PPFM: Image Denoising in Photon-Counting CT Using Single-Step Posterior Sampling Poisson Flow Generative Models

Dennis Hein, Staffan Holmin, Timothy Szczykutowicz, Jonathan S. Maltz, Mats Danielsson, Ge Wang, *Fellow, IEEE*, and Mats Persson

***Abstract*—Diffusion and Poisson flow models have shown impressive performance in a wide range of generative tasks, including low-dose CT (LDCT) image denoising. However, one limitation in general, and for clinical applications in particular, is slow sampling. Due to their iterative nature, the number of function evaluations (NFEs) required is usually on the order of $10 - 10^3$, both for conditional and unconditional generation. In this article, we present posterior sampling Poisson flow generative models (PPFMs), a novel image denoising technique for low-dose and photon-counting CT that produces excellent image quality whilst keeping NFE = 1. Updating the training and sampling processes of Poisson flow generative models (PFGMs)++, we learn a conditional generator which defines a trajectory between the prior noise distribution and the posterior distribution of interest. We additionally hijack and regularize the sampling process to achieve NFE = 1. Our results shed light on the benefits of the PFGM++ framework compared to diffusion models. In addition, PPFM is shown to perform favorably compared to current state-of-the-art diffusion-style models with NFE = 1, consistency models, as well as popular deep learning and nondeep learning-based image denoising techniques, on clinical LDCT images and clinical images from a prototype photon-counting CT system.**

***Index Terms*—Deep learning, denoising, diffusion models, photon-counting computed tomography (PCCT), Poisson flow generative models (PFGMs).**

I. Introduction

COMPUTED tomography (CT) is a widely used medical imaging modality providing cross-sectional images of the patient used to detect pathological abnormalities. CT is used as a tool both for diagnosis and treatment planning for a wide range of disease, such as stroke, cancer, and cardiovascular disease. However, the potential risk associated with ionizing radiation [1], [2] has spurred on a huge research endeavor to achieve images of high diagnostic quality while keeping the dose as low as reasonably achievable [3], [4]. Photon-counting CT (PCCT), based on the latest generation of CT detector technology, inherently contributes toward this objective as it is able to reduce dose via photon energy weighting and by largely eliminating the effects of electronic noise. This novel detector technology, in addition to improved low-dose imaging, yields major improvements in spatial and energy resolution [5], [6], [7], [8], [9], [10] both extremely valuable to provide accurate diagnosis. However, obtaining high resolution in either space or energy decreases the number of photons in each respective voxel or energy bin, and this unavoidably increases image noise. Hence, to materialize the full potential of the latest in X-ray CT detector technology there is an even higher demand for high quality image denoising techniques.

Existing image denoising techniques can roughly be categorized into: 1) iterative reconstruction [11], [12], [13], [14], [15], [16], [17]; 2) preprocessing methods [18]; and 3) postprocessing methods [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. Iterative reconstruction have proved to be successful in generating images with low noise levels while keeping important details intact. However, these methods are usually computationally expensive. Preprocessing methods approach the problem in the sinogram domain, prior to image reconstruction. The advantage of this method is that it will be agnostic to specific parameters used in the image reconstruction (kernel, matrix size, field of view (FOV), etc.). However, as the sinogram is in general of higher dimension than the reconstructed image, these approaches impose a higher compute requirement and may simply be unfeasible in certain applications.

Manuscript received 19 December 2023; revised 10 April 2024; accepted 30 May 2024. Date of publication 11 June 2024; date of current version 5 September 2024. This work was supported in part by MedTechLabs; in part by GE HealthCare; in part by the Swedish Research council under Grant 2021-05103; and in part by the Göran Gustafsson Foundation under Grant 2114. *(Corresponding author: Dennis Hein.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Swedish Ethics Review Agency under Application No. 2020-04638 and 2021-01092 and by the UW Madison Institutional Review board under Application No. 2022-1043 and performed in line with relevant Swedish and US legislation.

Dennis Hein, Mats Danielsson, and Mats Persson are with the Department of Physics, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden, and also with the MedTechLabs, BioClinicum, Karolinska University Hospital, 171 76 Stockholm, Sweden (e-mail: dhein@kth.se).

Staffan Holmin is with the Department of Clinical Neuroscience, Karolinska Institute, 171 77 Stockholm, Sweden, and also with the Department of Neuroradiology, Karolinska University Hospital, 171 76 Stockholm, Sweden.

Timothy Szczykutowicz is with the Department of Radiology, University of Wisconsin School of Medicine and Public Health, Madison, WI 53726 USA.

Jonathan S. Maltz is with the GE HealthCare, Grand View Boulevard, Waukesha, WI 53188 USA.

Ge Wang is with the Department of Biomedical Engineering, School of Engineering, Biomedical Imaging Center, Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRPMS.2024.3410092.

Digital Object Identifier 10.1109/TRPMS.2024.3410092

Postprocessing alleviates these issues by operating directly in the image domain. Popular postprocessing methods include nonlocal means (NLMs) [19], [20] and block-matching 3D (BM3D) [28] filtering, as well as deep learning-based methods [21], [22], [23], [24], [25], [26], [27], [29], [30], [31], [32], [33]. In particular, deep generative models have proved exceptionally capable in suppressing noise while preventing over-smoothing and thereby generating processed images with appealing noise characteristics [22], [23], [32], [33], [34], [35], [36], [37], [38]. It is also possible to combine pre- and postprocessing by considering both the image and sinogram domain within one method, as done in [39].

Diffusion and Poisson flow models are relatively recent deep generative models that have shown excellent performance on a wide range of tasks, showing remarkable success for unconditional [40], [41], [42], [43], [44], [45], [46], [47] and conditional image generation [32], [33], [43], [48], [49], [50], [51], [52], [53]. These families of generative models, lend themselves very well for inverse problem solving, ubiquitous in medical imaging, and have already been demonstrated on a range of problems in medical imaging [32], [33], [50], [53], [54]. Despite being based on two widely different underlying physical processes, EDM [45] (diffusion models) and Poisson flow generative models (PFGMs)++ [47] are intimately connected in theory and in practice. The generative processes both work by iteratively denoising images, starting from an initial prior noise distribution, following some physically meaningful trajectory. The former is inspired by nonequilibrium thermodynamics and the latter by electrostatics. PFGM++ realizes a generative model by treating N-dimensional data as electric charges in an $N + D$-dimensional augmented space. Tracing out the resulting electric field lines yields a trajectory, defined by an ordinary differential equation (ODE), from an easy-to-sample prior distribution to the data distribution of interest. Amazingly, the training and sampling processes of PFGM++ converges to that of EDM in the $D \to \infty$, $r = \sigma\sqrt{D}$ limit [47]. In other words, PFGM++ contains diffusion models as a special case. In addition, EDM and PFGM++ are also tightly connected in practice. As show in [47], the training and sampling algorithms introduced for EDM [45] can directly be applied to PFGM++ with just an updated prior noise distribution and a simple change of variables.

The iterative sampling process is a key feature of diffusion-style models, such as diffusion and Poisson flow models. This allows for a flexible tradeoff between compute and image quality as well as zero-shot editing of data. However, this is also a key limitation as more compute means slower sampling which may limit their use in real-time applications. Compared to single-step models, such as GANs [55], diffusion-style models may required on the order of $10 - 10^3$ times more compute to generate a sample, both for unconditional and conditional generation. Considering clinical CT image denoising as an example, a full 3-D volume may contain hundreds of slices that promptly need to be processed. Efforts to reduce the number of function evaluations (NFEs), and improve sampling speeds, include moving to efficient ODE samplers [44] and distillation techniques [57]. A recent development is consistency models [56], which builds upon of probability flow diffusion models and learns to map any point at any time-step to the trajectory's initial point. This is achieved by enforcing self-consistency: any two points on the same trajectory maps to the same initial point. A consistency model can be trained in distillation mode [consistency distillation (CD)], where a pretrained diffusion model is distilled into a single-step sampler, and in isolation mode (consistency training), where a consistency model is trained from scratch as a stand-alone model. Although yielding impressive results, there is a noticeable drop in performance when comparing the output from the consistency model with NFE $= 1$ to the underlying diffusion model with NFE$>$1. This drop in performance is smaller for CD than for consistency training and can be mitigated by taking a few more steps in the sampling process.

In this article, we propose a novel postprocessing denoising method that exploits the added robustness afforded by choosing D in the PFGM++ framework to achieve high image quality without the penalty of computationally costly sampling. The main contributions are as follows.

1) We present posterior sampling poisson flow generative models (PPFMs), a novel framework for image denoising in low-dose and photon-counting CT that produces excellent image quality whilst keeping NFE $= 1$. Using PFGM++ [47], originally developed for unconditional generation (noise-to-image), as starting point, we update the training and sampling processes, utilizing paired data to learn a conditional generator (image-to-image). Intuitively, instead of estimating an empirical electric field as in PFGM++ [47], we exploit the additional information afforded by paired data to estimate a "conditional" empirical electric field, which defines a trajectory from the prior noise distribution to the posterior distribution of interest. While not strictly necessary in order to get a sample from the desired posterior, we additionally hijack and regularize the sampling process. Using this formulation we can choose the hyperparameters such that NFE $= 1$.
2) We shed light on the benefits of using the PFGM++ framework with variable D compared to diffusion models with $D \to \infty$ fixed for the task of image denoising. The corresponding posterior sampling method based on diffusion models is contained as a special case ($D \to \infty$) in our proposed method and our results indicate that the PFGM++ framework, with D as an additional hyperparameter, yields significant performance gains.
3) We show that our proposed method outperforms current state-of-the-art diffusion-style models with NFE $= 1$, consistency models [56]. In addition to the state-of-the-art from the AI literature, we also compare our proposed method to previous popular supervised (RED-CNN [21], WGAN-VGG [23]), and nondeep learning-based (BM3D [28]) image denoising techniques. Our results indicate superior performance on clinical low-dose CT (LDCT) images and clinical

images from a prototype photon-counting CT scanner developed by GE HealthCare, Waukesha [60].

