Artificial Intelligence approaches for Monte Carlo simulation

David Sarrut
Ane Etxebeste, Albert Saporta, Nils Krah,
Théo Kaprelian, Jean-Michel Létang

CNRS - University of Lyon

CREATIS, CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, France

Deep Learning and Monte Carlo (med phys)

Several investigations in the previous years

TABLE 1 | AI-based applications r
properties such as energy, positic

Application
Dose computation
Dose denoising
SPECT scan-time reduction
CBCT scatter modelling
PET attenuation/scatter correctior
Detector response modelling
Source + phase space modelling
Event selection
Interaction position in scintillators

[1]*http://hdl.handle.net/11603/19255*
[2]*http://hdl.handle.net/2078.1/thesis:14550*

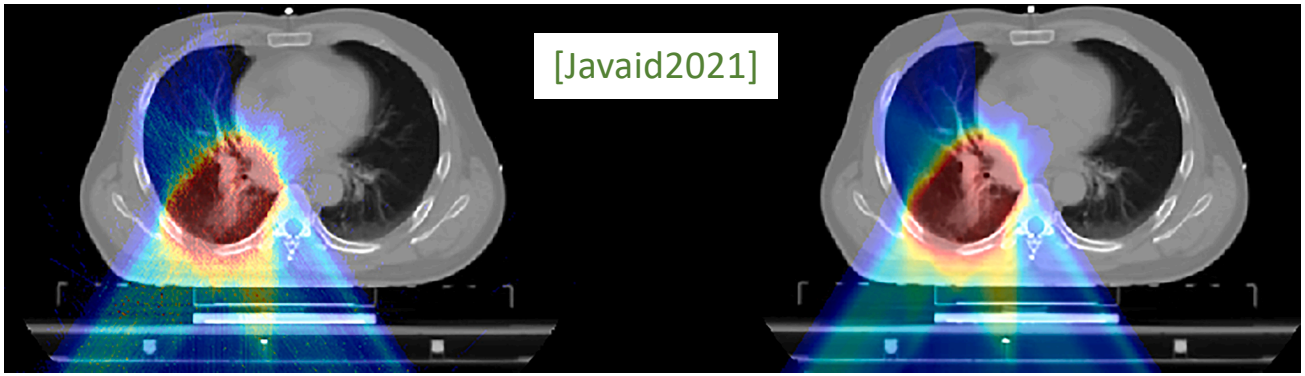


Fig. 3. A dose distribution on a sample lung test case. *Left* is the noisy dose distribution (MC simulation with 1×10^6 protons) and *right* is the reference dose distribution computed with 1×10^9 protons.

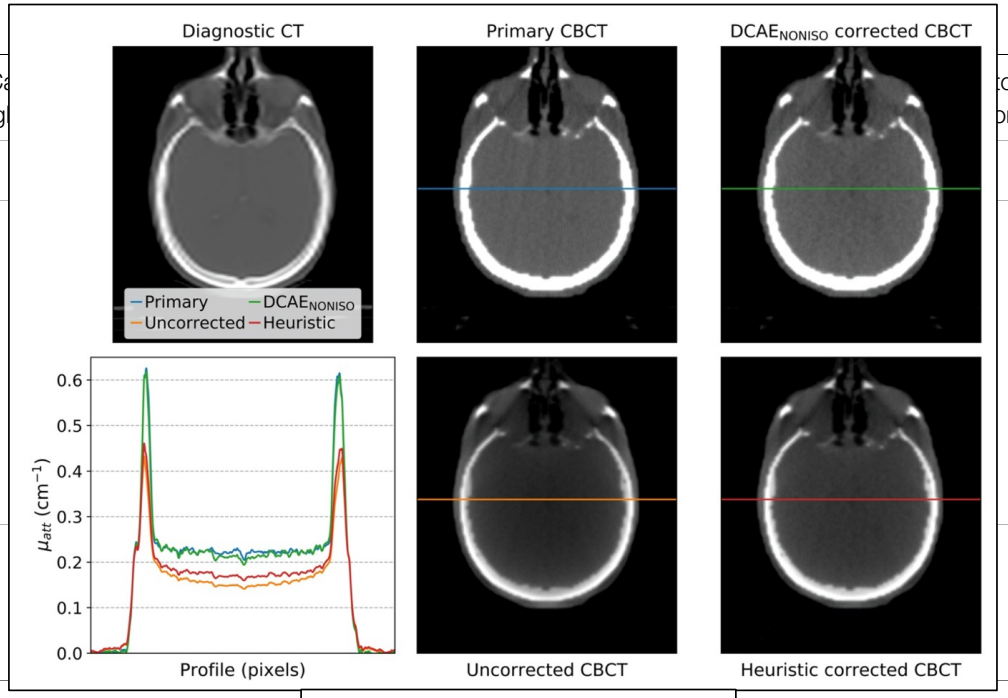
Deep Learning and Monte Carlo (med phys)

Several investigations in the previous years

TABLE 1 | AI-based applications related to Monte Ca … to a vector of particle properties such as energy, position, direction, weig… on.

Application	Main ML types
Dose computation	CNN, U-net
Dose denoising	CNN, U-net
SPECT scan-time reduction	CNN, U-net
CBCT scatter modelling	CNN, U-net
PET attenuation/scatter correction	CNN, U-net
Detector response modelling	GAN, MLP
Source + phase space modelling	GAN
Event selection	MLP, CNN
Interaction position in scintillators	MLP, CNN

[1]*http://hdl.handle.net/11603/19255*
[2]*http://hdl.handle.net/2078.1/thesis:14550*



[van den Heyden2020]

Deep Learning and Monte Carlo (med phys)

Several investigations in the previous years

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.

