Supplementary Material

A. Mathematical Preliminaries

Definition 1 (Contraction on $\mathbb{R}^n$)**.** *A function $\boldsymbol{f}:\mathbb{R}^n\mapsto\mathbb{R}^n$ is called a contraction mapping if there exists a real number $0\leq\lambda<1$ such that for all $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$,*

$$\|\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{f}(\boldsymbol{y})\|\leq\lambda\|\boldsymbol{x}-\boldsymbol{y}\| \tag{20}$$

Using the intermediate value theorem for the function $\boldsymbol{f}(\boldsymbol{x})$, we can easily see that $\boldsymbol{f}(\boldsymbol{x})$ is contracting with the rate $0\leq\lambda<1$ if $\boldsymbol{f}$ satisfies the following:

$$\sigma_{\max}\left(\frac{\partial\boldsymbol{f}(\boldsymbol{x})}{\partial\boldsymbol{x}}\right)\leq\lambda<1 \tag{21}$$

where $\sigma_{\max}(\boldsymbol{A})$ denotes the largest singular value of a matrix $\boldsymbol{A}$. Note that the contraction mapping in $\mathbb{R}^n$ is closely related to Lipschitz continuity, and indeed the function that satisfies (20) with any $\lambda>0$ is called λ-Lipschitz continuous function.

Now, we provide a theorem for discrete stochastic contraction, which is slightly modified from the contraction theorem of stochastic difference equation in [22].

Theorem A.1. *[22] Consider the stochastic difference equation:*

$$\boldsymbol{x}_{i+1}=\boldsymbol{f}(\boldsymbol{x}_i,i)+g(\boldsymbol{x}_i,i)\boldsymbol{w}_i \tag{22}$$

where $\boldsymbol{f}(\cdot,i)$ is a $\mathbb{R}^n\mapsto\mathbb{R}^n$ function, $g(\cdot,i)$ is a $\mathbb{R}^n\times\mathbb{N}\mapsto\mathbb{R}$ function for each $i\in\mathbb{N}$, and $\{\boldsymbol{w}_i,i=1,2,\cdots\}$ is a sequence of independent n-dimensional zero mean unit variance Gaussian noise vectors. Assume that the system satisfies the following two hypothesis:

(H1) The function $\boldsymbol{f}(\cdot,i)$ is contracting with factor λ in the sense of (21) for all $i\in\mathbb{N}$.

(H2) $\mathrm{Tr}(g(\boldsymbol{x},i)\boldsymbol{I}g(\boldsymbol{x},i))\leq C,\quad\forall\boldsymbol{x},i.$

Then, for two sample trajectory $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ that satisfies (22), we have

$$\mathbb{E}\|\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i\|^2\leq\frac{2C}{1-\lambda^2}+\lambda^{2i}\mathbb{E}\|\boldsymbol{x}_0-\tilde{\boldsymbol{x}}_0\|^2 \tag{23}$$

The following corollary is a simple consequence of Theorem A.1.

Corollary 1. *Consider the stochastic difference equation associated with the data fidelity term:*

$$\boldsymbol{x}'_{i+1}=\boldsymbol{f}(\boldsymbol{x}_i,i)+\sigma(\boldsymbol{x}_i,i)\boldsymbol{z}_i \tag{24}$$
$$\boldsymbol{x}_{i+1}=\boldsymbol{A}\boldsymbol{x}'_{i+1}+\boldsymbol{b} \tag{25}$$

where the $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ is a non-expansive linear mapping, and $\boldsymbol{f}(\boldsymbol{x},i)$ and $\sigma(\boldsymbol{x},i)$ satisfies (H1) and (H2). Then, for two sample trajectories $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ that satisfies (22), we have

$$\mathbb{E}\|\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i\|^2\leq\frac{2C\tau}{1-\lambda^2}+(\lambda)^{2i}\mathbb{E}\|\boldsymbol{x}_0-\tilde{\boldsymbol{x}}_0\|^2 \tag{26}$$

where $\tau=\frac{\mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})}{n}$.

Proof. After the application of (25), we have

$$\boldsymbol{x}_{i+1}=\underbrace{\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}_i,i)+\boldsymbol{b}}_{\tilde{\boldsymbol{f}}(\boldsymbol{x}_i,i)}+\sigma(\boldsymbol{x}_i,i)\boldsymbol{A}\boldsymbol{z}_i$$

Therefore, we have

$$\sigma_{\max}\left(\frac{\partial\tilde{\boldsymbol{f}}(\boldsymbol{x},i)}{\partial\boldsymbol{x}}\right)\leq\sigma_{\max}(\boldsymbol{A})\sigma_{\max}\left(\frac{\partial\boldsymbol{f}(\boldsymbol{x},i)}{\partial\boldsymbol{x}}\right)$$
$$=\lambda$$

as $\sigma_{\max}(\boldsymbol{A})\leq1$ for a non-expansive linear mapping. Furthermore, we have

$$\mathrm{Tr}(g(\boldsymbol{x},i)\boldsymbol{A}^T\boldsymbol{A}g(\boldsymbol{x},i))=g(\boldsymbol{x},i)^2\mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})$$
$$=\frac{\mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})}{n}C=C\tau$$

Therefore, we have

$$\mathbb{E}\|\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i\|^2\leq\frac{2C\tau}{1-\lambda^2}+\lambda^{2i}\mathbb{E}\|\boldsymbol{x}_0-\tilde{\boldsymbol{x}}_0\|^2 \tag{27}$$

□

Lemma A.1. *Let $s_\theta(\boldsymbol{x}_i,i)$ be a sufficiently expressive parameterized score function so that*

$$s_\theta(\boldsymbol{x}_i,t)=\frac{\partial}{\partial\boldsymbol{x}_i}\log p_{0i}(\boldsymbol{x}_i|\boldsymbol{x}_0) \tag{28}$$

Then, we have

$$\frac{\partial}{\partial\boldsymbol{x}_i}\boldsymbol{s}_\theta(\boldsymbol{x}_i,t)=-\frac{1}{b_i^2}\boldsymbol{I}. \tag{29}$$

where

$$b_i^2=\begin{cases}1-\bar{\alpha}_i, & \textit{(DDPM)}\\ \sigma_i^2-\sigma_0^2, & \textit{(SMLD)}\end{cases} \tag{30}$$

Proof. The forward diffusion is given by

$$\boldsymbol{x}_i=a_i\boldsymbol{x}_0+b_i\boldsymbol{z} \tag{31}$$

where $\boldsymbol{z}\sim\mathcal{N}(0,\boldsymbol{I})$ and (a_i,b_i) are defined in (5) and (10) for DDPM and SMLD, respectively. Using (28), we have

$$\frac{\partial}{\partial \boldsymbol{x}_i}\left(\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_i, i)\right)^T \tag{32}$$

$$= \frac{\partial}{\partial \boldsymbol{x}_i}\left(\frac{\partial}{\partial \boldsymbol{x}_i}\log p_{0i}(\boldsymbol{x}_i|\boldsymbol{x}_0)\right)^T \tag{33}$$

$$= \frac{\partial}{\partial \boldsymbol{x}_i}\left(\frac{\partial}{\partial \boldsymbol{x}_i}\Big(-\frac{\|\boldsymbol{x}_i - a_i\boldsymbol{x}_0\|^2}{2b_i^2}\Big)\right)^T \tag{34}$$

$$= \frac{\partial}{\partial \boldsymbol{x}_i}\Big(-\frac{\boldsymbol{x}_i - a_i\boldsymbol{x}_i}{b_i^2}\Big)^T \tag{35}$$

$$= -\frac{1}{b_i^2}\boldsymbol{I}, \tag{36}$$

where ^{T} denotes the transpose. This concludes the proof. □

B. Proof of Theorem 1

Let N be the standard reverse diffusion step when starting from $T = 1$. Then, the number of discretization step for our method is given $N' = Nt_0 < N$ so that t_0 can refer to the acceleration factor. We further define a new index $i = N' - j$ to convert the reverse diffusion index $j = N', \cdots, 1$ to a forward direction index $i = 0, 1, \cdots, N'$. This does not change the contraction property of the stochastic difference equation. Therefore, without loss of generality, we use the aforementioned contraction property of stochastic difference equation for the index $i = 0, 1, \cdots, N'$. Now, we are ready to provide the proof.

