Deep Learning for Passive Gamma Emission Tomography

Carlos Sanchez-Belenguer[1], Alvaro Casado-Coscolla[2], and Erik Wolfart[1]

[1]European Commission, Joint Research Centre (JRC), Ispra, Italy. [name.surname]@ec.europa.eu
[2]Seidor Italy Srl, Milan, Italy*. alvaro.casado-coscolla@ext.ec.europa.eu

Abstract

In this paper, we address the problem of generating and enhancing Passive Gamma Emission Tomography (PGET) data from a deep learning perspective. The PGET instrument has been developed for the verification of spent nuclear fuel and relies on image reconstruction and analysis algorithms to detect missing or substituted fuel pins. High quality simulations are required for validating and further improving such techniques. However, reproducing the behavior of the original instrument is not straight-forward: complex Monte Carlo simulations need to be evaluated in order to compute vast amounts of photon histories. This makes this task extremely time consuming, taking up to several days to compute a single measurement. We propose the use of Convolutional Neural Networks (CNNs) for complementing and speeding up such process. More specifically, in this work we introduce a U-Net autoencoder that learns the mapping between incomplete/noisy data and its corresponding full sinogram. To do so, we exploit the fact that sinograms are highly redundant: the contribution of a single pin can be observed from many different directions. Our CNN learns the underlying model of the data to effectively exploit these redundancies and to make informed predictions of complete PGET sinograms starting from partial views. The experimental evaluation of our trained system shows very accurate results using only a small fraction of data and with execution times below one second. The technique is suitable for efficiently generating synthetic PGET sinograms, as well as for enhancing real measurement data.

1 Introduction

The backend of the nuclear fuel cycle, in particular the interim and long-term storage of spent fuel assemblies (SFA), is currently one of the main challenges in nuclear safeguards. After the SFAs are removed from the reactor core, they are transferred to water ponds for cooling and ‘short-term’ storage. Over the coming years, large numbers of SFAs will be transferred to dry storage, either to interim storage facilities or for final disposal. Before the transfer, safeguards inspectors need to verify the operator’s SFA declaration, as the verification is practically impossible once the fuel has been loaded in the interim/final storage canisters. The inspector should verify that no fuel pin has been removed or replaced. The quantification of the radioactive activity for each pin would be of additional value. Due the high radiation level and other operational constraints, it is not possible to access the single pins, but the measurement has to be carried out for the entire assembly using a non-destructive method.

Since no adequate verification tools were previously available, the Passive Gamma Emission Tomography (PGET) tool was developed and approved by the IAEA for safeguards use in December 2017 [1–5]. PGET uses two opposing arrays of gamma detectors that measure the gamma radiation emitted by the fuel pins. The arrays rotate around the fuel assembly step-by-step, thus generating 2D sinograms. The gamma radiation is measured in several broad energy windows in order to identify different fission products in the SFA.

*Working under contract for JRC, Ispra

1.1 Previous works

The verification of SFAs using PGET sinograms relies on advanced image filtering, reconstruction and analysis techniques. Generally, the pipeline is divided into three consecutive stages, where the quality of the results achieved in one step heavily affects the subsequent ones: (1) data pre-processing for correcting the non-uniform response of the detectors, (2) cross-sectional image reconstruction of the assembly and (3) image segmentation for pin identification and characterization. The first step is typically addressed with a Gaussian filtering over summed sinograms as detailed in [1]. The second relies on either analytic [6] or algebraic image reconstruction techniques that may include the modelling of attenuation [7] or scattering [8, 9]. The last stage is supported by image segmentation algorithms that may vary from template matching [10] to the use of Convolutional Neural Networks (CNNs) [8], amongst others.

Tomographic image reconstruction and analysis is a well-known problem in other domains, like medicine [11]. However, the PGET imposes some unique constraints due to the high activity of the pins and self-attenuation that require the development of dedicated techniques. Most of the works addressing these developments rely on simulations for validating and benchmarking their results [7–9, 12]. Simulations are attractive for two main reasons: on the one hand, real measurements are scarce and sensitive. On the other hand, simulations can provide realistic data for various diversion scenarios that would otherwise be impractical with physical experiments.

Simulations are generally performed by using Monte Carlo radiation transport codes [13] due to their completeness and realistic results. However, Monte Carlo simulations are computationally intensive tasks that impose the need of advanced computers and that may take several days to be performed. The case of the PGET is particularly challenging due to the strong self-attenuation of the sources, the reduced size of the detectors together with their collimation and the distance to the source. All these constraints make that the photon current on one detector element can be up to eight orders of magnitude lower than the photon emission rate [14]. To address these issues, general optimization methods have been developed, like variance reduction techniques [15], or two-phased simulations that target specifically the PGET unique characteristics [14]. Nevertheless, simulations times remain in the order of days.

