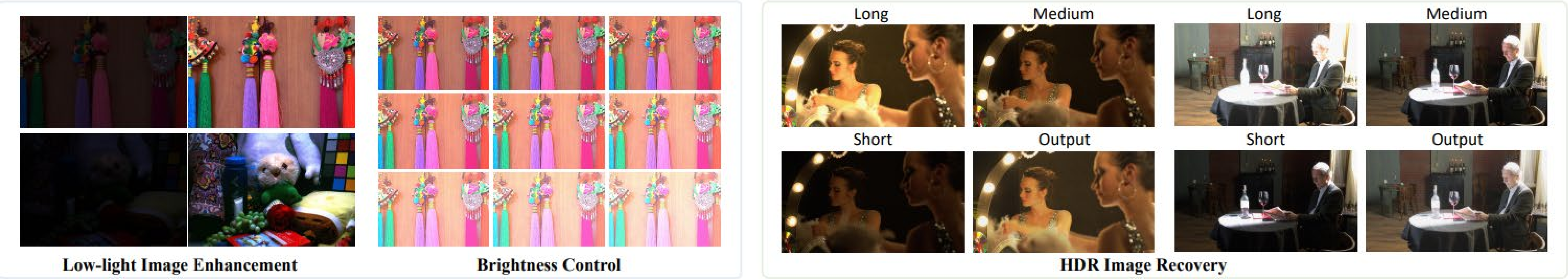
Denoising Diffusion Restoration Models, Bahjat Kawar, et al., NIPS22

Background

Linear and Multi-linear inverse problems



Non-linear inverse and blind degradation problems



Background

- DGP, DDRM are limited in specific degradation models
- We want to explore the generative prior thoroughly with unified restoration tasks

Methods	DGP [62]	DDRM [32]	GDP (Ours)
Prior	GAN	DDPM	DDPM
Linear	✔	✔	✔
Non-linear	✗	✗	✔
Blind	✗	✗	✔

