

Training End-to-End Unrolled Iterative Neural Networks for SPECT Image Reconstruction

Zongyu Li¹, Graduate Student Member, IEEE, Yuni K. Dewaraja, Member, IEEE,
and Jeffrey A. Fessler¹, Fellow, IEEE

Abstract—Training end-to-end unrolled iterative neural networks for single-photon emission computerized tomography (SPECT) image reconstruction requires a memory-efficient forward–backward projector for efficient backpropagation. This article describes an open-source, high-performance Julia implementation of a SPECT forward–backward projector that supports memory-efficient backpropagation with an exact adjoint. Our Julia projector uses only ~5% of the memory of an existing MATLAB-based projector. We compare unrolling a CNN-regularized expectation–maximization (EM) algorithm with end-to-end training using our Julia projector with other training methods, such as gradient truncation (ignoring gradients involving the projector) and sequential training, using XCAT phantoms and virtual patient (VP) phantoms generated from SIMIND Monte Carlo (MC) simulations. Simulation results with two different radionuclides (⁹⁰Y and ¹⁷⁷Lu) show that: 1) for ¹⁷⁷Lu XCAT phantoms and 2) ⁹⁰Y VP phantoms, training unrolled EM algorithm in an end-to-end fashion with our Julia projector yields the best reconstruction quality compared to other training methods and ordered-subset EM (OSEM), both qualitatively and quantitatively. For VP phantoms with ¹⁷⁷Lu radionuclide, the reconstructed images using end-to-end training are in higher quality than using sequential training and OSEM, but are comparable with using gradient truncation. We also find there exists a tradeoff between computational cost and reconstruction accuracy for different training methods. End-to-end training has the highest accuracy because the correct gradient is used in backpropagation; sequential training yields worse reconstruction accuracy, but is significantly faster and uses much less memory.

Index Terms—Backpropagatable forward–backward projector, end-to-end learning, quantitative single-photon emission computerized tomography (SPECT), regularized model-based image reconstruction.

Manuscript received 27 September 2022; revised 24 November 2022 and 19 January 2023; accepted 27 January 2023. Date of publication 30 January 2023; date of current version 3 April 2023. This work was supported in part by NIH under Grant R01 EB022075 and Grant R01 CA240706. (Corresponding author: Zongyu Li.)

This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of Michigan Institutional Review Board (IRB).

Zongyu Li and Jeffrey A. Fessler are with the Department of Electrical Engineering and Computer Science, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 USA (e-mail: zonyul@umich.edu; fessler@umich.edu).

Yuni K. Dewaraja is with the Department of Radiology, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 USA (e-mail: yuni@med.umich.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TRPMS.2023.3240934>.

Digital Object Identifier 10.1109/TRPMS.2023.3240934

I. INTRODUCTION

SINGLE-PHOTON emission computerized tomography (SPECT) is a nuclear medicine technique that images spatial distributions of radioisotopes and plays a pivotal role in clinical diagnosis, and in estimating radiation-absorbed doses in nuclear medicine therapies [1], [2]. For example, quantitative SPECT imaging with Lutetium-177 (¹⁷⁷Lu) in targeted radionuclide therapy (such as ¹⁷⁷Lu DOTATATE) is important in determining dose–response relationships in tumors and holds great potential for dosimetry-based individualized treatment. Additionally, quantitative Yttrium-90 (⁹⁰Y) bremsstrahlung SPECT imaging is valuable for safety assessment and absorbed dose verification after ⁹⁰Y radioembolization in liver malignancies. However, SPECT imaging suffers from noise and limited spatial resolution due to the collimator response; the resulting reconstruction problem is hence ill-posed and challenging to solve.

Numerous reconstruction algorithms have been proposed for SPECT reconstruction, of which the most popular ones are model-based image reconstruction algorithms, such as maximum-likelihood expectation–maximization (MLEM) [3] and ordered-subset EM (OSEM) [4]. These methods first construct a mathematical model for the SPECT imaging system, then maximize the (log-)likelihood for a Poisson noise model. Although MLEM and OSEM have achieved great success in clinical use, they have a tradeoff between recovery and noise. To address that tradeoff, researchers have proposed alternatives such as regularization-based (or maximum a posteriori in Bayesian setting) reconstruction methods [5], [6], [7]. For example, Panin et al. [5] proposed total variation (TV) regularization for SPECT reconstruction. However, TV regularization may lead to “blocky” images and oversmoothing the edges. One way to overcome blurring edges is to incorporate anatomical boundary side information from CT images [8], but that method requires accurate organ segmentation. Chun et al. [9] used nonlocal means (NLM) filters that exploit the self-similarity of patches in images for regularization, yet that method is computationally expensive and hence less practical. In general, choosing an appropriate regularizer can be challenging; moreover, these traditional regularized algorithms may lack generalizability to images that do not follow assumptions made by the prior.

With the recent success of deep learning (DL) and especially convolutional neural networks (CNNs), DL methods have been reported to outperform conventional algorithms

in many medical imaging applications, such as MRI [10], [11], [12], CT [13], [14], and PET reconstruction [15], [16], [17]. However, fewer DL approaches to SPECT reconstruction appear in the literature. Shao et al. [18] proposed ‘‘SPECTnet’’ with a two-step training strategy that learns the transformation from projection space to image space as an alternative to the traditional OSEM algorithm. Shao et al. [19] also proposed a DL method that can directly reconstruct the activity image from the SPECT projection data, even with reduced view angles. Mostafapour et al. [20] trained a neural network that maps nonattenuation-corrected SPECT images to those corrected by CT images as a post-processing procedure to enhance the reconstructed image quality.

Though promising results were reported with these methods, most of them worked in 2-D whereas 3-D is used in practice [18], [19]. Furthermore, there has yet to be an investigation of end-to-end training of CNN regularizers that are embedded in unrolled SPECT iterative statistical algorithms such as CNN-regularized expectation–maximization (EM). End-to-end training is popular in machine learning and other medical imaging fields such as MRI image reconstruction [21] and is reported to meet data-driven regularization for inverse problems [22]. But for SPECT image reconstruction, end-to-end training is nontrivial to implement due to its complicated system matrix. Alternative training methods have been proposed, such as sequential training [23], [24], [25], [26] and gradient truncation [27]; these methods were shown to be effective, though they could yield suboptimal reconstruction results due to approximations to the training loss gradient. Another approach is to construct a neural network that also models the SPECT system matrix, like in ‘‘SPECTnet’’ [18], but this approach lacks interpretability compared to algorithms like unrolled CNN-regularized EM, i.e., if one sets the regularization parameter to zero, then the latter becomes identical to the traditional EM.

As an end-to-end training approach has not yet been investigated for SPECT image reconstruction, this article first describes a SPECT forward–backward projector written in the open-source and high-performance Julia language that enables efficient auto-differentiation. Then, we compare the end-to-end training approach with other non-end-to-end training methods.

The structure of this article is as follows. Section II describes the implementation of our Julia projector and discusses end-to-end training and other training methods for the unrolled EM algorithm. Section III compares the accuracy, speed, and memory use of our Julia projector with Monte Carlo (MC) and a MATLAB-based projector, and then compares reconstructed images with end-to-end training versus sequential training and gradient truncation on different datasets (XCAT and virtual patient (VP) phantoms), using qualitative and quantitative evaluation metrics. Sections IV and V conclude this article and discuss future works.

Notation: Bold upper/lower case letters (e.g., \mathbf{A} , \mathbf{x} , \mathbf{y} , and \mathbf{b}) denote matrices and column vectors, respectively. Italics (e.g., μ , y , and b) denote scalars. y_i and b_i denote the i th element in vector \mathbf{y} and \mathbf{b} , respectively. \mathbb{R}^N and \mathbb{C}^N denote the N -dimensional real/complex normed vector

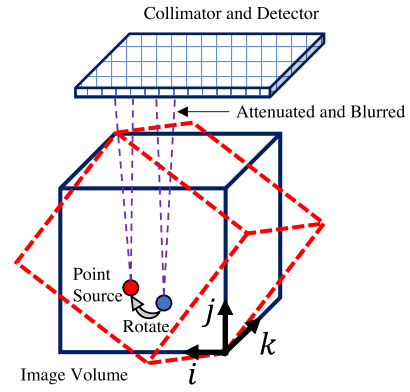


Fig. 1. SPECT imaging model for parallel-beam collimators, with attenuation and depth-dependent collimator point spread response.

space, respectively. $(\cdot)^*$ denotes the complex conjugate and $(\cdot)'$ denotes the Hermitian transpose.

II. METHODS

This section summarizes the Julia SPECT projector, a DL-based image reconstruction method, as well as the dataset used in experiments and other experiment setups.

A. Implementation of Julia SPECT Projector

Our Julia implementation of SPECT projector is based on [28], modeling parallel-beam collimator geometries. Our projector also accounts for attenuation and depth-dependent collimator response. We did not model the scattering events like Compton scatter and coherent scatter of high-energy gamma rays within the object. Fig. 1 illustrates the SPECT imaging system modeled in this article.