Application	Input type	Refs (among others)	Main ML types
Dose computation	image	[49, 63, 79, 85, 90, 104, 116, 117, 147]	CNN, U-net
Dose denoising	image	[43, 59, 71, 101, 103, 111, 131, 153][1]	CNN, U-net
SPECT scan-time reduction	image	[82, 119, 121]	CNN, U-net
CBCT scatter modelling	image	[27, 58, 60, 75, 79, 84, 87, 88, 140, 145, 152, 155]	CNN, U-net
PET attenuation/scatter correction	image	[6,	
Detector response modelling	particles	[126,	
Source + phase space modelling	particles	[108, 1	
Event selection	particles	[8, 12, 40, 46, 93, 98	
Interaction position in scintillators	various	[23, 33, 37, 99, 109,	

[1] *http://hdl.handle.net/11603/19255*
[2] *http://hdl.handle.net/2078.1/thesis:14550*

[Sarrut et al. Frontiers 2022]

frontiers

Artificial Intelligence for Monte Carlo simulation in medical physics

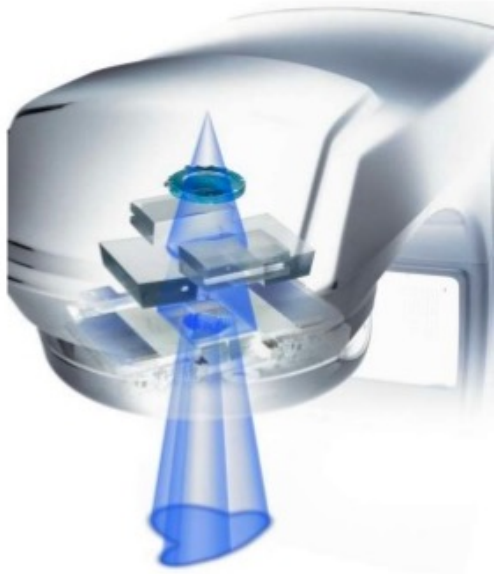
David Sarrut [1,*], Ane Etxebeste [1], Enrique Muñoz [2], Nils Krah [1,2] and Jean Michel Létang [1]

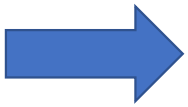
[1] *Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69373, Lyon, France.*
[2] *University of Lyon, Université Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, F-69622. Villeurbanne. France*