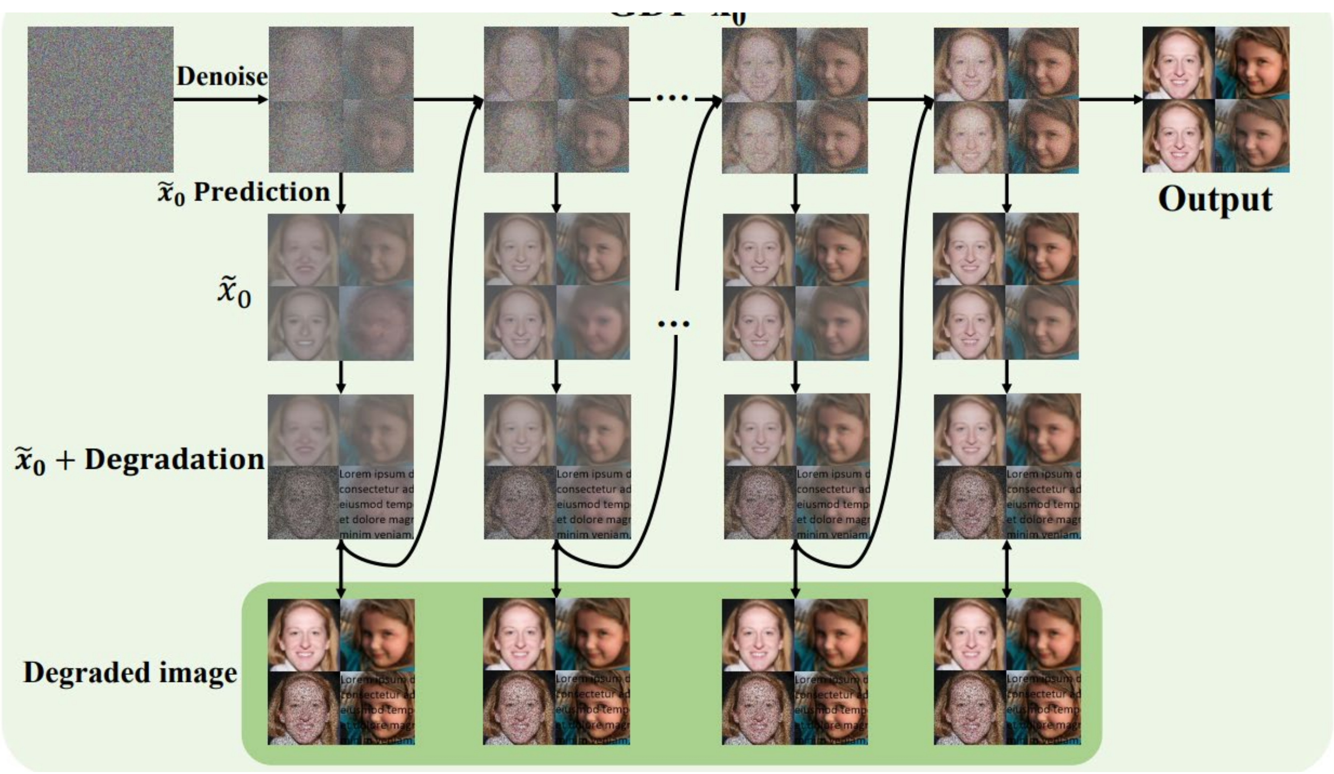
Outline

- Background
- Method
- Experiments
- Conclusion

GDP: Overview

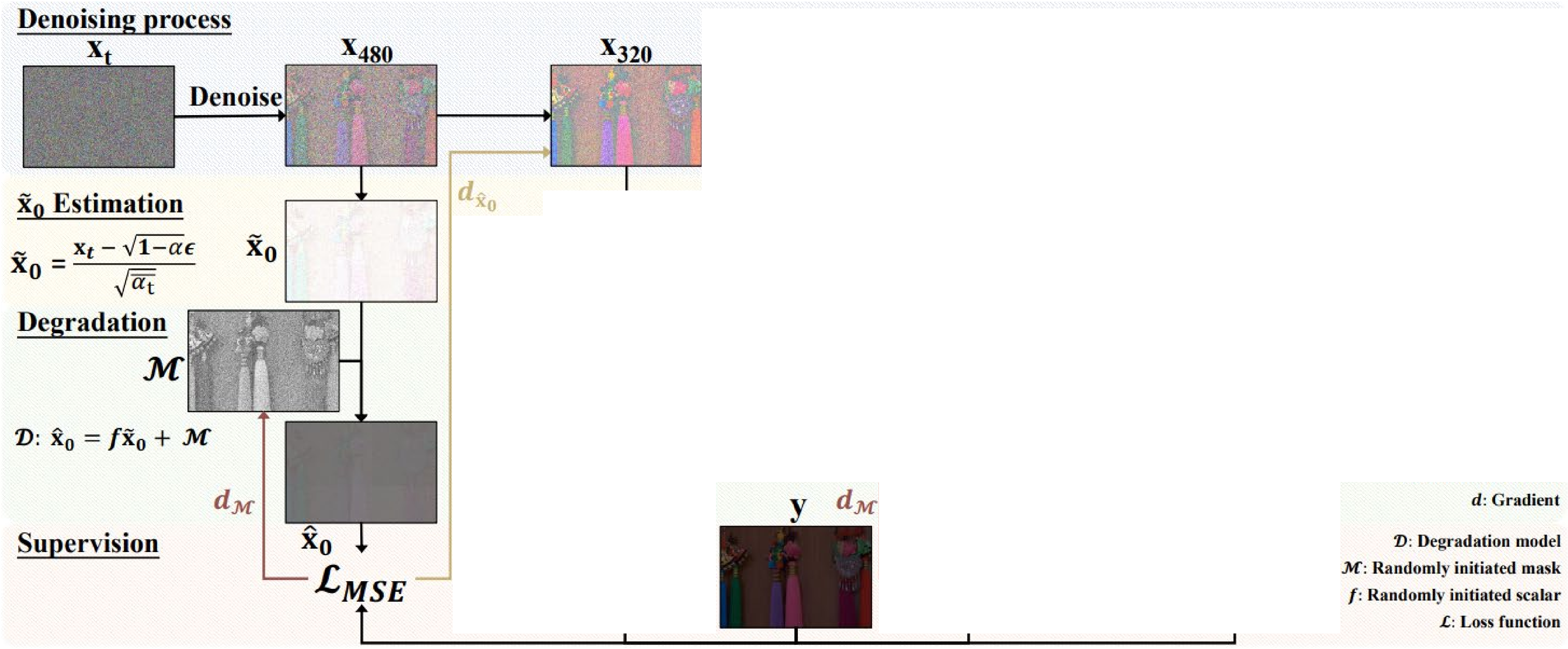
Similar to DGP: Degradation Alignment

Use intermediate $\tilde{x}_0$ to regularize the generation process



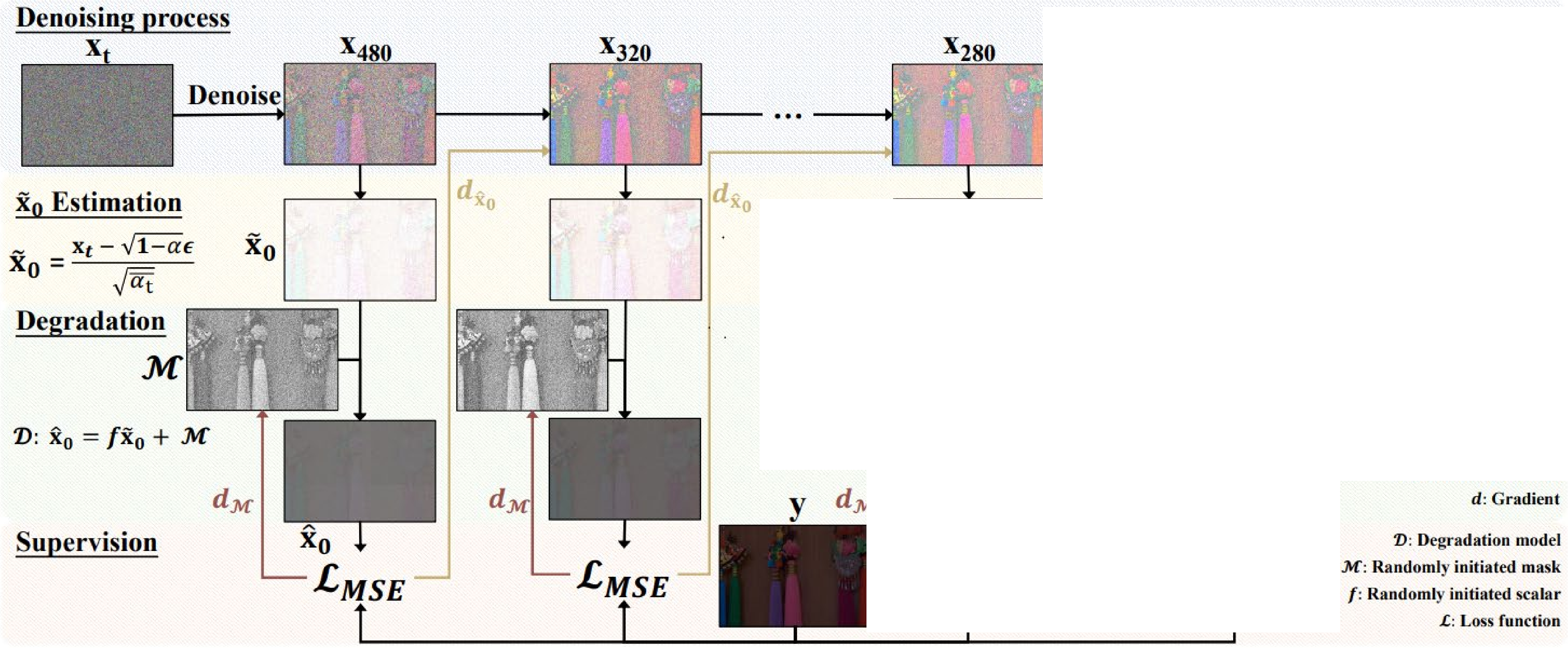
GDP: Overview

Use intermediate $\tilde{x}_0$ to regularize the generation process



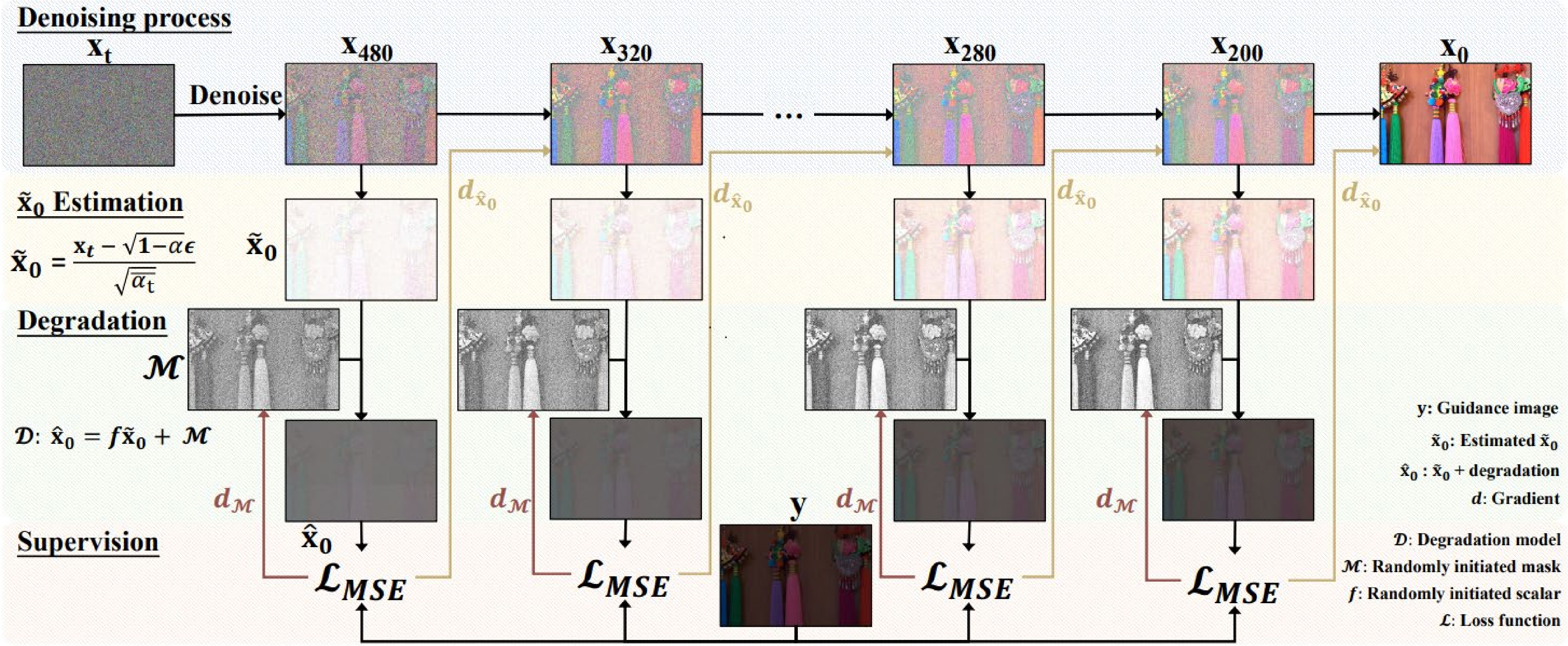
GDP: Overview

Use intermediate $\tilde{x}_0$ to regularize the generation process

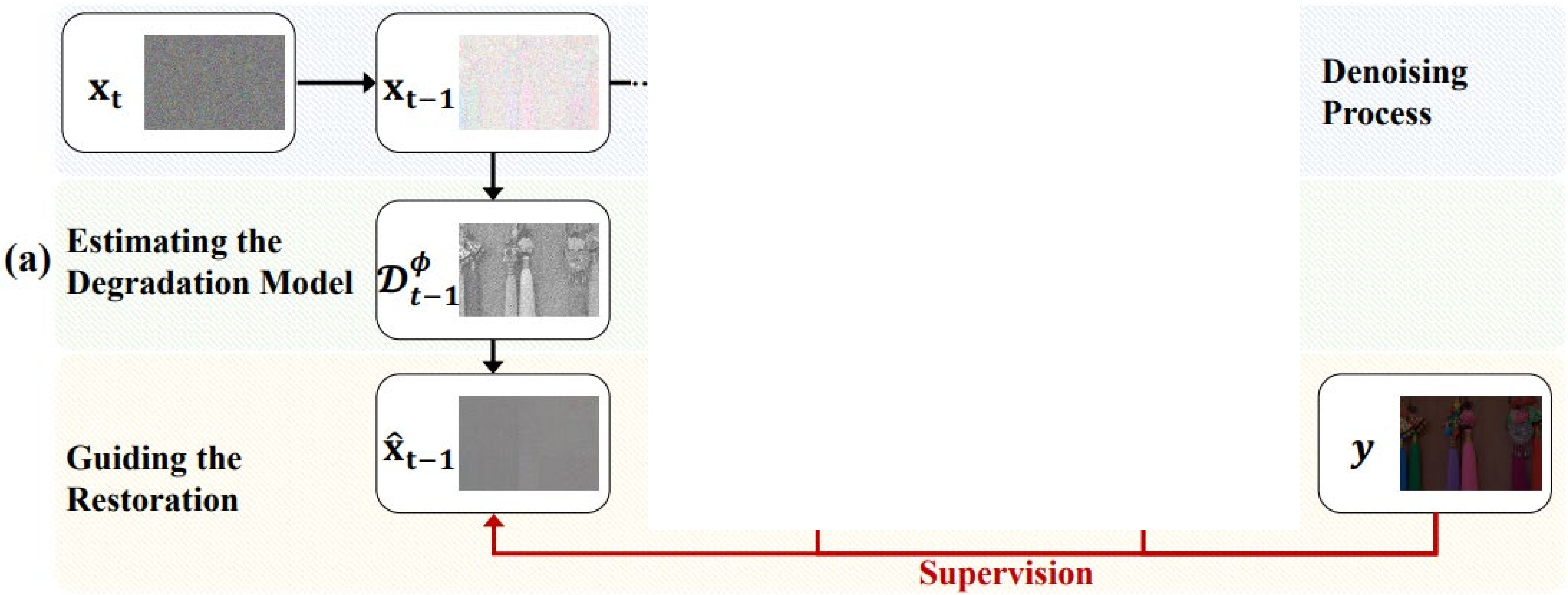


GDP: Overview

Use intermediate $\tilde{x}_0$ to regularize the generation process

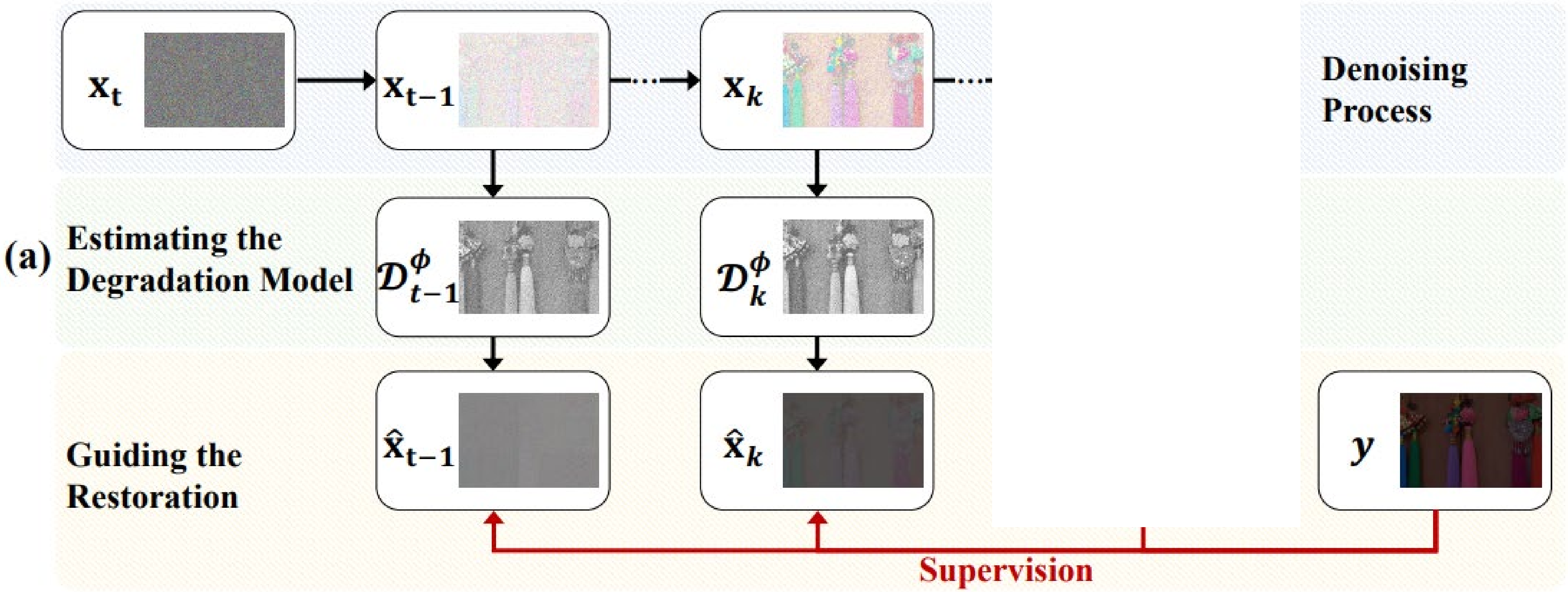


GDP: Optimize the degradation model



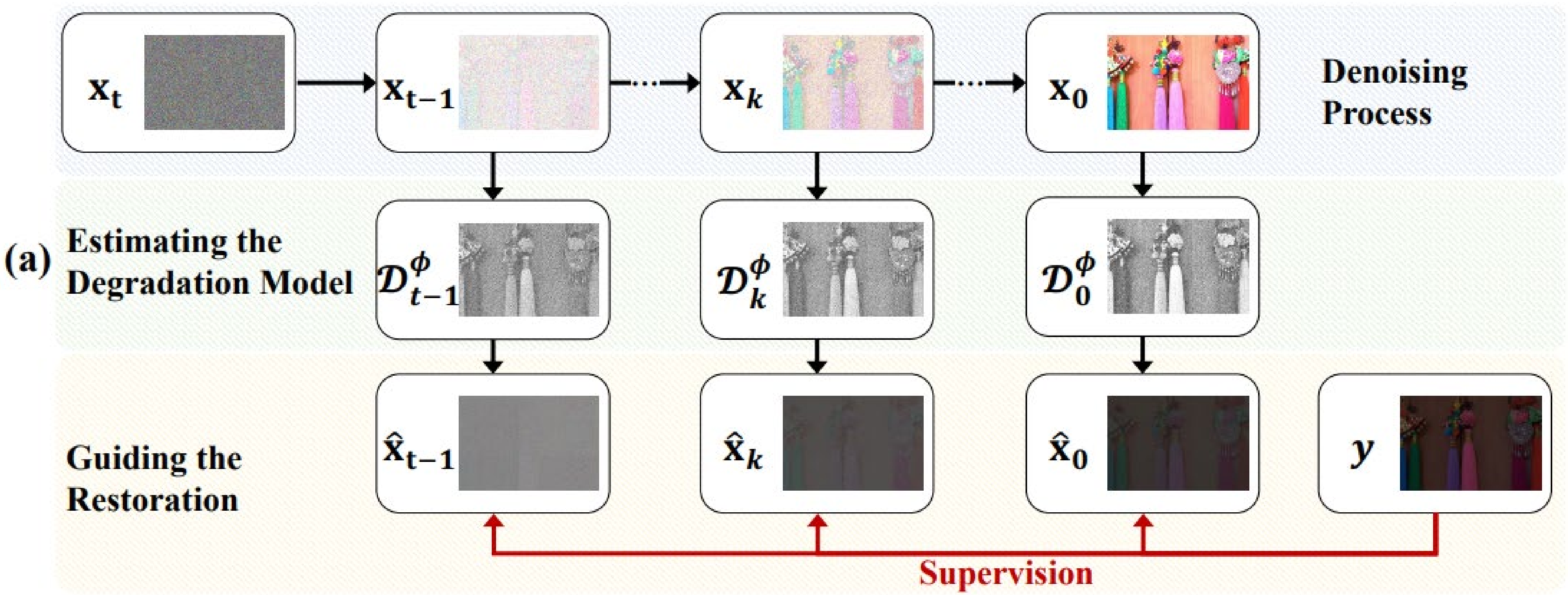
Simple but effective degradation model: $y = fx + \mathcal{M}$

GDP: Optimize the degradation model



Simple but effective degradation model: $y = fx + \mathcal{M}$

GDP: Optimize the degradation model



Simple but effective degradation model: $y = fx + \mathcal{M}$,

GDP: Method

DDPM denoising probability:

