AI meets Physics

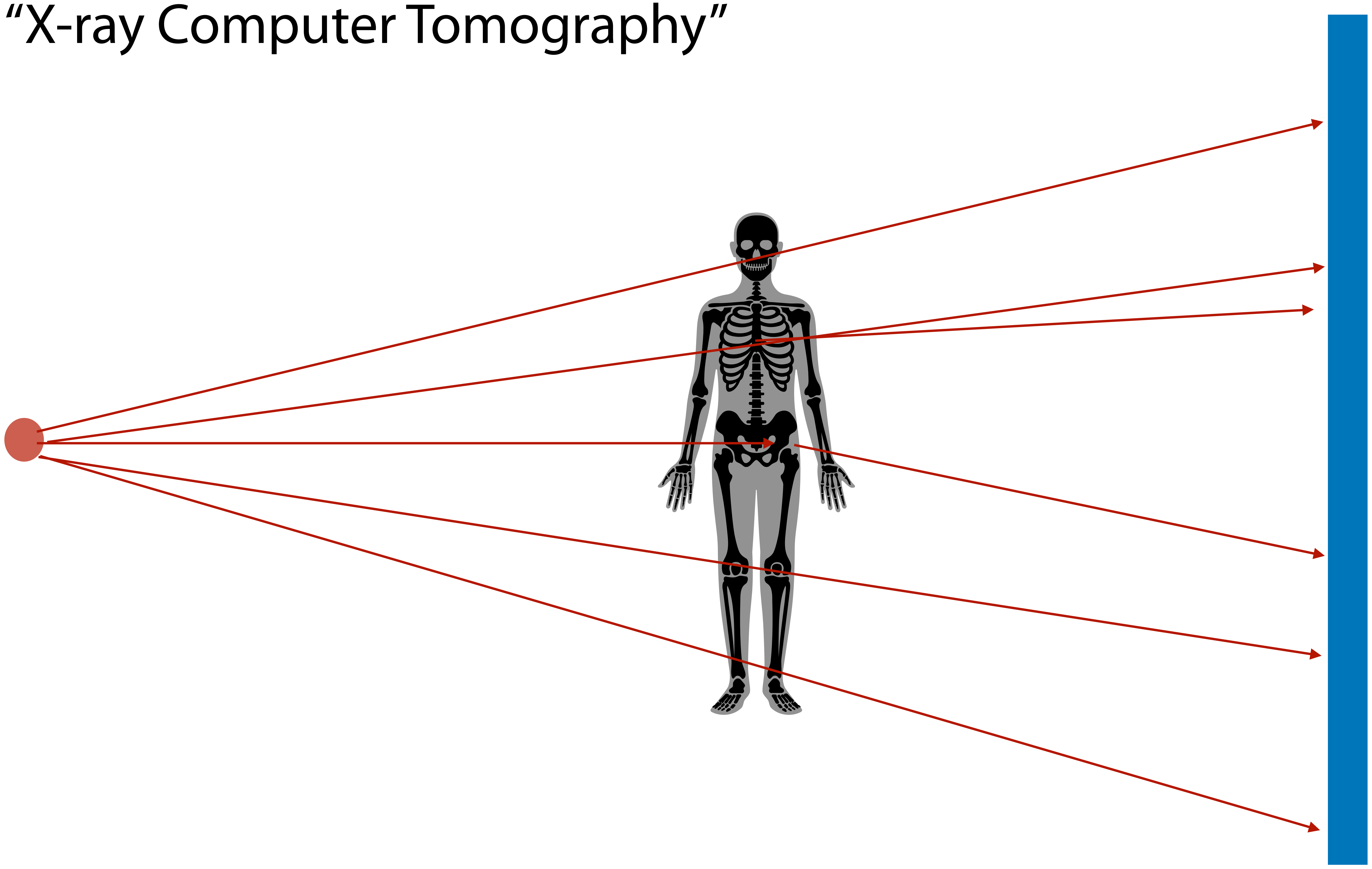
an overview of AI applications in the context of Monte Carlo particle transport simulations (in medical imaging and medical physics)

Nils Krah, INSA Lyon, CNRS, Léon Bérard Cancer Centre, France

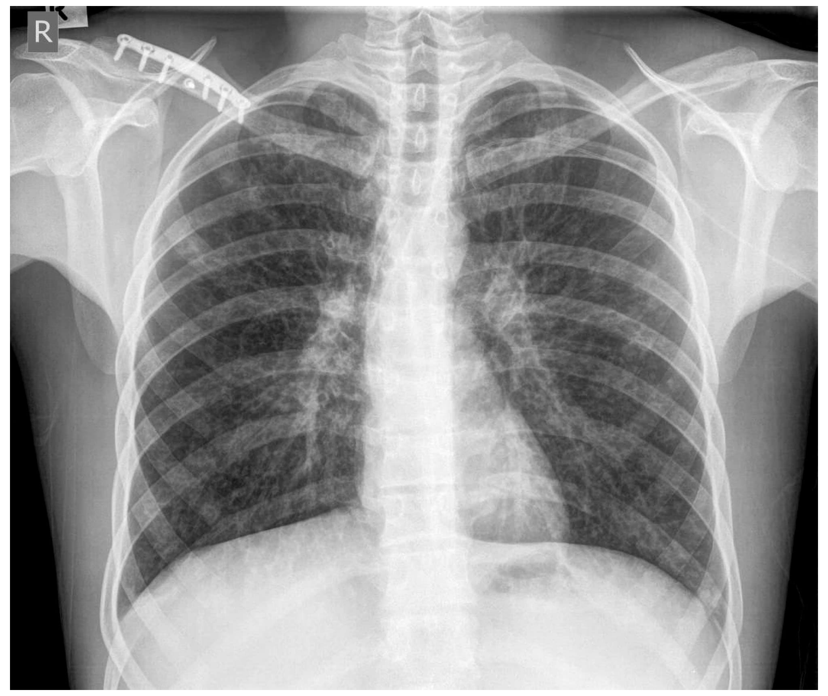
Context: Medical Imaging

Example: x-rays (gammas) traversing a human body

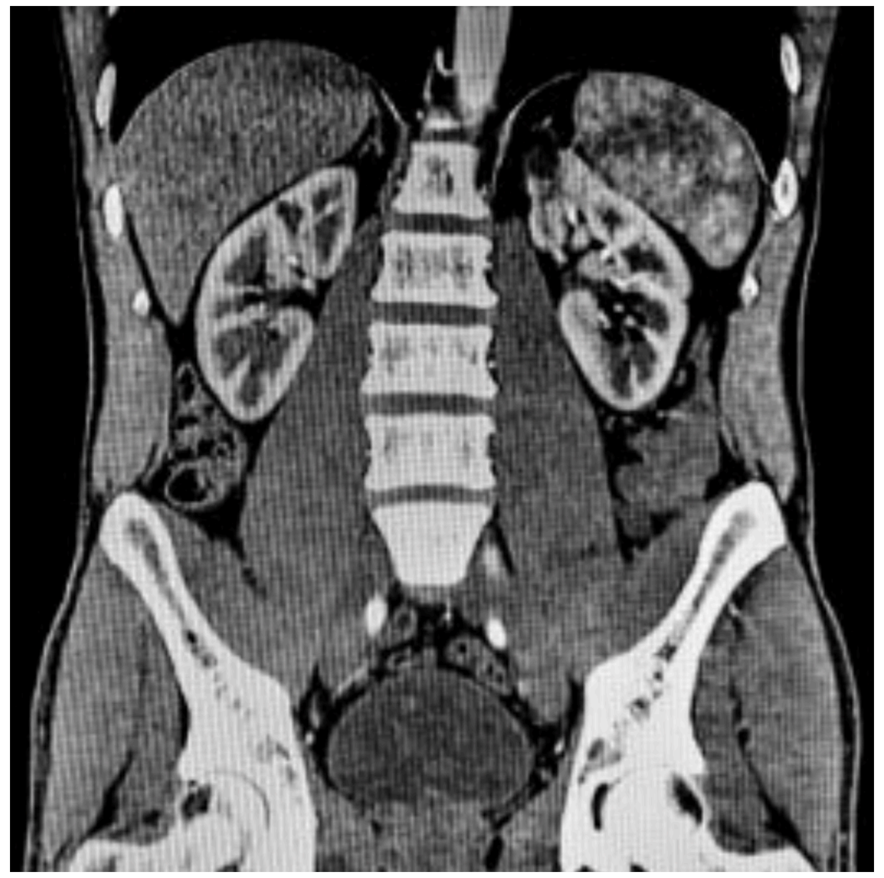
“X-ray Computer Tomography”



Intensity mage

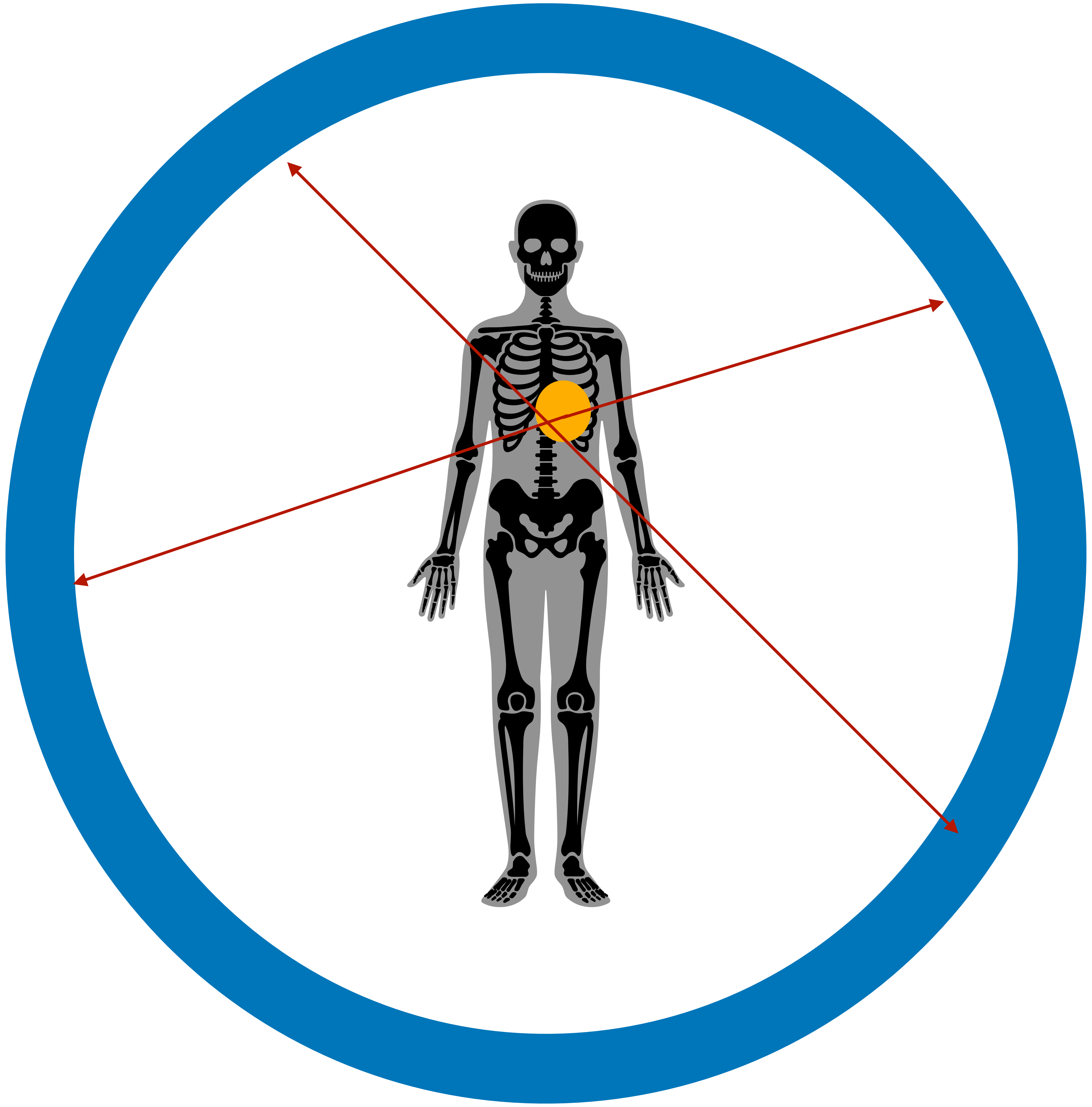


Reconstruct volumetric map of the attenuation coefficient

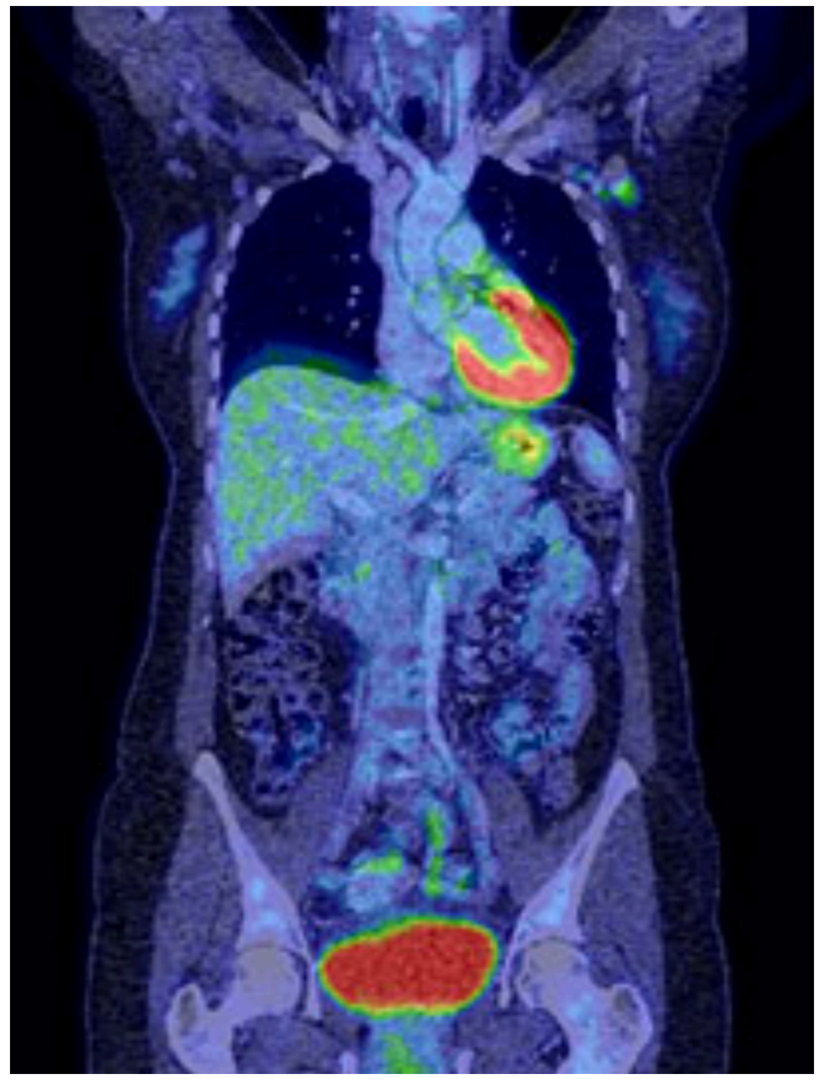


Context: Medical Imaging

- Example: gammas emitted from a body



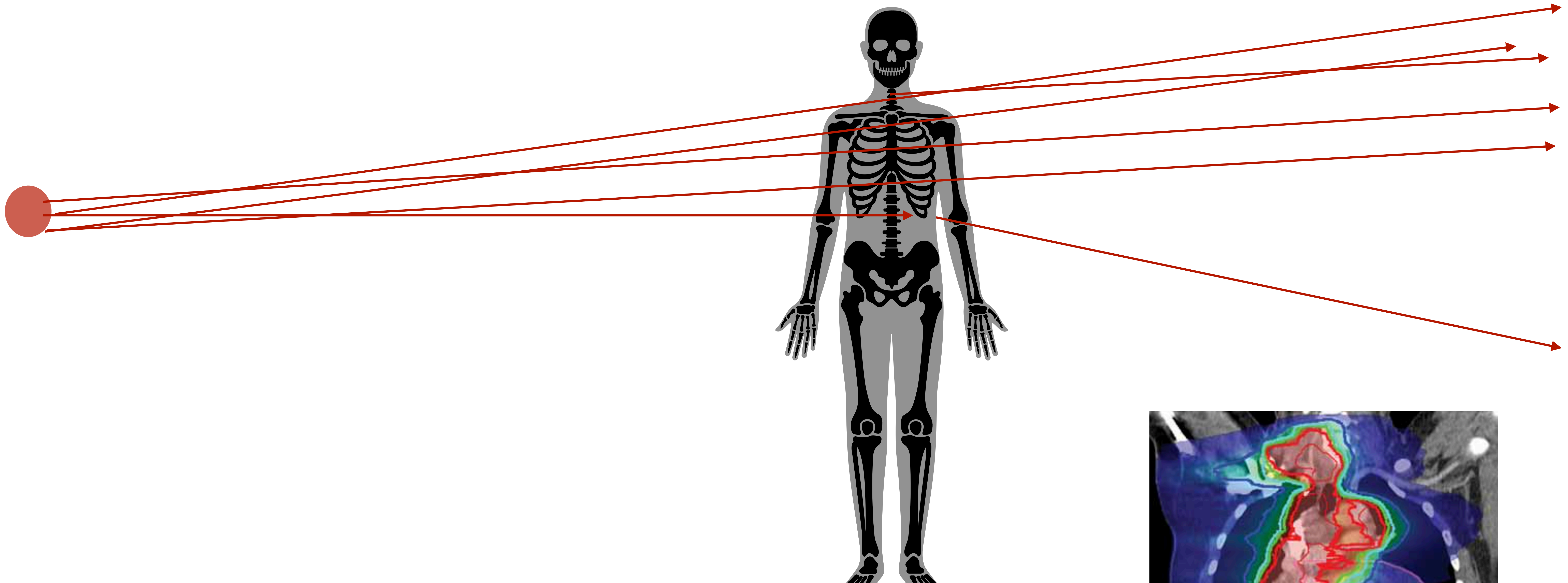
- Positron Emission Tomography (PET)
- Single Photon Emission Computed Tomography (SPECT)



