Artificial Intelligence for Monte Carlo Simulation in Medical Physics

David Sarrut1*, Ane Etxebeste1, Enrique Muñoz2, Nils Krah1,2 and Jean Michel Létang1

1University Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1294, Lyon, France, 2University of Lyon, Université Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, Villeurbanne, France

Monte Carlo simulation of particle tracking in matter is the reference simulation method in the field of medical physics. It is heavily used in various applications such as 1) patient dose distribution estimation in different therapy modalities (radiotherapy, protontherapy or ion therapy) or for radio-protection investigations of ionizing radiation-based imaging systems (CT, nuclear imaging), 2) development of numerous imaging detectors, in X-ray imaging (conventional CT, dual-energy, multi-spectral, phase contrast . . .), nuclear imaging (PET, SPECT, Compton Camera) or even advanced specific imaging methods such as proton/ion imaging, or prompt-gamma emission distribution estimation in hadrontherapy monitoring. Monte Carlo simulation is a key tool both in academic research labs as well as industrial research and development services. Because of the very nature of the Monte Carlo method, involving iterative and stochastic estimation of numerous probability density functions, the computation time is high. Despite the continuous and significant progress on computer hardware and the (relative) easiness of using code parallelisms, the computation time is still an issue for highly demanding and complex simulations. Hence, since decades, Variance Reduction Techniques have been proposed to accelerate the processes in a specific configuration. In this article, we review the recent use of Artificial Intelligence methods for Monte Carlo simulation in medical physics and their main associated challenges. In the first section, the main principles of some neural networks architectures such as Convolutional Neural Networks or Generative Adversarial Network are briefly described together with a literature review of their applications in the domain of medical physics Monte Carlo simulations. In particular, we will focus on dose estimation with convolutional neural networks, dose denoising from low statistics Monte Carlo simulations, detector modelling and event selection with neural networks, generative networks for source and phase space modelling. The expected interests of those approaches are discussed. In the second section, we focus on the current challenges that still arise in this promising field.

Keywords: AI, Monte Carlo simulation, medical physics, GAN, deep learning

1 INTRODUCTION

Techniques based on Deep Learning have seen huge interest for several years showing, in particular, significant progress in computer vision. Many medical applications have adopted them (see Shen et al. [132] for a recent review) and a lot of research is currently underway. These recent developments around Machine Learning in medical physics have found applications in the field
of Monte Carlo simulations. In this work, we will review and discuss the use of artificial intelligence, or more specifically machine Learning, for Monte Carlo simulation for particle transport especially in the context of medical physics. Links to other fields such as particle physics, nuclear physics or solid state physics also exist, but they would be beyond the scope of this work.

The article is structured in the following three parts: Sections 1.1 and 1.2 give a brief introduction of the principles of Monte Carlo simulation as well as deep learning, Section 2 presents a literature review in the context of medical physics, and Section 3 discusses on current challenges.

### 1.1 MONTE CARLO MODELLING IN MEDICAL PHYSICS

Monte Carlo codes in medical physics are similar to those used in high energy physics community (HEP). Specifically, the simulation engine simulates the transport of particles, mainly photons and light charged particles, across a set of geometrical objects made of well-defined materials, modeling the physical interactions between particles and matter. The transport is performed particle by particle in a step-by-step fashion. For all particles, at each step, stochastic models describing physical interactions (such as Compton scattering, photo-electric effect, ionisation, etc) are repeatedly evaluated based on databases of cross-sections. Thanks to this approach, the quantities determined via simulation, for example the absorbed dose distribution or the number of detected particles, are very accurately estimated even in complex geometries. Depending on the complexity of the simulated configuration, millions or even billions of particles should be tracked to reach an acceptable statistical convergence, making the whole process usually very long.

The use of Monte Carlo techniques for medical physics started to become increasingly popular in the late 1970s, in particular for the modelling of imaging systems in nuclear medicine, for the characterization of particle beam accelerators in radiotherapy, and for calculating the absorbed dose in patients for planning treatment [5, 14, 115, 130]. Since then, Monte Carlo simulations have become a widespread tool in research and development (R&D) for the design of nuclear imaging systems and dose calculation engines in treatment planning systems (TPS) [19, 41, 129, 139, 148].

An example of system development where Monte Carlo simulations are involved in the design, the control and test of the devices, and the tuning of reconstruction algorithms is the new generation of whole-body PET scanners. Prototypes currently under development include the EXPLORER [10] at UC Davis (United States), the PennPET [64] in Philadelphia (United States), the PET20.0 [146] in Ghent (Belgium) and J-PET [72] in Krakow (Poland). With regard to TPS, Monte Carlo simulations are often necessary to characterize the beam lines and the resulting particle phase spaces (photons or charged particles), to determine the dose point-kernels in analytical dose engines, or to directly calculate the absorbed dose in patients [129, 148]. The great accuracy of Monte Carlo calculations is particularly crucial for new radiotherapy protocols, such as hypo-fractionation [95], “flash” radiotherapy [74], and hadrontherapy (proton or ions [13, 50]) which involve very high dose rates and a high spatial conformation.