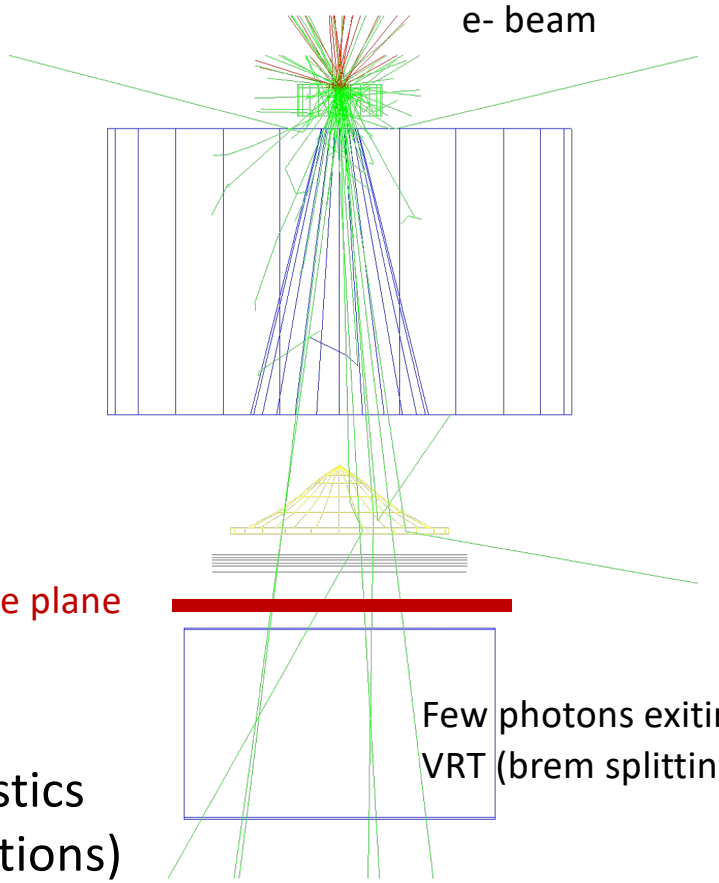
Example 1

Modeling a phase-space with a GAN

Radiation Therapy Linac head simulation





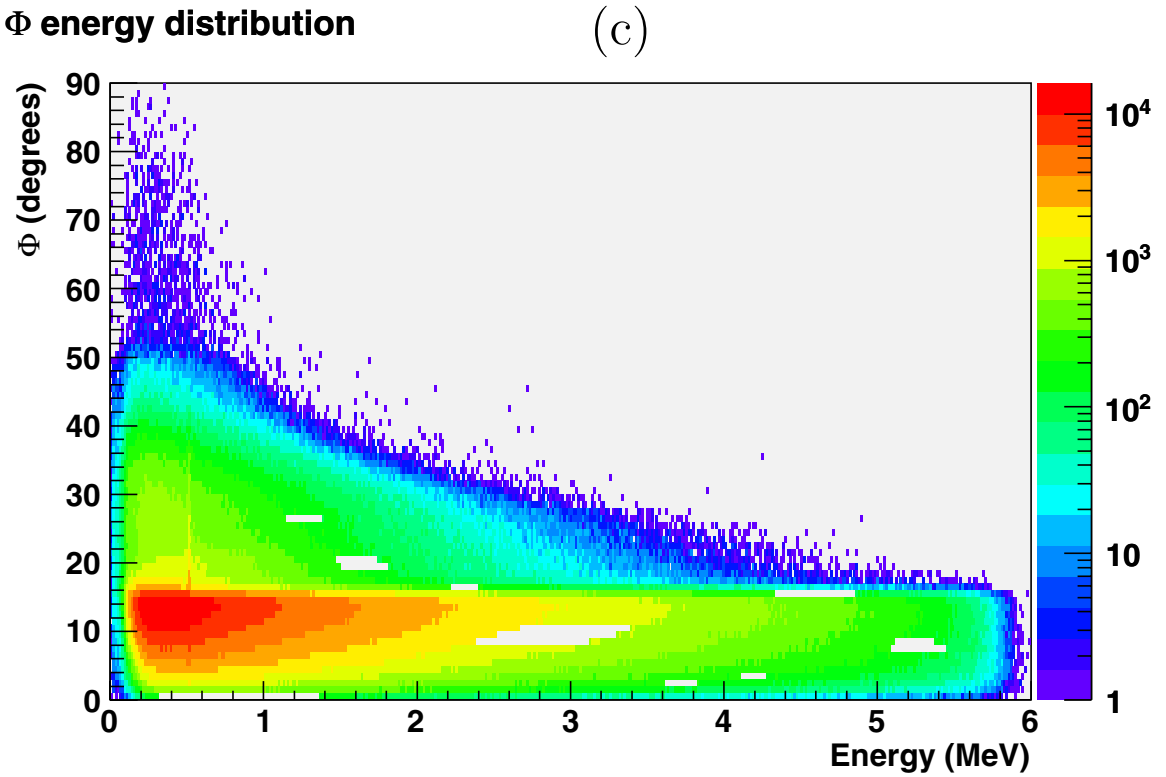


Goal: determine beam characteristics
(energy, position, direction distributions)

Phase Space (PHSP)

Store beam properties as **Phase Space**

- A PHSP is a list of particles (around 10^8, 10^9)
- Properties:
  E, x, y, z, dx, dy, dz, w, t



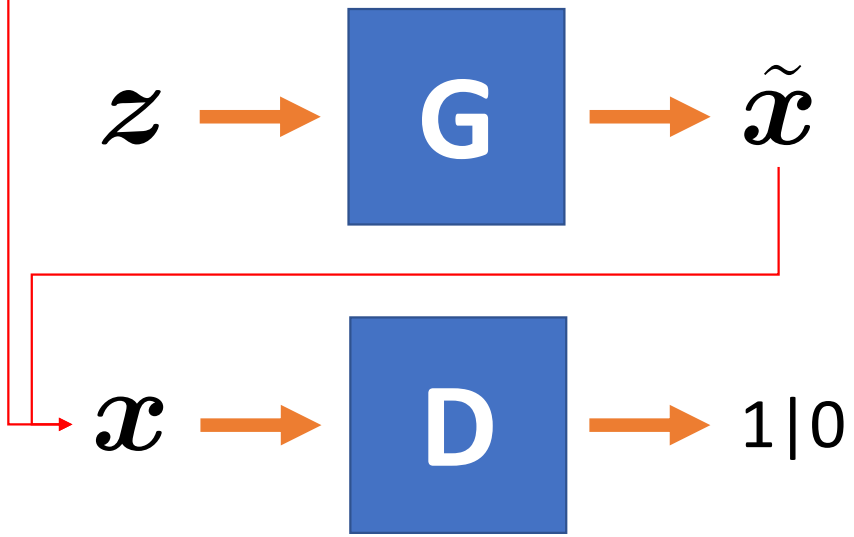
Example of dependance of direction ϕ and energy

GAN: Generative Adversarial Network

- Training dataset $\boldsymbol{x} \in \mathbb{R}^d$
  - Dimension d=7 (E, X, Y, Z, dX, dY, dZ)
  - Samples of unknown p_{real}
- Generator $G(\boldsymbol{z}; \boldsymbol{\theta}_G)$
- Discriminator $D(\boldsymbol{x}; \boldsymbol{\theta}_D)$

$$J_D(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G) = \mathbb{E}_{\boldsymbol{z}}[D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{x}}[D(\boldsymbol{x})]$$
$$J_G(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G) = -\mathbb{E}_{\boldsymbol{z}}[D(G(\boldsymbol{z}))]$$

Wasserstein GAN [Arjovsky 2017]

