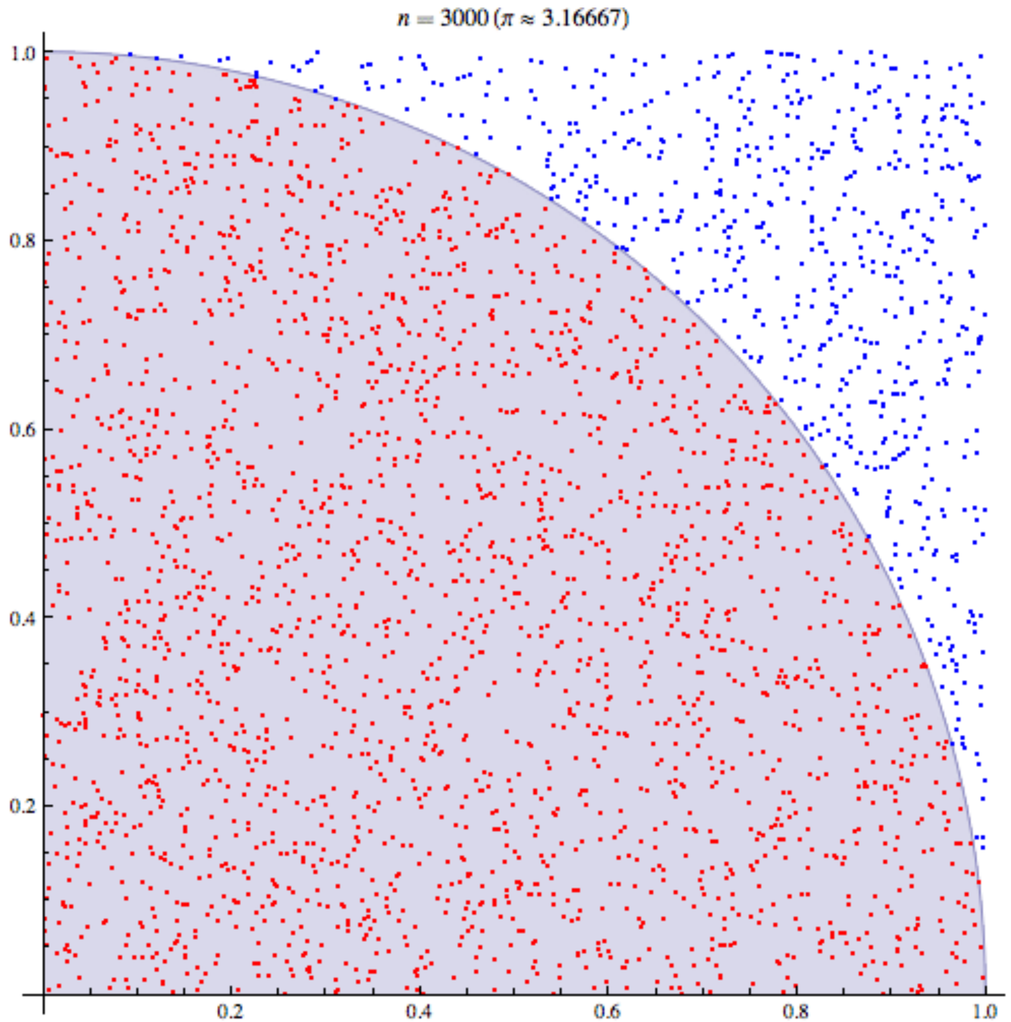
Artificial Intelligence approaches for Monte Carlo simulation

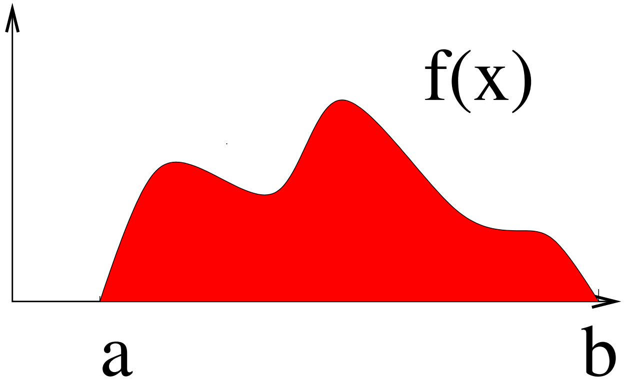
David Sarrut

University of Lyon

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

Monte Carlo





$$I = \int_a^b dx\, f(x) \qquad I \approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$$

Error: $\propto 1/\sqrt{N}$

Monte Carlo simulations

- Born during WW2
  - Stanislaw Ulam, John von Neumann … Manhattan Project

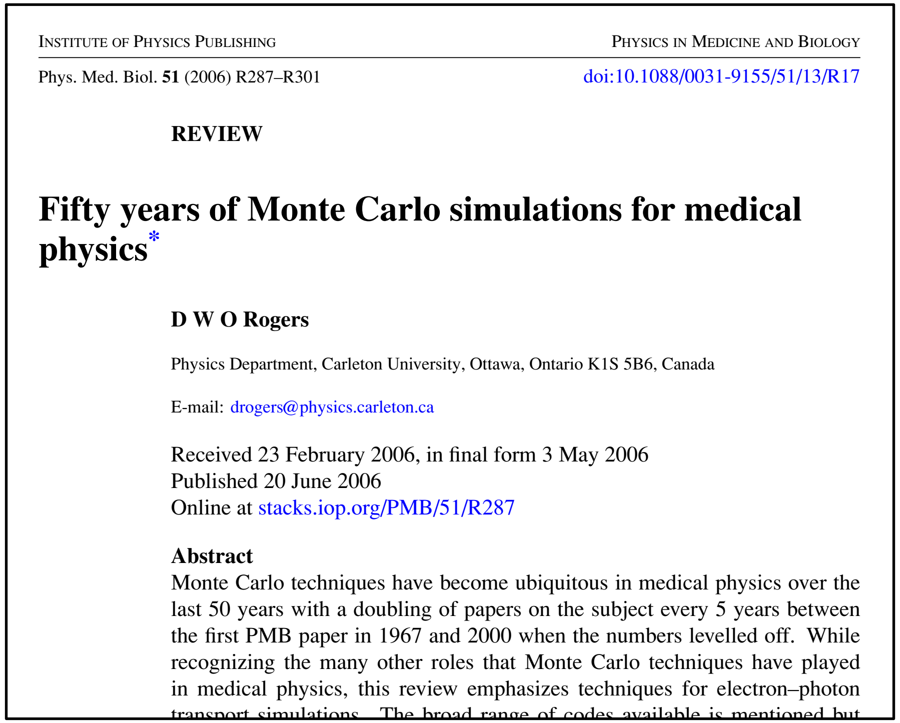
- MC in medical physics
  - Simulating radiation transport
  - Roots in the 70’, imaging systems (SPECT, PET) and Radiation Therapy
- Nowadays:
  - All TPS (Treatment Planning System)
  - All PET, SPECT ; Total-Body PET project (Explorer, etc)

100%

Monte Carlo simulations evolution

- More than 60 years of evolution
  - More accurate physical databases
  - More generic codes (MCNPX, EGSNRC, Penelope, Geant4, Gate)
  - Faster algorithms
  - Use of powerful computing infrastructures (cluster, GPU)

- However
  - Increasing need for detailed and accurate physical processing (TOF, SiPM, CZT, etc)
  - Still with long simulations times (need VRT)


INSTITUTE OF PHYSICS PUBLISHING PHYSICS IN MEDICINE AND BIOLOGY

Phys. Med. Biol. **51** (2006) R287–R301 doi:10.1088/0031-9155/51/13/R17

REVIEW

Fifty years of Monte Carlo simulations for medical physics*

D W O Rogers

Physics Department, Carleton University, Ottawa, Ontario K1S 5B6, Canada

E-mail: drogers@physics.carleton.ca

Received 23 February 2006, in final form 3 May 2006
Published 20 June 2006
Online at stacks.iop.org/PMB/51/R287

Abstract

Monte Carlo techniques have become ubiquitous in medical physics over the last 50 years with a doubling of papers on the subject every 5 years between the first PMB paper in 1967 and 2000 when the numbers levelled off. While recognizing the many other roles that Monte Carlo techniques have played in medical physics, this review emphasizes techniques for electron–photon transport simulations. The broad range of codes available is mentioned but

