Deep Equilibrium Diffusion Restoration with Parallel Sampling

(Supplementary Materials)

Jiezhang Cao[1], Yue Shi[1,2], Kai Zhang[3], Yulun Zhang[2], Radu Timofte[1,4], Luc Van Gool[1,5]
[1]ETH Zürich, [2]Shanghai Jiao Tong University, [3]Nanjing University, [4]University of Würzburg, [5]KU Leuven

Organization. In this paper, we organize our supplementary materials as follows. In Section A, we provide detailed proofs of our proposed Proposition 1. In Section C, we provide more results of our method. In Section B, we provide more visual comparisons. In Section D, we provide more details and visual results of initialization optimization. In Section E, the limitations and future work of our proposed method are discussed.

A. Proof of Proposition 1

Proof Based on existing diffusion model-based IR (*e.g.*, [68]), the estimated $\boldsymbol{x}_0$ at time-step t can be formulated as:

$$\boldsymbol{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right). \tag{14}$$

We fix the range-space as $\boldsymbol{A}^\dagger\boldsymbol{y}$ and leave the null-space unchanged, then $\boldsymbol{x}_0$ at time-step t can be further estimated by

$$\begin{aligned}
\hat{\boldsymbol{x}}_{0|t} =& \left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_{0|t} + \boldsymbol{A}^\dagger\boldsymbol{y} \\
=& \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t)\right) + \boldsymbol{A}^\dagger\boldsymbol{y} \\
=& \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{A}^\dagger\boldsymbol{y},
\end{aligned} \tag{15}$$

where the second line is based on Eqn. (14). Based on DDIM [61], $\boldsymbol{x}_{t-1}$ can be updated by

$$\begin{aligned}
\boldsymbol{x}_{t-1} =& \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_{0|t} + \gamma_t\sqrt{1-\bar{\alpha}_t}\eta\boldsymbol{\epsilon}_t + \gamma_t\sqrt{1-\bar{\alpha}_t}\sqrt{1-\eta^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) \\
:=& \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_{0|t} + \gamma_t c_t^1\boldsymbol{\epsilon}_t + \gamma_t c_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) \\
=& \sqrt{\bar{\alpha}_{t-1}}\left[\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{A}^\dagger\boldsymbol{y}\right] + \gamma_t c_t^1\boldsymbol{\epsilon}_t + \gamma_t c_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) \\
=& \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \left(\gamma_t c_t^2 - \frac{\sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{A}^\dagger\boldsymbol{y} + \gamma_t c_t^1\boldsymbol{\epsilon}_t \\
:=& \boldsymbol{\Psi}_t^1\boldsymbol{x}_t + \boldsymbol{\Psi}_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{\Psi}_t^3\boldsymbol{y} + \boldsymbol{\Psi}_t^4\boldsymbol{\epsilon}_t,
\end{aligned} \tag{16}$$

where ϵ_t is sampled from standard Gaussian distribution $\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, and γ_t is a parameter which can be set as 1. The third line is based on Eqn. (15). Let $c_t^1 = \sqrt{1-\bar{\alpha}_t}\eta$ and $c_t^2 = \sqrt{1-\bar{\alpha}_t}\sqrt{1-\eta^2}$, and we define $\boldsymbol{\Psi}_t^i, i = 1,\ldots,4$ as

$$\boldsymbol{\Psi}_t^1 = \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right), \quad \boldsymbol{\Psi}_t^2 = \gamma_t c_t^2 - \frac{\sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right), \quad \boldsymbol{\Psi}_t^3 = \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{A}^\dagger, \quad \boldsymbol{\Psi}_t^4 = \gamma_t c_t^1.$$

Based on the formulation of $\boldsymbol{x}_{t-1}$, we can write $\boldsymbol{x}_{t-2}$ as

$$\begin{aligned}
\boldsymbol{x}_{t-2} =& \boldsymbol{\Psi}_{t-1}^1\boldsymbol{x}_{t-1} + \boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1},t-1) + \boldsymbol{\Psi}_{t-1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1} \\
=& \boldsymbol{\Psi}_{t-1}^1\left(\boldsymbol{\Psi}_t^1\boldsymbol{x}_t + \boldsymbol{\Psi}_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{\Psi}_t^3\boldsymbol{y} + \boldsymbol{\Psi}_t^4\boldsymbol{\epsilon}_t\right) + \boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1},t-1) + \boldsymbol{\Psi}_{t-1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1} \\
=& \boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^1\boldsymbol{x}_t + \boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^4\boldsymbol{\epsilon}_t + \boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1},t-1) + \boldsymbol{\Psi}_{t-1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1} \\
=& \boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^1\boldsymbol{x}_t + \left(\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1},t-1)\right) + \left(\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^3\boldsymbol{y}\right) + \left(\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^4\boldsymbol{\epsilon}_t + \boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1}\right) \\
=& \frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \sum_{s=t-2}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\left(\boldsymbol{\Psi}_{s+1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{s+1},s+1) + \boldsymbol{\Psi}_{s+1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{s+1}^4\boldsymbol{\epsilon}_{s+1}\right) \\
&+ \boldsymbol{A}^\dagger\boldsymbol{A}\left(\boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1},t-1) + \boldsymbol{\Psi}_{t-1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1}\right) \\
:=& \frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \sum_{s=t-2}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I}-\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{z}_{s+1} + \boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{z}_{t-1},
\end{aligned} \tag{17}$$

where the second line follows Eqn. (16), the third line holds by the definition of $\boldsymbol{\Psi}_t^i$, and in the last line, we define $\boldsymbol{z}_s$ as:

$$\begin{aligned}\boldsymbol{z}_s =&\boldsymbol{\Psi}_s^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_s, s) + \boldsymbol{\Psi}_s^3\boldsymbol{y} + \boldsymbol{\Psi}_s^4\boldsymbol{\epsilon}_s \\ =&\left(\gamma_s c_s^2 - \frac{\sqrt{1-\bar{\alpha}_s}}{\sqrt{\alpha_s}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_s, s) + \sqrt{\bar{\alpha}_{s-1}}\boldsymbol{A}^\dagger\boldsymbol{y} + \gamma_s c_s^1\boldsymbol{\epsilon}_s, \end{aligned} \tag{18}$$

where the second line is based on the definitions of c_t^2 and $\boldsymbol{\Psi}_t^i, i = 2, 3, 4$. Similarly, we further write $\boldsymbol{x}_{t-3}$ as