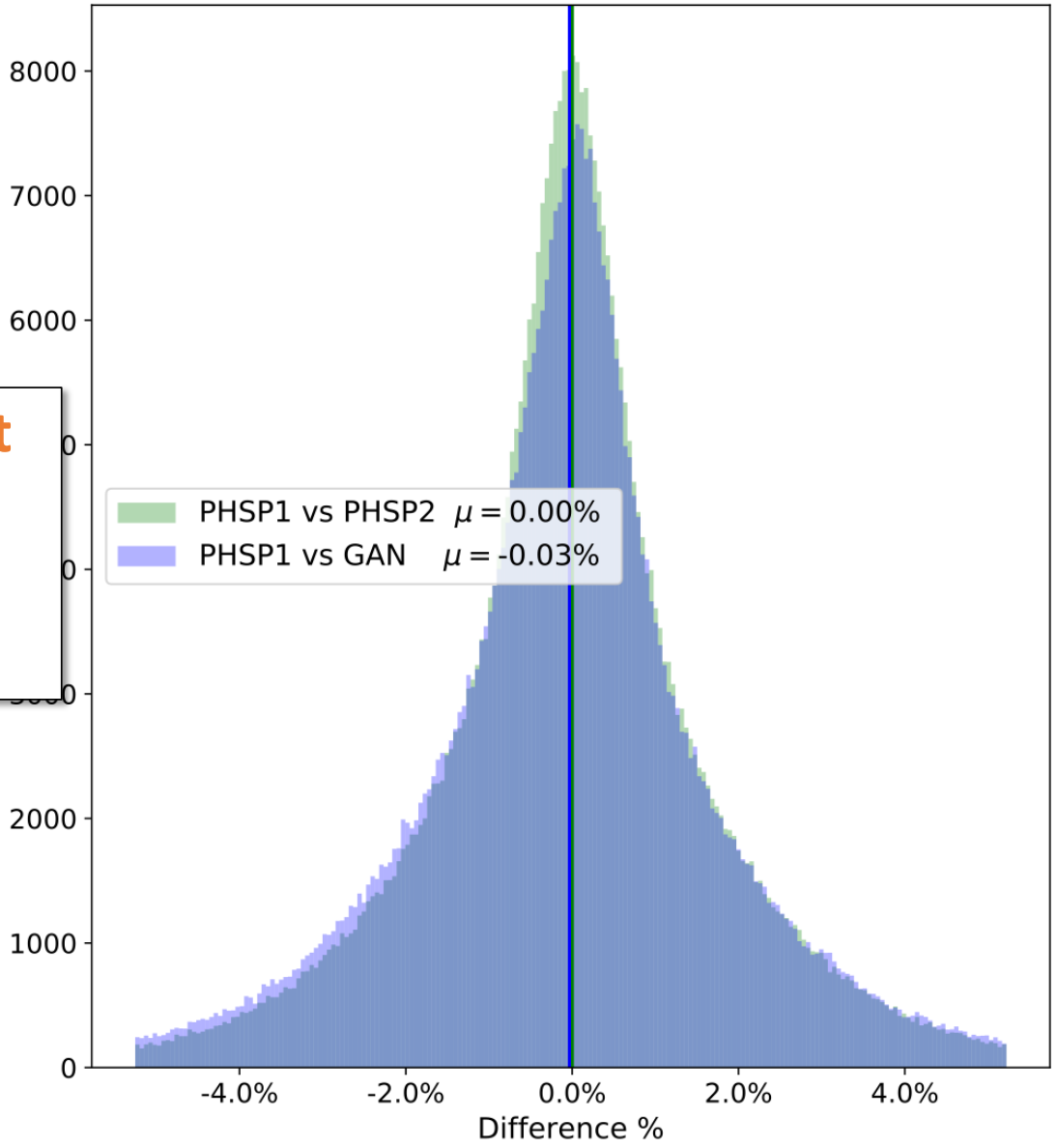


[Goodfellow 2014]

Results

- **Sufficient for dose but not perfect**
- **Smooth-out 511 keV peak**

- **2GB replaced by 10 MB**

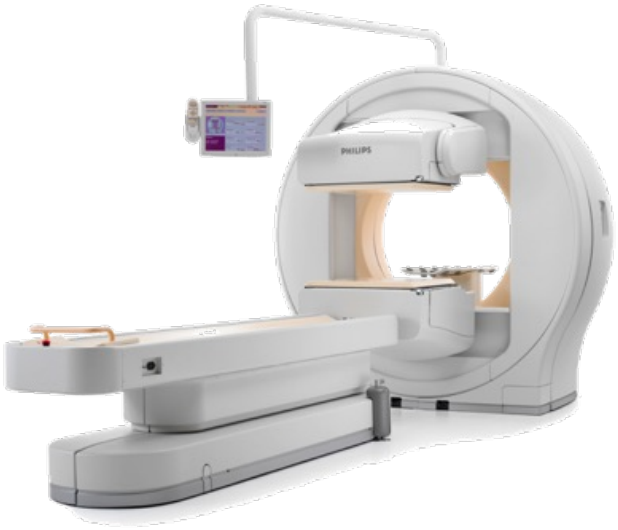


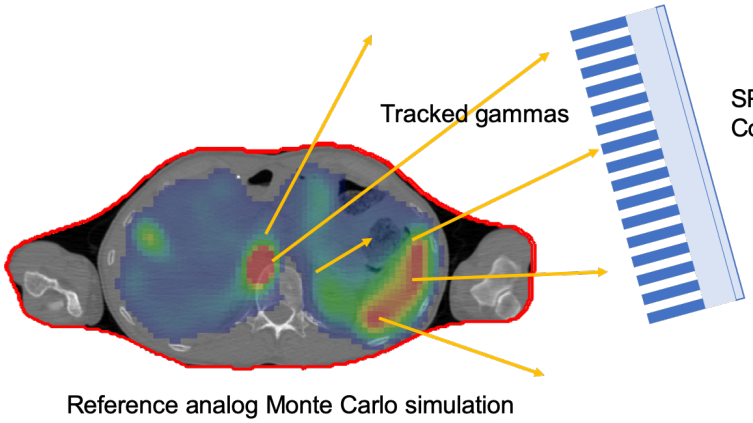
Example 2

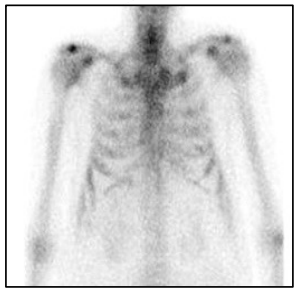
Modeling “forward model” and “detector response” with GAN and NN

