Low-Dose CT Using Denoising Diffusion Probabilistic Model for 20× Speedup

Wenjun Xia, Qing Lyu, Ge Wang
Department of Biomedical Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
{xia4, lyuq, wangge}@rpi.edu

Abstract

Low-dose computed tomography (LDCT) is an important topic in the field of radiology over the past decades. LDCT reduces ionizing radiation-induced patient health risks but it also results in a low signal-to-noise ratio (SNR) and a potential compromise in the diagnostic performance. In this paper, to improve the LDCT denoising performance, we introduce the conditional denoising diffusion probabilistic model (DDPM) and show encouraging results with a high computational efficiency. Specifically, given the high sampling cost of the original DDPM model, we adapt the fast ordinary differential equation (ODE) solver for a much-improved sampling efficiency. The experiments show that the accelerated DDPM can achieve 20x speedup without compromising image quality.

1 Introduction

Computed tomography (CT) plays an indispensable role in radiology due to its high-contrast resolution, fast imaging speed, and many utilities. With the increasing popularity of CT scans, there is a major safety concern on ionizing radiation from x-ray exposure to the patient. To reduce the x-ray radiation-induced risks during a CT scan, low-dose CT (LDCT) research has attracted a widespread attention so that the image quality of LDCT can be optimized despite the radiation dose reduction.

To solve the above challenge of LDCT, there have been numerous efforts made in the past few decades. A natural approach for LDCT is to post-process a noisy LDCT image. Inspired by established methods for natural image denoising, non-local means (NLM) and block-matching 3D (BM3D) were adapted for LDCT denoising [27, 15, 22], significantly improving the LDCT performance. Although these post-processing methods do remove LDCT image noise, the resultant image quality often hardly meets the clinical requirements, since it is difficult to model the noise contained in the LDCT images, and thus new artifacts can be frequently introduced. Another commonly used LDCT denoising approach is referred to as model-based iterative reconstruction (MBIR), which is to establish a data model in the projection domain and a content model in the image domain, and then minimize an objective function for model fitting iteratively. Especially, inspired by compressed sensing (CS) theory, the regularization model with total variation (TV) greatly improves the reconstruction quality [49]. Subsequently, many methods were proposed to solve the over-smoothness problem associated with TV minimization, including anisotropic TV [9], total generalized variation (TGV) [32], fractional order TV [51, 50], and so on. In addition to the TV regularization, other effective priors were also formulated, such as NLM [7], tight framework [16], low-rank [3], dictionary learning [46, 8], learned sparsifying transform [52], etc. These regularization-based methods can achieve a promising denoising performance by careful designing the regularization term and fine-tuning the hyperparameters. However, that means extensive experience and skills, and limited clinical applications.
With the recent development of the deep learning (DL) techniques [26], LDCT denoising was immediately identified as an earliest application target [42]. Chen et al. first published a convolutional neural network (CNN) for LDCT denoising and achieved remarkable performance [6]. Progress in this area has been rapid, including residual structures to improve both the training convergence and denoising performance [5, 21, 23, 19], and incorporate various loss functions and training strategies, such as adversarial loss [43, 48], perceptual loss [48], edge incoherence loss [38], etc. Also, in some studies DL and MBIR algorithms were combined for synergies [44, 4, 1, 18, 11, 45]. These methods deploy CNNs in iterative optimization schemes so that the priors and penalty parameters can be learned in a data-driven fashion.

Very recently, the denoising diffusion probabilistic model (DDPM) [39, 20, 12, 47] emerged as a generative model with the mode diversity and output quality superior to that of the generative adversarial network (GAN) [17, 1, 39, 20, 40, 41, 13, 31]. DDPM gradually perturbs an original data distribution until into a standard distribution, typically a normal distribution. Then, DDPM iteratively recovers the data distribution in a learned reverse diffusion process. Song et al. interpreted the forward and backward diffusion processes in the stochastic differential equation (SDE) framework [41]. The SDE-based diffusion model allows the learning of the reverse diffusion process in continuous time, and has been applied to a variety of tasks, such as image super-resolution [34, 37], image inpainting [29], image editing [30], image translation [10, 36], and so on.