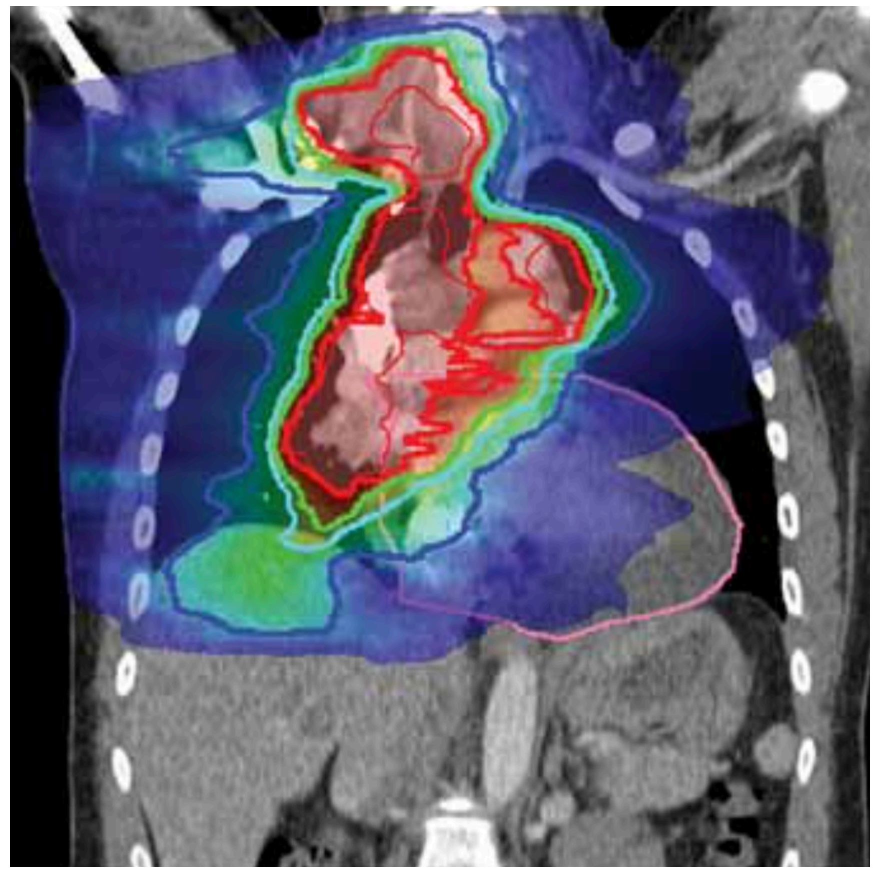
Reconstruction:
Source distribution

Context: External Radiotherapy

Ionizing radiation penetrating through/into a human body

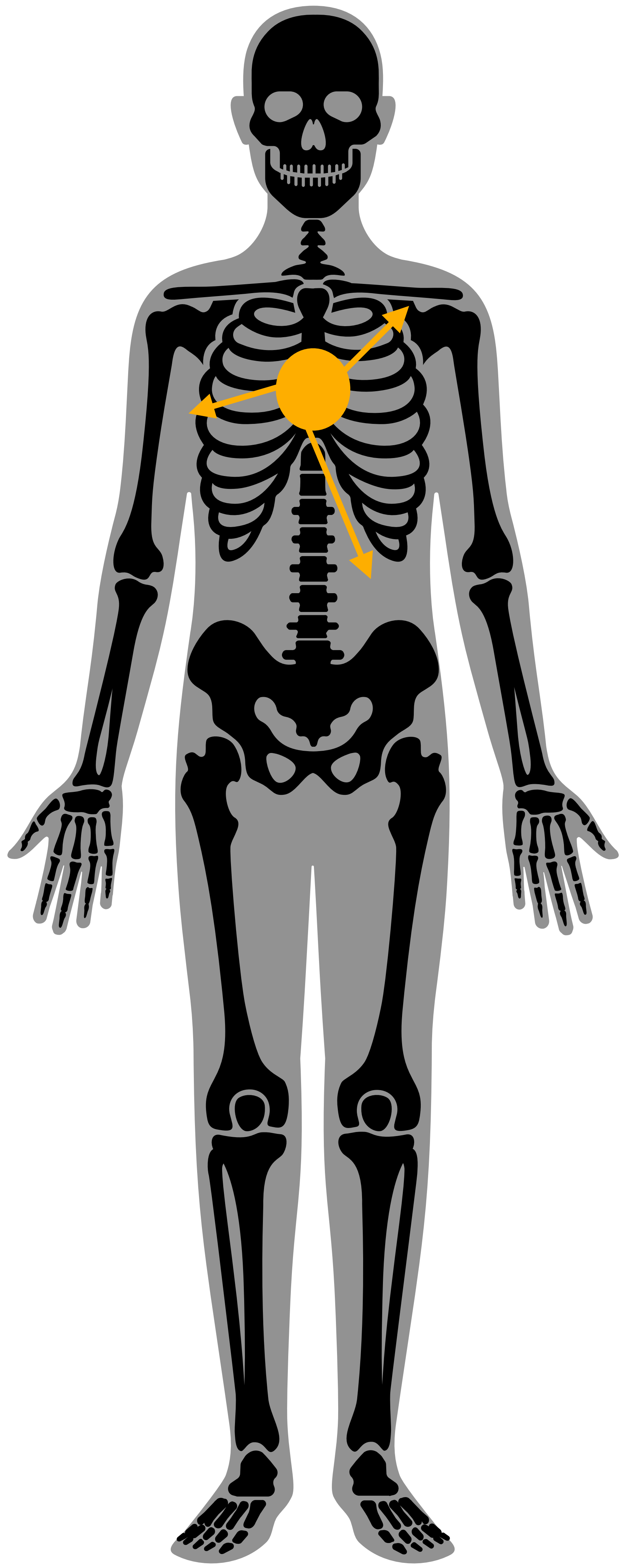


Goal: expose tumor to high dose, but
spare healthy tissue as much as possible.

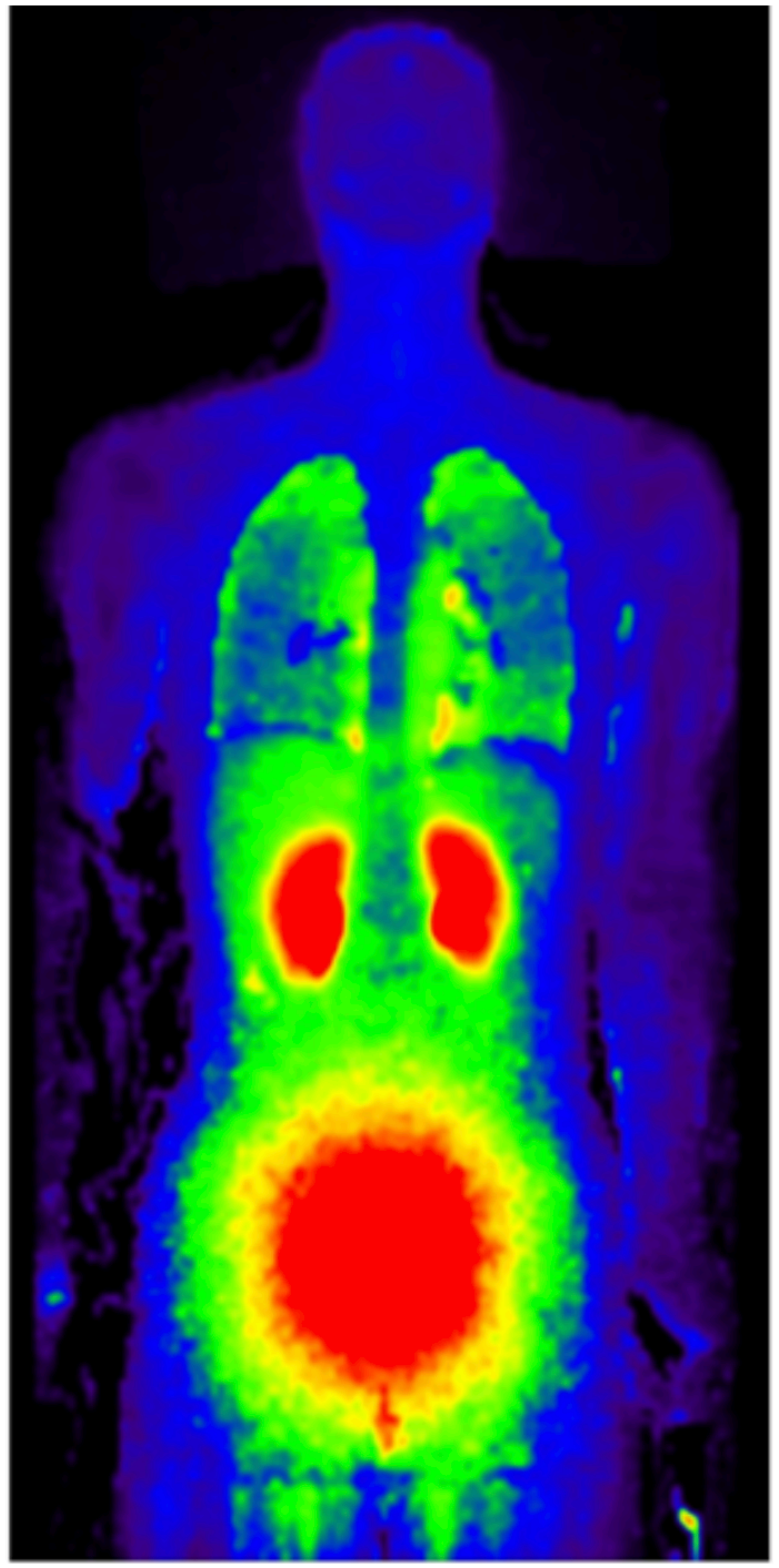


Context: Nuclear Medicine

- Ionizing radiation created inside a human body



Expose organs/cells affected by cancer to radiation dose.



Why Monte Carlo simulation?

In medical imaging:

- Design and optimize new imaging devices
- Generate data to test reconstruction algorithms
- Simulate scatter contribution
- Simulate attenuation maps in nuclear imaging

In external radiotherapy:

- Predict dose maps
- Recalculate treatments for quality assurance

In nuclear medicine:

- Predict dose

Examples of Monte Carlo codes

Multi-purpose:

- Geant4, FLUKA, MCNP, …

Applications built on top of Geant4:

- TOPAS
- GATE version 9.x
- new GATE 10

Application/physics specific:

- EGS (Electron, photon)
- Penelope (electron, photon)
- MCsquare (proton therapy)
- FRED (mainly fast dose calculation)
- GGEMS
- …

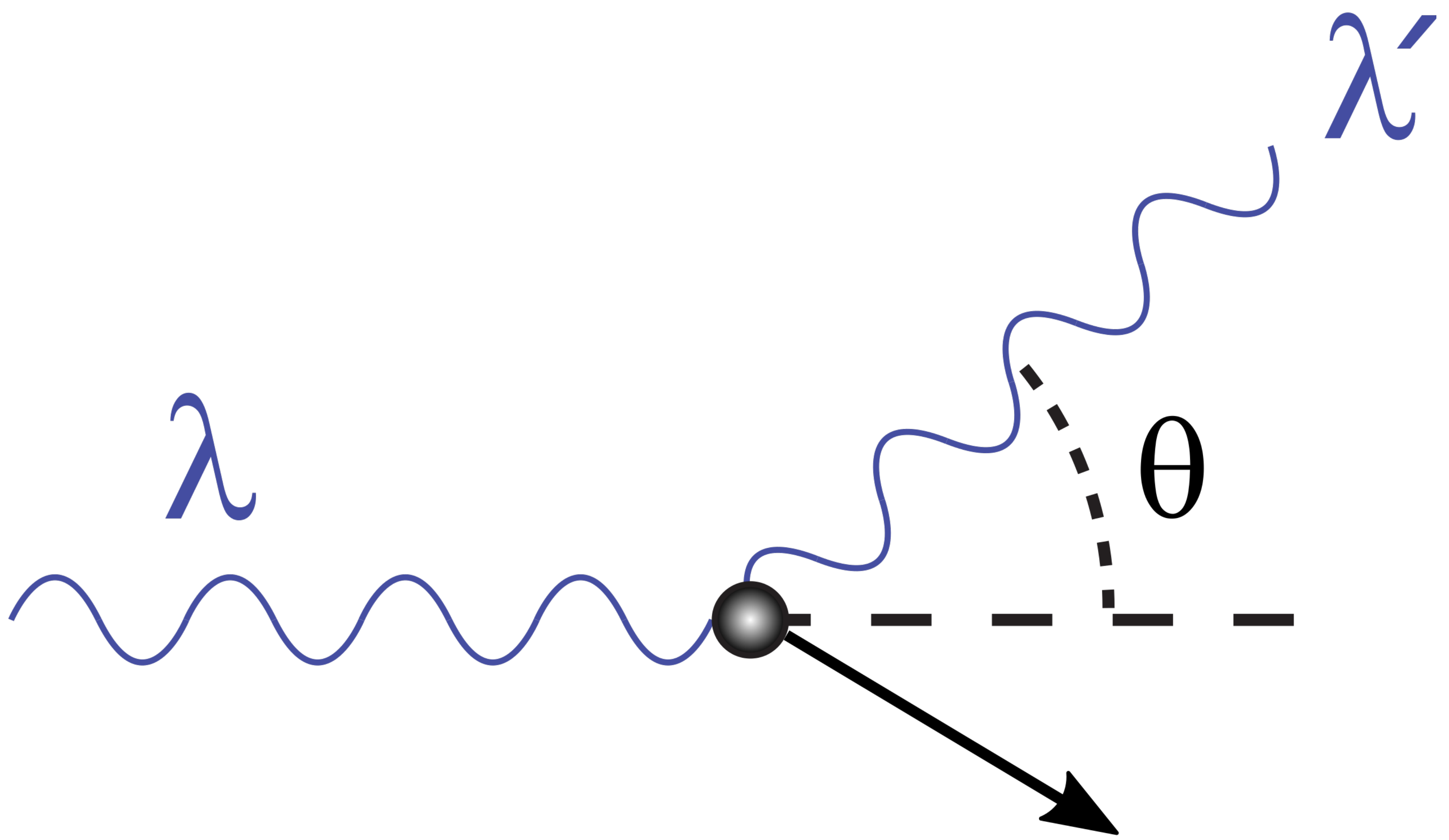
Physics perspective

- Particles traveling across a heterogeneous distribution of matter
- Undergo interactions with the matter

- Which interaction depends on material and projectile properties
- Result of interaction is stochastic

“Interaction”: Physics perspective

Example of interaction: Photon scattering off an electron - Compton scattering



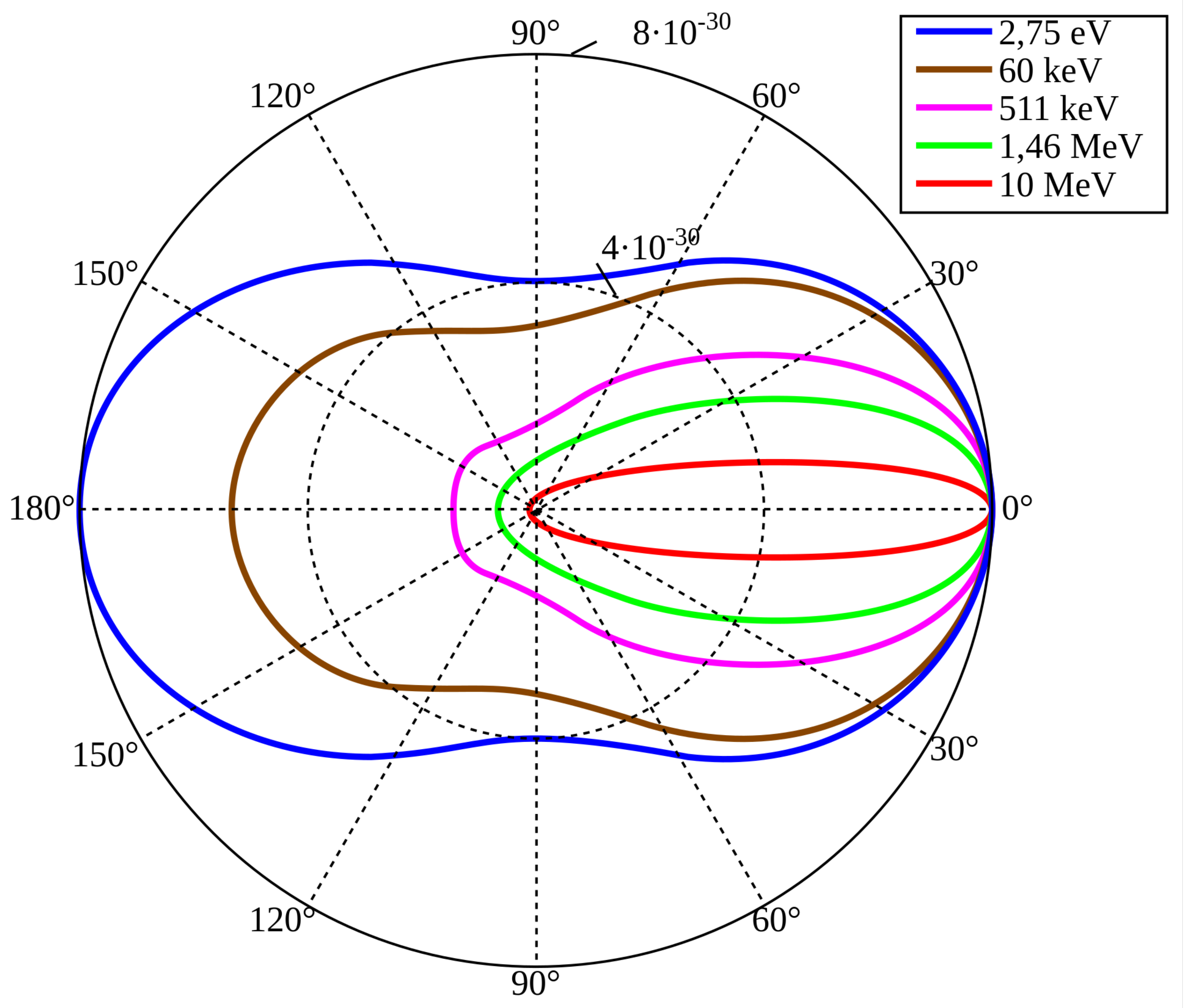
- Photon changes energy (wavelength) and direction
- Transfers some energy to target

"Interaction": Statistics perspective

Example of interaction: Compton scattering

Likelihood of a photon being scattered:

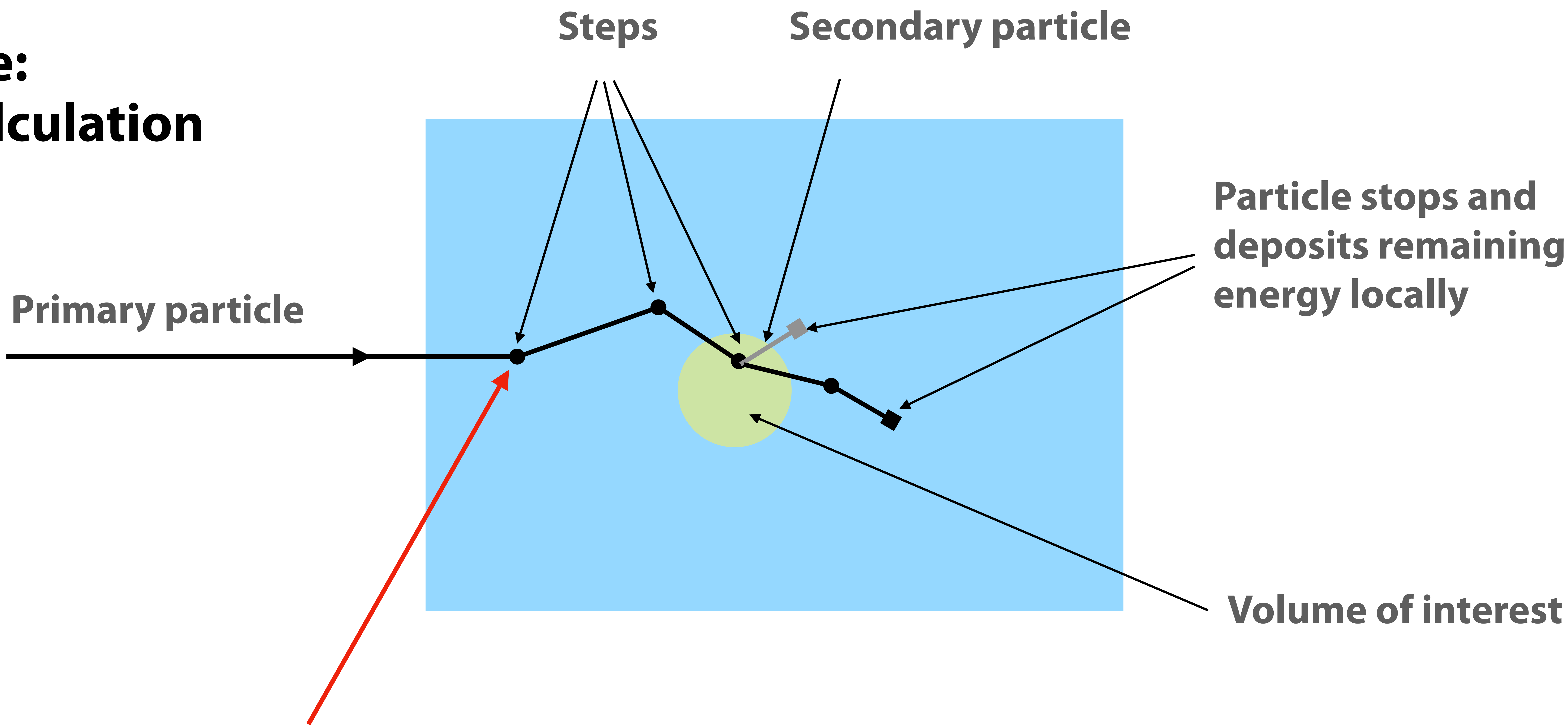
$$\frac{1}{2} r_e^2 \left(\frac{\lambda}{\lambda'}\right)^2 \left[\frac{\lambda}{\lambda'} + \frac{\lambda'}{\lambda} - \sin^2(\theta)\right]$$



Particle transport simulation

Step-wise propagation of a particle across a medium

Example:
Dose calculation

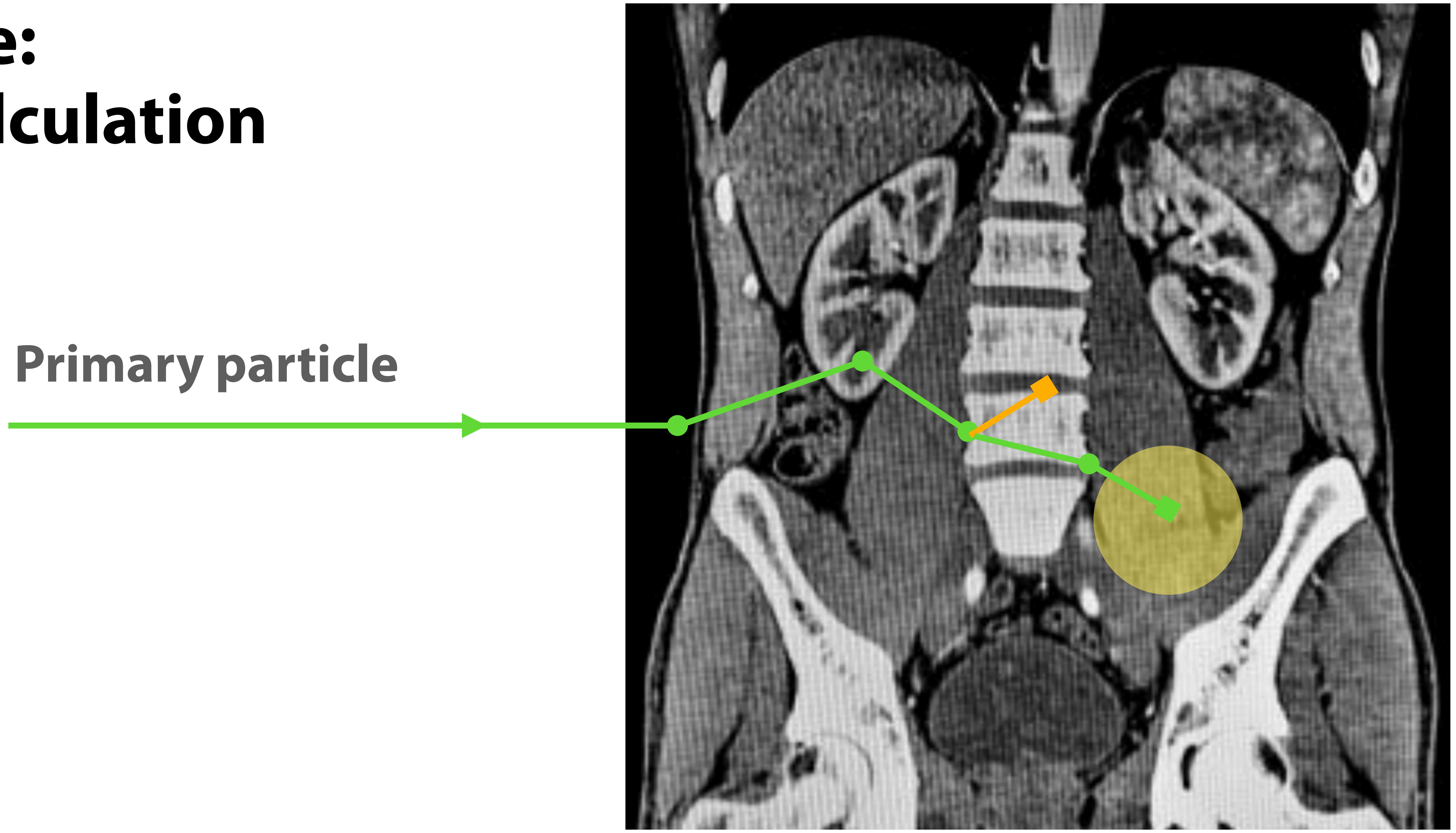


At each step: Evaluate and apply interaction models

Particle transport simulation

Patient = complex heterogeneous geometry

Example:
Dose calculation



Patient geometry usually parametrized via 3D discretized image: x-ray CT image

Ingredients of a Monte Carlo Simulation

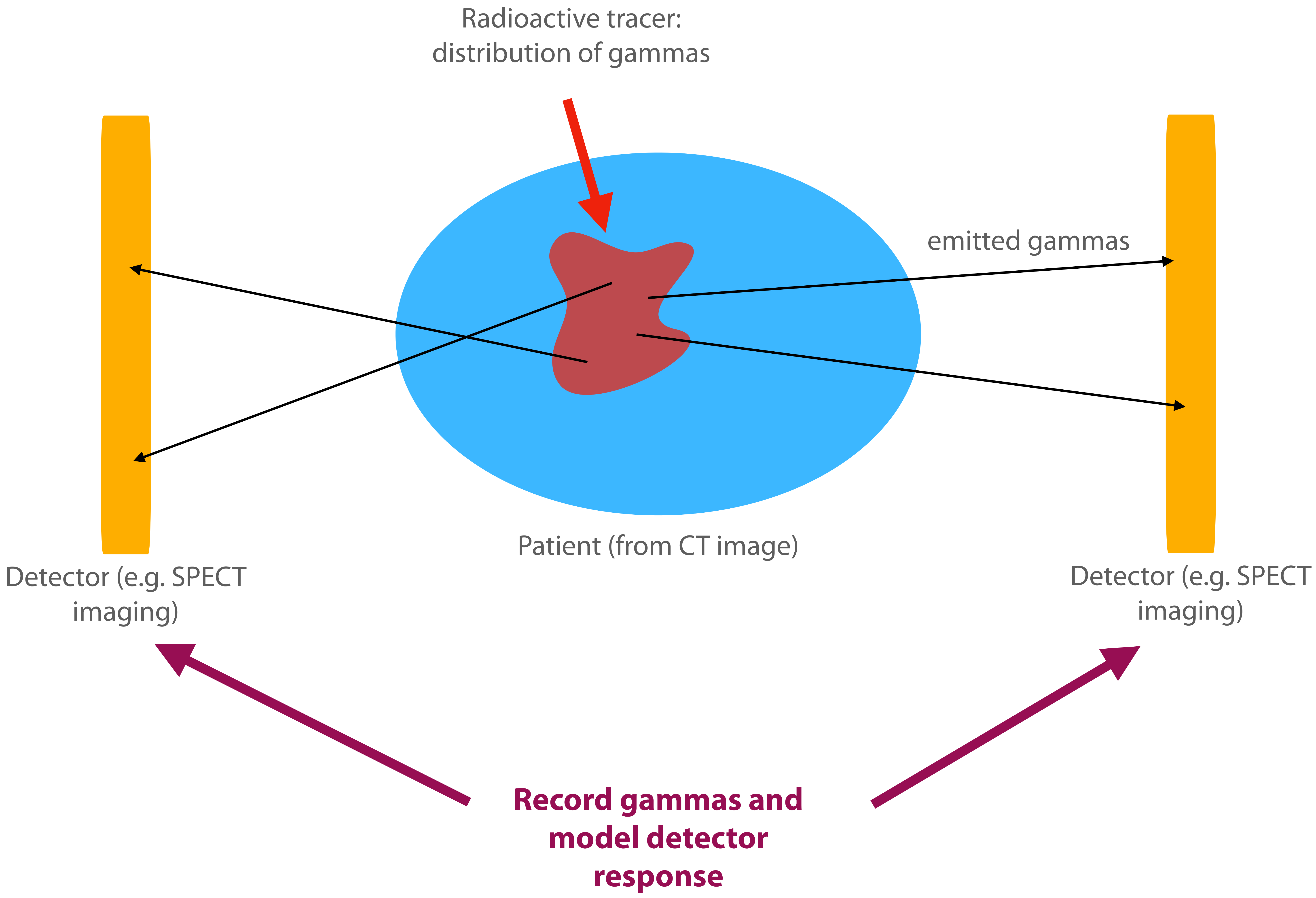
Source (gammas, ions, …)

Geometry (objects, beamline, patient …

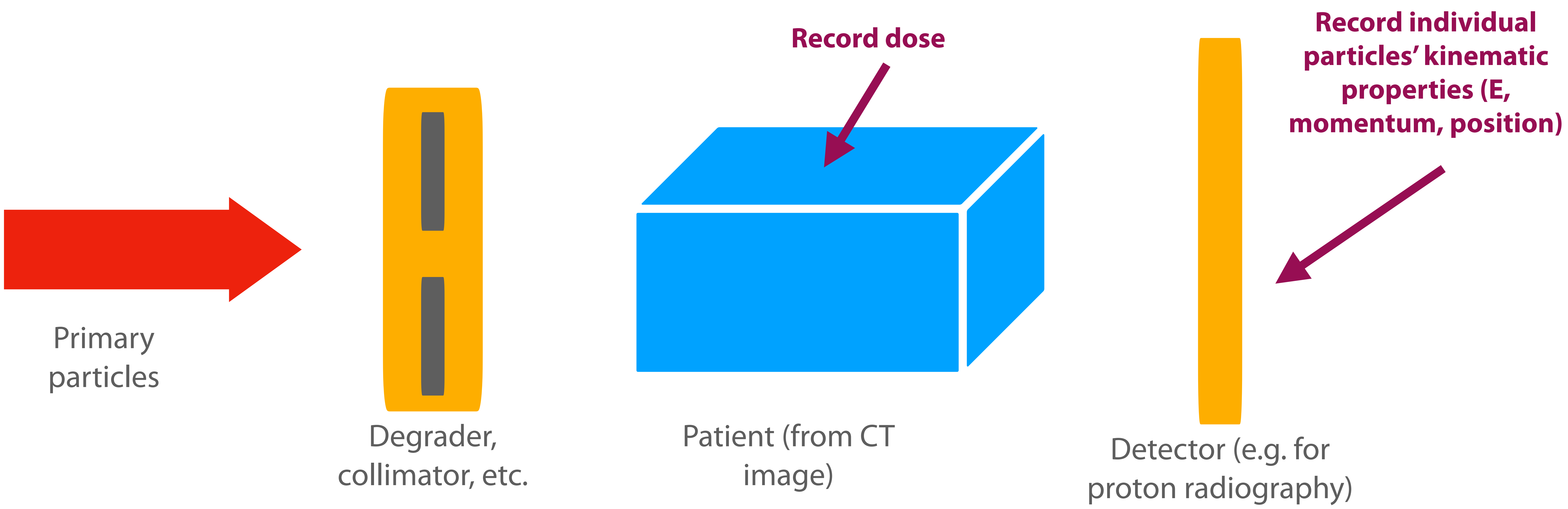
Physics (interactions of particles with the target, nuclear decay)

Output information about physics (dose, particle distribution, detector signal)

Schematic example of "Monte Carlo Simulation" …

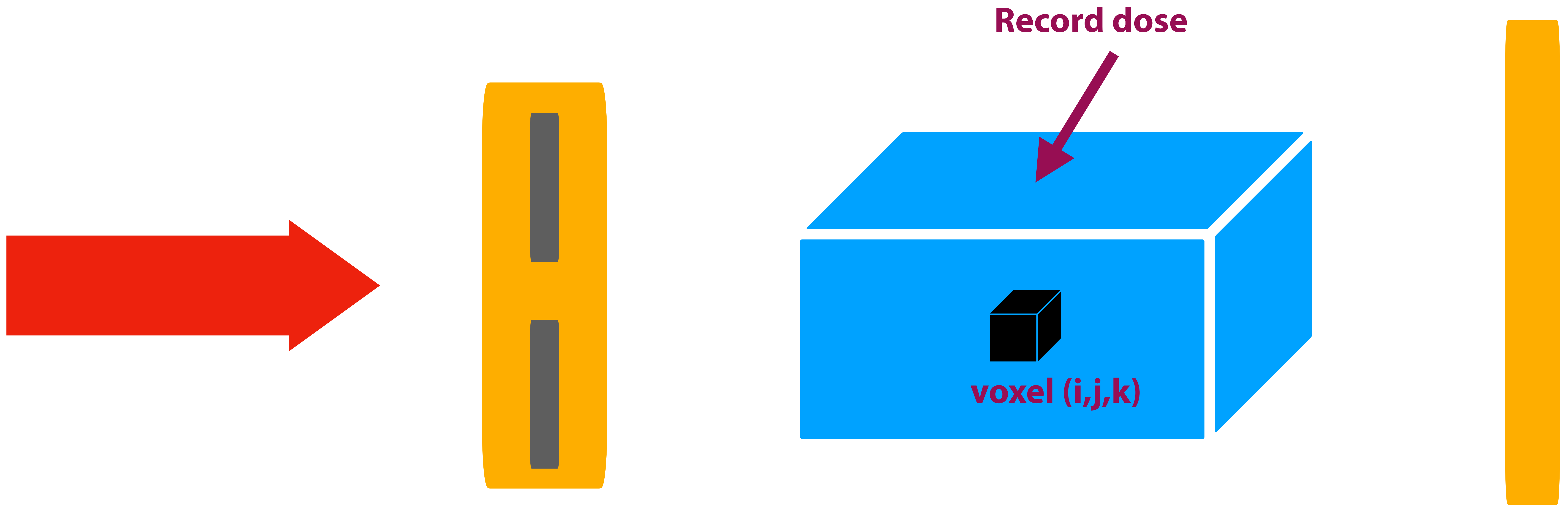


Example of "Monte Carlo Simulation" …



Monte Carlo simulations
(in medical physics/imaging)
are used to
sample probability distributions.

MC samples a probability distribution



Dose D in voxel i,j,k is:

$$D_{i,j,k} \propto \int p(\epsilon, \vec{x}_{i,j,k})\epsilon \mathrm{d}\epsilon$$

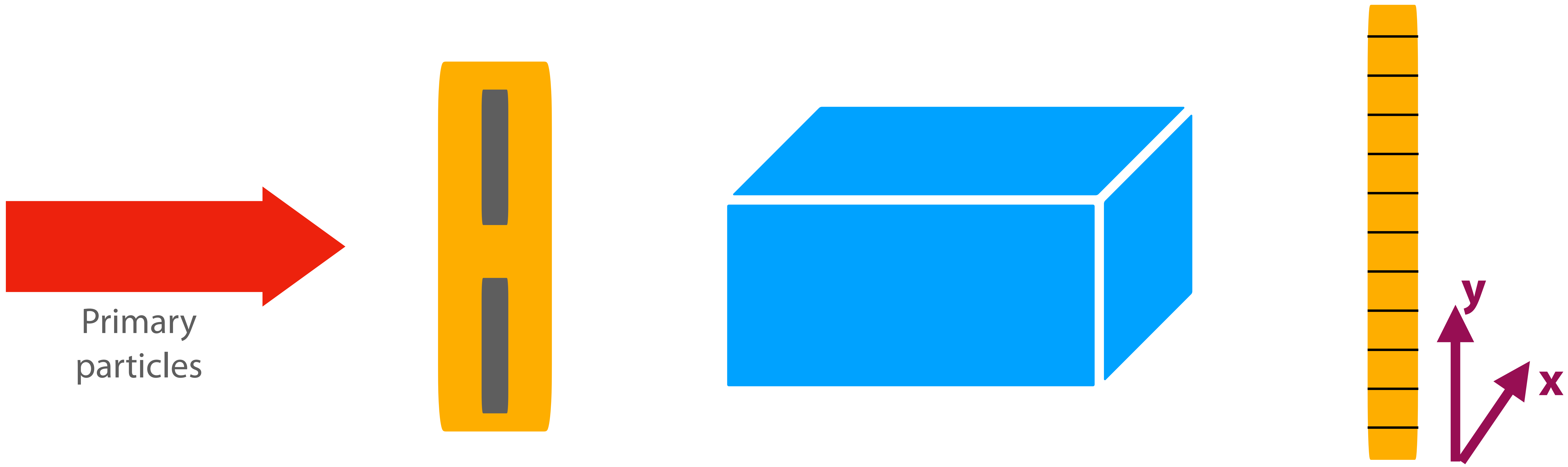
with ϵ deposited energy

Sampled by transporting N primary particles

→ Precision of D improves with 1/sqrt(N)

→ More particles = lower noise

MC samples a probability distribution



Projected image is obtained by binning:

$$I(x, y) \propto \int p(E, x, y)\, w(E)\, \mathrm{d}E$$

Sampled by transporting N primary particles

with E the particle's energy

and $w(E)$ the detector response

More particles = lower noise

Simulated images are spatially discretized probability distributions

where

1 image represents 1 distribution

Pros and cons of (multi-purpose) Monte Carlo

Accurate and “complete” physics

Versatile applications

+

Computationally intensive

−

Monte Carlo simulations can generate training datasets (images)

How to get around the “slowness”?

- Strip down and simplify physics (models, cuts)
- Implement task-specific optimized code
- Use parallel hardware; GPU
- Replace parts of the tracking by fast simulation
- Replace Monte Carlo completely by effective model
- Enhance low-statistics Monte Carlo output

How to get around the “slowness”?

- Strip down and simplify physics (models, cuts)
- Implement task-specific optimized code
- Use parallel hardware; GPU → next talk
- Replace parts of the tracking by fast simulation
- Replace Monte Carlo completely by effective model
- Enhance low-statistics Monte Carlo output

How to get around the “slowness”?