B.1. DDPM

In DDPM, the discrete version of the forward diffusion is given by Eq. (5), and the reverse diffusion is given by eq. (6). Here, $\boldsymbol{z}_\theta(\boldsymbol{x}, i)$ is trained by

$$\min_\theta \mathbb{E}_i \mathbb{E}_{\boldsymbol{x}(0)} \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\Big[\|\boldsymbol{z} - \boldsymbol{z}_\theta(\sqrt{\bar\alpha_i}\boldsymbol{x}(0) + \sqrt{1-\bar\alpha_i}\boldsymbol{z}, i)\|^2\Big]. \tag{37}$$

It was shown that $\boldsymbol{z}_\theta(\boldsymbol{x}, i)$ is a scaled version of the score function [34]:

$$\boldsymbol{s}_\theta(\boldsymbol{x}, i) = -\frac{1}{\sqrt{1-\bar\alpha_i}}\boldsymbol{z}_\theta(\boldsymbol{x}, i) \tag{38}$$

which leads to

$$\boldsymbol{x}_{i-1} = \underbrace{\frac{1}{\sqrt{\alpha_i}}\Big(\boldsymbol{x}_i + (1-\alpha_i)\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)\Big)}_{\boldsymbol{f}(\boldsymbol{x}_i, i)} + \sigma_i \boldsymbol{z}, \tag{39}$$

Thus, we have

$$\begin{aligned}\frac{\partial \boldsymbol{f}^T(\boldsymbol{x}_i, i)}{\partial \boldsymbol{x}_i} &= \frac{1}{\sqrt{\alpha_i}}\left(\boldsymbol{I} + (1-\alpha_i)\frac{\partial \boldsymbol{s}_\theta^T(\boldsymbol{x}_i, i)}{\partial \boldsymbol{x}_i}\right)\\ &= \frac{1}{\sqrt{\alpha_i}}\left(1 - \frac{1-\alpha_i}{1-\bar\alpha_i}\right)\boldsymbol{I}\\ &= \frac{1}{\sqrt{\alpha_i}}\frac{\alpha_i - \bar\alpha_i}{1-\bar\alpha_i}\boldsymbol{I}\\ &= \sqrt{\alpha_i}\frac{1-\bar\alpha_{i-1}}{1-\bar\alpha_i}\boldsymbol{I}\end{aligned}$$

Therefore, the contraction rate is given by

$$\lambda = \max_{i\in[N']}\sqrt{\alpha_i}\left(\frac{1-\bar\alpha_{i-1}}{1-\bar\alpha_i}\right) < 1 \tag{40}$$

as $0 < \alpha_i, \bar\alpha_i < 1$. Furthermore, we can easily show that

$$C = n\max_{i\in[N']}(1-\bar\alpha_i) = n(1-\bar\alpha_N),$$

as $\bar\alpha_i$ is decreasing with i.

B.2. SMLD: Discrete Version of VE-SDE

In discrete version of VE-SDE, the forward diffusion is given by (10). The associated reverse diffusion is given by (11). Thus, we have

$$\begin{aligned}\frac{\partial \boldsymbol{f}^T(\boldsymbol{x}_i, i)}{\partial \boldsymbol{x}_i} &= \boldsymbol{I} + (\sigma_i^2 - \sigma_{i-1}^2)\frac{\partial \boldsymbol{s}_\theta^T(\boldsymbol{x}_i, i)}{\partial \boldsymbol{x}_i}\\ &= \left(1 - \frac{\sigma_i^2 - \sigma_{i-1}^2}{\sigma_i^2 - \sigma_0^2}\right)\boldsymbol{I}\\ &= \frac{\sigma_{i-1}^2 - \sigma_0^2}{\sigma_i^2 - \sigma_0^2}\boldsymbol{I}\end{aligned}$$

and the contraction rate is given by

$$\lambda = \max_{i\in[N']}\frac{\sigma_{i-1}^2 - \sigma_0^2}{\sigma_i^2 - \sigma_0^2} < 1 \tag{41}$$

as σ_i is increasing with i. Furthermore, we can easily show that

$$C = n\max_{i\in[N']}\sigma_i^2 - \sigma_{i-1}^2$$

B.3. DDIM

The DDIM forward diffusion can be set identically to the forward diffusion of DDPM (5), whereas the reverse diffusion is given as (7). In fact, with a proper reparameterization, one can cast DDIM such that it is equivalent to the discrete version of VE-SDE without noise terms. More specifically, if we define the following reparametrization:

$$\bar{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i}{\sqrt{\bar\alpha_i}} \tag{42}$$

then (7) becomes

$$\bar{\boldsymbol{x}}_{i-1} = \bar{\boldsymbol{x}}_i + (\sigma_{i-1} - \sigma_i)\, \boldsymbol{z}_\theta(\boldsymbol{x}_i, i) \tag{43}$$

where

$$\sigma_i = \frac{\sqrt{1-\bar{\alpha}_i}}{\sqrt{\bar{\alpha}_i}} \tag{44}$$

Furthermore, the corresponding score function with respect to the reparameterization is

$$\boldsymbol{s}_\theta(\bar{\boldsymbol{x}}_i, i) = -\frac{\boldsymbol{z}_\theta(\boldsymbol{x}_i, i)}{\sigma_i} \tag{45}$$

so that we have

$$\bar{\boldsymbol{x}}_{i-1} = \bar{\boldsymbol{x}}_i - (\sigma_{i-1} - \sigma_i)\, \sigma_i \boldsymbol{s}_\theta(\bar{\boldsymbol{x}}_i, i) \tag{46}$$

The forward diffusion (5) can be equivalently represented by the reparameterization as:

$$\bar{\boldsymbol{x}}_i = \bar{\boldsymbol{x}}_0 + \sigma_i \boldsymbol{z} \tag{47}$$

as $\alpha_0 = 1$. Therefore, we have

$$\frac{\partial \boldsymbol{f}^T}{\partial \bar{\boldsymbol{x}}_i}(\bar{\boldsymbol{x}}_i) = \left(1 + \frac{\sigma_{i-1} - \sigma_i}{\sigma_i}\right) \boldsymbol{I} = \frac{\sigma_{i-1}}{\sigma_i} \boldsymbol{I} \tag{48}$$

and the contraction rate is given by

$$\lambda = \max_{i \in [N']} \frac{\sigma_{i-1}}{\sigma_i} < 1 \tag{49}$$

as σ_i is increasing with i. Furthermore, we can easily show that $C = 0$ as there is no noise term.

C. Proof of Theorem 2

For some of the proofs, we borrow more tight inequality to obtain the result. In fact, the inequality of stochastic contraction

$$\bar{\varepsilon}_{0,r} \leq \frac{2C\tau}{1-\lambda^2} + \lambda^{2N'} \bar{\varepsilon}_{N'} \tag{50}$$

is a rough estimation of recursive inequality [22]

$$\bar{\varepsilon}_{j-1,r} \leq \lambda_j^2 \bar{\varepsilon}_{j,r} + 2C_j\tau, \tag{51}$$

where $\bar{\varepsilon}_{j,r}$ denotes the estimation error between reverse conditional diffusion path down to j. Accordingly, we have

$$\bar{\varepsilon}_{0,r} \leq \bar{\varepsilon}_{N,r} \prod_{j=0}^{N} \lambda_j^2 + \sum_{j=1}^{N} \left(2C_j\tau \prod_{i=1}^{j-1} \lambda_i^2\right), \tag{52}$$

which is reduced to (50) when λ_j and C_j are uniformly bounded by λ and C, respectively.

Now, our proof strategy is as follows. We specify reasonable conditions on $\{\beta_i\}$ or $\{\sigma_i^2\}$, which are satisfied by the existing DDPM, SLMD, and DDIM scheduling approaches. Then, for any $0 < \mu \leq 1$, our goal is to to show that there exists N' such that

$$\bar{\varepsilon}_{0,r} \leq \mu\varepsilon_0,$$

and N' decreases as ε_0 gets smaller.