For the forward projector, at each rotation angle, we first rotate the 3-D image matrix $\mathbf{x} \in \mathbb{R}^{n_x \times n_y \times n_z}$ according to the third dimension by its projection angle θ_l (typically $2\pi(l-1)/n_{\text{view}}$); l denotes the view index, which ranges from 1 to n_{view} and n_{view} denotes the total number of projection views. We implemented and compared (results shown in Section III) both bilinear interpolation and 3-pass 1-D linear interpolation [29] with zero padding boundary condition for image rotation. For attenuation correction, we first rotated the 3-D attenuation map $\boldsymbol{\mu} \in \mathbb{R}^{n_x \times n_y \times n_z}$ (obtained from transmission tomography) also by θ_l , yielding a rotated 3-D array $\tilde{\boldsymbol{\mu}}(i, j, k; l)$, where i, j, k denotes the 3-D voxel coordinate. Assuming n_y is the index corresponding to the closest plane of \mathbf{x} to the detector, then we model the accumulated attenuation factor $\bar{\mu}$ for each view angle as

$$\bar{\mu}(i, j, k; l) = e^{-\Delta y \left(\frac{1}{2} \tilde{\mu}(i, j, k; l) + \sum_{s=j+1}^{n_y} \tilde{\mu}(i, s, k; l) \right)} \quad (1)$$

where Δy denotes the voxel size for the (first and) second coordinates. Next, for each y slice (an (x, z) plane for a given j index) of the rotated and attenuated image, we convolve with the appropriate slice of the depth-dependent point spread function $\mathbf{p} \in \mathbb{R}^{p_x \times p_z \times n_y \times n_{\text{view}}}$ using a 2-D fast Fourier transform (FFT). Here, we use replicate padding for both the i and k coordinates. The view-dependent PSF accommodates noncircular orbits. Finally, the forward-projection operation simply sums

Algorithm 1: SPECT Forward Projector

Input: 3D image $\mathbf{x} \in \mathbb{R}^{n_x \times n_y \times n_z}$,
 3D attenuation map $\boldsymbol{\mu} \in \mathbb{R}^{n_x \times n_y \times n_z}$,
 4D point spread function $\mathbf{p} \in \mathbb{R}^{p_x \times p_z \times n_y \times n_{\text{view}}}$,
 voxel size Δy .
Initialize: $\mathbf{v} \in \mathbb{R}^{n_x \times n_z \times n_{\text{view}}}$ as all zeros.
for $l = 1, \dots, n_{\text{view}}$ **do**
 $\tilde{\mathbf{x}} \leftarrow$ rotate \mathbf{x} by θ_l
 $\tilde{\boldsymbol{\mu}} \leftarrow$ rotate $\boldsymbol{\mu}$ by θ_l
 for $j = 1, \dots, n_y$ **do**
 $\tilde{\boldsymbol{\mu}} \leftarrow$ calculate by (1) using $\tilde{\boldsymbol{\mu}}$
 $\tilde{x}(i, j, k) * = \tilde{\boldsymbol{\mu}}(i, j, k)$
 $v(i, k, l) += \tilde{x}(i, j, k) \otimes p(i, k, j, l)$
 end
end
Output: projection views $\mathbf{v} \in \mathbb{R}^{n_x \times n_z \times n_{\text{view}}}$

Algorithm 2: SPECT Backward Projector

Input: Array of 2D projection views $\mathbf{v} \in \mathbb{R}^{n_x \times n_z \times n_{\text{view}}}$,
 3D attenuation map $\boldsymbol{\mu} \in \mathbb{R}^{n_x \times n_y \times n_z}$,
 4D point spread function $\mathbf{p} \in \mathbb{R}^{p_x \times p_z \times n_y \times n_{\text{view}}}$,
 voxel size Δy .
Initialize: $\mathbf{x} \in \mathbb{R}^{n_x \times n_y \times n_z}$ as all zeros.
for $l = 1, \dots, n_{\text{view}}$ **do**
 $\tilde{\boldsymbol{\mu}} \leftarrow$ rotate $\boldsymbol{\mu}$ by θ_l
 for $j = 1, \dots, n_y$ **do**
 $\tilde{\boldsymbol{\mu}} \leftarrow$ calculate by (1) using $\tilde{\boldsymbol{\mu}}$
 $\tilde{v}(i, k, l) \leftarrow$ adjoint of $v(i, k, l) \otimes p(i, k, j, l)$
 $\tilde{x}(i, j, k) \leftarrow \tilde{v}(i, k, l) \cdot \tilde{\boldsymbol{\mu}}(i, j, k; l)$
 end
 $\mathbf{x} +=$ adjoint rotate $\tilde{\mathbf{x}}$ by θ_l
end
Output: $\mathbf{x} \in \mathbb{R}^{n_x \times n_y \times n_z}$

the rotated, blurred, and attenuated activity image \mathbf{x} along the second coordinate j . Algorithm 1 summarizes the forward projector, where \otimes denotes a 2-D convolution operation.

All of these steps are linear, so hereafter, we use \mathbf{A} to denote the forward projector, though it is not stored explicitly as a matrix. As each step is linear, each step has an adjoint operation. So the backward projector \mathbf{A}' is the adjoint of \mathbf{A} that satisfies

$$\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{A}'\mathbf{y} \rangle \quad \forall \mathbf{x}, \mathbf{y}. \quad (2)$$

The exact adjoint of (discrete) image rotation is not simply a discrete rotation of the image by $-\theta_l$. Instead, one should also consider the adjoint of linear interpolation. For the adjoint of convolution, we assume the point spread function is symmetric along coordinates i and k so that the adjoint convolution operator is just the forward convolution operator along with the adjoint of replicate padding. Algorithm 2 summarizes the SPECT backward projector.

To accelerate the for-loop process, we used multithreading to enable projecting or backprojecting multiple angles at the same time. To reduce memory use, we preallocated necessary arrays and used fully in-place operations inside the for-loop in

forward and backward projection. To further accelerate auto-differentiation, we customized the chain rule to use the linear operator \mathbf{A} or \mathbf{A}' as the Jacobian when calling $\mathbf{A}\mathbf{x}$ or $\mathbf{A}'\mathbf{y}$ during backpropagation. We implemented and tested our projector in Julia v1.6; we also implemented a GPU version in Julia (using CUDA.jl) that runs efficiently on a GPU by eliminating explicit scalar indexing. For completeness, we also provide a PyTorch version but without multithreading support, in-place operations nor the exact adjoint of image rotation.

B. Unrolled CNN-Regularized EM Algorithm

Model-based image reconstruction algorithms seek to estimate image $\mathbf{x} \in \mathbb{R}^N$ from noisy measurements $\mathbf{y} \in \mathbb{R}^M$ with imaging model $\mathbf{A} \in \mathbb{R}^{M \times N}$. In SPECT reconstruction, the measurements \mathbf{y} are often modeled by

$$\mathbf{y} \sim \text{Poisson}(\mathbf{A}\mathbf{x} + \bar{\mathbf{r}}) \quad (3)$$

where $\bar{\mathbf{r}} \in \mathbb{R}^M$ denotes the vector of means of background events such as scatters. Combining regularization with the Poisson negative log likelihood yields the following optimization problem:

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\mathbf{x} \geq \mathbf{0}} f(\mathbf{x}) + R(\mathbf{x}), \\ f(\mathbf{x}) &\triangleq \mathbf{1}'(\mathbf{A}\mathbf{x} + \bar{\mathbf{r}}) - \mathbf{y}' \log(\mathbf{A}\mathbf{x} + \bar{\mathbf{r}}) \end{aligned} \quad (4)$$

where $f(\mathbf{x})$ is the data fidelity term and $R(\mathbf{x})$ denotes the regularizer. For DL regularizers, we follow [23] and formulate $R(\mathbf{x})$ as

$$R(\mathbf{x}) \triangleq \frac{\beta}{2} \|\mathbf{x} - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2 \quad (5)$$

where β denotes the regularization parameter; and $\mathbf{g}_{\boldsymbol{\theta}}$ denotes a neural network with parameter $\boldsymbol{\theta}$ that is trained to learn to enhance the image quality.

Based on (4), a natural reconstruction approach is to apply variable splitting with $\mathbf{u} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})$ and then alternatively update the images \mathbf{x} and \mathbf{u} as follows:

$$\begin{aligned} \mathbf{u}_{k+1} &= \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_k), \\ \mathbf{x}_{k+1} &= \arg \min_{\mathbf{x} \geq \mathbf{0}} f(\mathbf{x}) + \frac{\beta}{2} \|\mathbf{x} - \mathbf{u}_{k+1}\|_2^2 \end{aligned} \quad (6)$$

where subscript k denotes the iteration number. To minimize (6), we used the EM-surrogate from [30] as summarized in [23], leading to the following vector update:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \frac{1}{2\beta} \left(-\mathbf{d}(\beta) + \sqrt{\mathbf{d}(\beta)^2 + 4\beta \mathbf{x}_k \odot \mathbf{e}(\mathbf{x}_k)} \right) \quad (7) \\ \mathbf{d}(\beta) &\triangleq \mathbf{A}'\mathbf{1} - \beta \mathbf{u}_k, \quad \mathbf{e}(\mathbf{x}_k) \triangleq \mathbf{A}'(\mathbf{y} \oslash (\mathbf{A}\mathbf{x}_k + \bar{\mathbf{r}})) \end{aligned} \quad (8)$$

where \odot and \oslash denote element-wise multiplication and division, respectively. To compute \mathbf{x}_{k+1} , one must substitute $\hat{\mathbf{x}}_k$ back into $\mathbf{e}(\cdot)$ in (8), and repeat. Hereafter, we refer to (6) as one outer iteration and (7) as one inner EM iteration. Algorithm 3 summarizes the CNN-regularized EM algorithm.