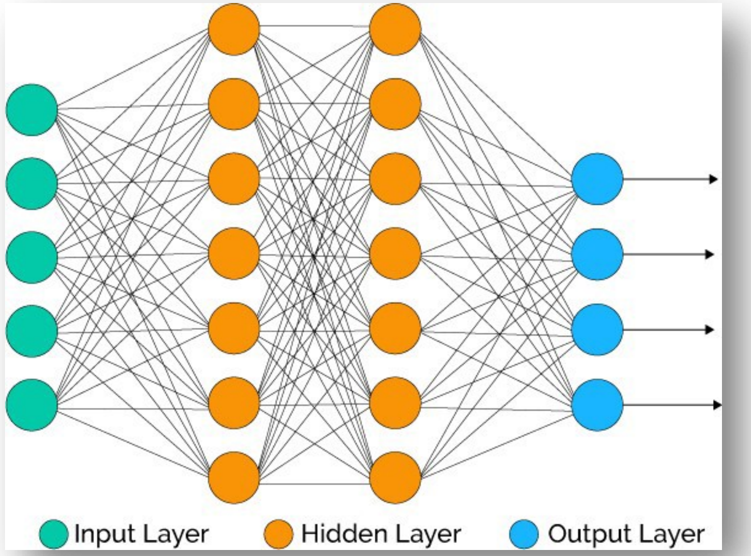
Artificial Intelligence (A.I)

- A.I. methods, image processing (photos, video)
- Deep Learning, neural network

- Medical physics:
    - Detection
    - Auto segmentation
    - Image generation (CT from MRI, CT from CBCT etc)
    - Image enhancement
    - Radiomics
    - etc …

Deep learning principle

- **Step1**: learn a model
  - Input training database (large), composed of numerous independent samples
  - Neural network architecture and learning methods



- **Step2**: use the model
  - Get input data, apply the NN

AI: a perfect fit for MC ?

Very short literature review

- **DL and dose estimation**
  - **[Lee2019, Götz2020, Roser2019, Nguyen2019, Liu2019]**
  - U-Net architecture, patch-based, predict dose
  - Large dataset variation ?
- **DL for dose computation denoising**
  - **[Peng2019, Fornander2019, Neph2019, Javaid2019, Madrigal2018]**
  - Towards less particles to track during MC simulation
  - Photon, proton dose. How to preserve dose gradient ?
  - Towards GAN ?
- **DL for detector and source modelling**
  - **[Sarrut2018, Sarrut2019, Zatcepin 2020]**
  - Depth-of-interaction resolution in pixellated PET detectors
- **DL for scatter modeling and correction reconstruction**
  - **[B van der Heyden2020, Lee2019, Maier2018, Sharp202]**
  - U-Net, dense scatter estimation

Here: use of **Deep Learning** with **Monte Carlo** simulation

- Articles from 2018, 2019, 2020
- Evolving field
- Investigations, may not be ready for clinic yet

- Training dataset size?
- Training dataset variability?
- Generalisation to other cases types?

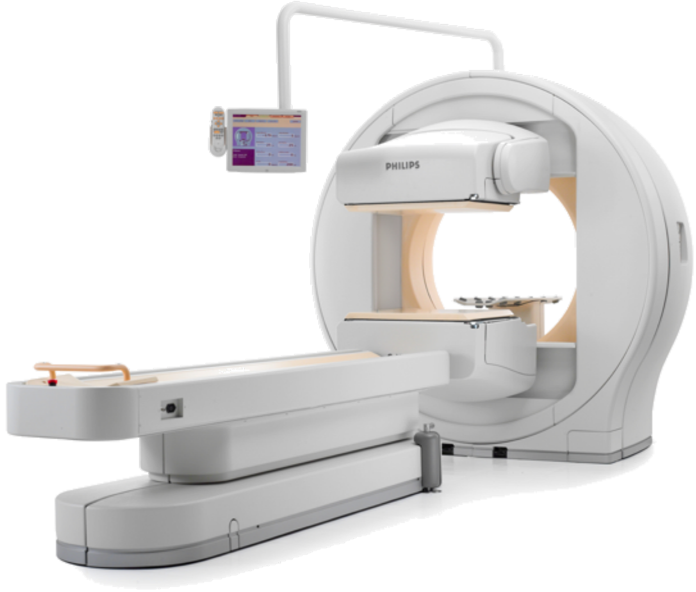
Examples of AI for Monte Carlo

- Example1: learning Angular Response Function for SPECT simulation
- Example2: learning Phase-Space for photon beam characterisation

Deep learning **within** Monte Carlo simulation

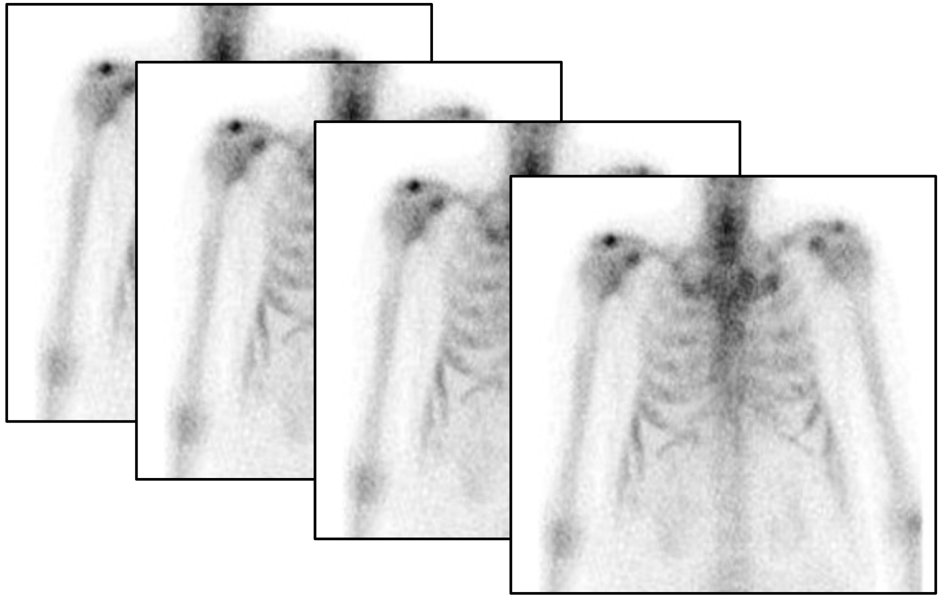
Example 1:
learning ARF for
SPECT simulation

SPECT Monte-Carlo simulation



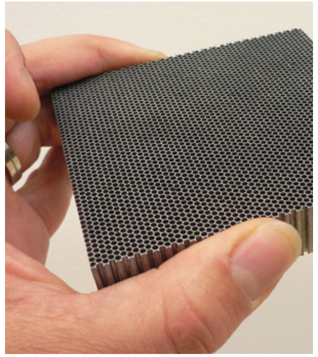
- Long computation time
- Around 10^{-4} particles reaching detector
- Brute-force approach up to few days computation

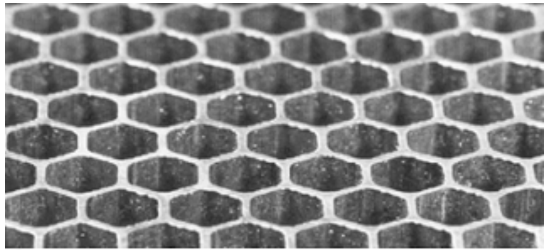
- Platforms:
  - SimSET [Harrison1993]
  - SIMIND [Ljungberg1989]
  - GATE/Geant4 [Sarrut2014]

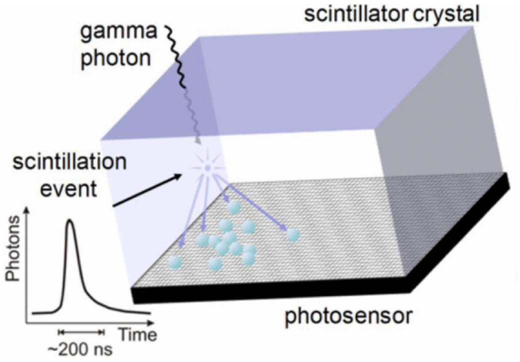


SPECT Monte-Carlo simulation

- Several proposed Variance Reduction Techniques (VRT) :
  - GIS: Geometrical Importance Sampling [Beenhouwer2009]
  - ARF: Angular Response Function [Song2005, Descourt2010, Rydeen2018]
  - MPS: Multiple Projection Sampling [Beenhouwer2008, Liu2008]
  - CFD: Convolution Based Forced Detection [Liu2008]
  - FFD: Fixed Forced Detection [Cajgfinger2017]



