

Monte Carlo simulation fused with target distribution modeling via deep reinforcement learning for automatic high-efficiency photon distribution estimation

JIANHUI MA,¹ ZUN PIAO,¹ SHUANG HUANG,¹ XIAOMAN DUAN,¹ GENGGENG QIN,¹ LINGHONG ZHOU,^{1,2} AND YUAN XU^{1,3}

¹School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China

²e-mail: smart@smu.edu.cn

³e-mail: yuanxu@smu.edu.cn

Received 26 October 2020; revised 17 December 2020; accepted 21 December 2020; posted 21 December 2020 (Doc. ID 413486); published 8 February 2021

Particle distribution estimation is an important issue in medical diagnosis. In particular, photon scattering in some medical devices extremely degrades image quality and causes measurement inaccuracy. The Monte Carlo (MC) algorithm is regarded as the most accurate particle estimation approach but is still time-consuming, even with graphic processing unit (GPU) acceleration. The goal of this work is to develop an automatic scatter estimation framework for high-efficiency photon distribution estimation. Specifically, a GPU-based MC simulation initially yields a raw scatter signal with a low photon number to hasten scatter generation. In the proposed method, assume that the scatter signal follows Poisson distribution, where an optimization objective function fused with sparse feature penalty is modeled. Then, an over-relaxation algorithm is deduced mathematically to solve this objective function. For optimizing the parameters in the over-relaxation algorithm, the deep Q-network in the deep reinforcement learning scheme is built to intelligently interact with the over-relaxation algorithm to accurately and rapidly estimate a scatter signal with the large range of photon numbers. Experimental results demonstrated that our proposed framework can achieve superior performance with structural similarity >0.94, peak signal-to-noise ratio >26.55 dB, and relative absolute error <5.62%, and the lowest computation time for one scatter image generation can be within 2 s. © 2021 Chinese Laser Press

<https://doi.org/10.1364/PRJ.413486>

1. INTRODUCTION

Particle distribution estimation is an important issue in medical diagnosis because the principle of that distribution in some crucial imaging devices follows the law of large numbers [1]. However, particle distribution estimation by only trillions of particle sampling is extremely time-consuming. Thus, approximating a target function model on the basis of target distribution feature analysis can substantially hasten particle estimation [2]. Particles fused with a target distribution feature for a few particles have a promising prospect in rapid statistical estimation without the loss of accuracy. As an important branch of particle distribution estimation, photon scattering is commonly involved in X-ray imaging, especially cone beam computed tomography (CBCT), which has been widely used in medical imaging given its high spatial resolution and low radiation dose and for applications such as dental CBCT, image-guided radiotherapy, and extremity CBCT [3]. Scatter occurs when photons

emitted from an X-ray source interact with a physical object and can extremely degrade the reconstructed image quality for clinical diagnosis, thereby leading to pixel-value inaccuracy. Various methods have been proposed for scatter removal, including the scatter kernel [4,5], beam stop array technique [6,7], and primary modulator [8]. Nevertheless, these approaches require additional equipment or increase the irradiation dose to the patient. The Monte Carlo (MC) particle sampling approach has been proved to estimate accurate scatter signals in a cost-efficient manner without the above disadvantages [9–11]. The said technique can precisely simulate the physical process of photon transport to produce all types of scatters composed of Compton, Rayleigh, and photoelectric effect. In reality, one X-ray tube generates approximately 1×10^{13} source photons for imaging one projection under one angle view. Therefore, the scatter simulation of photon transport by only MC particle sampling is extremely time-consuming, and this feature precludes its clinical application [12].

To enhance its computational efficiency, the scatter signal is typically simulated by an MC algorithm with a low photon number, so the generated signal is noisy and fragmentary. To overcome this problem, the noise-contaminated scatter signal is assumed to be governed by the Poisson distribution and sparse feature representation. An efficient denoising algorithm based on Poisson distribution and sparse feature fusion is proposed to smooth the signal along the spatial dimension. Nonetheless, the weight parameter between the Poisson distribution and sparse feature does not follow a one-size-fits-all trend for all cases with different photon numbers. Tuning parameters manually for all situations is a trial-and-error process, which calls for considerable time and effort. Moreover, that scheme hinders a noise removal algorithm from reaching an optimal solution of generating an accurate scatter signal via ultra-low photons.

In the past few years, deep-learning algorithms have achieved remarkable success across many fields, such as computer vision and pattern recognition [13–18]. Deep-learning methods build deep neural networks (DNN) to directly extract local and global features from the dataset, an approach that can avoid handcrafted selection [13]. Motivated by the powerful performance of deep learning, the Google DeepMind group integrated a neural network and reinforcement learning (RL) algorithm to play Atari games with human-level performance [19]. Afterwards, AlphaGo based on deep RL (DRL) defeated human masters in the ancient Chinese game Go and attracted global attention [20]. Shen *et al.* [21] proposed a parameter tuning policy network to adjust pixel-wise parameters in iterative computed tomography (CT) reconstruction and attain comparable or better image quality. DRL was also applied in radiotherapy for optimal dose adaptation and a treatment planning optimization problem [22–25]. These superior performances have demonstrated that the DRL algorithm can achieve a task analogous to human instincts. In DRL, an agent represented by a DNN interacts with the environment to explore all possible consequences of the action for the highest reward feedback. The agent acts according to their observation of the environment, which, in turn, is changed by the action and yields a new observation for the next step. Over many episodes, the agent is supposed to develop an optimal policy to obtain maximum rewards. Essentially, parameter tuning is a dynamic decision-making process, for which DRL is highly suitable. Inspired by the impressive achievements of pioneering work and the rationale of DRL, we propose a framework to realize automatic high-efficiency scatter estimation via the fusion of MC particle sampling, statistical distribution features, and a DRL scheme.

The remainder of this paper is organized as follows. Section 2 describes the automatic scatter estimation framework (ASEF), including its key components, namely, the MC particle sampling algorithm, Poisson distribution and sparse feature-based statistical distribution algorithm, and a DRL scheme. Network interpretability and implementation details are also introduced in the section. Section 3 evaluates the performance of the proposed framework qualitatively and quantitatively. Section 4 discusses and summarizes the strengths and drawbacks.

2. METHODS AND MATERIALS

A. Automatic Scatter Estimation Framework

In this study, we propose an ASEF that integrates an MC particle sampling algorithm (details depicted in Section 2.B), a statistical distribution model fused with sparse feature representation under the Poisson distribution assumption (Section 2.C), and a DRL scheme (Section 2.D). The entire framework is illustrated in Fig. 1. First, on the basis of the X-ray source energy spectrum and system geometry configuration, a graphic processing unit (GPU)-based MC simulation initially yields a raw scatter signal with a low photon number to hasten scatter generation. Second, assuming that the scatter signal follows Poisson distribution, an optimization objective function fused with sparse feature penalty is constructed. Then, an over-relaxation algorithm is deduced mathematically to solve this objective function. For optimizing the parameters in the over-relaxation algorithm, a neural network called the deep Q-network (DQN) is built to intelligently interact with the over-relaxation algorithm for optimal scatter image quality. Specifically, the well trained DQN, which possesses the optimal policy of the highest rewards, takes an action to change the parameters; then the over-relaxation algorithm yields a scatter image based on the adjusted parameters for the next action selection. After several tuning steps, the optimal parameters can be determined when the highest cumulative rewards of a sequence of steps are obtained. The scatter image yielded by the over-relaxation algorithm with the optimal parameters is referred to as optimal quality.

B. Monte Carlo Particle Sampling Algorithm

An in-house MC simulation tool with a polychromatic energy spectrum is utilized in this study for the MC particle sampling of the scatter signal. The energy spectrum describes the probability density of a source photon as a function of its energy. Such an energy spectrum can be specified by using the method developed by Boone *et al.* [26]. According to tabulated data including the attenuation coefficients and the cross sections of Compton scattering, Rayleigh scattering, and the photoelectric effect, each photon transporting through the imaging phantom is simulated to compute the particle distribution in the virtual detector. During the simulation, an indicator is carried by each photon that records if any scattering events have taken place during the transport. For different scattering events of different orders, the photon will be marked by a number as the indicator of the photon. For example, the indicators of all primal photons are zero, and all first order Compton photons are marked one as the indicator. The CT volume of the phantom is utilized to generate different material and density information on the basis of the CT value via threshold-based segmentation. The MC sampling scheme includes several realistic features, e.g., the polychromatic source spectrum, materials and thicknesses of the detector, and beam collimation.