In this paper, we introduce a conditional DDPM for LDCT denoising. The U-Net [20] is adopted to learn the reverse diffusion process of a normal-dose CT (NDCT) image conditioned on a LDCT counterpart. Then, by gradually sampling, cleaner images can be step by step recovered from a normal noise distribution under the condition of the LDCT image. The final image generated by DDPM can remove structural noise in the input LDCT image while preserving clinically important details. However, the sampling efficiency of the original DDPM model is low and can hardly meet clinical requirements. To alleviate this problem, here we use a fast ordinary differential equation (ODE) solver [28] for a much-boosted efficient sampling, 20× faster than that of the original DDPM.

2 Methodology

2.1 Supervised Deep LDCT Denoising

Currently, a typical DL-based LDCT denoising method trains a network to learn a mapping from LDCT images to NDCT images. Let us assume that \( x \in \mathbb{R}^N \) and \( y \in \mathbb{R}^N \) are paired LDCT and NDCT images, then the parameters of the network can be trained as follows:

\[
\min_\theta \| D_\theta(x) - y \|_2^2,
\]

where \( D_\theta : \mathbb{R}^N \to \mathbb{R}^N \) is the network defined by \( \theta \), which is a vector of parameters.

2.2 Conditional DDPM for LDCT Denoising

As shown in Fig. 1, the conditional DDPM for LDCT denoising consists of a forward process and a reverse process. In reference to [20], the forward process is defined as a Markov chain which
The neural network is trained with a variational lower bound (VLB) of the negative log-likelihood:

\[ q(y_{1:T}|y_0) = \prod_{t=1}^{T} q(y_t|y_{t-1}), \]

where

\[ q(y_t|y_{t-1}) = \mathcal{N}(y_t|\sqrt{1-\beta_t}y_{t-1}, \beta_tI). \]

According to the properties of the Gaussian distribution, the sampling operation can be directly performed at an arbitrary timestep \( t \) without any iteration:

\[ q(y_t|y_0) = \mathcal{N}(y_t|\sqrt{\alpha_t}y_0, (1-\alpha_t)I), \]

where \( \alpha_t = 1 - \beta_t \) and \( \alpha_t = \prod_{i=1}^{t} \alpha_i \). After the forward process, \( y_T \) will follow a standard normal distribution \( \mathcal{N}(0, I) \) when \( T \) is large enough. Then, it can be shown that the posterior distribution of \( y_{t-1} \) given \( y_t \) and \( y_0 \) can be formulated as follows:

\[
\begin{align*}
q(y_{t-1}|y_t, y_0) &= \mathcal{N}(y_{t-1}|\tilde{\mu}_t(y_t, y_0), \sigma_t^2 I), \\
\tilde{\mu}_t(y_t, y_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\alpha_t}y_t + \frac{\sqrt{\alpha_{t-1}}(1-\alpha_t)}{1-\alpha_t}y_0, \\
\sigma_t^2 &= \frac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\alpha_t}.
\end{align*}
\]

Similar to the forward process, the reverse process conditioned on a LDCT image \( x \) is also a Markov chain defined by

\[ p_\theta(y_{0:T}|x) = p(y_T) \prod_{t=1}^{T} p_\theta(y_{t-1}|y_t, x), \]

where

\[ p_\theta(y_{t-1}|y_t, x) = \mathcal{N}(y_{t-1}|\mu_\theta(y_t, x, t), \sigma_t^2 I), \]

where \( y_T \) is sampled from a standard normal distribution \( p(y_T) \sim \mathcal{N}(0, I) \), \( \mu_\theta \) is the expectation computed by a learned U-Net model \( \theta \).

