PET Image Denoising with Score-Based Diffusion Probabilistic Models

Chenyu Shen[1,2], Ziyuan Yang[2], and Yi Zhang[1(✉)]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, China
yzhang@scu.edu.cn

[2] College of Computer Science, Sichuan University, Chengdu, China

Abstract. Low-count positron emission tomography (PET) imaging is an effective way to reduce the radiation risk of PET at the cost of a low signal-to-noise ratio. Our study aims to denoise low-count PET images in an unsupervised mode since the mainstream methods usually rely on paired data, which is not always feasible in clinical practice. We adopt the diffusion probabilistic model in consideration of its strong generation ability. Our model consists of two stages. In the training stage, we learn a score function network via evidence lower bound (ELBO) optimization. In the sampling stage, the trained score function and low-count image are employed to generate the corresponding high-count image under two handcrafted conditions. One is based on restoration in latent space, and the other is based on noise insertion in latent space. Thus, our model is named the bidirectional condition diffusion probabilistic model (BC-DPM). Real patient whole-body data are utilized to evaluate our model. The experiments show that our model achieves better performance in both qualitative and quantitative aspects compared to several traditional and recently proposed learning-based methods.

Keywords: PET denoising · diffusion probabilistic model · latent space conditions

1 Introduction

Positron emission tomography (PET) is an imaging modality in nuclear medicine that has been successfully applied in oncology, neurology, and cardiology. By injecting a radioactive tracer into the human body, the molecular-level activity in tissues can be observed. To mitigate the radiation risk to the human body, it is essential to reduce the dose or shorten the scan time, leading to a low signal-to-noise ratio and further negatively influencing the accuracy of diagnosis.

Recently, the denoising diffusion probabilistic model (DDPM) [6,9,11] has become a hot topic in the generative model community. The original DDPM was designed for generation tasks, and many recent works have proposed extending it for image restoration or image-to-image translation. In supervised mode, Saharia

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43907-0_26.

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14220, pp. 270–278, 2023.
https://doi.org/10.1007/978-3-031-43907-0_26

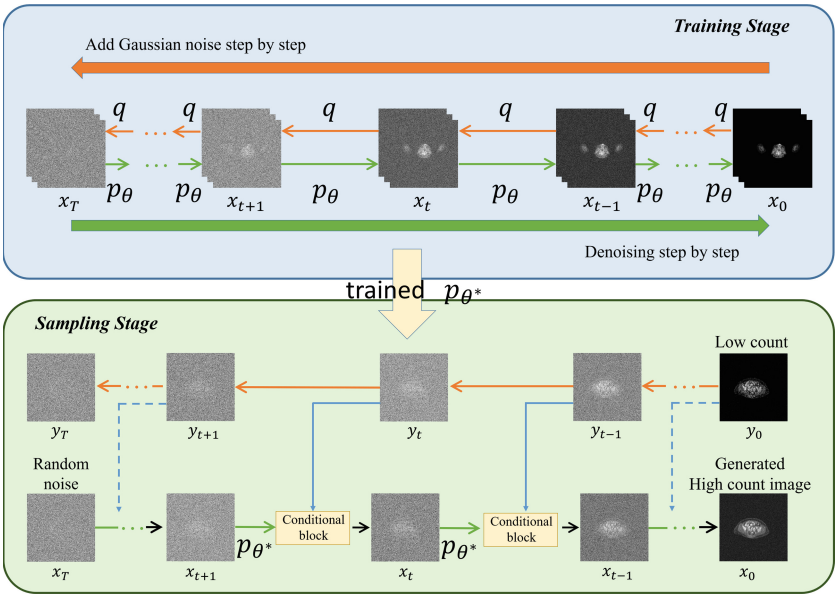

Fig. 1. Overview of our proposed BC-DPM model.

et al. [8] proposed a conditional DDPM to perform single-image super-resolution, which integrates a low-resolution image into each reverse step. In unsupervised mode, to handle the stochasticity of the generative process, Choi proposed iterative latent variable refinement (ILVR) [1] to guarantee the given condition in each transition, thus generating images with the desired semantics. DDPM has also been applied in medical imaging. To explore its generalization ability, Song *et al.* [12] proposed a fully unsupervised model for medical inverse problems, providing the measuring process and the prior distribution learned with a score-based generative model. For PET image denoising, Gong *et al.* [4] proposed two paradigms. One is directly feeding noisy PET images and anatomical priors (if available) into the score function network, which relies on paired high-quality and low-quality PET images. The other is feeding only MR images into the score function network while using noisy PET images in the inference stage under the assumption that PET image noise obeys a Gaussian distribution.