Code used for this article is available at: https://github.com/dennishein/cpfgmpp_PCCT_denoising.

II. Methods

A. Problem Formulation

The objective in this article is to generate high-quality reconstructions $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ of $\boldsymbol{y} \in \mathbb{R}^N$ from noise degraded $\boldsymbol{c} = \mathcal{F}(\boldsymbol{y}) \in \mathbb{R}^N$, where $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^N$ denotes the noise degradation operator, including factors, such as quantum noise [21], and $N := n \times n$. In the case of LDCT, $\boldsymbol{y}$ corresponds to the normal-dose CT (NDCT) and $\boldsymbol{c}$ to the LDCT image. In the case of photon-counting CT, $\boldsymbol{c}$ is the thin unprocessed slice and $\boldsymbol{y}$ is its noise suppressed counterpart. The problem of generating high-quality reconstructions $\hat{\boldsymbol{y}}$ of $\boldsymbol{y}$ from measurements $\boldsymbol{c}$ is typically ill-posed. It helpful to treat this as a statistical inverse problem, and we will assume that the data follow some prior distribution $\boldsymbol{y} \sim p(\boldsymbol{y})$. Our high-quality reconstruction is then a sample from the posterior $\hat{\boldsymbol{y}} \sim p(\boldsymbol{y}|\boldsymbol{c})$. This strategy for solving inverse problem is called posterior sampling. In this article, $\boldsymbol{y}$ will be treated as "ground truth" despite the fact that it may contain noise and artifacts.

B. Diffusion Models

Diffusion models [40], [41], [42], [43], [44], [45], originally inspired by nonequilibrium thermodynamics, work by first slowly transforming the data distribution to a noise distribution by iteratively adding Gaussian noise, and subsequently learning to run the process in reverse, slowly removing the noise. Building on the continuous-time probability flow ODE formulation in [43], Karras et al. [45] described this process as

$$d\boldsymbol{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p_{\sigma(t)}(\boldsymbol{x})dt \tag{1}$$

where $\sigma(t) \in [\sigma_{\min}, \sigma_{\max}]$ is a predefined, time-dependent, noise scale, and $\nabla_{\boldsymbol{x}} \log p_{\sigma(t)}(\boldsymbol{x})$ is the time-dependent score function of the perturbed data distribution. Moving the ODE forward and backward in time nudges the sample away from and toward the data distribution, respectively. Crucially, the ODE in (1) only depends on the data distribution via the time-dependent score function, an estimate of which can be obtained by minimizing the weighted denoising score matching [58] objective

$$\mathbb{E}_{\sigma\sim p(\sigma)}\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\mathbb{E}_{\boldsymbol{x}\sim p_\sigma(\boldsymbol{x}|\boldsymbol{y})} \left[\lambda(\sigma)||f_\theta(\boldsymbol{x},\sigma) - \nabla_{\boldsymbol{x}} \log p_\sigma(\boldsymbol{x}|\boldsymbol{y})||_2^2\right] \tag{2}$$

where $\lambda(\sigma)$ is a weighting function, $p(\sigma)$ is the training distribution of noise scales, $p(\boldsymbol{y})$ is the data distribution, and $p_\sigma(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}, \sigma^2\boldsymbol{I})$ is the Gaussian perturbation kernel, which samples perturbed data $\boldsymbol{x}$ from ground truth data $\boldsymbol{y}$. Once equipped with this estimate, we can generate an image by drawing an initial sample from the prior noise distribution and solving (1) using some numeric ODE solver.

C. PFGM++

Instead of estimating a time-dependent score function, as for score-based diffusion models, the objective of interest in PFGM++ is the high dimensional electric field

$$\boldsymbol{E}(\tilde{\boldsymbol{x}}) = \frac{1}{S_{N+D-1}(1)}\int \frac{\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}}{||\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}||^{N+D}} p(\boldsymbol{y})d\boldsymbol{y} \tag{3}$$

where $p(\boldsymbol{y})$ is the ground truth data distribution, $S_{N+D-1}(1)$ is the surface area of the unit $(N + D - 1)$-sphere, and $\tilde{\boldsymbol{y}} := (\boldsymbol{y}, \boldsymbol{0}) \in \mathbb{R}^{N+D}$ and $\tilde{\boldsymbol{x}} := (\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{R}^{N+D}$ are the augmented ground truth and perturbed data, respectively. The electric field lines, generated by the data treated as electric charges in the augmented space, define a surjection between the ground truth data distribution and a uniform distribution on the infinite $N+D$-dimensional hemisphere. Importantly, the electric field is rotationally symmetric on the D-dimensional cylinder $\sum_{i=1}^D z_i^2 = r^2 \;\; \forall r > 0$ and therefore a dimensionality reduction is possible [47]. In particular, it suffices to track the norm of the augmented variables $r = r(\tilde{\boldsymbol{x}}) := ||\boldsymbol{z}||_2$ and we can redefine $\tilde{\boldsymbol{y}} := (\boldsymbol{y}, 0) \in \mathbb{R}^{N+1}$ and $\tilde{\boldsymbol{x}} := (\boldsymbol{x}, r) \in \mathbb{R}^{N+1}$. Hence, the ODE of interest is

$$d\boldsymbol{x} = \boldsymbol{E}(\tilde{\boldsymbol{x}})_{\boldsymbol{x}} \cdot E(\tilde{\boldsymbol{x}})_r^{-1} dr \tag{4}$$

where $\boldsymbol{E}(\tilde{\boldsymbol{x}})_{\boldsymbol{x}} = [1/S_{N+D-1}(1)] \int (\boldsymbol{x} - \boldsymbol{y}/[||\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}||^{N+D}])p(\boldsymbol{y})d\boldsymbol{y}$, and $E(\tilde{\boldsymbol{x}})_r = [1/S_{N+D-1}(1)] \int (r/[||\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}||^{N+D}])p(\boldsymbol{y})d\boldsymbol{y}$, a scalar. Crucially, this symmetry reduction has converted the aforementioned surjection into a bijection between the ground truth data placed on the $r = 0$ ($\boldsymbol{z} = \boldsymbol{0}$) hyperplane and a distribution on the $r = r_{\max}$ hypercylinder [47]. PFGM++ employs a perturbation-based objective, akin to the denoising score matching objective in score-based diffusion models [43], [45]. In particular, for the perturbation kernel $p_r(\boldsymbol{x}|\boldsymbol{y})$, the objective is

$$\mathbb{E}_{r\sim p(r)}\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x}|\boldsymbol{y})}\left[\left\| f_\theta(\tilde{\boldsymbol{x}}) - \frac{\boldsymbol{x} - \boldsymbol{y}}{r/\sqrt{D}}\right\|_2^2\right] \tag{5}$$

where $p(r)$ is the training distribution over r. The key idea is that we can choose the perturbation kernel such that the minimizer of (5) matches (4). In particular, for $p_r(\boldsymbol{x}|\boldsymbol{y}) \propto 1/(||\boldsymbol{x} - \boldsymbol{y}||_2^2 + r^2)^{(N+D/2)}$, it is possible to show that the minimizer of (5) is $f_\theta^*(\tilde{\boldsymbol{x}}) = \sqrt{D}\boldsymbol{E}(\tilde{\boldsymbol{x}})_{\boldsymbol{x}} \cdot E(\tilde{\boldsymbol{x}})_r^{-1}$. Starting with an initial sample from $p_{r_{\max}}$ one can generate a sample for the target data distribution by solving $d\boldsymbol{x}/dr = \boldsymbol{E}(\tilde{\boldsymbol{x}})_{\boldsymbol{x}}/E(\tilde{\boldsymbol{x}})_r = f_\theta^*(\tilde{\boldsymbol{x}})/\sqrt{D}$ using some numeric ODE solver. Notably, Xu et al. [47] proved that the training and sampling processes of PFGM++ converges to that of EDM in the $D \to \infty, r = \sigma\sqrt{D}$ limit. Hence, PFGM++ accepts diffusion models as a special case.