$$\begin{aligned}\boldsymbol{x}_{t-3} =&\boldsymbol{\Psi}_{t-2}^1\boldsymbol{x}_{t-2} + \boldsymbol{\Psi}_{t-2}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-2}, t-2) + \boldsymbol{\Psi}_{t-2}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-2}^4\boldsymbol{\epsilon}_{t-2} \\ =&\boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^1\boldsymbol{x}_t + \left(\boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, t-1) + \boldsymbol{\Psi}_{t-2}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-2}, t-2)\right) \\ &+ \left(\boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-2}^3\boldsymbol{y}\right) + \left(\boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^1\boldsymbol{\Psi}_t^4\boldsymbol{\epsilon}_t + \boldsymbol{\Psi}_{t-2}^1\boldsymbol{\Psi}_{t-1}^4\boldsymbol{\epsilon}_{t-1} + \boldsymbol{\Psi}_{t-2}^4\boldsymbol{\epsilon}_{t-2}\right) \\ =&\frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \sum_{s=t-3}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\left(\boldsymbol{\Psi}_{s+1}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{s+1}, s+1) + \boldsymbol{\Psi}_{s+1}^3\boldsymbol{y} + \boldsymbol{\Psi}_{s+1}^4\boldsymbol{\epsilon}_{s+1}\right) \\ &+ \boldsymbol{A}^\dagger\boldsymbol{A}\left(\boldsymbol{\Psi}_{t-2}^2\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-2}, t-2) + \boldsymbol{\Psi}_{t-2}^3\boldsymbol{y} + \boldsymbol{\Psi}_{t-2}^4\boldsymbol{\epsilon}_{t-2}\right) \end{aligned} \tag{19}$$

$$:= \frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \sum_{s=t-3}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{z}_{s+1} + \boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{z}_{t-2}, \tag{20}$$

where the second line is according to the formulation of $\boldsymbol{x}_{t-2}$, and the third line is based on the definition of $\boldsymbol{\Psi}_t^i$. In the line of Eqn. (19), we can derive the following

$$\begin{aligned}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{\Psi}_{t+1}^3\boldsymbol{y} =&\sqrt{\bar{\alpha}_t}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{A}^\dagger\boldsymbol{y} = \boldsymbol{0}, \\ \boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{\Psi}_{T-k+1}^3\boldsymbol{y} =&\sqrt{\bar{\alpha}_{T-k}}\boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{y} = \sqrt{\bar{\alpha}_{T-k}}\boldsymbol{A}^\dagger\boldsymbol{y}, \\ \left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{\Psi}_{t+1}^2 =&\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\left(\gamma_t c_t^2 - \frac{\sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\right) \\ =&\gamma_t c_t^2\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right) - \frac{\sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right) \\ =&\left(\gamma_t c_t^2 - \frac{\sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}}\right)\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right), \\ \boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{\Psi}_{T-k+1}^2 =&\boldsymbol{A}^\dagger\boldsymbol{A}\left(\gamma_{T-k+1}c_{T-k+1}^2 - \frac{\sqrt{\bar{\alpha}_{T-k}(1-\bar{\alpha}_{T-k+1})}}{\sqrt{\bar{\alpha}_{T-k+1}}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\right) \\ =&\gamma_{T-k+1}c_{T-k+1}^2\boldsymbol{A}^\dagger\boldsymbol{A}. \end{aligned}$$

Based on the induction, we can write $\boldsymbol{x}_{T-k}$ as:

$$\boldsymbol{x}_{T-k} = \frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_T}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_T + \sum_{s=T-k}^{T-1}\frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{z}_{s+1} + \boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{z}_{T-k+1}. \tag{21}$$

We complete the proof. Actually, the above equation can also be written as

$$\boldsymbol{x}_{T-k} = \frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_T}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_T + \sum_{s=T-k}^{T-1}\frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A}\right)^{T-k-s}\boldsymbol{z}_{s+1}. \tag{22}$$

□

We extend Proposition 1 to the noisy inverse problem (*i.e.*, $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}_\sigma$) as follows.

Proposition 2 (Parallel sampling for noisy inverse problem) *Given a degradation matrix $\boldsymbol{A}$, a degraded image $\boldsymbol{y}$ and a Gaussian noise image $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and assume that the scale factor $\boldsymbol{\Sigma}_t$ for range-space is a constant diagonal matrix, i.e., $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_t$, for every state t, for $k \in [1, \dots, T]$, the state $\boldsymbol{x}_{T-k}$ can be predicted by previous states $\{\boldsymbol{x}_{T-k+1}, \dots, \boldsymbol{x}_T\}$, i.e.,*

$$\boldsymbol{x}_{T-k} = \frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_T}} \left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^k \boldsymbol{x}_T + \sum_{s=T-k}^{T-1} \frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_s}} \left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{T-k-s} \boldsymbol{z}_{s+1}, \tag{23}$$

where $\boldsymbol{z}_s = c_s^0 \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_s, s) + \sqrt{\bar{\alpha}_{s-1}}\boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y} + c_s^1\boldsymbol{\epsilon}_s$, the coefficients are defined as $c_s^0 := c_s^2 - \sqrt{(1-\bar{\alpha}_s)/\alpha_s}(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})$, $c_s^1 := \sqrt{1-\bar{\alpha}_s}\eta$ and $c_s^2 := \sqrt{1-\bar{\alpha}_s}\sqrt{1-\eta^2}, 0 \le \eta < 1$.

Proof For the noisy inverse problem, [68] estimates $\hat{\boldsymbol{x}}_{0|t}$ as

$$\begin{aligned}
\hat{\boldsymbol{x}}_{0|t} =& \boldsymbol{x}_{0|t} - \boldsymbol{\Sigma}_t\boldsymbol{A}^\dagger(\boldsymbol{A}\boldsymbol{x}_{0|t} - \boldsymbol{y}) \\
=& (\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{x}_{0|t} + \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y} && (24)\\
=& \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y} && (25)\\
=& \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{x}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})\cdot\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y}. && (26)
\end{aligned}$$

Let $c_t^0 := c_t^2 - \sqrt{(1-\bar{\alpha}_t)/\alpha_t}(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A})$, $c_t^1 := \sqrt{1-\bar{\alpha}_t}\eta$ and $c_t^2 := \sqrt{1-\bar{\alpha}_t}\sqrt{1-\eta^2}$, we define $\boldsymbol{z}_t$ with $\gamma_t = 1$ as

$$\begin{aligned}
\boldsymbol{z}_t =& \left(c_t^2 - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y} + c_t^1\boldsymbol{\epsilon}_t && (27)\\
:=& c_t^0\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{y} + c_t^1\boldsymbol{\epsilon}_t. && (28)
\end{aligned}$$

