Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model

Yinhuai Wang, Jiwen Yu, Jian Zhang

Presented by Jakub Grzywaczewski

MI2.AI Summer Seminar
at Warsaw University of Technology

May 12, 2025

Overview

1. Introduction

2. Theoretical background

3. Denoising Diffusion Null-Space Model

4. Enhanced Version: DDNM$^+$

5. Miscellaneous

QR Codes

(a) Arxiv

(b) Open Review

(c) GitHub

ZERO-SHOT IMAGE RESTORATION USING DENOISING DIFFUSION NULL-SPACE MODEL

Yinhuai Wang[1*], **Jiwen Yu**[1*], **Jian Zhang**[1,2]
[1]Peking University Shenzhen Graduate School, [2]Peng Cheng Laboratory
{yinhuai; yujiwen}@stu.pku.edu.cn, zhangjian.sz@pku.edu.cn

ICLR
International Conference On Learning Representations

ICLR 2023 Oral 6 Track 4:
Applications & Social Aspects of
Machine Learning & General Machine Learning

Authors

Yinhuai Wang
Ph.D student at HKUST

Jiwen Yu
Ph.D student at HKUST

Jian Zhang
Associate Professor at Peking University

Diffusion models 1/2

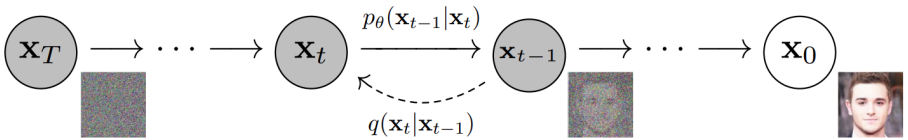


Figure: The directed graphical model considered in DDPM paper

- Forward process (from x_0 to x_T) as a Markov chain with transition probabilities $q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ according to variance schedule $\beta_1, \beta_2, \cdots, \beta_T$.
- Additionally because of the Markov property we get $q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$ where $\alpha_s := 1 - \beta_s$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.
- We want to learn the true $q(x_{t-1}|x_t)$. However $q(x_{t-1}|x_t)$ is not tractable, but $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \sigma_t I)$ is. Now we need a way to get x_0 using x_t.

Diffusion models 2/2

Given that $q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$ we can re-parametrize this to $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$ which we can revert to arrive at

$$\hat{x}_0(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t)\right) \tag{1}$$

where $\epsilon_\theta(x_t, t)$ is trained to approximate the true added noise ϵ.
Putting it all together into $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \sigma_t I)$ and re-parameterizing once again we get the DDPM sampler:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t, z \quad z \sim \mathcal{N}(0, I). \tag{2}$$

Full expressions for $\tilde{\mu}_t(x_t, x_0)$ and σ_t can be found in the DDPM [Ho et al., 2020] paper.

General Inverse Problems

Inverse Problem

Given an observation $y = A(x)$ and a known operator A recover the most plausible input $\hat{x} \in \mathbb{R}^n$ given some a priori distribution $q(x)$.

Key concepts:

- Consistency: Does $A(x) = y$ for the generated x?
- Realness: Does x come from the distribution q?

Linear Inverse Problem

Given an observation $y = Ax$ and a known linear operator A recover the most plausible input $\hat{x} \in \mathbb{R}^n$ given some a prior distribution $q(x)$.

For images this problem can be formulated as:

$$\hat{x} = \arg\min_x \frac{1}{2\sigma^2}||Ax - y||_2^2 + \lambda\mathcal{R}(x), \tag{3}$$

where the term $\frac{1}{2\sigma^2}||Ax - y||_2^2$ is responsible for data consistency and $\lambda\mathcal{R}(x)$ regularizes the realness of the generated $\hat{x}$ (fit to the a priori distribution).

Examples of Linear Inverse Problems

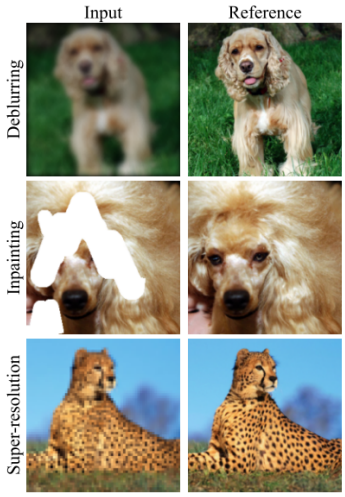


Figure: Part of a image from the I^{2}SB paper

- We assume the $q(x)$ here is a distribution of real images.
- Inverse Problems on Images are often called Image Restoration Problems.