In the MC particle sampling, the scatter angles θ for Rayleigh scattering are sampled from the expression for the differential cross sections as

$$\frac{d\sigma_R}{d(\cos \theta)} = \pi r_0^2 F(E, \theta, Z)^2 (1 + \cos^2 \theta), \quad (1)$$

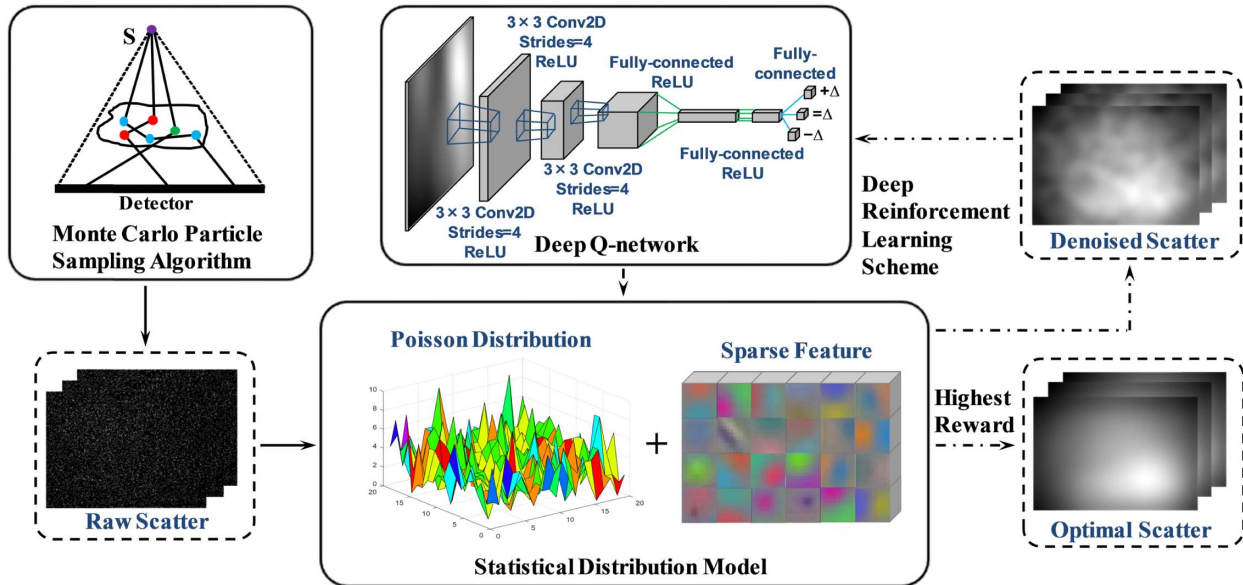


Fig. 1. Automatic scatter estimation framework. The MC algorithm generates raw scatter signals in terms of the X-ray source energy spectrum and system geometry configuration. The DRL scheme (denoted by the dashed black arrow) employs a deep Q-network to interact with the statistical distribution model to yield a satisfactory scatter image.

and the scatter angles θ for Compton scattering can be expressed as

$$\frac{d\sigma_C}{d(\cos \theta)} = \pi r_0^2 S(E, \theta, Z) [P(E, \theta) - P(E, \theta)^2 \sin^2 \theta + P(E, \theta)^3]. \quad (2)$$

Note that the photon energy is reduced after the Compton scattering, and $P(E, \theta)$ is the ratio of the energy E' before scattering and the current energy E :

$$P(E, \theta) = \frac{E'}{E} = \frac{1}{1 + \frac{E}{m_0 c^2} (1 - \cos \theta)}, \quad (3)$$

where r_0 , c , and m_0 separately indicate the electron radius, light speed, and resting mass of one electron in Eqs. (1)–(3). The functions $F(E, \theta, Z)$ and $S(E, \theta, Z)$ are the form factor and the incoherent scattering function. Data for $F(E, \theta, Z)$ and $S(E, \theta, Z)$ are generated from the gCTD package [27], which is a fast MC simulation package for patient-specific CBCT imaging or dose calculation. For shortening computation time, scatter angles θ were not sampled by calculating Eqs. (1) and (2) but were taken from the pre-calculated look-up table. Electron transport is neglected in the simulation, as it cannot reach the detector and will be absorbed in the phantom immediately. Moreover, the Woodcock tracking algorithm [28] is employed to avoid calculating the integral of photon path attenuation voxel by voxel.

C. Scatter Statistical Distribution Model Based on Poisson Distribution and Sparse Feature Representation

It has been proved that the measurement determined by the photon distribution statistics follows a Poisson distribution [29,30]. The scatter signal $\hat{S}(u)$, which is noise-contaminated, is expected to approach true scatter $S(u)$, and thus its proba-

bility density function with expectation $S(u)$ is defined as

$$P[x = \hat{S}(u)] = \frac{S(u)^{\hat{S}(u)}}{\hat{S}(u)!} e^{-S(u)}, \quad (4)$$

where $u = (i, j)$ indicates the detector coordinate, for which i and j denote the horizontal and vertical axes, respectively. Note that measurements $\hat{S}(u)$ are independent of each other, and thus

$$P(x = \hat{S}) = \int \frac{S^{\hat{S}}}{\hat{S}!} e^{-S} d\mathbf{u}. \quad (5)$$

Moreover, Eq. (5) is maximal when $\hat{S} \approx S$, and we minimize $-\log P(x = \hat{S})$ rather than maximizing $P(x = \hat{S})$. That is

$$\arg \min_S \int (S - \hat{S} \log S) d\mathbf{u}. \quad (6)$$

Given the ill-posed nature of Eq. (6), a total-variation (TV) regularization [31] is employed for the smoothness of the solution. Therefore, the formula based on Poisson distribution and sparse feature representation can be defined as follows:

$$S(u) = \arg \min_S \int (S - \hat{S} \log S) d\mathbf{u} + \frac{\beta}{2} \int |\nabla S|^2 d\mathbf{u}, \quad (7)$$

where the first data-fidelity term is designed to ensure Poisson distribution, and the second TV regularization term is formulated for noise suppression. β is a hyper-parameter controlling the relationship between the data-fidelity and regularization terms. As Eq. (7) is convex, the derivative of its optimal solution is equal to zero, that is,

$$\left(1 - \frac{\hat{S}}{S}\right) - \beta \nabla^2 S = 0. \quad (8)$$

Equation (8) can be discretized as

$$\left[1 - \frac{\hat{S}(i,j)}{S(i,j)}\right] - \beta \nabla^2 S(i,j) = 0, \quad (9)$$

where the Laplacian operator $\nabla^2 S(i,j) = \sum S(i,j) - 4S(i,j)$, and $\sum S(i,j)$ stands for $S(i-1,j) + S(i+1,j) + S(i,j-1) + S(i,j+1)$. Then, an over-relaxation iteration algorithm is employed to solve Eq. (7):

$$S^{(k+1)}(i,j) = (1 - \omega)S^{(k)}(i,j) + \frac{\omega}{4} \left[\sum S^{(k)}(i,j) - \frac{1}{\beta} \left(1 - \frac{\hat{S}(i,j)}{S^{(k)}(i,j)}\right) \right], \quad (10)$$

where the superscript k represents the iteration step, and ω is an empirical constant to speed up convergence. Finally, the noise free scatter signal S can be obtained by solving Eq. (10) iteratively.

D. Deep Reinforcement Learning

1. Double Deep Q-Network

Three hyper-parameters appear in Eq. (10): ω , β , and the number of iterations k to be determined. Typically, these variables are set to a constant to obtain smoothed scatter images even though the simulated photon number is different. Tuning parameters manually for all situations is a trial-and-error process, which calls for considerable time and effort. Moreover, this approach restricts the over-relaxation algorithm from exploring a global optimal solution for improved image quality. These quality-related parameters are not one-size-fits-all for all cases with different photon numbers. Consequently, we aim to utilize a DRL scheme to intelligently seek out satisfactory images like human behavior.