During the last decades, the performance issues of Monte Carlo simulations have been addressed from a different domain: Computer Graphics. In this field, real-time light transport simulations are required for achieving realistic renders, where the target refresh rate imposes processing times in the order of milliseconds. To satisfy this time constraint, this task has been divided into two steps: (1) a fast Monte Carlo simulation, where very few photons are simulated, producing very noisy results and (2) a de-noising technique that exploits the partial results of the previous stage to enhance the final result in an informed way. CNNs have proven to be perfect candidates for light transport Monte Carlo de-noising applications [16, 17], as they can effectively learn complex relations between input images (noisy, incomplete simulations) and target images (final simulations) and they are very easy to train for this task: the training set consists on pairs of incomplete/complete simulations.

One of the most common CNN architectures for segmentation, de-nosing and inpainting applications is the U-Net [18]. It is based on an auto-encoder architecture, where skip connections are added between the encoder and the decoder blocks in each level of the model. It has been applied to many fields, including general image de-noising [19, 20], SPECT de-noising [21] and also for de-noising Monte Carlo dose simulations [22].

1.2 Overview

In this paper we address the performance issues of Monte Carlo simulations for the PGET from the deep learning perspective: we propose a U-Net auto-encoder that learns the mapping between incomplete/noisy data and its corresponding full sinogram. We overcome the data hunger of Deep Neural Networks by introducing a fast Monte Carlo simulator that generates near real-time sinograms that are used for both, training and validation. Finally, we assess the degree of generalization achieved by training only with synthetic data. To do so, we evaluate our trained CNN with real acquisitions [23] and perform a quantitative analysis of the results.

The remainder of this paper is organized as follows: Section 2 explains our technique, distinguishing between the data simulation (Section 2.1) and the deep learning approach (Section 2.2). Section 3 presents the results achieved with our method and Section 4 outlines the main conclusions of our work and future developments.

2 Approach

Our approach is based on the fact that PGET sinograms are highly redundant and with a strong geometrical component: the gamma emissions of a single pin can be observed with different degrees of attenuation from all the orientations of the detector array. The goal is to generate a model that is aware of the spatial correlations of data within a single sinogram (i.e., a data-driven approach). This way, when facing situations with incomplete sinograms or low statistics, our model can perform informed interpolations in a holistic manner.

Training a CNN capable of learning such a complex model, whilst ensuring proper generalization, requires a large number of training samples. Unfortunately, traditional Monte Carlo neutron transport simulations are too slow and become impractical when aiming for generating thousands of training sinograms. We address this issue by proposing a novel near real-time simulator and by introducing data augmentation techniques during the training stage.

2.1 Data simulation

Our simulator is based on the Monte Carlo paradigm: sinograms are estimated by integrating all the detections produced after calculating vast amounts of photon histories. The more *effective photons* are simulated, the more accurate the final result will be. We consider an effective photon the photon that is originated from one pin of the SFA and that ends up producing a detection in the PGET tool. Since we aim for high performance in our simulations, the goal from a computational complexity point of view is to simulate as many effective photons as possible (i.e., ensure that most of the histories calculated end up in a detection), without biasing the final result and respecting the underlying randomness of Monte Carlo approaches.

To address this computational complexity premise and to optimize the implementation of our simulator, several acceleration strategies have been developed. This section provides an overall description of such strategies and some intuitions about their impact in the final result. A detailed explanation will be provided in a future publication due to space limitations and because we consider that simulation details are out of the scope of this paper.

The PGET tool considered in this work is the revised version detailed in [4], with 182 cadmium zinc telluride detectors and performing continuous-motion acquisitions that produce sinograms with 360 orientations. The assembly layouts modelled are the PWR 17x17, BWR SVEA-96, WWER-440, WWER-1000 and GE14 10x10. For the moment, our simulator is energy agnostic (i.e., monochromatic) and does not consider the detector response. The photon interactions considered are photon-pin and photon-water absorptions, due to the photoelectric effect, and photon-pin scattering. Cross sections of interactions are decoupled from nuclear physics and expressed in probabilistic terms. Finally, simulations are performed in a 2D space.