In this paper, we propose a conditional diffusion probabilistic model for low-count PET image denoising in an unsupervised manner without the Gaussian noise assumption or paired datasets. Our model is divided into two stages. In the training stage, we leverage the standard DDPM to train the score function network to learn a prior distribution of PET images. Once the network is trained, we transplant it into the sampling stage, in which we design two conditions to control the generation of high-count PET images given corresponding low-count PET images. One condition is that the denoised versions of low-count PET images are similar to high-count PET images. The other condition is that

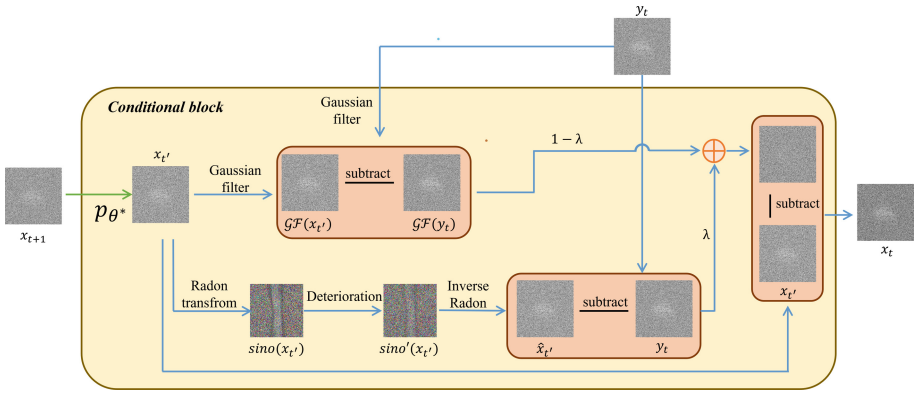

Fig. 2. The proposed conditional block in the sampling stage.

when we add noise to high-count PET images, they degrade to low-count PET images. As a result, our model is named the bidirectional condition diffusion probabilistic model (BC-DPM). In particular, to simulate the formation of PET noise, we add noise in the sinogram domain. Additionally, the two proposed conditions are implemented in latent space. Notably, Our model is 'one for all', that is, once we have trained the score network, we can utilize this model for PET images with different count levels.

2 Method

Letting $X \subset \mathcal{X}$ be a high-count PET image dataset and $Y \subset \mathcal{Y}$ be a low-count PET image dataset, x_0 and y_0 denote instances in X and Y, respectively. Our goal is to estimate a mapping $\mathcal{F}(\mathcal{Y}) = \mathcal{X}$, and the proposed BC-DPM provides an unsupervised technique to solve this problem. BC-DPM includes two stages. In the training stage, it requires only X without paired (X, Y), and in the sampling stage, it produces the denoised x_0 for a given y_0.

2.1 Training Stage

BC-DPM acts the same as the original DDPM in the training stage, it consists of a forward process and a reverse process. In the forward process, x_0 is gradually contaminated by fixed Gaussian noise, producing a sequence of latent space data $\{x_1, x_2, ..., x_T\}$, where $x_T \sim \mathcal{N}(0, \mathbf{I})$. The forward process can be described formally by a joint distribution $q(x_{1:T}|x_0)$ given x_0. Under the Markov property, it can be defined as:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\{\beta_1, \beta_2, ..., \beta_T\}$ is a fixed variance schedule with small positive constants and $\mathbf{I}$ represents the identity matrix. Notably, the forward process allows x_t to

be sampled directly from x_0:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{2}$$

where $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, $\alpha_t := 1 - \beta_t$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

The reverse process is defined by a Markov chain starting with $p(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)\mathbf{I}). \tag{3}$$

Given the reverse process, $p_\theta(x_0)$ can be expressed by setting up an integral over the $x_{1:T}$ variables $p_\theta(x_0) := \int p_\theta(x_{0:T}) dx_{1:T}$, and the parameter θ can be updated by optimizing the following simple loss function:

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\right\|^2]. \tag{4}$$

The $\epsilon_\theta(x_t, t)$ used in this paper heavily relies on that proposed by Dhariwal *et al.* [3]. The pseudocode for the training stage is given in Algorithm 1.