Examples of Non-Linear Inverse Problems

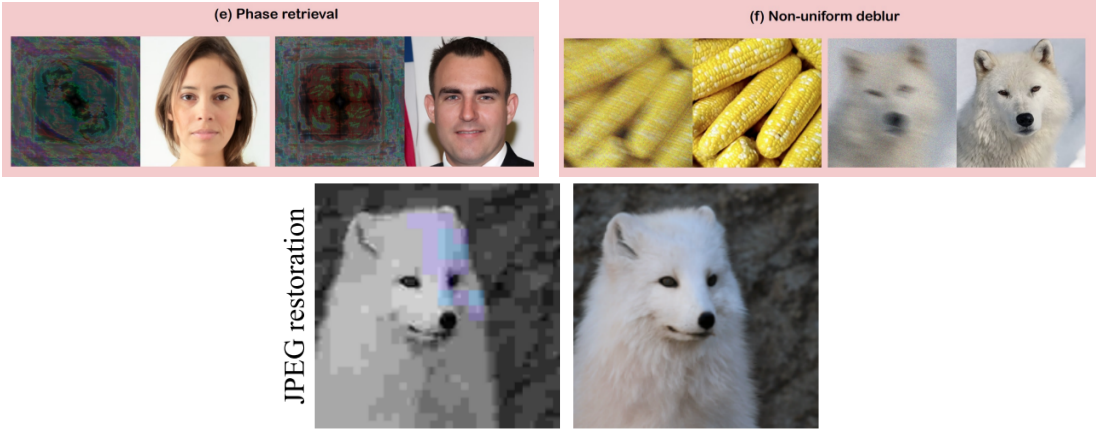


Figure: Images from DPS [Chung et al., 2022] and I^2SB [Liu et al., 2023] papers

Pseudoinverse

Moore-Penrose Pseudoinverse Definition

For $A \in \mathbb{R}^{m \times n}$ a pseudoinverse of A is defined as a matrix $A^{\dagger} \in \mathbb{R}^{n \times m}$ satisfying all Moore-Penrose conditions:

- $AA^{\dagger}A = A$,
- $A^{\dagger}AA^{\dagger} = A^{\dagger}$,
- $AA^{\dagger}$ and $A^{\dagger}A$ are Hermitian.

Those pseudoinverses can be created either manual or by using the SVD decomposition.

Pseudoinverse from SVD

Colorization. For colorization, we choose the degradation matrix $\mathbf{A} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$ for each pixel as we described in Sec. 3.2. Fig.9(g) demonstrates the example of colorization degradation.

Solve the Pseudo-Inverse Using SVD Considering we have a linear operator $\mathbf{A}$, we need to compute its pseudo-inverse $\mathbf{A}^\dagger$ to implement the algorithm of the proposed DDNM. For some simple degradation like inpainting, colorization, and SR based on average pooling, the pseudo-inverse $\mathbf{A}^\dagger$ can be constructed manually, which has been discussed in Sec. 3.2. For general cases, we can use the singular value decomposition (SVD) of $\mathbf{A}(= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)$ to compute the pseudo-inverse $\mathbf{A}^\dagger(= \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^\top)$ where $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^\dagger$ have the following relationship:

$$\mathbf{\Sigma} = diag\{s_1, s_2, \cdots\}, \mathbf{\Sigma}^\dagger = diag\{d_1, d_2, \cdots\}, \tag{21}$$

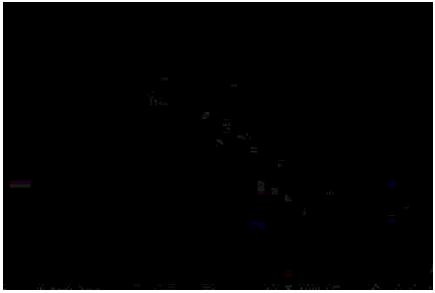
$$d_i = \begin{cases} \frac{1}{s_i} & s_i \neq 0 \\ 0 & s_i = 0 \end{cases}, \tag{22}$$

where s_i means the i-th singular value of $\mathbf{A}$ and d_i means the i-th diagonal element of $\mathbf{\Sigma}^\dagger$.