D. Posterior Sampling Poisson Flow Generative Models

Our proposed method, PPFM, builds on PFGM++, by updating both the training and sampling processes. There are many ways to obtain a conditional generator for diffusion models, as shown in [48]. The most straightforward of which is to simply feed the condition image $\boldsymbol{c}$ as an additional input to the network estimating the time-dependent score function. This has been used with great success empirically [51], [52], and [48] showed mathematically that this "trick" has a solid

Algorithm 1: Proposed PPFM Training. Adapted From PFGM++ [47] With Adjustments Highlighted in Blue

```
Sample Data $\{\boldsymbol{y}_i, \boldsymbol{c}_i\}_{i=1}^{\mathcal{B}}$ from $p(\boldsymbol{y}, \boldsymbol{c})$
Sample standard deviations $\{\sigma_i\}_{i=1}^{\mathcal{B}}$ from $p(\sigma)$
Sample $r$ from $p_r$: $\{r_i = \sigma_i \sqrt{D}\}_{i=1}^{\mathcal{B}}$
Sample radii $\{R_i = p_{r_i}(R)\}_{i=1}^{\mathcal{B}}$
Sample uniform angles $\{\boldsymbol{v}_i = \frac{\boldsymbol{u}_i}{||\boldsymbol{u}_i||_2}\}_{i=1}^{\mathcal{B}}, \boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
Get perturbed data $\{\hat{\boldsymbol{y}}_i = \boldsymbol{y}_i + R_i \boldsymbol{v}_i\}_{i=1}^{\mathcal{B}}$
Calculate loss $\ell(\boldsymbol{\theta}) = \sum_{i=1}^{\mathcal{B}} \lambda(\sigma_i) ||D_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_i, \sigma_i, \boldsymbol{c}_i) - \boldsymbol{y}_i||_2^2$
Update network parameters $\boldsymbol{\theta}$ using Adam
```

Algorithm 2: Proposed PPFM Sampling. Adapted From PFGM++ [47] With Adjustments Highlighted in Blue

```
Get initial data $\boldsymbol{x}_\tau = \boldsymbol{c}$
for $i = \tau, \ldots, T-1$ do
    $\boldsymbol{d}_i = (\boldsymbol{x}_i - D_\theta(\boldsymbol{x}_i, t_i, \boldsymbol{c}))/t_i$
    $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + (t_{i+1} - t_i)\boldsymbol{d}_i$
    if $i < T-1$ then
        $\boldsymbol{d}'_i = (\boldsymbol{x}_{i+1} - D_\theta(\boldsymbol{x}_{i+1}, t_{i+1}, \boldsymbol{c}))/t_{i+1}$
        $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + (t_{i+1} - t_i)(\frac{1}{2}\boldsymbol{d}_i + \frac{1}{2}\boldsymbol{d}'_i)$
    end
    $\boldsymbol{x}_{i+1} = w\boldsymbol{x}_{i+1} + (1-w)\boldsymbol{x}_\tau$
end
return $\boldsymbol{x}_T$
```

theoretical background and does yield a consistent estimator of the conditional time-dependent score function. We will move from an unconditional generator to a conditional one following this strategy. For conciseness, we will leave a theoretical treatment to future work and instead illustrate empirically that this adjusted objective generates samples from the desired posterior. In practice, as is the case for PFGM++, we will employ the training and sampling algorithms from EDM [45] using an updated prior noise distribution, the $r = \sigma\sqrt{D}$ hyperparameter translation formula, $\tilde{\boldsymbol{x}} := (\boldsymbol{x}, r)$, and the fact that EDM sets $\sigma(t) = t$. Since $dr = d\sigma\sqrt{D} = dt\sqrt{D}$, by a change of variable we have that $d\boldsymbol{x} = f_\theta^*(\tilde{\boldsymbol{x}})/\sqrt{D}dr = f_\theta^*(\tilde{\boldsymbol{x}})dt$. The training process of PPFM is presented in Algorithm 1 with updates to the original formulation in PFGM++ [47] highlighted in blue[1].

Formally, the updates in Algorithm 1 are sufficient to get a conditional generator. However, we found that additionally updating the sampling process can yield significant improvements in terms of sampling speed, a key issue for diffusion-style models. Hence, we propose to hijack and regularize the sampling process. Instead of running all the way from a sample from the prior noise distribution, we will hijack the sampling process at some $i = \tau \in \mathbb{Z}_+, \tau < T$ by simply inserting our condition image $\boldsymbol{x}_\tau = \boldsymbol{c}$. Intuitively, this approach relies on the assumption that there exists a $\sigma^* = t^* \in [\sigma_{\min}, \sigma_{\max}]$ such that the condition image approximately matches a point on the solution trajectory, that is $\boldsymbol{c} \approx \boldsymbol{x}_{\sigma^*}$ where we have used the notation $\boldsymbol{x}_{\sigma^*}$ to emphasize that the perturbed $\boldsymbol{x}$ depends on the σ. Recall that in this setting t and σ can be used interchangeably. We get the corresponding r^* via the alignment of hyperparameters formula $r = \sigma\sqrt{D}$. Despite applying to CT images reconstructed with a softer kernel, our results indicate that this assumption is satisfied in practice. Consequently, the for-loop will then run from $i = \tau$ instead of $i = 0$. With this additional hyperparameter τ we have that $\text{NFE} = 2 \cdot (T - \tau) - 1$, where T is the total number of steps, or noise-scales. Initial results injecting a forward diffused condition image using the Gaussian perturbation kernel, as in done in e.g., [49] for diffusion models, did not seem to improve the results whilst introducing additional stochasticity. Thus, we decided to go with this more simplistic, and novel, approach of directly injecting the condition image $\boldsymbol{c}$. Since T is inversely proportional to the step size, $h_i = |t_{i+1} - t_i|$, employed in the ODE solver, choosing a small T is equivalent to setting a large step size. More formally, following [45], we set $t_i = (\sigma_{\max}^{(1/\rho)} + (i/T - 1)(\sigma_{\min}^{(1/\rho)} - \sigma_{\max}^{(1/\rho)}))^\rho, i = 0, \ldots, T-1$. This means that we get quite aggressive denoising but it comes at the cost of a larger local error as the local error using the 2nd order method scales as $\mathcal{O}(h^3)$ with step size h. As noted in [47], PFGM++ is relatively less sensitive to step size than EDM [45] and our results will show that using PFGM++ framework allows us to push the hyperparameters to an extreme where we have a large step size yet achieve good performance. Finally, we add a regularization step. The particular regularizer used will depend on the inverse problem at hand. Since we are here interested in image denoising, simply applying the identity map suffices. Initial results using a low-pass filtered version of $\boldsymbol{x}_\tau$, as in e.g., [53] for diffusion models, did not improve performance. Hence, we opted to go with this more simplistic formulation. In other words, we will mix $\boldsymbol{x}_{i+1}$ with $\boldsymbol{x}_\tau = \boldsymbol{c}$, the input image we seek to denoise, using weight $w \in [0, 1]$. Our proposed PPFM sampling is shown in Algorithm 2, again with updates to PFGM++ [47] highlighted in blue. Together, Algorithms 1 and 2 yields our proposed method, PPFM.

III. Experiments

A. Datasets

1) Mayo Low-Dose CT Data: The dataset from the Mayo Clinic, used in the AAPM LDCT grand challenge [59], is used for training and validation. This publicly available clinical dataset contains images from 10 patients reconstructed using two different kernels and two different slice thicknesses on a 512 × 512 pixel grid. In this article, we use the data with slice thickness 1 mm and reconstruction kernel D30 (medium). We split the data into a training set containing the first 8 patients, with a total of 4800 slices, and a validation set containing the final 2 patients with a total of 1136 slices.

2) Photon-Counting CT Data: For test data, we use images gathered as a part of a clinical study of a GE prototype photon-counting system [60]. The patients were scanned at Karolinska Institute, Stockholm, and Sweden (case 1, effective diameter 28 cm and $\text{CDTI}_{\text{vol}} = 10.12$ mGy) and at the University of Wisconsin–Madison, Madison, WI (case 2, effective diameter

[1] Note that f_θ is estimated indirectly via D_θ.

TABLE I
KEY PARAMETERS USED FOR SCANNING PATIENTS ON PROTOTYPE PHOTON-COUNTING CT SYSTEM. THE PCCT DATA ARE USED FOR TESTING ONLY

	Tube current	Helical pitch	Rotation time	kVp
Case 1	255 mA	0.990:1	0.6 s	120
Case 2	290 mA	0.510:1	0.7 s	120

36 cm and $\text{CDTI}_{\text{vol}} = 27.64$ mGy) with parameters listed in Table I. We reconstructed 70 keV virtual monoenergetic images with filtered backprojection on a 512×512 pixel grid with 0.42 mm slice thickness.

B. Implementation Details

We train a network for each $D \in \{64\,128\}$ and for $D \to \infty$ for 100k iterations using Adam [61] with learning rate 2×10^{-4} and batch size of 32 on one NVIDIA A6000 48GB GPU. $D \to \infty$ is an important special case as this corresponds to the equivalent method based on diffusion models instead of PFGM++. We borrow the majority of the hyperparameters directly from [47]. We use DDPM++ with channel multiplier 128, channels per resolution [1, 1, 2, 2, 2, 2, 2], and self-attention layers at resolutions 16, 8, and 4. The only adjustment to the network architecture to move from a unconditional to a conditional generator, is to adjust the number of channels. The suggested preconditioning, exponential moving average (EMA) schedule, and nonleaky augmentation from [45] is used with an augmentation probability of 15%. We in addition set dropout probability to 10%. Following [47], we set $\sigma_{\min} = 0.002$, $\sigma_{\max} = 380$, and $\rho = 7$. The network is trained on randomly extracted 256×256 patches. Training on patches will lead to efficient training (lower graphics memory requirements) and additionally help prevent overfitting as training on randomly extracted patches serves as additional data augmentation. We train the network using mixed precision to further reduce the graphics memory requirements. τ, T, and w are crucial hyperparameters in Algorithm 2. As we only consider setups with NFE $= 1$ for our main results, $\tau = T - 1$ and hence, completely determined by T. We set T and w by grid search over $T \in \{4, 8, 16, 32, 64\}$ and $w \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ using learned perceptual image patch similarity (LPIPS) [63] on the validation set as selection criteria for each $D \in \{64\,128\}$ and for $D \to \infty$. This yields $T = 8$ and $w = 0.7$. We note that even though NFE $= 1$, this "single-step" configuration will also blend in the condition image, a second step. However, the time required for this operation is negligible and thus we still refer to this a single-step.