$$\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} | \boldsymbol{x}_t \quad \right) = \log \left(p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} | \boldsymbol{x}_t\right) \qquad \right)$$
$$\approx \log p(\boldsymbol{r})$$

Where r ~ $\mathcal{N}\left(\boldsymbol{r}; \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right) \qquad , \Sigma\right)$

GDP: Method

Conditional denoising probability[*]:

$$\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}\right) = \log \left(p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right) p\left(\boldsymbol{y} \mid \boldsymbol{x}_t\right)\right) + K_1$$
$$\approx \log p(\boldsymbol{r}) + K_2,$$

Where r ~$\mathcal{N}\left(\boldsymbol{r}; \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right) + \Sigma \boldsymbol{g}, \Sigma\right)$, and g= $\nabla_{\boldsymbol{x}_t} \log p\left(\boldsymbol{y} \mid \boldsymbol{x}_t\right)$

*proved by Diffusion models beat gans on image synthesis, Prafulla Dhariwal et al., NIPS2021

GDP: Method

Conditional denoising probability[*]:

$$\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}\right)=\log \left(p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right) p\left(\boldsymbol{y} \mid \boldsymbol{x}_t\right)\right)+K_1$$
$$\approx \log p(\boldsymbol{r})+K_2,$$

Where r ~ $\mathcal{N}\left(\boldsymbol{r} ; \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)+\Sigma \boldsymbol{g}, \Sigma\right)$, and g= $\nabla_{\boldsymbol{x}_t} \log p\left(\boldsymbol{y} \mid \boldsymbol{x}_t\right)$

Posterior probability definition:

$$p\left(\boldsymbol{y} \mid \boldsymbol{x}_t\right)=\frac{1}{Z} \exp \left(-\left[s \mathcal{L}\left(\mathcal{D}(\boldsymbol{x}_t), \boldsymbol{y}\right)+\lambda \mathcal{Q}(\boldsymbol{x}_t)\right]\right)$$

Reconstruction loss   Quality Enhancement loss

*proved by Diffusion models beat gans on image synthesis, Prafulla Dhariwal et al., NIPS2021

GDP-x_0

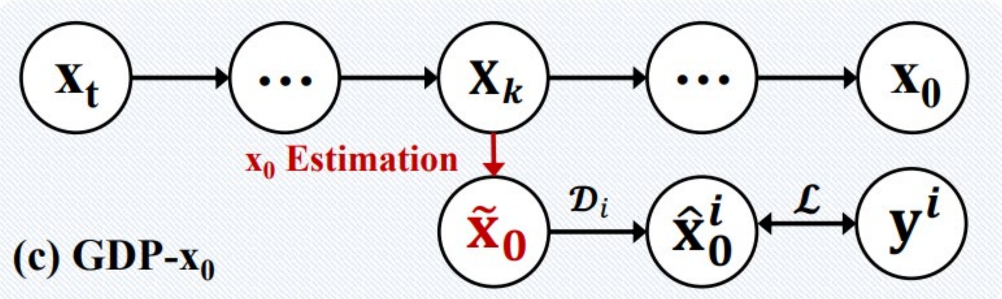
Calculate loss by "pseudo-clean" image $\tilde{x}_0$

Algorithm 2: GDP-$\boldsymbol{x}_0$: Conditioner guided diffusion sampling on $\tilde{\boldsymbol{x}}_0$, given a diffusion model $(\mu_\theta(\boldsymbol{x}_t), \Sigma_\theta(\boldsymbol{x}_t))$, corrupted image conditioner $\boldsymbol{y}$.