To train $\mathbf{g}_{\boldsymbol{\theta}}$, the most direct way is to unroll Algorithm 3 and train end-to-end with an appropriate target; this supervised approach requires backpropagating through the SPECT system model, which is not trivial to implement with previous

Algorithm 3: SPECT CNN-Regularized EM Algorithm

Input: 3D projection measurements y ,
 3D background measurements \bar{r} ,
 system model A , initial guess x_0 ,
 deep neural network g_θ ,
 outer iterations K .
for $k = 0, \dots, K - 1$ **do**
 $u_{k+1} = g_\theta(x_k)$
 $x_{k+1} \leftarrow$ repeat (7) until convergence tolerance or
 maximum # of inner iterations is reached
end
Output: x_K

SPECT projection tools due to the memory issues. Non-end-to-end training methods, e.g., sequential training [23], first train u_k by the target and then plug into (7) at each iteration. This method must use nonshared weights for the neural network per each iteration. Another method is gradient truncation [27] that ignores the gradient involving the system matrix A and its adjoint A' during backpropagation. Both of these training methods, though reported to be effective, may be suboptimal because they approximate the overall training loss gradients.

C. Phantom Dataset and Simulation Setup

We used simulated XCAT phantoms [31] and VP phantoms for experiment results presented in Section III. Each XCAT phantom was simulated to approximately follow the activity distributions observed when imaging patients after ^{177}Lu DOTATATE therapy. We set the image size to $128 \times 128 \times 80$ with voxel size $4.8 \times 4.8 \times 4.8 \text{ mm}^3$. Tumors of various shapes and sizes (5–100 mL) were located in the liver as is typical for patients undergoing this therapy.

For VP phantoms, we consider two radionuclides: ^{177}Lu and ^{90}Y . For ^{177}Lu phantoms, the true images were from PET/CT scans of patients who underwent diagnostic ^{68}Ga DOTATATE PET/CT imaging (Siemens Biograph mCT) to determine eligibility for ^{177}Lu DOTATATE therapy. The ^{68}Ga DOTATATE distribution in patients is expected to be similar to ^{177}Lu and hence can provide a reasonable approximation to the activity distribution of ^{177}Lu in patients for DL training purposes but at a higher resolution. The PET images had size $200 \times 200 \times 577$ and voxel size $4.073 \times 4.073 \times 2 \text{ mm}^3$ and were obtained from our Siemens mCT (resolution is 5–6 mm FWHM [32]) and reconstructed using the standard clinic protocol: 3-D OSEM with three iterations, 21 subsets, including resolution recovery, time of flight, and a 5-mm (FWHM) Gaussian post-reconstruction filter. The density maps were also generated using the experimentally derived CT-to-density calibration curve.

For ^{90}Y phantoms, the true activity images were reconstructed (using a previously implemented 3-D OSEM reconstruction with CNN-based scatter estimation [33]) from ^{90}Y SPECT/CT scans of patients who underwent ^{90}Y microsphere radioembolization in our clinic.

In total, we simulated 4 XCAT phantoms, 8 ^{177}Lu and 8 ^{90}Y VP phantoms. We repeated all of our experiments three times with different noise realizations. All image data have the University of Michigan Institutional Review Board (IRB) approval for retrospective analysis. For all simulated phantoms, we selected the center slices covering the lung, liver, and kidney corresponding to SPECT axial field of view (FOV) (39 cm).

Then, we ran the SIMIND MC program [34] to generate the radial position of SPECT camera for 128 view angles. The SIMIND model parameters for ^{177}Lu were based on ^{177}Lu DOTATATE patient imaging in our clinic (Siemens Intevo with medium energy collimators, a 5/8" crystal, a 20% photopeak window at 208 keV, and two adjacent 10% scatter windows) [35]. For ^{90}Y , a high-energy collimator, 5/8" crystal, and a 105–195-keV acquisition energy window was modeled as in our clinical protocol for ^{90}Y bremsstrahlung imaging. Next, we approximated the point spread function for ^{177}Lu and ^{90}Y by simulating the point source at six different distances (20, 50, 100, 150, 200, and 250 mm) and then fitting a 2-D Gaussian distribution at each distance. The camera orbit was assumed to be noncircular (auto-contouring mode in clinical systems) with the minimum distance between the phantom surface and detector set at 1 cm.

III. EXPERIMENT RESULTS

A. Comparison of Projectors

We used an XCAT phantom to evaluate the accuracy and memory efficiency of our Julia projector.

1) *Accuracy:* We first compared primary (no scatter events included) projection images and profiles generated by our Julia projector with those from MC simulation and the MATLAB projector. For the results of MC, we ran two SIMIND simulations for 1 billion histories using ^{177}Lu and ^{90}Y as radionuclide source, respectively. Each simulation took about 10 h using a 3.2-GHz 16-Core Intel Xeon W CPU on MacOS. The MATLAB projector was originally implemented and compiled in C99 and then wrapped by a MATLAB MEX file as a part of the Michigan Image Reconstruction Toolbox (MIRT) [36]. The physics modeling of the MATLAB projector was the same as our Julia projector except that it only implemented 3-pass 1-D linear interpolation for image rotation. Unlike the memory-efficient Julia version, the MATLAB version prerotates the patient attenuation map for all projection views. This strategy saves time during EM iterations for a single patient, but uses considerable memory and scales poorly for DL training approaches involving multiple patient datasets.

Fig. 2 compared the primary projections generated by different methods without adding Poisson noise. Visualizations of image slices and line profiles illustrate that our Julia projector (with rotation based on 3-pass 1-D interpolation) is almost identical to the MATLAB projector, while both give a reasonably good approximation to the MC. Using MC as reference, the normalized root-mean-square error (NRMSE) of Julia1-D/MATLAB/Julia2-D projectors were 7.9%/7.9%/7.6% for ^{177}Lu , respectively; while the NRMSE were 8.2%/8.2%/7.9%

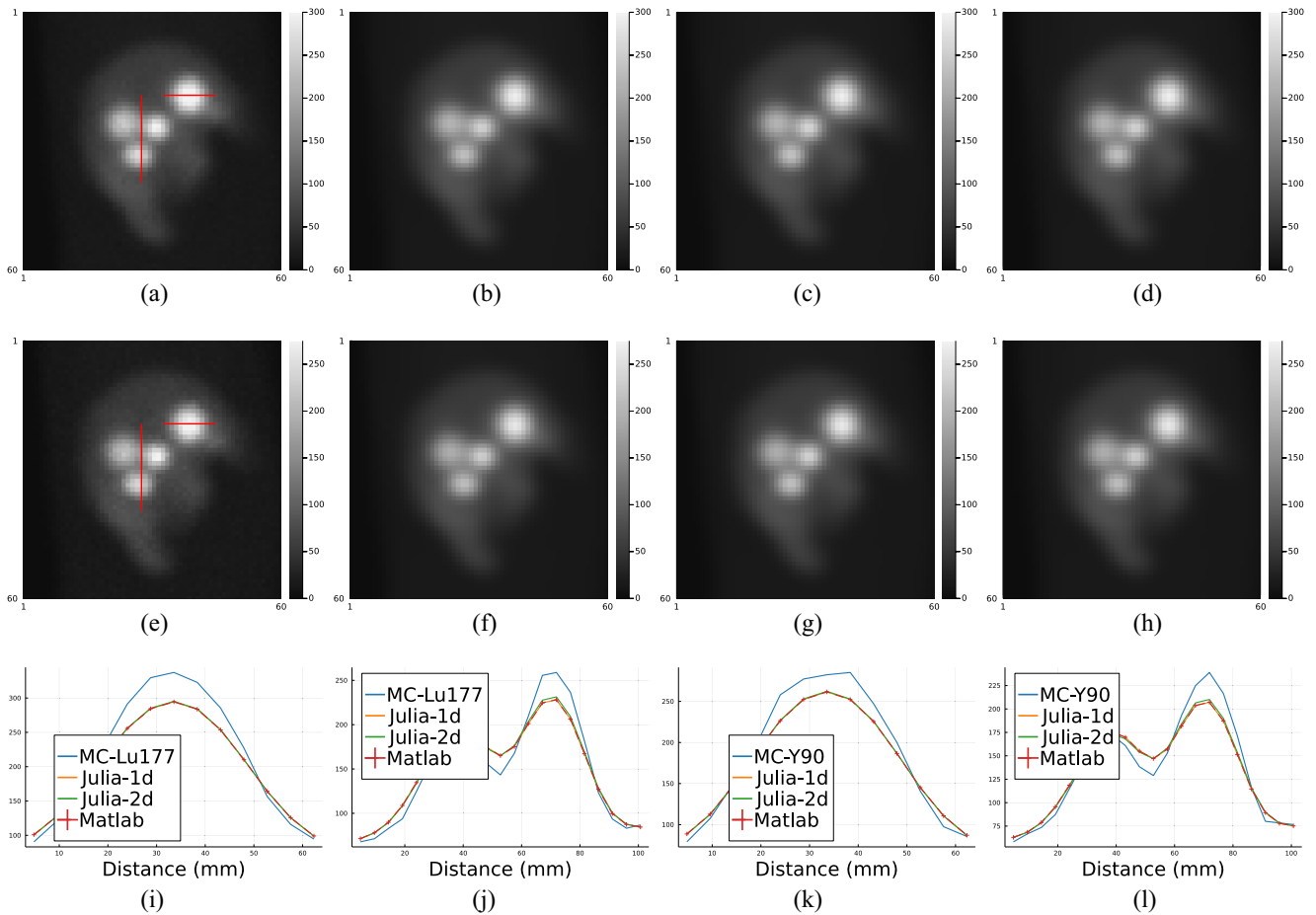


Fig. 2. Primary (scatter-free) projections generated by MC simulation, MATLAB projector, and our Julia projector with 3-pass 1-D linear interpolation and 2-D bilinear interpolation for image rotation, using ^{177}Lu and ^{90}Y radionuclides. (i)–(l) show line profiles across tumors as shown in (a) and (e), respectively. MC projections were scaled to have the same total activities as the MATLAB projector per field of view. (a) MC (^{177}Lu). (b) Julia 1-D (^{177}Lu). (c) Julia 2-D (^{177}Lu). (d) MATLAB (^{177}Lu). (e) MC (^{90}Y). (f) Julia 1-D (^{90}Y). (g) Julia 2-D (^{90}Y). (h) MATLAB (^{90}Y). (i) Horizontal profile (^{177}Lu). (j) Vertical profile (^{177}Lu). (k) Horizontal profile (^{90}Y). (l) Vertical profile (^{90}Y).