R&D activities in the field of Monte Carlo simulations have resulted in the development of generic computer codes, i.e. which allow the user to simulate a wide range of particles, energies, geometrical elements and physical interactions (EGSnrc [65], MCNPX [55], Penelope [120], Fluka [18], Geant4 [1, 4], Gate [57, 123, 124], etc.). The accuracy of the underlying physical models and cross-section databases has continuously been improved, also thanks to new experimental data. To counter the low efficiency of Monte Carlo simulation techniques, variance reduction techniques (VRT) [129, 139] have been developed and continue to be proposed to speed up computation times at a given precision.

The development of increasingly sophisticated acquisition systems and finer representations of patient data requires a complex modelling, costly in computer resources. Monte Carlo codes dedicated to a specific application also exist and usually offer a better computational performance than generic codes, but they are in general very restricted to the targeted applications. Most research teams rely on the latter for their work.

Monte Carlo simulation is inherently parallel because particle histories are treated as independent from each other. This is a major advantage for accelerating computations [148]. Powerful computing infrastructures (clusters) can thus be used by researchers to obtain Monte Carlo simulation results in an acceptable time. The recent enthusiasm for scientific computing on graphics cards (GPU) has also concerned several Monte Carlo developments [16, 45, 89, 118], but the codes ported to GPUs tend to be difficult to maintain and partly lose in generality by limiting themselves to well-defined applications.

### 1.2 Deep Learning Principles

Deep learning [17, 47, 76, 128] is a machine learning method performing supervised, non-supervised or semi-supervised learning tasks, in which the learning takes place across many different stages, as for example defined in [128]. It is most commonly accomplished using neural networks. A neural network is composed of connected neurons typically (but not necessarily) organized in layers. Connections between neurons have associated weights and a neuron is associated with an activation function which generates the neuron’s output, e.g. a non-linear function mapping from an open into a closed real domain (e.g., values bounded between zero and one). The input to a neuron’s activation function is the weighted sum over the outputs of all the connected neurons (belonging to the previous layer in a fully connected feedforward net), generating a complex mapping between the network’s inputs and outputs. In \( \text{Eq. 1} \), \( \mathbf{x}^{(i)} \), \( \mathbf{W}^{(i)} \) and \( b^{(i)} \) represent respectively the output, the weights matrix and the bias for the layer \( i \), whereas \( f \) stands for the activation function which is applied element-wisely. The number of neurons, the way they are connected (layers), the choice of activation functions and other parameters...
are referred to as the “network architecture”. The weights’ values of the connections are parameters that will be determined during the training phase.

\[ x^{(i+1)} = f(W^{(i)}x^{(i)} + b^{(i)}) \]  

(1)

Indeed, training the model means optimizing a value for every weight in order to adapt the network to handle a task. This learning process uses a training dataset as input which, for supervised learning, groups pairs of input-output samples. Optimizing the network is typically performed by stochastic gradient descent where weights are updated using backpropagation (in a feedforward network) that computes the gradient of a loss function with respect to the weights of the network. The loss function is chosen depending on the problem at hand, for example to quantify how well the current model prediction matches the training dataset or, indirectly, to measure a distance between current and expected distribution.

Convolutional neural networks [73, 77, 156] is a famous approach to deal with high dimension input data such as images. They are regularized versions of (fully connected) networks based on convolution kernels that slide along input features and provide activation when some specific type of feature is detected at some spatial position in the input image. Hence, shared weights and local connections allow reducing the number of parameters and can thereby simplify the training process, improve generalisation, and reduce overfitting. A CNN architecture is composed of several building blocks (convolution layer, pooling layer, fuller connected layer, activation function, loss function, etc) that must be selected and put together into a network for each task.

Generative Adversarial Networks (GANs) are special deep neural network architectures recently reported [48] that, once trained, can be used to generate data with similar statistics as the training set. A GAN consists of two models that are simultaneously trained: a generative model \( G \) that aims to generate a targeted data distribution, and a discriminative model \( D \) that estimates the probability that a sample came from the training data rather than from the generative model. The discriminator \( D \) is trained to maximize the probability of correctly identifying samples from the training data as real and those generated by \( G \) as fake. The generator \( G \) is trained to produce data samples distributed similarly to the data distribution. Once trained, the resulting \( G \) model is able produce set of samples that are supposed to belong to the underlying probability distribution of the targeted data represented by the training dataset. A review can, for example, be found in [31]. This type of architecture is frequently used in multiple applications, in particular in the synthesis of photorealistic images or, for example in the medical physics field, to generate synthetic CT from MR images [80]. In the field of Monte Carlo particle tracking simulations in medical physics, several works have been proposed and will be discussed in the next sections.

### 2 LITERATURE REVIEW

Within the High Energy Physics (HEP) community, a lot of effort has already been made to improve and accelerate Monte Carlo simulation with the help of Machine Learning (including Deep learning) for various applications, in particular around the Geant4 code [44]. Among various examples: simulation of particle showers [108], modelling the response of detectors [144], pairs of jet simulation at LHC [39], nuclear interaction modelling [30], condensed matter physics [25, 133], etc. Interested readers may, for example, refer to several reviews [2, 3, 22, 51, 114] or to https://iml-wg.github.io/HEPML-LivingReview.