The Q -learning algorithm is a widely used approach, for which the action-value function $Q_\pi(s, a)$ is defined as the expectation of the rewards sum of all possible steps from the current state s taking action a under policy π :

$$Q_\pi(s, a) = E(G_t | s_t = s, a_t = a), \quad (11)$$

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{m=1}^{\infty} \gamma^m r_{t+m}, \quad (12)$$

where G_t is the discounted rewards sum starting from step t , r_t is the reward at step t , and $\gamma \in [0, 1]$ is a discount factor. The Q -learning algorithm aims to explore an optimal policy to maximize the action-value function:

$$Q_\pi^*(s, a) = \max_{\pi} Q_\pi(s, a). \quad (13)$$

The optimal action-value function $Q_\pi^*(s, a)$ can be solved iteratively via the Bellman equation:

$$Q_\pi^*(s, a) = r + \gamma \max_{a'} Q_\pi^*(s', a'), \quad (14)$$

where r denotes the reward after adopting action a to the current state s , and s' is the next state following the current state s under action a . Equation (14) entails expensive computation when the state and action involve extensive dimensions. Thus, we adopt a convolutional neural network (CNN) $Q_\pi(s, a; W)$ to approximate the Q function $Q_\pi(s, a)$, such that a quadratic loss function defined as Eq. (15) can be minimized to force $Q_\pi(s, a; W)$ to approach optimal $Q_\pi^*(s, a; W)$:

$$L = E\{[r + \gamma \max_{a'} Q_\pi(s', a'; W) - Q_\pi(s, a; W)]^2\}. \quad (15)$$

To enhance the training stability of $Q_\pi(s, a; W)$, a separate target network $\hat{Q}_\pi(s, a; \hat{W})$, whose architecture is the same as the main network $Q_\pi(s, a; W)$, is constructed following the protocol in Ref. [19]. The weights \hat{W} of $\hat{Q}_\pi(s, a; \hat{W})$ will be fixed when the main network weight W is updated according to the gradient of loss function defined as

$$L = E\{[r + \gamma \max_{a'} \hat{Q}_\pi(s', a'; \hat{W}) - Q_\pi(s, a; W)]^2\}. \quad (16)$$

After several training steps, \hat{W} is updated with W ; that is, $\hat{W} = W$. The approach based on the Q -learning algorithm and DNN is referred to as DQN. The DQN tends to over-estimate the action-value function, so we use double DQN (DDQN) for improved robustness [32]. The DDQN utilizes the main network $Q_\pi(s, a; W)$ to select the action corresponding to the maximum Q -value, which is the output value of the network, namely, $a' = \arg \max_{a'} [Q_\pi(s', a'; W)]$. Then, action a' will be utilized in the target network $\hat{Q}_\pi(s', a'; \hat{W})$ to predict the Q -value. That is,

$$L = E\{[r + \gamma \hat{Q}_\pi\{s', \arg \max_{a'} [Q_\pi(s', a'; W)]; \hat{W}\} - Q_\pi(s, a; W)]^2\}. \quad (17)$$

2. Reward Function

In DRL, the agent aims to take step-by-step action toward the desired situation by obtaining highest rewards. Therefore, the reward is supposed to faithfully evaluate the quality of state s . In this study, state s represents the scatter image per angle. Consequently, we employ structural similarity (SSIM) [33] to construct the reward function. The SSIM is a perceptual metric that can quantitatively measure the similarity between two images x and y , focuses on luminance, contrast, and structure, and is defined as

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (18)$$

where μ_x and μ_y separately denote the average of x and y , σ_x^2 and σ_y^2 represent variance, and σ_{xy} is the covariance of x and y . A higher SSIM indicates greater similarity between two images. Therefore, the reward function consists of the SSIM of the state and ground truth formulated as

$$r_t = \text{sgn}[\text{SSIM}(s_{t+1}, s_{gt}) - \text{SSIM}(s_t, s_{gt})], \quad (19)$$

where s_{gt} denotes the ground truth of the scatter, and the first and second terms separately measure the SSIM of the scatter at step t with the ground truth and the SSIM between the scatter at step $t+1$ and ground truth. r_t will be positive if s_{t+1} is closer to the ground truth compared with the previous state s_t , such that it can inspire the DDQN to improve the generated scatter quality toward the desired image quality. sgn indicates the sign function, which transfers a positive number to 1 and a negative number to -1. As suggested in Ref. [19], the sign function is applied to rescale the reward for the scale limitation of the error derivatives.

3. Training Process of the DDQN

In this study, action a has three possible actions: increase or decrease the parameter by 20% and keep it invariant. Some

researches [21,23] specified the amount of parameter change with different amplitudes. Because the reward function defined in Eq. (19) utilizes the sign function to restrict the scale of the error derivatives, the network cannot differentiate between rewards of different magnitudes. Therefore, we arbitrarily select 20% as the parameter amplitude. We build our network $W = \{W_k, W_\omega, W_\beta\}$ containing three subnetworks to represent three parameters in the over-relaxation algorithm defined as Eq. (10). These subnetworks $\{W_k, W_\omega, W_\beta\}$ possess the same architecture illustrated in Fig. 2. We repeatedly train $\{W_k, W_\omega, W_\beta\}$ for N_{episode} times, and each episode has a series of steps. At step t , one of the three subnetworks $\{W_k, W_\omega, W_\beta\}$ is randomly selected with equal probability. Then, the ϵ -greedy algorithm is adopted to select an action to adjust the parameter. Specifically, we select a random action a_t with probability ϵ ; otherwise, the action corresponding to the highest Q -value is selected, i.e., $a_t = \arg \max_a [Q_\pi(s_t, a; W)]$. Afterward, the parameter is adjusted according to a_t , and the over-relaxation algorithm is solved to generate a new scatter s_{t+1} as the state for the next step. Equation (19) takes s_t and s_{t+1} to calculate the reward r_t . The dataset $\{s_t, a_t, r_t, s_{t+1}\}$ is then stored in the experience replay memory D to mitigate the correlation between the training dataset generated in successive steps. Subsequently, network W is trained by randomly sampling a mini-batch of dataset from D . $\{W_k, W_\omega, W_\beta\}$ will be updated via minimizing loss function in Eq. (17). Finally, update target network weights $\{\hat{W}_k, \hat{W}_\omega, \hat{W}_\beta\}$ with $\{W_k, W_\omega, W_\beta\}$, i.e., let $\{\hat{W}_k, \hat{W}_\omega, \hat{W}_\beta\} = \{W_k, W_\omega, W_\beta\}$ every N_{update} steps. The training process of the DDQN is summarized in Table 1, and the related parameter values are defined in Table 2.

E. DDQN Interpretability

The DDQN can predict an action under the optimal policy. However, the mechanism by which the DDQN takes action in terms of the current state remains unclear. To interpret the DDQN, a gradient-weighted class activation map (Grad-CAM) [34] is utilized to highlight crucial regions in the state for action prediction. The Grad-CAM can yield a localization map using the gradient of output with respect to the last

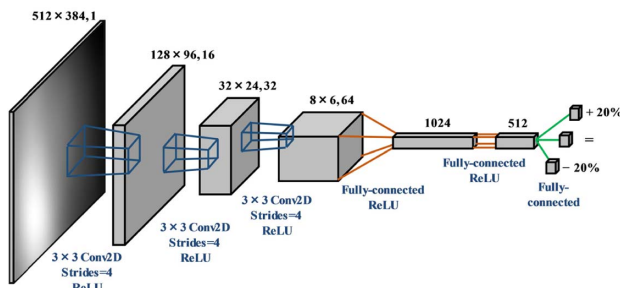


Fig. 2. Network architecture in the DDQN. The network takes a scatter image as input and predicts three possible actions for parameter adjustment. The number at the top denotes the feature map size and channel number, and the operations for each layer are presented at the bottom. For instance, the first hidden layer convolves 16 filters of 3×3 with stride four with the input layer followed by a rectified linear unit (ReLU) activation function, and the output layer is a fully connected linear layer with three outputs.