- Strip down and simplify physics (models, cuts)
- Implement task-specific optimized code
- Use parallel hardware; GPU → next talk
- Replace parts of the tracking by fast simulation → presented yesterday
- Replace Monte Carlo completely by effective model
- Enhance low-statistics Monte Carlo output

How to get around the "slowness"?

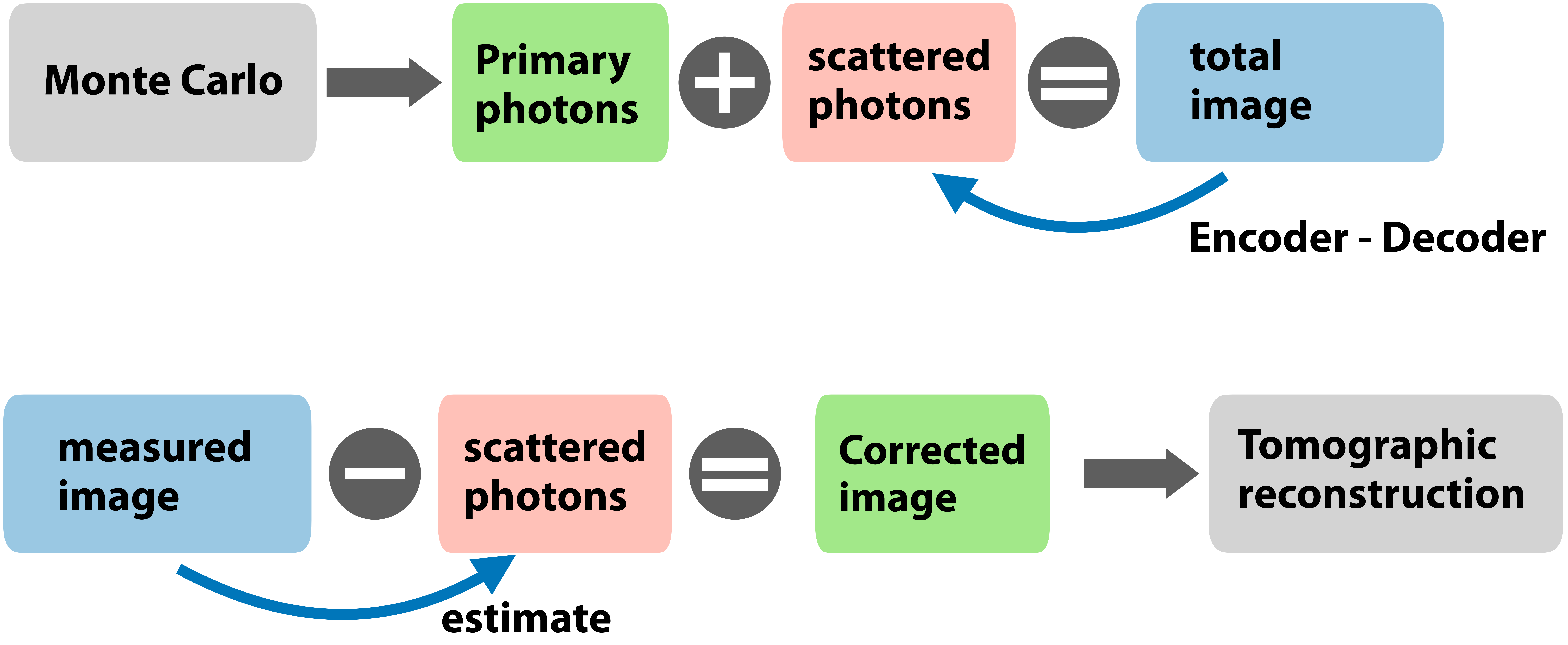
- Strip down and simplify physics (models, cuts)
- Implement task-specific optimized code
- Use parallel hardware; GPU → next talk
- Replace parts of the tracking by fast simulation → presented yesterday
- Replace Monte Carlo completely by effective model
- Enhance low-statistics Monte Carlo output

→ this talk

Scatter estimation/removal

- Scatter contribution in projections reduces image quality in SPECT, PET, CBCT
- Scatter can be simulated via Monte Carlo, but it is slow
- Idea: Train network to estimate scatter images from projections
- … and replace the Monte Carlo simulation

Scatter estimation/removal

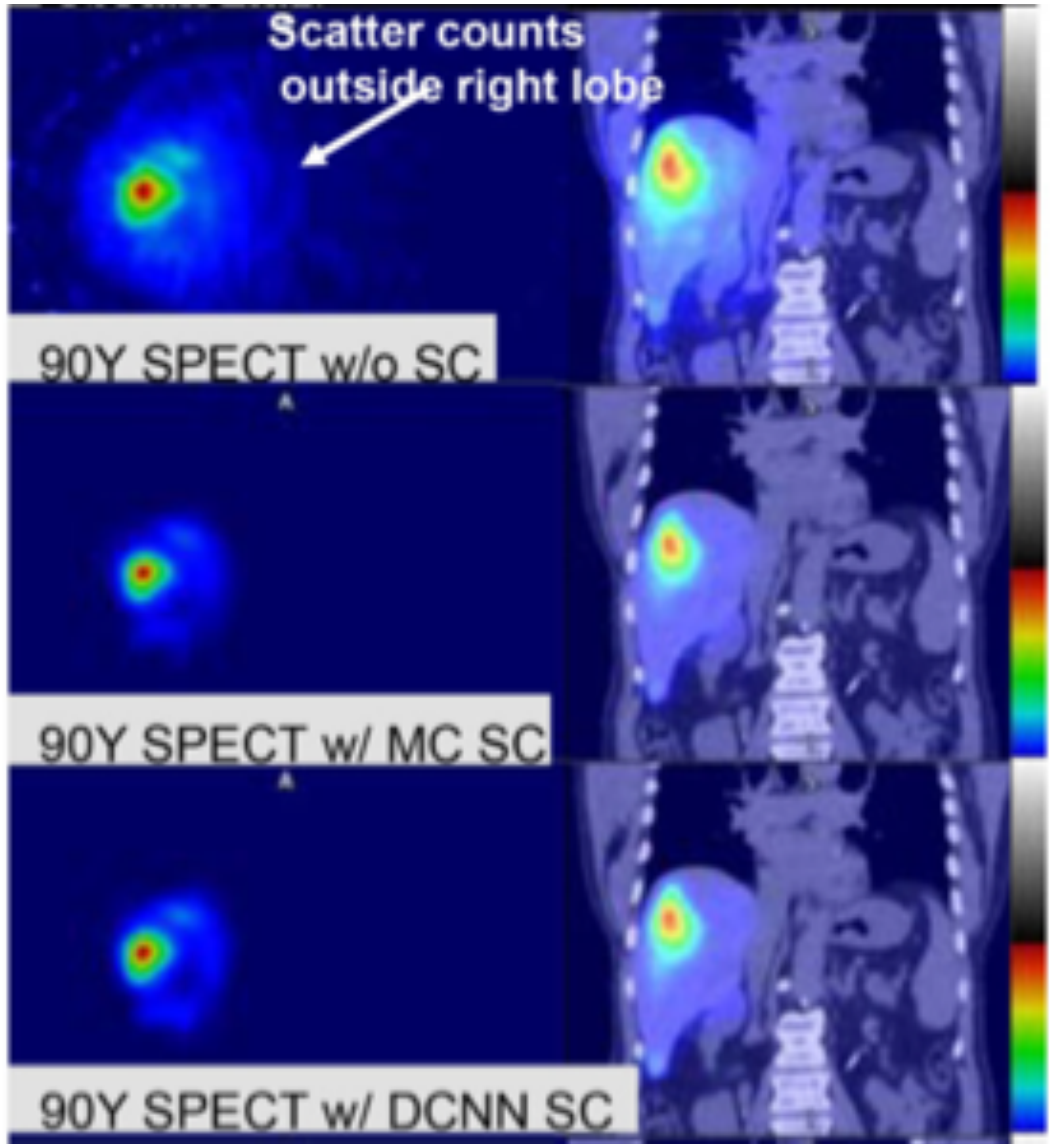


Scatter estimation/removal

Examples:

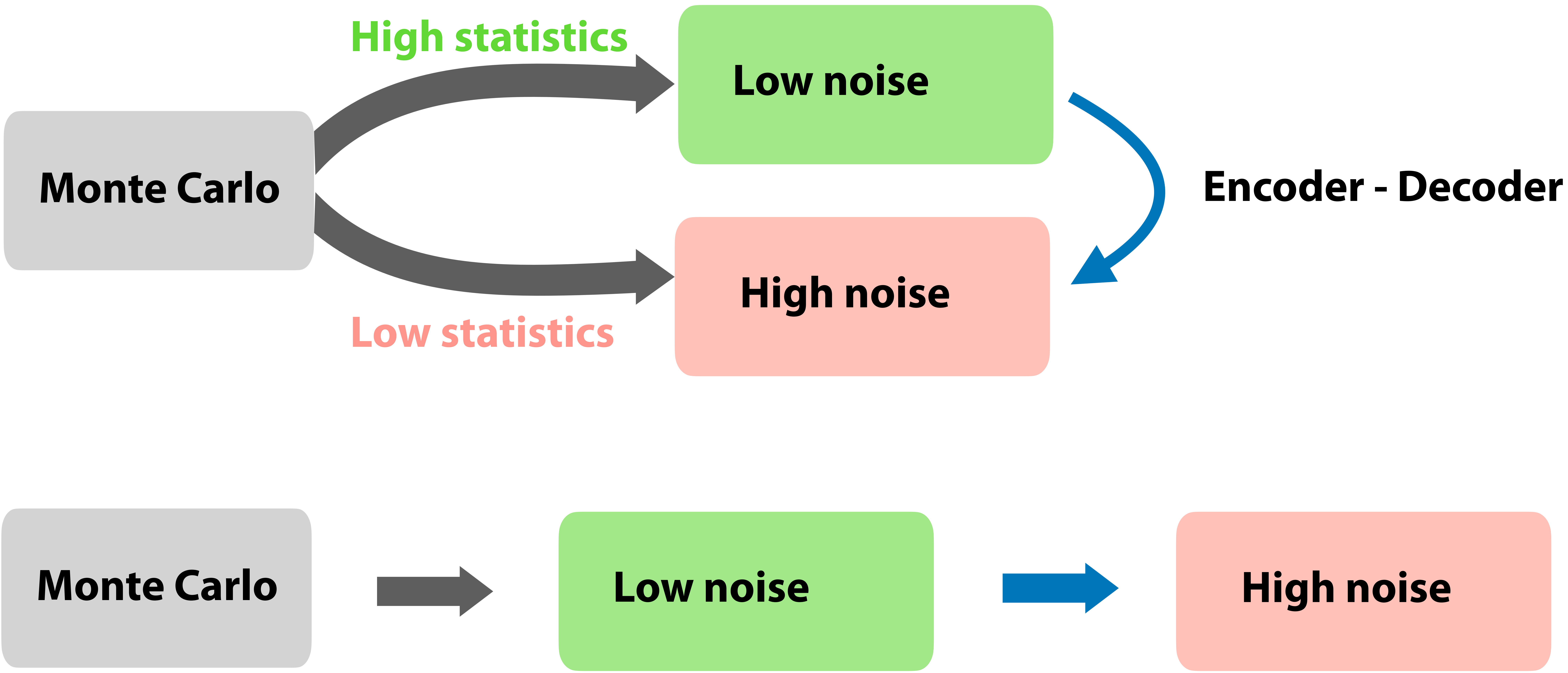
- Xiang et al, 2020, "A deep neural network for fast and accurate scatter estimation in quantitative SPECT/CT under challenging scatter conditions"
- Heyden et al., 2020, "Monte Carlo Based Scatter Removal Method for Non-isocentric Cone-Beam CT Acquisitions Using a Deep Convolutional Autoencoder"

[Lalonde2020]
[Lee2019]
[Maier2019]
[van der Heyden2020]
[Xiang2020]



[Xiang2020]

Enhance low-statistics Monte Carlo output



Enhance low-statistics Monte Carlo output

- Post-processing, CNN-based denoising
- Training dataset: pairs of high-noise/low-noise distributions
- Data: dose maps, SPECT/PET images
- 10–100 times fewer particles
- Dose gradients preserving?

[Fornander2019]
[Neph2019]
[Peng2019]
[Javaid2019]
[Reymann2019]
[Kontaxis2020]
[Javaid2021]
[Bai2021]

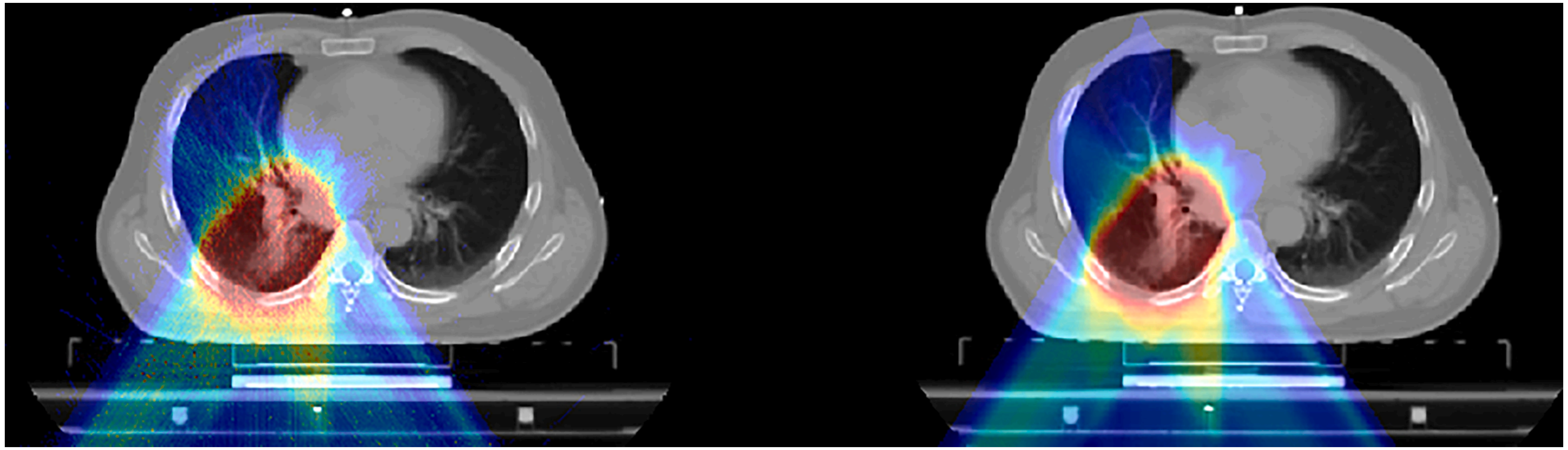


Fig. 3. A dose distribution on a sample lung test case. *Left* is the noisy dose distribution (MC simulation with 1×10^6 protons) and *right* is the reference dose distribution computed with 1×10^9 protons.

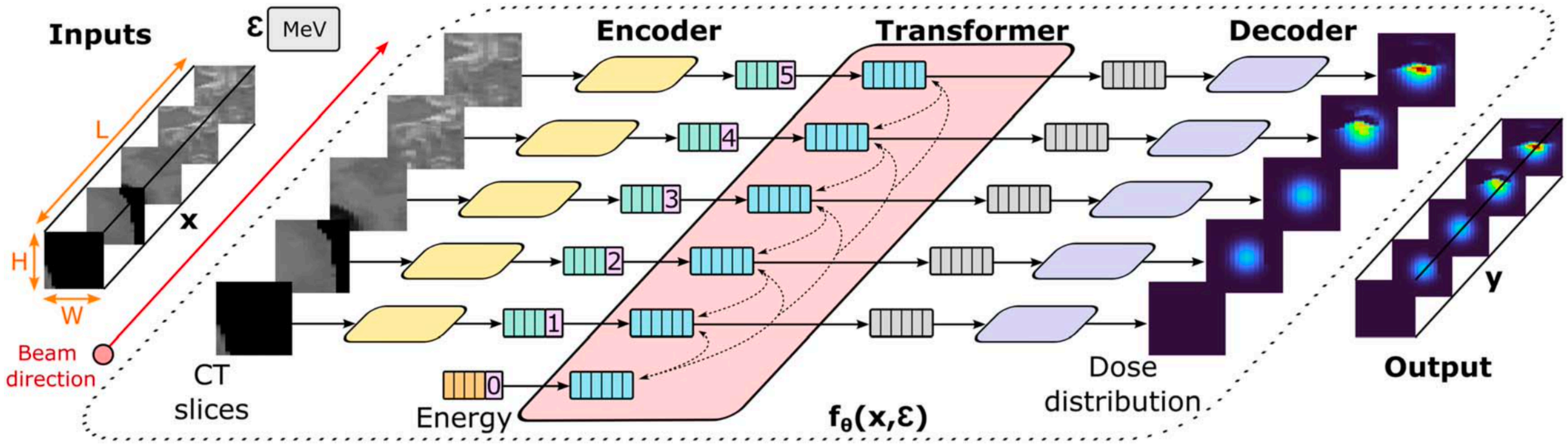
Javaid, U. et al., 2021, https://doi.org/10.1002/mp.13856