SPECT simulation

- Part1: from emission to patient exiting gamma
- Part2: track gamma inside the detector







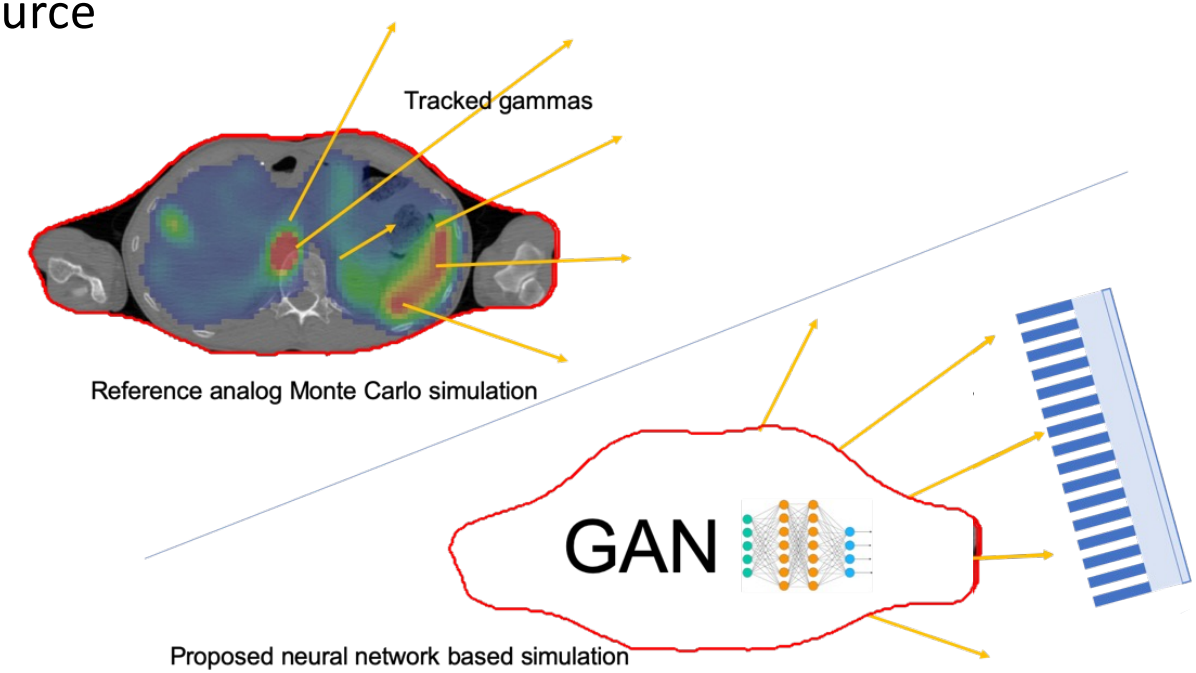
Training dataset

Train a GAN to produce exiting gamma from a given source

- Step1: run low stats MC, consider exiting gammas (in a phsp)
- Step2: train a GAN, use it as a source
- Step3: track to detector

Conditional GAN

- Condition = activity distribution
- Specific to a CT image
- Generic to any activity distribution

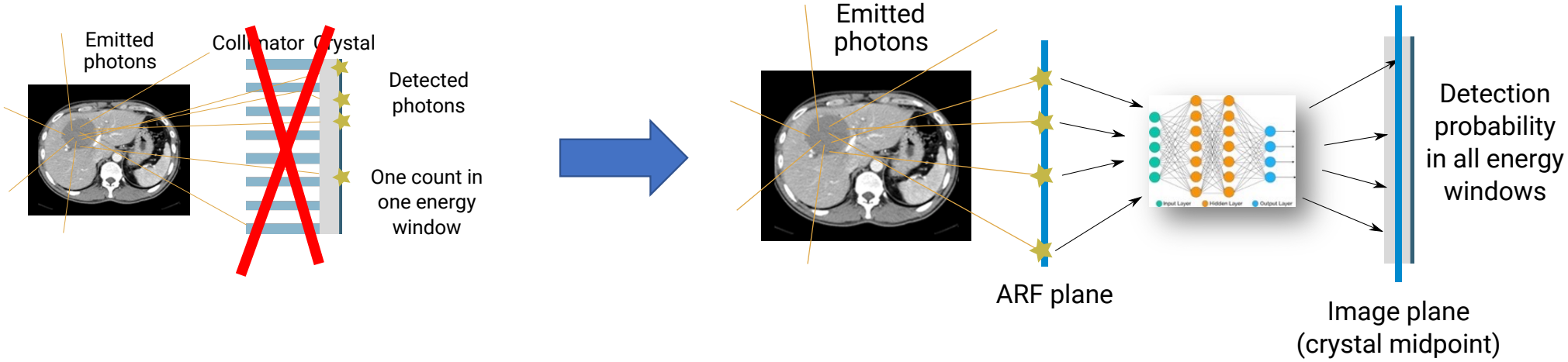


Detector response

[Song2005]
[Descourt2010]
[Rydeen2018]
[Sarrut2018]

