



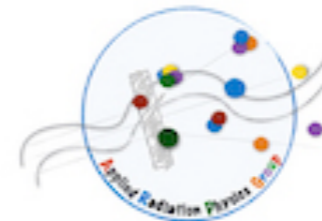
# Machine learning for medical applications of Physics

Carlo Mancini Terracciano  
[carlo.mancini.terracciano@roma1.infn.it](mailto:carlo.mancini.terracciano@roma1.infn.it)

AI-at-SLAC seminar  
January 16th



SAPIENZA  
UNIVERSITÀ DI ROMA



INFN

# Outline

---

- Two applications of Machine Learning:
  - MRI image analysis and segmentation
  - reproducing the final state of a low energy nuclear interaction model (BLOB)

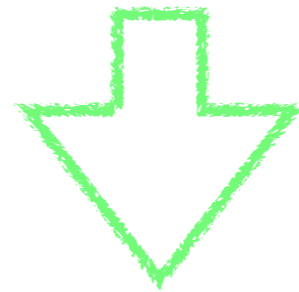


# MRI image analysis and segmentation

# Introduction

---

- The therapy for locally advanced rectal cancers is:
  - neoadjuvant chemo-radiotherapy (CRT)
  - followed by radical surgery



- local pelvic recurrence to rate lower than 10%

# However...

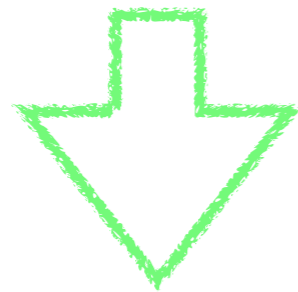
---

- This therapeutic approach is an **over-treatment** of many patients:
- those who do not respond to the treatment (non-responders)
  - whose early identification (2–3 weeks after the start of neoadjuvant CRT) might help clinicians in referring them to alternative treatments;
- patients with pathological complete response
  - who could benefit from either less invasive surgery (ie, transanal endoscopic microsurgery) or “wait-and-watch” strategy

# The purpose

---

- Recognise non-responders and complete responder patients during the CRT (before surgery)
- Finding new biomarkers



- Personalize the treatment plan

# The dataset available

---

- 55 patients with histologically confirmed rectal adenocarcinoma and locally advanced tumor stages II (cT3-4, N0, M0) and III (cT1-4, N+, M0)
- 3T MR images T2-weighted axial oblique (planes orthogonal to the rectum)
- Images acquired in three stages:
  - just after diagnosis (pre-CRT)
  - CRT response evaluation at early phase: CRT treatment was day 40 and tumour response was assessed with MRI at day 21 (mid-CRT)
  - pre-surgical analysis: 6 to 8 weeks after the end of CRT (post-CRT)
- gross specimen was analysed by pathologist

# The dataset available

---

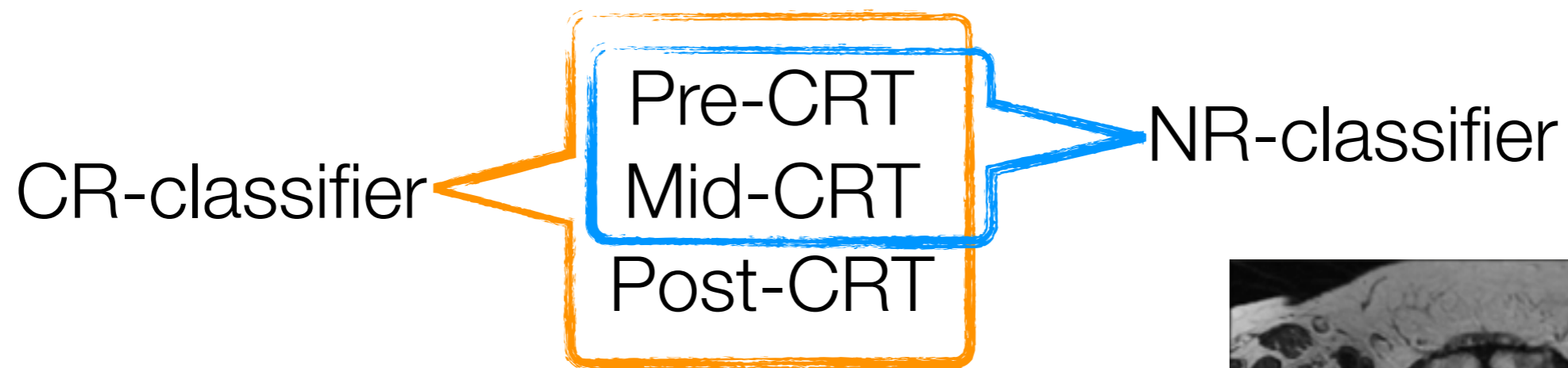
- gross specimen was analysed by pathologist
- Surgical specimen, analysed by pathologist, to assess the response to the therapy:
  - **16** patients were **Complete Responders (CR)**
  - **27** patients were **Partial Responders (PR)**
  - **12** patients were **Non Responders (NR)**