C.1. DDPM

Without loss of generality, we assume that ground truth image and the corrupted image are normalized within range $[0,1]$, i.e. $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in [0,1]^n$. Then, we have

$$\varepsilon_0 = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2 \leq n. \tag{53}$$

We choose N' such that

$$N'\beta_{N'} \geq 2\log\left(\frac{4n}{\mu\varepsilon_0}\right) \tag{54}$$

$$N'\beta_{N'} \leq \frac{\mu\varepsilon_0}{4n\tau}. \tag{55}$$

We separately investigate each term in (52). First, from theorem 1,

$$\begin{aligned}
\bar{\varepsilon}_{N,r} &= a_{N'}^2 \varepsilon + 2b_{N'}^2 n \\
&= \bar{\alpha}_{N'}\varepsilon_0 + (1-\bar{\alpha}_{N'})2n \\
&= 2n + \bar{\alpha}_{N'}(\varepsilon_0 - 2n) \\
&\leq 2n
\end{aligned}$$

where the last inequality comes from (53). Subsequently,

$$\begin{aligned}
&\sum_{j=1}^{N'} \left(2C_j\tau \prod_{i=1}^{j-1} \lambda_i^2\right) \\
&= \sum_{j=1}^{N'} \left(2n(1-\alpha_j)\tau \prod_{i=1}^{j-1} \lambda_i^2\right) \\
&\leq 2n\tau \sum_{j=1}^{N'} \beta_j \cdot 1 \\
&\leq 2n\tau N'\beta_{N'} \leq \frac{\mu\varepsilon_0}{2}.
\end{aligned}$$

where the first inequality comes from $\prod_{i=1}^{j-1} \lambda_i^2 \leq \prod_{i=1}^{j-1} 1 \leq 1$ and the last equality is from (55). Therefore,

$$\begin{aligned}
\bar{\varepsilon}_{0,r} &\leq \bar{\varepsilon}_{N',r} \prod_{j=0}^{N'} \lambda_j^2 + \sum_{j=1}^{N'} \left(2C_j\tau \prod_{i=1}^{j-1} \lambda_i^2\right) \\
&\leq 2n \cdot e^{-\frac{N'\beta_{N'}}{2}} + \frac{\mu\varepsilon_0}{2} \\
&\leq 2n \cdot \frac{\mu\varepsilon_0}{4n} + \frac{\mu\varepsilon_0}{2} \leq \mu\varepsilon_0,
\end{aligned} \tag{56}$$

where the third inequality holds by (54), and the inequality in (56) comes from Lemma C.1 (see below). Furthermore, from (55), we can see that N' becomes smaller for a smaller ε_0. This concludes the proof of DDPM.

Lemma C.1.

$$\prod_{j=1}^{N'} \lambda_j^2 \leq e^{-\frac{N'\beta_{N'}}{2}}.$$

Proof of Lemma C.1.

$$\begin{aligned}
\prod_{j=1}^{N'} \lambda_j^2 &= \prod_{j=1}^{N'} \alpha_j \cdot \frac{(1-\bar{\alpha}_{j-1})^2}{(1-\bar{\alpha}_j)^2} \\
&\leq \prod_{j=1}^{N'} \alpha_j \\
&\leq \left(\frac{1}{N'}\sum_{j=1}^{N'} \alpha_j\right)^{N'} \\
&= \left(1 - \frac{1}{N'}\sum_{j=1}^{N'} \beta_j\right)^{N'} \\
&= \left(1 - \frac{\beta_{N'}}{2}\right)^{N'}
\end{aligned}$$

where the first inequality comes from $\bar{\alpha}_j = \bar{\alpha}_{j-1}\alpha_j \leq \bar{\alpha}_{j-1}$, and the second inequality is the inequality of arithmetic and geometric means, and the third equality is from the linear increasing β_j from $\beta_0 = 0$. Finally, using

$$e^x \geq \left(1 + \frac{x}{N}\right)^N \quad \text{for } N \geq 1, |x| \leq N \tag{57}$$

we have

$$\prod_{j=1}^{N'} \lambda_j^2 \leq e^{-\frac{N\beta_{N'}}{2}},$$

This concludes the proof. □

C.2. SMLD

Assume that the minimum and maximum values of variance satisfy the following:

$$\sigma_{\min}^2 < \frac{\mu^{\frac{3}{2}}\varepsilon_0}{8n} \tag{58}$$

$$\sigma_{\max}^2 > \frac{\mu\varepsilon_0}{4n}. \tag{59}$$

Then, using (58),

$$\log\left(\frac{2}{\sqrt{\mu}}\right) < \log\left(\frac{\mu\varepsilon_0}{4n\sigma_{\min}^2}\right),$$

and thus

$$\frac{\log(2/\sqrt{\mu})}{\log(\sigma_{\max}^2/\sigma_{\min}^2)} < \frac{\log(\mu\varepsilon_0/4n\sigma_{\min}^2)}{\log(\sigma_{\max}^2/\sigma_{\min}^2)}. \tag{60}$$

In addition, from (59), we have

$$\frac{\mu\varepsilon_0}{4n\sigma_{\min}^2} < \frac{\sigma_{\max}^2}{\sigma_{\min}^2},$$

and hence

$$\frac{\log(\mu\varepsilon_0/4n\sigma_{\min}^2)}{\log(\sigma_{\max}^2/\sigma_{\min}^2)} < 1. \tag{61}$$

Combining (60) with (61), we arrive at

$$\frac{\log(2/\sqrt{\mu})}{\log(\sigma_{\max}^2/\sigma_{\min}^2)} < \frac{\log(\mu\varepsilon_0/4n\sigma_{\min}^2)}{\log(\sigma_{\max}^2/\sigma_{\min}^2)} < 1. \tag{62}$$

Now, we can choose N' such that it satisfies the following conditions:

$$\begin{aligned}
\frac{N'-1}{N-1} &\geq \frac{\log(2/\sqrt{\mu})}{\log(\sigma_{\max}^2/\sigma_{\min}^2)} \\
\frac{N'-1}{N-1} &\leq \frac{\log(\mu\varepsilon_0/4n\sigma_{\min}^2)}{\log(\sigma_{\max}^2/\sigma_{\min}^2)}
\end{aligned} \tag{63}$$

This leads to the following bounds

$$\begin{aligned}
\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{\frac{N'-1}{N-1}} &\geq \frac{2}{\sqrt{\mu}} \\
n\sigma_{\min}^2\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{\frac{N'-1}{N-1}} &\leq \frac{\mu\varepsilon_0}{4}.
\end{aligned} \tag{64}$$

On the other hand, in the geometric scheduling of noise, for all i, we have

$$\lambda = \sigma_{\min}^2\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{\frac{i-1}{N-1}} \Big/ \sigma_{\min}^2\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{\frac{i-2}{N-1}} \tag{65}$$

$$= \left(\frac{\sigma_{\min}^2}{\sigma_{\max}^2}\right)^{\frac{1}{N-1}} \text{ and}$$

$$C = n\max\sigma_i^2\left(1 - \frac{\sigma_{i-1}^2}{\sigma_i^2}\right) = n\sigma_{N'}^2(1-\lambda). \tag{66}$$

where

$$\sigma'_N = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{N'-1}{N-1}}.$$

Note that from (64),

$$2n\sigma_{N'}^2 = 2n\sigma_{\min}^2\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{\frac{N'-1}{N-1}} \leq \frac{\mu\varepsilon_0}{2}, \tag{67}$$

and

$$\left(\frac{\sigma_{\min}^2}{\sigma_{\max}^2}\right)^{\frac{2(N'-1)}{N-1}} \leq \frac{\mu}{4}. \tag{68}$$

Hence, by plugging in (66) to (50), we have

$$\begin{aligned}
\bar{\varepsilon}_{0,r} &\leq \frac{2C\tau}{1-\lambda^2} + \lambda^{2N'}\bar{\varepsilon}_{N'} \\
&= \frac{2n\sigma_N^2(1-\lambda)\tau}{(1+\lambda)(1-\lambda)} + \left(\frac{\sigma_{\min}^2}{\sigma_{\max}^2}\right)^{\frac{2N'}{N-1}}(\varepsilon_0 + 2n\sigma_{N'}^2) \\
&\leq 2n\sigma_{N'}^2\frac{\tau}{1+\lambda} + \left(\frac{\sigma_{\min}^2}{\sigma_{\max}^2}\right)^{\frac{2(N'-1)}{N-1}}(\varepsilon_0 + 2n\sigma_{N'}^2) \\
&\leq \frac{\mu\varepsilon_0}{2} + \frac{\mu}{4}\left(\varepsilon_0 + \frac{\mu\varepsilon_0}{2}\right) \\
&\leq \frac{\mu\varepsilon_0}{2} + \frac{\mu}{4}(\varepsilon_0 + \varepsilon_0) \\
&= \mu\varepsilon_0,
\end{aligned}$$

where the third inequality comes from the bounds in (67), (68), and the fact that $\tau = \frac{tr(A^TA)}{n} < 1$ for a non-expansive linear mapping A.