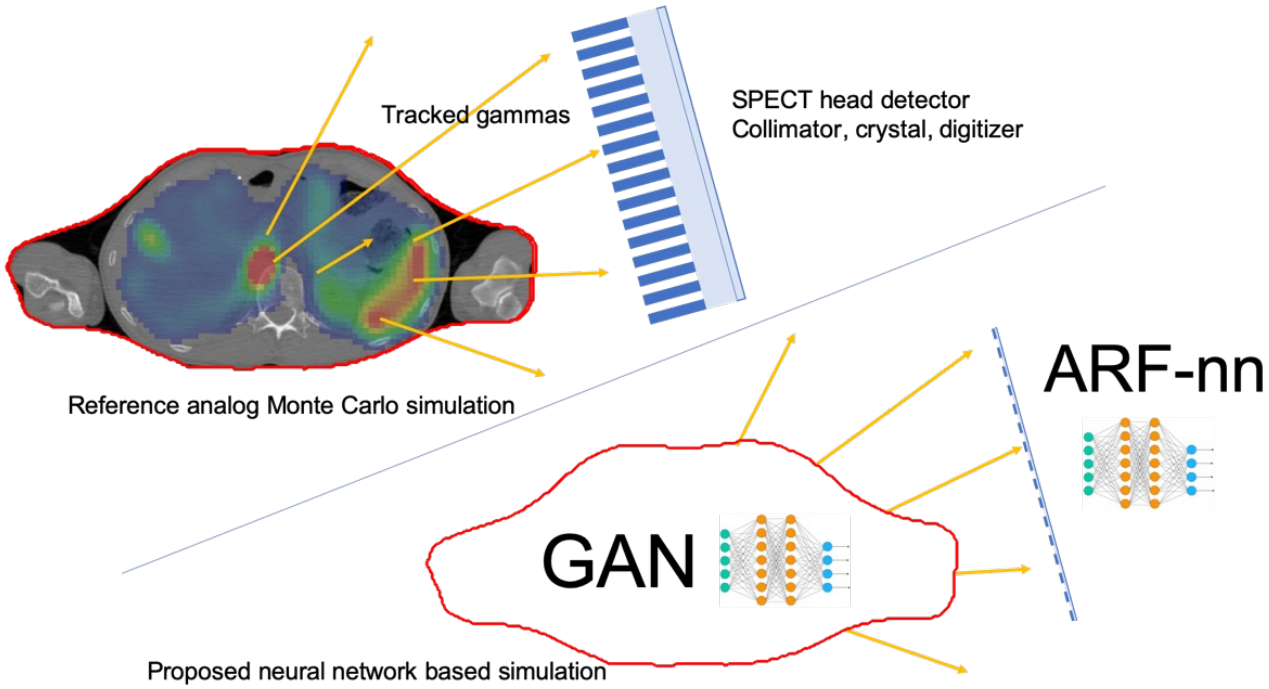
With Angular Response Function (ARF)