Table 1. DDQN Training Process

1.	Initialize main network weights W and target network weights \hat{W}
2.	For episode = 1, 2, ..., N_{episode} do
3.	For projection = 1, 2, ..., N_{prj} do
4.	Initialize $\{k_0, \omega_0, \beta_0\}$
5.	Generate s_1 using Eq. (10) with $\{k_0, \omega_0, \beta_0\}$
6.	For $t = 1, 2, \dots, N_{\text{step}}$ do
7.	Randomly select one subnetwork from $\{W_k, W_\omega, W_\beta\}$
8.	With probability ϵ select action a_t randomly
9.	Otherwise choose $a_t = \arg \max_a [Q_\pi(s_t, a; W)]$
10.	Adjust parameters $\{k_t, \omega_t, \beta_t\}$ according to a_t
11.	Generate s_{t+1} using Eq. (10) with $\{k_t, \omega_t, \beta_t\}$
12.	Compute reward r_t using Eq. (19)
13.	Store dataset $\{s_t, a_t, r_t, s_{t+1}\}$ in experience replay D
14.	Randomly sample a mini-batch of dataset from D
15.	Compute the gradient of loss function in Eq. (17)
16.	Update main network weights $W = \{W_k, W_\omega, W_\beta\}$
17.	For every N_{update} steps, let $\hat{W} = W$
18.	End For
19.	End For
20.	End For

Table 2. Parameters in the DDQN Training Phase

Parameters	Values	Descriptions
N_{episode}	100	Number of training episodes
N_{prj}	45	Number of training projections
N_{step}	30	Number of steps for each episode
N_{update}	20	Number of steps for target network weights update
D	2000	Capacity of experience replay memory
ϵ	[0.01, 1]	Probability of random action in ϵ -greedy algorithm
γ	0.6	Discount factor
l_r	0.001	Learning rate of gradient descent for main network
N_{batch}	64	Mini-batch samples for network training

convolution layer to locate regions that are much more important for DDQN decision making. More concretely, Grad-CAM initially calculates the gradient of the Q -value q^a with respect to the feature map A^k of the last convolutional layer whose width and height are separately indexed by i and j , namely, $\frac{\partial q^a}{\partial A_{ij}^k}$. Then, these gradients are backpropagated and global-average-pooled to attain weights α_k^a for every feature map, i.e., $\alpha_k^a = \frac{1}{Z} \sum_i \sum_j \frac{\partial q^a}{\partial A_{ij}^k}$. The Grad-CAM heatmap is the sum of the weighted feature maps and followed by a rectified linear unit (ReLU): $L_{\text{Grad-CAM}}^a = \text{ReLU}(\sum_k \alpha_k^a A^k)$. Finally, the Grad-CAM heatmap is upsampled to the input state size.

F. Implementation Details

In this study, we utilize one Nvidia TITAN Xp GPU and four Intel Core i7 3.6 GHz processors with 40 GB memory to implement the framework using TensorFlow [35]. Equation (10) is a noise removal algorithm in the projection domain. Therefore, the scatter S is a projection whose resolution is 512×384 . Six groups of scatter datasets are present with

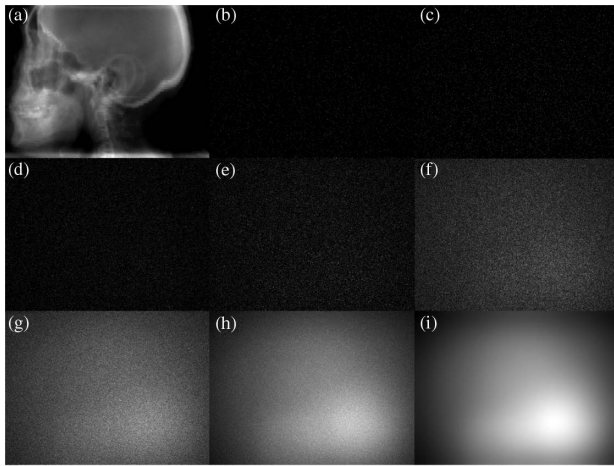


Fig. 3. (a) is the primary projection of the head and neck (H&N) patient; (b)–(i) represent raw scatter projections that are separately calculated by the MC particle sampling algorithm with source photons of 5×10^5 , 1×10^6 , 5×10^6 , 1×10^7 , 1×10^8 , 1×10^9 , 1×10^{10} , and 1×10^{12} for the same projection angle.

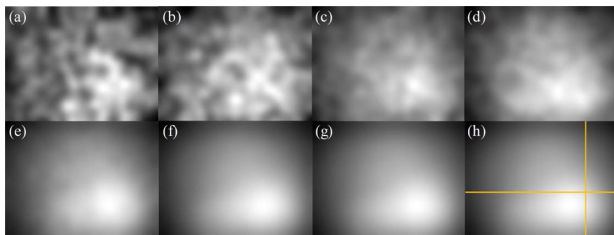


Fig. 4. (a)–(g) are the scatter images of Figs. 3(b)–3(h) smoothed by the over-relaxation smoothing algorithm; (h) corresponds to Fig. 3(i), which is considered a noise free scatter image and utilized as the ground truth.

different photon numbers of 5×10^5 , 1×10^6 , 5×10^6 , 1×10^7 , 1×10^8 , and 1×10^9 . Each dataset contains 90 projections. As shown in Fig. 3(i), a raw scatter signal generated from MC simulation with 1×10^{12} particles is almost noise free and taken as the ground truth. We randomly select 45 of 90 scatter images at 1×10^6 for training. The 45 images whose angles

differ from those of the training cases in the six groups are chosen as the testing cases. Therefore, the total number of testing cases is 270.

3. RESULTS

A. MC Particle Sampling Algorithm

Figures 3(b)–3(i) present the raw scatter signals of the projection produced by the MC particle sampling algorithm with different photon numbers. Figure 3(a) is the primary signal of the projection generated by a typical 100 kVp poly-energetic spectrum with about 100 energy channels. Clearly, Figs. 3(b)–3(h) severely suffer from noise contamination because of the deficiency of source photons (less than 1×10^{12}), whereas Fig. 3(i) shows the MC algorithm that can precisely yield a desirable quality scatter image after the 1×10^{12} photons was simulated. The quantitative evaluation of image similarity and time cost is shown in Tables 3 and 4, respectively. As presented in Table 4, simulating one projection via MC with 1×10^{11} photons costs 6402.60 s. The computation time of the MC particle sampling algorithm exponentially increases with the growth of the photon number. Therefore, directly producing an approved scatter signal through the MC algorithm would be impractical.

B. Over-Relaxation Smoothing Algorithm for Scatter Estimation

Figures 4(a)–4(g) are the corresponding smoothed scatter signals of Figs. 3(b)–3(h) manipulated by an over-relaxation smoothing algorithm with empirical parameters ($k = 700$, $\omega = 0.8$, and $\beta = 1 \times 10^{-11}$), which is denoted as *Empirical* for simplicity. The raw scatter projection with 1×10^{12} photons, as shown in Fig. 4(h), is employed as the ground truth for comparison. The over-relaxation smoothing algorithm can improve scatter image quality among all cases compared to the corresponding raw scatter images. Figures 4(f) and 4(g) are visually similar to Fig. 4(h). Figure 4(e) is slightly rough, whereas Figs. 4(a)–4(d) are severely noisy and fragmentary. The intensity profiles of Figs. 4(a)–4(h) along the row and column indicated by an orange line in Fig. 4(h) are plotted to further validate the performance of the over-relaxation smoothing algorithm. As shown in Figs. 5(a) and 5(b), the intensity profiles of Figs. 4(f) and 4(g) follow exactly the same trend as that in Fig. 4(h). Moreover, the profile lines of Figs. 4(a)–4(e) still

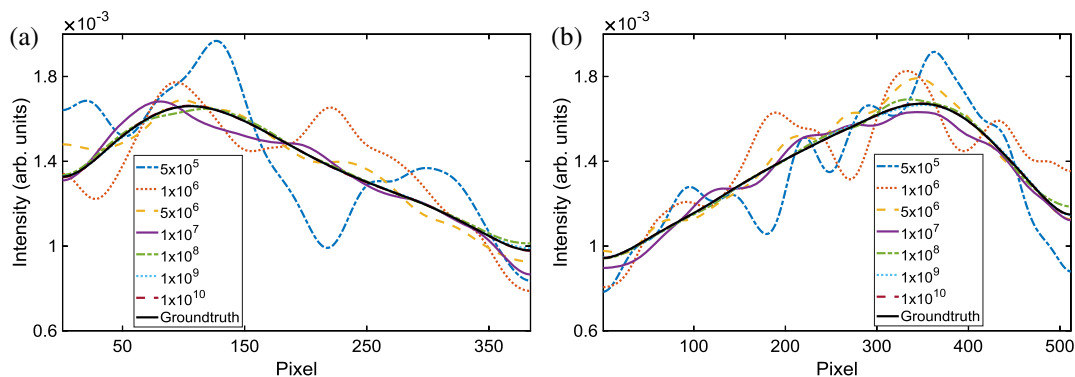


Fig. 5. Intensity profiles of Fig. 4 along the (a) horizontal and (b) vertical directions as denoted by the orange lines in Fig. 4(h).