C. Comparison to Other Methods

Consistency models [56] are the current state-of-the-art diffusion-style models with NFE $= 1$. However, as for the case of EDM [45] and PFGM++[47], the original formulation is for the problem of unconditional image generation. To the best of our knowledge, consistency models have never been used for conditional generation. Nevertheless, since they build upon diffusion models, and the CD approach in particular distills said diffusion model to a consistency model, one can reasonably surmise that the strategy of feeding the condition images $\boldsymbol{c}$ as additional input to the network to get a conditional generator will work well. Our empirical results support this hypothesis. Starting from the official implementation[2] we employ minimal adjustments in order to learn a conditional consistency model with $\boldsymbol{c}$ as additional input using the, CD approach. We opt for the CD, instead of consistency training, as this is the top performing approach in [56]. We train the networks on randomly extracted 256×256 patches from the 8 patients in the Mayo LDCT training data. All hyperparameters for training and sampling are set as in [56] for the LSUN 256×256 experiments[3], except for batch size with had to be reduced to 4 to fit on a single NVIDIA A6000 48GB GPU. We first train an EDM for 300k iterations, and subsequently distill it into a consistency model during 600k iterations. For data augmentation, we applied random rotations and mirrorings. It is worth pointing out that the network used for CD [56] has about 530M parameters whereas the network we use in this article has 47M parameters. In addition, it is trained for considerably more iterations. Hence, both sampling and training are considerably more time consuming. In particular, despite both achieving NFE $= 1$, our proposed PPFM offer 2.5 times faster sampling. Following [56], we will refer to this consistency model as CD.

In addition to the state-of-the-art from the AI literature, we also compare our proposed method to previous popular supervised and nondeep learning-based image denoising techniques. As an example of a popular nondeep learning-based technique we use a version of BM3D [28]. BM3D was shown to be the top performer for Mayo LDCT denoising in the category of nondeep learning-based image denoising techniques in [21]. We used bm3d.py[4] and set the parameter σ_{BM3D} equal to the standard deviation of a flat region-of-interest (ROI) in the LDCT validation data. For supervised techniques, we use RED-CNN [21] and WGAN-VGG [23]. RED-CNN was trained on over 10^6 extracted overlapping 55×55 patches from the 8 patients in the Mayo LDCT training data. The architecture is set as specified in [21]. WGAN-VGG was trained on randomly extracted 64×64 patches from the training set, with network architecture and other hyperparameters as in [23]. For both networks, we augment the data by applying random rotations and mirrorings during training. WGAN-VGG is an interesting comparison case as it is very similar in principle to the method proposed in this article. Both methods achieve image denoising via posterior sampling by adjusting the training processes of deep generative models, and thereby acquire conditional generators. The major difference is the deep generative model itself. WGAN-VGG [23] is based on GANs, which were the state-of-the-art deep generative models until the event of diffusion models and PFGM++, whereas our

[2] https://github.com/openai/consistency_models

[3] As specified in https://github.com/openai/consistency_models/blob/main/scripts/launch.sh.

[4] https://pypi.org/project/bm3d/

proposed method is based on PFGM++, a current state-of-the-art deep generative model. Despite being similar in principle, this difference leads to a myriad of important differences in practice. Notably, PFGM++ does not require adversarial training and is therefore much more stable to train.

D. Evaluation Methods

In addition to image quality assessment via visual inspection, we also consider three quantitative metrics of image quality. We employ the two most commonly used metrics in the CT denoising literature, namely structural similarity index (SSIM [62]) and peak signal-to-noise ratio (PSNR). These metrics are easy to use and very well established but they do not necessarily correlate well with human perception [63]. PSNR is inversely proportional to the ℓ_2 Euclidean distance. This simple pixel-wise metrics does not adequately capture nuances of human perception. This is particularly most evident for the case of blurring as a result of over-smoothing, which is inadequately penalized. On the other hand, SSIM is perceptually motivated; however, it is very difficult to model the complex processes underlying human perception and therefore is also falls short. Zhang et al. [63] suggested using pretrained convolutional neural networks (CNNs) as feature extractors, as is the case for perceptual loss functions, to develop a metric of image similarity that closely corresponds to human perception. They call this metric LPIPS and demonstrate on a series of different datasets, using different pretrained CNNs, how LPIPS better corresponds to human perception than traditional metrics, such as SSIM and PSNR. In this article, we use the official implementation of LPIPS[5] with AlexNet [64] as feature extractor. To move from RGB to grayscale images, we use the standard approach of simply feeding a triplet of repeated grayscale images as input to the pretrained network.

E. Results

Qualitative results, along side with LPIPS, SSIM, and PSNR, for a representative case from the Mayo LDCT validation data are available in Figs. 1 and 2. This patient is of additional interest due to a metastasis in the liver. To emphasize this lesion, we include a magnified version of the ROI in Fig. 1 in Fig. 2. BM3D, shown in (c), does a good job suppressing noise and recovering details. However, this comes at a cost of artifacts that makes the image appear smudgy. RED-CNN, shown in (d), does an exceptional job of suppressing noise whilst keeping key details intact. Nevertheless, the denoising is too aggressive and the noise is suppressed well below the level in the NDCT image, shown in (a). This over-smoothing is expected since RED-CNN is trained with a simple pixel-wise ℓ_2-loss. WGAN-VGG, shown in (e), on the other hand, does a very good job at suppressing noise while producing noise characteristics aligned with that of the NDCT image. At first glance, CD, shown in (f), seems to perform exceedingly well. However, at closer inspection, especially in Fig. 2, one can see several details that appear different for CD than for all the other images, including NDCT and LDCT. We highlighted one such detail with a yellow arrow. CD seems to have added a feature that is not visible in the LDCT nor NDCT image. Seemingly convincing, but factually inaccurate, claims are commonly referred to as "hallucinations" in the large language models (LLMs) literature.[6] We will adopt this terminology to mean inaccurate addition, or removal, of features. Results for our proposed method are available in (g)–(i). PPFM, with $D = 128$ and $D = 64$, does an exceptional job of suppressing noise whilst keeping key details intact and accurately reproducing the noise characteristics of the NDCT image. Comparing (g), with $D \to \infty$, to D finite, in (h) and (i), emphasizes the effect of the added robustness afforded by choosing D in PFGM++ framework. For small T, or equivalently a large step size, PPFM with $D \to \infty$ breaks down whereas PPFM with D finite yield good results.

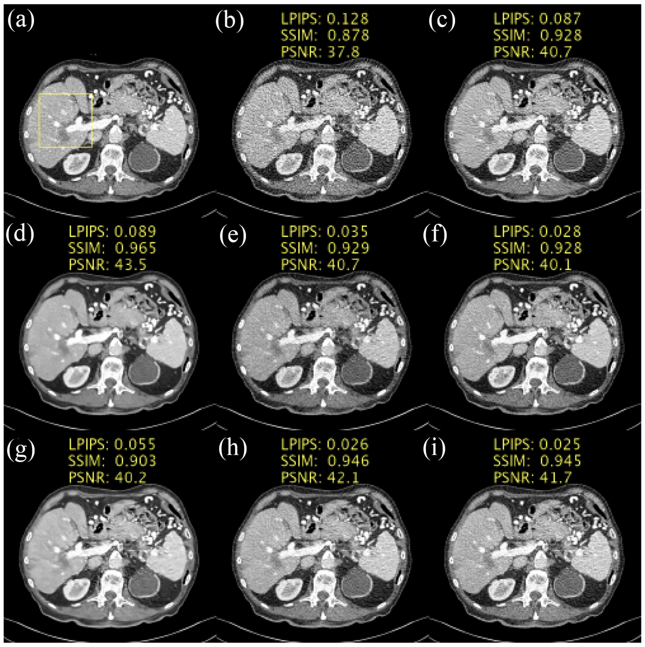

Fig. 1. Results on the Mayo low-dose CT validation data. Abdomen image with a metastasis in the liver. (a) NDCT. (b) LDCT. (c) BM3D [28]. (d) RED-CNN [21]. (e) WGAN-VGG [23]. (f) CD [56]. (g) PPFM ($D \to \infty$). (h) PPFM ($D = 128$). (i) PPFM ($D = 64$). Yellow box indicating ROI shown in Fig. 2. 1 mm-slices. Window setting [−160, 240] HU.

The mean and standard deviation of LPIPS, SSIM, and PSNR over the entire Mayo LDCT validation set are available in Table II. We additionally include the average time, in seconds, per slice for each method. The top performer in terms of SSIM and PSNR is RED-CNN. This is not entirely unexpected since RED-CNN is trained to minimize the ℓ_2-loss between patches from the NDCT and LDCT images. However, as noted above, SSIM and PSNR do not necessarily correspond well with human perception—in particular when it comes to over-smoothing. WGAN-VGG combines a perceptual loss with an adversarial loss in order to generate a denoised image from a posterior that is "close," in a certain sense, to the distribution of the NDCT images. The overall noise characteristics, texture, and level, more closely resembles that

[5] https://github.com/richzhang/PerceptualSimilarity

[6] See, for instance, reference [65] for an overview.