The neural network is trained with a variational lower bound (VLB) of the negative log-likelihood:

\[ \mathcal{L}_{VLB} = \mathbb{E}_q \left[ -\log \frac{p_\theta(y_{0:T}|x)}{q(y_{1:T}|y_0)} \right] \geq \mathbb{E} \left[ -\log p_\theta(y_0|x) \right]. \]

After inserting the Gaussian density function into Eq. (3) and removing the constant, the objective function can be rewritten as

\[ \mathcal{L} = \mathbb{E}_{x,y} \mathbb{E}_{\epsilon|y} \left[ \frac{1}{2\sigma_t^2} \| \tilde{\mu}_t(y_t, y_0) - \mu_\theta(y_t, x, t) \|^2 + \epsilon \right]. \]

Specifically, the forward process can be implemented as

\[ y_t = \sqrt{\alpha_t}y_0 + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim \mathcal{N}(0, I), \]

Then, the posterior expectation in Eq. (5) can be rewritten as

\[ \tilde{\mu}_t(y_t, y_0) = \tilde{\mu}_t \left( y_t, \frac{1}{\sqrt{\alpha_t}}(y_t - \sqrt{1-\alpha_t}\epsilon) \right) = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon). \]

Instead of directly learning the expectation, the U-Net model \( D_\theta \) is trained to predict the noise vector \( \epsilon \). Then, the objective function Eq. (9) becomes

\[ \mathcal{L} = \mathbb{E}_{x,y} \mathbb{E}_{\epsilon|y} \left[ \frac{(1-\alpha_t)^2}{2\sigma_t^2 \alpha_t(1-\alpha_t)} \| \epsilon - D_\theta(\sqrt{\alpha_t}y_0 + \sqrt{1-\alpha_t}\epsilon, x, t) \|^2 \right]. \]

According to [20], we remove the weight of the \( \ell_2 \) norm for better training performance. Finally, for inference the sampling operation \( y_{t-1} \sim p_\theta(y_{t-1}|y_t) \) can be computed as follows:

\[ y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( y_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}D_\theta(y_t, x, t) \right) + \sigma_t z, z \sim \mathcal{N}(0, I). \]

For clarify, Algorithms 1 and 2 are presented in the pseudo codes for training and inference, respectively.
To cast the discrete diffusion processes of DDPM in the continuous form, Kingma [24] proved that the forward process for $t \in \{1, 2, ..., T\}$ following the distribution Eq. (3) is equivalent to a stochastic differential equation (SDE) starting from $y_0 \sim p(y_0)$ for any $t \in [0, T]$:

$$dy = f(t)y dt + g(t)dw,$$

where $w_t \in \mathbb{R}^N$ is the standard Wiener process, and

$$f(t) = \frac{1}{\sqrt{\alpha_t}} \cdot \frac{\partial \log \eta_t}{\partial t}, \quad g^2(t) = \frac{\partial \sigma_t^2}{\partial t} - 2 \frac{\partial \log \eta_t}{\partial t} \sigma_t^2,$$

where $\eta_t = \sqrt{\alpha_t}$. Starting from $y_T \sim p(y_T)$, the reverse process can be formulated as a reverse-time SDE [41]:

$$dy = \left[ f(t)y - g^2(t)\nabla_y \log q_t(y) \right] dt + g(t)dw,$$

where $w_t \in \mathbb{R}^N$ is the standard Wiener process for the reverse time from $T$ to 0, $\log q_t(y)$ is called the score function and can be learned with a neural network [41]. The reverse process of DDPM can be regarded as a specific form of Eq. (16), in which the noise prediction model $D_\theta(y_t, x, t)$ is to learn $-\sigma_t \nabla_y \log q_t(y)$. The sampling of the above SDE with a large step will hardly converge because of the randomness of the Wiener process [33]. In [41], Song et al. proved that the SDE shown in Eq. (16) has the same marginal distribution at each time $t$ as that of a probability flow ODE, which can be sampled with a larger step:

$$dy = \left[ f(t)y - \frac{1}{2} g^2(t)\nabla_y \log q_t(y) \right] dt,$$

Substituting the noise prediction model into Eq. (17), the reverse process can be implemented as

$$dy = \left[ f(t)y + \frac{\sigma_t^2(t)}{2 \sigma_t} D_\theta(y_t, x, t) \right] dt,$$
Algorithm 3: DPM-Solver-1.