for ^{90}Y . We also compared the OSEM reconstructed images using Julia (2-D) and MATLAB projectors, where we did not observe notable difference, as shown in Fig. 3. The overall NRMSD between MATLAB and Julia (2-D) projector for the whole 3-D OSEM reconstructed image ranged from 2.5% to 2.8% across three noise realizations.

2) *Speed and Memory Use*: We compared the memory use and compute times between our Julia projector (with 2-D bilinear interpolation) and the MATLAB projector using different number of threads when projecting a $128 \times 128 \times 80$ image. Fig. 4 shows that our Julia projector has comparable computing time for a single projection with 128 view angles using different number of CPU threads while using only a very small fraction of memory ($\sim 5\%$) and preallocation time ($\sim 1\%$) compared to the MATLAB projector.

3) *Adjoint of Projector*: We generated a set of random numbers to verify that the backprojector is an exact adjoint of the forward projector. Specifically, we generated the system matrix of size $(8 \times 6 \times 7) \times (8 \times 8 \times 6)$ using random (non-negative) attenuation maps and random (symmetric) PSF. Fig. 5 compares the transpose of the forward projector to the backprojector. As shown in Fig. 5(d), the Frobenius norm error of our backprojector agrees well with the regular transpose within an accuracy of 10^{-6} across 100 different realizations,

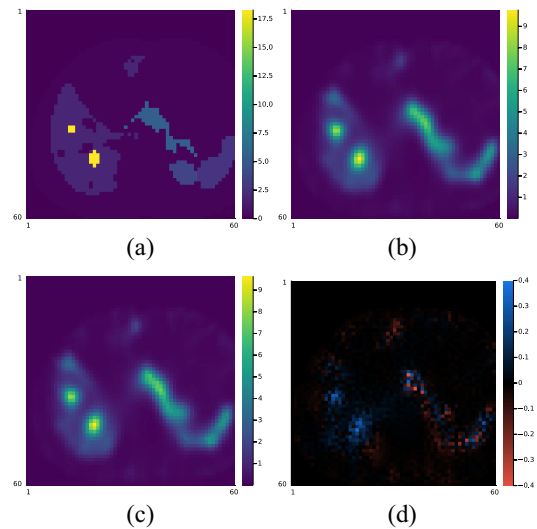


Fig. 3. Comparison of one slice of the $128 \times 128 \times 80$ OSEM reconstruction (16 iterations, 4 subsets) using MATLAB and Julia (2-D interpolation) projectors. (a) True activity. (b) OSEM-MATLAB. (c) OSEM-Julia 2-D. (d) Difference of (b) and (c).

as expected for 32-bit floating-point calculations. A more comprehensive comparison is available in the code tests at <https://github.com/JuliaImageRecon/SPECTrecon.jl>.

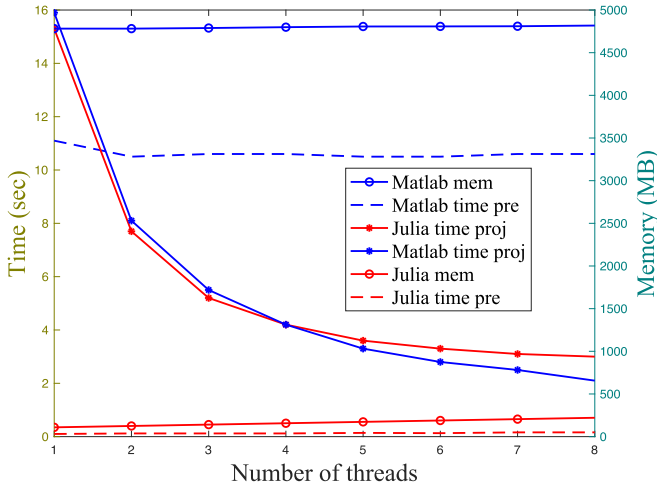


Fig. 4. Time and memory comparison between the MATLAB projector and our Julia projector for projecting 128 view angles of a $128 \times 128 \times 80$ image. “time pre” denotes the time cost for preallocating necessary arrays before projection; “time proj” denotes the time cost for a single projection; and “mem” denotes the memory usage. All methods were tested on MacOS with a 3.8-GHz 8-Core Intel Core i7 CPU.

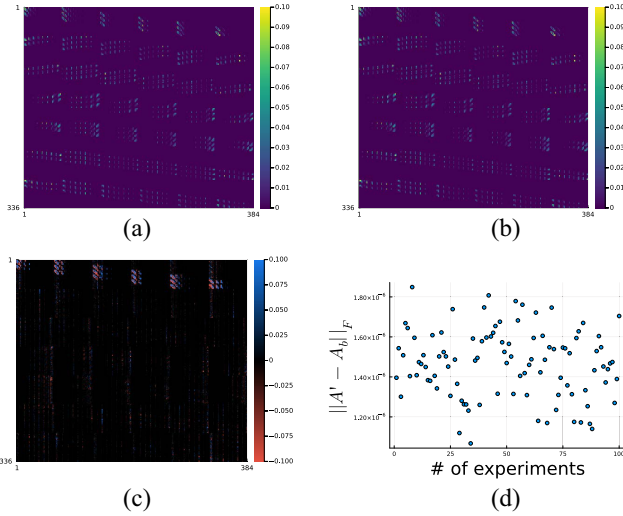


Fig. 5. Accuracy of the backprojector. In (d), A' denotes the regular transpose of A ; and A_b denotes the backprojector. (a) Regular transpose. (b) Backprojector. (c) Difference of (a) and (b) multiplied by 10^7 . (d) Frobenius norm error.

B. Comparison of CNN-Regularized EM Using Different Training Methods

This section compares end-to-end training with other training methods that have been used previously for SPECT image reconstruction, namely, the gradient truncation and sequential training. The training targets were simulated activity maps on ^{177}Lu XCAT phantoms and ^{177}Lu & ^{90}Y VP phantoms. We implemented an unrolled CNN-regularized EM algorithm with three outer iterations, each of which had one inner iteration. Only three outer iterations were used (compared to previous works such as [27]) because we used the 16-iteration 4-subset OSEM reconstructed image as a warm start for all reconstruction algorithms. We set the regularization parameter [defined in (5)] as $\beta = 1$. The regularizer was a 3-layer 3-D CNN,

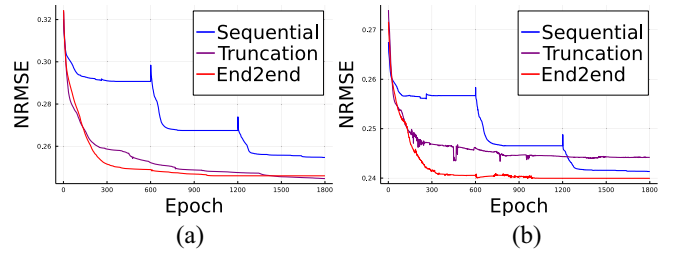


Fig. 6. Training and validation loss of three backpropagation methods. (a) Training loss. (b) Validation loss.

where each layer had a $3 \times 3 \times 3$ convolutional filter followed by ReLU activation (except the last layer), and hence had 657 trainable parameters in total. We added the input image \mathbf{x}_k to the output of CNN following the common residual learning strategy [37]. End-to-end training and gradient truncation could also work with a shared weights CNN approach, but were not included here for fair comparison purpose, since the sequential training only works with nonshared weights CNN. All the neural networks were initialized with the same parameters (drawn from a Gaussian distribution) and trained on an Nvidia RTX 3090 GPU for 600 epochs by minimizing mean-square error (loss) using AdamW optimizer [38] with a constant learning rate 0.002.