These simplifications impose a limit on the accuracy of our simulations when compared with traditional MCNP simulations. However, considering that our approach is data-driven and the strong role that geometry plays in the final sinograms, the goal of the simulator is not to produce perfect sinograms given a SFA detailed description in nuclear terms (initial enrichment, burnup, cooling time), but to generate data that captures the main characteristics of real sinograms (see Figure 1). This will allow our CNN to learn most of the parameters of the problem we are

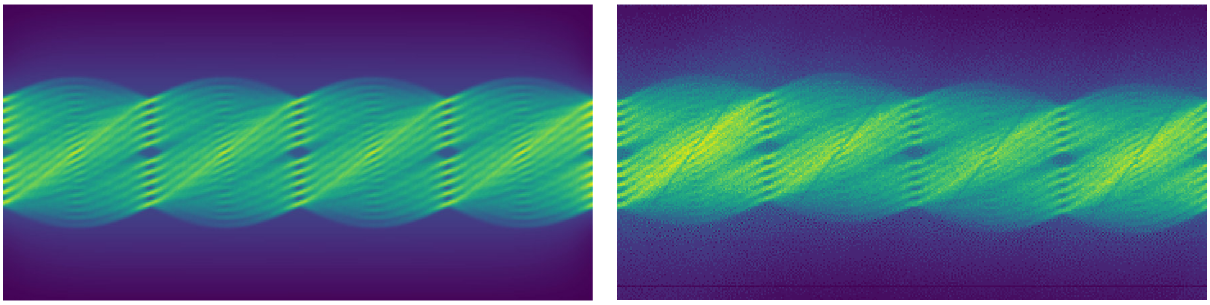

Figure 1: Comparison between our simulations (left) and real data (right). Notice how, despite the simplifications introduced, the degree of resemblance between the two sinograms is considerable, allowing a CNN to learn most of the features of the problem by using only simulations.

facing. Then, if it needs to be applied to a specific set of simulations (or real acquisitions), a final fine-tuning stage will be necessary, where the trained model will be re-trained with few real samples to fit the new requirements, as demonstrated in [24]. It is also worth noting that, from a complexity point of view, our simulations pose a more challenging learning problem, as all pin parameters (emission, absorption and scattering probabilities) are completely decoupled and can take random values ranging between 0% and 100%.

For accelerating simulations, we decouple the photon interactions inside the assembly from the photon detection in the detector array: i) in a first step, we efficiently simulate the interactions (absorption and scattering) to compute the exit position and direction (in 2D) of each photon leaving the fuel assembly; ii) a specific detector element can observe this photon if the orientation of detector array is within a certain range. The size of the angular range differs between the detector elements and thus we can compute the 'probability' with which each detector element will observe the photon. Therefore, a single photon simulation generates a continuous probability distribution contributing to a set of different sinogram pixels, i.e. the required number of simulated photons is several orders smaller than in a traditional approach where many photons need to be simulated to obtain a single detection.

Finally, we take advantage on the fact that this type of simulations fall into the area of computational problems known as *embarrassingly parallel*. Depending on the way we use the simulator we perform full parallelization at pin level (in case we are simulating a single sinogram) or at sinogram level (in case we are batch-simulating a training dataset).

2.2 Deep learning approach

Our approach aims to enhance PGET data. In this work we face the two specific use-cases illustrated in Figure 2: de-noising low statistics sinograms and addressing the limited view problem. The first one typically happens when the acquisition time was too short or when the Monte Carlo simulation was early-stopped. The second one attempts to *fill the gaps* when only a subset of sensor readings are available. In the deep learning domain, these two problems fall, respectively, into the de-noising and inpainting applications.

The architecture of our CNN (Figure 3) is based on the U-Net auto-encoder, with 4 down-sample blocks, 4 up-sample blocks and their corresponding skip connections. This architecture imposes a constraint in the data dimensionality: input sinograms need to be down-scaled 4 times by a factor of 2 and then up-scaled again. No fractional parts are allowed during such operations and, generally, powers of two are desirable with, ideally, squared aspect ratios. Consequently, as raw sinograms are 360 pixel wide (one per orientation) and 182 pixel high (one per detector), we pad them into 512x512 images before passing them through the CNN. For the vertical padding, the two horizontal bands are filled with zeroes. For the horizontal padding, we exploit the symmetries of sinograms to fill in the two vertical bands with their corresponding counts from