# The dataset available

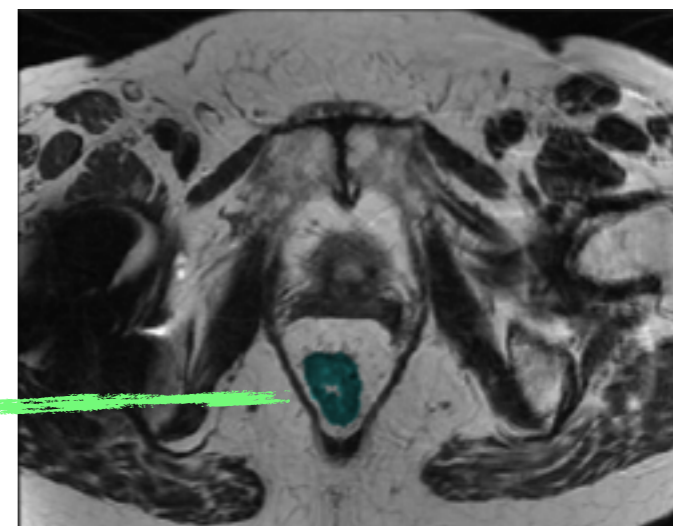
---

- T2-weighted MRI from high field (3T) scanner
- ~30 slices along the rectum axis



## **Region of interest (ROI)**

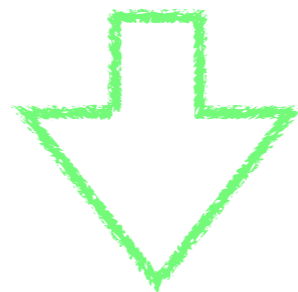
drawn by hand by the radiologist  
contains suspect tumor tissue  
Usually 7-10 slice per scan



# The goal

---

- Automatically stratify the response to CRT before surgery
- Identify Complete Responders after CRT to (possibly) avoid surgery (e.g. wait and watch strategy)
- Identify Non Responders during CRT to address them to a more effective strategy