Deep learning based dose estimation

Example: Proton beam therapy

- Predict stack of 2D dose distribution along the beam

Training data: Monte Carlo

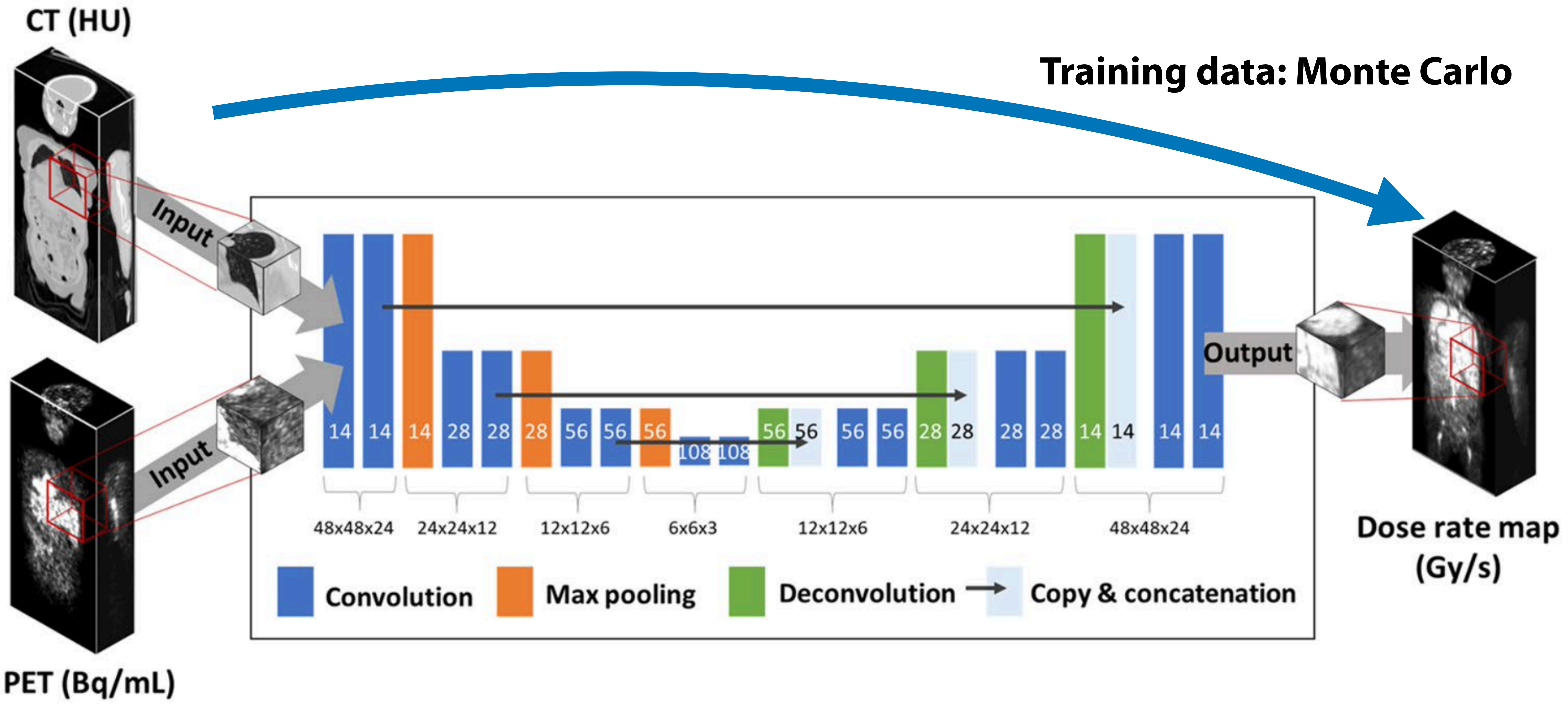


Pastor-Serrano, O., & Perkó, Z. (2022). https://doi.org/10.1088/1361-6560/ac692e

Deep learning based dose estimation

Example: Nuclear medicine

- Predict 3D dose distribution given CT and PET images



Lee, M. S. et al., (2019). https://doi.org/10.1038/s41598-019-46620-y

Models to synthetically generate (pseudo) images

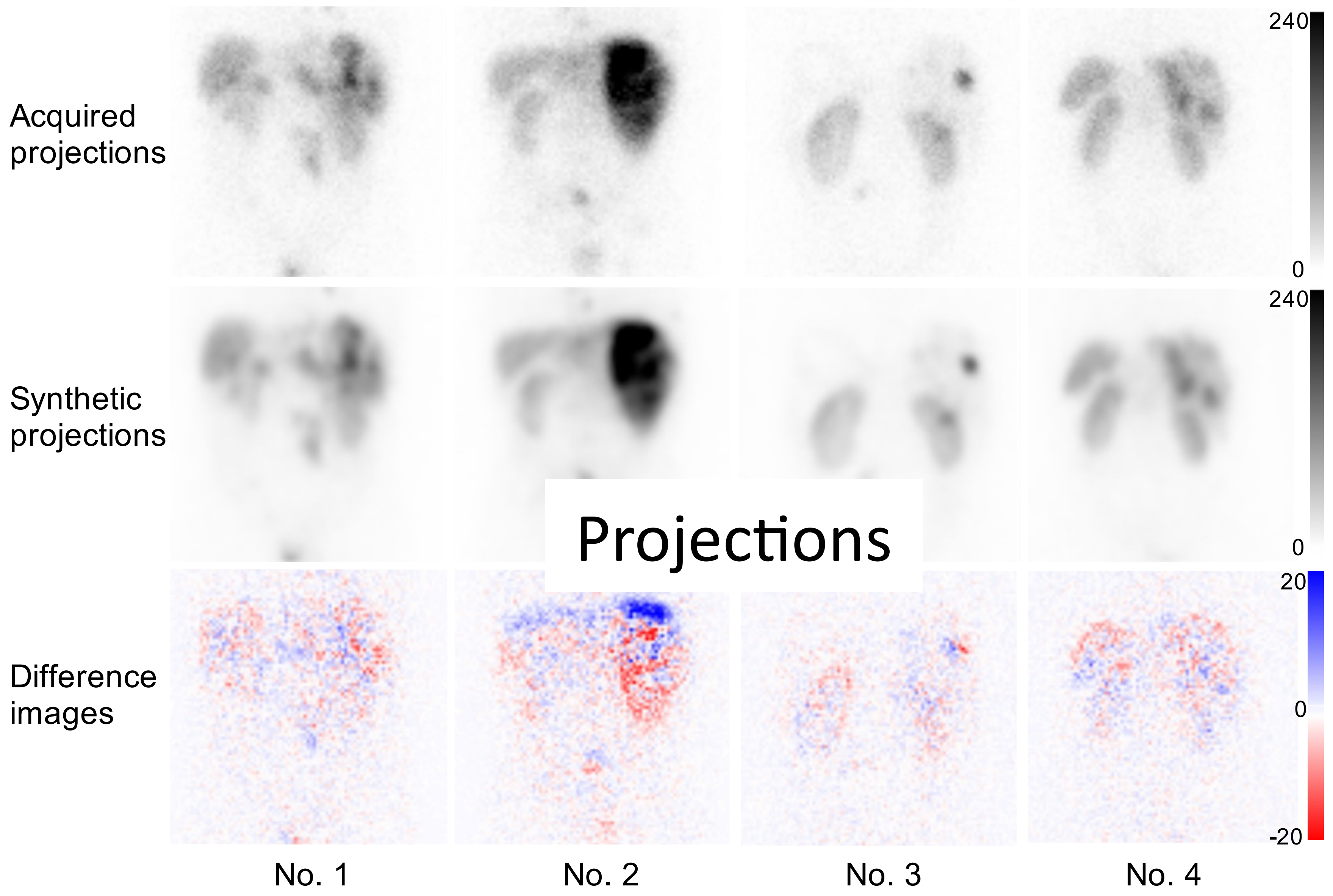
(not strictly Monte Carlo, but medical imaging)

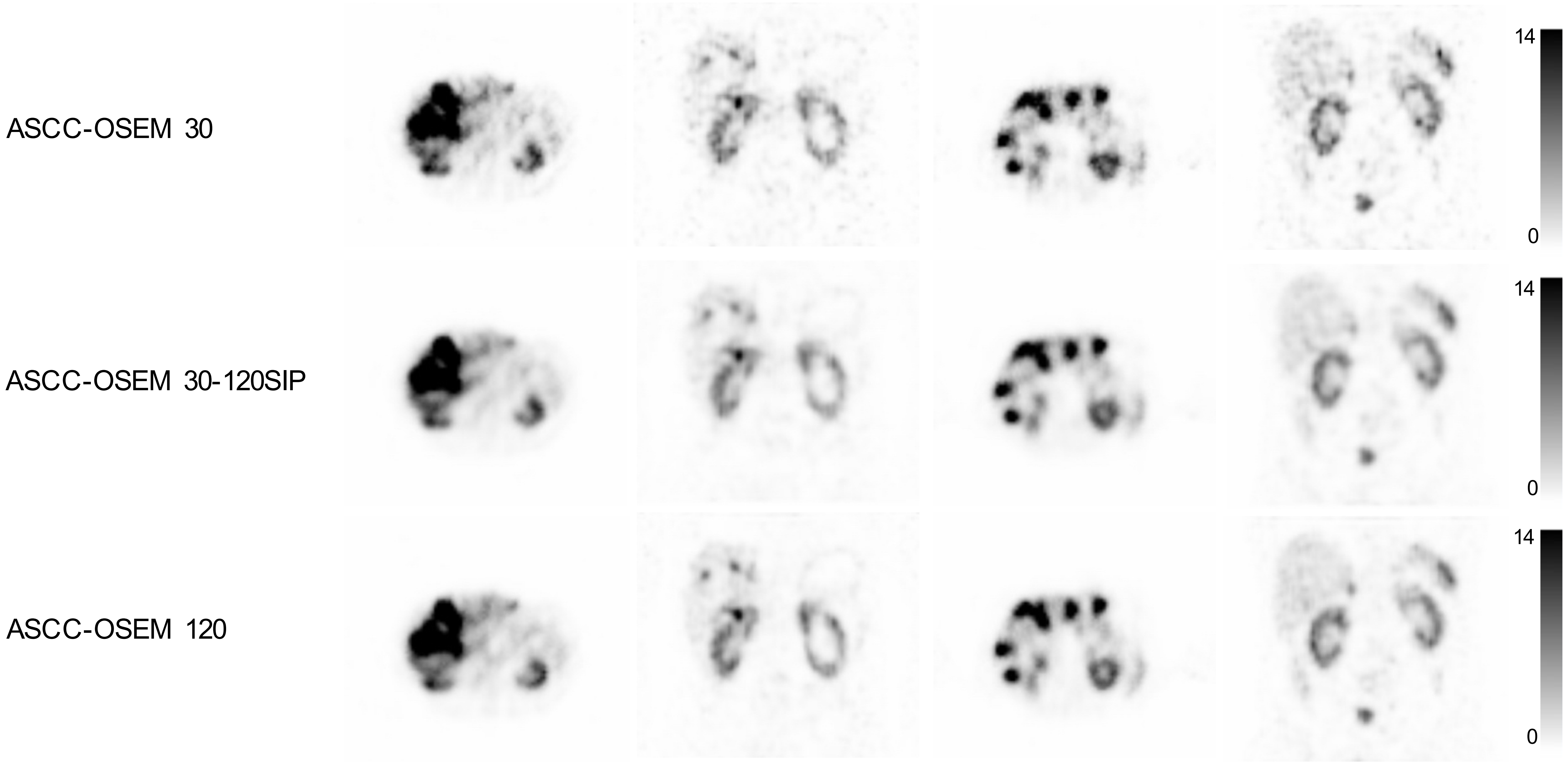
Generative intermediate SPECT projections

Long 3D SPECT acquisition with rotation gantry, around 120 angles

Rydén et al, 2020: Synthetic intermediate projections

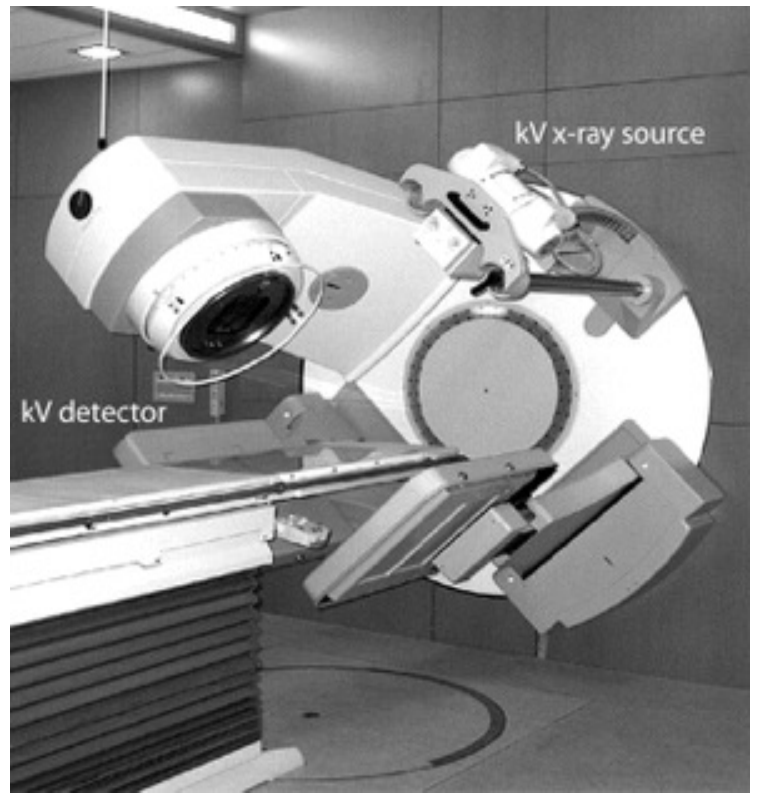
… to be used during the reconstruction



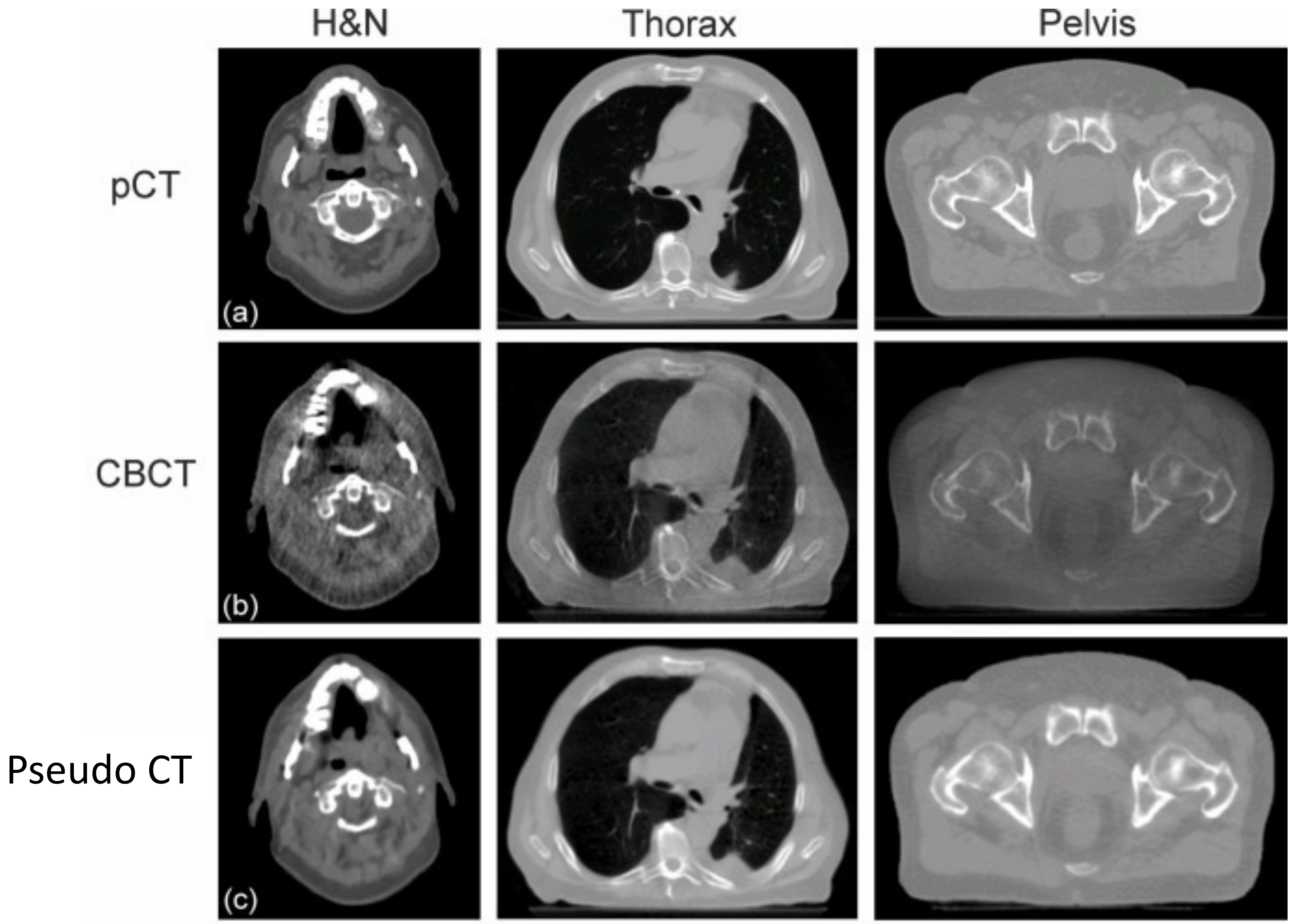


GAN for pseudoCT from cone beam CT

conventional (fan beam) x-ray CT



cone beam x-ray CT



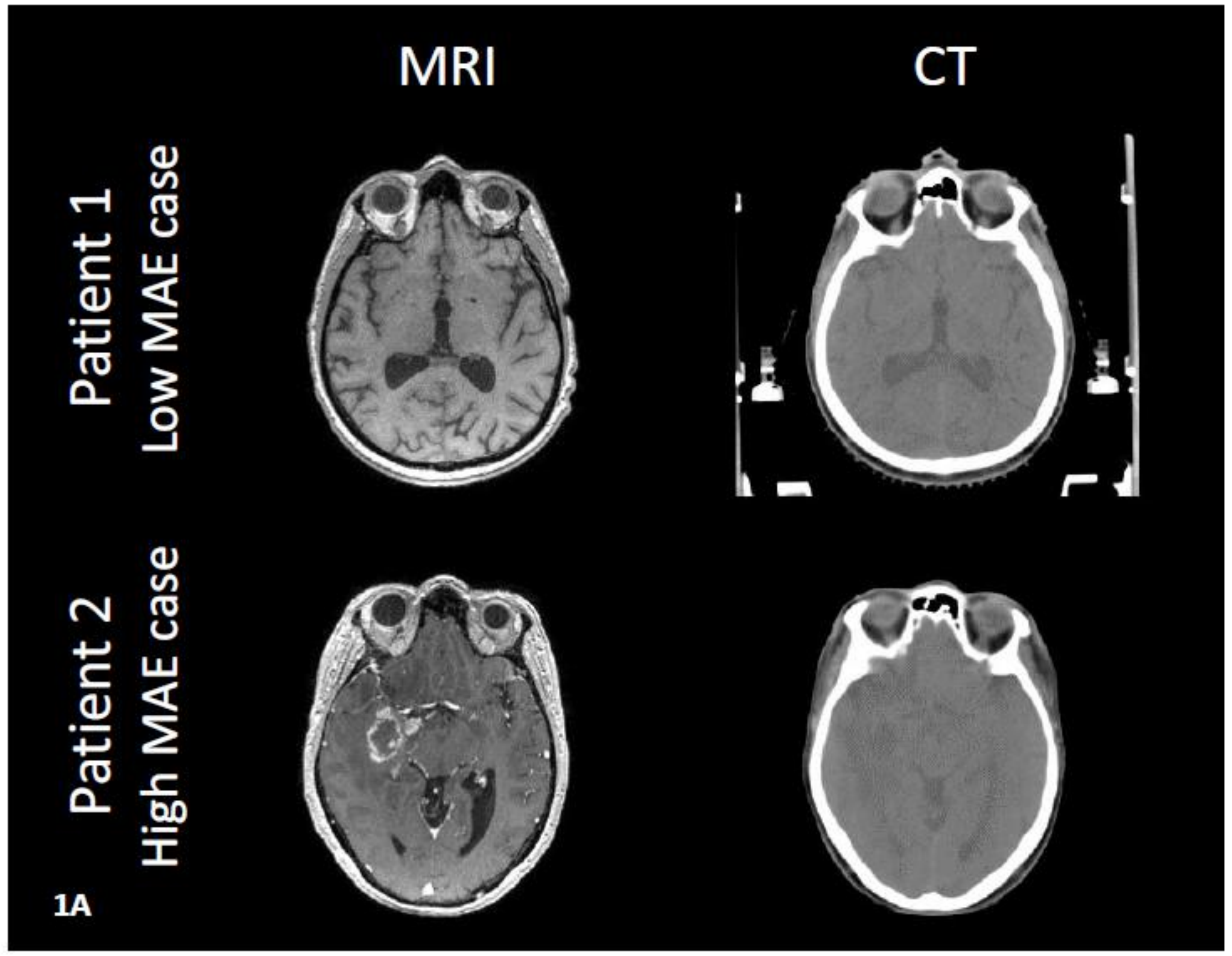
[Liang2019], [Kinda2020]: CycleGAN

GAN for pseudoCT from Magnetic Resonance Imaging (MRI)