TABLE II
MEAN AND STANDARD DEVIATION OF LPIPS, SSIM, AND PSNR IN THE LOW-DOSE CT VALIDATION SET. IN ADDITION TO AVERAGE TIME, IN SECONDS, TO EVALUATE A SINGLE SLICE. ↓ MEANS LOWER IS BETTER. ↑ MEANS HIGHER IS BETTER. BEST RESULTS IN BOLD

	LPIPS (↓)	SSIM (↑)	PSNR (↑)	Avg. time per slice (↓)
LDCT	0.075 ± 0.02	0.94 ± 0.02	41.5 ± 1.6	
BM3D [28]	0.050 ± 0.01	0.97 ± 0.01	45.0 ± 1.6	3.38×10^{0}
RED-CNN [21]	0.048 ± 0.02	$\mathbf{0.98} \pm 0.01$	$\mathbf{46.8} \pm 1.2$	7.40×10^{-2}
WGAN-VGG [23]	0.019 ± 0.01	0.96 ± 0.01	43.2 ± 0.9	1.34×10^{-3}
CD [56]	0.013 ± 0.00	0.96 ± 0.01	43.1 ± 1.0	4.44×10^{-2}
PPFM				
$D \to \infty$	0.025 ± 0.01	0.93 ± 0.01	42.0 ± 0.7	1.77×10^{-2}
$D = 128$	0.012 ± 0.00	$\mathbf{0.98} \pm 0.01$	45.8 ± 1.4	1.86×10^{-2}
$D = 64$	$\mathbf{0.010} \pm 0.00$	0.97 ± 0.01	45.4 ± 1.4	1.47×10^{-2}

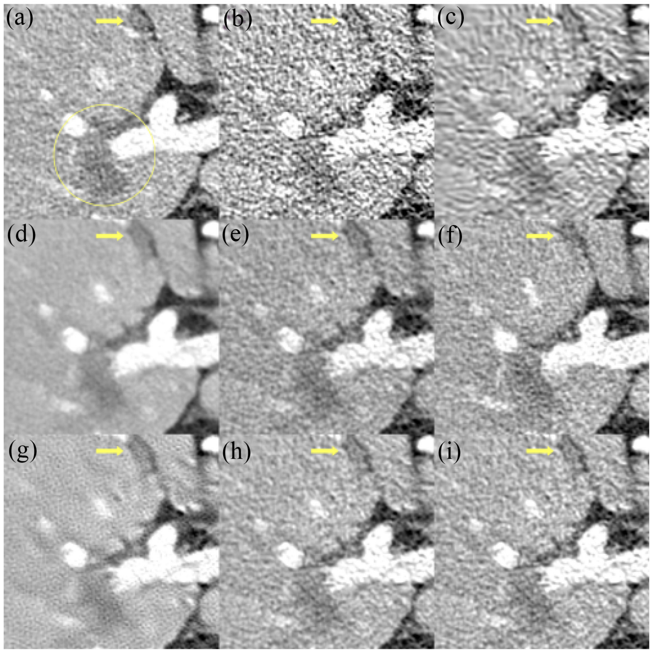

Fig. 2. ROI in Fig. 1 magnified to emphasize details. (a) NDCT. (b) LDCT. (c) BM3D [28]. (d) RED-CNN [21]. (e) WGAN-VGG [23]. (f) CD [56]. (g) PPFM ($D \to \infty$). (h) PPFM ($D = 128$). (i) PPFM ($D = 64$). Yellow circle added to emphasize lesion. Yellow arrow placed to emphasize detail. 1 mm-slices. Window setting [−160, 240] HU.

of the NDCT image for WGAN-VGG than for RED-CNN. We can see that, accordingly, the LPIPS is significantly lower (better) for WGAN-VGG than RED-CNN. The overall top performer in terms of LPIPS is our proposed method, PPFM, with $D = 64$.

To shed light on the individual components of our proposed sampler, we conduct an ablation study with results available Fig. 3(a) and (b) shows the NDCT and LDCT images, respectively. In (c), we turn off hijacking and regularization. As was also seen in Fig. 4, the sampler breaks down in this setting. The same holds true in (e), where we regularize but have turned off the hijacking. Comparing (c) to (d), we can see that hijacking plays a pivotal role in our proposed sampler. For the setting consider here, with $T = 8$, hijacking allows us to move from a total breakdown to a very pleasing image. Regularizing is also shown to be beneficial, it helps prevent over-smoothing resulting from aggressive denoising, a consequence of choosing a large step size, as can be seen when comparing (d) to (f). Hence, hijacking and regularization, hijacking in particular, is what enables excellent image quality whilst keeping NFE = 1. In order words, hijacking can help break the dependence on large T for good image quality.

The proposed method is trained in a supervised manner to directly yield a conditional estimator. Hence, as mentioned above, neither hijacking nor regularization is strictly necessary. Instead, one can simply draw an initial sample from the prior noise distribution and then solve the ODE to generate a sample from the desired posterior. To illustrate empirically that this is indeed the case, we set $\tau = 0$, $w = 1$, and replace the first line with an initial sample from the prior noise distribution, $p_{r_{\max}}$, in Algorithm 2. Hence, except for the fact that the network takes the condition image as an additional input, Algorithm 2 is exactly as in PFGM++ [47]. We show results for $T \in \{40, 100, 250, 500\}$ in Fig. 4. Consistent with expectations, the performance improves as T, the total number of steps, gets larger for the $D \to \infty$ case. Interestingly, for the $D = 64$ case, LPIPS improves, SSIM is slightly worse, and PSNR roughly stays the same. Crucially, for $D = 64$, we can see that for $T \geq 40$ our high-quality reconstruction is a good approximation of the ground truth image, that is $\hat{\boldsymbol{y}} \approx \boldsymbol{y}$. For the $D \to \infty$ case a significantly larger T is required to get good results. Moreover, comparing results in Figs. 1 and 2(i) with Fig. 4(f) illustrates the added benefit of the proposed sampler. In Fig. 4, a sample from the desired posterior is generated by starting from an initial sample from the prior noise distribution and then running Algorithm 2 with $\tau = 0$ and $w = 1$. This corresponds to employing the PFGM++ ($D = 64$), updated to the conditional case via supervised learning, as is. Compared to our proposed method in Figs. 1 and 2(i), SSIM and PSNR are marginally worse whereas LPIPS is better. Notably, NFE = 1 for (i) in Figs. 1 and 2 whereas NFE=999 in Fig. 4(f). In other words, the proposed sampling algorithm achieves comparable results quantitatively, and arguably superior results qualitatively, with up to 999× faster sampling. These improvements are made possible by our proposed sampling method which greatly regularizes the sampling process, enforcing consistency with the LDCT image, $\boldsymbol{c}$. As also shown in the ablation study in Fig. 3, the key part is the hijacking, which converts the sampling problem from a stochastic mapping from a noise vector to a deterministic mapping from the LDCT image.

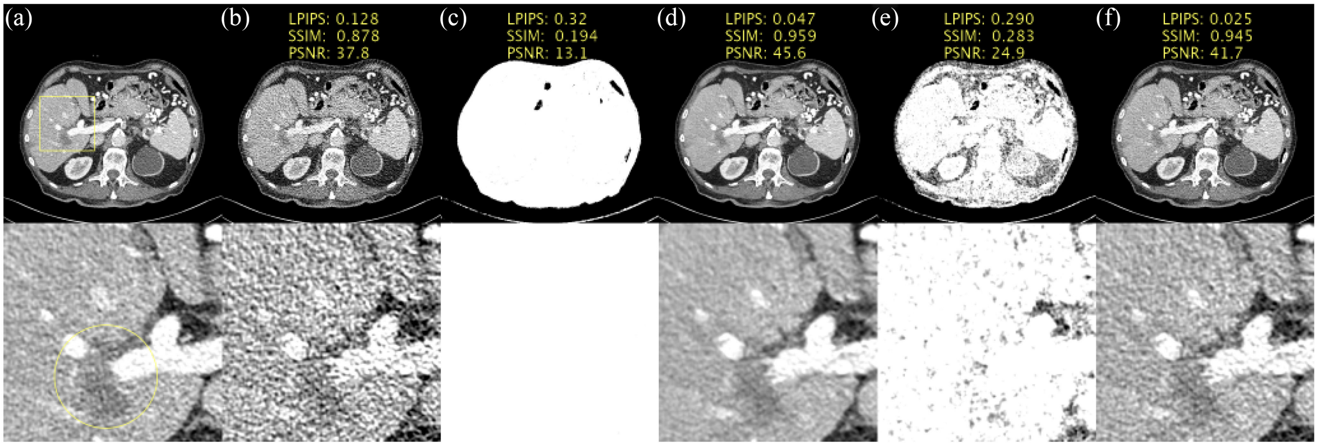

Fig. 3. Ablation study of PPFM sampler with $D = 64$ and $T = 8$. (a) NDCT. (b) LDCT. (c) No hijacking and no regularization. (d) Hijacking but no regularization ($\tau = T - 1, w = 1$). (e) No hijacking but regularization ($\tau = 0, w = 0.7$, and $\boldsymbol{x}_0$ is a sample from the prior noise distribution). (f) Hijacking and regularization ($\tau = T - 1, w = 0.7$). Yellow circle added to emphasize lesion. 1 mm-slices. Window setting [−160, 240] HU.