To our knowledge, no review has been proposed for the medical physics field. In the following sections, we thus review works which combine machine learning with Monte Carlo simulations in the medical physics field. Of course, particle transport simulation via Monte Carlo in HEP and medical physics share many similarities. Exchange among researchers working in these different fields would be desirable in order to share new knowledge and discoveries. Some of the works reviewed in the following deal with input data that are not an image, but are related to sets of particle properties. Table 1 summarizes the type of input that is considered for each application. The motivation behind many of the presented works is to speed up the computation, e.g. dose calculations or image reconstructions, to the order of minutes rather than hours or days. Other motivation is to improve detector quality by better event selection or reconstruction.

### TABLE 1 | AI-based applications related to Monte Carlo simulations and their corresponding input data type. The word “particles” as input type refers to a vector of particle properties such as energy, position, direction, weight, etc. CNN stands for convolutional neural networks and MLP stands for multi-layer perceptron.

<table>
<thead>
<tr>
<th>Application</th>
<th>Input type</th>
<th>Refs (among others)</th>
<th>Main ML types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECT scan-time reduction</td>
<td>image</td>
<td>[82, 119, 121]</td>
<td>CNN, U-net</td>
</tr>
<tr>
<td>CBCT scatter modelling</td>
<td>image</td>
<td>[27, 58, 60, 75, 79, 84, 87, 88, 140, 145, 152, 155]</td>
<td>CNN, U-net</td>
</tr>
<tr>
<td>PET attenuation/scatter correction</td>
<td>image</td>
<td>[6, 9]</td>
<td>CNN, U-net</td>
</tr>
<tr>
<td>Detector response modelling</td>
<td>particles</td>
<td>[126, 144]</td>
<td>GAN, MLP</td>
</tr>
<tr>
<td>Source + phase space modelling</td>
<td>particles</td>
<td>[108, 125, 127]</td>
<td>GAN</td>
</tr>
<tr>
<td>Event selection</td>
<td>particles</td>
<td>[8, 12, 40, 46, 93, 98, 100, 102, 107, 157]</td>
<td>MLP, CNN</td>
</tr>
<tr>
<td>Interaction position in scintillators</td>
<td>various</td>
<td>[23, 33, 97, 99, 109, 110, 122, 150, 154]</td>
<td>MLP, CNN</td>
</tr>
</tbody>
</table>

1http://hdl.handle.net/11603/19255
2http://hdl.handle.net/2078.1/thesis/14550
2.1 AI-Based Dose Computation

Different studies have used Monte Carlo simulations and CNNs to estimate the dose distribution in imaging and radiotherapy. The general idea is to develop a fast neural network as an alternative to computationally intensive simulations. Typically, dose distributions computed with Monte Carlo simulations are used to generate large training and validation sets from CT images and treatment plans. For example, Lee et al. [79] proposed deep learning-based methods to estimate the absorbed dose distribution for internal radiation therapy treatments, i.e., where the radiation source is a radionuclide injected into the patient. A CNN was trained from PET and CT image patches associated with their corresponding dose distributions computed by GATE simulation [124] and considered as ground truth. The training database was composed of 10 patients with eight PET/CT timepoints after intravenous injection of 68Ga-NOTA-RGD, from 1 to 62 min post-injection. The network architecture was based on a U-net structure [116]. The first part of the network performed image downsampling operations (contracting path) and the second part, image upsampling (expansive path). The U-net considers both PET and CT as input data to predict the dose. It was operated on image patches rather than on full images because, in the studied scenario, the dose is mainly deposited locally (millimeters) around the source voxels, allowing memory and computation time gain. Note that local dose deposit would not be a valid assumption in case of radiation with larger range (high energy photons for example). The voxel dose rate errors between CNN-estimated and Monte Carlo-estimated dose were found to be less than 3% and was obtained within a few minutes compared to hours with Monte Carlo. Similarly, Götz [49] presented a hybrid method combining a U-net with empirical mode decomposition technique. The method takes as input CT images and corresponding absorbed dose maps estimated with the MIRD protocol (organ S-value [21]) from SPECT images for 177Lu internal radiation therapy treatment. Again, results seem very good, better than the fast Dose-Volume-Kernel (DVK) method [20] and faster than Monte Carlo.

Principles relatively similar to those developed in the previous examples related to internal radiation therapy have also been applied to external radiation therapy, i.e., where the radiation source is an external particle beam generated by an accelerator. Kalantzis et al. [63] performed a feasibility study of a multi-layer perceptron (MLP) to convert a 2D fluence map obtained from an electronic portal imaging device (EPID) to a dose map for IMRT, replicating conventional convolution kernel in TPS. Nguyen et al. [104] proposed to perform 3D radiotherapy dose prediction for head and neck cancer patients with a hierarchical densely connected U-net deep learning architecture, with prediction error lower than 10%. Liu et al. [85] developed a deep learning method for prediction of 3D dose distribution of helical tomotherapy for nasopharyngeal cancer leading to less than 5% prediction error.