Finally, we can easily see that the value N' satisfying (63) decreases as ε_0 decreases.

C.3. DDIM

In DDIM, we have $C_j = 0$ for Eq. (52). Let σ_0 and N' satisfy the following:

$$\sigma_0^2 \leq \frac{\mu\varepsilon_0}{4n} \tag{69}$$

$$\sigma_{N'}^2 \geq \frac{\varepsilon_0}{2n} \tag{70}$$

Then, we have

$$\begin{aligned}
\bar{\varepsilon}_0, r &\leq \bar{\varepsilon}_{N,r}\prod_{j=1}^{N}\lambda_j^2 \\
&\leq (\varepsilon_0 + \sigma_{N'}^2 2n)\cdot\frac{\sigma_0^2}{\sigma_{N'}^2} \\
&\leq \mu\varepsilon_0
\end{aligned}$$

where the second equality comes from $\lambda_j = \sigma_{j-1}/\sigma_j$ and the last equality comes from Eqs. (69) and (70).

We can also easily see that the minimum value N' satisfying (70) decreases as ε_0 decreases, as σ_i^2 is an increasing sequence in DDIM.

D. Implementation detail

In this section, we provide detailed explanation of discrete version of CCDF for each application. Again, the number of discretization step for our method is given $N' = Nt_0 < N$ where t_0 refers to the acceleration factor.

D.1. Super-resolution and Image Inpainting

For these problems, we employ the discretized version of the VP-SDE, which has shown impressive results on conditional generation [5, 8]. Namely, we use DDPM [10], with several strategies introduced in improved DDPM (IDDPM) [20] for both training the score function and for reverse diffusion procedure.

The modified reverse diffusion is given by

$$\boldsymbol{x}'_{i-1} = \frac{1}{\sqrt{\alpha_i}}\Big(\boldsymbol{x}_i + (1-\alpha_i)\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)\Big) + \sqrt{\sigma_i}\boldsymbol{z}, \tag{71}$$

where σ_i is given by

$$\sigma_i = \exp(v\log\beta_i + (1-v)\log\tilde{\beta}_i), \tag{72}$$

letting model variance to be learnable in a range $[\beta_i, \tilde{\beta}_i]$, where $\tilde{\beta}_i$ is given by $\tilde{\beta}_i = \frac{1-\bar{\alpha}_{i-1}}{1-\bar{\alpha}_i}\beta_i$. In (72), v is the learnable parameter so that it can be trained using the variational lower-bound penalty introduced in [10, 20].

Specifically, v and the score function $\boldsymbol{s}_\theta$ are trained using the following objective

$$L_{total}(\theta, v) = L_{simple}(\theta) + \lambda L_{VLB}(v), \tag{73}$$

where $L_{simple}(\theta)$ is given in (37) and we apply stop-gradient for the L_{VLB} so that the gradient of the loss contributes only to estimating the model variance.

For the training of score function, we use a U-Net architecture as used in [20] with the loss function as given in (73). Multi-headed attention [36] was used only at the 16×16 resolution. Linear beta noise scheduling [10] with $\beta_{\text{min}} = 0.0001$ and $\beta_{\text{max}} = 0.02$ were used, with $N = 1000$ discretization. We train the model with a batch size of 2, and a static learning rate of `1e-4` with Adam [14] optimizer for 5M steps. Exponential moving average (EMA) rate of 0.9999 was applied to the model.

For super-resolution, we define a blur kernel $\boldsymbol{h}_D$ which is defined by successive applications of the downsampling filter by a factor D, and upsampling filter by a factor D. This can be represented as a matrix multiplication:

$$\boldsymbol{P}\boldsymbol{x}' := \boldsymbol{h}_D * \boldsymbol{x}'. \tag{74}$$

where $\boldsymbol{x}'$ denotes intermediate estimate from the reverse diffusion. Then, we use the following data consistency iteration:

$$\boldsymbol{x}_i = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}'_i + \hat{\boldsymbol{x}}_i, \tag{75}$$

where $\boldsymbol{x}_i$ is the current estimate, and $\hat{\boldsymbol{x}}_i$ is the forward propagated image from the initial measurement $\hat{\boldsymbol{x}}(0)$:

$$\hat{\boldsymbol{x}}_i = \sqrt{\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z} \tag{76}$$

Therefore, we have

$$\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{P}, \quad \boldsymbol{b} = \hat{\boldsymbol{x}}_i.$$

We can easily see that $\sigma_{\max}(\boldsymbol{A}) \leq 1$ for the normalized filter $\boldsymbol{h}_D$.

Similarly, for the case of image inpainting, $\boldsymbol{P}$ is just a diagonal matrix with 1 at the measured locations and 0 on the unmeasured locations so that $\sigma_{\max}(\boldsymbol{A}) \leq 1$.

The resulting pseudo-code implementation of the algorithm is given in Algorithm 1.

Algorithm 1 Accelerated Super-resolution / inpainting (VP, Markov)

Require: $\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0, N', \{\alpha_i\}_{i=1}^{N'}, \{\sigma_i\}_{i=1}^{N'}, \boldsymbol{s}_\theta$
$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\boldsymbol{x}_{N'} \leftarrow \sqrt{\bar{\alpha}_{N'}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{N'}}\boldsymbol{z}$ ▷ Forward diffusion
for $i = N'$ to 1 **do** ▷ Reverse diffusion
  $\boldsymbol{x}'_{i-1} \leftarrow \frac{1}{\sqrt{\alpha_i}}(\boldsymbol{x}_i + (1-\alpha_i)\boldsymbol{s}_\theta(\boldsymbol{x}_i, i))$
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} + \sigma_i \boldsymbol{z}$ ▷ Unconditional update
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{x}}_i \leftarrow \sqrt{\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z}$
  $\boldsymbol{x}_{i-1} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}_{i-1} + \hat{\boldsymbol{x}}_i$
  ▷ Measurement consistency
end for
return $\boldsymbol{x}_0$

D.2. DDIM for Super-resolution/Inpainting

Note that we can use the same score function trained for DDPM, and use it in DDIM sampling [30]. Here, we study the effect on combining DDIM together with the proposed method to achieve even further acceleration. All we need to do is modify the unconditional update step, arriving at Algorithm 2.