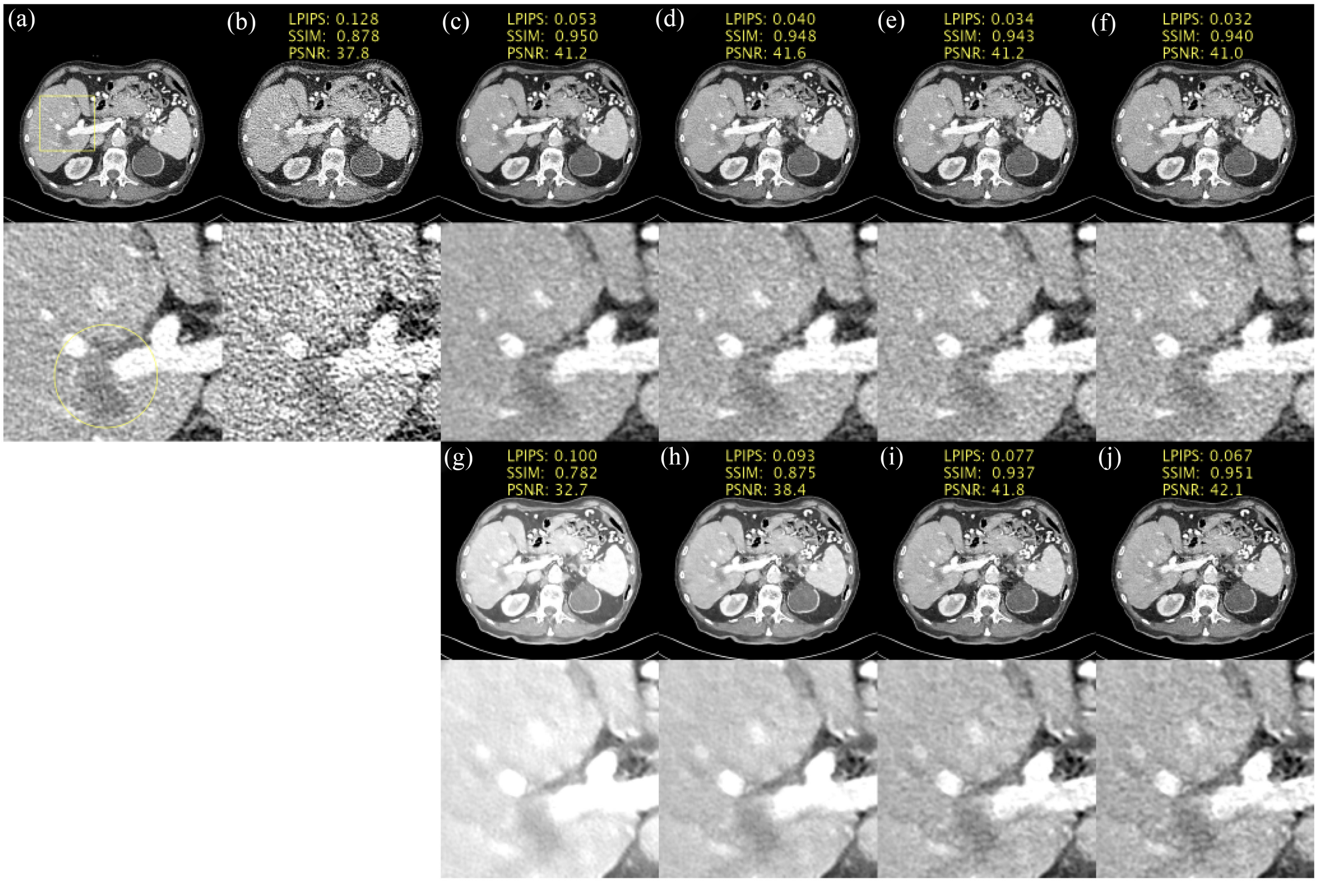

Fig. 4. Results without hijacking and regularization (i.e., $w = 1$, $\tau = 0$, and $\boldsymbol{x}_\tau = \boldsymbol{x}_0$ is a sample from the prior noise distribution in Algorithm 2). (a) NDCT. (b) LDCT. (c) $T = 40$ ($D = 64$). (d) $T = 100$ ($D = 64$), (e) $T = 250$ ($D = 64$). (f) $T = 500$ ($D = 64$). (g) $T = 40$ ($D \to \infty$). (h) $T = 100$ ($D \to \infty$). (i) $T = 250$ ($D \to \infty$). (j) $T = 500$ ($D \to \infty$). Yellow circle added to emphasize lesion. 1 mm-slices. Window setting [−160, 240] HU.

This greatly reduces the complexity of the mapping and thus enables high image quality with significantly fewer NFE. Of particular interest in medical imaging, this greatly reduces small variations in the output that leads to worse pixel-wise consistency, negatively effecting the quantitative metrics. In addition to hijacking, we also have the regularization step which further enforces pixel-wise consistency, in addition to ensuring consistency in the noise characteristics, as also shown in Fig. 3. Notably, the same conclusion does not hold true for the $D \to \infty$ case as can be seen comparing Fig. 4(j) with Figs. 1 and 2(g). In this case, the NFE=999 setup outperforms our proposed sampling. That we get good results for D finite, but not with $D \to \infty$ (EDM) is likely attributed to the added robustness over diffusion models of the PFGM++ framework.

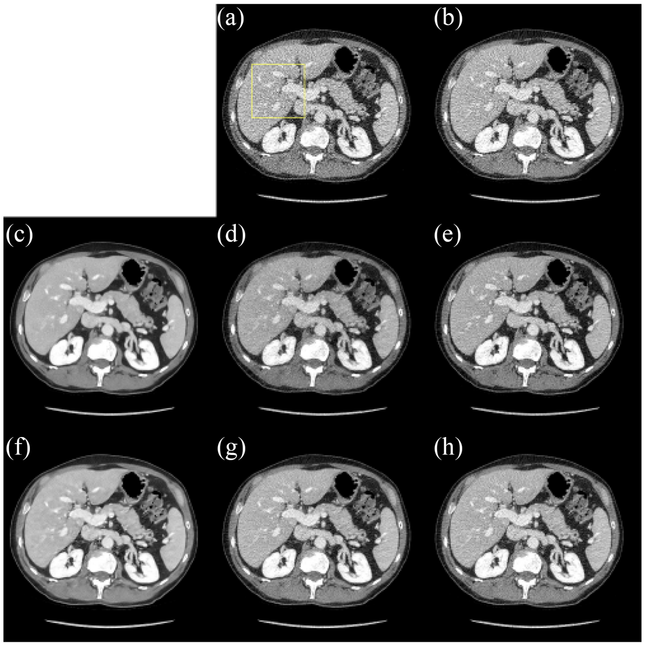

Fig. 5. Results for the PCCT test data: case 1. (a) Unprocessed. (b) BM3D [28]. (c) RED-CNN [21]. (d) WGAN-VGG [23]. (e) CD [56]. (f) PPFM ($D \to \infty$). (g) PPFM ($D = 128$). (h) PPFM ($D = 64$). No ground truth available. Yellow box indicating ROI shown in Fig. 6. 0.42 mm-slices. Window setting [−160, 240] HU.

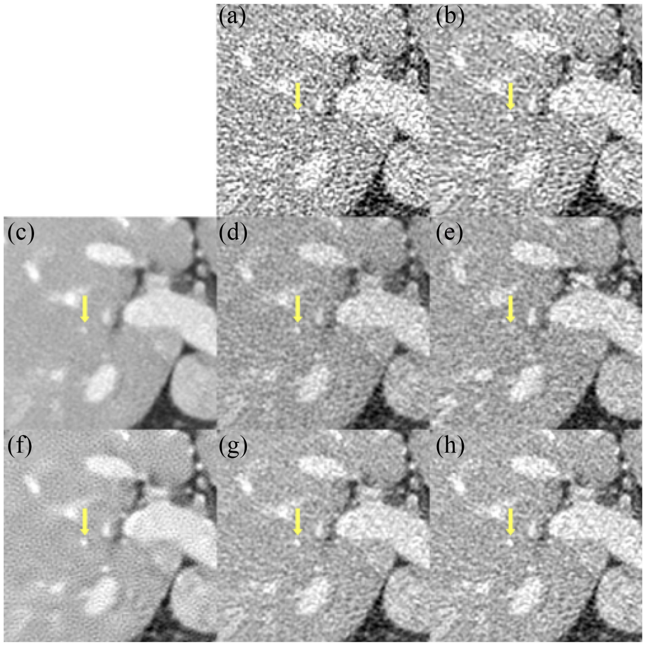

Fig. 6. ROI in Fig. 5 magnified to emphasize details. (a) Unprocessed. (b) BM3D [28]. (c) RED-CNN [21]. (d) WGAN-VGG [23]. (e) CD [56]. (f) PPFM ($D \to \infty$). (g) PPFM ($D = 128$). (h) PPFM ($D = 64$). No ground truth available. Yellow arrow placed to emphasize detail. 0.42 mm-slices. Window setting [−160, 240] HU.

Hence, using PFGM++ instead of diffusion models is key in making the NFE = 1 case viable.

Results for a representative case from the PCCT test data, case 1, are available in Figs. 5 and 6. Since these data are clinical images from a prototype photon-counting system, there are no images available to play the role of ground truth, and we will therefore have to resort to visual inspection as means of accessing image quality. Larger details can reasonably be distinguished from statistical variation in the noise; however, this is very difficult for smaller, lower contrast, details. With that caveat, since no ground truth is available, we simply define a good result as an image which preserves details visible in the unprocessed image, shown in (a), but with a lower noise level. BM3D, shown in (b), seems to generalize quite poorly. The noise level in (b) similar to that in (a) with additional artifacts that makes the image appear smudgy. This may be due to differences in noise characteristics in the validation data, where we measured σ_{BM3D}, and the test data. RED-CNN, WGAN-VGG, and CD, on the other hand, shown in (c)–(e), respectively, seem to generalize well from the LDCT data to the photon-counting CT test data. We have placed a yellow arrow on a detail of interest. This feature is clearly visible is all cases, including the unprocessed image, but it is missing for CD, shown in (e). Hence, although it is difficult to say definitively without a ground truth, this seems to indicate that CD removed a genuine feature. The proposed method is shown in (f)–(h). As was the case for the Mayo LDCT validation data, there is a major performance boost for D finite, shown in (g) and (h), compared to $D \to \infty$, shown in (f).