Other developments for imaging dose and brachytherapy have been proposed. For example, Roser et al. [117] use a U-Net fed with first order fluence maps computed by fast ray-casting in order to estimate the total dose exposure including scattered radiation during image-guided x-ray procedures. The CNN was trained using smoothed results of MC simulations as output and ray-casting simulations of identical imaging settings and patient models as inputs. As a result, the proposed CNN estimated the skin dose with an error of below 10% for the majority of test cases. The authors conclude by stating that the combination of CNN and MC simulation has the potential to decrease the computational complexity of accurate skin dose estimation. As an example in brachytherapy, Mao et al. [90] investigated a CNN-based dose prediction models, using structure contours, prescription and delivered doses as training data, for prostate patients and cervical patients. Predictions were found to be very close to those from MC, with less than few percent differences for various dosimetry indexes (CTV).

At the current stage, it is unlikely that dose distributions predicted via DL will be used as the main dose computation method in clinical practice because the dose is expected to be estimated from physically plausible effects and modeling and not really by learning processes. Nevertheless, it may be useful for plan checking consistency, fast plan comparison or to guide plan optimization.

2.2 Deep Learning Based Monte Carlo Denoising

Instead of mapping from some kind of image data (e.g., patient CT, SPECT image) to a dose distribution, deep learning methods have also been developed as a post-processing step to Monte Carlo dose computations to reduce the noise in dose maps due to inherent statistical fluctuations in the deposited dose per voxel. Indeed, Monte Carlo denoising methods have been studied for a long time and have shown to be able to reduce computation time by smoothing statistical fluctuations [101]. The noise of the Monte Carlo computed dose is related to the variance on the deposited energy and decreases as the number of simulated particles, $N$, increases, specifically at a $1/N$ rate. Hence, a large number of iterations is required to reach low fluctuation dose estimation, in particular in low dose regions where $N$ is small. For simulation of detectors such as CT or PET/SPECT images, noise is generally considered as Poisson noise. Lowering the number of simulated particles translates into a net gain in computation speed. Several filtering methods [35, 36, 42, 66, 92] (among others) have been employed in denoising, such as 3D wavelet-based, advanced mean-median filtering, anisotropic diffusion and so on. In general, the methods manage to reduce dose fluctuations while preserving mean dose, but the effective acceleration depends significantly on the characteristics of the dose distribution [101].

The principle of CNN-based denoising is to feed a network with pairs of high-noise/low-noise dose distributions obtained from low and high statistics Monte Carlo simulations with the goal to generate low-noise dose maps from noisy ones. In many cases, the CNN architecture is derived from U-Net, but other architectures such as Dense-Net [54] or Conveying-Path Convolutional Encoder-decoder (CPCE [131]) were studied as well. CNN-based denoising has been applied to photon [43, 71, 103, 111] and proton dose [59] for various indications including

---

1http://hdl.handle.net/11603/19255
brain, head and neck, liver, lung, prostate, and to dose delocalization due to charged particles within MRI (in magnetic resonance-guided radiation therapy [29]). Evaluations were performed based on peak signal-to-noise ratio (PSNR), Dose Volume Histogram (DVH [53]) or gamma index ([81, 86]) as comparison metrics. Results were generally very encouraging. The CNN produced noise equivalent dose maps with approximately 10–100 times fewer particles than originally needed [11, 153]. Some difficulties remain: results depend on the size and complexity of the training datasets and it is to be seen how the method can be generalised to other datasets, e.g., how well does a network which was mainly trained on head and neck patients perform on prostate patients. Furthermore, denoised dose maps must preserve dose gradients and it is not yet fully clear how to guarantee this.

In SPECT and PET imaging, the image noise is (partly) related to the scan-time duration. Reducing the scan-time directly improves the clinical workflow and decreases involuntary motion during scanning on the one hand, but on the other hand increases image noise. Different denoising approaches based on DL have been proposed such as [32, 82, 121]. In particular, Ryden et al. [119] proposed an approach based on sparse projection data sampling where intermediate projections were interpolated using a deep CNN to avoid image degradation. DL-based denoising methods were also investigated for low dose CT imaging [111, 131, 151, 153]. More generally, deep learning image denoising methods may be a source of inspiration in this field [142].

### 2.3 AI for Modelling Scatter

DL-based methods have also been applied to cone-beam CT (CBCT) imaging. The main issues to be addressed in this modality are the poor image quality and the artefacts due to scatter. These arise because the imager panel, which is two-dimensional without anti-scatter grid, not only captures the attenuated primary photons from the x-ray source, but also those originating from coherent and incoherent scatter within the patient. For accurate image reconstruction, the scatter contribution would need to be known and subtracted from the raw projection images. In practice, this is impossible because the imager panel only provides a non-discriminative integrated intensity signal. A Monte Carlo simulation, on the other hand, can specifically tag scattered photons so that perfect scatter-free projections can be obtained via simulation. In fact, some earlier works on CBCT scatter correction rely on Monte Carlo simulation to estimate the scatter contribution in raw projection [58]. However, the direct Monte Carlo simulation of kV photons is too slow to be integrated into a clinical image reconstruction software, although heavy use of variance reduction techniques might improve this [88].