Anti-examples of Pseudoinverse

One paper stated that if we select A to be the JPEG Encoder than $A^\dagger$ is the JPEG Decoder. I was not convinced.

(a) Original image x



(b) $A^\dagger Ax - A^\dagger AA^\dagger Ax$

Figure: The absolute error between $A^\dagger Ax$ and $A^\dagger AA^\dagger Ax$ with normalized images to $[0, 1]$ is 41.25.

Range-Null Space Decomposition

Using pseudoinverse of a matrix A one can decompose x as:

$$x \equiv A^{\dagger}Ax + (I - A^{\dagger}A)x, \tag{4}$$

where $A^{\dagger}Ax$ is range-space part and the $(I - A^{\dagger}A)$ is the null-space part.

Denoising Diffusion Null-Space Model (DDNM)

Given a diffusion model trained on real images for noise prediction we can modify the $x_{0|t}$ by considering the range-null space decomposition and replacing the range-space part with $A^\dagger y$. That is if we consider an inverse problem $y = Ax$ at each step of the diffusion we have access to $x_{0|t}$ which we modify to

$$\hat{x}_{0|t} = A^\dagger y + (I - A^\dagger A)x_{0|t}. \tag{5}$$

Essentially allowing the diffusion to take place only in the null-space of the matrix A. This operation ensures consistency at each step:

$$A\hat{x}_{0|t} = AA^\dagger y + A(I - A^\dagger A)x_{0|t} = AA^\dagger y = AA^\dagger Ax = Ax = y. \tag{6}$$

Algorithm of DDNM

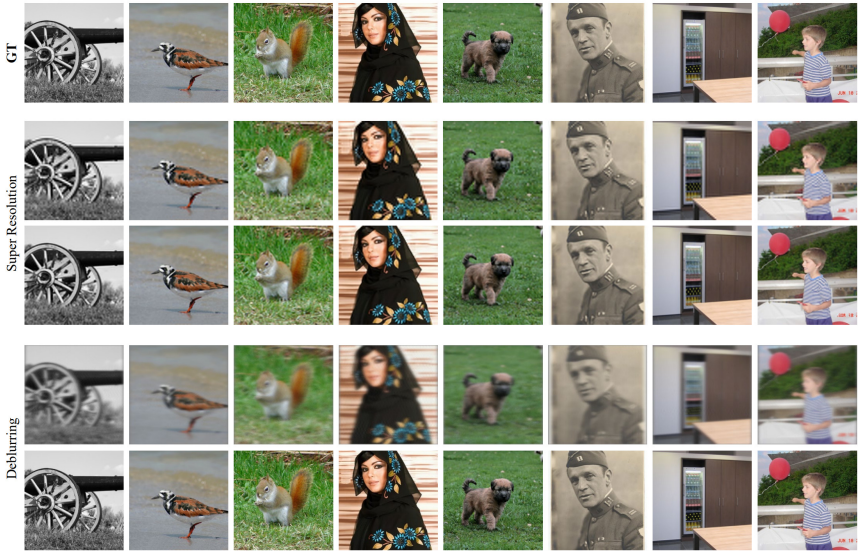
Algorithm 1 Sampling of DDPM

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \mathcal{Z}_\theta(\mathbf{x}_t, t) \sqrt{1 - \bar{\alpha}_t} \right)$
  $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_{0|t})$
end for
return $\mathbf{x}_0$