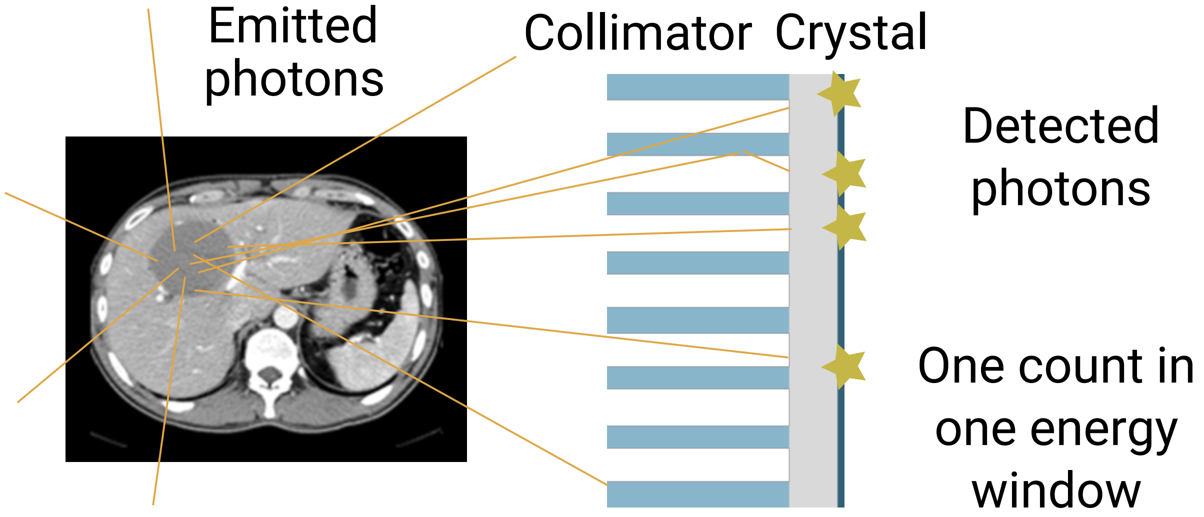


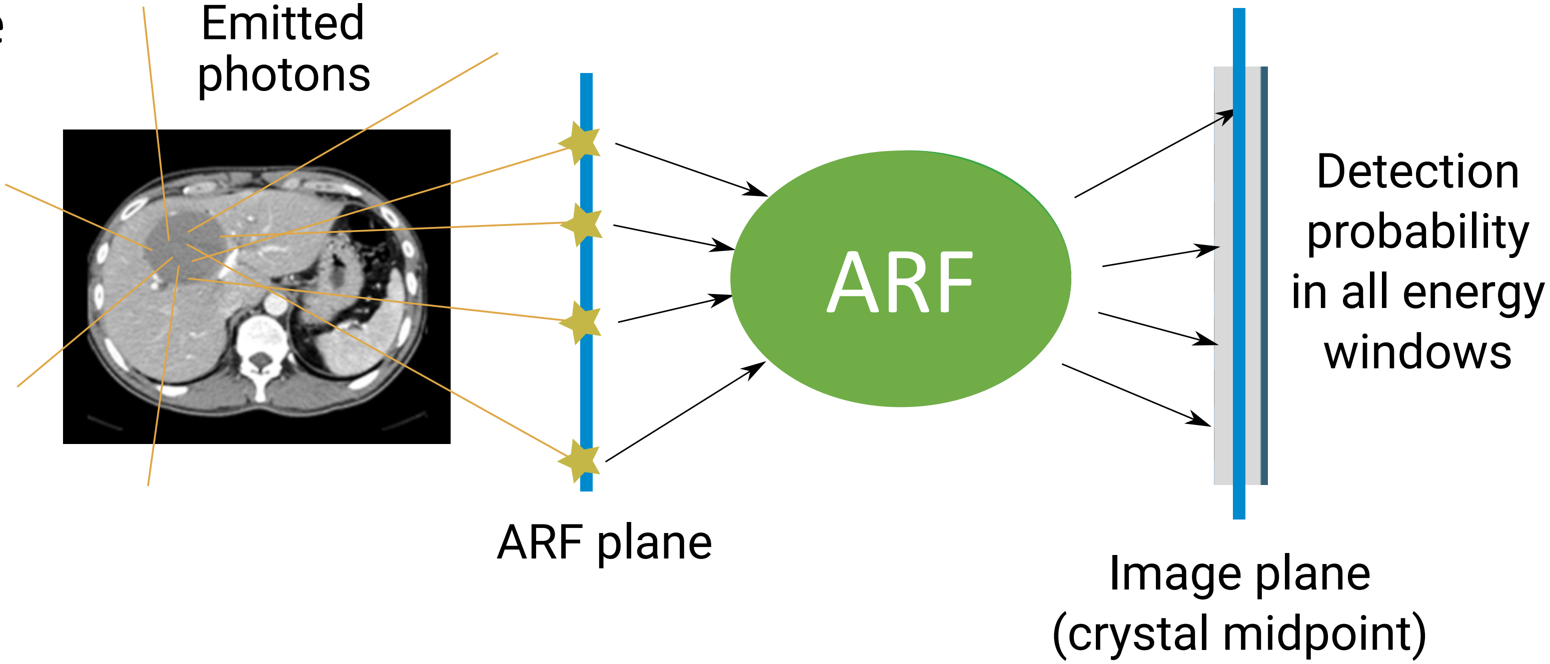


[Braga2014]

ARF: principles

- Angular Response Function
- Replace SPECT head detection with tabulated response
  - Incident particle at ARF plane use tables to get energy windows probabilities
- Assume:
  - Spatially invariant
  - Detection depends on direction + energy





ARF

Replace histogram tables by a neural network

- Advantages:
  - ARF tables needed to be computed only once
  - Variance reduction: probability instead of counts
  - Efficient, speedup x20-100 [Song2005, Descourt2010]

- Drawbacks:
  - ARF tables needed for every detector configurations
  - Large dataset needed to compute tables, 10^8 to 10^{11} [Rydeen2018]
  - Choice of table binning (3D histogram) not clear
  - Speedup not explicitly evaluated

Artificial neural network

- Learn a predictive model from a training dataset $h(\mathbf{x}) = \mathbf{y}$
- *Training dataset*:
  simulation, large source, complete energy spectra, complete detector (collimator/crystal)
  10^8 to 10^9 particles + **Russian Roulette**

$$\mathbf{x} = (E, \theta, \phi)$$

- *Input space*:
  particles energy and direction at the collimator entrance plane

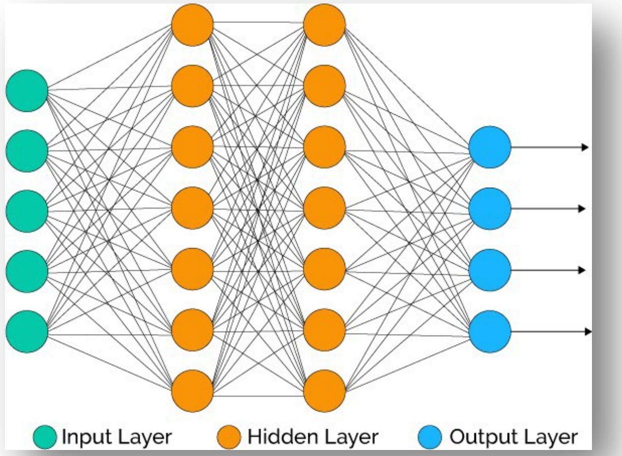
$$h(E, \theta, \phi) = y_i$$

Gives probability y_i for an incoming photon to be detected in the i^{th} energy window

Artificial neural network architecture

- 3 hidden linear fully connected layers
- 400 neurons by layer
- Activation function: ReLu
- Loss function: multiclass cross-entropy
- Optimisation: Adam [Kingma2014] (max 1000 iterations)
- Batch size: 5000 samples $\alpha = 0.0001$
- Adaptive learning rate

~ 10^5 weights (2 MB)