Recent works propose to use deep convolutional networks which learn from CBCT projections simulated via Monte Carlo. They generate estimated scatter images (projections) as output based on raw projections as input [75, 78, 87, 145]. Once trained, the network can replace the Monte Carlo simulation and be used as scatter estimator within the image reconstruction workflow. The technical details of the networks vary, but all report promising results with significant higher CNR (Contrast to Noise Ratio) compared to previous heuristic methods. It is worth mentioning that these methods rely on Monte Carlo simulations for training where primary photons can be distinguished from scattered ones and could not be easily trained on experimentally acquired projections which cannot directly provide explicit scatter images to learn from.

Other authors have reported CBCT scatter correction methods based on deep learning which operate in the image domain [27, 60, 84, 140, 152, 155]. More specifically, they take a CBCT image as input and generate a synthetic CT image as output, i.e. they estimate how a CT image of the patient anatomy described by the CBCT image would have looked like. These synthetic CT images seem to contain much fewer artefacts than the original CBCT images.

Attenuation and scatter correction in the image domain for PET imaging has also been proposed using deep convolutional neural networks [6, 97]. Datasets to train the networks consisted in experimentally acquired images, but in principle, these image-based scatter correction studies would also work on Monte Carlo generated data that may help to create large databases.

### 2.4 AI for Modelling Imaging Detector Response

The works presented so far rely on the output of Monte Carlo simulations, but they do not alter the simulation itself. The works in this section, on the other hand, replace part of a Monte Carlo simulation in an attempt to accelerate it. More specifically, the proposed works model the particle transport through part of the geometrical components implemented in the simulation. In contrast to the previous methods, the model's input and output are not necessarily images, but may be sets of particle properties (energy, position, etc).

To our knowledge, few works have been published on this topic in the medical physics field. One example was recently proposed to speed up simulation by modelling the response of a detector: instead of explicitly simulating the particle transport in the detector, this is emulated by the network. For example, one idea could be to speed-up simulations of SPECT imaging by modelling the collimator-detector response function (CDRF) that combines the cumulative effects of all interactions in the imaging head and may be approximated with Angular Response Functions (ARF) [38, 119, 126, 135]. In [126], the tabulated model of the CDRF has been replaced by a deep neural network trained to learn ARF of a collimator-detector system. The method has been shown to be efficient and to provide variance reduction that speeds up the simulation. Speed-up compared to pure Monte-Carlo was between 10 and 3,000: ARF methods are more efficient for low count areas (speed-up of 1,000–3,000) than for high count areas (speedup of 20–300) and more efficient for high energy radionuclides (such as 131I) that show large collimator penetration.

### 2.5 AI for Monte Carlo Source Modelling

Recent works in the medical physics field have explored the use of generative networks, GANs in particular, to model particle source
distributions and potentially speed up Monte Carlo simulations [125, 127]. In the proposed methods, the training data set is a phase space file generated by an analog MC simulation which contains properties (energy, position, direction) of all particles reaching a specific surface. Once the GAN is trained, the resulting network G acts as a compact and fast phase space generator for the MC simulations, replacing a large file of several gigabytes by a NN (G) of several megabytes. G has the ability to quickly generate a large number of particles which allows the user to speed-up the simulations significantly (up to a few orders of magnitude depending on the simulation configuration). In the first case [127], the GAN method was used to learn the distribution of particles exiting a the nozzle of a therapeutic linear electron accelerator (linac), and to model a brachytherapy treatment where the network learned the source distribution generated by seeds in the prostate region. Simulations performed with the GAN as a phase space generator showed a very good dosimetric accuracy compared to the real phase space. In the second case [125], the authors proposed to apply this approach to a more complex particle distribution, namely that of particles exiting a patient in a SPECT acquisition. Results showed that images of complex sources with low error compared to the reference image reconstructed from real phase space data were feasible.

It should be mentioned, although beyond the scope of this review, that several works in the HEP community have also shown how generative models may be very useful to model high-dimensional distributions. Among others, Paganini et al. [108] proposed a GAN model to simulate computer intensive electromagnetic showers in a multi-layer calorimeter, and de Oliveira et al. [34] also explored GAN to produce jet images (2D representations of energy depositions from particles interacting with a calorimeter). Both methods report large computational speedups compared to conventional Monte Carlo simulations.

2.6 Deep Learning in Nuclear Imaging

DL has also been explored in the context of nuclear imaging (PET, SPECT, Compton, etc.) - a field where Monte Carlo simulation plays a vital role in designing and validating imaging systems and reconstruction algorithms. Many of the proposed DL methods focus on post processing steps of raw data acquired by the imaging system which impact image quality.

In PET, for example, NNs have been investigated to identify random data points arising from annihilation events which lead to image noise [107] and for the correct sequence identification of PET events with multiple interactions of an annihilation photon in several detector elements, in which the first interaction position must be identified in order to recover the actual line of response [93, 102]. NNs have also been used to estimate the two-dimensional interaction position in the monolithic scintillator crystal in PET imagers, or a three-dimensional position when the depth of interaction (DoI) is estimated as well. The investigated NNs have yielded results with better spatial resolution [37, 122], higher uniformity across the crystal volume [98] or faster implementation [149] compared to other existing methods (e.g. maximum likelihood [112] or nearest neighbours [141], among many others).