Algorithm 1: Training stage.

repeat
- $x_0 \sim q(x_0)$
- $t \sim Uniform(1, 2, ..., T)$
- $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- Update θ by optimizing
  $\mathbb{E}_{t,x_0,\epsilon}[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\right\|^2]$

until convergence

2.2 Sampling Stage

The main difference between BC-DPM and the original DDPM lies in the sampling stage. Due to the stochasticity of the reverse process $p_\theta(x_{0:T})$, it is difficult for the original DDPM to generate images according to our expectation. To overcome this obstacle, the proposed BC-DPM models $p_\theta(x_0|c)$ given condition c instead of modeling $p_\theta(x_0)$ as

$$p_\theta(x_0|c) = \int p_\theta(x_{0:T}|c) dx_{1:T}, \quad p_\theta(x_{0:T}|c) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, c). \tag{5}$$

Condition c derives from specific prior knowledge from the high-count PET image x_0 and the low-count PET image y_0. With c, BC-DPM can control the generation of x_0 given y_0.

Then, the core problem is to design a proper condition c. A natural choice is $\mathcal{D}(y_0) \approx x_0$, that is, the restoration task itself. We must clarify that it will not

cause a 'deadlock' for the following two reasons. One is that the final form of the condition $\mathcal{D}(y_0) \approx x_0$ does not involve x_0, and the other is that we choose a relatively simple denoiser in the condition, which can be viewed as a 'coarse to fine' operation. In practice, we utilize a Gaussian filter $\mathcal{GF}(\cdot)$ as the denoiser in this condition. However, the Gaussian filter usually leads to smoothed images. Based on this property, we observe that the PSNR value between $\mathcal{GF}(y_0)$ and x_0 is usually inferior to that between $\mathcal{GF}(y_0)$ and $\mathcal{GF}(x_0)$, which means that the condition $\mathcal{GF}(y_0) \approx \mathcal{GF}(x_0)$ is more accurate than $\mathcal{GF}(y_0) \approx x_0$. Thus, we choose $\mathcal{GF}(y_0) \approx \mathcal{GF}(x_0)$ in our experiments.

However, if we only utilize the above condition, the training is unstable, and distortion may be observed. To address this problem, another condition needs to be introduced. The above condition refers to denoising, so conversely, we can consider adding noise to x_0; that is, $y_0 \approx \mathcal{A}(x_0)$. According to the characteristics of PET noise, Poisson noise is used in the sinogram domain instead of the image domain. We define this condition as $\mathcal{P}^\dagger(Po(\mathcal{P}(x_0) + r + s)) \approx y_0$, where $\mathcal{P}$, Po, $\mathcal{P}^\dagger$, r and s represent the Radon transform, Poisson noise insertion, inverse Radon transform, random coincidence and scatter coincidence, respectively.

Now, we have two conditions $\mathcal{GF}(y_0) \approx \mathcal{GF}(x_0)$ and $\mathcal{P}^\dagger(Po(\mathcal{P}(x_0)+r+s)) \approx y_0$ from the perspectives of denoising and noise insertion, respectively. Since the conditions involve x_0, we have to convert the conditions from the original data space into latent space under certain circumstances to avoid estimating x_0. Let us denote each transition in the reverse process under global conditions as:

$$\begin{aligned} p_\theta(x_{t-1}|x_t, c_1, c_2) \ = & p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_0) = \mathcal{GF}(y_0), \\ & \mathcal{P}^\dagger(Po(\mathcal{P}(x_0) + r + s)) = y_0). \end{aligned} \tag{6}$$