Besides line profiles for qualitative comparison, we also used mean activity error (MAE) and NRMSE as quantitative evaluation metrics, where MAE is defined as

$$\text{MAE} \triangleq \left| 1 - \frac{\frac{1}{n_p} \sum_{j \in \text{VOI}} \hat{\mathbf{x}}[j]}{\frac{1}{n_p} \sum_{j \in \text{VOI}} \mathbf{x}_{\text{true}}[j]} \right| \times 100\% \quad (9)$$

where n_p denotes the number of voxels in the voxels of interest (VOI). $\hat{\mathbf{x}}$ and \mathbf{x}_{true} denote the reconstructed image and the true activity map, respectively. The NRMSE is defined as

$$\text{NRMSE} \triangleq \frac{\sqrt{\frac{1}{n_p} \sum_{j \in \text{VOI}} (\hat{\mathbf{x}}[j] - \mathbf{x}_{\text{true}}[j])^2}}{\sqrt{\frac{1}{n_p} \left(\sum_{j \in \text{VOI}} \mathbf{x}_{\text{true}}[j] \right)^2}} \times 100\%. \quad (10)$$

All activity images were scaled by a factor that normalized the whole activity to 1 MBq per FOV before comparison. All quantitative results (Tables I–III) were averaged across three different noise realizations.

1) *Loss Function, Computing Time, and Memory Use:* We compared the training and validation loss using sequential training, gradient truncation, and end-to-end training. We ran 1800 epochs for each method on ^{177}Lu XCAT phantoms with the AdamW optimizer [38]. Fig. 6 shows that the end-to-end training achieved the lowest validation loss while it had comparable training loss with the gradient truncation (which became lower at around 1400 epochs). For visualization, we concatenated the first 600 epochs of each outer iteration for the sequential training method, as shown by the spikes in the sequential training curve. We ran 600 epochs for each algorithm for subsequent experiments because the validation losses were pretty much settled at around 600 epochs.

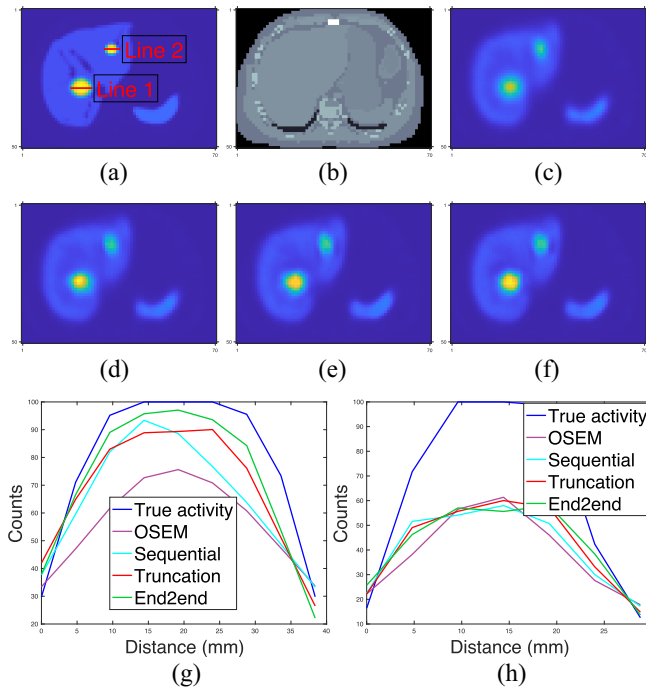


Fig. 7. Qualitative comparison of different training methods and OSEM tested on ^{177}Lu XCAT phantoms. (a)–(c) True activity map, attenuation map, and OSEM reconstruction (16 iterations and 4 subsets); (d)–(f) regularized EM using sequential training, gradient truncation, and end-to-end training, respectively; and (g) and (h) line profiles in (a).

We also compared the computing time of each training method. We found that for MLEM with three outer iterations and one inner iteration, where each outer iteration had a 3-layer CNN, sequential training took 48.6 s to complete a training epoch; while gradient truncation took 327.1 s and end-to-end training took 336.3 s. Under the same experiment settings, we found sequential training took less than 1 GB of memory to backpropagate through one outer iteration; compared to approximately 6 GB used in gradient truncation and end-to-end training that backpropagated through three outer iterations.

2) *Results on ^{177}Lu XCAT Phantoms:* We evaluated the CNN-regularized EM algorithm with three training methods on 4 ^{177}Lu XCAT phantoms we simulated. We generated the primary projections by calling forward operation of our Julia projector and then added uniform scatters with 10% of the primary counts before adding Poisson noise. Of the 4 phantoms, we used 2 for training, 1 for validation, and 1 for testing.

Fig. 7 shows that the end-to-end training yielded incrementally better reconstruction of the tumor in the liver center over OSEM, sequential training, and gradient truncation. Fig. 7(g) also illustrates this improvement by the line profile across the tumor. For the tumor at the top-right corner of the liver, all methods had comparable performance; this can be attributed to the small tumor size (5 mL) for which partial volume (PV) effects associated with SPECT resolution are higher; and hence its recovery is even more challenging.

Table I demonstrates that the CNN-regularized EM algorithm with all training methods (sequential training, gradient

TABLE I
AVERAGE (\pm STANDARD DEVIATION) MAE (%) AND NRMSE (%) ACROSS THREE NOISE REALIZATIONS OF ^{177}Lu XCAT PHANTOMS

MAE (%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion 1 (67mL)	12.5 \pm 0.6	6.7 \pm 1.8	2.8 \pm 0.9	2.1 \pm 1.1
Lesion 2 (10mL)	20.2 \pm 0.9	11.5 \pm 4.1	10.8 \pm 0.9	9.7 \pm 1.1
Lesion 3 (9mL)	25.6 \pm 0.6	18.8 \pm 0.4	15.2 \pm 0.9	12.8 \pm 1.0
Lesion 4 (5mL)	43.0 \pm 0.6	40.0 \pm 1.2	38.8 \pm 0.8	38.7 \pm 0.7
Liver	6.4 \pm 0.7	6.2 \pm 1.5	4.6 \pm 1.1	3.7 \pm 1.2
Lung	2.4 \pm 0.7	2.2 \pm 0.4	0.7 \pm 0.6	0.9 \pm 0.5
Spleen	14.2 \pm 0.9	12.6 \pm 2.4	8.9 \pm 0.7	9.3 \pm 1.5
Kidney	15.9 \pm 1.0	15.1 \pm 1.2	14.4 \pm 1.4	13.6 \pm 1.6
NRMSE (%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion 1 (67mL)	27.3 \pm 0.3	21.7 \pm 1.3	18.9 \pm 0.6	18.3 \pm 0.6
Lesion 2 (10mL)	26.8 \pm 0.6	19.2 \pm 2.2	16.4 \pm 0.4	16.3 \pm 0.8
Lesion 3 (9mL)	28.4 \pm 0.4	22.8 \pm 0.8	18.3 \pm 0.7	16.3 \pm 0.7
Lesion 4 (5mL)	43.5 \pm 0.5	41.1 \pm 1.3	40.0 \pm 0.7	40.2 \pm 0.6
Liver	28.5 \pm 0.1	25.0 \pm 0.8	24.3 \pm 0.3	24.5 \pm 0.3
Lung	32.1 \pm 0.1	31.2 \pm 1.1	29.5 \pm 0.3	30.4 \pm 0.4
Spleen	25.7 \pm 0.3	22.8 \pm 1.1	20.4 \pm 0.4	19.9 \pm 0.6
Kidney	40.8 \pm 0.3	39.7 \pm 0.4	39.7 \pm 0.2	39.2 \pm 0.3

truncation, and end-to-end training) consistently had lower reconstruction error than the OSEM method. Among all training methods, the proposed end-to-end training had lower MAE over nearly all lesions and organs than other training methods. The relative reduction in MAE by the end-to-end training was up to 32% (for lesion 3) compared to sequential training. End-to-end training also had lower NRMSE for most lesions and organs and was otherwise comparable to other training methods. The relative improvement compared to sequential training was up to 29% (for lesion 3).

3) *Results on ^{177}Lu VP Phantoms:* Next, we present test results on 8 ^{177}Lu VP phantoms. Out of 8 ^{177}Lu phantoms, we used 4 for training, 1 for validation, and 3 for testing.

Fig. 8 shows that the improvement of all learning-based methods was limited compared to OSEM, which was also evident from line profiles. For example, in Fig. 8(g), where the line profile was drawn on a small tumor. We found that OSEM yielded a fairly accurate estimate already, and we did not observe as much improvement as we had seen on ^{177}Lu XCAT phantoms for end-to-end training or even learning-based methods. Table II also demonstrates this observation. The OSEM method had substantially lower MAE and NRMSE compared to the errors shown for ^{177}Lu XCAT data (cf. Table I). Moreover, the end-to-end training method had comparable accuracy with gradient truncation. For example, gradient truncation was the best on lesion, liver, and lung in terms of MAE; end-to-end training had the lowest NRMSE on lesion, liver, lung, kidney, and spleen. Perhaps this could be due to the loss function used for training, i.e., MSE loss was used in our experiments so that end-to-end training might yield lower NRMSE. A more comprehensive study would be needed to verify this conjecture.

4) *Results on ^{90}Y VP Phantoms:* We also tested with 8 ^{90}Y VP phantoms. Of the 8 phantoms, we used 4 for training, 1 for validation, and 3 for testing.