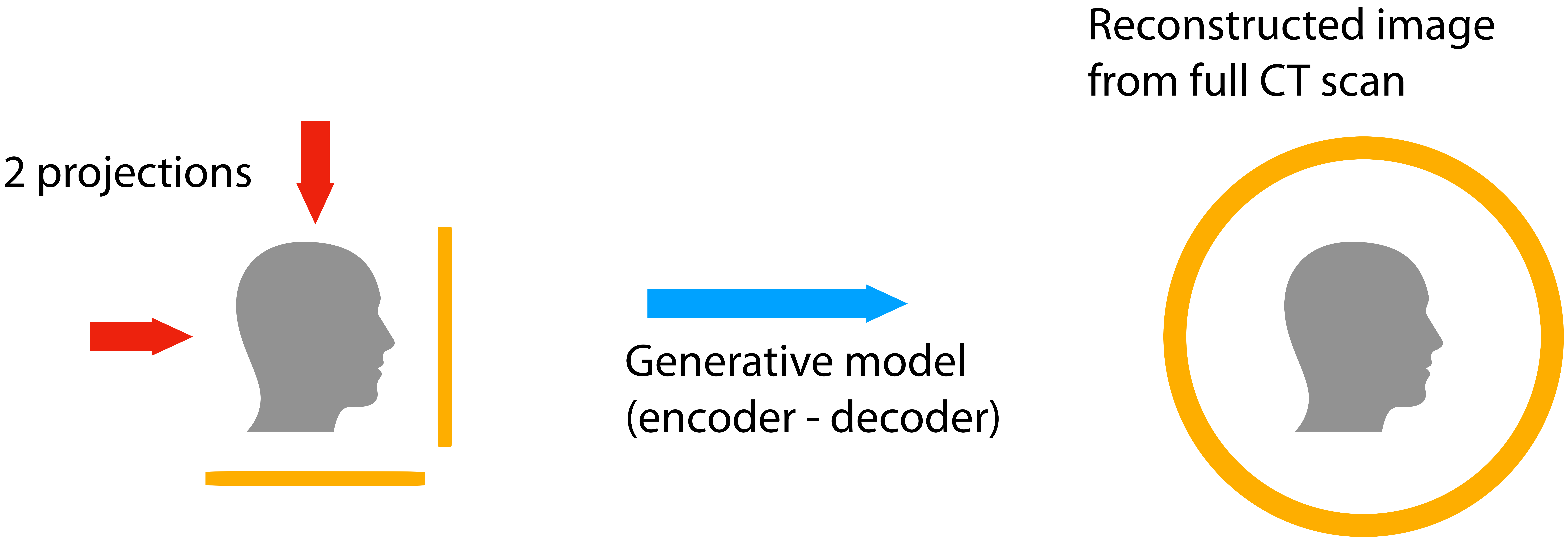
- Use case: input for dose computation in radiotherapy

x-ray CT

MRI

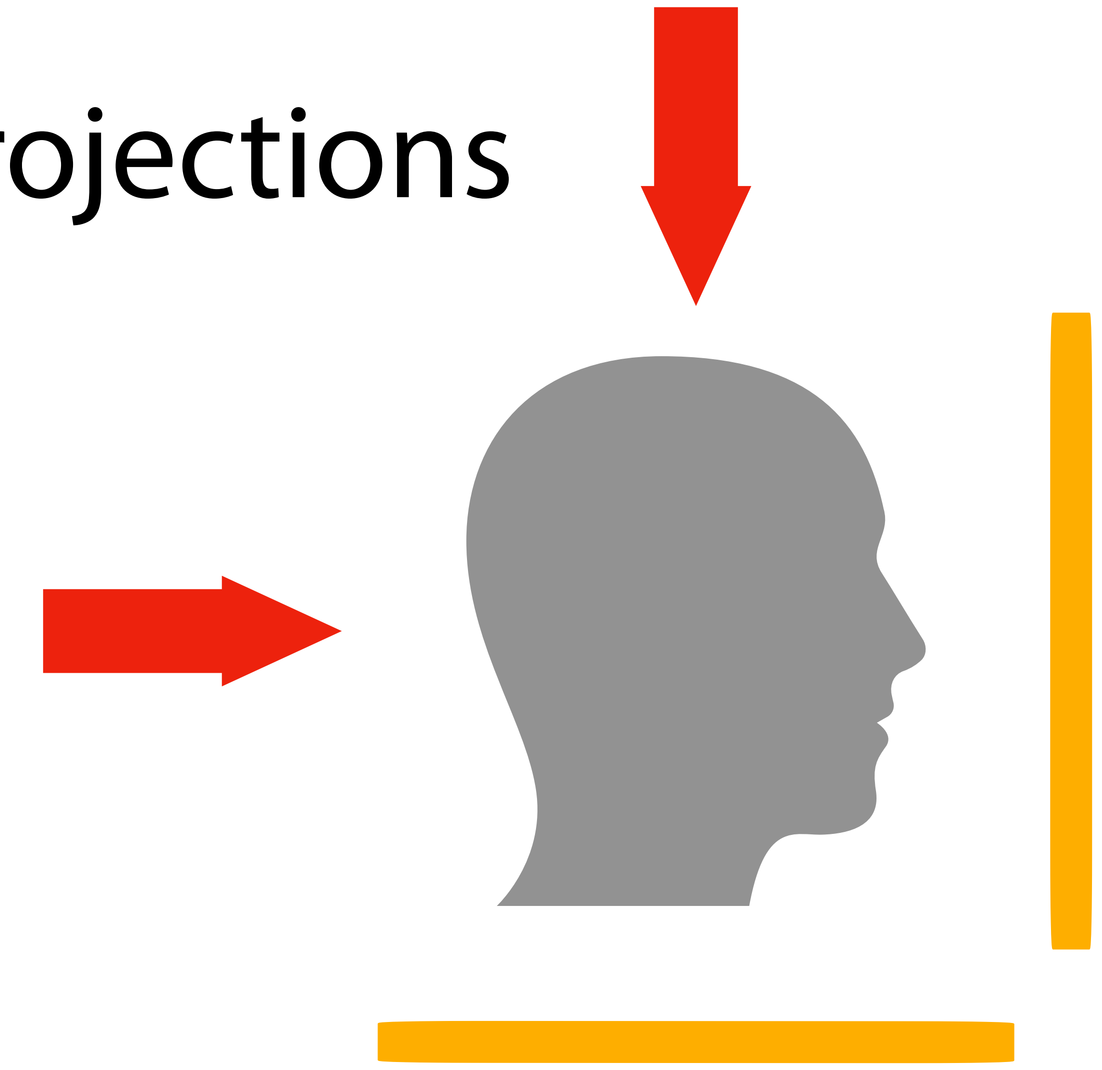


Heads up when generating images …



Now we can get patient CTs from only 2 projections! Awesome.

2 projections



Generative model
(encoder - decoder)

Reconstructed image
from full CT scan

Now we can get patient CTs from only 2 projections! Awesome.

Would you buy this?

Question: Up to which point is it legitimate to generate images?

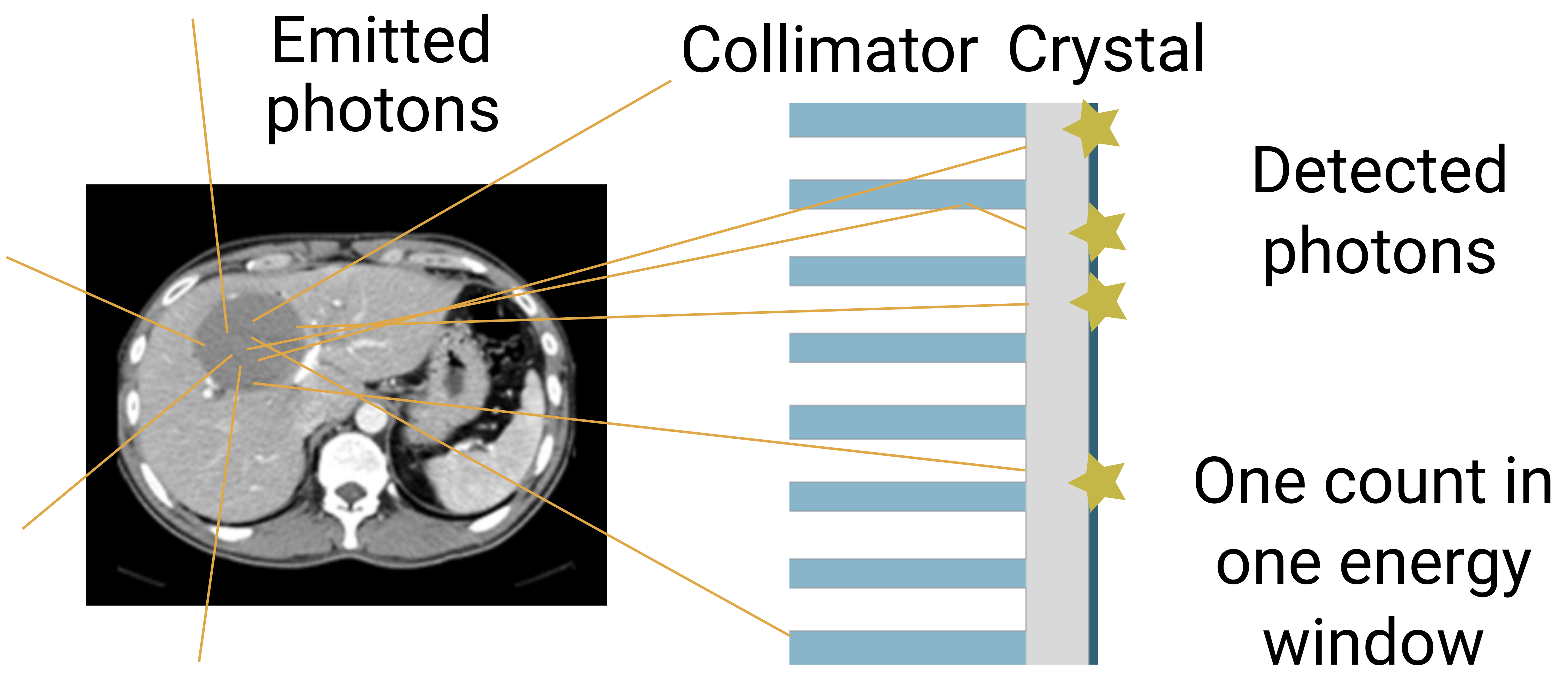
Question: Up to which point is it legitimate to generate images?

Tentative answer: Depends if we transform/convert information or invent it.

Learning the detector response function in Monte Carlo simulation of SPECT imaging

Angular Response Function (ARF)

- SPECT imager usually has a collimator with holes to filter out photons according to their direction.
- Energy windows are applied at the electronic acquisition level.



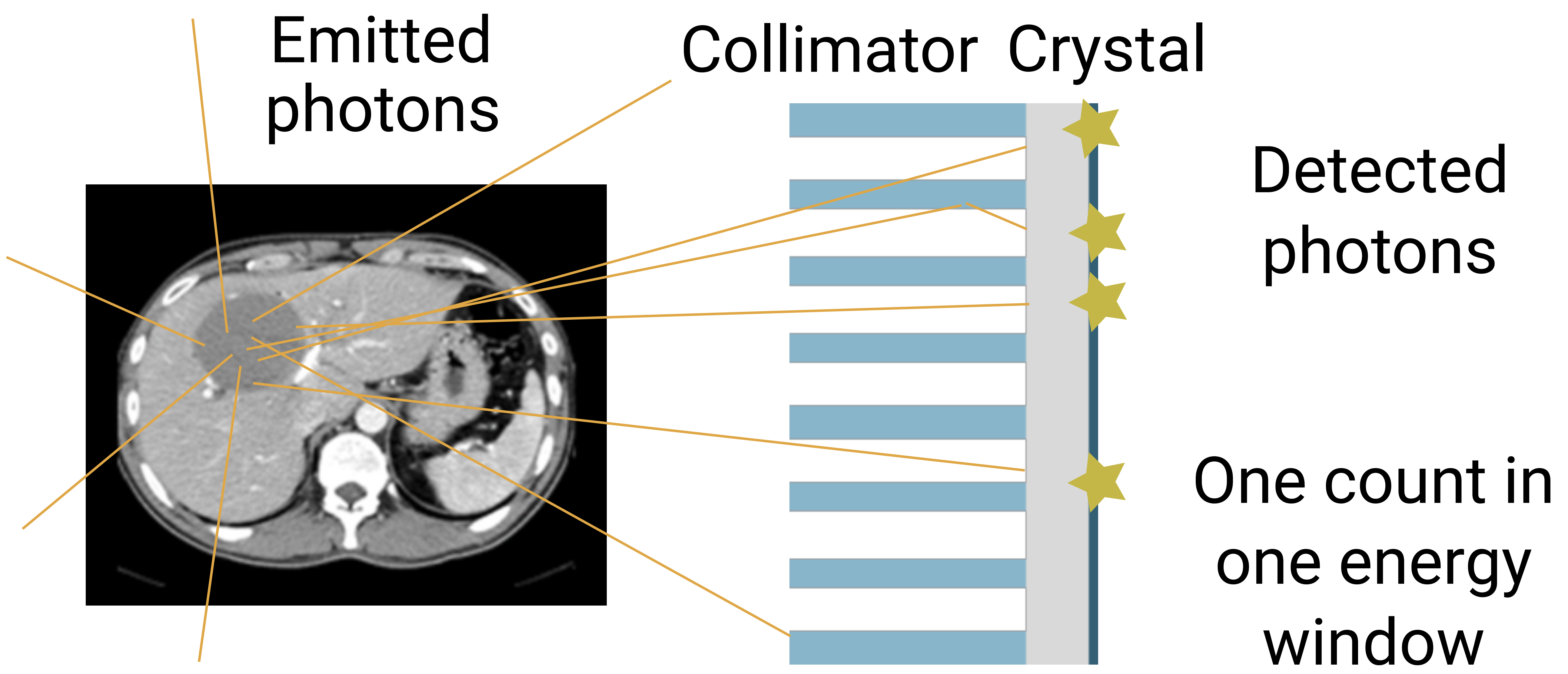
Detection likelihood

$$p(\theta, \phi, E)$$

Likelihood in window "i"

$$\int_{E_{i,\text{low}}}^{E_{i,\text{high}}} p(\theta, \phi, E)\mathrm{d}E$$

Angular Response Function (ARF)



Detection likelihood

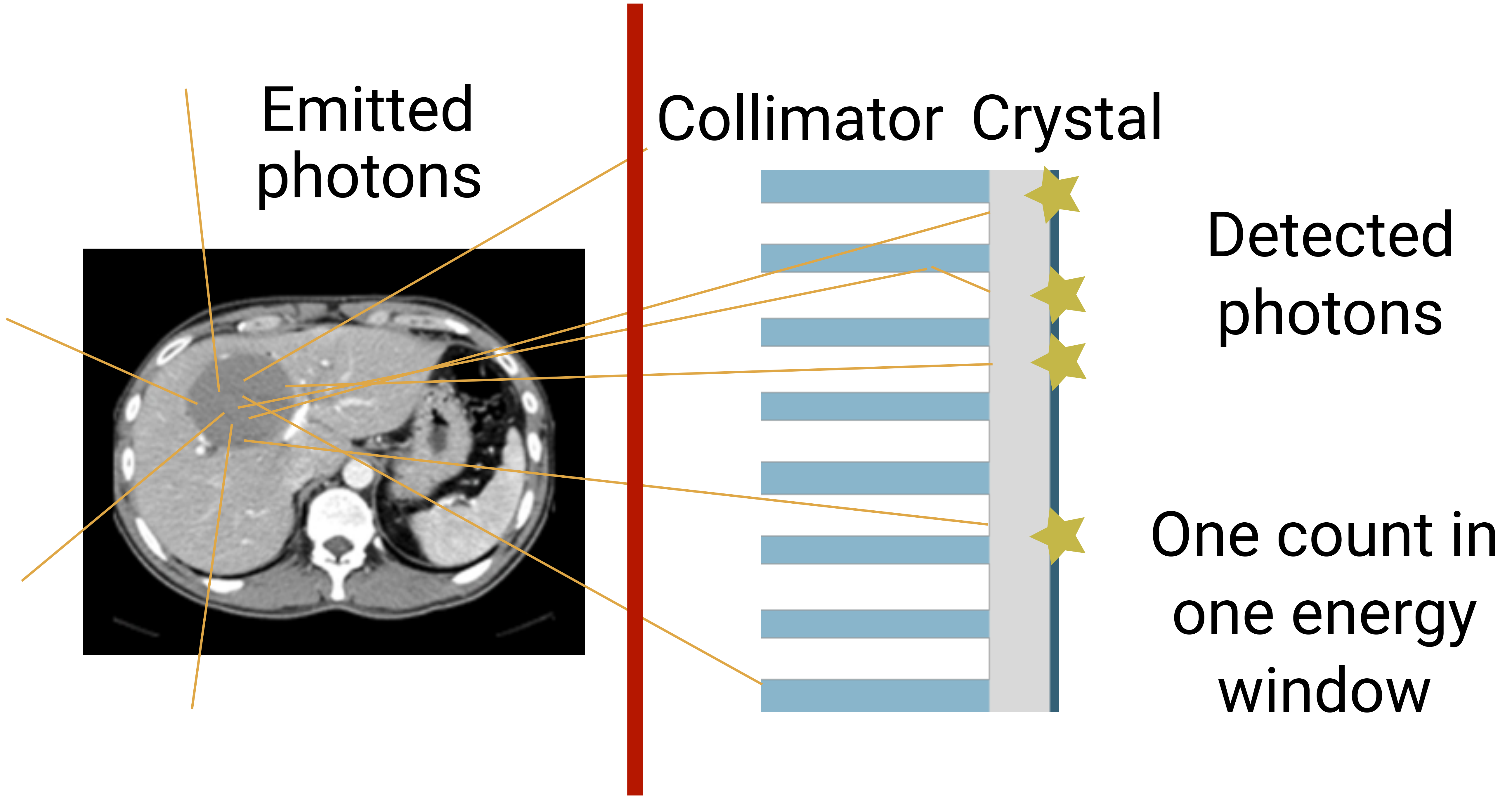
$$p(\theta, \phi, E)$$

Likelihood in window "i"

$$\int_{E_{i,\mathrm{low}}}^{E_{i,\mathrm{high}}} p(\theta, \phi, E)\mathrm{d}E$$

Angular Response Function (ARF)

- Instead of transporting the photons across the detector, we score them according to the likelihood determined from their properties at a virtual plain.