In Eq. (2), x_t can be represented by a linear combination of x_0 and ϵ. Then, we can express x_0 with x_t and ϵ:

$$x_0 \approx f_\theta(x_t, t) = (x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t))/\sqrt{\bar{\alpha}_t}. \tag{7}$$

Similarly, applying the same diffusion process to y_0, we have $\{y_1, y_2, ..., y_T\}$, and y_0 can be expressed with y_t and ϵ:

$$y_0 \approx f_\theta(y_t, t) = (y_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(y_t, t))/\sqrt{\bar{\alpha}_t}. \tag{8}$$

Replacing x_0 and y_0 with $f_\theta(x_t, t)$ and $f_\theta(y_t, t)$ in Eq. (6), respectively, we have:

$$\begin{aligned} p_\theta(x_{t-1}|x_t, c_1, c_2) \ \approx & \mathbb{E}_{q(y_{t-1}|y_0)}[p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_{t-1}) = \mathcal{GF}(y_{t-1}), \\ & \mathcal{P}^\dagger(Po(\mathcal{P}(x_{t-1}) + r + s)) = y_{t-1})]. \end{aligned} \tag{9}$$

Assume that

$$\begin{aligned} x_{t-1} \ = & (1-\lambda)(\mathcal{GF}(y_{t-1}) + (\mathcal{I} - \mathcal{GF})(x'_{t-1})) \\ & + \lambda(y_{t-1} + x'_{t-1} - \mathcal{P}^\dagger(Po(\mathcal{P}(x_{t-1}) + r + s)), \end{aligned} \tag{10}$$

where x'_{t-1} is sampled from $p_\theta(x'_{t-1}|x_t)$, and $\lambda \in [0,1]$ is a balancing factor between the two conditions. Thus, we have

$$\begin{aligned}&\mathbb{E}_{q(y_{t-1}|y_0)}[p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_{t-1}) = \mathcal{GF}(y_{t-1}), \mathcal{P}^\dagger(Po(\mathcal{P}(x_{t-1}) + r + s)) = y_{t-1})] \\ \approx &p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_{t-1}) = \mathcal{GF}(y_{t-1}), \mathcal{P}^\dagger(Po(\mathcal{P}(x_{t-1}) + r + s)) = y_{t-1}).\end{aligned} \tag{11}$$

Finally, we have

$$\begin{aligned}&p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_0) = \mathcal{GF}(y_0), \mathcal{P}^\dagger(Po(\mathcal{P}(x_0) + r + s)) = y_0) \\ =&p_\theta(x_{t-1}|x_t, \mathcal{GF}(x_{t-1}) = \mathcal{GF}(y_{t-1}), \mathcal{P}^\dagger(Po(\mathcal{P}(x_{t-1}) + r + s)) = y_{t-1}),\end{aligned} \tag{12}$$

which indicates that under the assumption of Eq. (10), the global conditions on (x_0, y_0) can be converted to local conditions on (x_{t-1}, y_{t-1}) in each transition from x_t to x_{t-1}.

Now, given a low-count PET image y_0, to estimate x_0, we can sample from white noise x_T using the following two steps iteratively. The first step is to generate an immediate x'_{t-1} from $p_\theta(x'_{t-1}|x_t)$. The second step is to generate x_{t-1} from x'_{t-1} using Eq. (10). In practice, we note that there is no need to operate the two local conditions in each transition; instead, we only need the last l transitions. Generally speaking, The larger l is, the more blurred the image will be. As l decreases, the image gets more noisy. We provide the sampling procedure of BC-DPM in Algorithm 2.

Algorithm 2: Sampling stage.

Input: low-count PET image y_0, parameter θ from the training stage, hyper-parameters λ and l
Output: high-count PET image x_0
$x_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, ..., 1$ **do**
  if $t <= l$ **then**
    sample x'_{t-1} from $p_\theta(x'_{t-1}|x_t)$
    sample y_{t-1} from $q(y_{t-1}|y_0)$
    update x_{t-1} using Equation (10)
  else
    sample x_{t-1} from $p_\theta(x'_{t-1}|x_t)$
end for
return x_0