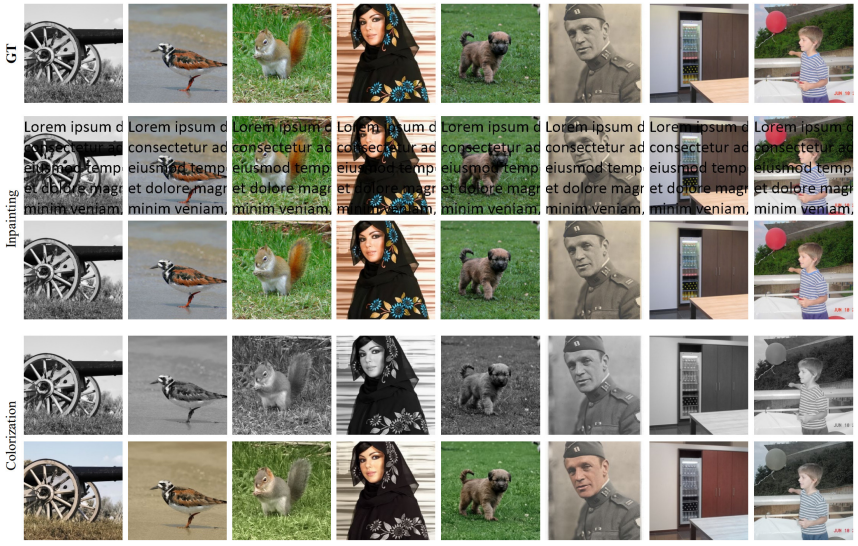
Algorithm 1 Sampling of DDNM

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \mathcal{Z}_\theta(\mathbf{x}_t, t) \sqrt{1 - \bar{\alpha}_t} \right)$
  $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}_{0|t}$
  $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$
end for
return $\mathbf{x}_0$

Qualitative Results of DDNM 1/2



Qualitative Results of DDNM 2/2



Quantitative Results

ImageNet	4× SR	Deblurring	Colorization	CS 25%	Inpainting
Method	PSNR↑/SSIM↑/FID↓	PSNR↑/SSIM↑/FID↓	*Cons*↓/FID↓	PSNR↑/SSIM↑/FID↓	PSNR↑/SSIM↑/FID↓
$\mathbf{A}^\dagger\mathbf{y}$	24.26 / 0.684 / 134.4	18.56 / 0.6616 / 55.42	0.0 / 43.37	15.65 / 0.510 / 277.4	14.52 / 0.799 / 72.71
DGP	23.18 / 0.798 / 64.34	N/A	- / 69.54	N/A	N/A
ILVR	27.40 / **0.870** / 43.66	N/A	N/A	N/A	N/A
RePaint	N/A	N/A	N/A	N/A	31.87 / **0.968** / 12.31
DDRM	27.38 / 0.869 / 43.15	43.01 / 0.992 / 1.48	260.4 / 36.56	19.95 / 0.704 / 97.99	31.73 / 0.966 / 4.82
DDNM(ours)	**27.46 / 0.870/ 39.26**	**44.93 / 0.994 / 1.15**	**42.32 / 36.32**	**21.66 / 0.749 / 64.68**	**32.06 / 0.968 / 3.89**
CelebA	4× SR	Deblurring	Colorization	CS 25%	Inpainting
Method	PSNR↑/SSIM↑/FID↓	PSNR↑/SSIM↑/FID↓	*Cons*↓/FID↓	PSNR↑/SSIM↑/FID↓	PSNR↑/SSIM↑/FID↓
$\mathbf{A}^\dagger\mathbf{y}$	27.27 / 0.782 / 103.3	18.85 / 0.741 / 54.31	0.0 / 68.81	15.09 / 0.583 / 377.7	15.57 / 0.809 / 181.56
PULSE	22.74 / 0.623 / 40.33	N/A	N/A	N/A	N/A
ILVR	31.59 / **0.945** / 29.82	N/A	N/A	N/A	N/A
RePaint	N/A	N/A	N/A	N/A	35.20 / 0.981 /14.19
DDRM	**31.63 / 0.945** / 31.04	43.07 / 0.993 / 6.24	455.9 / 31.26	24.86 / 0.876 / 46.77	34.79 / 0.978 /12.53
DDNM(ours)	**31.63 / 0.945 / 22.27**	**46.72 / 0.996 / 1.41**	**26.25 / 26.44**	**27.56 / 0.909 / 28.80**	**35.64 / 0.982 /4.54**

Table 1: Quantitative results of zero-shot IR methods on **ImageNet**(*top*) and **CelebA**(*bottom*), including five typical IR tasks. We mark N/A for those not applicable and **bold** the best scores.