Input: Corrupted image $\boldsymbol{y}$, gradient scale s, degradation model $\mathcal{D}_\phi$ with randomly initiated parameters ϕ, learning rate l for optimizable degradation model, distance measure $\mathcal{L}$, optional quality enhancement loss $\mathcal{Q}$, quality enhancement scale λ.
Output: Output image $\boldsymbol{x}_0$ conditioned on $\boldsymbol{y}$

Sample $\boldsymbol{x}_T$ from $\mathcal{N}(0, \mathbf{I})$
for *t from T to 1* **do**
- $\mu, \Sigma = \mu_\theta(\boldsymbol{x}_t), \Sigma_\theta(\boldsymbol{x}_t)$
- $\tilde{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$
- $\mathcal{L}^{total}_{\phi,\tilde{\boldsymbol{x}}_0} = \mathcal{L}(\boldsymbol{y}, \mathcal{D}_\phi(\tilde{\boldsymbol{x}}_0)) + \mathcal{Q}(\tilde{\boldsymbol{x}}_0)$
- $\phi \leftarrow \phi - l\nabla_\phi \mathcal{L}^{total}_{\phi,\tilde{\boldsymbol{x}}_0}$
- Sample $\boldsymbol{x}_{t-1}$ by $\mathcal{N}\left(\mu + s\nabla_{\tilde{\boldsymbol{x}}_0}\mathcal{L}^{total}_{\phi,\tilde{\boldsymbol{x}}_0}, \Sigma\right)$

end
return $\boldsymbol{x}_0$



$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

$$\tilde{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

GDP: Loss

- Reconstruction Loss:
  - MSE, SSIM…

$$p(\boldsymbol{y} \mid \boldsymbol{x}_t) = \frac{1}{Z} \exp\left(-\left[s\mathcal{L}\left(\mathcal{D}(\boldsymbol{x}_t), \boldsymbol{y}\right) + \lambda \mathcal{Q}(\boldsymbol{x}_t)\right]\right)$$

Reconstruction loss   Quality Enhancement loss

- Quality Enhancement Loss (options for each task)
  - Exposure Control Loss: $L_{\exp} = \frac{1}{U} \sum_{k=1}^{U} |R_k - E|$
  - Color Constancy Loss: $L_{\text{col}} = \sum_{\forall (m,n) \in \varepsilon} (Y^m - Y^n)^2, \varepsilon = \{(R,G),(R,B),(G,B)\}$
  - Illumination Smoothness Loss: $L_{tv_{\mathcal{M}}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{c \in \xi} (|\nabla_h \mathcal{M}_n^c| + \nabla_v \mathcal{M}_n^c |)^2, \xi = \{R,G,B\}$
  - Maybe some good IQA metrics, all of them are adopted from ZeroDCE*

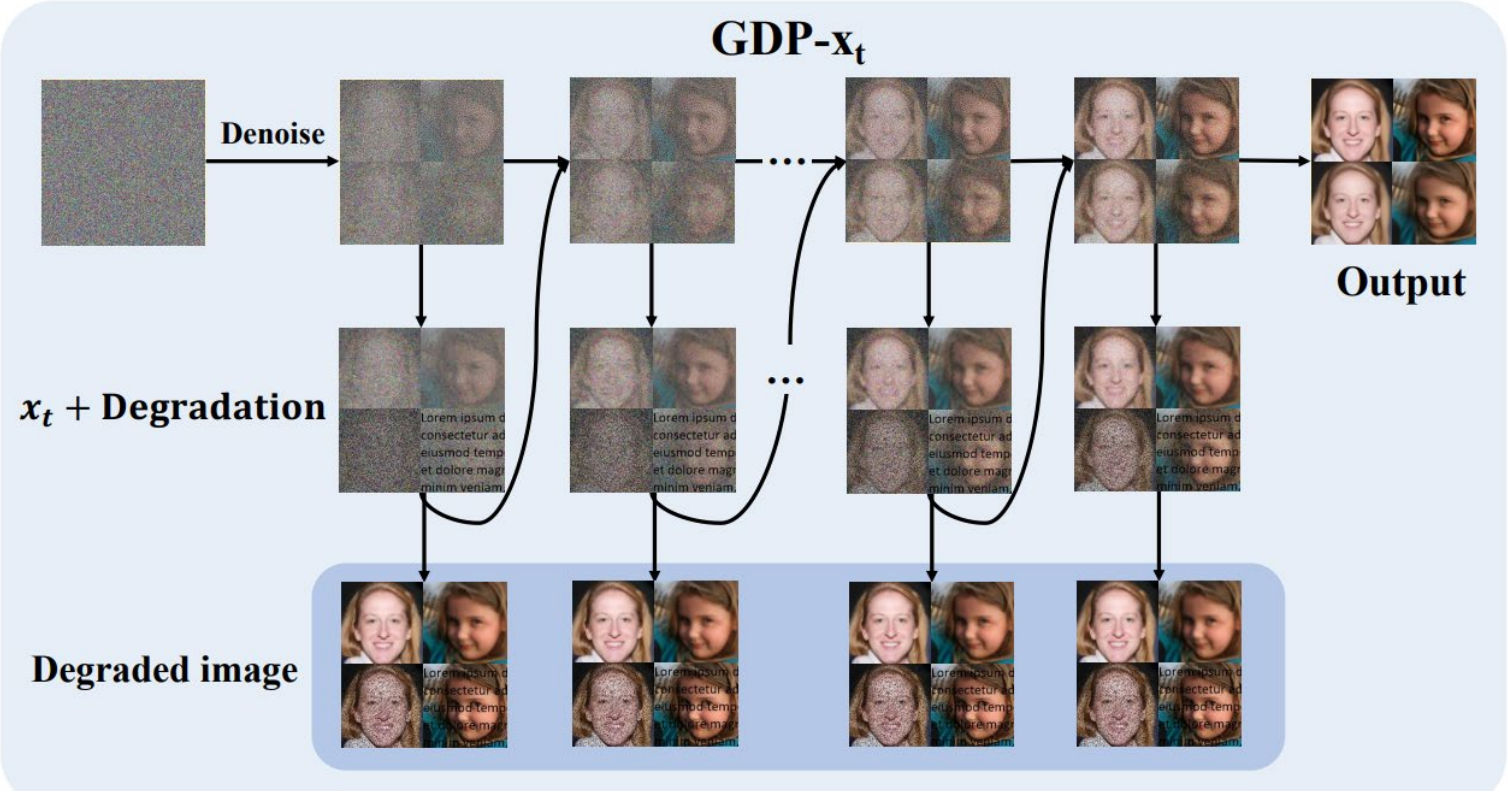
Zero-reference deep curve estimation for low-light image enhancement, Chunle Guo et al., CVPR2020

Outline

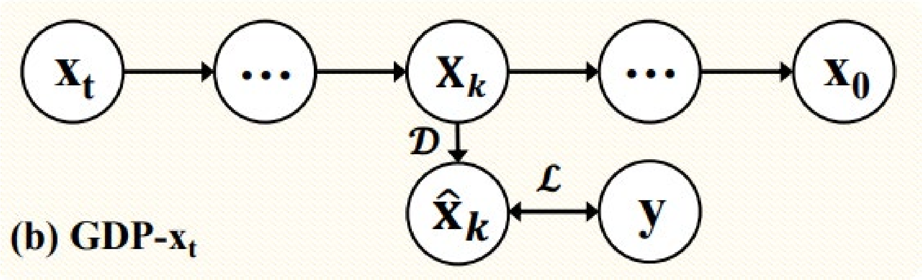
- Background
- Method
- Experiments
- Conclusion

Experiments: Another Variant

Add regularization on x_t, not on $\tilde{x}_0$



GDP-x_t



Algorithm 1: GDP-x_t with fixed degradation model: Conditioner guided diffusion sampling on x_t, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, corrupted image conditioner y.