Modeled as a neural network

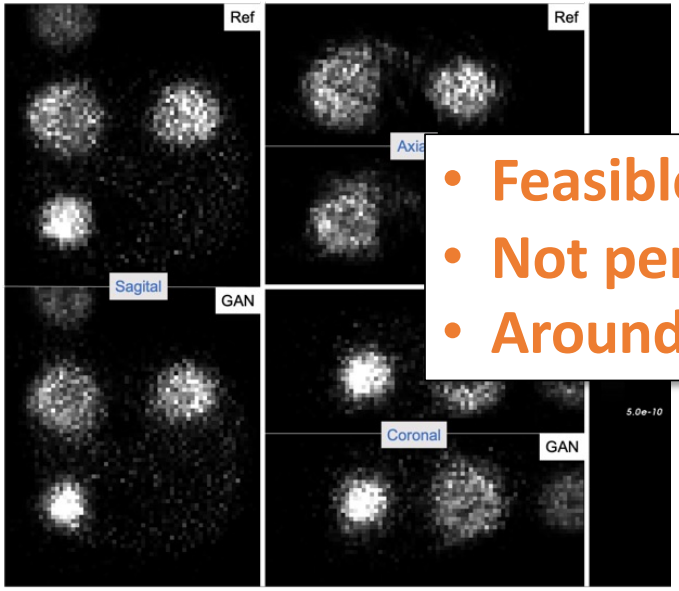


GAN as a “forward” model

The GAN produce exiting gamma from any activity source

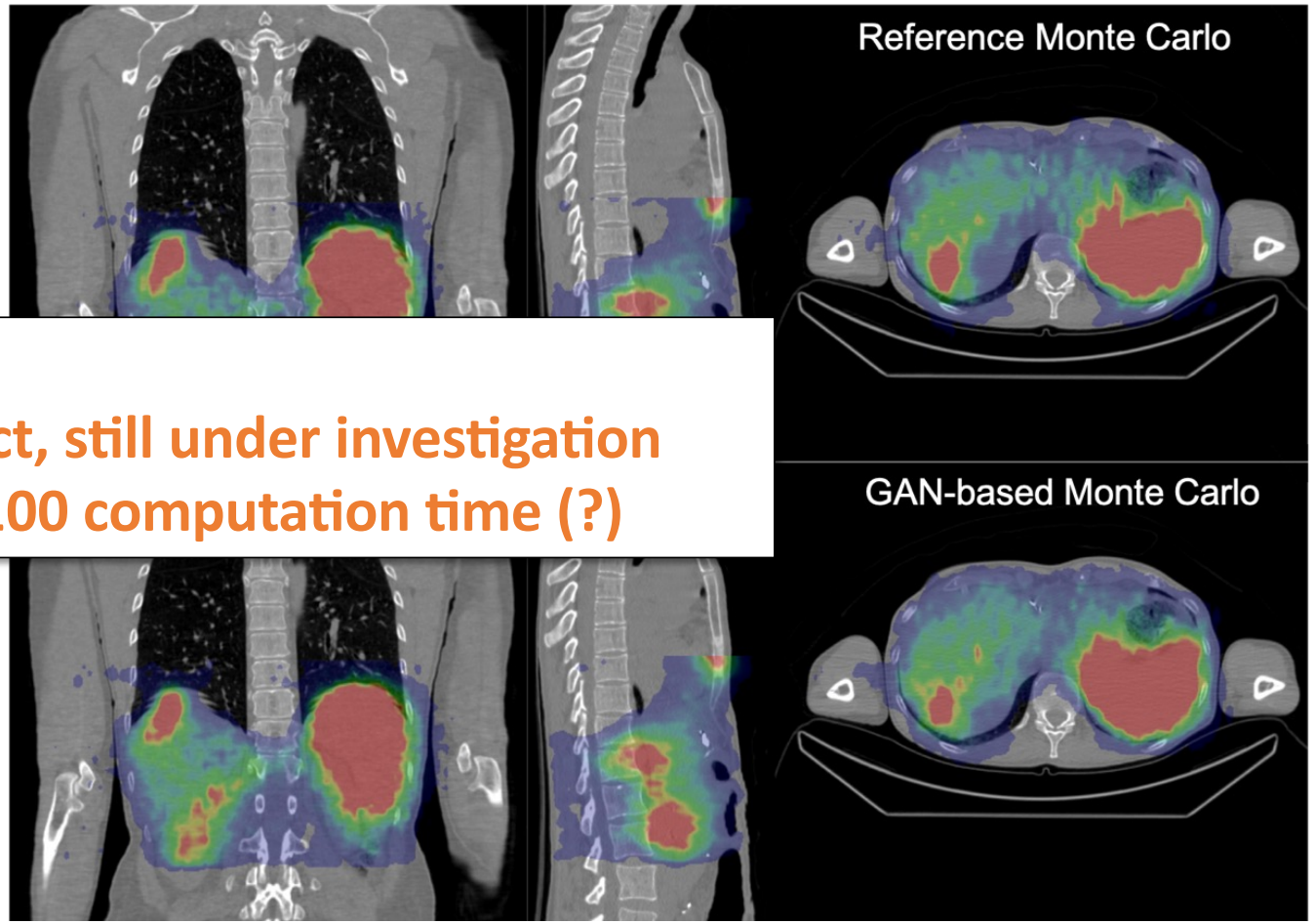


Results



Test1: 3D reconstruction

- **Feasible**
- **Not perfect, still under investigation**
- **Around x100 computation time (?)**



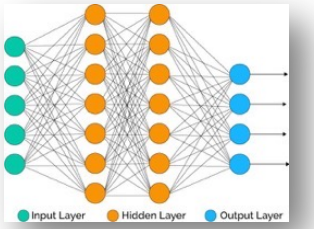
Test2: 3D reconstruction

Technical: implementation

How to integrate DL within Geant4

D.L. / Geant4 integration

Goal: neural network integration in Geant4 simulation



- Geant4 is C++, plain CPU, multithreads
- Most of AI toolkit's front-ends are Python, GPU
- **Training**: external to Geant4 (for the moment)
- **Inference**: within Geant4

PyTorch NVIDIA

Geant4 integration (#1)

PyTorch C++ lib (*libTorch*)

- Use the compiled *libTorch* library
- Convert the trained nn into a ".pt" file (pretrained weights)
- Load net : `nn = torch::jit::load(path);`
- Run net : `output = nn.forward(input).toTensor();`
- Convert input and output tensors
- GPU not straightforward

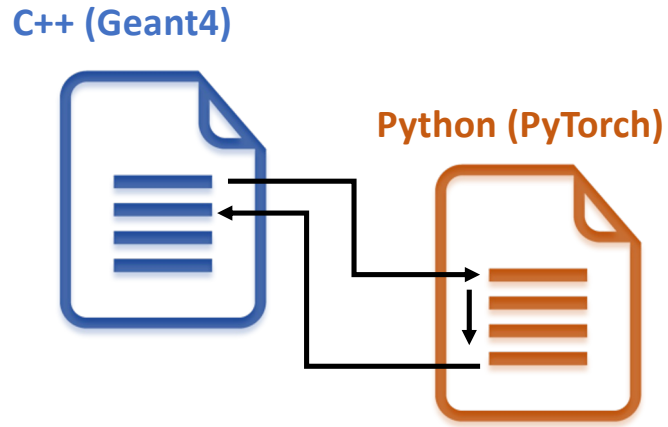
Av: integrated

Inc: long development/maintenance time, complex API

Geant4 integration (#2)