Input: Time sequence \( \{ t_i \in [0, 1] | i = 0, 1, ..., M \} \)

Output: \( y_{1:M} \)

Load \( D_\theta \)
\[ y_T \sim N(0, I) \]
\[ y_{00} = y_T \]

for \( i = 1, 2, ..., M \) do
\[ h_i = \lambda_{t_i} - \lambda_{t_{i-1}} \]
\[ s_i = t_{\lambda} \left( \lambda_{t_{i-1}} + \lambda_{t_i} \right) \]
\[ u_i = \frac{\eta_i}{\eta_{t_{i-1}}} y_{t_{i-1}} - \sigma_{s_i} \left( e^{h_i} - 1 \right) D_\theta(y_{t_{i-1}}, x, t_{i-1}) \]
\[ y_i = \frac{\eta_i}{\eta_{t_i}} y_{t_i} - \sigma_{s_i} \left( e^{h_i} - 1 \right) D_\theta(u_i, x, s_i) \]
end

Algorithm 4: DPM-Solver-2.

Input: Time sequence \( \{ t_i \in [0, 1] | i = 0, 1, ..., M \} \)

Output: \( y_{1:M} \)

Load \( D_\theta \)
\[ y_T \sim N(0, I) \]
\[ y_{00} = y_T \]

for \( i = 1, 2, ..., M \) do
\[ h_i = \lambda_{t_i} - \lambda_{t_{i-1}} \]
\[ s_i = t_{\lambda} \left( \lambda_{t_{i-1}} + \lambda_{t_i} \right) \]
\[ u_i = \frac{\eta_i}{\eta_{t_{i-1}}} y_{t_{i-1}} - \sigma_{s_i} \left( e^{h_i} - 1 \right) D_\theta(y_{t_{i-1}}, x, t_{i-1}) \]
\[ y_i = \frac{\eta_i}{\eta_{t_i}} y_{t_i} - \sigma_{s_i} \left( e^{h_i} - 1 \right) D_\theta(u_i, x, s_i) \]
end

which is a semi-linear ODE \([28]\), whose solution at time \( t \) can be obtained with the variation of constants formula \([2]\):
\[ y_t = \exp \left( \int_s^t f(\tau) d\tau \right) y_s + \int_s^t \left[ \exp \left( \int_\tau^t f(\tau') d\tau' \right) \frac{\sigma^2(\tau)}{2\sigma_t} D_\theta(y_\tau, x, \tau) \right] d\tau. \] (19)

Let \( \lambda_t = \log(\eta_t/\sigma_t) \), Eq. \([19]\) can be further simplified into
\[ y_t = \frac{\eta_t}{\eta_s} y_s + \eta_t \int_s^t \left[ \frac{d\lambda_\tau}{d\tau} \frac{\sigma_\tau}{\eta_\tau} D_\theta(y_\tau, x, \tau) \right] d\tau. \] (20)

The predefined \( \lambda_t \) can be obtained with a strictly decreasing function of \( t \), denoted \( \lambda(t) \), which has an inverse function \( t = t_\lambda(\lambda) \). Then, by changing the time variable \( t \) into the parameter variable \( \lambda \) and denoting \( y_\lambda := y_{t_\lambda(\lambda)} \) and \( \hat{D}_\theta(y_\lambda, x, \lambda) := D_\theta(y_{t_\lambda(\lambda)}, x, t_\lambda(\lambda)) \), Eq. \([20]\) can be rewritten as
\[ y_t = \frac{\eta_t}{\eta_s} y_s + \eta_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{D}_\theta(y_\lambda, x, \lambda) d\lambda, \] (21)