Detection likelihood

$$p(\theta, \phi, E)$$

Likelihood in window “i”

$$\int_{E_{i,\mathrm{low}}}^{E_{i,\mathrm{high}}} p(\theta, \phi, E)\mathrm{d}E$$

Angular Response Function (ARF)

Detection likelihood

$$p(\theta, \phi, E)$$

Likelihood in window "i"

$$p_i(\theta, \phi) = \int_{E_{i,\mathrm{low}}}^{E_{i,\mathrm{high}}} p(\theta, \phi, E)\mathrm{d}E$$

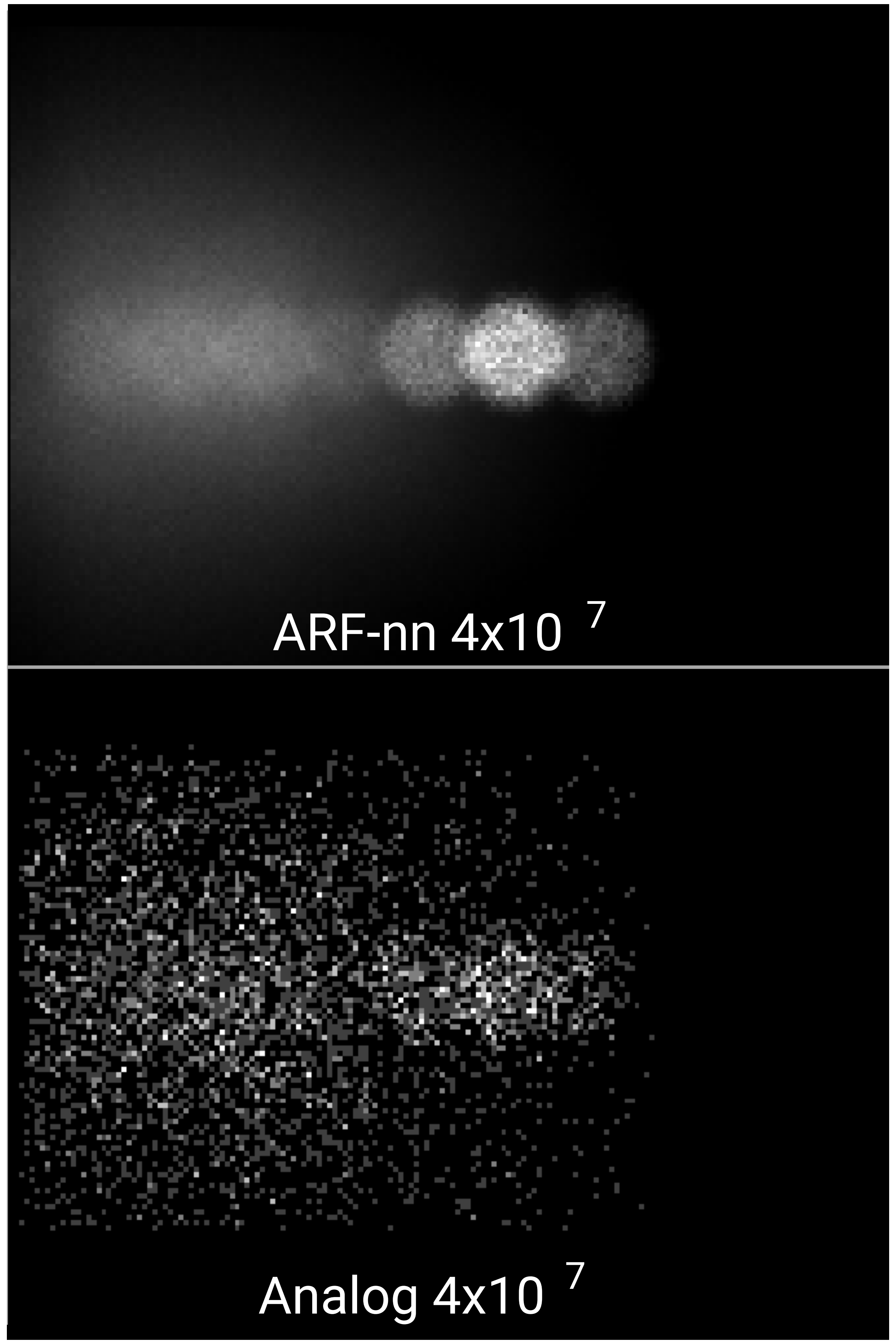
Parametrize as multi-dimensional histogram

Train a neural network to predict p_i from theta and phi

acts as variance reduction technique because network is a smooth non-linear function

Angular Response Function (ARF)

Image of spherical sources



Neural network based ARF

ARF acts as variance reduction technique

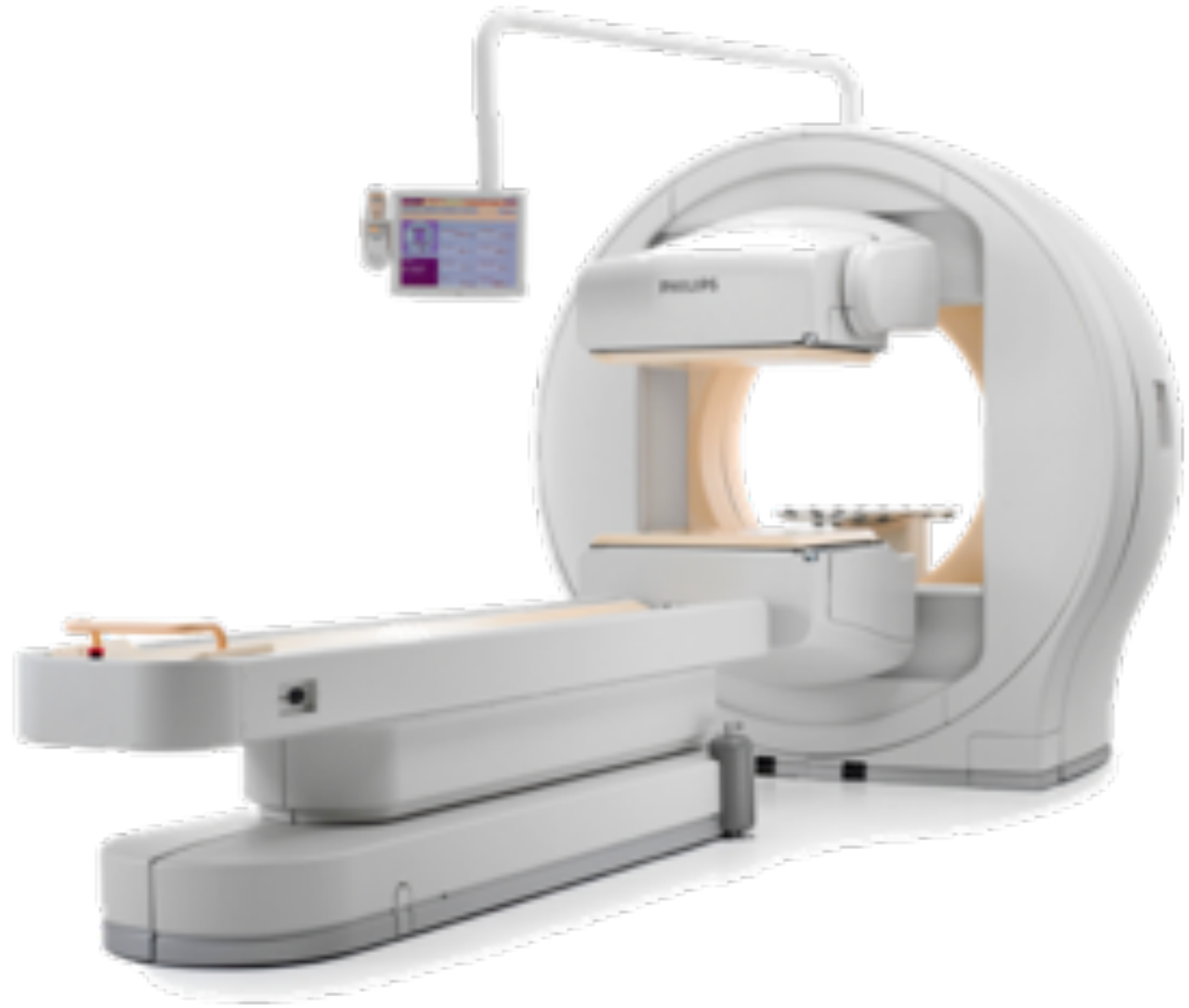
Explicit (analog) Monte Carlo simulation

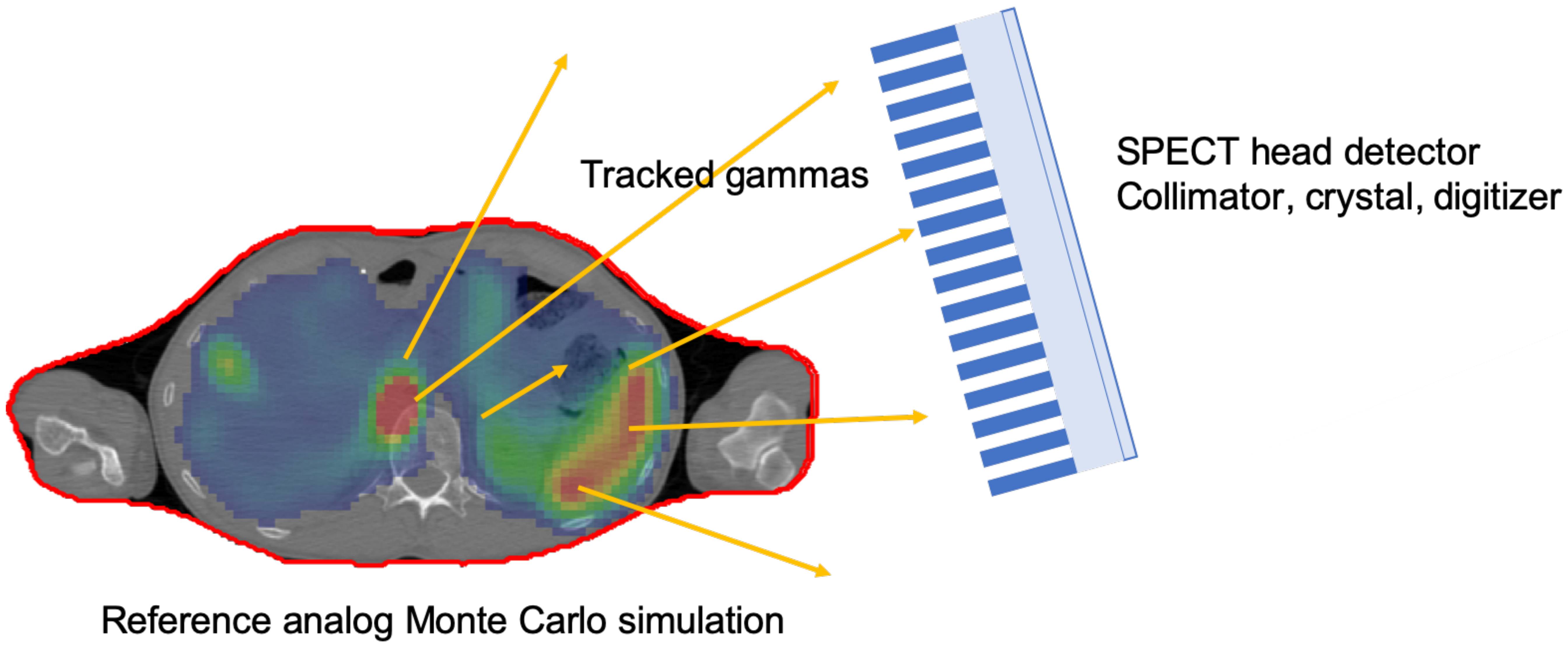
Sarrut, D., Krah, N., et al., 2018, https://doi.org/10.1088/1361-6560/aae331

Learning phase-space distributions for Monte Carlo simulations in nuclear imaging

GAN to accelerate SPECT simulation

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

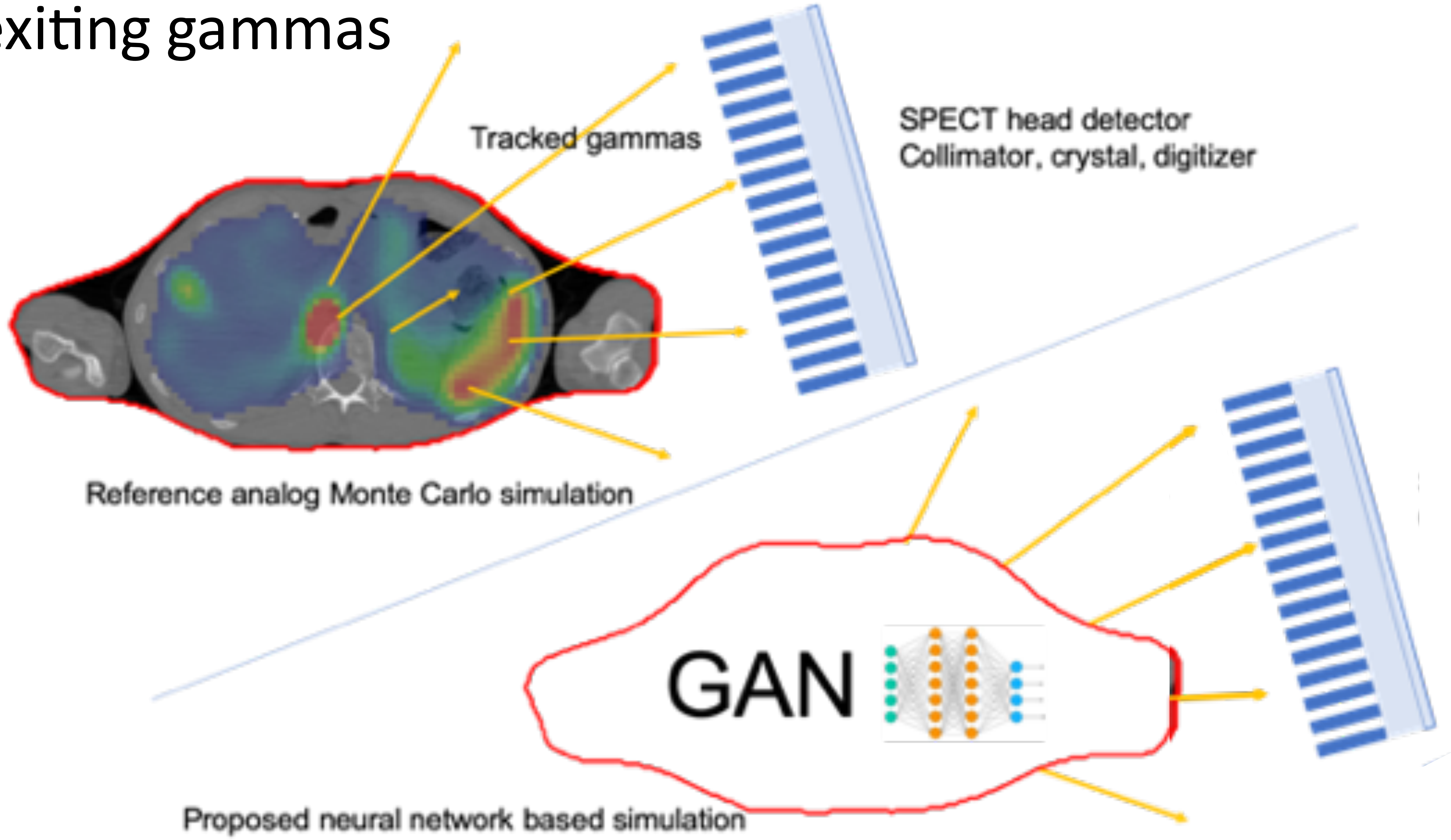




Training the GAN based on Monte Carlo data

Train a GAN to produce exiting gamma from a given source

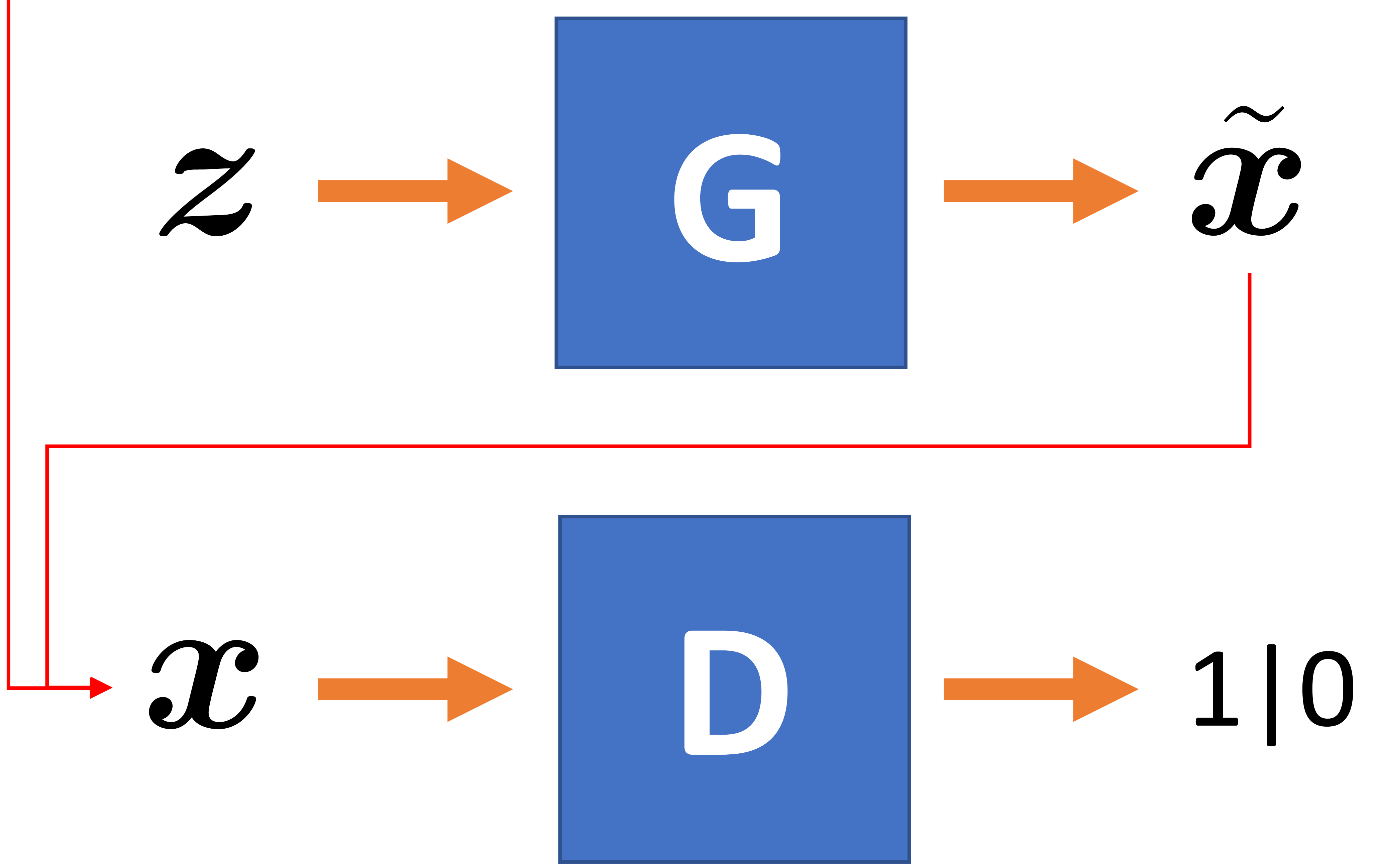
- Step1: run low stats MC, consider exiting gammas
- Step2: train a GAN
- Step3: use GAN a source