- Two different classifiers

# Software tool

---

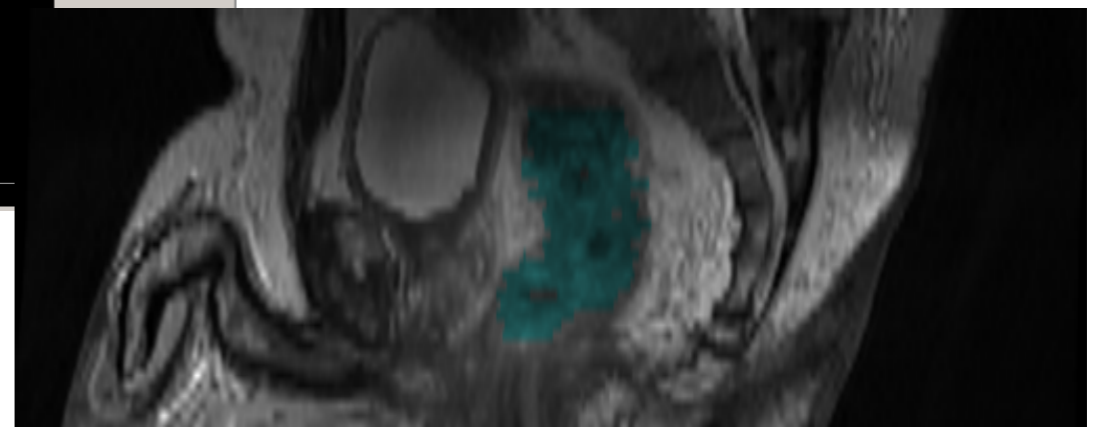
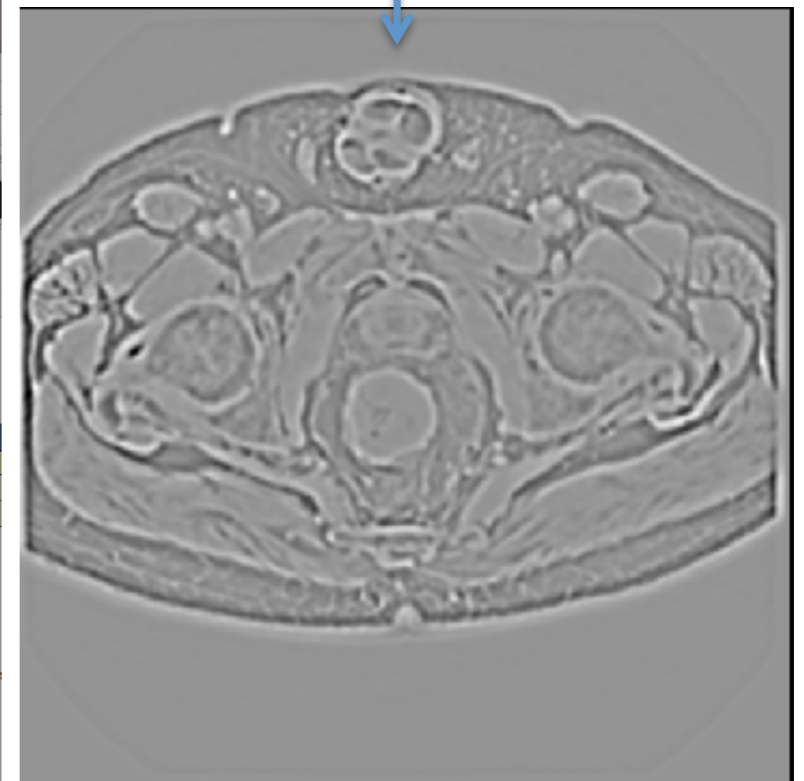
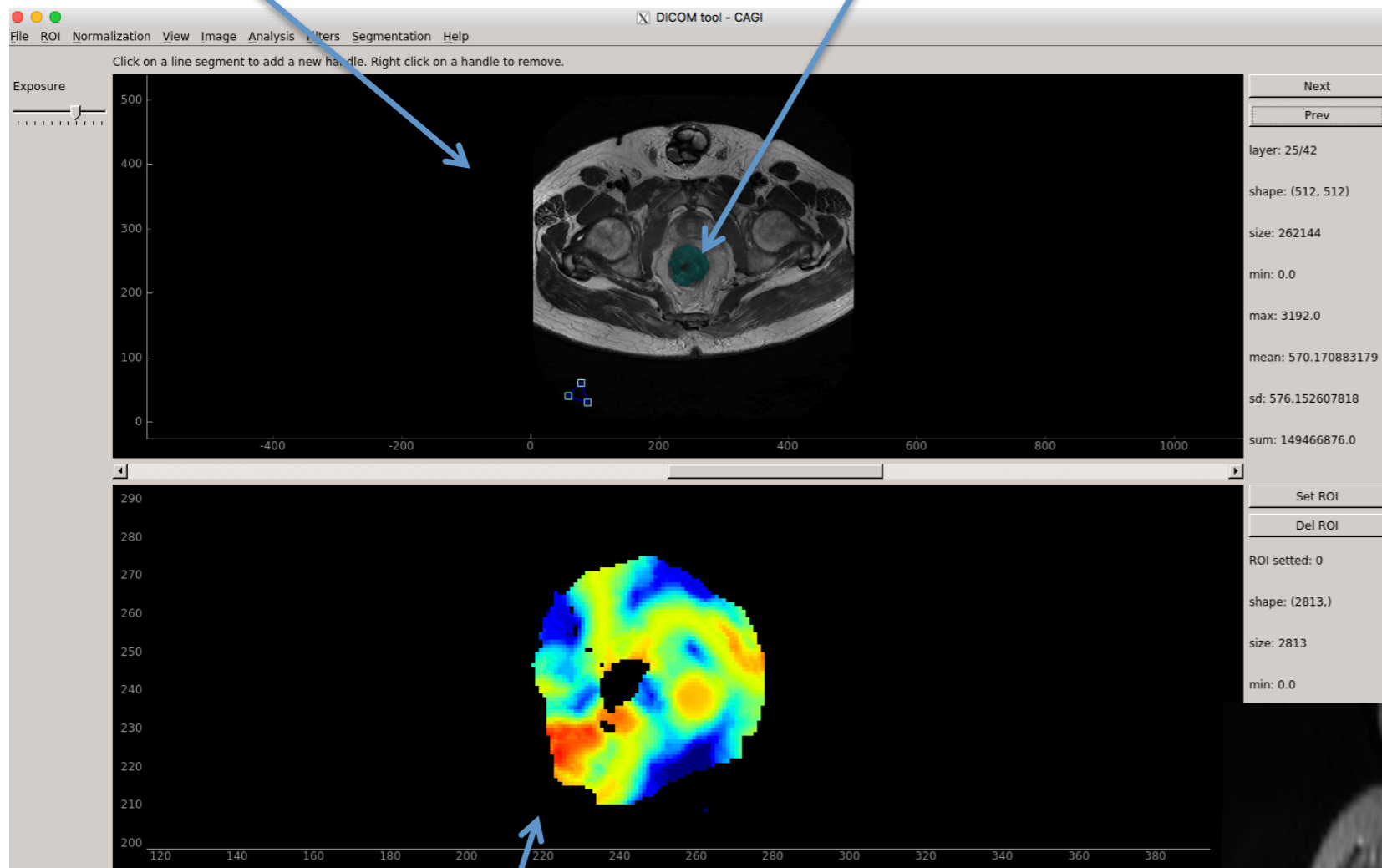
- We developed a custom open-source package in python  
[https://github.com/carlomt/dicom\\_tools](https://github.com/carlomt/dicom_tools)  
<https://pypi.org/project/dicom-tools/>
- The software is able to:
  - read and visualize images in dicom format - import and visualize ROI
  - draw new ROIs
  - allows grey-level intensity normalization (histogram matching and to a pre-defined ROI)
  - implements image filters
  - implements segmentation tools
  - implements texture analysis