where the integral \( \int e^{-\lambda} \hat{D}_\theta d\lambda \) is called the exponentially weighted integral of \( \hat{D}_\theta \) \([28]\). This integral can be calculated numerically by Taylor expansion. According to the order of Taylor expansion, Lu et al. \([28]\) provided three solvers for the flow probability ODE. Algorithms \([3, 4]\) and \([5]\) are the ODE solvers obtained from the first-order, second-order and third-order Taylor expansion, which are called DPM-Solver-1, DPM-Solver-2 and DPM-Solver-3, respectively. As shown in the pseudo codes of the algorithms, there are \( k \) times of functional evaluation in DPM-Solver-k per step. The higher order solver has a faster convergence speed so that it takes fewer steps to achieve satisfactory results \([28]\). Therefore, with a limited number of functional evaluations (NFE), the DPM-Solver-3 is recommended. After applying DPM-Solver-3, if the remaining NFEs can no longer be divisible by 3, we can apply DPM-Solver-1 or DPM-Solver-2.

The DDPM model is trained with discrete time steps, while the DPM-Solver solves the SDE in a continuous fashion. Nevertheless, we can still apply the DPM-Solver to an already trained DDPM
Algorithm 5: DPM-Solver-3.

Input: Time sequence \( \{t_i \in [0, 1] | i = 0, 1, ..., M \} \)

Output: \( \tilde{y}_{tM} \)

Load \( D_\theta \)

\( y_T \sim \mathcal{N}(0, I) \)

\( y_{t0} = y_T, \ r_1 = \frac{1}{3}, \ r_2 = \frac{2}{3} \)

for \( i = 1, 2, ..., M \) do

\( h_i = \lambda_{t_i} - \lambda_{t_{i-1}} \)

\( s_{2i-1} = t_x(\lambda_{t_{i-1}} + \tau_1 h_i), \ s_{2i} = t_x(\lambda_{t_{i-1}} + \tau_2 h_i) \)

\( u_{2i-1} = \frac{n_{2i-1}}{n_{t_{i-1}}} \tilde{y}_{t_{i-1}} - \sigma_{s_{2i-1}} (e^{\tau_1 h_{i}} - 1) D_\theta(\tilde{y}_{t_{i-1}}, x, t_{i-1}) \)

\( v_{2i-1} = D_\theta(u_{2i-1}, x, s_{2i-1}) - D_\theta(\tilde{y}_{t_{i-1}}, x, t_{i-1}) \)

\( u_{2i} = \frac{n_{2i}}{n_{t_{i-1}}} \tilde{y}_{t_{i-1}} - \sigma_{s_{2i}} (e^{\tau_2 h_i} - 1) D_\theta(\tilde{y}_{t_{i-1}}, x, t_{i-1}) - \frac{\sigma_{s_{2i}} \tau_2}{r_1} \left( e^{\tau_2 h_{i}} - 1 \right) v_{2i-1} \)

\( v_{2i} = D_\theta(u_{2i}, x, s_{2i}) - D_\theta(\tilde{y}_{t_{i-1}}, x, t_{i-1}) \)

\( \tilde{y}_i = \frac{n_i}{n_{t_{i-1}}} \tilde{y}_{t_{i-1}} - \sigma_i (e^{h_{i}} - 1) D_\theta(\tilde{y}_{t_{i-1}}, x, t_{i-1}) - \frac{\sigma_i \tau_2}{r_2} \left( e^{h_{i}} - 1 \right) v_{2i} \)

end

Figure 2: The results obtained using different methods. (a) NDCT, (b) LDCT, (c) U-Net, (d) DDPM, (e) DPM-Solver 15 NFE, and (f) DPM-Solver 50 NFE. The display window is consistently set to [-160, 240] HU.

model. When applying the DPM-solver, we discretize the continuous time sequence into DDPM steps:

\[ t_d = 1000 \cdot \max \left( t_c - \frac{1}{T} \right), \]  \hspace{1cm} (22)

where \( t_d \) and \( t_c \) are discrete and continuous time variables respectively, and \( T \) is the total number of discrete time steps.