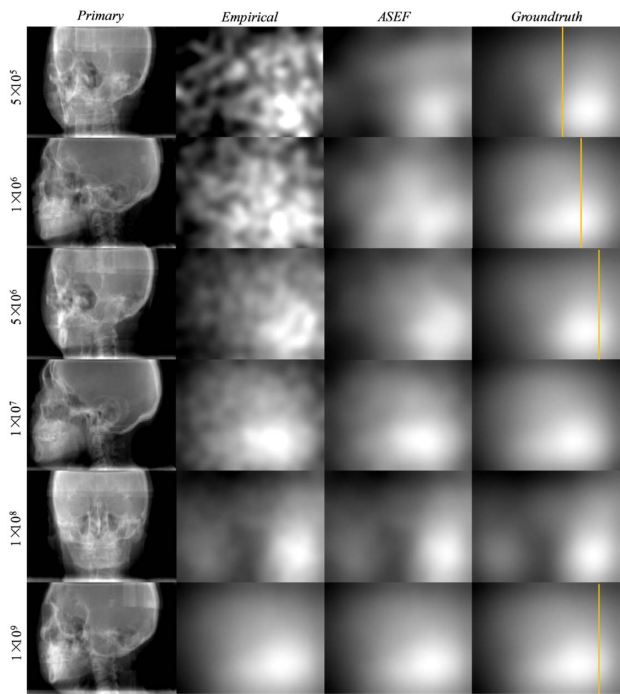


Fig. 6. From top to bottom: six testing results with 5×10^5 , 1×10^6 , 5×10^6 , 1×10^7 , 1×10^8 , and 1×10^9 source photons. From left to right: primary signals, smoothed scatter signals restored by the over-relaxation algorithm with empirical parameters, smoothed scatter signals restored by the proposed framework, and the ground truth.

oscillate. Therefore, the over-relaxation smoothing algorithm performs well when the photon number is no less than 1×10^9 .

C. Scatter Estimation Performance Comparison

To validate its scatter estimation performance, the proposed framework named *ASEF* is compared with *Empirical* among six testing groups with dissimilar photon numbers. As depicted in Fig. 6, *Empirical* alleviates the noise component to some extent but remains inferior compared to *ASEF*. Although both *Empirical* and *ASEF* results degrade as the number of source photons decreases, *ASEF* scatter images are much more consistent in quality improvement. For the most challenging case with 5×10^5 source photons in the first row of Fig. 6, the *Empirical* scatter image is severely distorted after the over-relaxation smoothing algorithm. By contrast, the *ASEF* can substantially remove noise and restore the true scatter signal from raw scatter. The *ASEF* scatter image is visually smoother and similar to the ground truth.

The intensity profiles of Fig. 6 along the vertical direction exhibit the scatter variation trend shown in Fig. 7. Figures 7(a)–7(c) indicate that the *ASEF* can follow a similar trend to the ground truth, whereas the profile of *Empirical* oscillates diversely because of the existence of noise. Figure 7(d) suggests that both *Empirical* and *ASEF* can obtain satisfactory scatter image quality for the 1×10^9 case. However, *ASEF* possesses a powerful capability of exploring optimal solutions for different source photons (Figs. 6 and 7).

For quantitative evaluation, the SSIM, peak signal-to-noise ratio (PSNR), and relative absolute error (RAE) are employed, in which PSNR and RAE are defined as

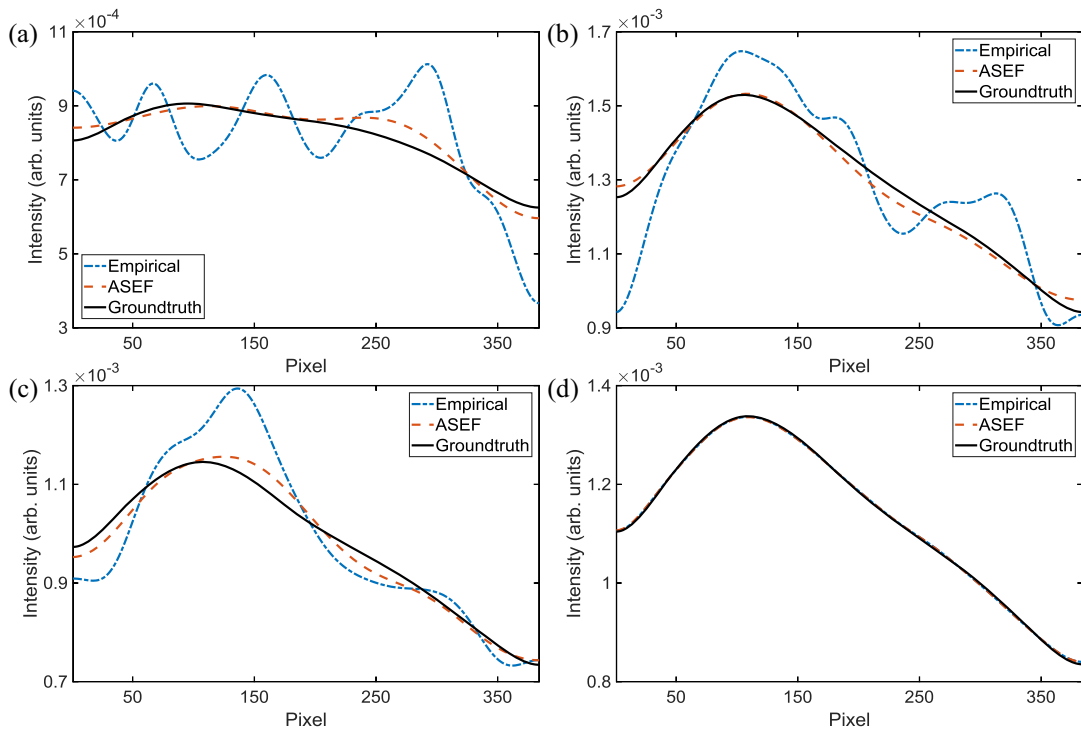


Fig. 7. (a)–(d) Intensity profiles of the first, second, third, and last rows in Fig. 6. The locations of the profiles (a)–(d) are denoted by orange lines at the last column of Fig. 6.

Table 3. SSIM, PSNR, and RAE Statistics (avg. \pm std.) among All Testing Cases^a

Photon Number	SSIM (1 = Best)				PSNR (dB)				RAE (%)			
	<i>Empirical</i>		<i>ASEF</i>		<i>Empirical</i>		<i>ASEF</i>		<i>Empirical</i>		<i>ASEF</i>	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.	avg.	std.	avg.	std.
5×10^5	0.79	4.70×10^{-2}	0.94	2.36×10^{-2}	21.54	0.85	26.55	1.34	12.03	1.27×10^{-2}	5.62	1.27×10^{-2}
1×10^6	0.88	3.73×10^{-2}	0.96	1.67×10^{-2}	23.99	0.72	29.05	1.22	8.52	9.65×10^{-3}	4.22	6.53×10^{-3}
5×10^6	0.97	8.83×10^{-3}	0.99	3.85×10^{-3}	30.26	0.91	33.76	1.03	3.81	4.69×10^{-3}	2.42	3.25×10^{-3}
1×10^7	0.98	4.31×10^{-3}	0.99	2.02×10^{-3}	33.19	0.83	36.05	0.89	2.68	3.14×10^{-3}	1.87	2.35×10^{-3}
1×10^8	0.99	4.96×10^{-4}	0.99	3.97×10^{-4}	43.03	0.82	43.96	0.73	0.84	9.31×10^{-4}	0.74	7.36×10^{-4}
1×10^9	0.99	4.84×10^{-5}	0.99	4.64×10^{-5}	52.97	0.91	53.12	0.89	0.27	3.26×10^{-4}	0.26	3.06×10^{-4}

^aBold indicates the better result for each photon number case under the same metric.

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right), \quad (20)$$

$$\text{RAE} = \sum_{i=1}^m \sum_{j=1}^n \frac{|s(i,j) - s_{gt}(i,j)|}{|s_{gt}(i,j)|}, \quad (21)$$

where $\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n [s(i,j) - s_{gt}(i,j)]^2$. MAX denotes the maximum value of scatter signal s with dimension $m \times n$.