In Compton imaging devices, ML has been investigated for sequence ordering of multiple-interaction events [157] and for signal and background discrimination of Compton camera data in the context of prompt gamma imaging [100]. It is also worth mentioning that DL-based methods have been studied for event selection in data measured by radiation detectors, in particular in HEP, as shown for example in [51]. Applications include detectors at the LHC [12], neutrino-dedicated detectors [8, 40] or measurement of gamma-rays in astrophysics [46]. We refer the reader to [2, 24, 51] for an overview in HEP field.

3 CURRENT CHALLENGES

Monte Carlo based particle transport codes are a central tool for many research questions and applications. This is certainly true for medical physics, the area we concentrate on here, as well as for other fields. We have shown in the previous sections that deep learning methods can be useful for various tasks during simulations, in particular to reduce the computation time (denoising, scatter modelling), but also to model complex systems (detector, source modelling) or perform advanced event selection. However, before those methods can replace conventional methods, especially in industrial or clinical settings, several challenges must be addressed.

Conventional methods in the context of Monte Carlo particle transport simulations are usually instructed by knowledge about the underlying physics processes. This results in specific mathematical or statistical models, usually containing parameters to be adjusted, e.g. based on a calibration or reference measurements. Neural networks, on the other hand, are effectively physics or model agnostic and simply learn properties from a given training dataset. Therefore, there is no a priori guarantee that a trained NN provides a plausible representation of the physics underlying the learned processes. At the same time, there are usually quantitative requirements associated with MC simulation tasks, e.g. accurate estimate of deposited energy or dose, accurate estimated particle properties, accurate phase space distributions. All of the following challenges are linked to the requirement that DL methods in MC simulations be reproducibly accurate and that accuracy can be evaluated.

Challenge 1: Quality of Data

One conventional challenge in DL is related to the limited database size, or its limited variability and adequacy to the learning process. There are several pitfalls such as: non homogeneous data, difficult data curation, insufficient representativity, etc. However, when learning from Monte Carlo data, the size of the training dataset could, in principle, be only limited by the computation time. As the latter can quickly become prohibitive, data augmentation may be used anyway. When learning from MC data, the quality of the learnt process becomes strictly tightened to the quality of the simulation itself. If the modelled system contains error or bias, it will be present in the training dataset and learnt by the neural network. Simulation results
must therefore be properly validated to avoid bias (see next section) and comparison with experimental data, if feasible, is required.

**Challenge 2: Performance Metric and Uncertainty**

Evaluation of a trained neural network is conventionally performed by splitting the dataset into three separated parts for training, validation, and test. The model is first trained on the training dataset in order to optimize weights values according to the loss function. The validation dataset is successively used to provide unbiased validation of the final model during the training in order to tune hyperparameters (e.g., number of layers, number of epoch) and prevent overfitting (when the loss function still decreases in the training dataset, but starts to increase in the validation set). The test dataset provides final unbiased validation. The validation process in the context of Monte Carlo simulations may be different from traditional computer vision applications (photos, cinema, games etc) where visual perception is assessed. Here, quantitative validation of physical quantities is needed. The figures of merit to evaluate usually depend on the kind of application for which the network is developed. For example, for dose computation, standard criteria such as the “gamma index” \[81, 86\] or Dose Volume Histograms \[53\] could be used. It is to be explored how (clinically) relevant metrics and tolerance levels might be incorporated into the training and validation process. Ideally, final validation of a network should not only be performed against simulated data, but also against experimental data. Furthermore, collaborative open datasets \[26\] or challenges (such as \[113\]), yet specific to medical physics applications, may be useful.

One of the advantages of Monte Carlo simulations is that they are able to easily associate an (uncertainty) error with the simulated data. The MC statistical uncertainty (e.g. \[28\]) could hence be used to provide a target tolerance. In the context of medical physics, the uncertainty of data produced by generative networks needs to be carefully studied and understood, especially if those networks are to (partly) replace conventional MC simulations. In more practical terms, what are the noise properties of DL-generated images or dose distributions compared to their MC simulated counterparts? Two forms of uncertainty have been proposed which are referred to as *epistemic* and *aleatoric* \[69, 91\], where epistemic is the reducible (related to the lack of training data in certain areas of the input domain) and aleatoric the irreducible part of uncertainty (dealing with the potential intrinsic randomness of the real data generating process). As an example, approaches based on Bayesian neural networks \[70, 106\], by its ability to give an estimation of the uncertainty, may be an interesting lead.

**Challenge 3: Neural Network Architecture and Hyperparameters**

A challenge when working with deep neural networks is to select the appropriate network structure and capacity, i.e., number of neurons, number of layers, type of activation, etc., for a given problem and adjust the training process with appropriate hyperparameter values. Underfitting may occur if the model is too simple (not enough capacity) or too much regularized, so it tends to have poor predictive performance. Overfitting can be an issue which occurs e.g. when a too flexible network (too many parameters) learns structure in the data which merely derives from noise or other artefacts rather than true information. Several regularization methods, such as adding a penalty term on the weights (e.g., L1, L2) in the loss, or using dropout regularization \[138\], that randomly ignores some layer outputs, might help preventing overfitting issues and improving model generalization (capacity to perform on inputs not previously seen by the network).