Similar to Proposition 1, based on DDIM [61], $\boldsymbol{x}_{t-1}$, $\boldsymbol{x}_{t-2}$ and $\boldsymbol{x}_{t-3}$ can be calculated by

$$\begin{aligned}
\boldsymbol{x}_{t-1} =& \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)\boldsymbol{x}_t + \boldsymbol{z}_t && (29)\\
=& \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^1\boldsymbol{x}_t + \sum_{s=t-1}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{t-1-s}\boldsymbol{z}_{s+1} && (30)\\
\boldsymbol{x}_{t-2} =& \frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^2\boldsymbol{x}_t + \sum_{s=t-2}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-2}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{t-2-s}\boldsymbol{z}_{s+1} && (31)\\
\boldsymbol{x}_{t-3} =& \frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^3\boldsymbol{x}_t + \sum_{s=t-3}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-3}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{t-3-s}\boldsymbol{z}_{s+1}. && (32)
\end{aligned}$$

According to the induction, we can write

$$\boldsymbol{x}_{t-k} = \frac{\sqrt{\bar{\alpha}_{t-k}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^k\boldsymbol{x}_t + \sum_{s=t-k}^{t-1}\frac{\sqrt{\bar{\alpha}_{t-k}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{t-k-s}\boldsymbol{z}_{s+1}. \tag{33}$$

Let t be T, the $\boldsymbol{x}_{T-k}$ can be estimated by

$$\boldsymbol{x}_{T-k} = \frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_T}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^k\boldsymbol{x}_T + \sum_{s=T-k}^{T-1}\frac{\sqrt{\bar{\alpha}_{T-k}}}{\sqrt{\bar{\alpha}_s}}\left(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{A}^\dagger\boldsymbol{A}\right)^{T-k-s}\boldsymbol{z}_{s+1}. \tag{34}$$

We complete the proof of Proposition 2.

□

B. More Implementation Details and Quantitative Results

More implementation details. Our method is a zero-shot diffusion model-based IR method and thus does not need training DEQ function and diffusion models. Algorithm 1 is a standard implementation of Anderson acceleration. In Algorithm 1, we sample a noise $\boldsymbol{x}_T$ from Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and copy it into multiple copies. The noises are the same and they are the initializations in the fixed point solver, denoted by $\boldsymbol{x}^0_{0:T-1}$, as shown in Figure 2. After K iterations, the sampling chain will converge to $\boldsymbol{x}^*_{0:T-1}$ where $\boldsymbol{x}^*_0$ is our desired result. In Algorithm 2, we provide an initialization optimization via DEQ inversion. The initialization is the same as Algorithm 1. For every initialization, we first use $\mathrm{RootSolve}(\cdot)$ to obtain $\boldsymbol{x}^*_0$ and its loss. Then we use the DEQ inversion Eqn. (13) to update $\boldsymbol{x}_T$. This update process terminates once the gradient norm falls below a default threshold within a sufficiently large step S. To further improve the performance, we use [18] for SR on ImageNet, and use [84, 86] for colorization, to provide the prior information in the intermediate state.

Super-Resolution. In the main paper, we present the SR results for $\times 2$ and $\times 4$ scales. In this experiment, we extend to other scales. We compare our method with DPS [20], DiffPIR [87], DDRM [41] and DDNM [68] on ImageNet. Additionally, we use bicubic upscaling as a baseline for SR. The quantitative results are shown in Table B1. For larger scales $\times 8$ and $\times 16$, our method demonstrates significant superiority over most methods across various metrics. Specifically, when compared with the competitive IR method DDNM, our method surpasses it by an LPIPS margin of up to 0.024 and a PSNR margin of up to 1.55 dB. Furthermore, our method requires only 15 sampling steps, compared to other methods.

Methods	$\times 8$ SR			$\times 16$ SR		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
bicubic	21.45	0.512	0.504	19.31	0.431	0.648
DDRM [41]	22.83	0.578	0.444	20.05	0.468	0.577
DPS [20]	18.30	0.382	0.520	16.40	0.315	0.572
DiffPIR [87]	20.40	0.440	0.476	17.72	0.347	0.553
DDNM [68]	22.36	0.558	0.414	20.02	0.459	0.563
DeqIR (Ours)	**23.91**	**0.630**	**0.390**	**21.04**	**0.510**	**0.546**

Table B1. Comparisons of zero-shot **SR** methods on ImageNet.

Deblurring. In the main paper, we have provided deblurring results for Gaussian and anisotropic kernels. In this experiment, we further consider a uniform kernel to evaluate the performance of all models. Specifically, we evaluate the zero-shot IR methods, including DDRM [41], DPS [20], DiffPIR [87], and DDNM [68], and employ $\boldsymbol{A}^\dagger \boldsymbol{y}$ as a baseline. The quantitative results in Table B2 demonstrate that our method outperforms others on all datasets. In comparison with DDNM [68], our method exhibits a PSNR improvement of up to 0.65 dB.

Methods	ImageNet			CelebA-HQ		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
$\boldsymbol{A}^\top \boldsymbol{y}$	18.76	0.455	0.574	20.30	0.652	0.446
DDRM [41]	36.97	0.953	0.080	40.72	0.972	0.060
DPS [20]	20.68	0.497	0.414	25.44	0.709	0.260
DiffPIR [87]	22.71	0.494	0.499	28.11	0.777	0.264
DDNM [68]	38.66	0.968	0.050	43.13	0.981	0.044
DeqIR (Ours)	**39.31**	**0.967**	**0.047**	**43.32**	**0.982**	**0.042**

Table B2. Comparisons of zero-shot **deblurring (uniform)** methods.

Inpainting. In the main paper, we provided quantitative results for the image inpainting task specifically on CelebA-HQ dataset. In this experiment, we extend our evaluation to ImageNet, comparing our method against state-of-the-art (SOTA) inpainting methods, including DDRM [41] and DDNM [68]. Additionally, we establish $\boldsymbol{A}^\dagger \boldsymbol{y}$ as a baseline for comparison. Furthermore, we explore various inpainting masks, such as text and stripe masks, and present the corresponding results on ImageNet in Table B3. Notably, our method exhibits significant performance improvement over the diffusion model DDRM [41]. Moreover, with a fixed number of timesteps (25), our method surpasses DDNM [68], albeit falls short compared to DDNM with 100 timesteps, as it incorporates more sampling information.

Methods	Text mask			Stripe mask		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
$\boldsymbol{A}^\dagger \boldsymbol{y}$	14.41	0.659	0.461	8.82	0.838	0.197
DDRM [41]	32.22	0.952	0.052	24.07	0.736	0.358
DDNM-25 [68]	32.71	0.958	0.039	25.58	0.788	0.280
DDNM-100 [68]	**33.94**	**0.964**	**0.031**	27.71	**0.838**	**0.197**
DeqIR (Ours)	33.03	0.958	0.046	**28.53**	0.816	0.213

Table B3. Comparisons of zero-shot **inpainting** methods on ImageNet.