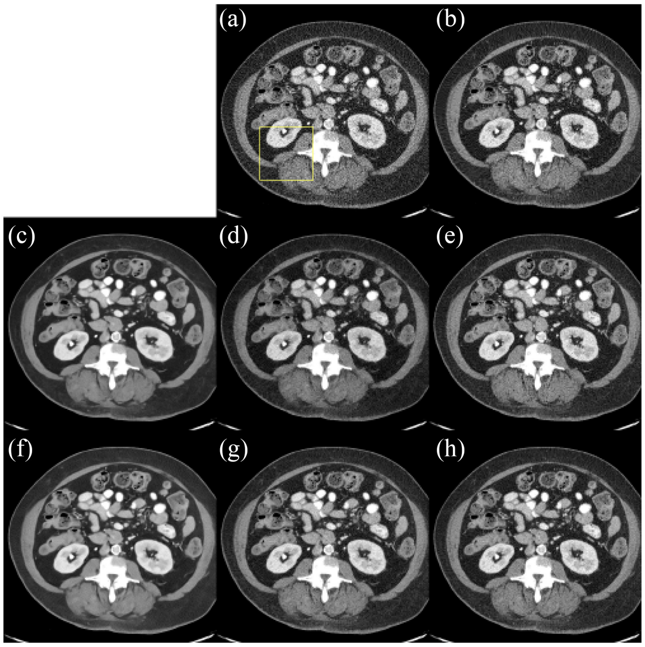

Fig. 7. Results for the PCCT test data: case 2. (a) Unprocessed. (b) BM3D [28]. (c) RED-CNN [21]. (d) WGAN-VGG [23]. (e) CD [56]. (f) PPFM ($D \to \infty$). (g) PPFM ($D = 128$). (h) PPFM ($D = 64$). No ground truth available. Yellow box indicating ROI shown in Fig. 8. 0.42 mm-slices. Window setting [−160, 240] HU.

We show the results on the second PCCT test case in Fig. 7, with a magnified version of the ROI shown in Fig. 8. We have also placed a yellow arrow in Fig. 8 to draw attention to specific details. We note that BM3D, shown in (c), seems to be doing a better job in terms of noise suppression that in Fig. 5. Differences in performance in the two test cases is most

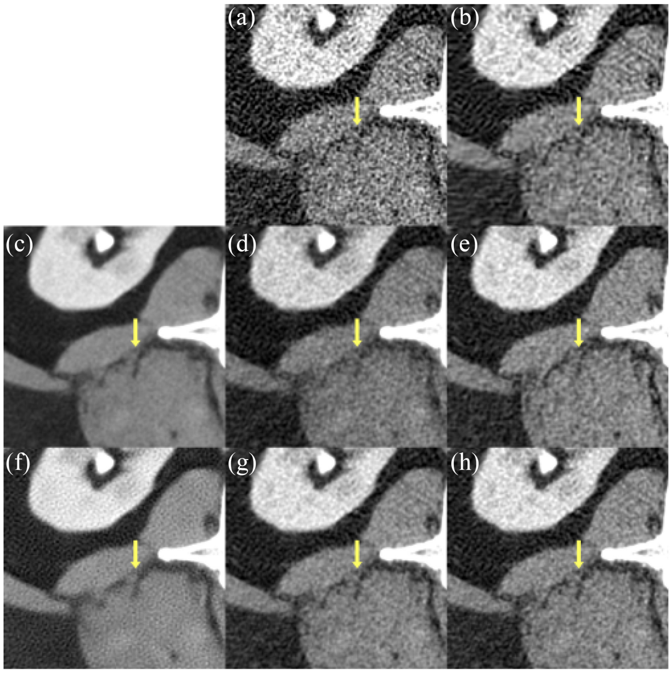

Fig. 8. ROI in Fig. 7 magnified to emphasize details. (a) Unprocessed. (b) BM3D [28]. (c) RED-CNN [21]. (d) WGAN-VGG [23]. (e) CD [56]. (f) PPFM ($D \to \infty$). (g) PPFM ($D = 128$). (h) PPFM ($D = 64$). No ground truth available. Yellow arrow placed to emphasize detail. 0.42 mm-slices. Window setting [−160, 240] HU.

likely due to differences in noise characteristics. RED-CNN, WGAN-VGG, and CD, shown in (c)–(e), respectively, all do a good job suppressing the noise while preserving details. The main difference is the characteristics, texture, and level, of the resulting noise. In particular, RED-CNN, is notably very smooth. The $D \to \infty$ case, shown in (f), over-smooths the image, reducing the contrast of key details, while introducing a strange texture. On the other hand, for finite D, PPFM results in images with realistic noise level and texture, and preserve key details. Moreover, we can see that the contrast of the fat in the back muscle, marked by the yellow arrow, is significantly better preserved using the proposed method with finite D, shown in (g) and (h), than for WGAN-VGG, shown in (d).

IV. Discussion and Conclusion

Despite appearing similar to a conventional end-to-end denoising method in the NFE $= 1$ case, our proposed approach is fundamentally different since the network is trained to estimate a high dimensional electric field (D finite, PFGM++ case) or a score function ($D \to \infty$, diffusion model case) and a sample from the posterior is subsequently obtained by following a solution trajectory via a discretized ODE. Our proposed method hijacks said solution trajectory in order to achieve NFE $= 1$ sampling. In addition, even though we show results for the special case for $\tau = T - 1$, Algorithm 2 is more general and thus the suggested approach also accepts NFE>1 sampling.

It is likely the case that one achieves better performance using a multistep sampler, trading off compute for image quality. Since we were here interested in the single-step case, only limited time was spend exploring the hyperparameters space for $\tau \neq T - 1$. In this preliminary search, we were unable to find a combination of T, τ and w outperforming our current hyperparameters in terms of LPIPS on the Mayo LDCT validation set. Notably, enforcing $\tau = T - 1$ greatly reduces the size of hyperparameter space since we only need the tuple (T, w) instead of (τ, T, w). It is left to future research to explore the extent to which there is a penalty in performance due to enforcing $\tau = T - 1$, and thereby achieving NFE $= 1$.

The added robustness of PFGM++ has already been demonstrated in previous work [47]. However, the results in Fig. 4 seem to be somewhat exaggerated given the authors experience with diffusion and Poisson flow models. In particular, given the ODE formulation in EDM [45], it is definitely possible to produce decent results for EDM with $T \approx 40$. In this case, we required around $T = 500$ to achieve reasonable results. We additionally validated these results by training a network with $D = 256 \times 256 * 10^3 >> 256 \times 256 = N$ to approximate the $D \to \infty$ case whilst staying the in the PFGM++ framework. This produced results roughly on par of what can be seen Fig. 4. This large discrepancy could be due to, for instance, robustness to network size. We leave it to future work to investigate how much this gap is narrowed by hyperparameter tuning.

Since we are interested in PCCT, the ultimate objective is to get an image denoising technique that works for spectral CT. Extending PPFM to the spectral case can be done in many different ways. One possibility is to simply expand the number of channels for each data point. Instead of feeding a single-energy image, one can use pairs of basis images or virtual monoenergetic images at two different energy levels. Assessing whether such an update would be sufficient, or if further updates are required to obtain a spectral CT denoiser is an interesting avenue for future research.

Finally, this is a 2-D image denoising method. As such, due to the nature of CT data, we are leaving an abundance of useful information on the table by not considering adjacent slices. We surmise that it should be relatively straight forward to extend the proposed method to a 3-D denoiser and thus leave this to future work.

In conclusion, we have presented PPFM, a novel image denoising technique for low-dose and photon-counting CT. Our proposed method updates the training and sample processes of PFGM++ [47] to get an conditional generator which is able to achieve high image quality without the penalty of computationally costly sampling. In particular, our proposed method is a single-step sampler, that is NFE $= 1$. Our results shed light on the benefits of building upon the PFGM++ framework, where D is a tunable hyperparameter, compared to diffusion models where $D \to \infty$ is fixed. In particular, we demonstrate that the corresponding setup with a diffusion model fails. Our results demonstrate favorable performance compared to current state-of-the-art diffusion-style models with NFE $= 1$, consistency models, as well as several popular deep learning-based and conventional postprocessing techniques on clinical LDCT images and clinical images from a prototype photon-counting CT system.

ACKNOWLEDGMENT

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests with the work reported in this article: DH discloses research collaboration with GE HealthCare. MP discloses research collaboration and license fees, GE HealthCare.

REFERENCES

[1] A. B. de González and S. Darby, "Risk of cancer from diagnostic X-rays: Estimates for the U.K. and 14 other countries," *Lancet*, vol. 363, no. 9406, pp. 345–351, 2004

[2] D. J. Brenner and E. J. Hall, "Computed tomography an increasing source of radiation exposure," *New England J. Med.*, vol. 357, no. 22, pp. 2277–2284, 2007.

[3] G. Wang, J. C. Ye, and B. De Man, "Deep learning for tomographic image reconstruction," *Nature Mach. Intell.*, vol. 2, no. 12, pp. 737–748, 2020.