When images are the input, convolution operations can be added in between network layers. Moreover, several architectures such as the well known UNet \[116\] or pix2pix conditional adversarial networks \[56\] (among others) have been proposed. When a network is used to bypass the Monte Carlo simulation and use as input an image, e.g. a patient CT or a PET image, conventional convolution filters can be applied. However, when particle properties are the input, the nature of each entry in the property vector may be different (position, energy, time, etc.). It remains to be studied how meaningful convolution operations can be defined on such an inhomogeneous input space, or if it may be applied only partly, e.g. on a single dimension such as energies. Furthermore, some particle properties such as charge or atomic weight are bound to be integer number rather than reals and may require specific processes, such as one-hot encoding as used for example in \[126\]. Finally, conservation laws or other physical principles might pose constraints which need to be built into the network optimisation, either by a specific architecture or an adapted loss function.

**Challenge 4: Generative Models, Generative Adversarial Network**

To simulate particle transport through a medium, a Monte Carlo code must generate particles according to some probability distribution. This can be the initial phase space distribution of the particle source, but also an intermediate step which creates new particles as a result of interactions with the target, e.g. inelastic nuclear scattering. In conventional Monte Carlo methods, this is done by sampling from a cumulative probability distribution and the accuracy with which the distribution is modelled and parametrized directly impacts the accuracy of the simulation results. Source particles can also be sampled explicitly from tabulated phase space files. In this context, generative models represent a new way to replace conventional particle generation methods in a Monte Carlo simulation. Understanding and mastering the technical aspects of such methods represents an important challenge.

Our review concentrated on GAN (Section 2.5) which have been explored for Monte Carlo simulation in medical physics. Many variants of GAN have been proposed to improve performance or to adapt to various applications (auxiliary GAN, bidirectional GAN, conditional GAN, Cycle GAN, InfoGAN, etc). Despite a large number of successful results, GAN have been shown to be notoriously difficult to train, suffering from several pitfalls: mode collapse, vanishing...
gradient, instability. To tackle these issues, various formulations based on different metrics, such as the Wasserstein distance [7] and regularization methods [52, 61, 62, 94] have been proposed. An in-depth study of the most suitable variants for Monte Carlo simulations remains to be undertaken. For example, is it possible to obtain a precise and reliable modelling of all spatial characteristics of the dose distributions [67]? Can a GAN model a Linac gamma source precisely enough to include 511 keV peak [127]? Alternative generative learning processes, such as VAE (Variational AutoEncoders, see for example [68]) or, recently, Score-Based diffusion Generative Models [105, 136, 137, 143] may have a larger role to play in distribution modelling within Monte Carlo. In particular, VAE networks are designed to compress the input information into a constrained multivariate latent distribution (encoding) to reconstruct it as accurately as possible (decoding). Although VAEs seem generally less efficient than GAN in the field of photo-realistic image synthesis, it could be an interesting alternative to GAN in the medical physics field. Additionally, transfer learning may also be of interest where a model already trained on a given dataset may be adapted through training on another dataset.

The problem of reproducing a probability distribution by generative networks such as GAN arises far beyond the simple source modelling. In Monte Carlo simulations, certain interactions between particles, in particular nuclear processes, are based on very elaborate statistical distributions which require a lot of computing time, and generative networks would have a role to play. For instance, Bayesian neural networks have been proposed to improve mass predictions of nuclear models [106] and quantify the prediction uncertainty which becomes larger when the network is extrapolated away from the training region.

Finally, it is interesting to observe some subtle differences between GAN in computer vision and for tasks such as particle generation in a Monte Carlo simulation. In a computer vision application where a GAN generates images, it is mainly of interest that each image be as realistic as possible. In a Monte Carlo simulation, any generated particles must be true to nature and the physical properties of the beam must be respected. What really counts is whether the distribution of many generated particles is correct. The corresponding question in the computer vision application would be e.g., whether the GAN generates the correct proportion of long-haired brown dogs compared to short-haired black ones, albeit all of them individually might be realistic. In more technical terms, an image has a much higher dimension, i.e., number of pixels, than the vector of physical properties describing a particle. Out of the space of all images (including images with random noise), only a very small and sparse subspace contains realistic images, i.e., containing pixels which depict a desired kind of object. A particle distribution instead densely fills a relatively large portion of the full phase space. These differences likely impact the way GANs and other generative models perform in Monte Carlo simulations as opposed to computer vision tasks and will deserve more detailed attention.

**Challenge 5: Explicability and Interpretability**

Deep neural networks are sometimes criticized as being black boxes, or in other words for not providing direct insight into the way they link input and output. As an example: when modelling the response properties of a detector explicitly via a physics-motivated analytical model, the mathematical form of the model together with its parameters inform the user directly which kind of events will be detected in which fashion. In contrast, a deep neural network trained on Monte Carlo simulated data does not offer this transparency. The underlying reason is that a neural network is a highly flexible non-linear function whose parameters are the neuron weights optimized to best represent the training data. As the weights have no a priori meaning attached to them, they are difficult to interpret.