# dicom\_tool

T2w slice

Region of interest (ROI) selected by hand by radiologists

Laplacian of Gaussian filter



local Shannon entropy

# dicom\_tool

The screenshot displays the DICOM tool application interface. The main window, titled "DICOM tool - ANFR", features a central image viewer showing an axial MRI scan of a human torso. A teal-colored region of interest (ROI) is highlighted on the scan. To the left of the image viewer is a vertical slider labeled "Exposure" with a value of 400. Below the image viewer is a horizontal axis with numerical labels from -200 to 700. To the right of the image viewer is a panel with several buttons: "Next", "Prev", "Set ROI", and "Del ROI". Below these buttons are statistics for the ROI: "layer: 22/31", "shapa: (512, 512)", "size: 262144", "min: 0.0", "max: 3254.0", "mean: 567.6141586303711", "sd: 502.80333555297534", and "sum: 154038526.0".

In the foreground, a terminal window titled "dicom\_tool interactive shell." displays the following text:

```
dicom_tool interactive shell.  
  
The dicom_tool application instance is called 'instance'  
so you can access its data members via 'instance.data'.  
If you loaded dicom files the content is stored in 'instance.data',  
the ROI in 'instance.ROI'.  
  
Python 3.6.5 (default, May 18 2018, 18:13:15)  
Type "copyright", "credits" or "license" for more information.  
  
IPython 5.7.0 -- An enhanced Interactive Python.  
?      -> Introduction and overview of IPython's features.  
%quickref -> Quick reference.  
help    -> Python's own help system.  
object? -> Details about 'object', use 'object??' for extra details.  
  
In [2]: instance.goToLayerWithLargerROI()  
  
In [3]: type(instance.data[22])  
Out[3]: numpy.ndarray  
  
In [4]:
```

It allows to open a python interactive shell from which you can call methods and have full access to memory

# Texture analysis

---

- Idea: different kinds of tissues have different texture
- We considered so far:
  - parameters from grey-level intensity histogram
  - Haralick parameters from grey-level co-occurrence matrix (dissimilarity contrast, energy, correlation, homogeneity)
  - Local Shannon entropy

# Textural features used in the analysis

---

First Order	Second Order
Mean	Minimum of the Shannon Entropy
Standard Deviation	Maximum of the Shannon Entropy
Skewness	Mean of the Shannon Entropy
Kurtosis	Standard Deviations of the Shannon Entropy
	Haralick Homogeneity
	Haralick Correlation
	Haralick Contrast
	Haralick Energy
	Haralick Dissimilarity