Scaling Range-Space Correction to Support Noisy Image Restoration We consider noisy IR problems in the form of $\mathbf{y}{=}\mathbf{A}\mathbf{x} + \mathbf{n}$, where $\mathbf{n}{\in}\mathbb{R}^{d\times 1}{\sim}\mathcal{N}(\mathbf{0}, \sigma_{\mathbf{y}}^2\mathbf{I})$ represents the additive Gaussian noise and $\mathbf{A}\mathbf{x}$ represents the clean measurement. Applying DDNM directly yields

$$\hat{\mathbf{x}}_{0|t} = \mathbf{A}^{\dagger}\mathbf{y} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{x}_{0|t} = \mathbf{x}_{0|t} - \mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{A}\mathbf{x}) + \mathbf{A}^{\dagger}\mathbf{n}, \tag{16}$$

where $\mathbf{A}^{\dagger}\mathbf{n} \in \mathbb{R}^{D\times 1}$ is the extra noise introduced into $\hat{\mathbf{x}}_{0|t}$ and will be further introduced into $\mathbf{x}_{t-1}$. $\mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{A}\mathbf{x})$ is the correction for the range-space contents, which is the key to *Consistency*. To solve noisy image restoration, we propose to modify DDNM (on Eq. 13 and Eq. 14) as:

$$\hat{\mathbf{x}}_{0|t} = \mathbf{x}_{0|t} - \boldsymbol{\Sigma}_t\mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{y}), \tag{17}$$

$$\hat{p}(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_{0|t}) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_t(\mathbf{x}_t, \hat{\mathbf{x}}_{0|t}), \boldsymbol{\Phi}_t\mathbf{I}). \tag{18}$$

$\boldsymbol{\Sigma}_t \in \mathbb{R}^{D\times D}$ is utilized to scale the range-space correction $\mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{y})$ and $\boldsymbol{\Phi}_t \in \mathbb{R}^{D\times D}$ is used to scale the added noise $\sigma_t\boldsymbol{\epsilon}$ in $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$. The choice of $\boldsymbol{\Sigma}_t$ and $\boldsymbol{\Phi}_t$ follows two principles: (i) $\boldsymbol{\Sigma}_t$ and $\boldsymbol{\Phi}_t$ need to assure the total noise variance in $\mathbf{x}_{t-1}$ conforms to the definition in $q(\mathbf{x}_{t-1}|\mathbf{x}_0)$ (Eq. 5) so the total noise can be predicted by $\mathcal{Z}_{\boldsymbol{\theta}}$ and gets removed; (ii) $\boldsymbol{\Sigma}_t$ should be as close as possible to $\mathbf{I}$ to maximize the preservation of the range-space correction $\mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{y})$ so as to maximize the *Consistency*.

Time Travel Trick

As the forward diffusion process is tractable through the transition probabilities $q(x_t|x_{t-1}$ and equivalently $q(x_{t+L}|x_t)$ for all $t + L < T$, where T is the number of timesteps, one can travel back in time and use the knowledge of the "future" to correct the "past".

Authors argue that this modification helps with keeping the null-space harmonized with the rest of the image in the final result.

DDNM$^+$ Algorithm

Algorithm 1 Sampling of DDNM

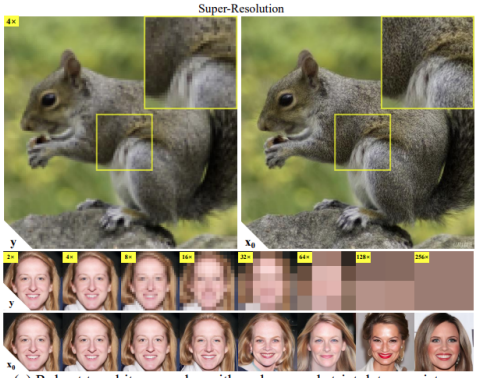
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\sqrt{1-\bar{\alpha}_t}\right)$
  $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{x}_{0|t}$
  $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$
return $\mathbf{x}_0$