Fig. 9 compares the reconstruction quality between the OSEM and CNN-regularized EM algorithm using sequential

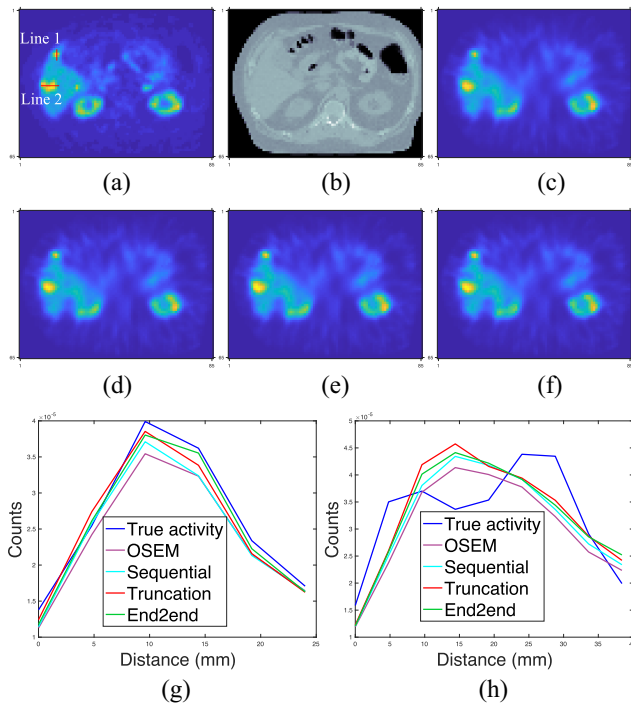


Fig. 8. Qualitative comparison of different training methods and OSEM tested on ¹⁷⁷Lu VP phantoms. (g) and (h) correspond to line profiles marked in (a). (a) True activity. (b) Attenuation map. (c) OSEM. (d) Sequential. (e) Truncation. (f) End2end. (g) Line 1 profile. (h) Line 2 profile.

TABLE II
AVERAGE(±STANDARD DEVIATION) MAE(%) AND NRMSE(%) ACROSS THREE NOISE REALIZATIONS OF ¹⁷⁷Lu VP PHANTOMS

MAE(%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion (6-152mL)	11.1 ± 2.5	9.4 ± 3.2	6.7 ± 2.4	7.3 ± 2.8
Liver	4.8 ± 0.1	4.5 ± 0.2	3.4 ± 0.6	4.0 ± 0.2
Healthy liver	4.1 ± 0.1	4.1 ± 0.1	3.5 ± 0.6	4.1 ± 0.2
Lung	3.4 ± 0.1	3.0 ± 0.2	2.4 ± 0.7	3.0 ± 0.5
Kidney	5.2 ± 0.3	4.3 ± 0.1	2.6 ± 0.1	2.3 ± 0.2
Spleen	0.8 ± 0.2	0.6 ± 0.1	1.3 ± 0.6	1.2 ± 0.4
NRMSE(%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion (6-152mL)	16.1 ± 2.2	14.9 ± 2.4	14.3 ± 1.7	14.2 ± 2.1
Liver	15.9 ± 0.2	15.3 ± 0.1	15.5 ± 0.6	15.3 ± 0.1
Healthy liver	16.8 ± 0.1	16.6 ± 0.1	17.3 ± 0.5	17.1 ± 0.3
Lung	22.3 ± 0.3	22.1 ± 0.4	22.0 ± 0.4	21.9 ± 0.5
Kidney	17.4 ± 0.1	16.8 ± 0.1	16.4 ± 0.3	16.3 ± 0.5
Spleen	13.5 ± 0.2	12.4 ± 0.3	12.3 ± 0.7	12.3 ± 0.5

training, gradient truncation, and end-to-end training. Visually, the end-to-end training reconstruction yields the closest estimate to the true activity. This is also evident through the line profiles [subfigures (m) and (n)] across the tumor and the liver.

Table III reports the MAE and NRMSE for lesions and organs across all testing phantoms. Similar to the qualitative assessment (Fig. 9), the end-to-end training also produced lower errors consistently across all testing lesions and organs. For instance, compared to sequential training/gradient truncation, the end-to-end training relatively reduced MAE on average by 8.7%/7.2%, 18.5%/11.0%, and 24.7%/16.1% for lesion, healthy liver, and lung, respectively. The NRMSE was also relatively reduced by 6.1%/3.8%, 7.2%/4.1%, and

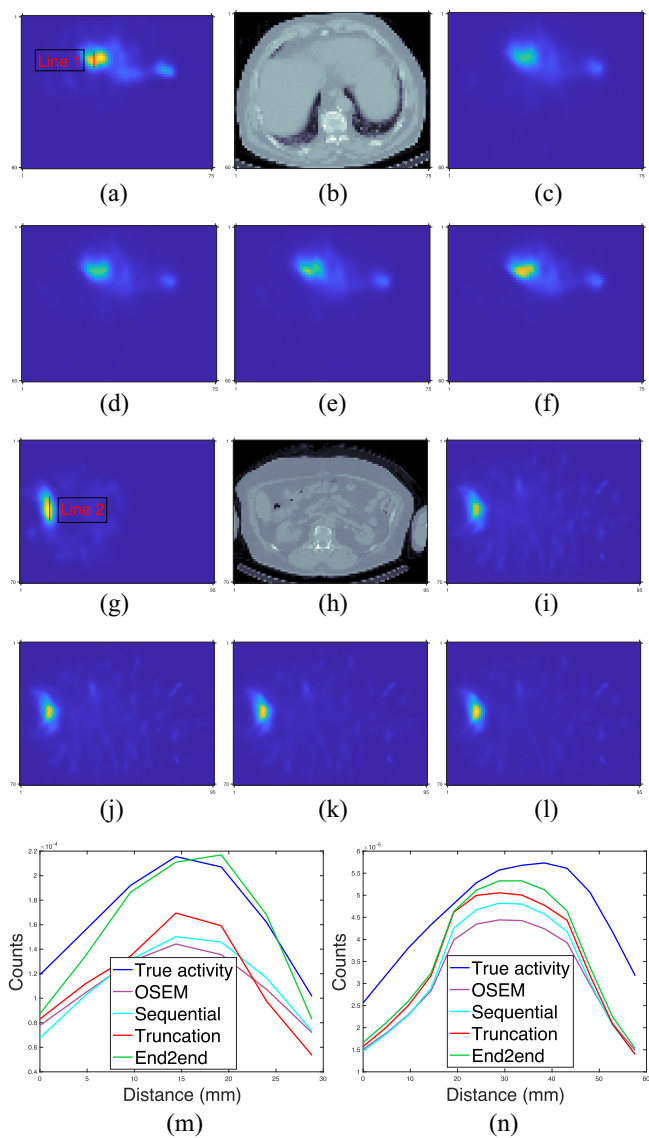


Fig. 9. Qualitative comparison of different training methods and OSEM tested on ⁹⁰Y VP phantoms. (a)–(f) and (g)–(l) show two slices from two testing phantoms. (m) and (n) correspond to line profiles in (a) and (g), respectively.

TABLE III
AVERAGE(±STANDARD DEVIATION) MAE(%) AND NRMSE(%) ACROSS THREE NOISE REALIZATIONS OF ⁹⁰Y VP PHANTOMS

MAE(%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion (3-356mL)	32.5 ± 1.3	25.3 ± 1.3	24.9 ± 1.0	23.1 ± 1.8
Liver	25.0 ± 0.1	18.7 ± 0.1	17.8 ± 1.3	15.6 ± 3.6
Healthy liver	25.1 ± 0.2	23.8 ± 0.5	21.8 ± 1.2	19.4 ± 3.1
Lung	88.4 ± 2.1	64.9 ± 1.6	58.3 ± 6.6	48.9 ± 8.4
NRMSE(%)				
Lesion/Organ	OSEM	Sequential	Truncation	End2end
Lesion (3-356mL)	35.3 ± 1.5	29.6 ± 1.4	28.9 ± 1.1	27.8 ± 1.2
Liver	29.9 ± 0.4	22.7 ± 0.1	22.1 ± 0.9	21.2 ± 1.5
Healthy liver	31.6 ± 0.4	27.9 ± 0.3	27.0 ± 0.9	25.9 ± 2.0
Lung	62.4 ± 1.3	59.2 ± 1.1	57.3 ± 3.0	55.6 ± 4.6

6.1%/3.0% for lesion, healthy liver, and lung, respectively. All learning-based methods consistently had lower errors than the OSEM method.

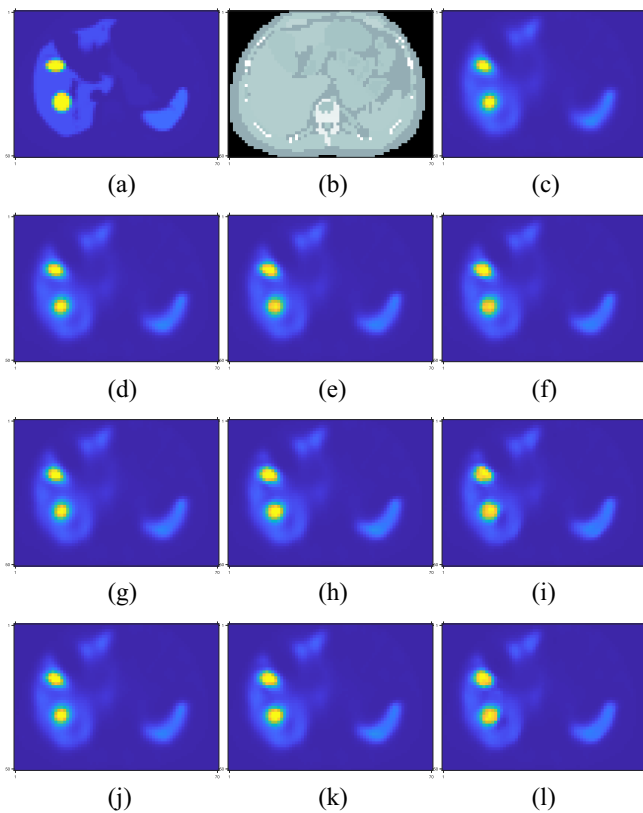


Fig. 10. Visualization of intermediate iteration results of different training methods. (a) True activity; (b) attenuation map; (c) OSEM; (d)–(f) sequential training; (g)–(i) gradient truncation; and (j)–(l) end-to-end training.