D.3. MRI reconstruction

For the task of MRI reconstruction, we ground our work on Score-MRI [6], and modify the previous algorithm for our purpose. The algorithm is given in Algorithm 3. Specifically, we use variance exploding (VE-SDE) with predictor-corrector (PC) sampling which gives optimal results for MR reconstruction. For the step size of the corrector (Langevin dynamics) step, we use the following

$$\epsilon_i = 2r \frac{\|\boldsymbol{z}\|_2}{\|\boldsymbol{s}_\theta(\boldsymbol{x}_i, \sigma_i)\|_2}, \tag{77}$$

Algorithm 2 Accelerated Super-resolution / inpainting (VP, markov) + DDIM

Require: $\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0, N', \{\alpha_i\}_{i=1}^{N'}, \{\sigma_i\}_{i=1}^{N'}, \boldsymbol{s}_\theta$
$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\boldsymbol{x}_i \leftarrow \sqrt{\bar{\alpha}_i}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z}$ ▷ Forward diffusion
for $i = N_0$ to 1 **do** ▷ Reverse diffusion
  $\boldsymbol{x}_{i-1} \leftarrow \frac{1}{\sqrt{\alpha_i}}\boldsymbol{x}_i +$
  $\left(\frac{1-\bar{\alpha}_i}{\sqrt{\alpha_i}} - \sqrt{(1-\bar{\alpha}_i)(1-\bar{\alpha}_{i-1})}\right)\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$
  ▷ Unconditional update
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{x}}_i \leftarrow \sqrt{\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z}$
  $\boldsymbol{x}_{i-1} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}_{i-1} + \hat{\boldsymbol{x}}_i$
  ▷ Measurement consistency
end for
return $\boldsymbol{x}_0$

with $r = 0.16$ set as constant. For training the score function, we use the following minimization strategy:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U(\eta,1)} \mathbb{E}_{\boldsymbol{x}(0)\sim p_0} \mathbb{E}_{\boldsymbol{x}(t)\sim\mathcal{N}(\boldsymbol{x}(0),\sigma^2(t)\boldsymbol{I})} \Big[\left\| \sigma(t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), t) - \frac{\boldsymbol{x}(t) - \boldsymbol{x}(0)}{\sigma(t)} \right\|_2^2 \Big], \tag{78}$$

with $\eta = 1\mathrm{e}{-5}$, and

$$\sigma(t) = \sigma_{\min} \left(\frac{\sigma_{max}}{\sigma_{\min}}\right)^t, \tag{79}$$

with $\sigma_{\min} = 0.01, \sigma_{max} = 378$. We construct a modified U-Net model introduced in [34], namely `ncsnpp`. Adam optimizer is used for optimization, with a static learning rate of $2\mathrm{e}{-4}$ for 5M steps. EMA rate of 0.999 is used, and gradient clipping is applied with the maximum value of 1.0.

In compressed sensing MRI, the subsampled k-space data $\boldsymbol{y}$ is obtained from underlying image $\boldsymbol{x}$ as:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{F}\boldsymbol{x} \tag{80}$$

where $\boldsymbol{F}$ denote the Fourier transform and its inverse, $\boldsymbol{D}$ is a diagonal matrix indicating the k-space sampling location and $\boldsymbol{y}$ is the original zero-filled k-space data.

The associated data consistency imposing operator is then defined by

$$\boldsymbol{x}_i = (\boldsymbol{I} - \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F})\boldsymbol{x}'_i + \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{y} \tag{81}$$

where $\boldsymbol{F}^{-1}$ is the inverse Fourier transform. Therefore, we have

$$\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F}, \quad \boldsymbol{b} = \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{y}.$$

Again, we can easily see $\sigma_{\max}(\boldsymbol{A}) \leq 1$ as the Fourier transform is orthonormal.

Algorithm 3 Accelerated MR reconstruction (VE, PC)

Require: $\boldsymbol{x}_0, \boldsymbol{y}, N', \{\sigma_i\}_{i=1}^{N'}, \{\epsilon_i\}_{i=1}^{N'}, \boldsymbol{s}_\theta$

$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

$\boldsymbol{x}_{N'} \leftarrow \boldsymbol{x}_0 + \sigma_{N'}\boldsymbol{z}$ ▷ Forward diffusion

for $i = N'$ to 1 **do** ▷ Reverse diffusion

  $\boldsymbol{x}'_{i-1} \leftarrow \boldsymbol{x}_i + (\sigma_i^2 - \sigma_{i-1}^2)\boldsymbol{s}_\theta(\boldsymbol{x}_i, \sigma_i)$

  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

  $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}$ ▷ Predictor

  $\boldsymbol{x}_{i-1} = (\boldsymbol{I} - \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F})\boldsymbol{x}_i + \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{y}$ ▷ Measurement consistency

  $\boldsymbol{x}'_{i-1} \leftarrow \boldsymbol{x}_{i-1} + \epsilon_i \boldsymbol{s}_\theta(\boldsymbol{x}_i, \sigma_i)$

  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

  $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} + \sqrt{2\epsilon_i}\boldsymbol{z}$ ▷ Corrector

  $\boldsymbol{x}_{i-1} = (\boldsymbol{I} - \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F})\boldsymbol{x}_i + \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{y}$ ▷ Measurement consistency

end for

return $\boldsymbol{x}_0$

The corresponding pseudo-code implementation is shown in Algorithm 3.

All training and inference algorithms were implemented in PyTorch, and were performed on a single RTX 3090 GPU.

E. Additional Experiments

E.1. Super-resolution

Comparison study. In Fig. E.1, we compare the results of super-resolution using fairly large number of diffusion steps, as opposed to using only 20 number of diffusion steps as shown in Fig. 4. This is a region where ILVR is known to perform well, as opposed to the few-step setting. While in Fig. E.1, ILVR uses 1000 steps of diffusion, the proposed method only uses 100, 200, and 300 steps of diffusion for $\times 4$, $\times 8$, and $\times 16$, respectively. Nevertheless, the quality of reconstruction does not degrade, thanks to the contraction property of CCDF.

Incorporation of DDIM. We provide additional SR results using CCDF + DDIM. In Figure E.2, we show an experiment with the FFHQ dataset, where we compare the combination of ILVR + DDIM, and proposed method + DDIM. For ILVR + DDIM, in order to reduce the number of iterations, we choose larger discretization steps used in DDIM. For the proposed method, we fix $N = 50$, and reduce the value of t_0 to achieve less iterations. In the figure, we confirm that our method can be used together with DDIM to create high-fidelity samples with as small as 5 reverse diffusion iterations, even when it comes down to extreme cases of SR $\times 8$ or SR $\times 16$. Additionally, we observe that the results with $t_0 \leq 0.5$ is *superior* to the $t_0 = 1.0$ counterparts, again confirming our theory.

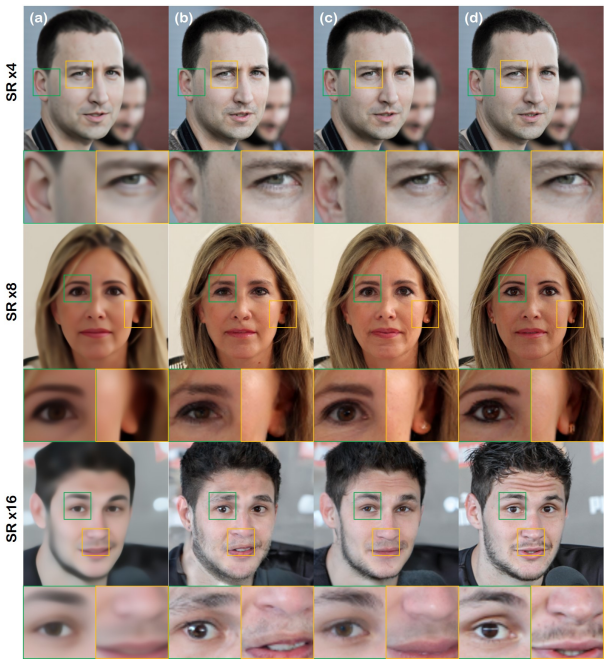

Figure E.1. Comparison on SR task ($\times 4$, $\times 8$, $\times 16$ for the 1^{st}, 2^{nd}, and 3^{rd} row): (a) ESRGAN [37], (b) ILVR [5] (1000 steps), (c) proposed method (100, 200, 300 steps for $\times 4$, $\times 8$, $\times 16$ SR), (d) Ground Truth

	SR	method	5	10	25	50
FFHQ	$\times 4$	ILVR +DDIM	120.53	114.61	87.15	81.85
		proposed +DDIM	72.34	**69.39**	78.83	82.72
	$\times 8$	ILVR +DDIM	147.44	115.30	101.37	93.72
		proposed +DDIM	91.84	**85.43**	87.43	94.89
	$\times 16$	ILVR +DDIM	147.44	115.30	101.37	93.72
		proposed +DDIM	91.84	**85.43**	87.43	94.89
AFHQ	$\times 4$	ILVR +DDIM	63.79	55.57	40.22	30.57
		proposed +DDIM	17.57	**17.19**	20.87	30.22
	$\times 8$	ILVR +DDIM	106.94	67.06	51.75	45.96
		proposed +DDIM	35.03	**31.70**	35.62	45.17
	$\times 16$	ILVR +DDIM	163.98	94.68	69.60	65.33
		proposed +DDIM	70.02	59.01	**49.36**	64.61

Table E.1. FID(↓) scores on FFHQ and AFHQ test set for SR task with DDIM by varying the number of iterations.