The SSIM, PSNR, and RAE are treated as three metrics to compare the smoothed scatter results using *Empirical* and *ASEF* methods. The result across all testing cases is summarized in Table 3. It is observed that the RAE of *ASEF* is minimal among all different orders of photon numbers, whereas the SSIM and PSNR are maximal compared to the *Empirical* method. Table 3 suggests that for the 5×10^5 source photons case referred to as the most challenging case in this study, the *ASEF* can achieve satisfactory performance with $\text{SSIM} > 0.94$, $\text{PSNR} >$

26.55 dB, and $\text{RAE} < 5.62\%$. Conversely, the SSIM, PSNR, and RAE of *Empirical* are 0.79, 21.54 dB, and 12.03%, respectively. The boxplots of the metric difference between *Empirical* and *ASEF* are plotted in Figs. 8(a)–8(c). We define metric difference as $\text{metric}_{\text{diff}} = \text{metric}_{\text{Empirical}} - \text{metric}_{\text{ASEF}}$. Note that the black line in the box denotes the median of difference statistics for each case. In the figure, the absolute value of the difference statistically becomes larger in the SSIM, PSNR, and RAE metrics with the decrease of the source photons. Figure 8(d) shows the SSIM comparison of *Empirical* and *ASEF* methods, where it is observed that *ASEF* is more robust among different photon number cases.

To demonstrate the efficiency of the proposed framework, we recorded the computation time of the MC particle sampling algorithm and DRL scheme for one scatter image generation using different photon numbers. MC and DRL are speeded up with one Nvidia TITAN Z GPU with 6 GB memory. Table 4 indicates that the computation burden of the MC algorithm exponentially reduces as the number of photons

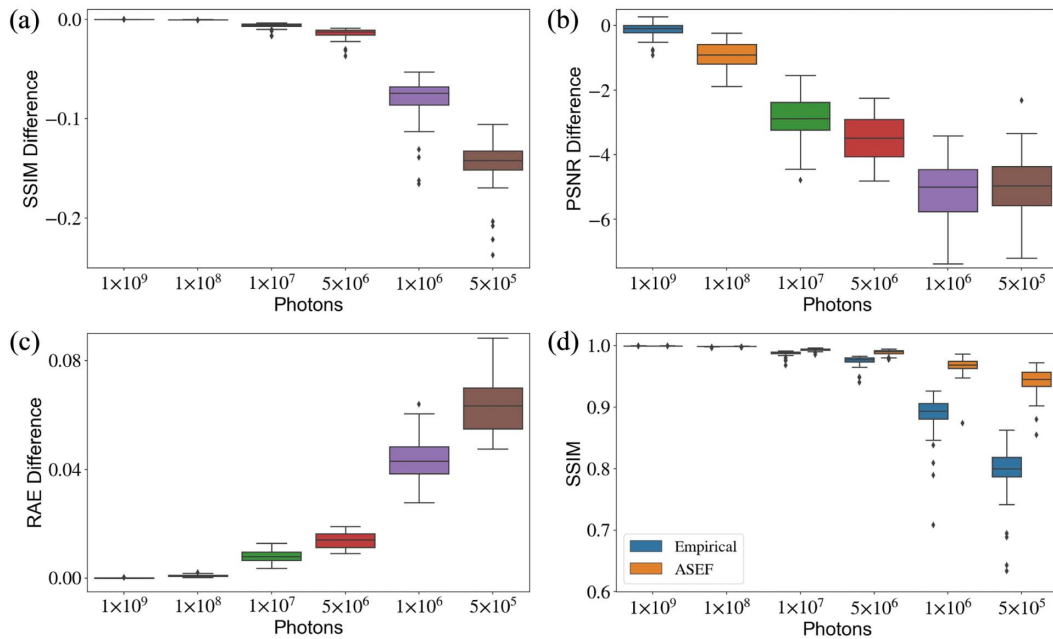


Fig. 8. (a)–(c) indicate boxplots of the metric difference of SSIM, PSNR, and RAE between *Empirical* and *ASEF* for all testing cases. $\text{metric}_{\text{diff}} = \text{metric}_{\text{Empirical}} - \text{metric}_{\text{ASEF}}$, where metric denotes SSIM, PSNR, and RAE, respectively. (d) is the boxplot of the SSIM comparison of *Empirical* and *ASEF*.

Table 4. Computation Time for One Scatter Image of a Prostate Patient across Different Photon Numbers

	Computation Time (s)							
	5×10^5	1×10^6	5×10^6	1×10^7	1×10^8	1×10^9	1×10^{10}	1×10^{11}
MC	0.43	0.45	0.57	0.83	5.94	60.00	633.95	6402.60
DRL	8.98	4.80	1.94	0.98	0.32	0.29	0.29	0.29
Total	9.41	5.25	2.51	1.81	6.26	60.29	634.24	6402.89

decrease. The DRL scheme is an interaction process of the neural network and the noise removal algorithm. The computation cost of the neural network is negligible, so the computation time of DRL is dominated by the noise removal algorithm. Note that the DRL process is comprised of a sequence of steps, where the raw scatter with fewer photons takes DRL more steps to find out the optimal parameters. To improve the computational efficiency, the DRL framework is initialized with empirical parameters. When the source photons are no less than 1×10^9 , DRL just takes one step to reach the optimal solution, so the computation time of DRL for the last three photon numbers in Table 4 is fixed. Briefly, the overall computation time of the proposed *ASEF* can be reduced to ~ 1.81 s without loss of quality, an outcome which is extremely fast in the MC simulation domain.

D. Automatic Scatter Estimation Process

Figure 9 presents the result during the automatic scatter estimation of the *ASEF* for a testing case. Figures 9(a)–9(c) are smoothed scatter images at Steps 1, 7, and 13, respectively. Visually, the image quality of the scatter signal is gradually ameliorated with a sequence of tuning steps. This outcome implies that the DDQN has learned an optimal policy to wisely make decisions for parameter adjustment. The SSIM and RAE results over steps are shown in Figs. 9(d) and 9(e). As expected, the SSIM increases step by step, whereas the RAE decreases over steps. Thus, the *ASEF* can improve scatter image quality intelligently.

E. Grad-CAM Heatmap

The Grad-CAM heatmaps of the three subnetworks $\{W_k, W_\omega, W_\beta\}$ are simultaneously produced when *ASEF* is verified in the testing case. These heatmaps are upsampled to match the original scatter image size for exhibition and nor-

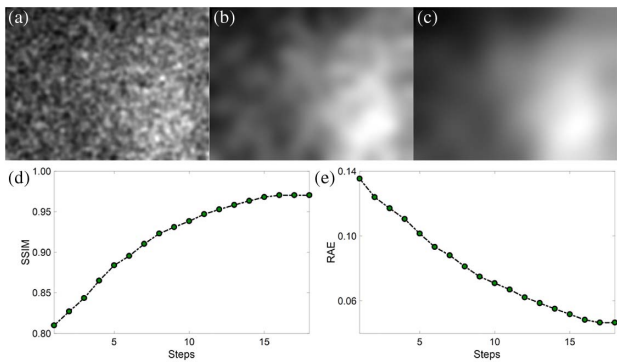


Fig. 9. Automatic scatter estimation process for a testing case. (a)–(c) are smoothed scatter images at Steps 1, 7, and 13, respectively. (d) and (e) separately plot the SSIM and RAE over steps.

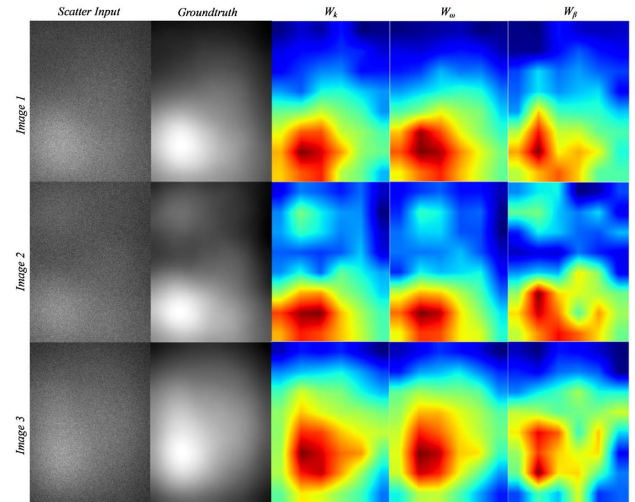


Fig. 10. Different scatter images. From left to right: scatter projection input, the ground truth of the scatter image at the first column, and Grad-CAM heatmaps of three subnetworks $\{W_k, W_\omega, W_\beta\}$.

malized in [0,1], where the prominent region denoted as the red area makes critical contributions to action choice. $\{W_k, W_\omega, W_\beta\}$ take the first column of Fig. 10 to predict an action and produce the Grad-CAM heatmaps for each subnetwork shown in the last three columns. The second column is the corresponding ground truth of the first column for visualization comparison. The rows in Fig. 10 represent different scatter images. Note that the high-intensity region of the scatter image in the first row is at the bottom left, whereas the high-intensity area at the third row is close to the image center. The image at the second row has two highlighted regions, for which the major and minor areas are separately located at the bottom and top left. The salient region of the heatmap is approximately located in a high scatter intensity area. The Grad-CAM heatmaps follow a similar scatter intensity distribution of the input state (first column), thereby implying that all three subnetworks consider higher values as crucial features for action prediction. This inference is reasonable because higher scatter intensities are more likely to suffer from noise contamination compared to lower scatter intensity signals.