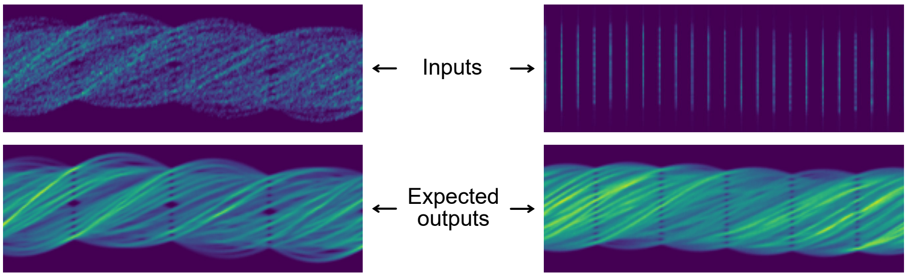

Figure 2: Use-cases faced in this approach: de-noising low statistics sinograms (left) and addressing the limited view problem (right). In both cases, the goal is to train a CNN that is able to extrapolate the information contained in the upper sinogram (input) in order to predict the lower one (expected output) as accurately as possible.

the original sinogram (e.g., the left band covers the detector orientations ranging between -76 and 0 degrees, which can be retrieved from the last 76 columns of the original sinogram). This way, we avoid hard variations in the input data that may mislead the feature learning of the CNN (padding with zeroes horizontally would introduce two sharp edges between the sinogram and the padding vertical bands).

To improve the speed of the learning stage, sinograms are linearly normalized into the range $[0, 1]$. Additionally, to improve generalization and exploiting the sinogram symmetries, in each training iteration sinograms are randomly shifted horizontally. This augmentation technique is equivalent to rotating the SFA in the simulator and allows to retrieve 360 equivalent training samples for each simulation performed (each pixel shift corresponds to a one degree rotation).

The training consists on inputting a low-quality sinogram (either a noisy one or one with limited views) and comparing the prediction of the CNN with the target sinogram. Based on the differences, the back-propagation algorithm updates the weights of the CNN and the process continues until convergence is detected (i.e., the CNN's performance does not improve anymore). The loss function used for the training is the mean squared error over the per-pixel differences between the target and predicted sinograms.

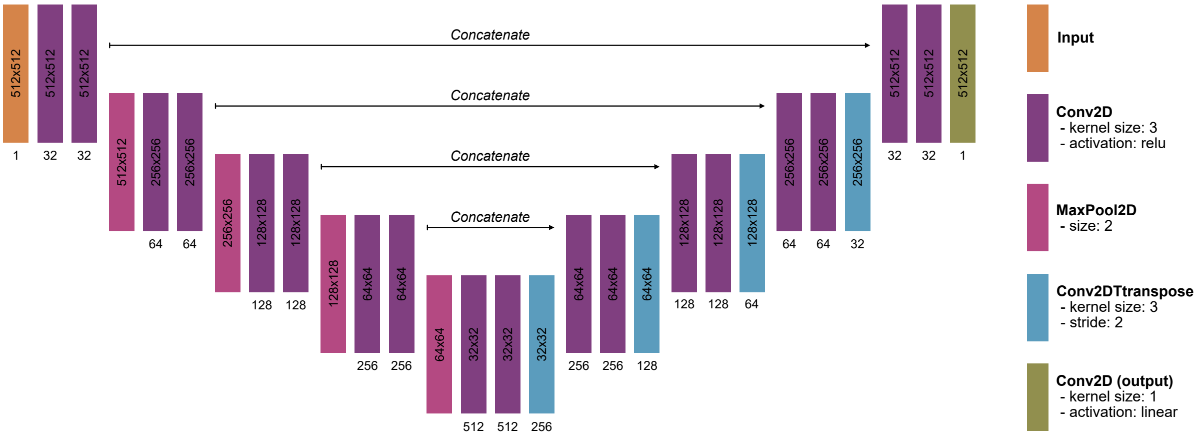

Figure 3: U-Net architecture used. The numbers inside each layer indicate the input size in the two major dimensions. The number below indicates the number of channels for the input and output units or the number of filters for the convolutional hidden layers.

3 Results

The simulations and the CNN training were performed using a computer with an nVidia GeForce RTX 3090 GPU and an Intel i9-10900X processor, with 10 physical cores (20 logical processors) at a maximum clock speed of 4.5 GHz.

For training and evaluating the performance of our CNN we simulated 10,000 sinograms. Those simulations included samples of the 5 SFA layouts modelled, where the placement of the assembly was randomly set, both in terms of orientation and position wrt. the center of the tool, but always ensuring a valid setup (i.e., no collisions between the SFA and the PGET tool). At pin level, the overall number of full pins was randomly selected, considering a minimum occupancy of 50%. Pin emission, absorption and scattering probabilities were set to random values where no correlation was forced (the only constraint was that the sum of absorption and scattering probabilities had to be lower than 1).

Calculating the LUT table of the simulator that models detection probabilities for the PGET instrument took less than 10 minutes and produced a binary file of 995 MB. This needs to be calculated only once per instrument specifications and gets loaded by the simulator at bootstrap.