More evaluation metrics (FID). In addition to PSNR, SSIM and LPIPS, we also consider FID, which is a commonly used evaluation metric for assessing the quality of generated images. Although FID is a useful metric in many real-world scenarios, there are certain circumstances where it may not be appropriate. In particular, I do not use FID to evaluate the image quality for super-resolution and deblurring in the main paper because the ImageNet and CelebA-HQ datasets with 100 classes are relatively small, and FID requires a sufficiently large dataset to accurately estimate the statistics of the data distribution. If the dataset is small, the FID score may not be reliable due to high variance in the estimated statistics. In contrast, we use FID in the colorization task since we can manipulate inputs using various Gaussian noises by altering seeds. Nevertheless, we retain the ability to compute FID scores for super-resolution and deblurring tasks. In general, lower FID indicates higher quality of images. As shown in Table B4, our method has superior FID performance compared to other approaches. Furthermore, we observe a similar trend between FID and LPIPS in our comparisons, as illustrated in Table 1 in the main paper. Hence, we can confidently rely on these FID results for assessing super-resolution and deblurring tasks.

Methods	Bicubic SR		Deblurring	
	2×	4×	Gaussian	anisotropic
Baseline	48.94	134.11	117.87	182.49
DGP [55]	172.72	273.65	231.00	267.71
DPS [20]	142.31	166.36	121.60	119.37
DiffPIR [87]	43.19	96.28	73.50	145.52
DDRM [41]	26.56	94.92	6.43	10.82
DDNM [68]	20.87	86.29	2.57	5.61
DeqIR (Ours)	**13.89**	**59.70**	**2.41**	**5.40**

Table B4. FID (↓) results of SR and deblurring tasks on ImageNet.

Application of Proposition 2. Based on Proposition 2, we apply our method to the task of restoring noisy images. Specifically, our approach addresses noisy super-resolution for scaling factors of $2\times, 4\times$, and $8\times$, denoted as Ours-σ. In our experiments, we set the noise level to $\sigma = 0.2$. As shown in Table B5, our method consistently achieves higher PSNR values compared to DDNM [68] across various scaling factors. Visual comparisons are provided in Figure B1, illustrating our method's capability to reduce noise and partially restore textures, leveraging the derived sampling formulation in Proposition 2.

Scale	DDNM [68]	**Ours-σ**
2×SR	24.91	**25.23**
4×SR	22.13	**22.40**
8×SR	19.70	**19.95**

Table B5. PSNR for noisy SR

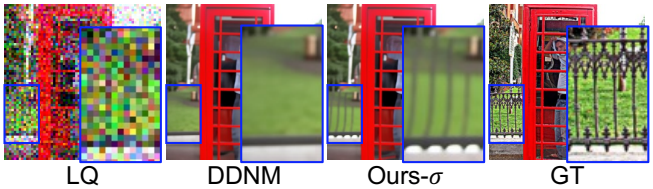

Figure B1. Results for noisy 4×SR.

GPU VRAM usage. In addition to comparing running times as presented in the main paper, we also analyze the running time, GPU memory usage, and PSNR values of our method to understand the trade-offs between image quality and computational costs. We conducted experiments with different timesteps for anisotropic deblurring on the ImageNe dataset, as shown in Table B6. As timesteps increase, image quality improves, albeit at the expense of increased inference time and GPU memory usage. Notably, for a timestep of 20, our method achieves the highest PSNR value. However, it also exhibits the longest inference time and requires the largest GPU memory allocation. Given constraints on available memory, one may choose a timestep of 15. This choice allows for the generation of images with a comparable PSNR to that of a timestep of 20, while having more efficient inference times.

Methods	Ours-10	Ours-15	Ours-20
Time (s)	12.11	16.53	21.19
Memory (G)	12.07	16.32	18.06
PSNR (dB)	38.58	39.21	39.47

Table B6. Comparisons of time, memory and PSNR.

C. More Qualitative Results

C.1. More Results on Super-Resolution

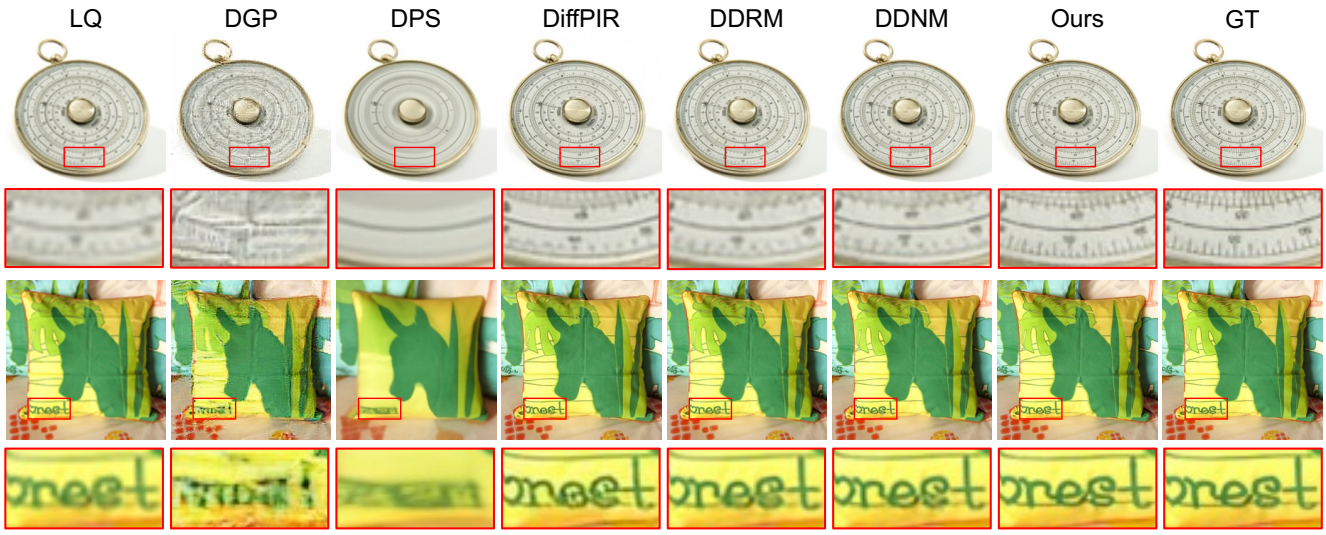

Figure C2. Qualitative results of image super-resolution ($\times 2$) methods on ImageNet.