PYTORCH

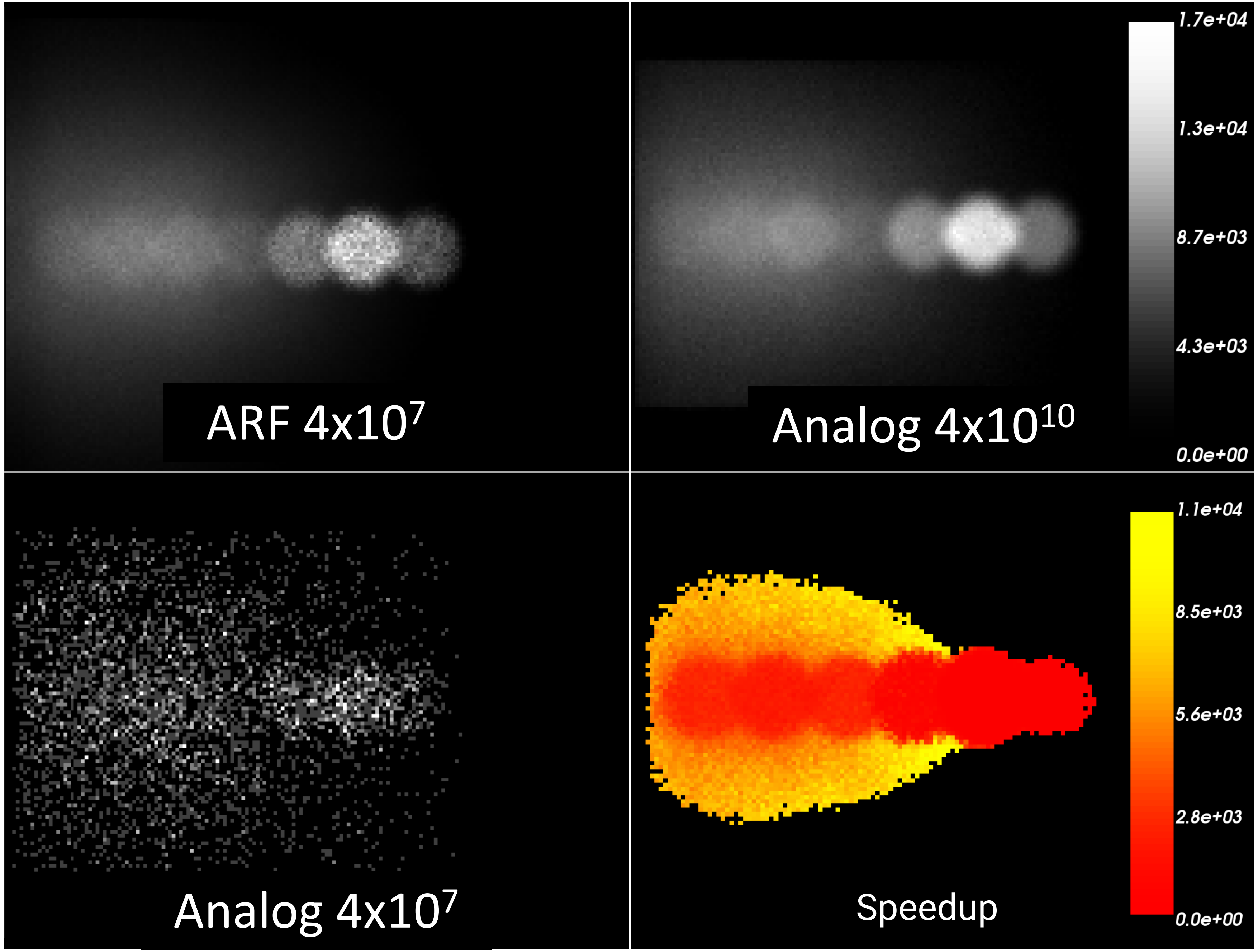
NVIDIA

Results

- Simulation of 7 circular sources of different energies
- Efficiency

$$\varepsilon_k = \frac{1}{t \times \sigma_k^2}$$

Speedup: 20 – 1000



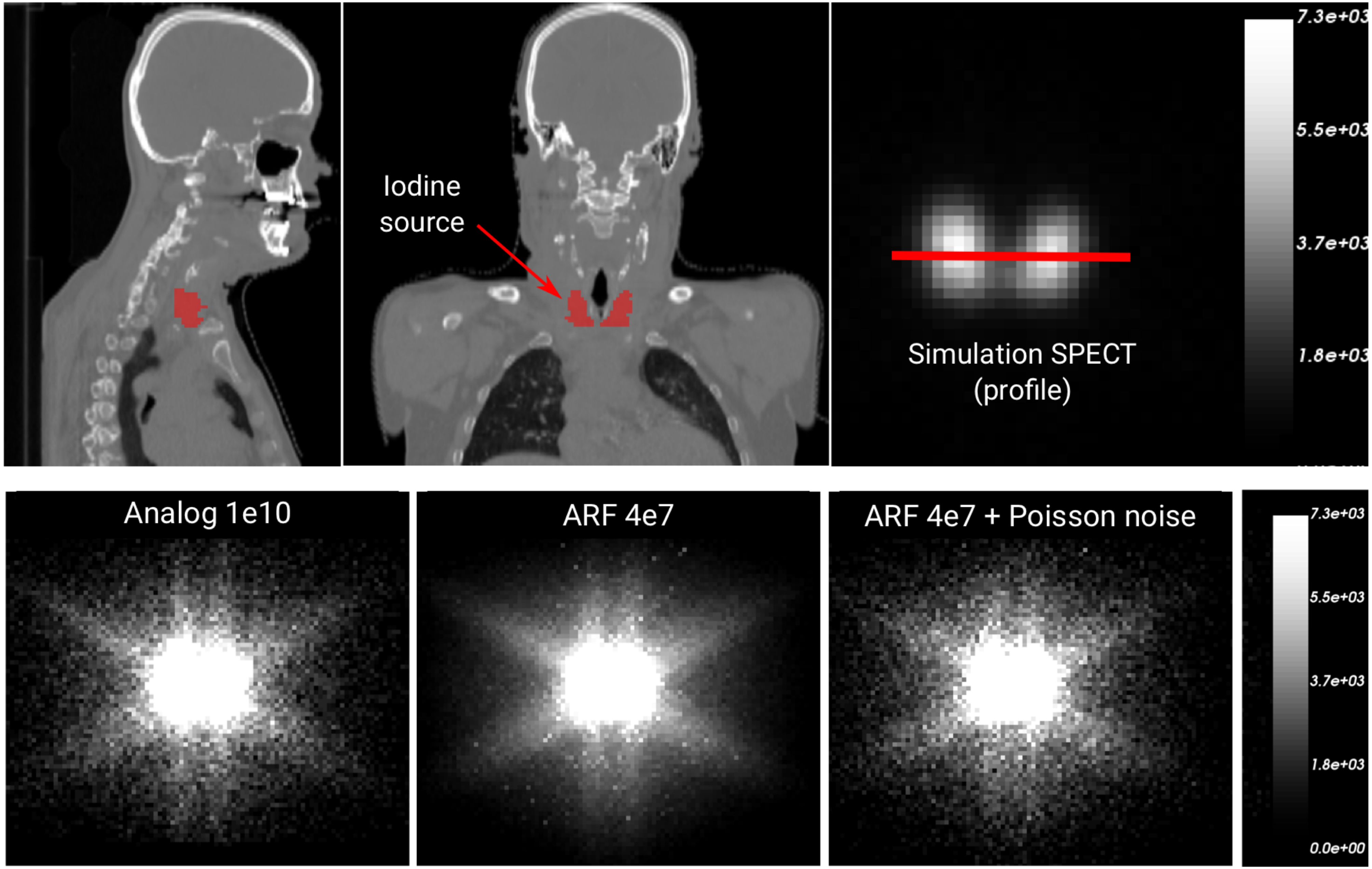
Results

25 days CPU time with 10^{10} particles

vs

2.5 hours with 4.10^7 particles

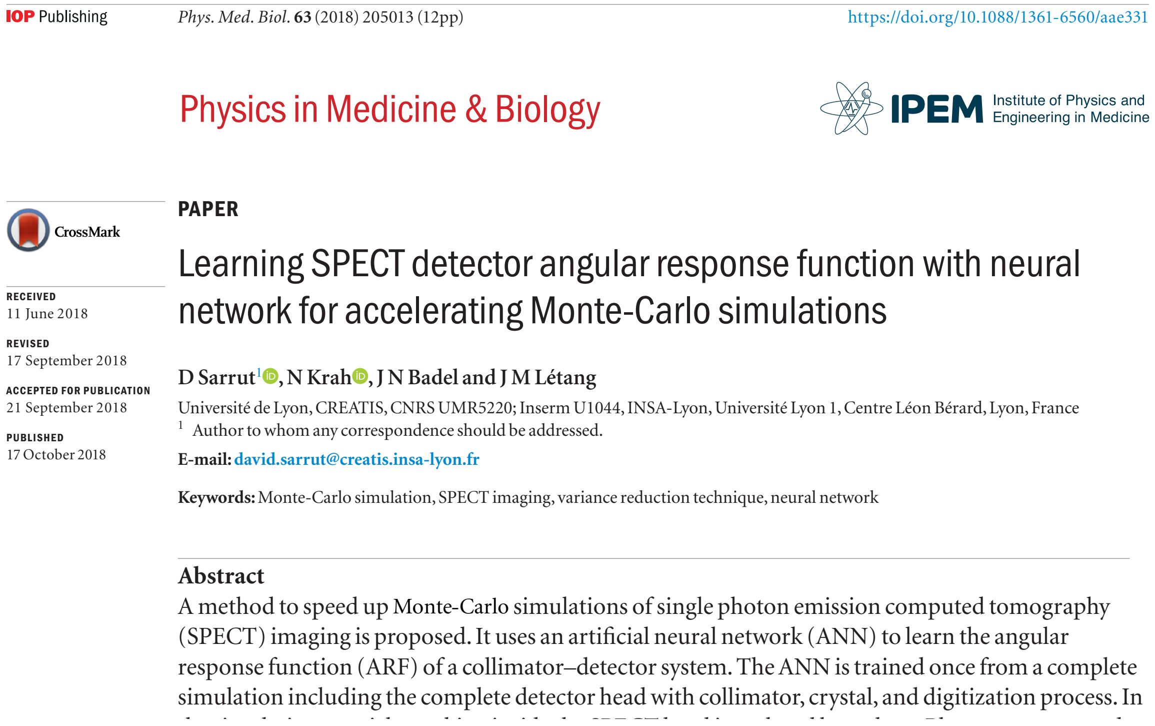
(> x200)