Figure 1 illustrates the whole model. In the training stage, q denotes fixed Gaussian noise for the forward process, and p_θ denotes a learned transition in the reverse process. Once p_θ is trained, it is moved to the sampling stage. In the sampling stage, we first use the same q to diffuse y_0 to $\{y_1, y_2, ..., y_T\}$. Then, we start with white noise x_T followed by a transition from x_{t+1} to x_t for each $t \in \{0, 1, 2, ..., T-1\}$. Each transition consists of p_θ and a conditional

block. p_θ is responsible for sampling an immediate x'_t from x_{t+1}. Then, the conditional block takes x'_t and y_t as inputs and outputs x_t. Figure 2 shows the detailed structure of the conditional block. There are two parallel branches. One calculates the difference between $\mathcal{GF}(x'_t)$ and $\mathcal{GF}(y_t)$, which represents the condition of denoising, and the other computes the difference between $\hat{x}'_t$ and y_t, where $\hat{x}'_t$ is derived by adding noise to x'_t in the sinogram domain. Then, we sum the two branches weighted by λ and subtract x'_t to output the final result x_t.

3 Experiment

3.1 Experimental Setup

To evaluate the proposed method, real clinical data downloaded from TCIA were tested [2]. The computer simulation modeled the geometry of a CTI ECAT 921 PET scanner, and the system matrix P was modeled using Siddon's refined method to calculate the ray path integral [5]. To simulate low-count PET images, we first generated a noise-free sinogram by forward projecting the original data, obtaining a sinogram with a matrix size of 160 (radial bins) $\times$ 192 (azimuthal angles). Then, uniform random events were added to the noise-free sinogram as background, which accounted for 20% of total true coincidences. Independent Poisson noise with different levels was injected, raising the total number of events to 1M, 0.3M, and 0.1M, respectively. Finally, these sinograms were reconstructed by the ML-EM algorithm with 100 iterations. We used 3000 2D slices from 60 patients as the training set and 400 slices from another 10 patients as the test set.

Our method was implemented with PyTorch on a GeForce GTX 1080Ti GPU. We trained the network using the AdamW algorithm with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $weight\_decay = 0.01$. The learning rate was set to 0.0001, and the batch size was 8. In our experiments, similar to DDPM, we set the number of diffusion steps to $T = 1000$. For the variance schedule in the forward process, we employed a linear schedule from $\beta_1 = 0.0001$ to $\beta_T = 0.02$. In the sampling stage, we evenly sampled 100 steps from 1 to T and then performed generation only on these 100 steps, reducing the number of steps from 1000 to 100 by employing the trick in [7]. For the count levels of 1M, 0.3M, and 0.1M in the real clinical data study, we set l to 5, 10, and 15, respectively. In all cases, we set $\lambda = 0.2$ to balance the two conditions. As the diffusion model can generate different results due to stochasticity, we ran the model five times and used the average of the five results as the final result.

We compared our method with two conventional methods, Gaussian Filter and BM3D, and two unsupervised/unpaired methods, Noise2Void with parameter transfer (N2V-PT) [10] and unsupervised CycleWGAN [13].

3.2 Experimental Results

Figure 3 shows the results using various methods at three count levels. It can be observed that our method obtains the best performance in all cases. At the 1M

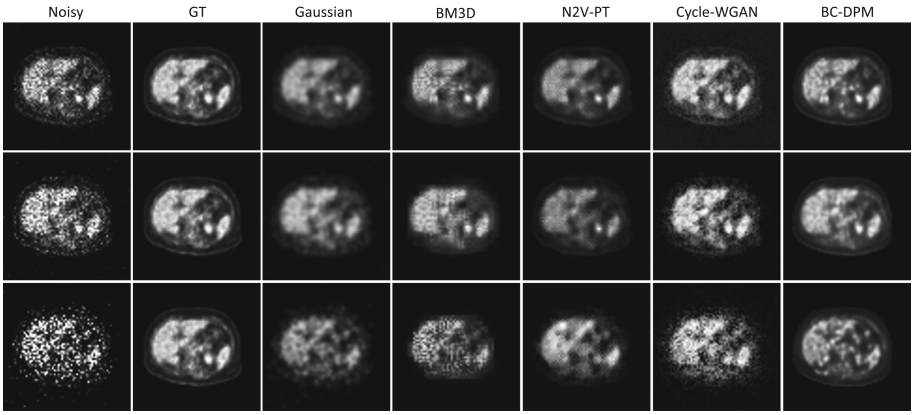

Fig. 3. Denoising results from different methods. The first row is under a count level of 1M, the second row is under a count level of 0.3M, and the third row is under a count level of 0.1M.