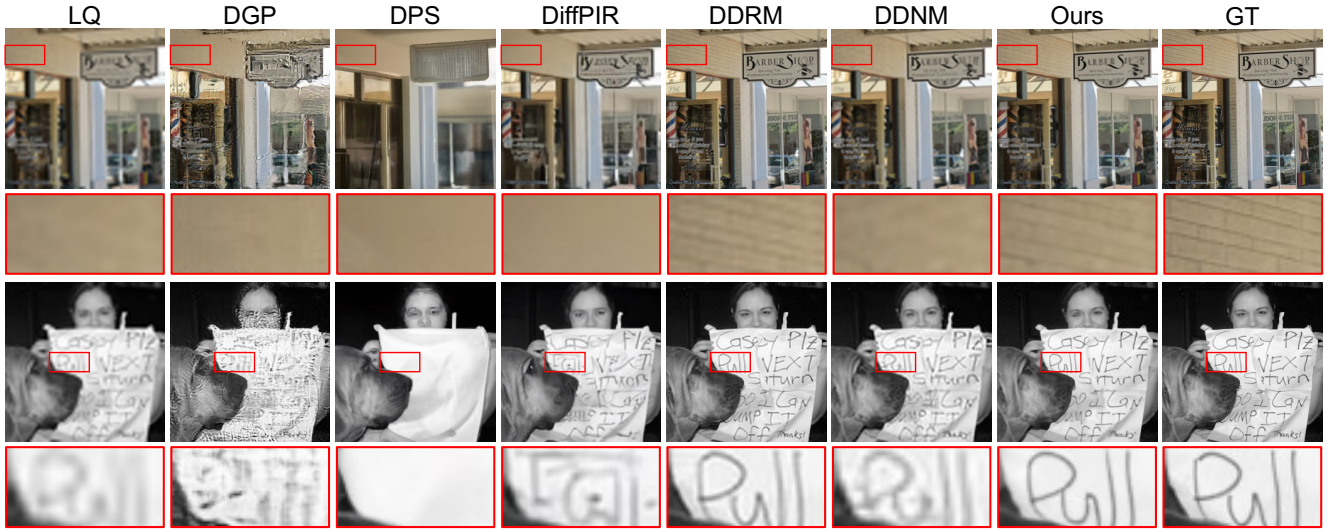

Figure C3. Qualitative results of image super-resolution ($\times 4$) methods on ImageNet.

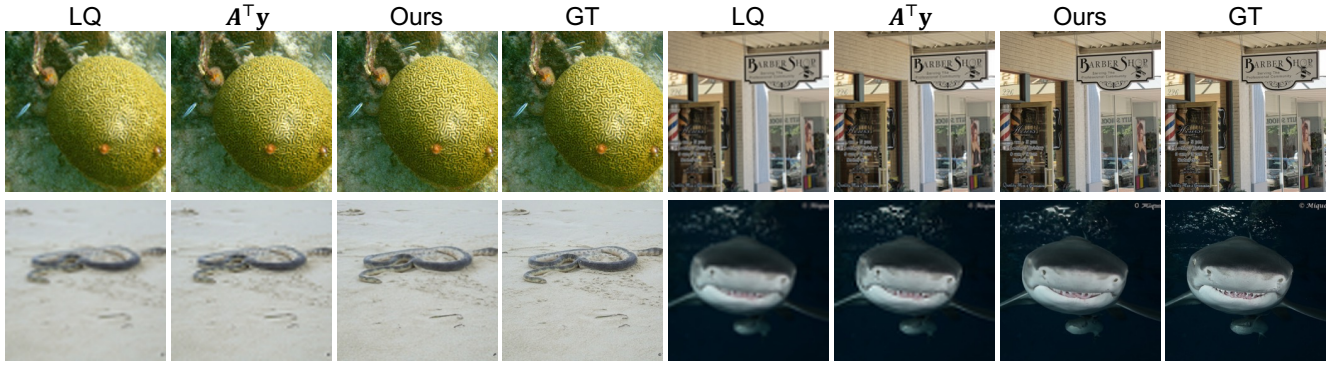

Figure C4. Our qualitative results of image super-resolution ($\times 2$ (above) and $\times 4$ (bottom)) on ImageNet.

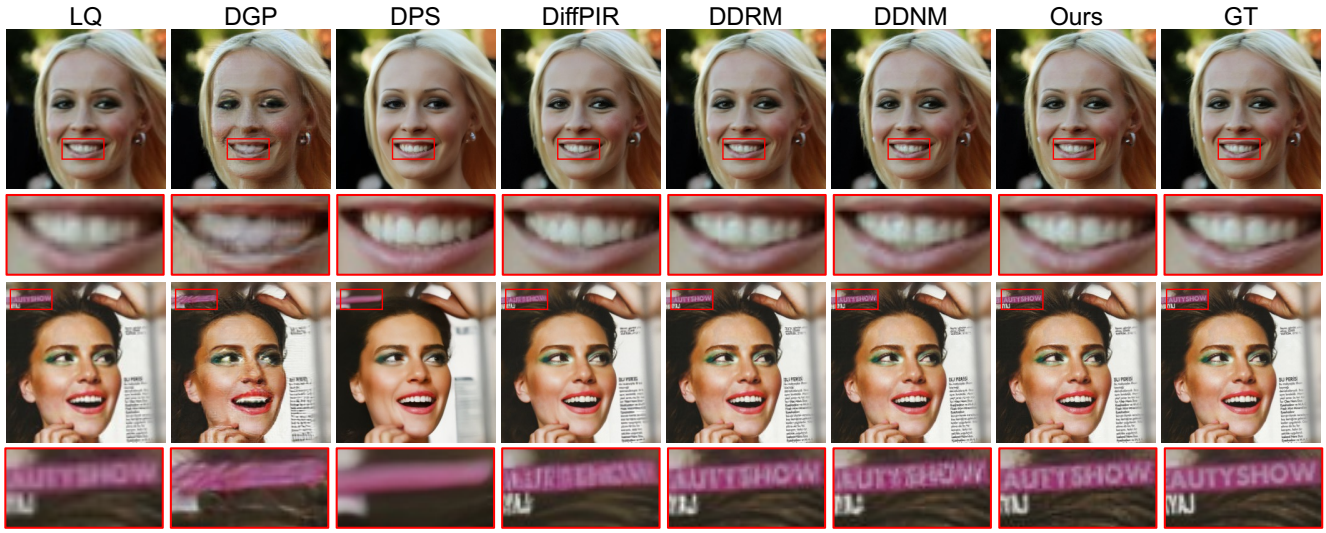

Figure C5. Qualitative results of image super-resolution ($\times 2$) methods on CelebA-HQ.