Example 1: conclusion

- Alternative approach to ARF by histogram using Artificial Neural Network
- Similar efficiency, require less data to build, more consistent (binning)
- Different noise distribution, need to add Poisson noise
- Available in GATE (open-source) www.opengatecollaboration.org

« *Learning SPECT detector angular response function with neural network for accelerating Monte-Carlo simulations* »
D. Sarrut, N. Krah, JN. Badel, JM. Létang, **Physics in Medicine and Biology**, 2018



IOP Publishing — Phys. Med. Biol. **63** (2018) 205013 (12pp) — https://doi.org/10.1088/1361-6560/aae331

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

CrossMark

RECEIVED
11 June 2018

REVISED
17 September 2018

ACCEPTED FOR PUBLICATION
21 September 2018

PUBLISHED
17 October 2018

PAPER

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

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

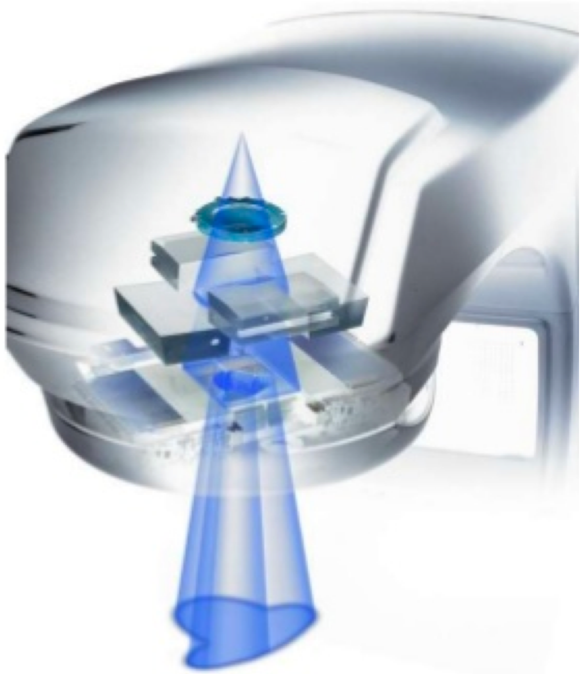
Keywords: Monte-Carlo simulation, SPECT imaging, variance reduction technique, neural network

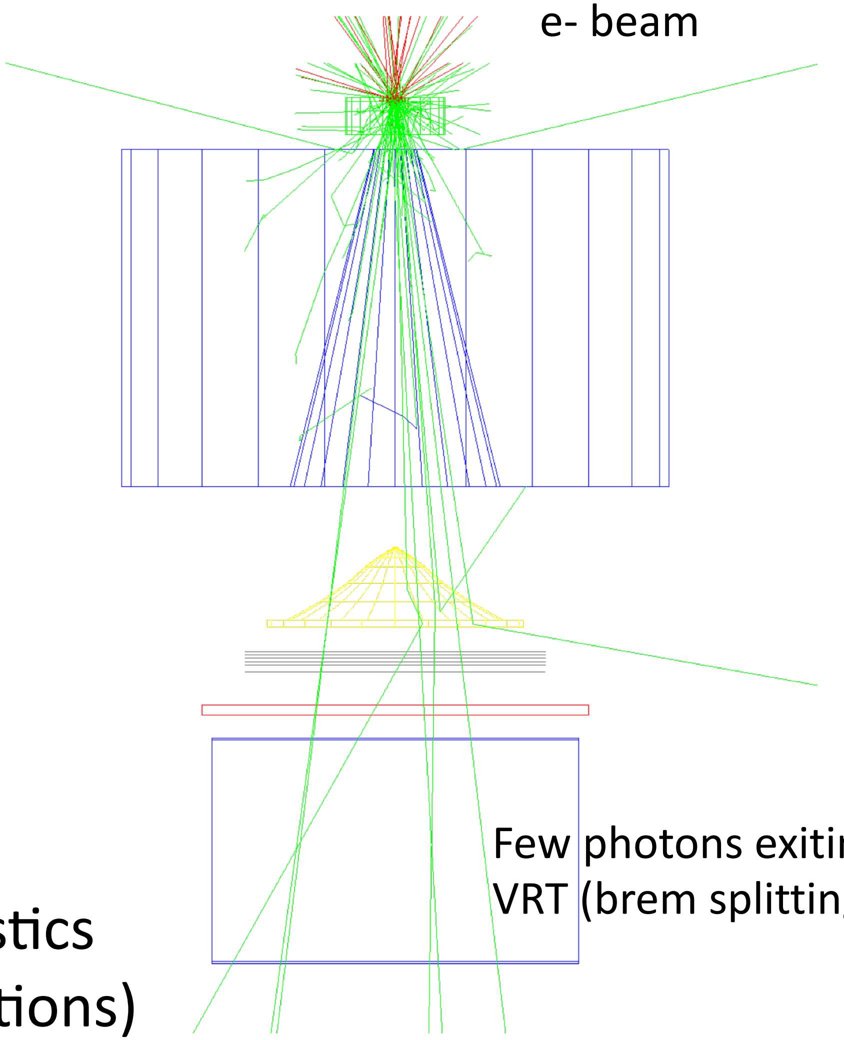
Abstract

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

Example 2:
learning Linac phase-space

Radiation Therapy Linac head simulation





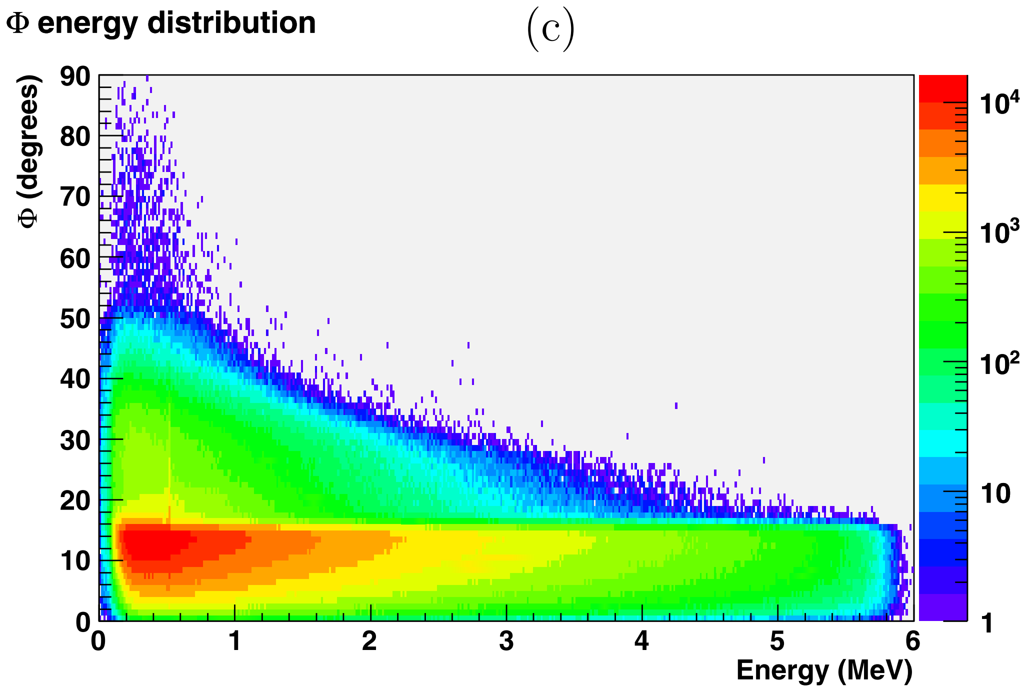
Goal: determine beam characteristics
(energy, position, direction distributions)

Phase Space (PHSP)

- Store beam properties as **Phase Space**
  - A PHSP is a list of particles (around 1e8, 1e9)
  - Properties: E, x, y, z, dx, dy, dz, w

- Advantages:
  - Computed only once
  - Fast to use
  - Can be shared

- Drawback
  - Several GB
  - When a cluster is used, should be shared among workers
  - Limited number of particles