Algorithm 2 Sampling of DDNM$^+$

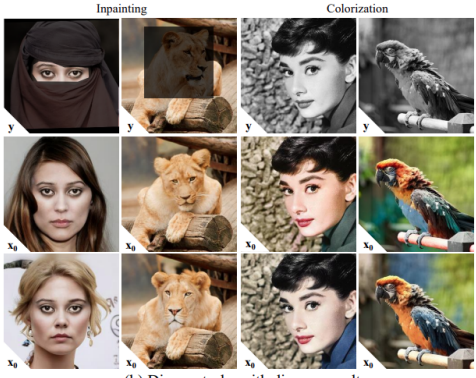
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $L = \min\{T - t, l\}$
  $\mathbf{x}_{t+L} \sim q(\mathbf{x}_{t+L}|\mathbf{x}_t)$
  for $j = L, ..., 0$ **do**
    $\mathbf{x}_{0|t+j} = \frac{1}{\sqrt{\bar{\alpha}_{t+j}}} \left(\mathbf{x}_{t+j} - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_{t+j}, t+j)\sqrt{1-\bar{\alpha}_{t+j}}\right)$
    $\hat{\mathbf{x}}_{0|t+j} = \mathbf{x}_{0|t+j} - \boldsymbol{\Sigma}_{t+j}\mathbf{A}^\dagger(\mathbf{A}\mathbf{x}_{0|t+j} - \mathbf{y})$
    $\mathbf{x}_{t+j-1} \sim \hat{p}(\mathbf{x}_{t+j-1}|\mathbf{x}_{t+j}, \hat{\mathbf{x}}_{0|t+j})$
return $\mathbf{x}_0$

However the official implementation differs slightly from this pseudo-code and the number of NFE is not $T \cdot L$ but rather $T \cdot (r + 1)$ where r is the number of repeats. I can summarize this as: do l diffusion steps ($t \to t - l$), go back in time by l ($t - l \to t$), do l diffusion steps once again, repeat this loop r times, than go further.

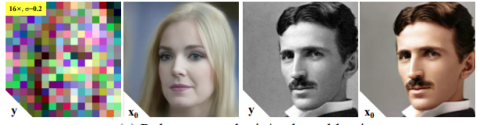
DDNM$^+$ Qualitative Results



(a) Robust to arbitrary scales with realness and strict data consistency



(b) Diverse tasks with diverse results



(c) Robust to synthetic/real-world noise

(d) Flexible in solving complex degradations

DDNM$^+$ Quantitative Results

CelebA	16× SR σ=0.2	C σ=0.2	CS ratio=25% σ=0.2	32× SR	C	CS ratio=10%
Method	PSNR↑/SSIM↑/FID↓	FID↓	PSNR↑/SSIM↑/FID↓	PSNR↑/SSIM↑/FID↓	FID↓	PSNR↑/SSIM↑/FID↓
DDNM	13.10 / 0.2387 / 281.45	216.74	17.89 / 0.4531 / 82.81	17.55 / 0.437 / 39.37	22.79	15.74/ 0.275 / 110.7
DDNM$^+$	**19.44 / 0.712 / 58.31**	**46.11**	**25.02 / 0.868 / 51.35**	**18.44 / 0.501 / 37.50**	**18.23**	**26.33 / 0.741 / 47.93**

Table 2: Ablation study on denoising improvements (*left*) and the time-travel trick (*right*). C represents the colorization task. σ denotes the noise variance on $\mathbf{y}$.

Mask-Shift Trick

Figure 5: 4×SR using Mask-Shift trick, DDNM. Input size: 64×256; output size: 256×1024.

Here we propose a simple but effective trick to perfectly solve this problem. Let's take the above example. First, we divide $\mathbf{y}$ into 8 parts $[\mathbf{y}^{(0)}, ..., \mathbf{y}^{(7)}]$, each part is a 64×32 image. In the first turn, we take $[\mathbf{y}^{(0)},\mathbf{y}^{(1)}]$ as the input and use DDNM to get the SR result $\mathbf{x}$, we further divide it as $[\mathbf{x}^{(0)},\mathbf{x}^{(1)}]$. Next, we need to run 6 turns of DDNM. For the ith turn, we use $[\mathbf{y}^{(i)},\mathbf{y}^{(i+1)}]$ as the input and divide the intermediate result $\hat{\mathbf{x}}_{0|t}$ as $[\hat{\mathbf{x}}_{0|t}^{(left)}, \hat{\mathbf{x}}_{0|t}^{(right)}]$ and replace its left by

$$\hat{\mathbf{x}}_{0|t} = [\mathbf{x}^{(i)}, \hat{\mathbf{x}}_{0|t}^{(right)}], \tag{20}$$