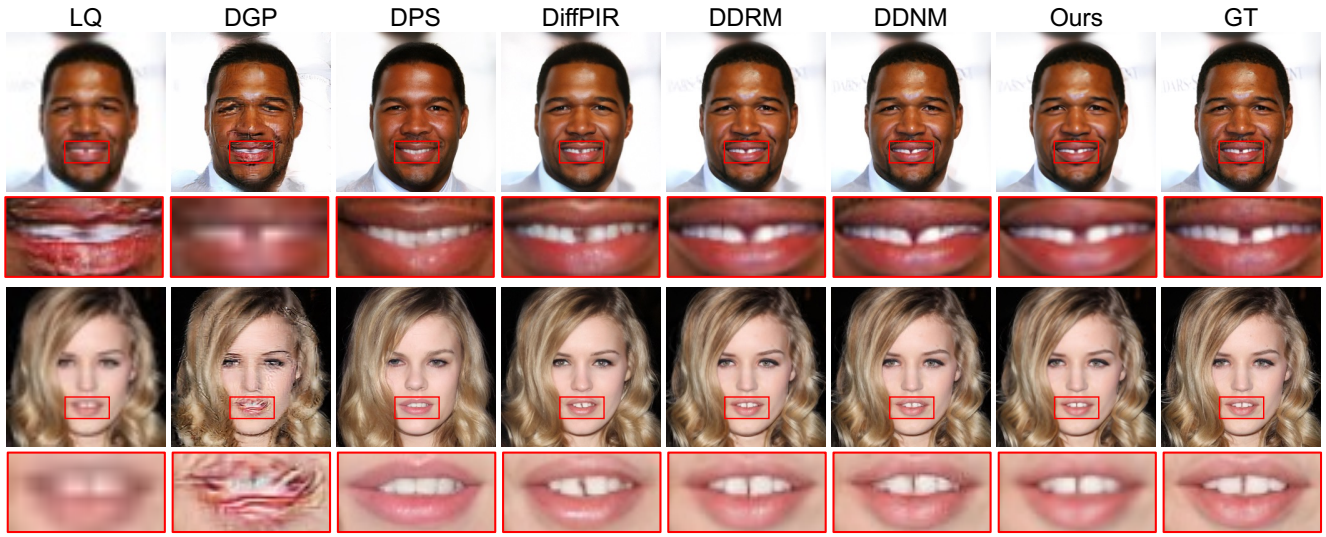

Figure C6. Qualitative results of image super-resolution ($\times 4$) methods on CelebA-HQ.

Figure C7. Our qualitative results of image super-resolution ($\times 2$ (above) and $\times 4$ (bottom)) on CelebA-HQ.

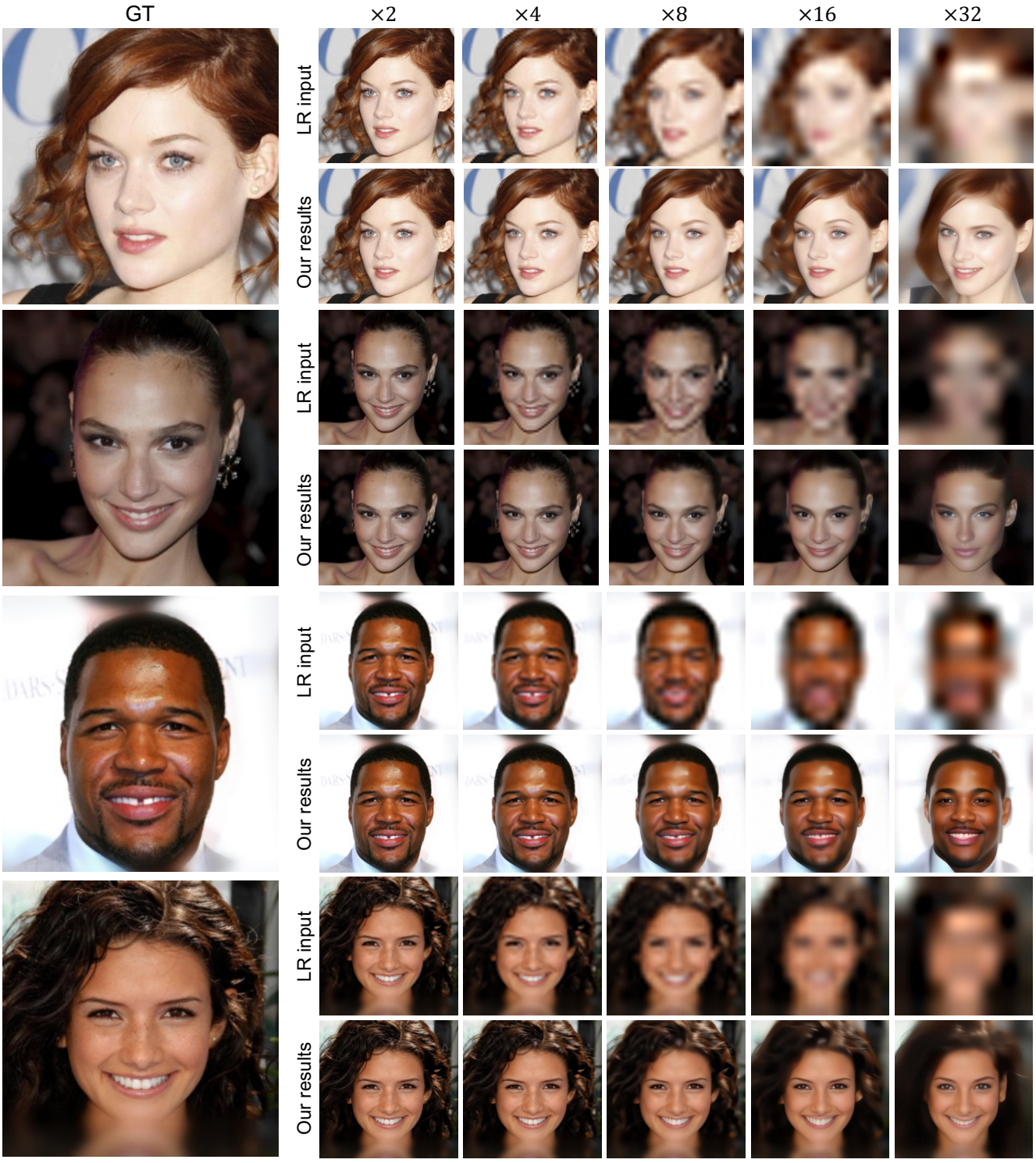

Figure C8. Our qualitative results of image super-resolution for different scales on CelebA-HQ.