---

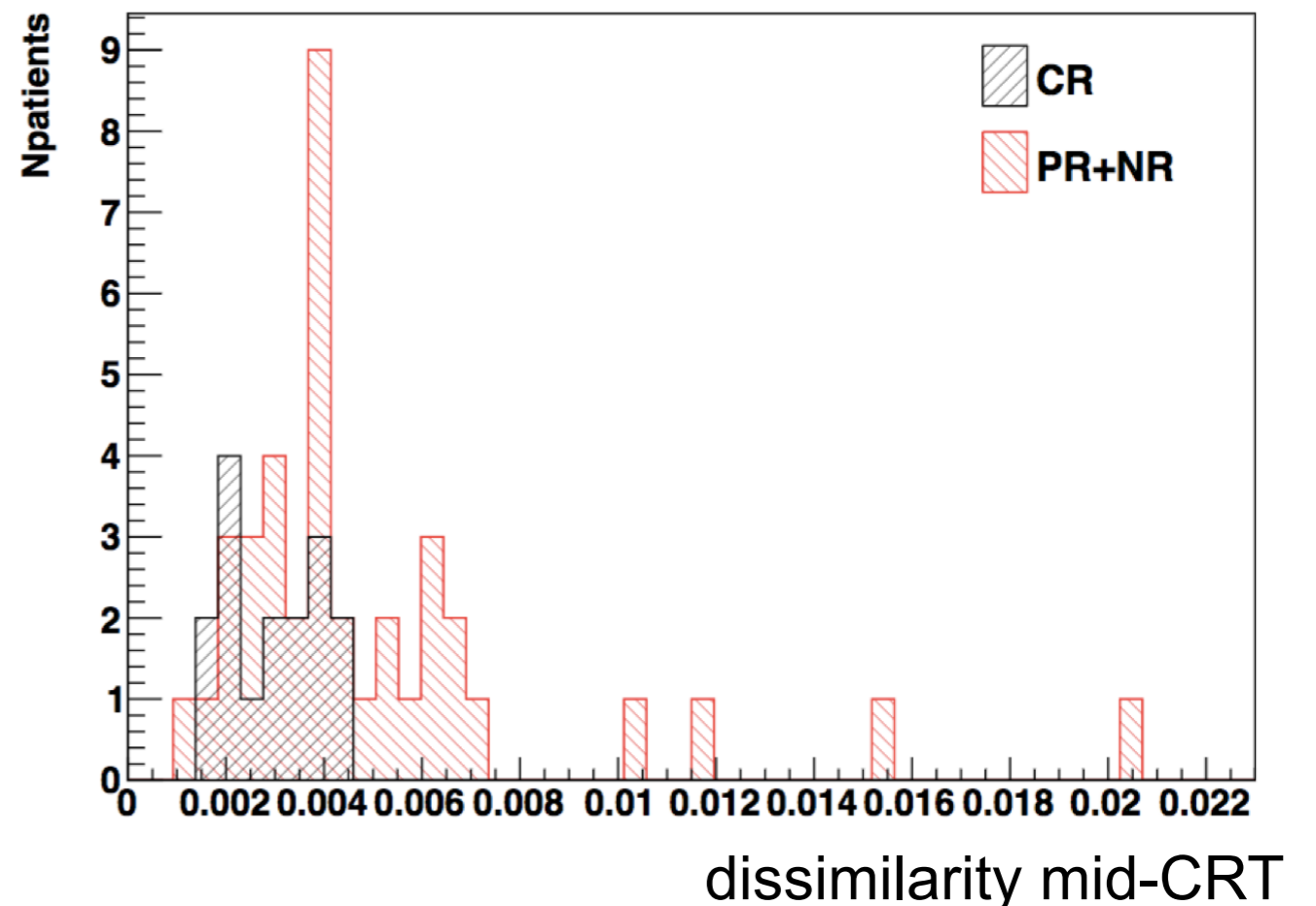
- First order parameters are the statistical parameters of the grey-level intensity histogram
- The second order parameters were computed after re-scaling the image to 8-bit (for computational reason)

# Results

---

- Some textural parameters are significantly different on average for the various classes of patients
- Distributions are wide and don't allow a separation of the classes on a patient basis