Input: Corrupted image y, gradient scale s, degradation model $\mathcal{D}$, distance measure $\mathcal{L}$, optional quality enhancement loss $\mathcal{Q}$, quality enhancement scale λ.

Output: Output image x_0 conditioned on y

Sample x_T from $\mathcal{N}(0, \mathbf{I})$

for *t from T to 1* **do**

  $\mu, \Sigma = \mu_\theta(x_t), \Sigma_\theta(x_t)$

  $\mathcal{L}_{x_t}^{total} = \mathcal{L}(y, \mathcal{D}(x_t)) + \mathcal{Q}(x_t)$

  Sample x_{t-1} by $\mathcal{N}(\mu + s\nabla_{x_t}\mathcal{L}_{x_t}^{total}, \Sigma)$

end

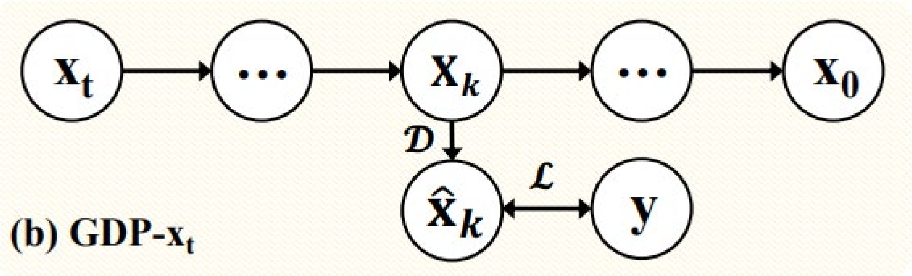
return x_0

$$\log p(y \mid x_t) = -\log Z - s\mathcal{L}(\mathcal{D}(x_t), y) - \lambda\mathcal{Q}(x_t)$$

$$\nabla_{x_t}\log p(y \mid x_t) = -s\nabla_{x_t}\mathcal{L}(\mathcal{D}(x_t), y) - \lambda\nabla_{x_t}\mathcal{Q}(x_t).$$

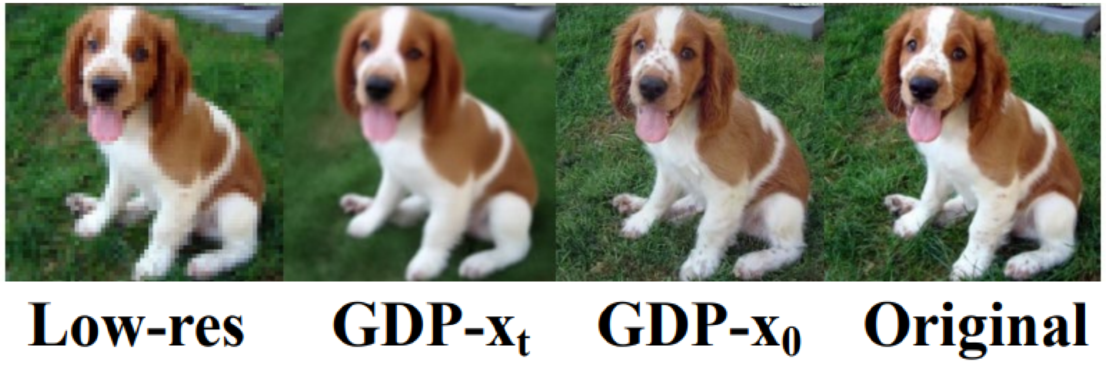
GDP-x_t

A naive MSE loss or perceptual loss will make x_t deviate from its original noise magnitude and do harm to the generation.