F. Generalization Verification

All of the above training and testing cases are the scatter images of a head and neck (H&N) patient; since the proposed framework can improve image quality dynamically in a manner similar to human intelligence, we utilize scatter images of a prostate patient to further verify it. As depicted by the primary signals in Fig. 11, imaging of the pelvis is so large that it exceeds the field

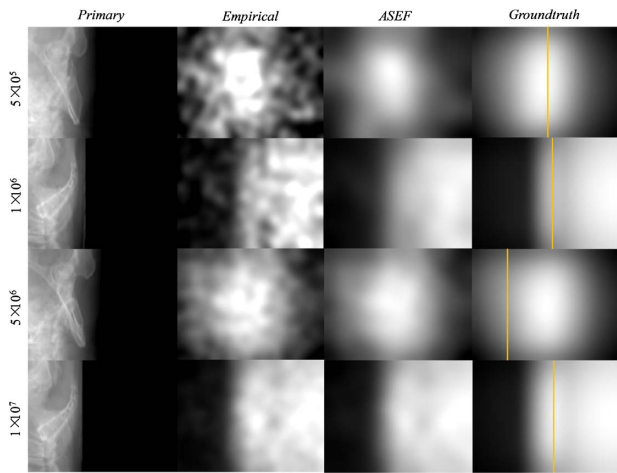


Fig. 11. From top to bottom: four prostate cases with 5×10^5 , 1×10^6 , 5×10^6 , and 1×10^7 source photons. From left to right: primary signals, smoothed scatter signals restored by the over-relaxation algorithm with empirical parameters, smoothed scatter signals restored by the proposed framework, and the ground truth.

of view (FOV) of a CBCT scan. Thus, the projections of the prostate structure are usually acquired by means of half-fan scanning protocol. It is observed that *ASEF* is superior compared with *Empirical* for all prostate cases. Moreover, the intensity profiles of *ASEF* in Fig. 12 are more similar to the corresponding profiles lines of ground truth, in comparison to the tortuous *Empirical* lines.

4. DISCUSSION

As verified in Fig. 3 and Table 4, producing high-quality particle distribution only by the MC algorithm is time-consuming, although the process is speeded up using GPUs. Reducing source photons can considerably improve computational efficiency but lead to noise contamination on the scatter signals. Therefore, an efficient noise removal algorithm based on Poisson distribution and sparse feature fusion is proposed to solve the above issue. The suggested algorithm can effectively make the quality of the scatter signal close to the undistorted signal but degrade gradually for a lower photon number (Figs. 4 and 5). High photon number cases (source photons of over 1×10^9) can be used for experimental study, which requires good quality at the cost of computation time. Meanwhile, ultra-low photon number cases are applied for clinics since the computation time is considered as a crucial factor in clinics. The DRL scheme is employed to boost the performance of the noise removal algorithm for ultra-low photon number cases. In this way, an automatic scatter estimation scheme suitable for the large range of photon numbers is constructed.

In the study, Figs. 6 and 7 depict the proposed framework that can produce satisfactory scatter signals with different source photons. Table 3 summarizes the results of SSIM, PSNR, and RAE across all testing H&N cases and indicates that *ASEF* can achieve good performance with SSIM > 0.94, PSNR > 26.55 dB, and RAE < 5.62%. Figure 8 further proves that our framework was robust across different photon numbers. The computational time of *ASEF* run on the GPU as recorded in Table 4 reveals that our approach is

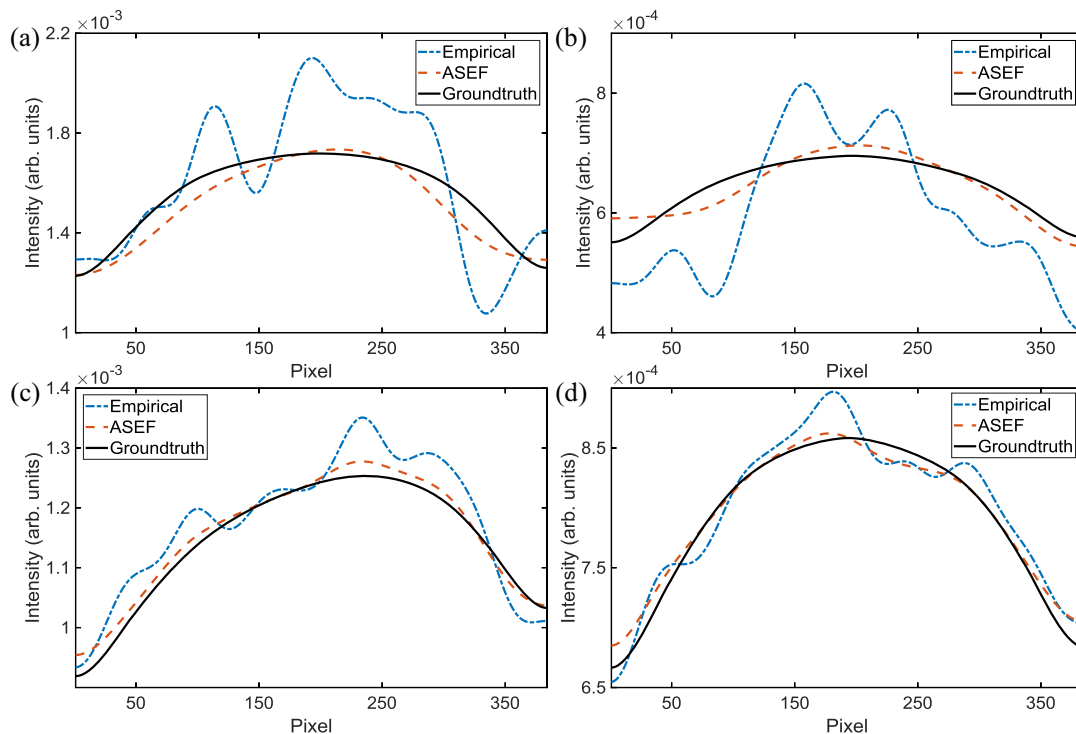


Fig. 12. (a)–(d) Intensity profiles of the four prostate cases presented in Fig. 11. Profile locations are outlined by orange lines in the last column of Fig. 11.

highly efficient with ~ 1.81 s for one scatter projection generation.

The Grad-CAM heatmap was applied to visualize how the DDQN makes action decisions. The region crucial for action prediction is identified in Fig. 10. Obviously, all three subnetworks had conformal intensity distributions similar to the scatter signal, thereby suggesting that the network tends to focus on high scatter intensity areas in view of the fact that the effect of noise on the higher-value region is greater than that in the lower-value counterpart.

We also validated the generalization of the proposed framework on a prostate patient while the DDQN was trained with scatter images of an H&N patient. The scatter signal of prostate patient is much sparser than the H&N scatter signal because the former is acquired by a half-fan scanning protocol. Four prostate scatter projections with 5×10^5 , 1×10^6 , 5×10^6 , and 1×10^7 source photons are validated to exhibit the performance. As shown in Fig. 11, the scatter projections by the proposed framework across all cases with different photon numbers were visually superior, and the intensity profiles in Fig. 12 demonstrated the desired performance as well.