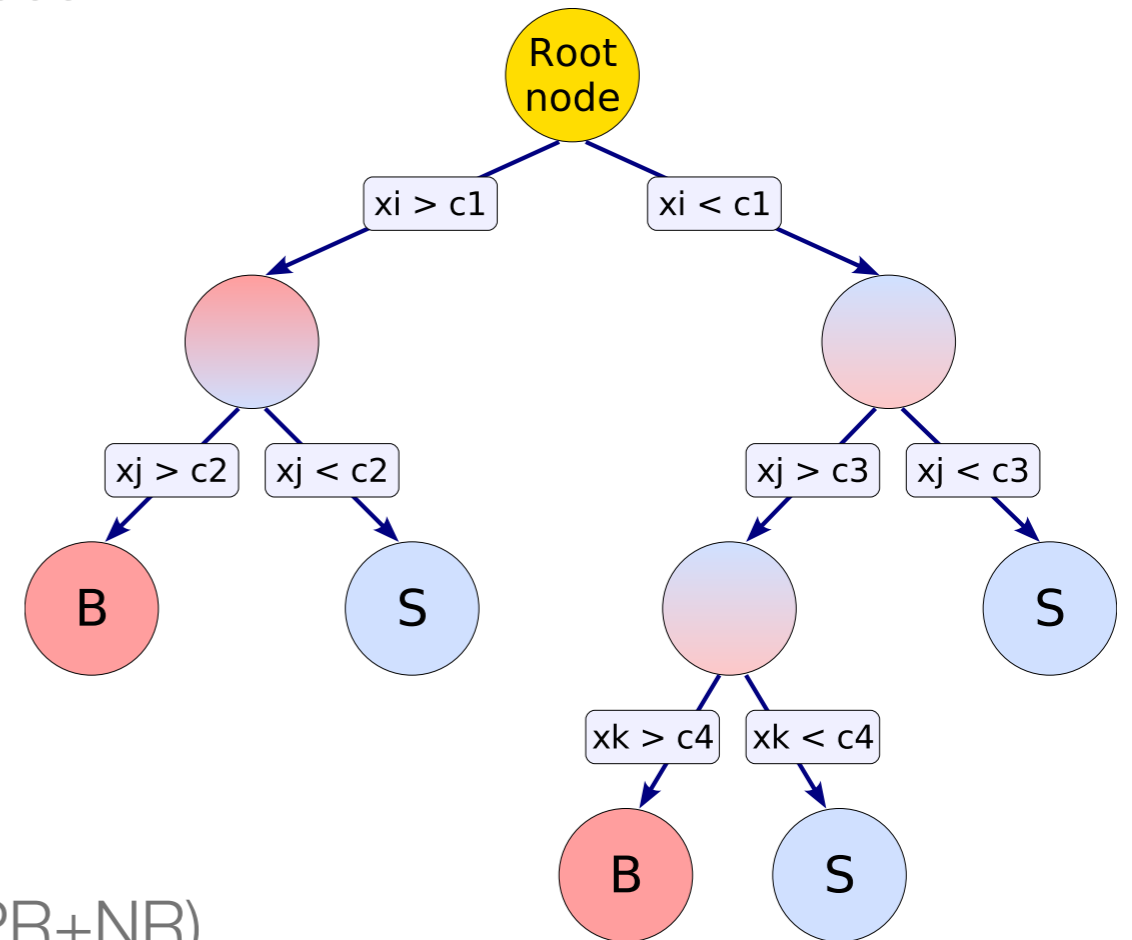
- It is unknown if these widths are due to instrumental effects or to biological variations





# Random Forest

- A classifier from the family of the decision trees
- Robust with small data samples
- 2000 trees, bagging fraction = 0.6
- Training cohort = 28 patients
- Validation cohort = 27 patients
- Two classifiers:
  - CR classifier: discriminate CR vs other (PR+NR)  
Use pre-CRT, mid-CRT, post-CRT features
  - NR classifier: discriminate NR vs other (CR+NR)  
Use pre-CRT, mid-CRT



[Image from TMVA Users' guide]

# Texture analysis results

- Most significant textural features (among O(100) features in total)

CR vs PR+NR discrimination

Parameter	CR	PR+NR
min(EntropyPost)/minEntropy(Pre) (3D)	$0.97 \pm 0.13$	$2.49 \pm 0.40$
min(EntropyPost)-minEntropy(Pre) (3D)	$-0.038 \pm 0.03$	$0.14 \pm 0.05$
min(EntropyPost)/minEntropy(Mid) (3D)	$1.07 \pm 0.13$	$2.41 \pm 0.42$
ROI area Pre (2D)	$2537 \pm 202$	$4016 \pm 366$
dissimilarity Mid (2D)	$0.00297 \pm 0.00025$	$0.00505 \pm 0.00060$