Training the GAN based on Monte Carlo data

[Goodfellow et al 2014]

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



Generalize GAN to arbitrary source distribution: Conditional GAN

Train with one given **phantom** (CT, patient) …

… but with **homogeneous** activity

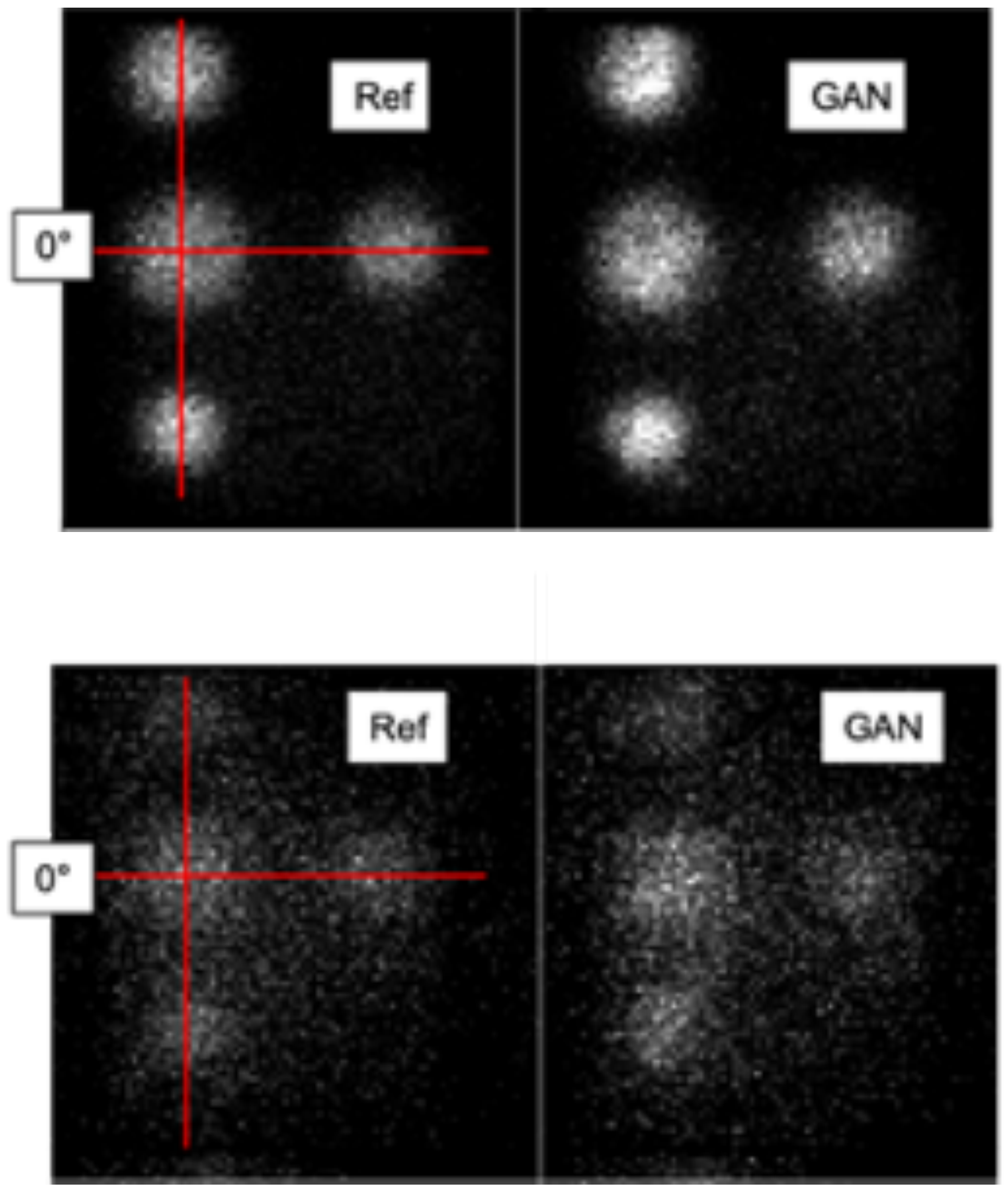
Conditional input activity map.

$$\mathcal{L}_C = \mathbb{E}[C(G(\mathbf{z}|\mathbf{y}))] - \mathbb{E}[C(\mathbf{x}|\mathbf{y})] + \underbrace{\lambda\mathbb{E}[(\left\|\nabla_{\hat{\mathbf{x}}|\mathbf{y}}C(\hat{\mathbf{x}}|\mathbf{y})\right\|_2 - 1)^2]}_{GP \quad regularization}$$

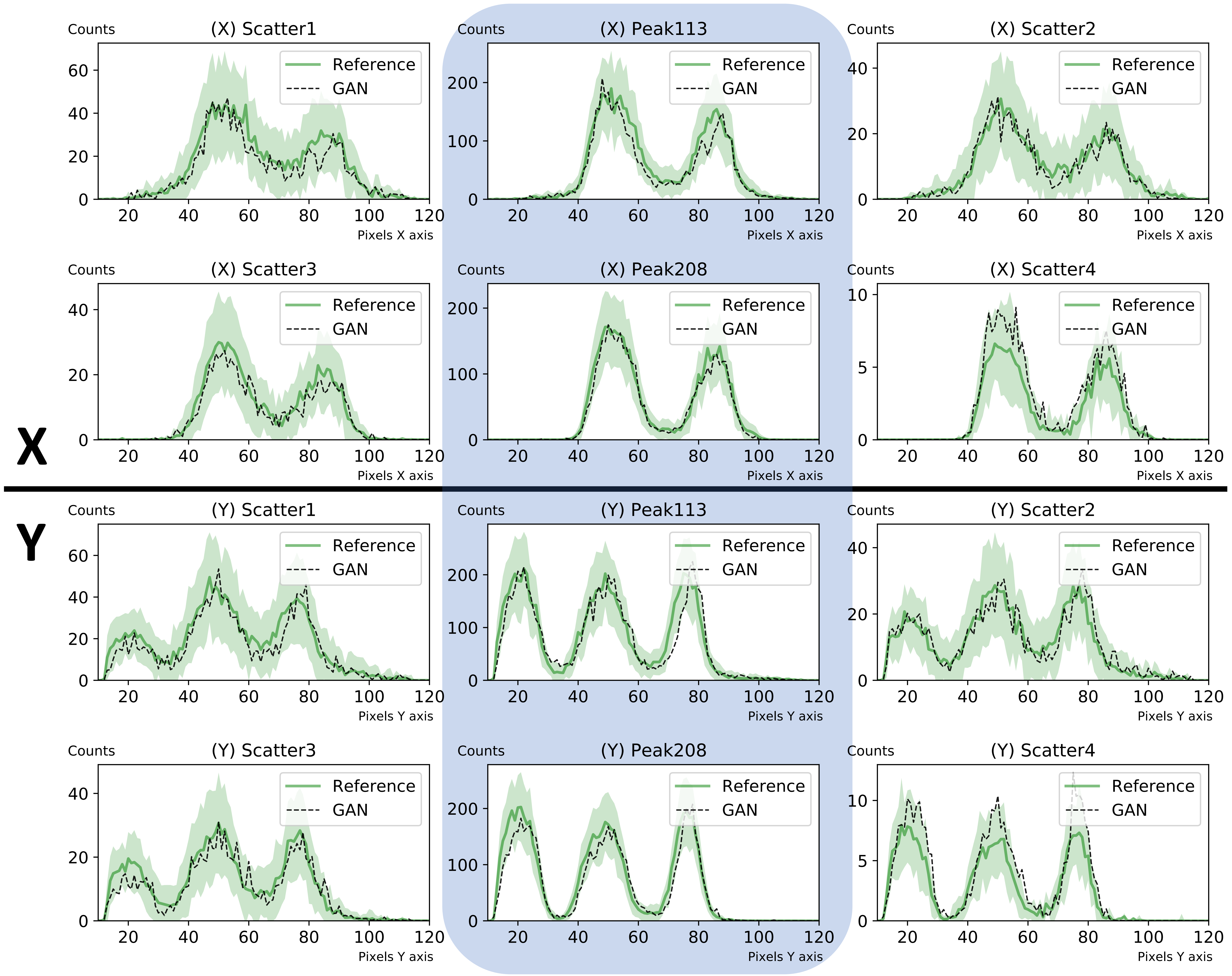
$$\mathcal{L}_G = -\mathbb{E}[C(G(\mathbf{z}|\mathbf{y}))]$$

[Saporta et al, PMB 2022]

Results

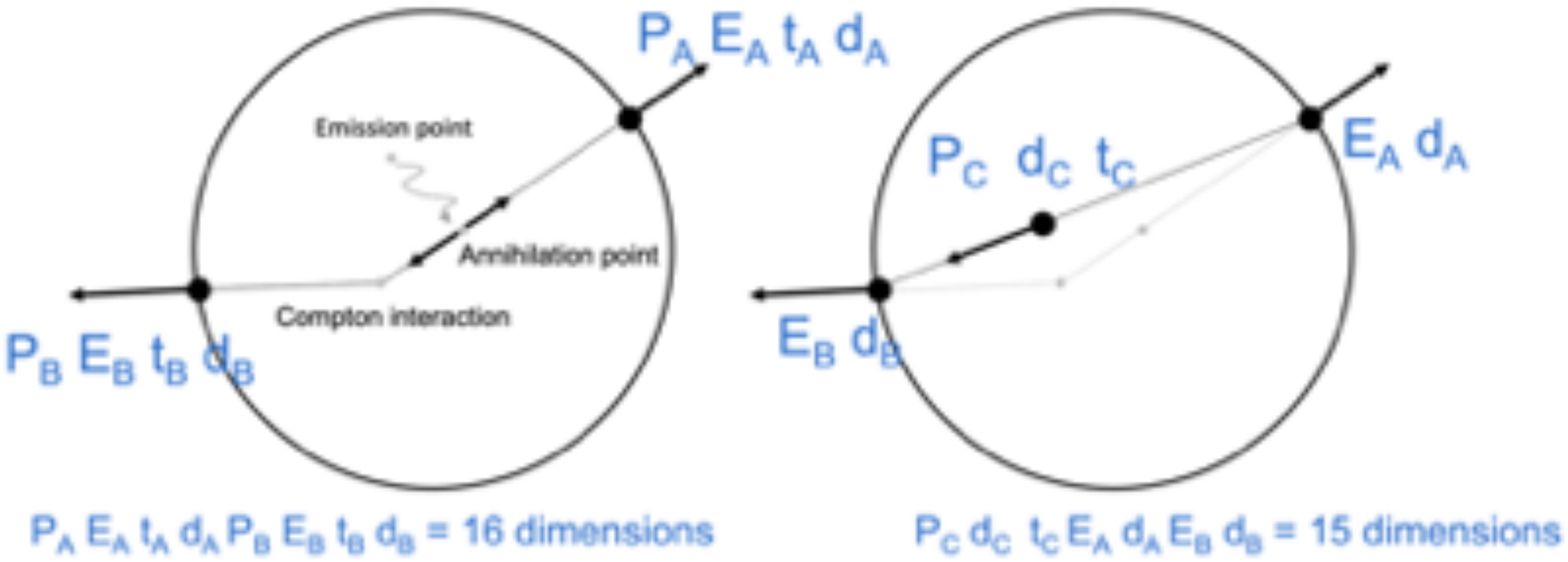


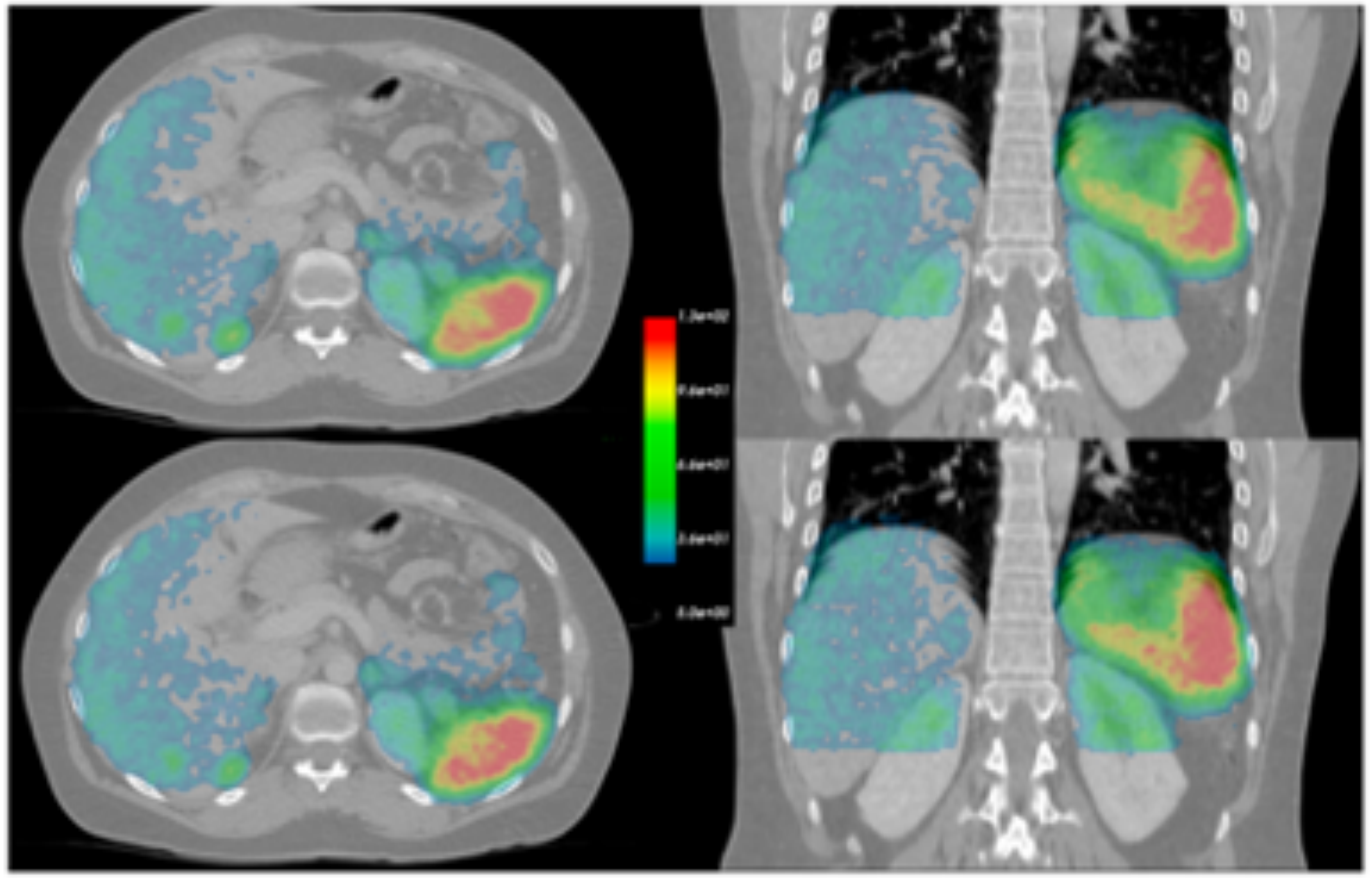
2D projections
Lu177 (1 peak & scatter)



GAN in PET simulation

Back-to-back GAN photon source for PET Monte Carlo simulation





[Sarrut et al, PMB 2023]

Recall:

Simulated images are spatially discretized probability distributions
where
1 image represents 1 distribution

Recall:

Simulated images are spatially discretized probability distributions
where
1 image represents 1 distribution

G(z) $\rightarrow$ Image

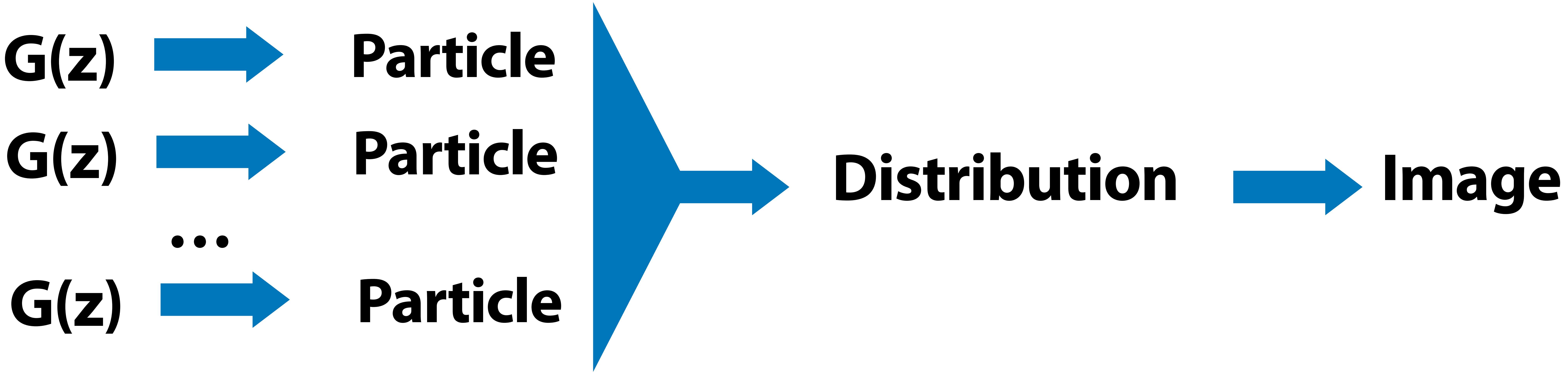
Recall:

Simulated images are spatially discretized probability distributions
where
1 image represents 1 distribution

G(z) → Image

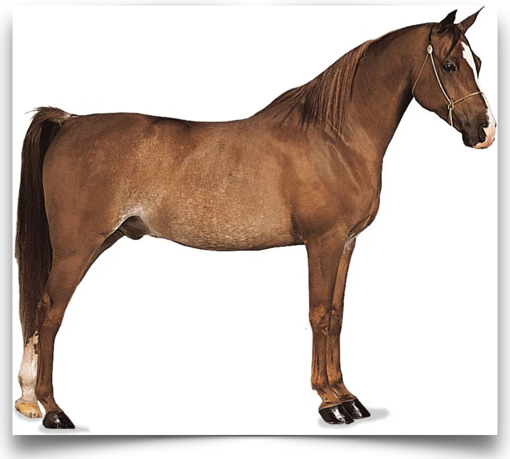
Generators produce 1 image from 1 vector z

Different when generating particles in Monte Carlo:



Inaccurate sampling of distribution of G(z) has systematic impact on image (artifacts)

In other words …

G(z)


G(z)


G(z)


G(z)


Each horse looks realistic

“G works well”

although G might slightly
overrepresent brown horses ….

… now with particles

G(z) Particle

G(z) Particle

G(z) Particle

G(z) Particle

Each particle has realistic energy

“G works well” (?)

But high energy particles are oversampled

Incorrect image

Curious to try Monte Carlo simulations yourself?

GATE 10 (beta)

- Non-beta release this summer
- Static input files replaced by Python scripts

```
pip install opengate
```

- Multi-thread, PyTorch
- Faster, embeddable, powerful
- SPECT, PET, Linac, proton, dose …
- More than 150 tests and examples

Open and Open-source

https://github.com/OpenGate/opengate

Thank you very much

Thank you very much

CREATIS

Lyon, France