Table 1. PSNR and SSIM values of various methods of patient data under different count levels.

Methods	1M		0.3M		0.1M	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Gaussian	32.9406	0.9713	31.5443	0.9611	30.9818	0.9531
BM3D	32.0151	0.9564	31.6693	0.9527	28.2746	0.9117
N2V-PT	34.4333	0.9767	33.4353	0.9695	31.0254	0.9545
CycleWGAN	35.1976	0.9821	33.6778	0.9653	31.6183	0.9500
BC-DPM	**35.6297**	**0.9831**	**33.9297**	**0.9700**	**32.1648**	**0.9621**

count level, the noise is small, and all methods obtain adequate results. At the 0.3M count level, the noise becomes higher. The Gaussian filter compromises the details for noise reduction. N2V-PT exhibits strange patterns due to the violation of the pixel independence assumption in PET noise. At the extremely low-count level, 0.1M, the Gaussian filter and N2V-PT cannot obtain clinically useful results, while our method can accurately recover some details due to its strong capacity for generation under these conditions. Table 1 reports the quantitative results, showing that our proposed BC-DPM outperforms other methods in terms of both PSNR and SSIM.

4 Conclusion

In conclusion, a PET denoising model based on diffusion probabilistic models is proposed in this paper. Our model is trained in an unsupervised manner and denoises low-count PET images without any anatomical prior as a reference. To

enable the DPM to generate high-count PET images from corresponding low-count PET images, we design bidirectional conditions derived from relations between the low-count image and the potential high-count image. One condition is that the denoised low-count image approximates the high-count image. The other is that after adding noise, the high-count image approximates the low-count image. For implementation, we transfer the bidirectional conditions to latent space, which helps free the model from its dependence on the high-count image. Experiments on real clinical data demonstrate that our model is superior in noise suppression and detail preservation to other state-of-the-art methods.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 62271335; in part by the Sichuan Science and Technology Program under Grant 2021JDJQ0024; and in part by the Sichuan University "From 0 to 1" Innovative Research Program under Grant 2022SCUH0016.

References

1. Choi, J., et al.: ILVR: conditioning method for denoising diffusion probabilistic models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 14347–14356 (2021)
2. Clark, K., et al.: The cancer imaging archive (TCIA): maintaining and operating a public information repository. J. Digit. Imag. **26**, 1045–1057 (2013)
3. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. Proc. Adv. Neural Inf. Process. Syst. **34**, 8780–8794 (2021)
4. Gong, K., et al.: PET image denoising based on denoising diffusion probabilistic models. arXiv preprint arXiv:2209.06167 (2022)
5. Han, G., Liang, Z., You, J.: A fast ray-tracing technique for TCT and ECT studies. In: Proceedings of the IEEE Nuclear Science Symposium, vol. 3, pp. 1515–1518 (1999)
6. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. **33**, 6840–6851 (2020)
7. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: Proceedings of the International Conference on Machine Learning, vol. 139, pp. 8162–8171 (2021)
8. Saharia, C., et al.: Image super-resolution via iterative refinement. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
9. Sohl-Dickstein, J., et al.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of the International Conference on Machine Learning, pp. 2256–2265 (2015)
10. Song, T.A., et al.: Noise2Void: unsupervised denoising of PET images. Phys. Med. Biol. **66** (2021)
11. Song, Y., et al.: Score-based generative modeling through stochastic differential equations. In: Proceedings of the International Conference on Learning Representations (2021)
12. Song, Y., et al.: Solving inverse problems in medical imaging with score-based generative models. In: Proceedings of the International Conference on Learning Representations (2022)
13. Zhou, L., et al.: Supervised learning with cyclegan for low-dose FDG PET image denoising. Med. Image Anal. **65**, 101770 (2020)