Each simulation was performed in 1,024 sequential iterations where, for each iteration, a maximum of 3,000 photon histories where computed per pin (based on its emission probability). The results of each iteration where integrated into a single sinogram that, at the end of the simulation, is considered to be the ground truth for both, our training and our evaluation experiments. Intermediate results were stored for iterations 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512. Figure 2 shows the results of the first iteration (top-left) and the full simulation (bottom-left). After iteration 1024 no significant improvement in the results was observed. The total simulation time for the 10,000 sinograms was around 8 hours, setting an average time per simulation below 3 seconds. An additional set of 1,000 sinograms was generated for validation purposes (i.e., those sinograms were excluded during the training stage).

The training was performed with Tensorflow in mini-batches of 8 samples, using an Adam optimizer with its default learning rate (10^{-3}) and with a HeNormal weight initialization for the convolutional layers. In total we trained 14 different CNNs: 7 for the de-noising problem, where each net was facing a different level of noise in the input data and another 7 for the limited view problem, where each net was facing a different number views in the input data. For the de-nosing ones, during the training we inputted the intermediate results saved by our simulator at iterations 1, 2, 4, 8, 16, 32 and 64. For the limited view ones, during the training we inputted the final sinogram (iteration 1024), horizontally sub-sampled with factors of 2, 4, 6, 8, 10, 12 and 15. In both cases, the target output was the corresponding full final sinogram. Figure 2 shows one example of the training data used for the CNN that learnt the de-nosing for iteration 1 (left) and another example of the training data used for the CNN that learnt the limited view problem with a factor of 15 (right).

Each CNN was trained for 50 epochs (i.e., 50 training iterations over the complete training set). For each epoch, sinograms where augmented by performing random shifts in the horizontal axis (as explained in Section 2.2). The full training of each CNN took less than 3 hours, as it was early-stopped at epoch 50 due to the presence of over fitting signs in the learning curves.

For each validation sinogram (i.e., a sinogram that was not part of the training set), we assess the accuracy of our trained networks by comparing the cross-sectional reconstruction of the ground truth (simulation at iteration 1024) against the cross-sectional reconstruction of the prediction produced by each trained CNN after inputting an incomplete sinogram (either a noisy one or one with limited views). Reconstructions are performed with the inverse radon transform provided by the scikit-image library in Python, that relies on the Filtered Back Projection (FBP) algorithm [6]. The metric used for assessing the similarity between reconstructions is the Structural Similarity Index Measure (SSIM) [25]. The higher the SSIM between ground truth and predictions, the more accurate the results.

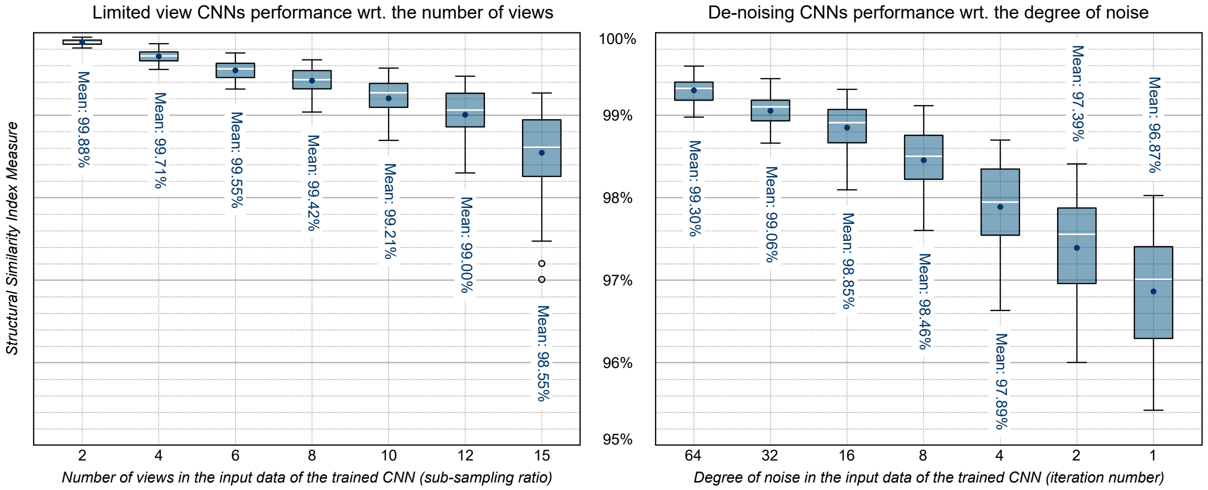

Figure 4: Overall accuracy of all our trained CNN's. Boxes represent the quartile distribution of the performance achieved by each CNN after evaluation. The white line represents the median SSIM, whilst the blue dot represents the mean SSIM.