x_k: noisy image with a specific noise magnitude

y : corrupted image with no noise or noises of different magnitude



Experiments

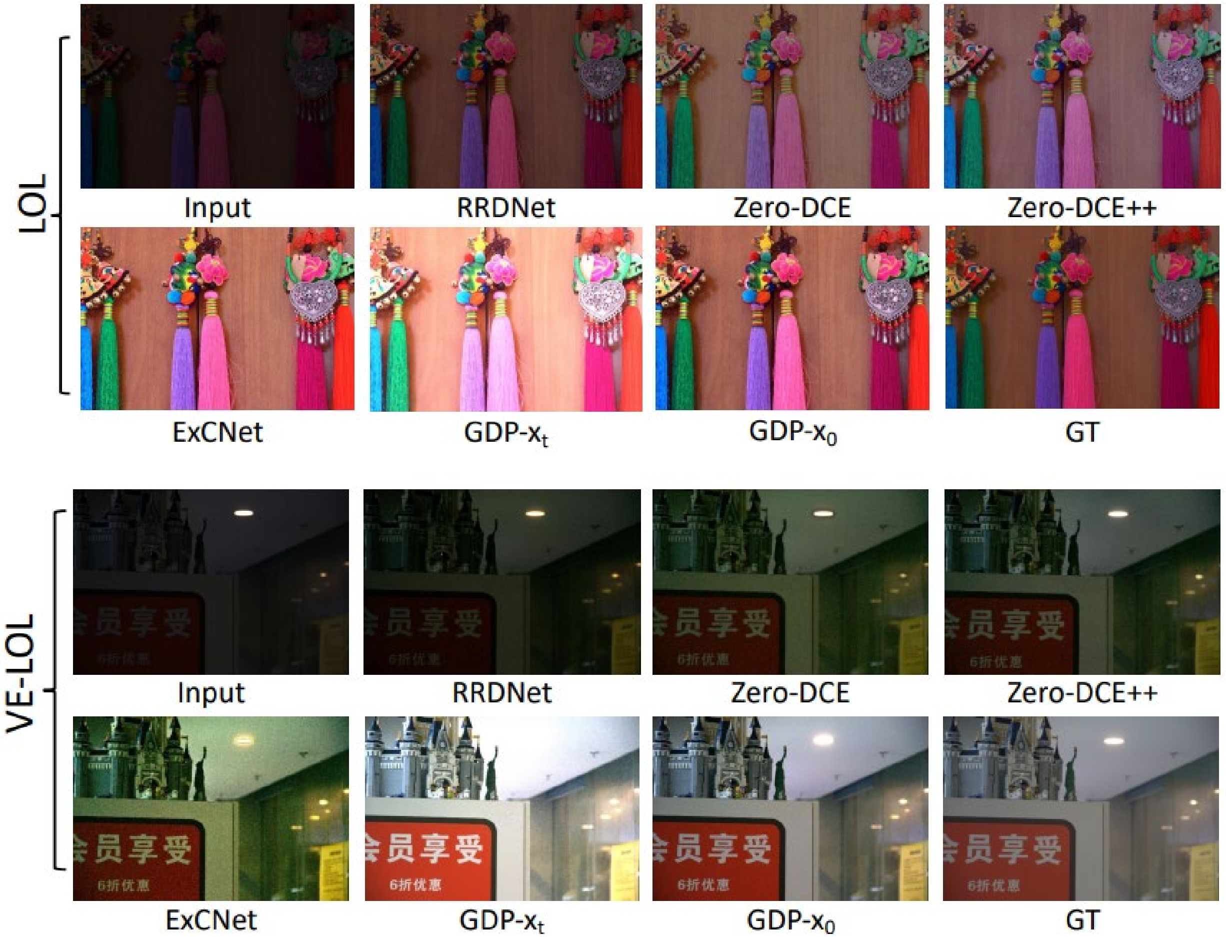
Qualitative comparison on colorization

Color Constancy Loss

$$L_{\text{col}} = \sum_{\forall(m,n)\in\varepsilon} (Y^m - Y^n)^2, \varepsilon = \{(R,G),(R,B),(G,B)\}$$

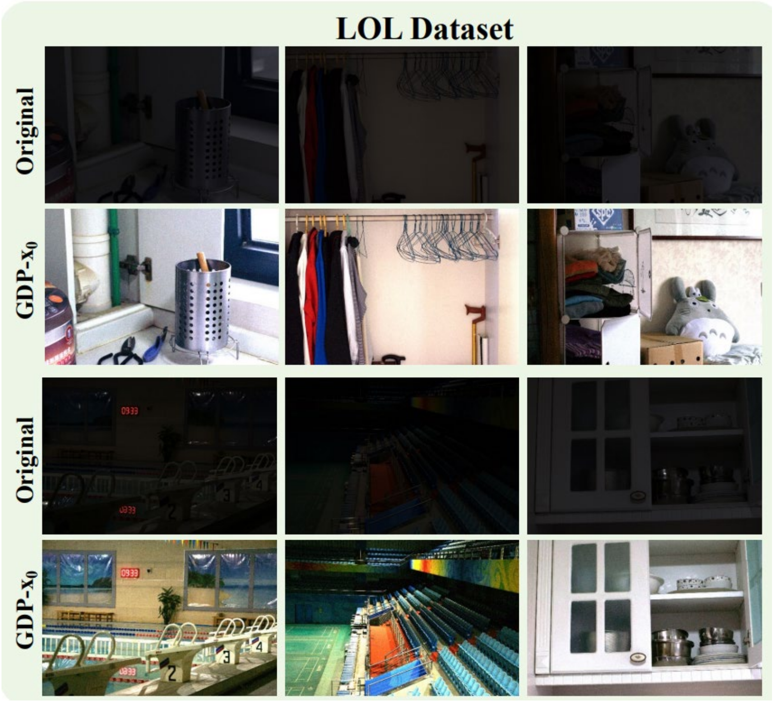
Experiments

Qualitative comparison of image enlighten task



Illumination Smoothness Loss:

$$L_{tv_{\mathcal{M}}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c\in\xi}(|\nabla_h \mathcal{M}_n^c| + \nabla_v \mathcal{M}_n^c\ |)^2, \xi = \{R, G, B\}$$



Experiments

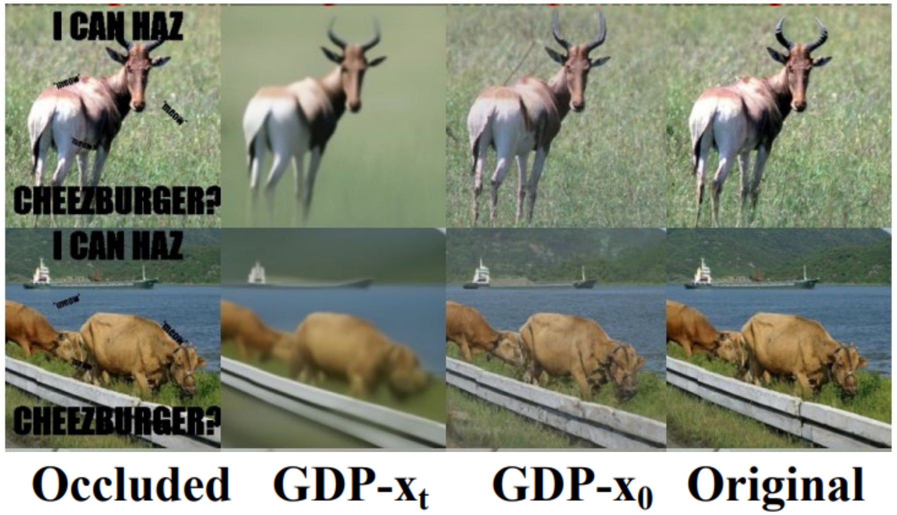
Qualitative comparison of image enlighten task

Exposure Control Loss:

$$L_{\exp} = \frac{1}{U} \sum_{k=1}^{U} |R_k - E|$$

Experiments

Qualitative comparison of inpainting task



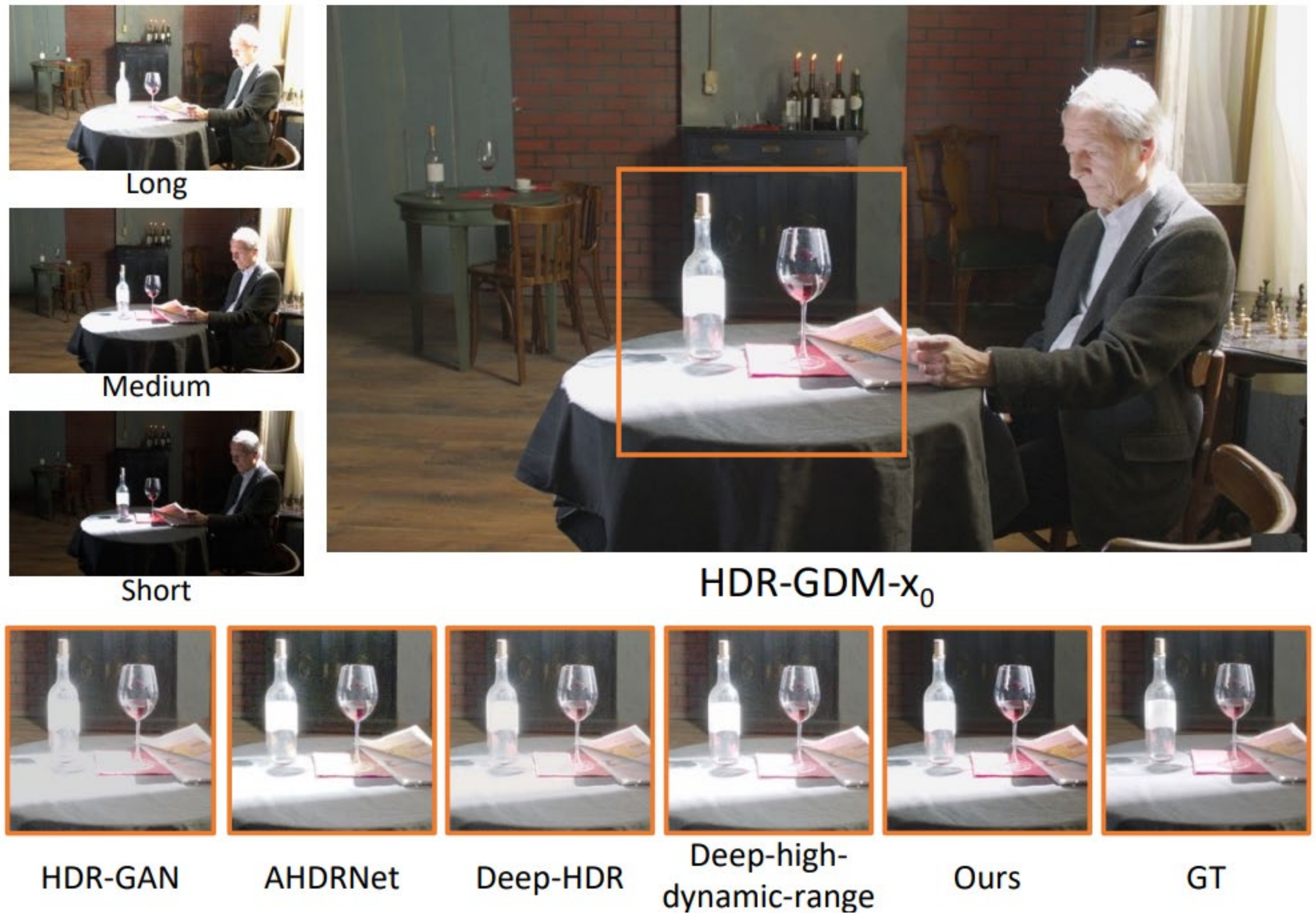
GDP-x_t may generate blurry images

Experiments

Qualitative comparison on HDR Recovery task

Exposure Control Loss:

$$L_{\text{exp}} = \frac{1}{U}\sum_{k=1}^{U} |R_k - E|$$



Experiments

Qualitative comparison on Multi-degradation tasks

Experiments

Quantitative comparison of linear image restoration tasks on ImageNet 1k

Method	4× Super-resolution				Deblur				25% Inpainting				Colorization			
	PSNR ↑	SSIM ↑	Consistency ↓	FID ↓	PSNR ↑	SSIM ↑	Consistency↓	FID ↓	PSNR ↑	SSIM ↑	Consistency↓	FID ↓	PSNR ↑	SSIM ↑	Consistency ↓	FID ↓
DGP [62]	21.65	0.56	158.74	152.85	26.00	0.54	475.10	136.53	27.59	0.82	414.60	60.65	18.42	0.71	305.59	94.59
SNIPS [33]	22.38	0.66	21.38	154.43	24.73	0.69	60.11	17.11	17.55	0.74	587.90	103.50	-	-	-	-
RED [69]	24.18	0.71	27.57	98.30	21.30	0.58	63.20	69.55	-	-	-	-	-	-	-	-
DDRM [32]	**26.53**	**0.78**	19.39	40.75	**35.64**	**0.98**	50.24	4.78	34.28	0.95	**4.08**	24.09	**22.12**	0.91	37.33	47.05
GDP-x_t	24.27	0.67	80.32	64.67	25.86	0.75	54.08	5.00	31.06	0.93	8.80	20.24	21.30	0.86	75.24	66.43
GDP-x_0	24.42	0.68	**6.49**	**38.24**	25.98	0.75	**41.27**	**2.44**	**34.40**	**0.96**	5.29	**16.58**	21.41	**0.92**	**36.92**	**37.60**

Low psnr/ssim, high FID

GDP-x_0 performs well than GDP-x_t

Experiments

Quantitative comparison of image enlighten task

Learning	Methods	LOL [88]					VE-LOL-L [47]					LoLi-Phone [41]	
		PSNR ↑	SSIM↑	FID↓	LOE↓	PI↓	PSNR↑	SSIM↑	FID ↓	LOE↓	PI↓	LOE↓	PI↓
Supervised learning	LLNet [50]	17.91	0.76	169.20	384.21	4.10	17.38	0.73	124.98	291.59	5.54	343.34	5.36
	LightenNet [43]	10.29	0.45	90.91	273.21	7.09	13.26	0.57	82.26	199.45	7.29	500.22	6.63
	Retinex-Net [88]	17.24	0.55	129.99	513.28	8.63	16.41	0.64	135.20	421.41	8.62	542.29	8.23
	MBLLEN [52]	17.90	0.77	122.69	175.10	8.39	15.95	0.70	105.74	114.91	7.45	137.34	6.46
	KinD [104]	17.57	0.82	74.52	377.59	7.41	18.07	0.78	80.12	253.79	7.51	265.47	6.84
	KinD++ [102]	17.60	0.80	100.15	712.12	7.96	16.80	0.74	101.23	421.97	7.98	382.51	7.71
	TBFEN [51]	17.25	0.83	90.59	367.66	8.29	18.91	0.81	91.30	276.65	8.02	214.30	7.34
	DSLR [46]	14.98	0.67	183.92	272.68	7.09	15.70	0.68	124.80	271.63	7.27	281.25	6.99
Unsupervised learning	EnlightenGAN [29]	17.44	0.74	82.60	379.23	8.78	17.45	0.75	86.51	311.85	8.27	373.41	7.26
Self-supervised learning	DRBN [92]	15.15	0.52	94.96	692.99	5.53	18.47	0.78	88.10	268.70	6.15	285.06	5.31
Zero-shot learning	ExCNet [99]	**16.04**	0.62	111.18	220.38	8.70	16.20	0.66	115.24	225.15	8.62	359.96	7.95
	Zero-DCE [23]	14.91	**0.70**	81.11	245.54	8.84	**17.84**	**0.73**	85.72	194.10	8.12	214.30	7.34
	Zero-DCE++ [42]	14.86	0.62	86.22	302.06	7.08	16.12	0.45	86.96	313.50	7.92	308.15	7.18
	RRDNet [106]	11.37	0.53	89.09	127.22	8.17	13.99	0.58	83.41	94.23	7.36	92.73	7.20
	GDP-x_t	7.32	0.57	238.92	364.15	8.26	9.45	0.50	152.68	194.49	7.12	508.73	8.06
	GDP-x_0	13.93	0.63	**75.16**	**110.39**	**6.47**	13.04	0.55	**78.74**	**79.08**	**6.47**	**75.29**	**6.35**

Outline

- Background
- Method
- Experiments
- Conclusion

Conclusion

- Image restoration -> conditional generation
- Tackle the linear inverse, non-linear and blind problems.
- Use intermediate $\tilde{x}_0$ to regularize the generation process
  - Easy but effective way to insert the condition

Thanks for listening!