With the rapid development of deep-learning algorithms, it is intuitive and straightforward to build end-to-end CNN for scatter removal. Moreover, several researches [36,37] have demonstrated successful applications of this scheme. However, some weaknesses and limitations exist to some extent. Training the CNN requires numerous paired data for which scatter projections calculated by the MC algorithm are generally considered as output labels for supervised learning. As mentioned before, yielding an excellent scatter image with adequate source photons will take substantial time; not to mention, hundreds of scatter projections are required to avoid over-fitting [38]. In addition, scatter signals are highly dependent on the CT geometry configuration and X-ray source energy spectrum [39–42], and the CNN model trained on a dataset may be inappropriate for other types of datasets. Note that only 45 scatter images are used in our DRL training phase. Because of the DRL training strategy illustrated in Table 1, the training datasets stored in the experience replay pool with the capacity of 2000 are $\{s_t, a_t, r_t, s_{t+1}\}$ rather than 45 images. Furthermore, the training datasets are dynamically updated as the steps change, so there are abundant training data for the neural network model.

5. CONCLUSION

In this study, we proposed an ASEF integrating an MC particle simulation algorithm, statistical distribution model fused with sparse feature representation under Poisson distribution, and a DRL scheme for high-efficiency photon distribution estimation. Experimental results demonstrated that our proposed framework has superior performance for the high-efficiency scatter estimation of large range photon numbers.

Funding. National Key Research and Development Program of China (2016YFA0202003); National Natural Science Foundation of China (61971463); Ministry of Science and Technology Planning Project of Guangdong

(2015B020233002, 2015B020233005); Guangzhou Science and Technology Plan Project (202002030385).

Acknowledgment. We thank Professor Xun Jia and his MC simulation group from the Department of Radiation Oncology at University of Texas Southwestern Medical Center for their assistance with comments and support on this work.

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. P. J. Verheijen, "Statistical distributions in particle technology," in *Particle Technology Course* (Department of Chemical Engineering/ Technical University Delft, 2001).
2. D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN'94)* (1994), pp. 55–60.
3. L. A. Dawson and D. A. Jaffray, "Advances in image-guided radiation therapy," *J. Clin. Oncol.* **25**, 938–946 (2007).
4. L. A. Love and R. A. Kruger, "Scatter estimation for a digital radiographic system using convolution filtering," *Med. Phys.* **14**, 178–185 (1987).
5. J. Star-Lack, M. Sun, A. Kaestner, R. Hassanein, G. Virshup, T. Berkus, and M. Oelhafen, "Efficient scatter correction using asymmetric kernels," *Proc. SPIE* **7258**, 72581Z (2009).
6. R. Ning, X. Tang, and D. Conover, "X-ray scatter correction algorithm for cone beam CT imaging," *Med. Phys.* **31**, 1195–1202 (2004).
7. J. S. Maltz, B. Gangadharan, M. Vidal, A. Paidi, S. Bose, B. A. Faddegon, M. Aubin, O. Morin, J. Pouliot, and Z. Zheng, "Focused beam-stop array for the measurement of scatter in megavoltage portal and cone beam CT imaging," *Med. Phys.* **35**, 2452–2462 (2008).
8. L. Zhu, N. R. Bennett, and R. Fahrig, "Scatter correction method for X-ray CT using primary modulation: theory and preliminary results," *IEEE Trans. Med. Imaging* **25**, 1573–1587 (2006).
9. G. Poludniowski, P. M. Evans, V. N. Hansen, and S. Webb, "An efficient Monte Carlo-based algorithm for scatter correction in keV cone-beam CT," *Phys. Med. Biol.* **54**, 3847–3864 (2009).
10. E. Mainegra-Hing and I. Kawrakow, "Fast Monte Carlo calculation of scatter corrections for CBCT images," *J. Phys. Conf. Ser.* **102**, 012017 (2008).
11. X. Jia, H. Yan, L. Cervino, M. Folkerts, and S. B. Jiang, "A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections," *Med. Phys.* **39**, 7368–7378 (2012).
12. Y. Xu, Y. Chen, Z. Tian, X. Jia, and L. Zhou, "Metropolis Monte Carlo simulation scheme for fast scattered X-ray photon calculation in CT," *Opt. Express* **27**, 1262–1275 (2019).
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
14. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3431–3440.
15. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2015), pp. 234–241.
16. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
17. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4700–4708.
18. W. Zhao, B. Han, Y. Yang, M. Buyyounouski, S. L. Hancock, H. Bagshaw, and L. Xing, "Incorporating imaging information from deep

- neural network layers into image guided radiation therapy (IGRT)," *Radiotherapy Oncol.* **140**, 167–174 (2019).
19. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature* **518**, 529–533 (2015).
 20. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot, "Mastering the game of Go with deep neural networks and tree search," *Nature* **529**, 484–489 (2016).
 21. C. Shen, Y. Gonzalez, L. Chen, S. B. Jiang, and X. Jia, "Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning," *IEEE Trans. Med. Imaging* **37**, 1430–1439 (2018).
 22. H. H. Tseng, Y. Luo, S. Cui, J. T. Chien, R. K. Ten Haken, and I. El Naqa, "Deep reinforcement learning for automated radiation adaptation in lung cancer," *Med. Phys.* **44**, 6690–6705 (2017).
 23. C. Shen, Y. Gonzalez, P. Klages, N. Qin, H. Jung, L. Chen, D. Nguyen, S. B. Jiang, and X. Jia, "Intelligent inverse treatment planning via deep reinforcement learning, a proof-of-principle study in high dose-rate Brachytherapy for cervical cancer," *Phys. Med. Biol.* **64**, 115013 (2019).
 24. C. Shen, L. Chen, Y. Gonzalez, and X. Jia, "Improving efficiency of training a virtual treatment planner network via knowledge-guided deep reinforcement learning for intelligent automatic treatment planning of radiotherapy," arXiv:2007.12591 (2020).
 25. C. Shen, D. Nguyen, L. Chen, Y. Gonzalez, R. McBeth, N. Qin, S. B. Jiang, and X. Jia, "Operating a treatment planning system using a deep-reinforcement learning-based virtual treatment planner for prostate cancer intensity-modulated radiation therapy treatment planning," *Med. Phys.* **47**, 2329–2336 (2020).
 26. J. M. Boone and J. A. Seibert, "Accurate method for computer-generating tungsten anode X-ray spectra from 30 to 140 kV," *Med. Phys.* **24**, 1661–1670 (1997).
 27. X. Jia, H. Yan, X. Gu, and S. B. Jiang, "Fast Monte Carlo simulation for patient-specific CT/CBCT imaging dose calculation," *Phys. Med. Biol.* **57**, 577–590 (2012).
 28. E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth, "Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry," in *Conference on Applications of Computing Methods to Reactor Problems* (1965), pp. 557–579.
 29. E. Jonsson, S. C. Huang, and T. Chan, "Total variation regularization in positron emission tomography," CAM report 9848 (1998).
 30. T. Le, R. Chartrand, and T. J. Asaki, "A variational approach to reconstructing images corrupted by Poisson noise," *J. Math. Imaging Vis.* **27**, 257–263 (2007).
 31. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* **60**, 259–268 (1992).
 32. H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," arXiv:1509.06461 (2015).
 33. 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.* **13**, 600–612 (2004).
 34. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-Cam: visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 618–626.
 35. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "TensorFlow: a system for large-scale machine learning," in *12th Symposium on Operating Systems Design and Implementation* (2016), pp. 265–283.
 36. M. Knaup, J. Kuntz, S. Sawall, and M. Kachelrieß, "Real-time scatter estimation for medical CT using the deep scatter estimation: method and robustness analysis with respect to different anatomies, dose levels, tube voltages, and data truncation," *Med. Phys.* **46**, 238–249 (2019).
 37. D. C. Hansen, G. Landry, F. Kamp, M. Li, C. Belka, K. Parodi, and C. Kurz, "ScatterNet: a convolutional neural network for cone-beam CT intensity correction," *Med. Phys.* **45**, 4916–4926 (2018).
 38. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
 39. B. Ohnesorge, T. Flohr, and K. Klingenberg-Regn, "Efficient object scatter correction algorithm for third and fourth generation CT scanners," *Eur. Radiol.* **9**, 563–569 (1999).
 40. M. Baer and M. Kachelrieß, "Hybrid scatter correction for CT imaging," *Phys. Med. Biol.* **57**, 6849–6867 (2012).
 41. W. Zhao, S. Brunner, K. Niu, S. Schafer, K. Royalty, and G. H. Chen, "Patient-specific scatter correction for flat-panel detector-based cone-beam CT imaging," *Phys. Med. Biol.* **60**, 1339–1365 (2015).
 42. M. Sun and J. Star-Lack, "Improved scatter correction using adaptive scatter kernel superposition," *Phys. Med. Biol.* **55**, 6695–6720 (2010).