C. Results at Intermediate Iterations

One potential problem associated with end-to-end training (and gradient truncation) is that the results at intermediate iterations could be unfavorable, because they are not directly trained by the targets [39]. Here, we examined the images at intermediate iterations and did not observe such problems as illustrated in Fig. 10, where images at each iteration gave a fairly accurate estimate to the true activity. Perhaps under the shallow-network setting (e.g., three layers used here, with only three outer iterations), the network for each iteration was less likely to overfit the training data. Another reason could be due to the nonshared weights setting so that the network could learn suitable weights for each iteration.

IV. DISCUSSION

Training end-to-end CNN-based iterative algorithms for SPECT image reconstruction requires memory efficient forward–backward projectors so that backpropagation can be less computationally expensive. This work implemented a new SPECT projector using Julia that is an open-source, high-performance, and cross-platform language. With comparisons between MC and a MATLAB-based projector, we verified the accuracy, speed, and memory efficiency of our Julia projector. These favorable properties support efficient backpropagation when training end-to-end unrolled iterative reconstruction algorithms. Most modern DL algorithms process multiple data batches in parallel, so memory efficiency is

of great importance for the efficient training and testing neural networks. To that extent, our Julia projector is much more suitable than the MATLAB-based projector.

We used the CNN-regularized EM algorithm as an example to test end-to-end training and other training methods on different datasets, including ^{177}Lu XCAT phantoms, ^{177}Lu and ^{90}Y VP phantoms. Simulation results demonstrated that end-to-end training improved reconstruction quality on these datasets. For example, end-to-end training improved the MAE of lesion/liver in ^{90}Y phantoms by 8.7%/16.6% and 7.2%/12.4% compared to sequential training and gradient truncation. This improvement could be attributed to the correct gradient was used in backpropagation. Although the end-to-end training yielded the lowest reconstruction error on both ^{177}Lu XCAT phantoms and ^{90}Y VP phantoms, the reconstruction errors on ^{177}Lu VP phantoms were comparable with the gradient truncation. This could be due to the choice of loss functions and CNN architectures in the EM algorithm, which we will explore in the future. Also, we noticed that the recovery of the nonuniform activity in VP phantoms was generally higher than activity for the XCAT phantom (MAE reported in Tables I and II) because the assigned “true” activities at the boundaries of organs did not drop sharply, and instead, were blurred out. And therefore the OSEM algorithm was fairly competitive as reported in Table II; in ^{90}Y VP results, the OSEM performed worse than learning-based methods, which could be attributed to the high downscatter associated with ^{90}Y SPECT due to the continuous bremsstrahlung energy spectrum. We found all learning methods did not work very well for small tumors (e.g., 5 mL), potentially due to the worse PV effect. Reducing PV effects in SPECT images has been studied extensively [40], [41]. Recently, Xie et al. [42] trained a deep neural network to learn the mapping between PV-corrected and noncorrected images. Incorporating their network into our reconstruction model using transfer learning is an interesting future direction.

Although promising results were shown in previous sections, this work has several limitations. First, we did not test numerous hyperparameters and CNN architectures, nor with a wide variety of phantoms and patients for different radionuclides therapies. Second, our experiments used OSEM images as warm start to the CNN-regularized EM algorithm, where the OSEM itself was initialized with a uniform image. We did not investigate using other images such as uniform images as the start of the EM algorithm. Using a uniform image to initialize the network would likely require far more network iterations which would be very expensive computationally and therefore impractical. Additionally, this article used fixed regularization parameter [β in (5)] rather than declaring β as a trainable parameter. We compared different methods for backpropagation, which requires using the same cost function (4) for a fair comparison. If one set β as a trainable parameter, then different methods could learn different β values, leading to different cost functions. However, the investigation of trainable β values is an interesting future work. Another limitation is that we did not investigate more advanced parallel computing methods such as distributed computing using multiple computers to further accelerate our Julia implementation of

SPECT forward–backward projector. Such acceleration is feasible using existing Julia packages if needed. The compute times reported in Fig. 4 shows that the method needs a few seconds per 128 projection views using eight threads, which is already feasible for scientific investigation.

We also found there exists a tradeoff between computational cost and reconstruction accuracy for different training methods. End-to-end training yielded reconstruction results with the lowest MAE and NRMSE because the correct gradient was used during backpropagation. Sequential training yielded worse results, but it was significantly faster and more memory efficient than the end-to-end training method. It is notably faster because it splits the whole training process and trains each of neural networks separately, and its backpropagation does not involve terms associated with the MLEM algorithm, so sequential training is actually equivalent to training that neural network alone without considering the MLEM. Sequential training also used much less memory because the training was performed iteration by iteration, one network by one network, and hence the memory limitation did not depend on the number of unrolled iterations in the MLEM algorithm.

V. CONCLUSION

This article presents a Julia implementation of backpropagatable SPECT forward–backward projector that is accurate, fast, and memory efficient compared to MC and a previously developed analytical MATLAB-based projector. Simulation results based on ^{177}Lu XCAT phantoms, ^{90}Y and ^{177}Lu VP phantoms demonstrate that: 1) end-to-end training yielded reconstruction images with the lowest MAE and NRMSE when tested on XCAT phantoms and ^{90}Y VP phantoms, compared to other training methods (such as sequential training and gradient truncation) and OSEM and 2) for ^{177}Lu VP phantoms, the end-to-end training method yielded better results than sequential training and OSEM; but was rather comparable with gradient truncation. We also found there exists a tradeoff between computational cost and reconstruction accuracy in different training methods (e.g., end-to-end training and sequential training). These results indicate that end-to-end training, which is feasible with our developed Julia projector, is worth investigating for SPECT reconstruction.

ACKNOWLEDGMENT

All the authors declare that they have no known conflicts of interest in terms of competing financial interests or personal relationships that could have an influence or are relevant to the work reported in this article. Code for reproducing the results is available at <https://github.com/ZongyuLi-umich/end-to-end-spect>.