C.2. More Results on Image Deblurring

Figure C9. Our qualitative results of image deblurring (gauss) on ImageNet.

Figure C10. Our qualitative results of image deblurring (anisotropic) on ImageNet.

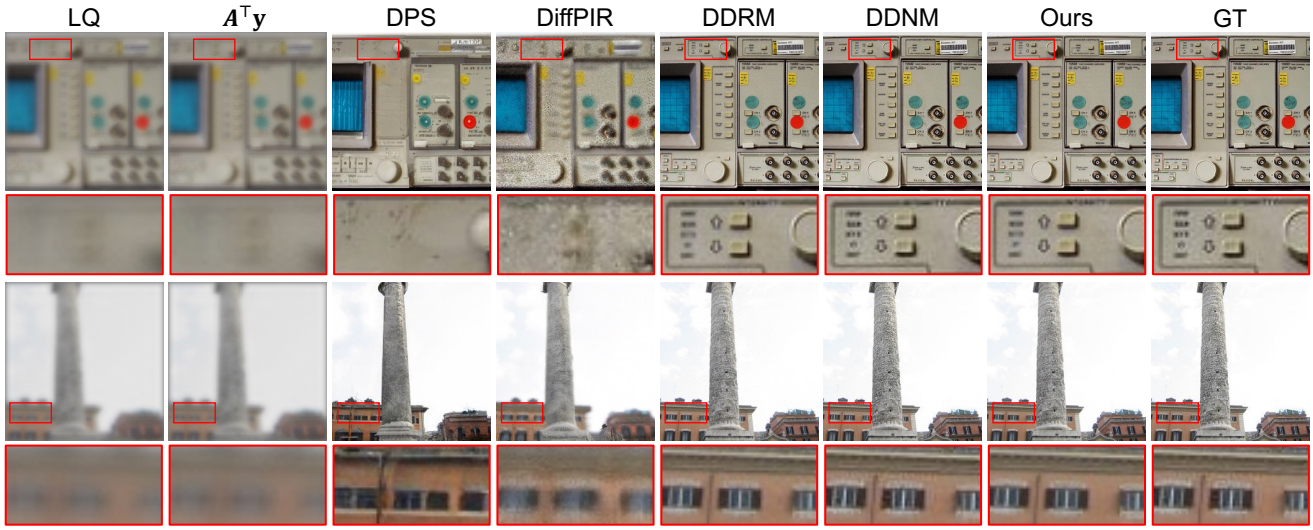

Figure C11. Our qualitative results of image deblurring (uniform) on ImageNet.

C.3. More Results on Image Inpainting

Figure C12. Our qualitative results of image inpainting on ImageNet.

Figure C13. Our qualitative results of image inpainting on CelebA-HQ.

C.4. More Results on Image Colorization

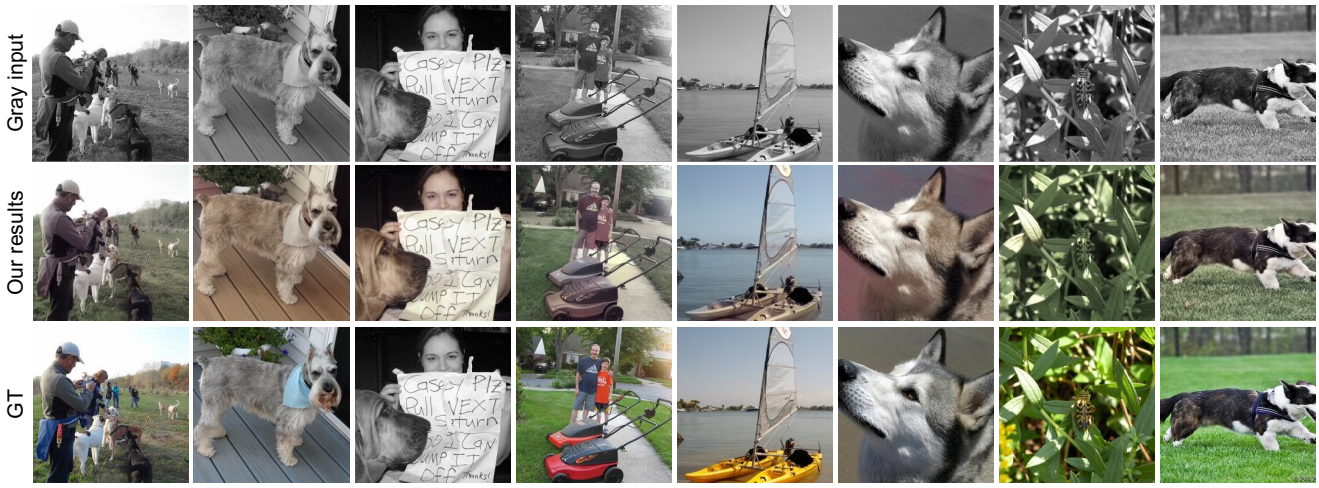

Figure C14. Our qualitative results of image colorization on ImageNet.

Figure C15. Our qualitative results of image colorization on CelebA-HQ.

C.5. More Results on Old Photo Restoration

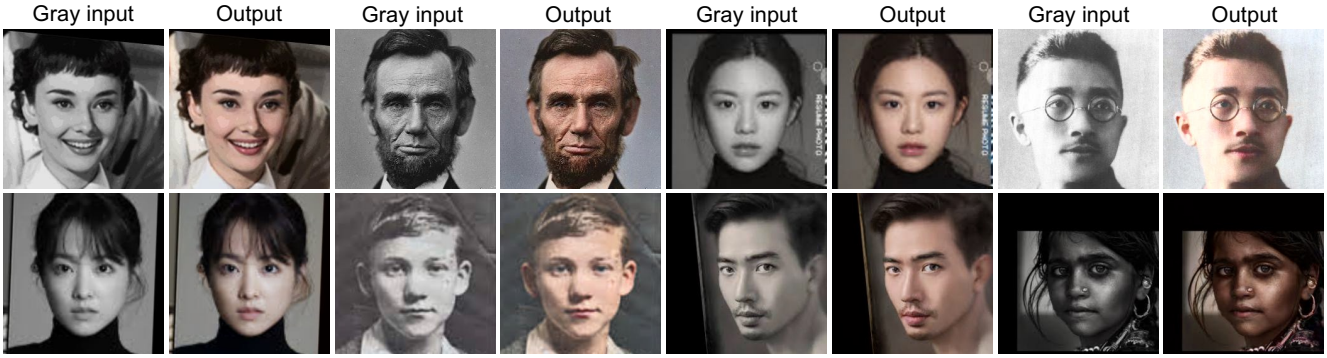

Figure C16. Our qualitative results of image colorization on real-world images.