The same trend can also be seen via quantitative metrics in Table E.1. Using limited number of diffusion steps, FID score in the case of ILVR+DDIM grows exponentially,

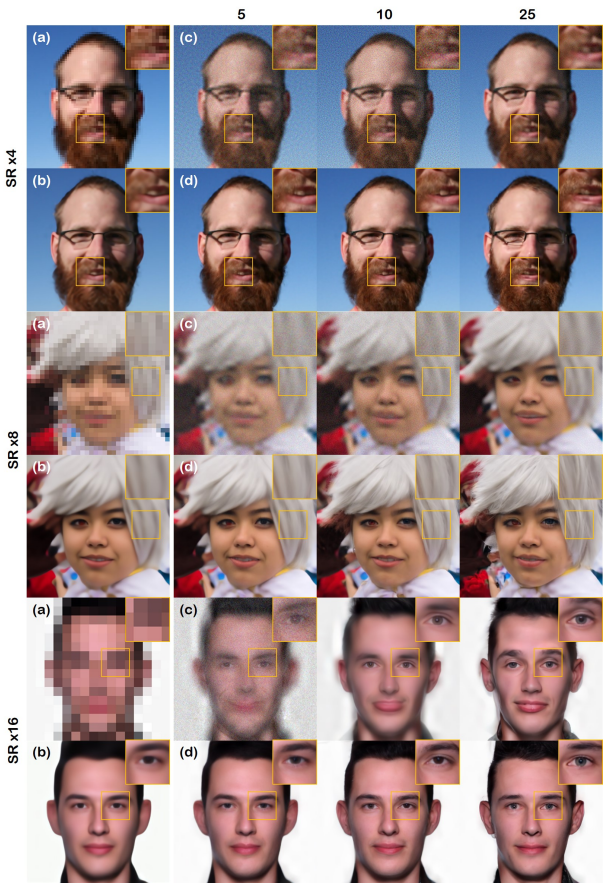

Figure E.2. Results on SR task of FFHQ dataset with proposed method + DDIM. Numbers on top indicate number of reverse diffusion iterations. (a) LR image, (b) ground truth, (c) ILVR + DDIM, (d) CCDF + DDIM.

as we decrease the number of steps taken. Contrarily, our method is able to *improve* the metric by quite a margin, as opposed to using full diffusion with 50 steps in total. This trend is indeed similar to the experiments performed with DDPM.

Experiments with ImageNet. ImageNet [7] contains diverse categories of natural images, and are known to be much harder to model, due to its highly multimodal nature. We try to examine if CCDF scales even to this challenging task, using a pre-trained model provided in the guided-diffusion github repository[3]. As with other experiments with FFHQ or AFHQ dataset, we train an ESRGAN model for each SR factor, and use it as our initialization strategy. In Fig. F.2, we can see that our CCDF strategy outperforms ILVR using full reverse diffusion, and also vastly improves the image quality of ESRGAN, which is our initialization.

t_0	0.05	0.1	0.2	0.5	0.75	1.0 [34]
Box 96	46.03	**45.93**	45.99	46.14	48.05	48.61
Box 128	50.41	50.05	**49.77**	51.65	54.49	59.27
Box 160	61.77	59.62	**57.99**	61.04	67.50	78.50

Table E.2. FID(↓) scores on FFHQ test set for inpainting task with varying t_0 values. $t_0 = 1.0$ is the baseline method without any acceleration used in [34]. Numbers in boldface, and underline indicate the best, and the second best scores.

E.2. Inpainting

Dependence on t_0. As in Table 2, we compare the FID score of reconstructions for the inpainting task, as we vary the t_0 values in table E.2. We notice similar results from the SR task, in the sense that there always exist $t_0 \in (0, T)$ which gives higher scores than using full diffusion. With relatively small boxes, we see that $t_0 = 0.1$ is optimal, whereas we typically need more diffusion steps for larger boxes.

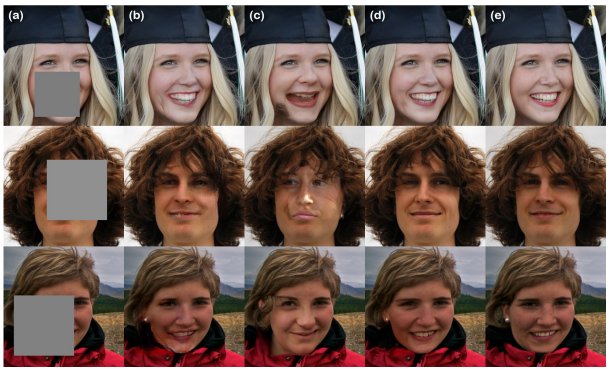

Figure E.3. Comparison on inpainting task: (a) Input image, (b) SN-PatchGAN [40], (c) score-SDE [34] using 1000 steps from $T = 1$, (d) CCDF using 200 steps from $t_0 = 0.2$, (e) Ground Truth.

Comparison study. We compare the proposed CCDF strategy with SN-PatchGAN [40], and score-SDE [34] using 1000 steps in Fig. E.3. For SN-PatchGAN in Fig. E.3 (b), we often see highly unrealistic details e.g. near the mouth. Note that SN-patchGAN serves as the initialization point for CCDF in inpainting. Leveraging this imperfect initialization, the proposed method is able to provide reconstructions that are highly realistic, as can be seen in fig. E.3 (d). It is also notable that score-SDE using full diffusion more often than not produces results that are incoherent with the known regions (see second row of Fig. E.3 (c)), while the proposed method stably outputs coherent results.

Ablation study. We perform an ablation study comparing the effect of different initialization strategy. Table E.3

[3] https://github.com/openai/guided-diffusion

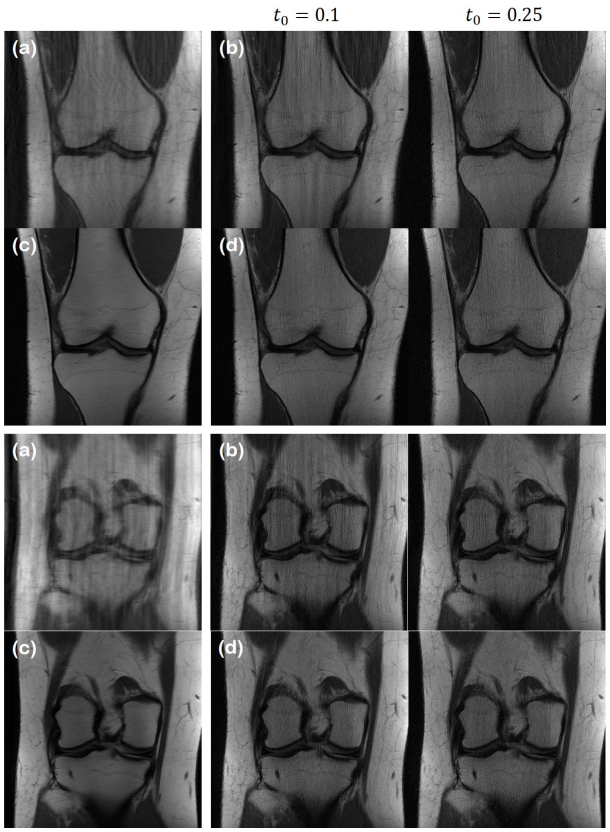

Figure E.4. Ablation study on using different initializations for forward diffusion ($\times 6$ 1D Gaussian sampling). (a) Vanilla initialization, (b) corresponding results with the proposed method. (c) NN initialization, (d) corresponding results with the proposed method.