NR vs CR+PR discrimination

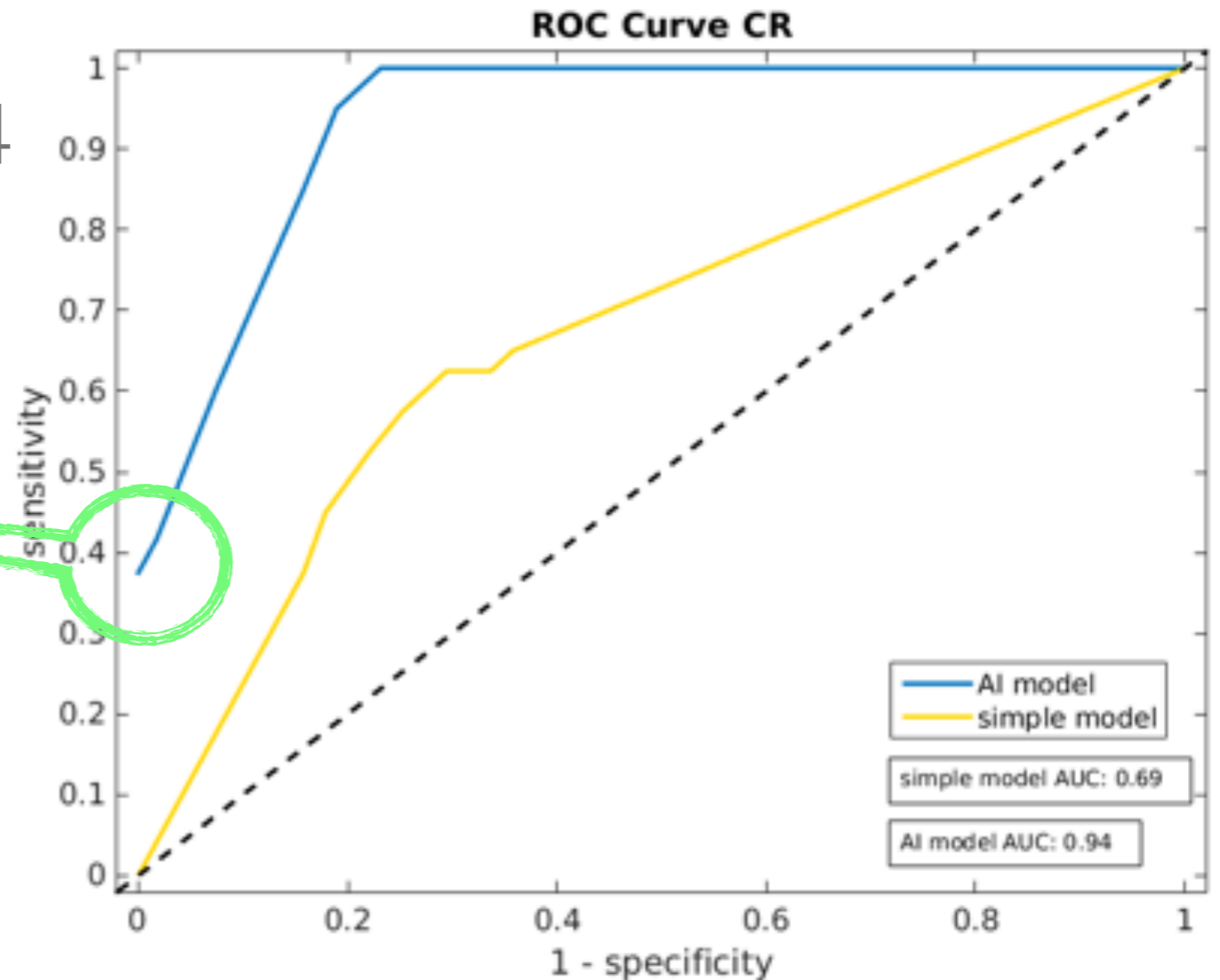
Parameter	NR	CR+PR
standard deviation Pre (3D)	$123 \pm 11$	$172 \pm 8$
mean(Entropy) Pre (2D)	$0.618 \pm 0.061$	$0.814 \pm 0.028$
stdDev(Entropy) Pre (2D)	$0.4180 \pm 0.0082$	$0.469 \pm 0.011$
max(Entropy) Pre (2D)	$1.870 \pm 0.078$	$2.248 \pm 0.056$

Notes:

- ranking based on the p-value of a t-student test
- Pre=pre-CRT, Mid=mid-CRT, Post=post-CRT

# Random Forest results

- CR Classifier accuracy
- Area Under Curve = 0.94
- Sensitivity  
 $38 \pm 15\%$