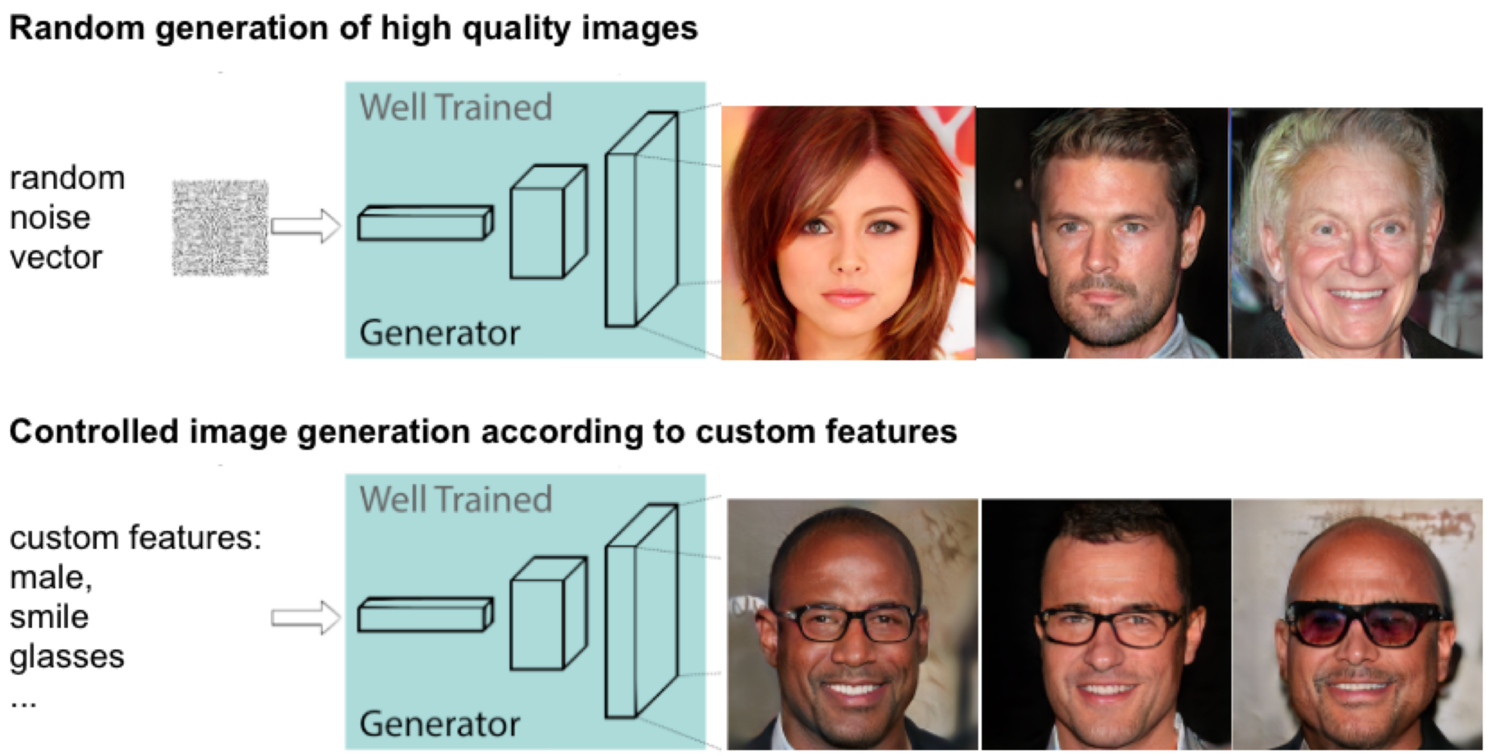
Example of dependence of direction ϕ and energy.

GAN: Generative Adversarial Network

[Goodfellow, 2014]

Goal: « learn » a multidimensional probability distribution

Initial application :
artificial images generator



https://www.thispersondoesnotexist.com
https://www.thiscatdoesnotexist.com
https://youtu.be/2edOMMREazo?t=37

GAN: Generative Adversarial Network

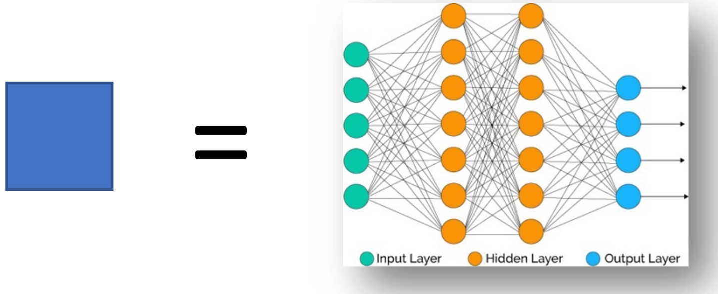
- Training dataset $\boldsymbol{x} \in \mathbb{R}^d$
  - Dimension d=7 (E, X, Y, Z, dX, dY, dZ)
  - Samples of an unknown distribution p_{real}

- Generator $G(\boldsymbol{z}; \boldsymbol{\theta}_G)$

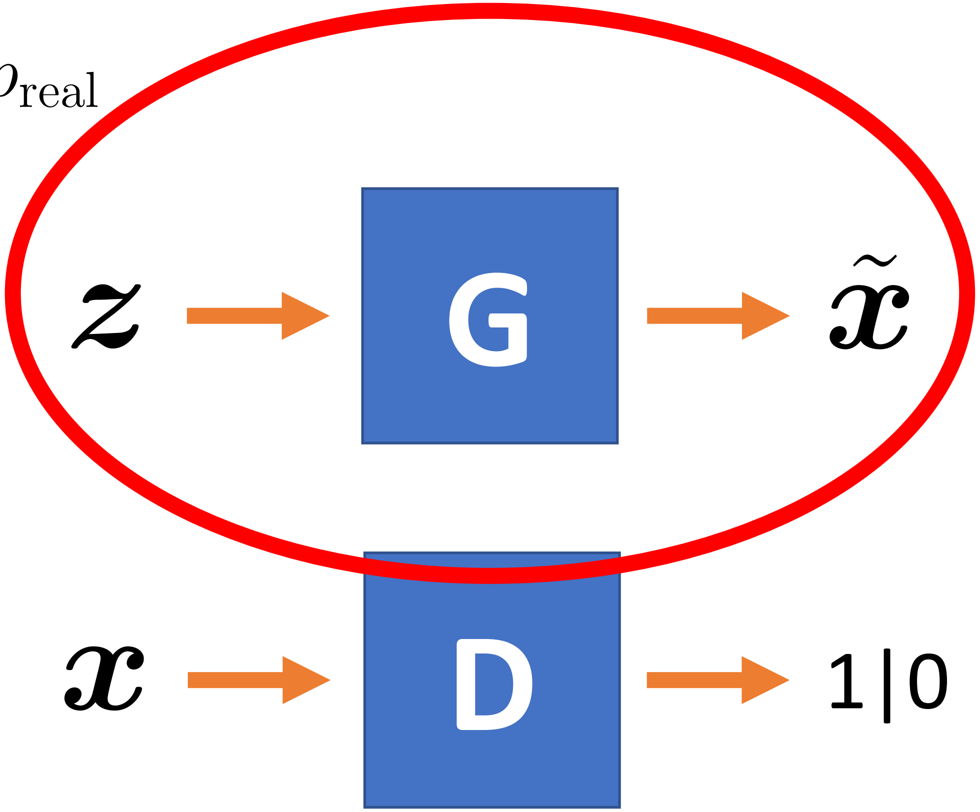
- Discriminator $D(\boldsymbol{x}; \boldsymbol{\theta}_D)$

GAN: Generative Adversarial Network

- Training dataset $\boldsymbol{x} \in \mathbb{R}^d$
  - Dimension d=7 (E, X, Y, Z, dX, dY, dZ)
  - Samples of an unknown distribution p_{real}



Neural network



Alternate G and D optimisation updates

Loss function

- GAN notoriously difficult to train
- Alternative formulations: Wasserstein GAN [Arjovsky 2017]

- "Earth-mover" distance (EMD) : cost of the optimal transport
- Un-tracktable in practice, but approximated:

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

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

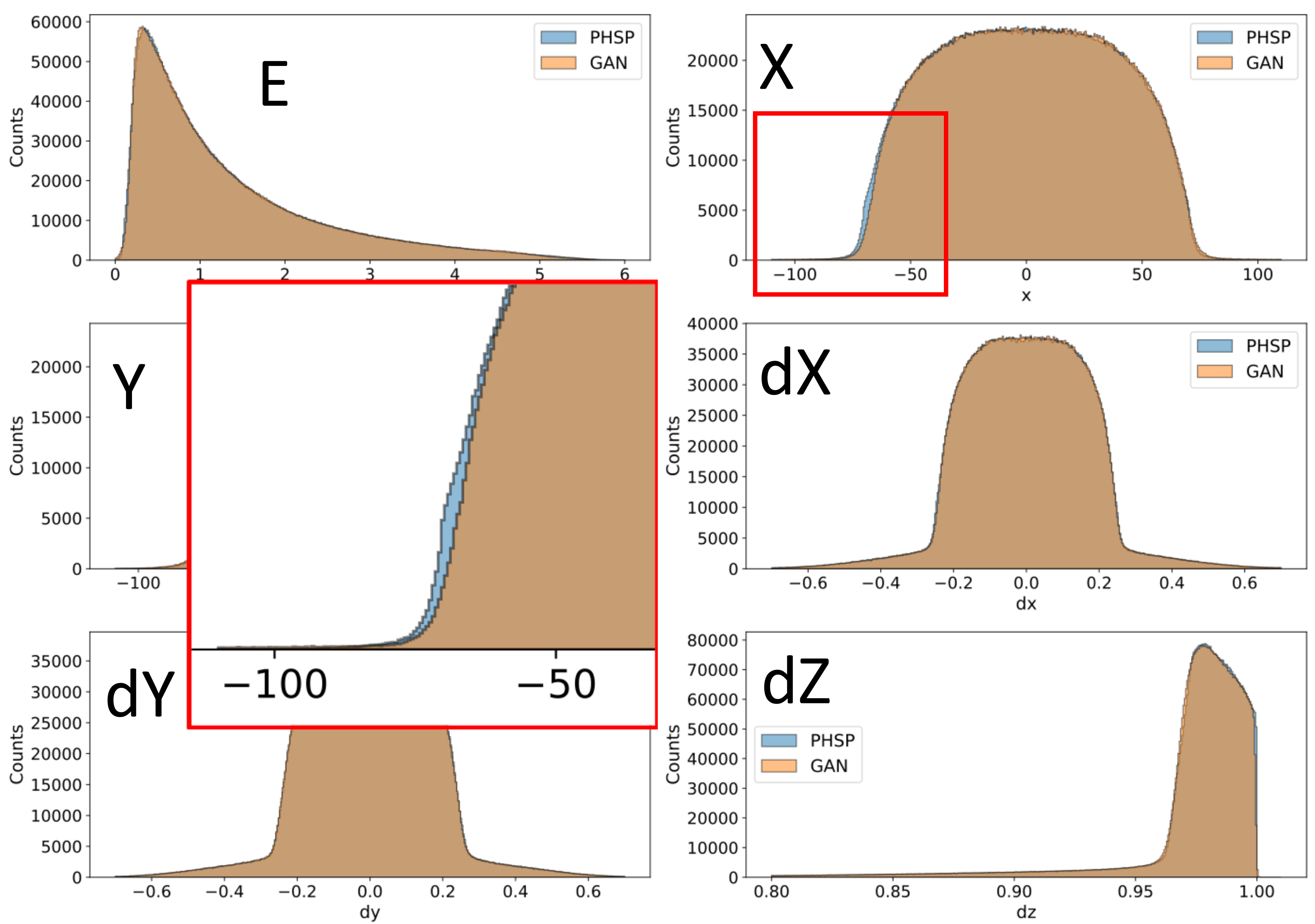
Experiments

PHSP from IAEA web site

PHSP	Size	Nb of particles
Elekta PRECISE 6MV	2 files of 3.9 GB	1.3×10^8 photons each file
CyberKnife IRIS 60mm	2 files of 1.6 GB	5.8×10^7 photons each file

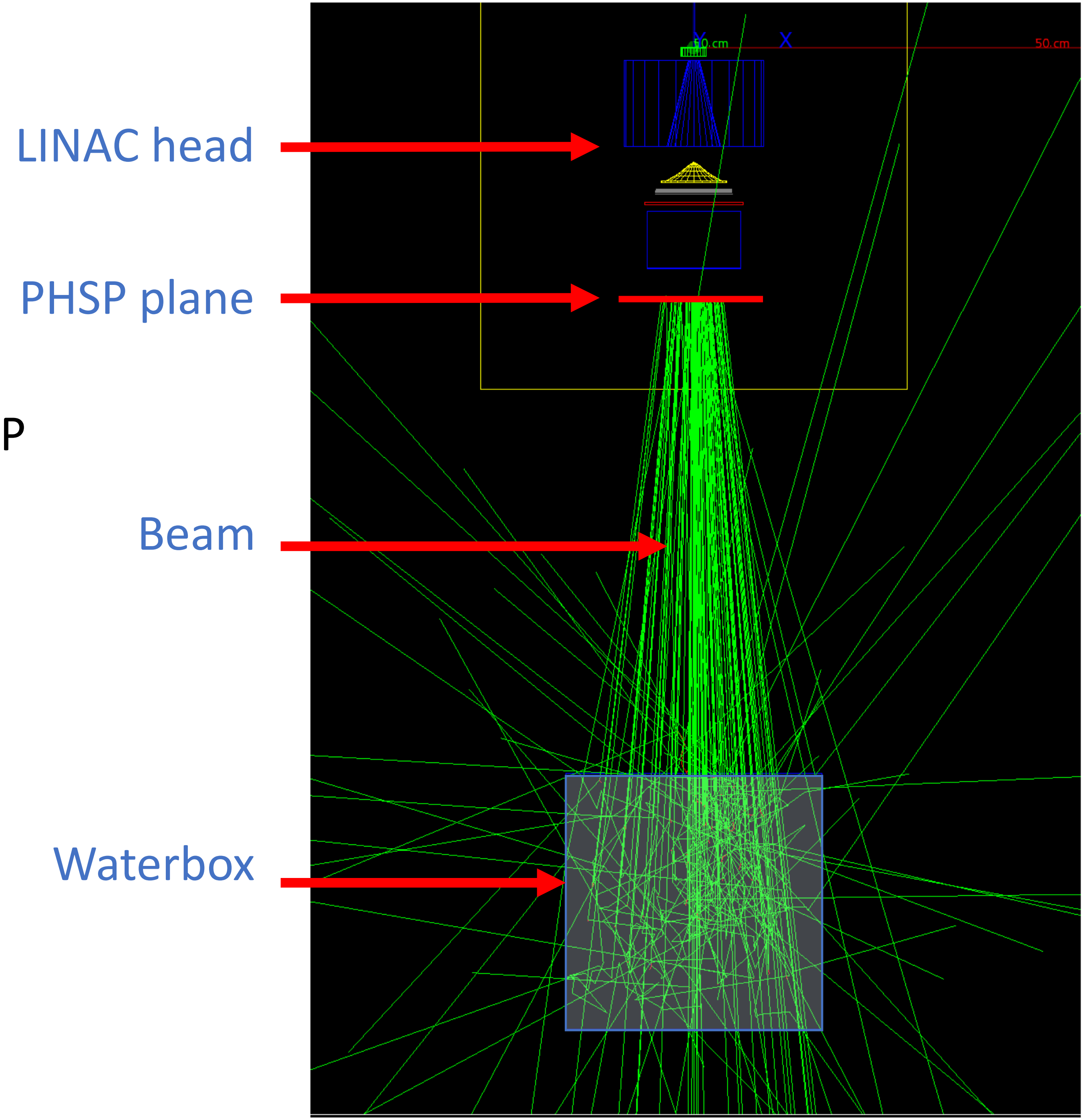
Results

Marginal distributions of the 6 parameters obtained from the reference PHSP and from the GAN, for Elekta 6MV linac.



Results

- Dose distribution in water from PHSP
  10^8 primary photons
- Compare dose between:
  1. PHSP1 vs PHSP2
  2. PHSP1 vs GAN
- Voxel by voxel dose comparison