DDNM Test of Time

Table 2. **4× super-resolution** results w.r.t different filters. We report FID and Classifier Accuracy (CA, unit:%) on a pre-trained ResNet50. In all Tables 2 to 5, dark-colored rows denote methods requiring additional information such as corruption operators, as opposed to conditional diffusion models like Palette and our I^{2}SB.

Filter	Method	FID ↓	CA↑
Pool	DDRM (Kawar et al., 2022a)	14.8	64.6
	DDNM (Wang et al., 2022b)	9.9	67.1
	IIGDM (Song et al., 2022)	3.8	72.3
	ADM (Dhariwal & Nichol, 2021)	3.1	**73.4**
	CDSB (Shi et al., 2022)	13.0	61.3
	I^{2}SB (Ours)	**2.7**	71.0
Bicubic	DDRM (Kawar et al., 2022a)	21.3	63.2
	DDNM (Wang et al., 2022b)	13.6	65.5
	IIGDM (Song et al., 2022)	3.6	**72.1**
	ADM (Dhariwal & Nichol, 2021)	14.8	66.7
	CDSB (Shi et al., 2022)	13.6	61.0
	I^{2}SB (Ours)	**2.8**	70.7

Table 4. **Inpainting** results w.r.t different masks.

Mask	Method	FID-10k ↓	CA↑
Center 128×128	DDRM (Kawar et al., 2022a)	24.4	62.1
	IIGDM (Song et al., 2022)	7.3	**72.6**
	DDNM (Wang et al., 2022b)	15.1	55.9
	Palette (Saharia et al., 2022)	6.1	63.0
	CDSB (Shi et al., 2022)	50.5	49.6
	I^{2}SB (Ours)	**4.9**	66.1
Freeform 10%-20%	DDRM (Kawar et al., 2022a)	9.7	67.6
	DDNM (Wang et al., 2022b)	3.2	73.6
	Palette (Saharia et al., 2022)	4.0	73.7
	CDSB (Shi et al., 2022)	8.5	71.2
	I^{2}SB (Ours)	**2.9**	**74.9**
Freeform 20%-30%	DDRM (Kawar et al., 2022a)	8.6	71.9
	IIGDM (Song et al., 2022)	5.3	**75.3**
	DDNM (Wang et al., 2022b)	4.2	70.8
	Palette (Saharia et al., 2022)	4.1	71.8
	CDSB (Shi et al., 2022)	16.5	64.5
	I^{2}SB (Ours)	**3.2**	73.4

Table 5. **Deblurring** results w.r.t different kernels.

Kernel	Method	FID-10k ↓	CA↑
Uniform	DDRM (Kawar et al., 2022a)	9.9	68.0
	DDNM (Wang et al., 2022b)	**3.0**	**75.5**
	Palette (Saharia et al., 2022)	4.1	74.0
	CDSB (Shi et al., 2022)	15.5	65.1
	I^{2}SB (Ours)	3.9	73.7
Gaussian	DDRM (Kawar et al., 2022a)	6.1	72.5
	DDNM (Wang et al., 2022b)	**2.9**	**75.6**
	Palette (Saharia et al., 2022)	3.1	75.4
	CDSB (Shi et al., 2022)	7.7	71.1
	I^{2}SB (Ours)	3.0	75.0

Figure: Relevant evaluation results from the I^{2}SB paper

QR Codes

(a) Arxiv

(b) Open Review

(c) GitHub

References

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2022).
Diffusion posterior sampling for general noisy inverse problems.
arXiv preprint arXiv:2209.14687.

Ho, J., Jain, A., and Abbeel, P. (2020).
Denoising diffusion probabilistic models.
Advances in neural information processing systems, 33:6840–6851.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. (2023).
I ′2 sb: Image-to-image schr\" odinger bridge.
arXiv preprint arXiv:2302.05872.

The End