3 Experiments and Results

To evaluate the performance of our proposed method for LDCT denoising, the 2016 NIH-AAPM-Mayo Clinic Low-Dose CT Grand Challenge dataset was chosen. The dataset has 2,378 paired NDCT and LDCT images with a thickness of 3mm from 10 patients. We selected 1,923 paired images from 8 patients as the training set, and 455 paired images from the remaining 2 patients as the test set. The image matrix was resampled to be 256x256. The total number of time steps \( T \) was set to 1,000. The model was trained using the Adam optimizer \cite{25} with a learning rate of \( 1 \times 10^{-4} \). The training process converged well after \( 5 \times 10^5 \) iterations on a computing server equipped with
Figure 3: The absolute error maps of the results in reference to the NDCT image. (a) LDCT, (b) U-Net, (c) DDPM, (d) DPM-Solver 15 NFE, and (e) DPM-Solver 50 NFE. The display window is [0, 80] HU.

Table 1: Average quantitative results of different methods.

| Method        | PSNR  | SSIM  |
|---------------|-------|-------|
| LDCT          | 42.81 | 0.9667|
| U-Net         | 44.08 | 0.9823|
| DDPM          | 46.49 | 0.9873|
| DPM-Solver 15 NFE | 40.41 | 0.9423|
| DPM-Solver 50 NFE | 46.72 | 0.9880|

Nvidia GTX 1080 Ti x4 GPUs. After the model was trained, we used DDPM, DPM-Solver 15 NFE and DPM-Solver 50 NFE to sample the test set respectively.

Fig. 2 shows the results obtained using different methods. It can be seen that U-Net [35] failed to eliminate image noise completely. With DDPM, image noise was effectively removed while the structures were well kept. With only 15 NFE, the DPM-Solver could not denoise LDCT images well. Because of a too few number of sampling steps, residual noise and newly generated noise are evident in the resultant images sampled with DPM-Solver 15 NFE. Remarkably, after NFE was increased to 50, the DPM-solver delivered a competitive performance relative to that of DDPM. It can be seen that in the magnified region of interest (ROI), DPM-Solver 50 NFE well maintained the vasculature, which was blurred by U-Net. Quantitatively, DPM-Solver 50 NFE outperformed DDPM. Fig. 3 shows the absolute error maps of the abdominal results shown in Fig. 2 in reference to the NDCT image. It further demonstrates the efficient and effective sampling performance of the DPM-Solver, whose error map is the least noisy.

Table 1 shows the averaged quantitative evaluation of different methods over the whole test set. It can be seen that DDPM and DPM-Solver 50 NFE have the best quantitative scores. Especially, DPM-Solver 50 NFE quantitatively outperforms DDPM, which means that sampling with the DPM-Solver gives a superior noise suppression performance.

Table 2 shows the averaged computational costs of different methods. Without any need for iterative sampling, U-Net has the highest processing efficiency. However, the denoising performance of U-Net is of limited value in clinical applications. DDPM requires a very high inference cost, which is hardly acceptable in practice. The DPM-Solver allows an excellent acceleration, while keeping or even improving the sampling quality. It can be seen that DPM-Solver 50 NFE achieves a 20× speedup compared to DDPM while maintaining denoising performance.
Table 2: Averaged inference costs of different methods

| Method          | Cost (second per image) |
|-----------------|-------------------------|
| U-Net           | 0.08                    |
| DDPM            | 62.17                   |
| DPM-Solver 15 NFE | 0.95                   |
| DPM-Solver 50 NFE | 3.12                   |