	SR	$t_0 =$	0.1	0.2	0.5	1.0
FFHQ	$\times 4$	vanilla	78.39	66.77	64.25	63.14
		NN init.	**60.90**	**60.91**	64.04	63.31
	$\times 8$	vanilla	116.42	93.06	82.39	78.91
		NN init.	78.13	**75.76**	79.34	77.34
	$\times 16$	vanilla	184.70	135.20	96.15	92.32
		NN init.	101.79	92.59	**88.09**	88.49
AFHQ	$\times 4$	vanilla	19.14	18.66	18.08	18.70
		NN init.	**15.53**	17.14	19.06	18.10
	$\times 8$	vanilla	48.87	39.88	33.28	34.84
		NN init.	33.47	**32.30**	33.65	33.50
	$\times 16$	vanilla	96.01	72.22	47.42	47.28
		NN init.	63.27	51.13	**44.18**	45.17

Table E.3. FID(↓) scores for SR tasks with different initialization strategies.

shows the difference in the results when using vanilla initialization with the corrupted image, and NN initialization. We see that with all t_0, NN initialization performs marginally better than vanilla initialization. The difference becomes clearer as we decrease the value of t_0 to 0.1. The same ablation study was performed also for MRI reconstruction task, and is illustrated in Figure E.4. We see similar trend as in the SR task.

Furthermore, we provide additional qualitative results of each task on various datasets, focusing mainly on showing the trend of reconstruction results as we vary the value of t_0. In Figure F.3, we compare the achievable image quality by fixing the number of reverse diffusion steps to 20. Consistent with what we saw in Figure 4, we see that our method largely outperforms the other diffusion model-based methods. In Figure F.4, Figure F.5 respectively, we see that we can stably arrive at a feasible solution with different values of $t_0 \in [0.1, 0.5]$, typically requiring higher values of t_0 for severer degradation.

F. Validity of assumption

In Lemma A.1, we assumed that $s_\theta(\boldsymbol{x}_i, t) = \frac{\partial}{\partial \boldsymbol{x}_i} \log p_{0i}(\boldsymbol{x}_i|\boldsymbol{x}_0)$. In this section, we briefly show that the assumption is valid. In the theoretic side, [1] showed that given an *optimal* reconstruction function $r^*_\sigma(\boldsymbol{x}_t)$ trained with denoising autoencoder loss asymptotically behaves as

$$r^*_\sigma(\boldsymbol{x}_t) = \boldsymbol{x}_t + \sigma^2 \frac{\partial \log p(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} + o(\sigma^2), \quad \sigma \to 0$$
$$s^*_\theta(\boldsymbol{x}_t) = \frac{r^*_\sigma(\boldsymbol{x}_t) - \boldsymbol{x}_t}{\sigma^2} = \frac{\partial \log p(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} + o(1), \quad \sigma \to 0,$$

where the equation emphasizes the behavior of the optimal *score* function, regarding how it was parameterized. This asymptotic behavior hints that the error will be small, especially when σ approaches zero. In our case, $\sigma \to 0$ as $t \to 0$, and hence the behavior holds near $t = 0$.

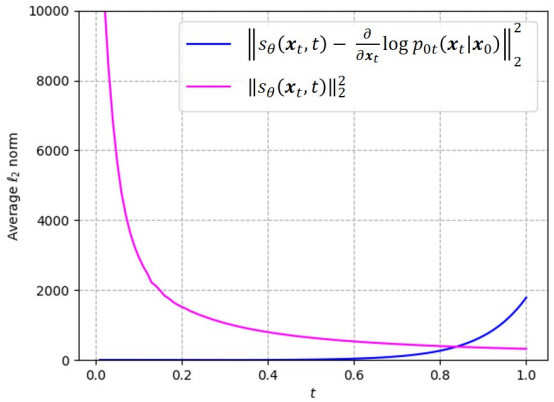

Figure F.1. Average value of $\|s_\theta(\boldsymbol{x}_t, t) - \frac{\partial}{\partial \boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2$, and the norm of $\|s_\theta(\boldsymbol{x}_t, t)\|_2^2$ at time t, on fastMRI 320×320 test set (1K samples).

In order to numerically validate such error, we conducted an experiment to calculate the actual average error norm, il-

lustrated in Fig. F.1. Here, we see that the error norm mostly stays at very low values across the range. We do observe that the magnitude of error inevitably increases when the noise level is too large, so the error grows where $t > 0.5$. Nevertheless, we note that most of our contraction analysis stays in the $t \in [\epsilon, 0.5]$ regime, and hence the assumption made in Lemma A.1. is practical.

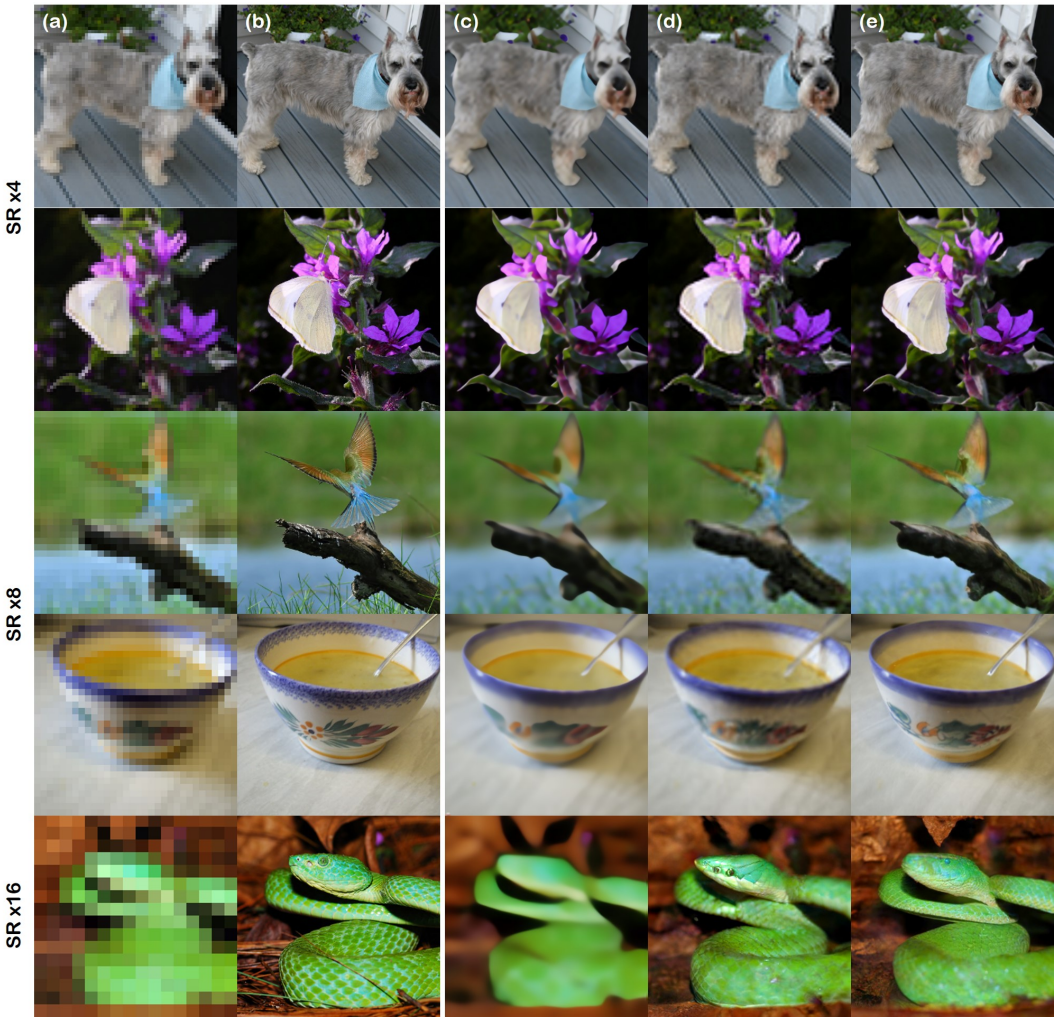

Figure F.2. Results of SR task on ImageNet256 validation dataset. (a) LR image, (b) ground truth, (c) ESRGAN, (d) ILVR (1000 steps), (e) CCDF (100, 200, 300 steps for $\times 4$, $\times 8$, $\times 16$ SR, respectively.)