Figure 4 shows the overall accuracy of our trained CNN's. The left plot corresponds to the limited view case, where the sub-sampling ratio increases along the X axis. The right plot corresponds to the de-noising case, where the degree of noise in the input data increases along the X axis. In both cases, the complexity increases along the X axis. Figure 5 shows some examples of the outputs produced by our trained CNN's compared to the ground truth results.

Considering the results shown in Figure 4 and the fact that the execution time of our trained CNNs is of a fraction of a second, the potential of this technique for Monte Carlo simulations becomes evident: in the most extreme case, a simulation could be early stopped when only 1/1024 photon histories have been calculated. By passing this intermediate result to our CNN trained for iteration 1, the outcome would be a noise-free sinogram that is (on average) 96.87%

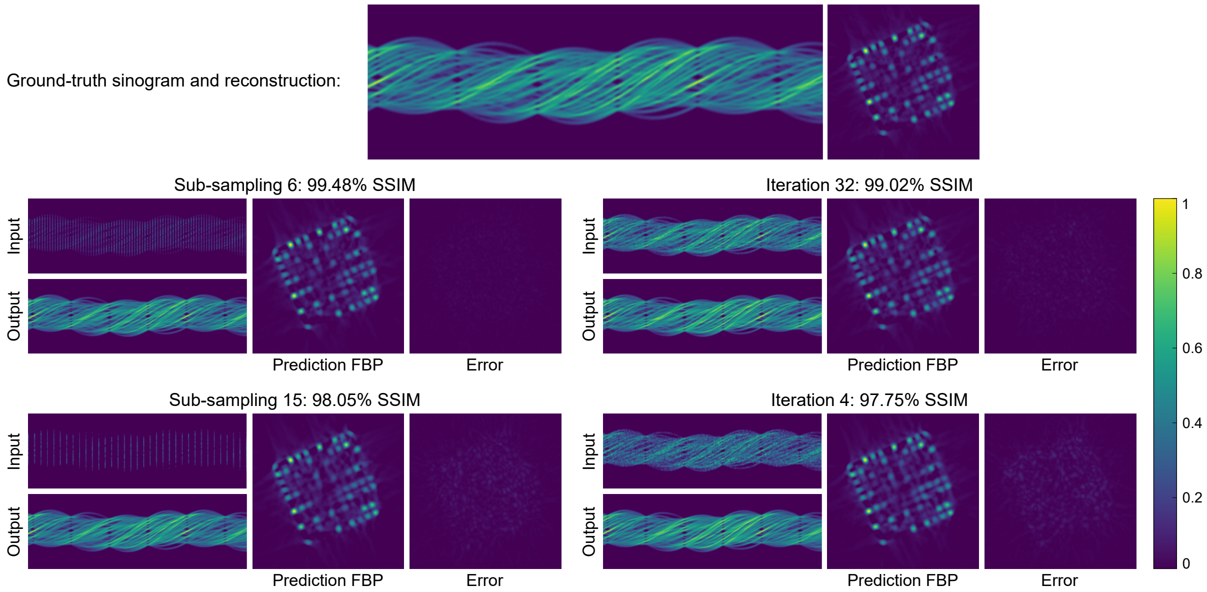

Figure 5: Examples of the outputs produced by our CNNs (bottom) compared to the ground truth data (top). The left images show two examples of CNNs trained to perform the de-noising from iterations 32 and 4 whilst the right images show two CNNs trained to work with partial views with sub-sampling factors of 6 and 15.

accurate. This means an increase of performance of three orders of magnitude with an accuracy loss of around 3%. If the target error were below 1%, the increase of performance would be of a factor of 12, for the limited view use-case, or of a factor of 32, for the de-noising use-case.

To further assess the degree of generalization of our technique and to validate the simulation results, we tested our CNN trained for limited views with a factor of 6 on real data samples from [23]. To do so, we created a synthetic problem by subsampling the original sinograms by a factor of 6 and compared the reconstructions of the outputs of our CNN wrt the ones for the full sinograms. The overall mean SSIM was of 95.31%, which is around 4 points below the accuracy we achieved with simulated data (99.55%). However, as Figure 6 shows, the final quality of the results remains extremely high and, considering that our simulations had no background noise, were performed in 2D, with monochromatic photons and with no detector response modelling, we conclude that the degree of resilience achieved by our models is promising. Previous works [**?**] have shown that fine-tuning with very small amounts of either, real or MCNP data, would significantly improve these results.