REFERENCES

- [1] S. James et al., “Current status of radiopharmaceutical therapy,” *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 109, no. 4, pp. 891–901, 2021, doi: [10.1016/j.ijrobp.2020.08.035](https://doi.org/10.1016/j.ijrobp.2020.08.035).
- [2] Y. Dewaraja et al., “MIRD pamphlet no. 23: Quantitative SPECT for patient-specific 3-dimensional dosimetry in internal radionuclide therapy,” *J. Nucl. Med.*, vol. 53, no. 8, pp. 1310–1325, 2012, doi: [10.2967/jnumed.111.100123](https://doi.org/10.2967/jnumed.111.100123).
- [3] L. A. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography,” *IEEE Trans. Med. Imag.*, vol. MI-1, no. 2, pp. 113–122, Oct. 1982, doi: [10.1109/TMI.1982.4307558](https://doi.org/10.1109/TMI.1982.4307558).
- [4] H. M. Hudson and R. S. Larkin, “Accelerated image reconstruction using ordered subsets of projection data,” *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–609, Dec. 1994, doi: [10.1109/42.363108](https://doi.org/10.1109/42.363108).
- [5] V. Y. Panin, G. L. Zeng, and G. T. Gullberg, “Total variation regulated EM algorithm [SPECT reconstruction],” *IEEE Trans. Nucl. Sci.*, vol. 46, no. 6, pp. 2202–2210, Dec. 1999, doi: [10.1109/23.819305](https://doi.org/10.1109/23.819305).
- [6] J. A. Fessler, “Penalized weighted least-squares image reconstruction for positron emission tomography,” *IEEE Trans. Med. Imag.*, vol. 13, no. 2, pp. 290–300, Jun. 1994, doi: [10.1109/42.293921](https://doi.org/10.1109/42.293921).
- [7] D. S. Lalush and B. M. W. Tsui, “A generalized Gibbs prior for maximum a posteriori reconstruction in SPECT,” *Phys. Med. Biol.*, vol. 38, no. 6, pp. 729–741, 1993, doi: [10.1088/0031-9155/38/6/007](https://doi.org/10.1088/0031-9155/38/6/007).
- [8] Y. K. Dewaraja, K. F. Koral, and J. A. Fessler, “Regularized reconstruction in quantitative SPECT using CT side information from hybrid imaging,” *Phys. Med. Biol.*, vol. 55, no. 9, pp. 2523–2539, 2010, doi: [10.1088/0031-9155/55/9/007](https://doi.org/10.1088/0031-9155/55/9/007).
- [9] S. Y. Chun, J. A. Fessler, and Y. K. Dewaraja, “Non-local means methods using CT side information for I-131 SPECT image reconstruction,” in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. Rec. (NSSMIC)*, 2012, pp. 3362–3366. doi: [10.1109/NSSMIC.2012.6551766](https://doi.org/10.1109/NSSMIC.2012.6551766).
- [10] G. Zeng et al., “A review on deep learning MRI reconstruction without fully sampled k-space,” *BMC Med. Imag.*, vol. 21, no. 1, p. 195, 2021, doi: [10.1186/s12880-021-00727-9](https://doi.org/10.1186/s12880-021-00727-9).
- [11] G. Yang et al., “DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction,” *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018, doi: [10.1109/TMI.2017.2785879](https://doi.org/10.1109/TMI.2017.2785879).
- [12] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, “Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss,” *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018, doi: [10.1109/TMI.2018.2820120](https://doi.org/10.1109/TMI.2018.2820120).
- [13] J. Minnema et al., “A review on the application of deep learning for CT reconstruction, bone segmentation and surgical planning in oral and maxillofacial surgery,” *Dentomaxillofac. Radiol.*, vol. 51, no. 7, 2022, Art. no. 20210437, doi: [doi:10.1259/dmfr.20210437](https://doi.org/10.1259/dmfr.20210437).
- [14] H. Chen et al., “LEARN: Learned experts’ assessment-based reconstruction network for sparse-data CT,” *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1333–1347, Jun. 2018, doi: [10.1109/TMI.2018.2805692](https://doi.org/10.1109/TMI.2018.2805692).
- [15] A. J. Reader, G. Corda, A. Mehranian, C. D. Costa-Luis, S. Ellis, and J. A. Schnabel, “Deep learning for PET image reconstruction,” *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 1–25, Jan. 2021, doi: [10.1109/TRPMS.2020.3014786](https://doi.org/10.1109/TRPMS.2020.3014786).
- [16] K. Kim et al., “Penalized PET reconstruction using deep learning prior and local linear fitting,” *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1478–1487, Jun. 2018, doi: [10.1109/TMI.2018.2832613](https://doi.org/10.1109/TMI.2018.2832613).
- [17] A. Mehranian and A. J. Reader, “Model-based deep learning PET image reconstruction using forward–backward splitting expectation–maximization,” *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 54–64, Jan. 2021, doi: [10.1109/TRPMS.2020.3004408](https://doi.org/10.1109/TRPMS.2020.3004408).
- [18] W. Shao, S. P. Rowe, and Y. Du, “SPECTnet: A deep learning neural network for SPECT image reconstruction,” *Ann. Transl. Med.*, vol. 9, no. 9, p. 819, 2021, doi: [10.21037/atm-20-3345](https://doi.org/10.21037/atm-20-3345).
- [19] W. Shao, M. G. Pomper, and Y. Du, “A learned reconstruction network for SPECT imaging,” *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 26–34, Jan. 2021, doi: [10.1109/TRPMS.2020.2994041](https://doi.org/10.1109/TRPMS.2020.2994041).
- [20] S. Mostafapour et al., “Deep learning-guided attenuation correction in the image domain for myocardial perfusion SPECT imaging,” *J. Comput. Des. Eng.*, vol. 9, no. 2, pp. 434–447, Apr. 2022, doi: [10.1093/jcde/qwac008](https://doi.org/10.1093/jcde/qwac008).
- [21] G. Shen, K. Dwivedi, K. Majima, T. Horikawa, and Y. Kamitani, “End-to-end deep image reconstruction from human brain activity,” *Front. Comput. Neurosci.*, vol. 13, p. 21, Apr. 2019, doi: [10.3389/fncom.2019.00021](https://doi.org/10.3389/fncom.2019.00021).
- [22] S. Mukherjee, M. Carioni, O. Öktem, and C.-B. Schönlieb, “End-to-end reconstruction meets data-driven regularization for inverse problems,” in *Advances in Neural Information Processing Systems*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2021, pp. 21413–21425. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/file/b2df0a0d4116c55f81fd5aa1ef876510-Paper.pdf>

- [23] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, "Improved low-count quantitative PET reconstruction with an iterative neural network," *IEEE Trans. Med. Imag.*, vol. 39, no. 11, pp. 3512–3522, Nov. 2020, doi: [10.1109/TMI.2020.2998480](https://doi.org/10.1109/TMI.2020.2998480).
- [24] A. Sahiner, M. Mardani, B. Ozturkler, M. Pilanci, and J. M. Pauly, "Convex regularization behind neural reconstruction," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19. [Online]. Available: <https://openreview.net/forum?id=VErQxgyrbfn>
- [25] B. Ozturkler, A. Sahiner, M. Pilanci, S. Vasanaawala, J. Pauly, and M. Mardani, "Scalable and interpretable neural MRI reconstruction via layer-wise training," in *Proc. Int. Soc. Magn. Reson. Med.*, 2021. [Online]. Available: <https://index.mirasmart.com/ISMRM2021/PDFfiles/1953.html>
- [26] G. Corda-D'Incan, J. A. Schnabel, and A. J. Reader, "Memory-efficient training for fully unrolled deep learned PET image reconstruction with iteration-dependent targets," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 6, no. 5, pp. 552–563, May 2022, doi: [10.1109/TRPMS.2021.3101947](https://doi.org/10.1109/TRPMS.2021.3101947).
- [27] A. Mehranian and A. J. Reader, "Model-based deep learning PET image reconstruction using forward-backward splitting expectation-maximisation," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2019, pp. 1–4, doi: [10.1109/NSS/MIC42101.2019.9059998](https://doi.org/10.1109/NSS/MIC42101.2019.9059998).
- [28] G. L. Zeng and G. T. Gullberg, "Frequency domain implementation of the three-dimensional geometric point response correction in SPECT imaging," in *Proc. Conf. Rec. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, vol. 3, 1991, pp. 1943–1947. doi: [10.1109/NSSMIC.1991.259256](https://doi.org/10.1109/NSSMIC.1991.259256).
- [29] E. V. R. Di Bella, A. B. Barclay, R. L. Eisner, and R. W. Schafer, "A comparison of rotation-based methods for iterative reconstruction algorithms," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 6, pp. 3370–3376, Dec. 1996, doi: [10.1109/23.552756](https://doi.org/10.1109/23.552756).
- [30] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 132–137, Mar. 1995, doi: [10.1109/42.370409](https://doi.org/10.1109/42.370409).
- [31] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Med. Phys.*, vol. 37, no. 9, pp. 4902–4915, 2010, doi: [10.1118/1.3480985](https://doi.org/10.1118/1.3480985).
- [32] A. T. Soderlund, J. Chaal, G. Tjio, J. J. Totman, M. Conti, and D. W. Townsend, "Beyond 18F-FDG: Characterization of PET/CT and PET/MR scanners for a comprehensive set of positron emitters of growing application— ^{18}F , ^{11}C , ^{89}Zr , ^{124}I , ^{68}Ga , and ^{90}Y ," *J. Nucl. Med.*, vol. 56, no. 8, pp. 1285–1291, 2015, doi: [10.2967/jnumed.115.156711](https://doi.org/10.2967/jnumed.115.156711).
- [33] H. Xiang, H. Lim, J. A. Fessler, and Y. K. Dewaraja, "A deep neural network for fast and accurate scatter estimation in quantitative SPECT/CT under challenging scatter conditions," *Eur. J. Nucl. Med. Mol. Imag.*, vol. 47, no. 13, pp. 2956–2967, 2020, doi: [10.1007/s00259-020-04840-9](https://doi.org/10.1007/s00259-020-04840-9).
- [34] M. Ljungberg, "The SIMIND Monte Carlo program." 2012. [Online]. Available: <https://doi.org/10.1201/b13073-8>
- [35] Y. K. Dewaraja et al., "A pipeline for automated voxel dosimetry: Application in patients with multi-SPECT/CT imaging after ^{177}Lu -peptide receptor radionuclide therapy," *J. Nucl. Med.*, vol. 63, no. 11, pp. 1655–1672, 2022, doi: [10.2967/jnumed.121.263738](https://doi.org/10.2967/jnumed.121.263738).
- [36] J. Fessler, "Michigan image reconstruction toolbox." 2022. [Online]. Available: <https://github.com/JeffFessler/mirt>
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778. doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [38] I. Loshchilov and F. Hutter, "Decoupled weight decay Regularization," in *Proc. ICLR*, 2019, pp. 1–18.
- [39] F. Knoll, "Rise of the machines (in MR image reconstruction)," in *Proc. IMA Workshop Comput. Imag.*, 2019. [Online]. Available: http://helper.ipam.ucla.edu/publications/dlm2020/dlm2020_15990.pdf
- [40] P. H. Pretorius and M. A. King, "Diminishing the impact of the partial volume effect in cardiac SPECT perfusion imaging," *Med. Phys.*, vol. 36, no. 1, pp. 105–115, 2009, doi: [10.1118/1.3031110](https://doi.org/10.1118/1.3031110).
- [41] A. Grings, C. Jobic, T. Kuwert, and P. Ritt, "The magnitude of the partial volume effect in SPECT imaging of the kidneys: A phantom study," *EJNMMI Phys.*, vol. 9, p. 18, Mar. 2022, doi: [10.1186/s40658-022-00446-2](https://doi.org/10.1186/s40658-022-00446-2).
- [42] H. Xie et al., "Segmentation-free partial volume correction for cardiac SPECT using deep learning," *J. Nucl. Med.*, vol. 63, no. S2, p. 2264, 2022.