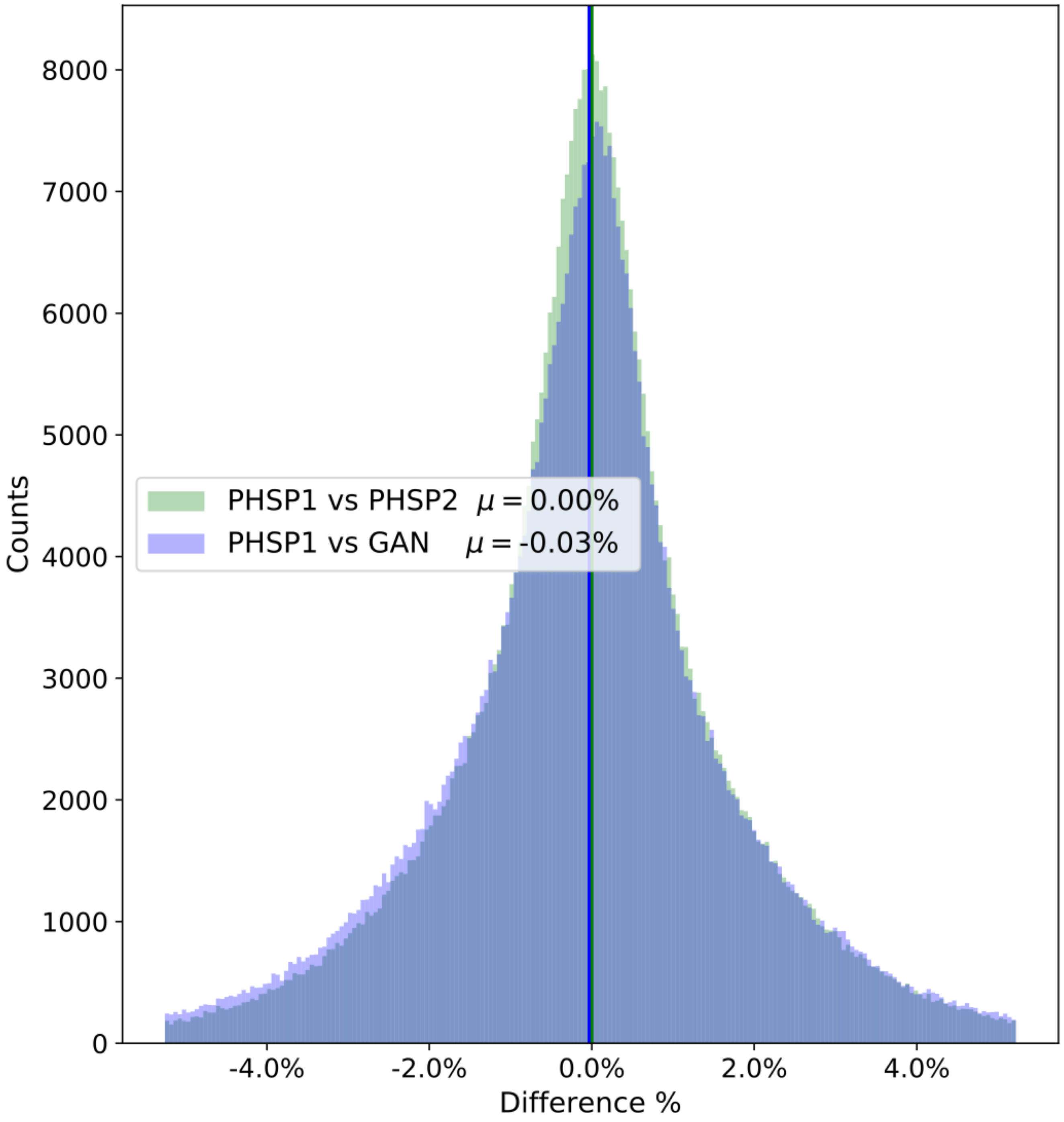


Results

Distributions of relative differences between
- PHSP1 and PHSP2
- PHSP1 and GAN

Vertical lines indicate the mean differences

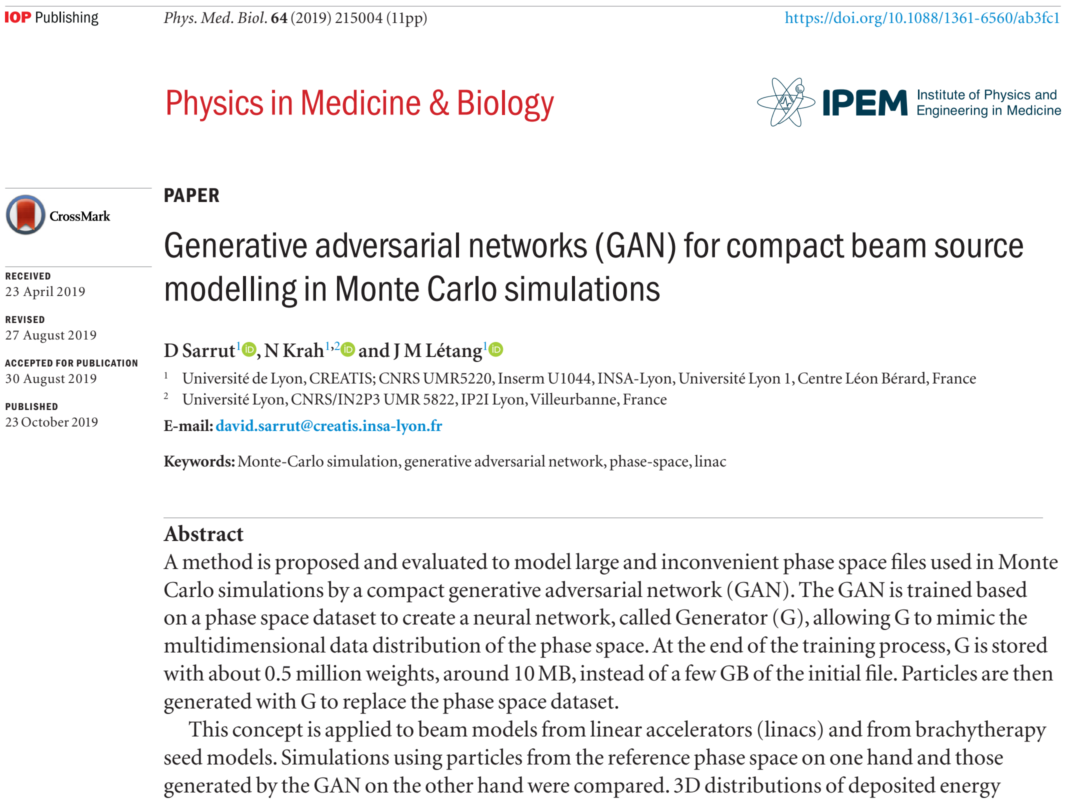
Difference relative to the prescribed dose



Example 2: conclusion

- Using GAN to represent a Phase-Space is feasible
- Final GAN model: few MB (vs PHSP = 4 GB)
- Sufficient for dose computation
- Training is difficult: hyperparameters, 511 keV peak, …
- Available in GATE www.opengatecollaboration.org
- Perspectives :
  - Could it be learned from less particles ?
  - Detailed statistical analysis in progress
  - Other applications of GAN within MC simulations

« *Generative Adversarial Networks (GAN) for compact beam source modelling in Monte Carlo simulations* »
D. Sarrut, N. Krah, JN. Badel, JM. Létang, **Physics in Medicine and Biology**, 2019



IOP Publishing Phys. Med. Biol. 64 (2019) 215004 (11pp) https://doi.org/10.1088/1361-6560/ab3fc1

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

CrossMark

RECEIVED
23 April 2019

REVISED
27 August 2019

ACCEPTED FOR PUBLICATION
30 August 2019

PUBLISHED
23 October 2019

PAPER

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

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

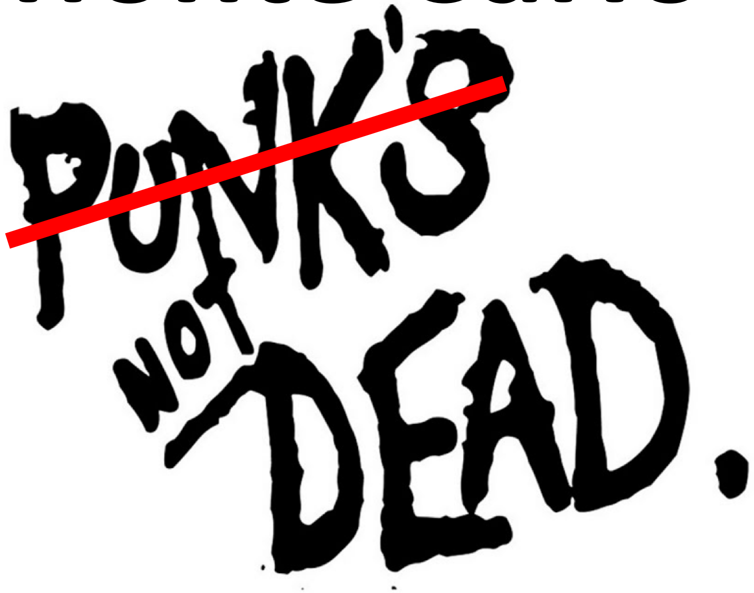
A method is proposed and evaluated to model large and inconvenient phase space files used in Monte Carlo simulations by a compact generative adversarial network (GAN). The GAN is trained based on a phase space dataset to create a neural network, called Generator (G), allowing G to mimic the multidimensional data distribution of the phase space. At the end of the training process, G is stored with about 0.5 million weights, around 10 MB, instead of a few GB of the initial file. Particles are then generated with G to replace the phase space dataset.

This concept is applied to beam models from linear accelerators (linacs) and from brachytherapy seed models. Simulations using particles from the reference phase space on one hand and those generated by the GAN on the other hand were compared. 3D distributions of deposited energy

General conclusion

- **AI** may (also) be useful with **MC**
  - ARF, GAN for phase-space, …
  - Faster, smoother, stronger
- Still experimental, currently under heavy investigations
- New challenges
  - Learning dataset size ?
  - Learning time ?
  - Convergence guarantee ?
  - Final Accuracy ?

Monte Carlo



Thanks for your attention !

CREATIS Lyon, France

GATE opengatecollaboration.org

LyriCAN

cnrs dépasser les frontières

PRIMES UNIVERSITE DE LYON

CENTRE DE LUTTE CONTRE LE CANCER LEON BERARD