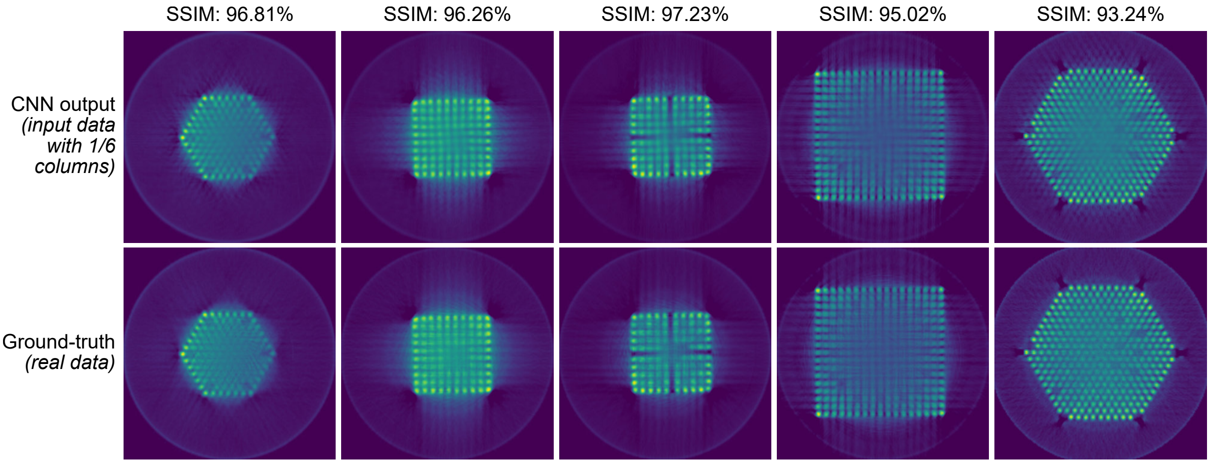

Figure 6: Examples of the outputs produced by our CNN with real data, considering the limited view problem with a subsampling factor of 6.

4 Conclusions and outlook

In this paper we have presented a novel technique that aims to generating and enhancing PGET data from the deep learning perspective. We have introduced a fast simulator that, by modelling photon-detector visibility from a probabilistic point of view, allows to generate Monte Carlo simulations with simplified physics in execution times below 3 seconds. Using these data, we have trained 14 different CNNs that solve the de-noising and limited view problems with different degrees of completeness. As results have shown, our technique is able to speed-up traditional Monte Carlo radiation transport simulations by several orders of magnitude with a limited impact in the accuracy of the final results. We have further tested our approach by applying it to real data without re-training the CNNs for it. Results have shown that the degree of generalization achieved using only synthetic data is promising, despite the limitations of our simulator.

Future works will address three main topics: (1) Further improvement of the PGET simulator to model, among others, the detector response, particle interactions in 3D and the energy tracking of the photons. (2) Fine-tuning our trained deep networks to improve their performance with measured or MCNP data. (3) Addressing practical use cases on top of the proposed architecture to face the verification tasks that nuclear inspectors perform on real inspections.

References

[1] Camille Bélanger-Champagne, Pauli Peura, Paula Eerola, Tapani Honkamaa, Timothy White, Mikhail Mayorov, and Peter Dendooven. Effect of gamma-ray energy on image quality in passive gamma emission tomography of spent nuclear fuel. In *IEEE Transactions on Nuclear Science*, volume 66, pages 487–496, 2019.

[2] Tapani Honkamaa, Ferenc Levai, Reinhard Berndt, Peter Schwalbach, Stefano Vaccaro, and Asko Turunen. A prototype for passive gamma emission tomography. In *2014 Symposium on International Safeguards*, page 281, Vienna (Austria), 10 2014.

[3] Timothy White, Mikhail Mayorov, Nikhil Deshmukh, Erin Miller, L. Eric Smith, Joakim Dahlberg, and Tapani Honkamaa. Spect reconstruction and analysis for the inspection of spent nuclear fuel. In *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–2, 2017.

[4] Timothy White, Alain Lebrun, Tapani Honkamaa, Mikhail Mayorov, Pauli Peura, Joakim Dahlberg, Jens Keubler, Victor Ivanov, and Asko Turunen. Application of passive gamma emission tomography (PGET) for the verification of spent nuclear fuel. In *59th Annual Conference of the Institute of Nuclear Materials Management*, Baltimore, MD, USA, 07 2018.

[5] Mikhail Mayorov, Timothy White, Alain Lebrun, Joerg Brutscher, Jens Keubler, André Birnbaum, Victor Ivanov, Tapani Honkamaa, Pauli Peura, and Joakim Dahlberg. Gamma emission tomography for the inspection of spent nuclear fuel. In *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–2, 2017.