C.6. Results on Other Image Restoration Tasks

In the main paper, we have conducted some typical image restoration (IR) tasks, including super-resolution, image deblurring, image inpainting and colorization. For other IR tasks, we show that our method can be used in the compressed sensing task. Specifically, we use the Walsh-Hadamard sampling matrix with a 0.5 compression ratio. We show the visual results on ImageNet and CelebA-HQ in Figures C17 and C18. As we can see, with the severely compressed inputs, our method is able to recover the high-frequency details while preserving the inherent identity of the content.

Figure C17. Our qualitative results of compressed sensing on ImageNet.

Figure C18. Our qualitative results of compressed sensing on CelebA-HQ.

C.7. Arbitrary Size Image Restoration

In our primary experiments, the input size is set at $3\times256\times256$. However, for real-world applications, the input sizes may vary. To address this variability, we demonstrate the adaptability of our model to accommodate inputs of arbitrary sizes. As an illustration, we use an input size of $3\times256\times1024$, a methodology that can be extended to accommodate diverse sizes. In line with the approaches in [48, 68], we segment larger images into multiple overlapping patches and conduct individual tests on each. Our method differs from [68] as our overlapped patches are smaller, resulting in reduced computational costs. Finally, we consolidate the generated outputs to form conclusive results.

Figure C19. Our qualitative results of arbitrary-size image restoration on real-world images.

D. More Results on Initialization Optimization

In the super-resolution, we set S as 500 for convergence, and we use the ℓ_2 loss to guide the generation process. In the loss, we use the classical IR model (*e.g.*, SwinIR [48]) as the supervision information such that the generated images can be close to the given supervision information. In the colorization, we set S as 2k for convergence, and we use the perceptual loss to guide the generation process. The results are shown in Figures D20 and D21. With the initialization optimization, PSNR can be further improved, and the colorization has guidance to generate according to the reference images.

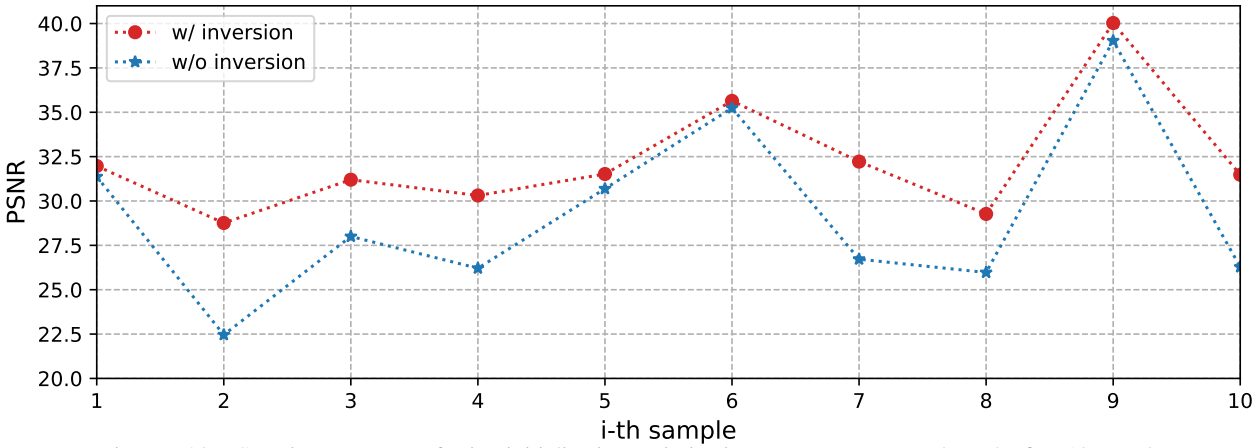

Figure D20. PSNR improvement of using initialization optimization on ImageNet (we show the first 10 samples).

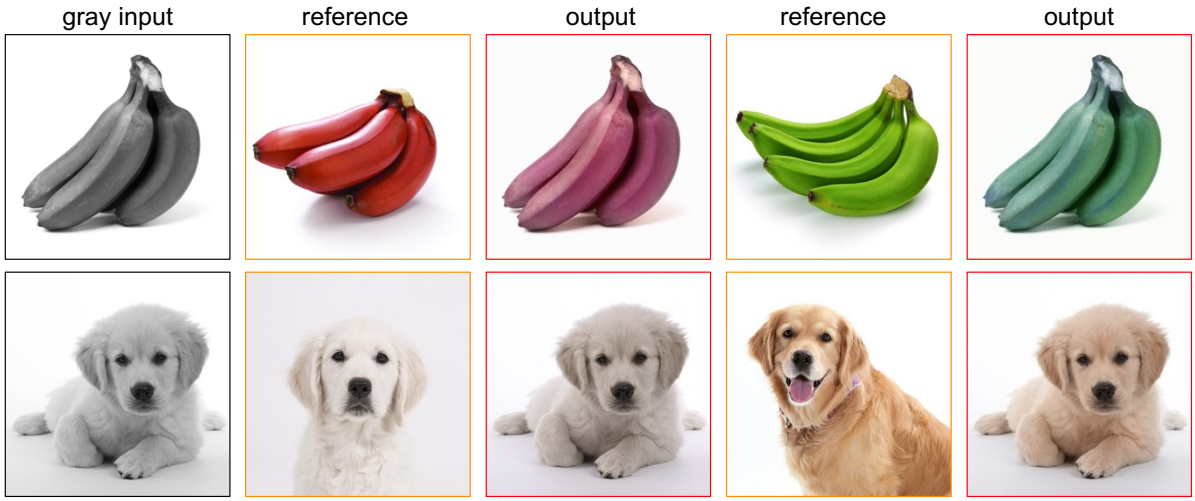

Figure D21. Our qualitative results of colorization with initialization optimization.

E. Limitations and Future Work

There remain many limitations that can be studied in the future.

- Our method processes multiple timestep images in parallel, leading to large memory in GPU. It is necessary to reduce the memory and boost the inference time.
- Our parallel sampling requires explicit forms of the degradation matrix $\boldsymbol{A}$ which is linear. For unknown, complex and non-linear, we need to design pseudo-inverse by hand. This way requires multiple attempts and is cumbersome. In addition, one can approximate the degradation matrix $\boldsymbol{A}$ by training a network on constructed pair data.
- Our initialization optimization via DEQ inversion needs many iterations for convergence. How to reduce the iterations and accelerate remains a future work.
- Our method is a new zero-shot image restoration method. Moreover, the IR performance depends on the pre-trained denoiser. The performance is not better than some supervised learning methods on a specific task. With the help of our model inversion, one can train our DeqIR model to improve the performance of IR.