In the new GATE system
- Simulation described in Python (no more macros)
- (Part of) Geant4 is exposed with *pybind11*

- Callbacks from C++ to Python during simulation
  - Conventional PyTorch code
  - Copy/convert data from C++ to Python
- Accumulate data **before** callback
  e.g. only every 10^5 ”hits”



Av: easier development, GPU

Inc: (callback is slow but infrequent), some data copy

IOP Publishing Phys. Med. Biol. 63 (2018) 205013 (12pp)

[Sarrut et al. PMB 2018]

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

PAPER

Learning SPECT detector angular response function with neural network for accelerating Monte-Carlo simulations

RECEIVED
11 June 2018

REVISED
17 September 2018

ACCEPTED FOR PUBLICATION
21 September 2018

PUBLISHED
17 October 2018

D Sarrut[1], N Krah, J N Badel and J M Létang

Université de Lyon, CREATIS, CNRS UMR5220; Inserm U1044, INSA-Lyon, Université Lyon 1, Centre Léon Bérard, Lyon, France

[1] Author to whom any correspondence should be addressed.

E-mail: david.sarrut@creatis.insa-lyon.fr

Keywords: Monte-Carlo simulation, SPECT imaging, variance reduction technique, neural network

Abstract

A method to speed up Monte-Carlo simulations of single photon emission computed tomography (SPECT) imaging is proposed. It uses an artificial neural network (ANN) to learn the angular response function (ARF) of a collimator–detector system. The ANN is trained once from a complete simulation including the complete detector head with collimator, crystal, and digitization process. In

IOP Publishing Phys. Med. Biol. 64 (2019) 215004 (11pp)

[Sarrut et al. PMB 2019]

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

PAPER

Generative adversarial networks (GAN) for compact beam source modelling in Monte Carlo simulations

RECEIVED
23 April 2019

REVISED
27 August 2019

ACCEPTED FOR PUBLICATION
30 August 2019

PUBLISHED
23 October 2019

D Sarrut[1], N Krah[1,2] and J M Létang[1]

[1] Université de Lyon, CREATIS; CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, Centre Léon Bérard, France

[2] Université Lyon, CNRS/IN2P3 UMR 5822, IP2I Lyon, Villeurbanne, France

E-mail: david.sarrut@creatis.insa-lyon.fr

Keywords: Monte-Carlo simulation, generative adversarial network, phase-space, linac

Abstract

A method is proposed and evaluated to model large and inconvenient phase space files used in Monte Carlo simulations by a compact generative adversarial network (GAN). T
on a phase space dataset to crea
multidimensional data distribu
with about 0.5 million weights,
generated with G to replace the
This concept is applied to be
seed models. Simulations using
generated by the GAN on the o

[Sarrut et al. Frontiers 2022]

frontiers

Artificial Intelligence for Monte Carlo simulation in medical physics

David Sarrut [1,*], Ane Etxebeste [1], Enrique Muñoz [2], Nils Krah [1,2] and Jean Michel Létang [1]

Lyon 1, UJM-Saint Etienne,
73, Lyon, France.
1, CNRS/IN2P3, IP2I Lyon,

IOP Publishing Phys. Med. Biol. 66 (2021) 055014

[Sarrut et al. PMB 2021]

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

PAPER

Modeling complex particles phase space with GAN for Monte Carlo SPECT simulations: a proof of concept

RECEIVED
5 October 2020

REVISED
14 January 2021

ACCEPTED FOR PUBLICATION
21 January 2021

PUBLISHED
19 February 2021

D Sarrut[1], A Etxebeste[1], N Krah[1,2] and JM Létang[1]

[1] Université de Lyon, CREATIS, CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, Centre Léon Bérard 69373, France

[2] University of Lyon, Université Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, F-69622, Villeurbanne, France

E-mail: david.sarrut@creatis.insa-lyon.fr

Keywords: Monte-Carlo simulation, generative adversarial network, SPECT, phase space

Abstract

A method is proposed to model by a generative adversarial network the distribution of particles exiting a patient during Monte Carlo simulation of emission tomography imaging devices. The resulting compact neural network is then able to generate particles exiting the patient, going towards the detectors, avoiding costly particle tracking within the patient. As a proof of concept, the method is evaluated for single photon emission computed tomography (SPECT) imaging and combined with

Modeling families of particle phase-space distributions with conditional GAN for Monte Carlo SPECT simulations

Albert Saporta[1], Ane Etxebeste[1], Théo Kaprelian[1], Jean Michel Létang[1], David Sarrut[1]

1. Université de Lyon; CREATIS; CNRS UMR5220; Inserm U1294; INSA-Lyon; Université Lyon 1, Lyon, France

[Saporta, PMB 2022 (revision)]

Abstract.

We propose a method to model families of phase-space distributions of particles emitted within a phantom during Monte Carlo simulation of SPECT imaging devices with a conditional Generative Adversarial Network (condGAN). The condGAN is trained on a phase-space dataset containing, in addition to the energy, time, position and direction of emitted particles, a vector of conditions composed of four dimensions:

Concl

AI and M

- Train
- Fast p
- Phase

- Limitati
  - Still u
  - Uncle
  - Need

g dataset
on (e.g. 511 keV peak)

- Perspec...
  - **Conditional GAN**: to learn a family of phase-spaces [Saporta2022 PMB]

Thanks for your attention !

CREATIS Lyon, France