4 Conclusion

In this paper, we have adapted a novel conditional DDPM model for LDCT denoising. The denoising performance outperforms the traditional CNN-based method. Furthermore, to accelerate the sampling operation of DDPM, we have applied a fast ODE solver. It is shown that the ODE Solver has both a high denoising performance and a high sampling efficiency. Specifically, the ODE solver can achieve a $20\times$ speedup without compromising the denoising performance. The superior performance of the ODE solver for DDPM sampling means a great potential for clinical applications.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
[2] Kendall Atkinson, Weimin Han, and David E Stewart. *Numerical solution of ordinary differential equations*. John Wiley & Sons, 2011.
[3] Jian-Feng Cai, Xun Jia, Hao Gao, Steve B Jiang, Zuowei Shen, and Hongkai Zhao. Cine cone beam ct reconstruction using low-rank matrix factorization: algorithm and a proof-of-principle study. *IEEE transactions on medical imaging*, 33(8):1581–1591, 2014.
[4] Hu Chen, Yi Zhang, Yunjin Chen, Junfeng Zhang, Weihua Zhang, Huaiqiang Sun, Yang Lv, Peixi Liao, Jiliu Zhou, and Ge Wang. Learn: Learned experts’ assessment-based reconstruction network for sparse-data ct. *IEEE transactions on medical imaging*, 36(12):2524–2535, 2017.
[5] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535, 2017.
[6] Hu Chen, Yi Zhang, Weihua Zhang, Peixi Liao, Ke Li, Jiliu Zhou, and Ge Wang. Low-dose ct via convolutional neural network. *Biomedical optics express*, 8(2):679–694, 2017.
[7] Yang Chen, Dazhi Gao, Cong Nie, Limin Luo, Wufan Chen, Xindao Yin, and Yazhong Lin. Bayesian statistical reconstruction for low-dose x-ray computed tomography using an adaptive-weighting nonlocal prior. *Computerized Medical Imaging and Graphics*, 33(7):495–500, 2009.
[8] Yang Chen, Xindao Yin, Luyao Shi, Huazhong Shu, Limin Luo, Jean-Louis Coatrieux, and Christine Toumoulin. Improving abdomen tumor low-dose CT images using a fast dictionary learning based processing. *Physics in Medicine & Biology*, 58(16):5803, 2013.
[9] Zhiqiang Chen, Xin Jin, Liang Li, and Ge Wang. A limited-angle ct reconstruction method based on anisotropic tv minimization. *Physics in Medicine & Biology*, 58(7):2119–2141, 2013.
[10] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.
[11] Il Yong Chun, Zhengyu Huang, Hongki Lim, and Jeff Fessler. Momentum-net: Fast and convergent iterative neural network for inverse problems. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
[12] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022.
[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.
[14] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[15] P Fumene Feruglio, Claudio Vinegoni, J Gros, A Sbarbati, and R Weissleder. Block matching 3d random noise filtering for absorption optical projection tomography. *Physics in Medicine & Biology*, 55(18):5401, 2010.

[16] Hao Gao, Hengyong Yu, Stanley Osher, and Ge Wang. Multi-energy ct based on a priori rank, intensity and sparsity model (prism). *Inverse problems*, 27(11):115012, 2011.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[18] Harshit Gupta, Kyong Hwan Jin, Ha Q Nguyen, Michael T McCann, and Michael Unser. Cnn-based projected gradient descent for consistent ct image reconstruction. *IEEE transactions on medical imaging*, 37(6):1440–1453, 2018.

[19] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[21] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[22] Dongwoo Kang, Piotr Slomka, Ryo Nakazato, Jonghye Woo, Daniel S Berman, C-C Jay Kuo, and Damini Dey. Image denoising of low-radiation dose coronary ct angiography by an adaptive block-matching 3d algorithm. In *Medical Imaging 2013: Image Processing*, volume 8669, pages 671–676. SPIE, 2013.

[23] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[24] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[25] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[26] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[27] Shanzhou Niu, Yang Gao, Zhaoying Bian, Jing Huang, Wufan Chen, Gaohang Yu, Zhengrong Liang, and Jianhua Ma. Sparse-view x-ray ct reconstruction via total generalized variation regularization. *Physics in Medicine & Biology*, 59(12):2997–3017, 2014.

[28] Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.