Monte Carlo simulation, on the other hand, is based on physics models with meaningful parameters and a thereby described quantitative relationship between input and output. Clearly, the randomized and iterative evaluation of a multitude of physics models makes the final simulation output complex in certain cases, but the underlying mechanism remains explicitly defined. A challenge when using deep neural networks in the context of Monte Carlo simulations is therefore to gain insight in and control over the workings of the network. This leads to the concepts of interpretation and explanation.

Definitions of these terms can be found in [96], namely, an interpretation is the mapping of an abstract concept into a domain that the human can make sense of. An explanation is the collection of features of the interpretable domain that have contributed for a given example to produce an output. It is important to note that both terms apply to trained networks. Picking up the example of the detector response (Section 2.4), an interpretation links a specific detector response, e.g. the detection window in a SPECT imager, to the particle properties, i.e. its energy, direction etc. In this same example, the explanation is the collection of properties which have led a specific particle to be associated with a certain detector response? In this sense, explanation and interpretation are expected to aid with the validation of deep neural networks in terms of physical plausibility.

The difficulty of visualizing and studying explanation and interpretation of a network grows with the dimension of the input data. When the input is merely a vector with a particle’s kinematic properties, i.e. with six or seven entries, the relevance of each of them for a given network decision can still be interpreted “manually”. For high dimensional input such as CT images, other methods must be employed, e.g. activation maximization with an expert [15, 134]. For interpretation, gradient based methods such as deep Taylor expansion and backward propagation techniques such as layer-wise relevance propagation should be mentioned here [9].

A rich literature about machine learning interpretation methods exists [83], with a large part of the methods exploiting the gradient information flowing through the layers of the network in order to highlight their impact. Investigating and developing interpretation and explanation techniques in the context of Monte Carlo simulations to make DNN sufficiently “transparent” will be one of the challenges to address.

### 4 CONCLUDING REMARK

There may be a methodological change associated with the use of deep learning methods in medical physics simulation: to some extent, instead of mathematically mastering the phenomenon under...
investigation, the modelling relies on a large amount of data to learn from heuristically. However, the Monte Carlo simulation which generates the training data needs to be skillfully set-up and evaluated in the first place. For the moment, even if it is envisioned that deep learning can improve simulations, it does not seem certain that it can always replace Monte Carlo. As the use of deep learning methods evolves, physics-driven dataset modelling, i.e., a mix between modelling based on large datasets and understanding of the underlying physics, will become increasingly important.

**DATA AVAILABILITY STATEMENT**

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/OpenGATE/Gate.

**REFERENCES**


**AUTHOR CONTRIBUTIONS**

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

**FUNDING**

This work was performed within the framework of the SIRIC LYriCAN Grant INCa-INSERM-DGOS-12563, the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007), the MOCAMED project (ANR-20-CE45-0025) and the POPEYE ERA PerMed 2019 project (ANR-19-PERM-0007–04).

Frontiers in Physics | www.frontiersin.org 9 October 2021 | Volume 9 | Article 738112

Sarrut et al. AI for Monte Carlo Simulation
83. Linardatos P, Papastefanopoulos V, and Kotsiantis S. Explainable AI: A
80. Lei Y, Harms J, Wang T, Liu Y, Shu HK, Jani AB, et al. MRI-only Based
78. Lee H, and Lee J. A Deep Learning-Based Scatter Correction of Simulated
73. Krizhevsky A, Sutskever I, and Hinton GE. Advances in Neural Information
70. Lalone D, Winey B, Verburg I, Pagannetti H, and Sharp GC. Evaluation of
69. Lomonaco S, Ventrella F, and Ciminiera P. Monte Carlo simulation of
63. Li X, Hui X, Fu Y, Li J, Liu L, and Guo X. A Deep Learning-Based Scatter
61. doi:10.1118/1.598248
60. Lei Y, Harms J, Wang T, Liu Y, Shu HK, Jani AB, et al. MRI-only Based
58. Lee H, and Lee J. A Deep Learning-Based Scatter Correction of Simulated
57. Lalone D, Winey B, Verburg I, Pagannetti H, and Sharp GC. Evaluation of
56. Lalone D, Winey B, Verburg I, Pagannetti H, and Sharp GC. Evaluation of
55. Linardatos P, Papastefanopoulos V, and Kotsiantis S. Explainable AI: A
54. Linardatos P, Papastefanopoulos V, and Kotsiantis S. Explainable AI: A
53. doi:10.1016/j.physletb.2018.01.002
52. Li X, Hui X, Fu Y, Li J, Liu L, and Guo X. A Deep Learning-Based Scatter
47. Low DA, Harms WB, Mutic S, and Purdy JA. A Technique for the
44. doi:10.1038/nature14539
42. Kowalski P, Wiącek K, Wiącek W, and Cichy M. A Proposal for an Analysis
34. Kowalski P, Wiącek K, Wiącek W, and Cichy M. A Proposal for an Analysis
33. doi:10.1088/0031-9155/48/17/303


**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Sarrut, Etxebeste, Muñoz, Krah and Létang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.