[4] L. R. Koetzier et al., "Deep learning image reconstruction for CT: Technical principles and clinical prospects," *Radiology*, vol. 306, no. 3, pp. e221257–e221257, 2023.

[5] M. J. Willemink, M. Persson, A. Pourmorteza, N. J. Pelc, and D. Fleischmann, "Photon-counting CT: Technical principles and clinical prospects," *Radiology*, vol. 289, no. 2, pp. 293–312, 2018.

[6] T. Flohr, M. Petersilka, A. Henning, S. Ulzheimer, J. Ferda, and B. Schmidt, "Photon-counting CT review," *Phys. Med.*, vol. 79, pp. 126–136, Nov. 2020.

[7] M. Danielsson, M. Persson, and M. Sjölin, "Photon-counting x-ray detectors for CT," *Phys. Med. Biol.*, vol. 66, no. 3, pp. 03TR01–03TR01, 2021.

[8] S. S. Hsieh, S. Leng, K. Rajendran, S. Tao, and C. H. McCollough, "Photon counting CT: Clinical applications and future developments," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 4, pp. 441–452, Jul. 2021.

[9] K. Higashigaito et al., "Contrast-enhanced abdominal CT with clinical photon-counting detector CT: Assessment of image quality and comparison with energy-integrating detector CT," *Acad. Radiol.*, vol. 29, no. 5, pp. 689–697, 2022.

[10] K. Rajendran et al., "First clinical photon-counting detector CT system: Technical evaluation," *Radiology*, vol. 303, no. 1, pp. 130–138, 2022.

[11] J. Wang, T. Li, H. Lu, and Z. Liang, "Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose X-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 25, no. 10, pp. 1272–1283, Oct. 2006.

[12] J. B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, "A three dimensional statistical approach to improved image quality for multislice helical CT," *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, Nov. 2007.

[13] E. Y. Sidky and X. Pan, "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization," *Phys. Med. Biol.*, vol. 53, no. 17, pp. 4777–4807, Sep. 2008.

[14] H. Yu, S. Zhao, E. A. Hoffman, and G. Wang, "Ultra-low dose lung CT perfusion regularized by a previous scan," *Acad. Radiol.*, vol. 16, no. 3, pp. 363–373, Mar. 2009.

[15] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang, "Low-dose CT reconstruction via edge-preserving total variation regularization," *Phys. Med. Biol.*, vol. 56, no. 18, p. 5949, 2011.

[16] J. W. Stayman, H. Dang, Y. Ding, and J. H. Siewerdsen, "PIRPLE: A penalized-likelihood framework for incorporation of prior images in CT reconstruction," *Phys. Med. Biol.*, vol. 58, no. 21, pp. 7563–7582, 2013.

[17] G. L. Zeng and W. Wang, "Does noise weighting matter in CT iterative reconstruction?" *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 1, no. 1, pp. 68–75, Jan. 2017.

[18] P. J. La Riviere, "Penalized-likelihood sinogram smoothing for low-dose CT," *Med. Phys.*, vol. 32, no. 6, pp. 1676–1683, Jun. 2005.

[19] J. Ma et al., "Low-dose computed tomography image restoration using previous normal-dose scan," *Med. Phys.*, vol. 38, no. 10, pp. 5713–5731, 2011.

[20] Y. Zhang, Y. Xi, Q. Yang, W. Cong, J. Zhou, and G. Wang, "Spectral CT reconstruction with image sparsity and spectral mean," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 510–523, Dec. 2016.

[21] H. Chen et al., "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2524–2535, Dec. 2017

[22] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Isgum, "Generative adversarial networks for noise reduction in low-dose CT," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2536–2545, Dec. 2017.

[23] Q. Yang et al., "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.

[24] H. Shan et al., "Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction," *Nature Mach. Intell.*, vol. 1, no. 6, pp. 269–276, 2019.

[25] B. Kim, M. Han, H. Shim, and J. Baek, "A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose CT images," *Med. Phys.*, vol. 46, no. 9, pp. 3906–3923, 2019.

[26] K. Kim, S. Soltanayev, and S. Y. Chun, "Unsupervised training of denoisers for low-dose CT reconstruction without full-dose ground truth," *IEEE J. Sel. Top.*, vol. 14, no. 6, pp. 1112–1125, Oct. 2020.

[27] N. Yuan, J. Zhou, and J. Qi, "Half2Half: Deep neural network based CT image denoising without independent reference data," *Phys. Med. Biol.*, vol. 65, no. 21, pp. 215020–215020, Nov. 2020.

[28] Y. Mäkinen, L. Azzari, and A. Foi, "Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching," *IEEE Trans. Image Process.*, vol. 29, pp. 8339–8354, 2020.

[29] Z. Li, S. Zhou, J. Huang, L. Yu, and M. Jin, "Investigation of low-dose ct image denoising using unpaired deep learning methods," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 2, pp. 224–234, Mar. 2022.

[30] S. Wang, Y. Yang, Z. Yin, and A. S. Wang, "Noise2Noise for denoising photon counting CT images: Generating training data from existing scans," *Proc. SPIE*, 2023, Art. no. 1246304.

[31] C. Niu et al., "Noise suppression with similarity-based self-supervised deep learning," *IEEE Trans. Med. Imag.*, vol. 42, no. 6, pp. 1590–1602, Jun. 2023.

[32] X. Liu, Y. Xie, J. Cheng, S. Diao, S. Tan, and X. Liang, "Diffusion probabilistic priors for zero-shot low-dose CT image denoising," 2023, *arXiv:2305.15887*.

[33] M. Tivnan et al., "Fourier diffusion models: A method to control MTF and NPS in score-based stochastic image generation," 2023, *arXiv:2303.13285*.

[34] Z. Zhang, L. Yu, X. Liang, W. Zhao, and L. Xing, "TransCT: Dual-path transformer for low dose computed tomography," in *Proc. MICCAI*, 2021, pp. 55–64.

[35] D. Wang, F. Fan, Z. Wu, R. Liu, F. Wang, and H. Yu, "CTformer: Convolution-free Token2Token dilated vision transformer for low-dose CT denoising," *Phys. Med. Biol.*, vol. 68, no. 6, 2023, Art. no. 65012.

[36] L. Yang, Z. Li, R. Ge, J. Zhao, H. Si, and D. Zhang, "Low-dose CT denoising via sinogram inner-structure transformer," *IEEE Trans. Med. Imag.*, vol. 42, no. 4, pp. 910–921, Apr. 2023.

[37] Y. Lei, C. Niu, J. Zhang, G. Wang, and H. Shan, "CT image denoising and deblurring with deep learning: Current status and perspectives," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 8, no. 2, pp. 153–172, Feb. 2024.

[38] A. Bousse et al., "Systematic review on learning-based spectral CT," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 8, no. 2, pp. 113–137, Feb. 2024.

[39] C. Niu, M. Li, X. Guo, and G. Wang, "Self-supervised dual-domain network for low-dose CT denoising," in *Proc. 14th Develop. X-Ray Tomogr.*, 2022, pp. 85–91.

[40] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. ICML*, 2015, pp. 2256–2265.

[41] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020, pp. 1–12.

[42] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. ICML*, 2021, pp. 8162–8171.

[43] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2021, pp. 1–36.

[44] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. ICLR*, 2021, pp. 1–22.

[45] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. NeurIPS*, 2022, pp. 1–13.

[46] Y. Xu, Z. Liu, M. Tegmark, and T. Jaakkola, "Poisson flow generative models," in *Proc. NeurIPS*, 2022, pp. 1–33.

[47] Y. Xu, Z. Liu, Y. Tian, S. Tong, M. Tegmark, and T. Jaakkola, "PFGM++:Unlocking the potential of physics-inspired generative models," 2023, *arXiv:2302.04265*.

[48] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Conditional image generation with score-based diffusion models," 2021, *arXiv:2111.13606*.

[49] H. Chung, B. Sim, and J. C. Ye, "Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction," 2021, *arXiv:2112.05146*.

[50] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," 2021, *arXiv:2111.08005*.

[51] C. Saharia et al., "Palette: Image-to-image diffusion models," in *Proc. SIGGRAPH*, 2022, pp. 1–29.

[52] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4713–4726, Apr. 2023.

[53] H. Chung, E. S. Lee, and J. C. Ye, "MR image denoising and super-resolution using regularized reverse diffusion," *IEEE Trans. Med. Imag.*, vol. 42, no. 4, pp. 922–934, Apr. 2023.

[54] R. Ge, Y. He, C. Xia, Y. Chen, D. Zhang, and G. Wang, "JCCS-PFGM: A novel circle-supervision based poisson flow generative model for multiphase CECT progressive low-dose reconstruction with joint condition," 2023, *arXiv:2306.07824*.

[55] I. Goodfellow et al., "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[56] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," 2023, *arXiv:2303.01469*.

[57] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," 2022 *arXiv:2202.00512*.

[58] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, 2011.

[59] (Am. Assoc. Phys. Med., Alexandria, VA, USA). *Low Dose CT Grand Challenge*. 2017. [Online]. Available: https://www.aapm.org/grandchallenge/lowdosect/

[60] H. Almqvist et al., "Initial clinical images from a second-generation prototype silicon-based photon-counting computed tomography system," *Acad. Radiol.*, vol. 31, no. 2, pp. 572–581, 2023.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, 2004.

[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, 2018, pp. 586–595.

[64] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*.

[65] K. Tian, E. Mitchell, H. Yao, C. D. Manning, and C. Finn, "Fine-tuning language models for factuality," 2023, *arXiv:2311.08401*.