[6] Jerry L. Prince and Jonathan M. Links. Medical imaging: Signals and systems (prince, j.l. and links, j.m.; 2006) [book review]. *IEEE Signal Processing Magazine*, 25(1):152–153, 2008.

[7] Rasmus Backholm, Tatiana A. Bubba, Camille Bélanger-Champagne, Tapio Helin, Peter Dendooven, and Samuli Siltanen. Simultaneous reconstruction of emission and attenuation in passive gamma emission tomography of spent nuclear fuel. *Inverse Problems and Imaging*, 14(2):317–337, 2020.

[8] Ming Fang, Yoann Altmann, Daniele Della Latta, Massimiliano Salvatori, and Angela Di Fulvio. Quantitative imaging and automated fuel pin identification for passive gamma emission tomography. *Scientific Reports*, 11(1), jan 2021.

[9] Ahmed Karam Eldaly, Ming Fang, Angela Di Fulvio, Stephen McLaughlin, Mike E. Davies, Yoann Altmann, and Yves Wiaux. Bayesian activity estimation and uncertainty quantification of spent nuclear fuel using passive gamma emission tomography. *Journal of Imaging*, 7(10), 2021.

[10] Anna Davour, Staffan Jacobsson Svärd, Peter Andersson, Sophie Grape, Scott Holcombe, Peter Jansson, and Mats Troeng. Applying image analysis techniques to tomographic images of irradiated nuclear fuel assemblies. *Annals of Nuclear Energy*, 96:223–229, 2016.

[11] Yannick Berker and Yusheng Li. Attenuation correction in emission tomography using the emission data—a review. *Medical Physics*, 43(2), 2 2016.

[12] L Smith, Staffan Jacobsson Svärd, Vladimir Mozin, Peter Jansson, Erin Miller, Tim White, Nikhil Deshmukh, Holly Trellue, Rick Wittman, Anna Davour, Sophie Grape, Peter Andersson, Stefano Vaccaro, Scott Holcombe, and Tapani Honkamaa. A viability study of

gamma emission tomography for spent fuel verification: Jnt 1955 phase i technical report. 10 2016.

[13] T. Goorley, M. James, Thomas Booth, F. Brown, J. Bull, Lawrence Cox, J. Durkee, J. Elson, Michael Fensin, R. Forster, J. Hendricks, H.G. Hughes, R. Jonns, B. Kiedrowski, Roger Martz, S. Mashnik, Gregg Mckinney, D. Pelowitz, R. Prael, and T. Zukaitis. Initial mcnp6 release overview - mcnp6 version 1.0. *Nucl. Technol.*, 164, 06 2013.

[14] Topias Kähkönen. Evaluating the viability of Serpent in Passive Gamma Emission Tomography (PGET) radiation transport simulations, 2021.

[15] Thomas E. Booth, R. Arthur Forster, and Roger L. Martz. Mcnp variance reduction developments in the 21st century. *Nuclear Technology*, 180(3):355–371, 2012.

[16] Steve Bako, Thijs Vogels, Brian Mcwilliams, Mark Meyer, Jan NováK, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4), jul 2017.

[17] Michael Mara, Morgan McGuire, Benedikt Bitterli, and Wojciech Jarosz. An efficient denoising algorithm for global illumination. In *Proceedings of High Performance Graphics*, New York, NY, USA, July 2017. ACM.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[19] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E. Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021.

[20] Fan Jia, Wing Hong Wong, and Tieyong Zeng. Ddunet: Dense dense u-net with applications in image denoising. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 354–364, 2021.

[21] Maximilian P. Reymann, Tobias Würfl, Philipp Ritt, Bernhard Stimpel, Michal Cachovan, A. Hans Vija, and Andreas Maier. U-net for spect image denoising. In *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–2, 2019.

[22] Umair Javaid, Kevin Souris, Damien Dasnoy-Sumell, Ana Barragán Montero, Sheng Huang, and John Lee. Denoising of monte carlo dose distributions using unet. *Medical Physics*, 46:E120–E120, 06 2019.

[23] International Atomic Energy Agency (IAEA). IAEA Tomography Reconstruction and Analysis Challenge, 2019. Available at https://ideas.unite.un.org/iaea-tomography/Page/Home. Accessed: 01-02-2023.

[24] Carlos Sanchez-Belenguer, Erik Wolfart, and Andrea Favalli. Deep learning driven analysis of gamma ray spectra of nuclear material. In *Proceedings of the INMM 63rd Annual Meeting*, 2022.

[25] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.