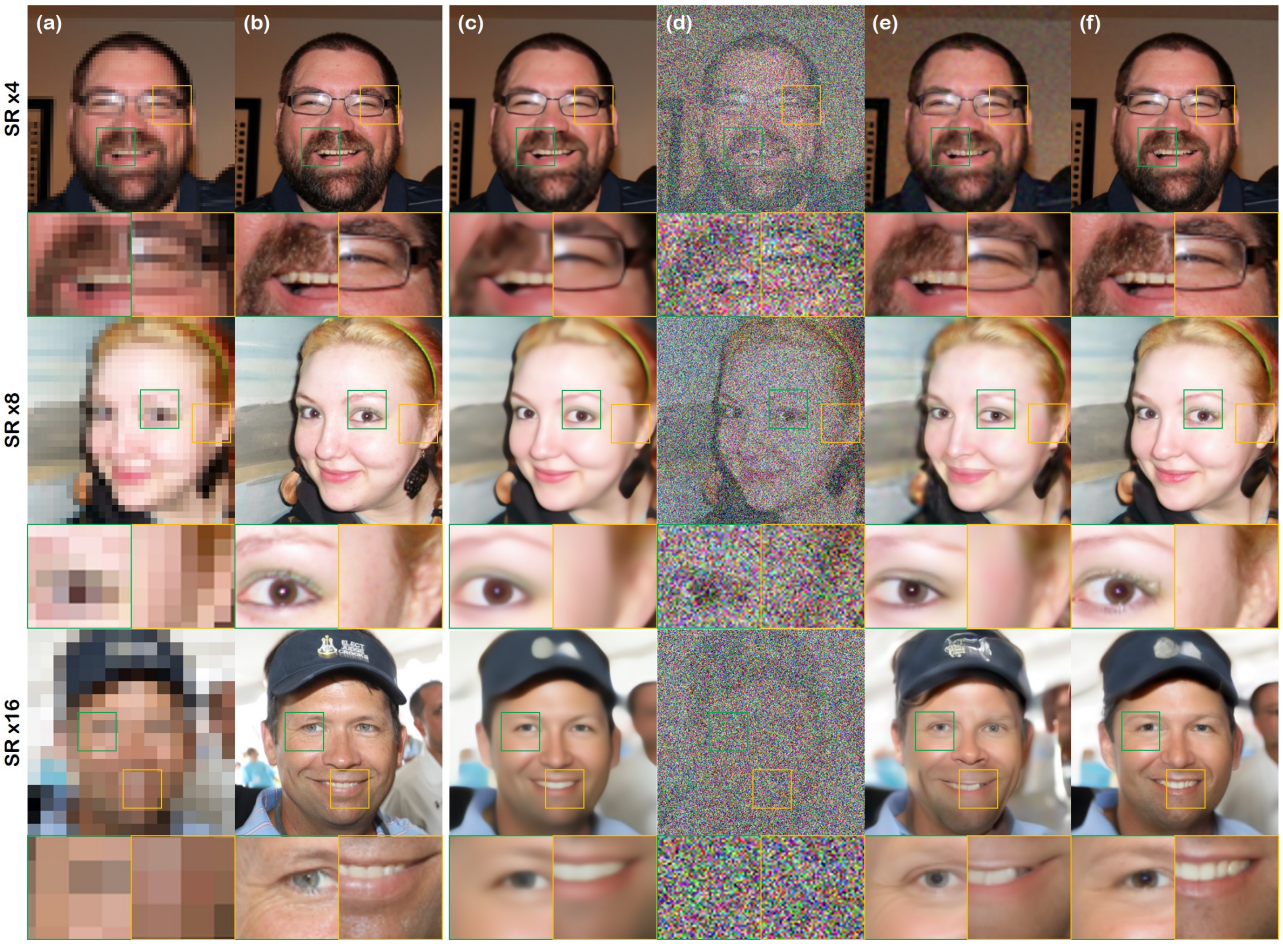

Figure F.3. Results of super-resolution on FFHQ 256×256 data. The first, second and third row denote $\times 4$ SR, $\times 8$ SR, and $\times 16$ SR, respectively. (a) LR input, (b) Ground Truth, (c) ESRGAN [37], (d) SR3 [25] with 20 diffusion steps ($N = 20, \Delta t = 0.05$), (e) ILVR [5] with 20 diffusion steps($N = 20, \Delta t = 0.05$), (f) proposed method (CCDF) with 20 diffusion steps ($N = 100, t_0 = 0.2$).

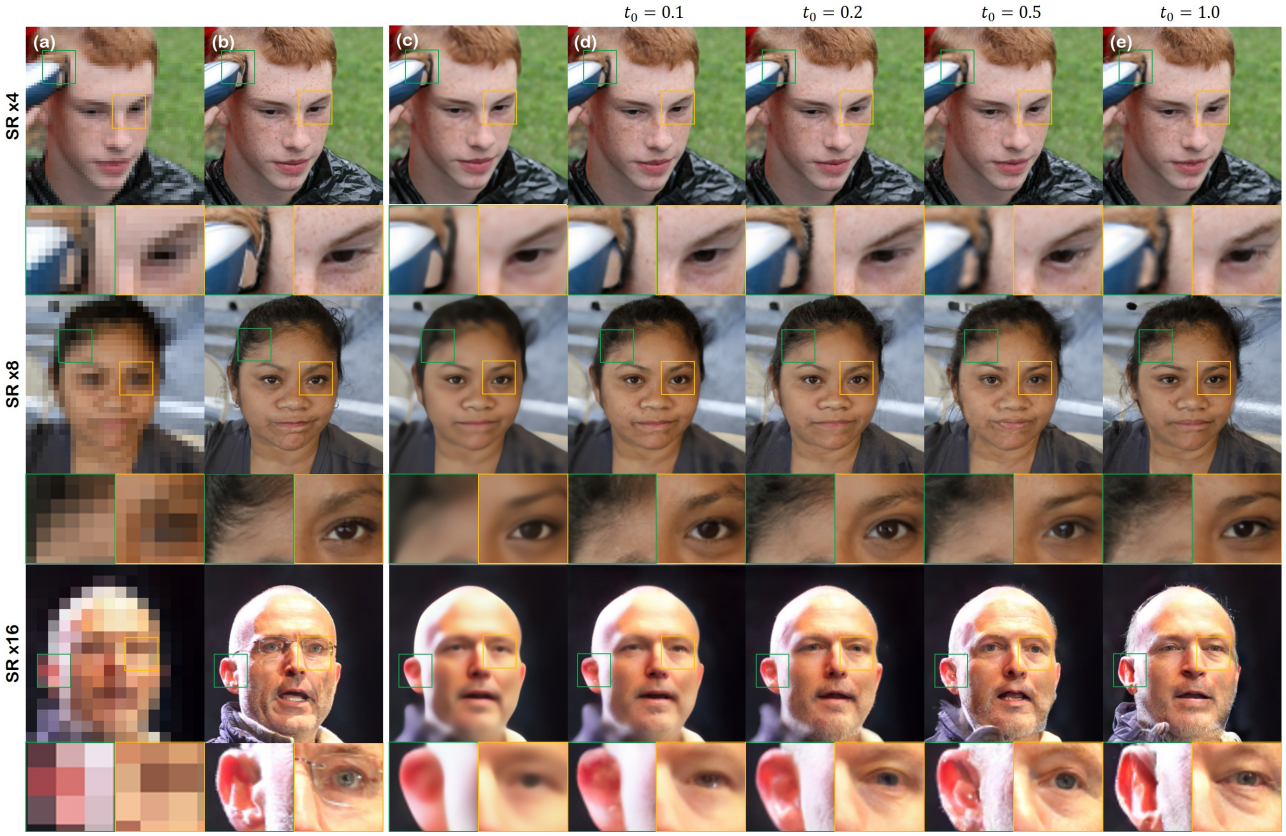

Figure F.4. Results of super-resolution on FFHQ 256×256 data. The first, second and third row denote $\times 4$ SR, $\times 8$ SR, and $\times 16$ SR, respectively. (a) LR input, (b) Reference, (c) ESRGAN [37], (d) proposed method (CCDF) with varying t_0 values, and (e) ILVR ($t_0 = 1.0$) [5].

Figure F.5. Additional results of inpainting on FFHQ 256×256 data. The first, second and third row denote masks of size 96×96, 128×128 and 160×160, respectively. (a) Masked image, (b) SN-patchGAN [40], (c) proposed method (CCDF) with varying t_0 values, and (d) Score-SDE ($t_0 = 1.0$) [34]