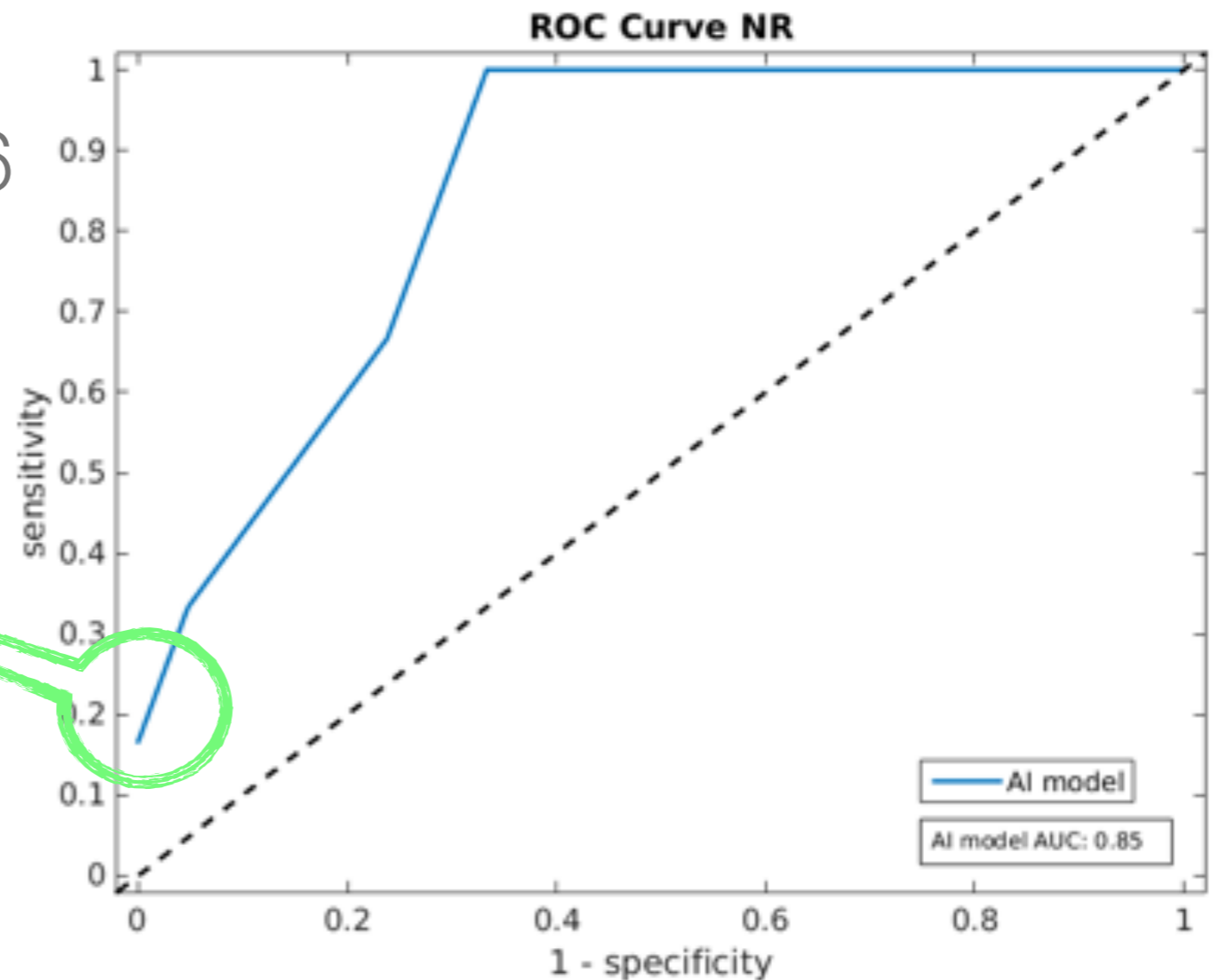
simple model =  
classification based only  
on tumor volume

**sensitivity** = true positive rate  
=> efficiency

**specificity** = true negative rate  
=> 1-contamination

# Random Forest results

- NR Classifier accuracy
- Area Under Curve = 0.86
- Sensitivity  
 $18 \pm 8\%$



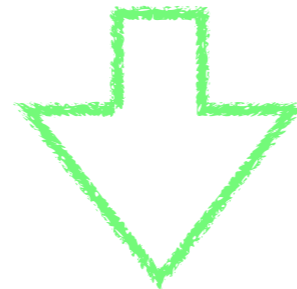
**sensitivity** = true positive rate  
=> efficiency

**specificity** = true negative rate  
=> 1-contamination

# Conclusions

---

- Machine Learning based analysis of quantitative features of T2-w MRI images is sensitive to response to CRT in rectal cancer
- Distributions of features are wide  
instrumental effects or biological variations?

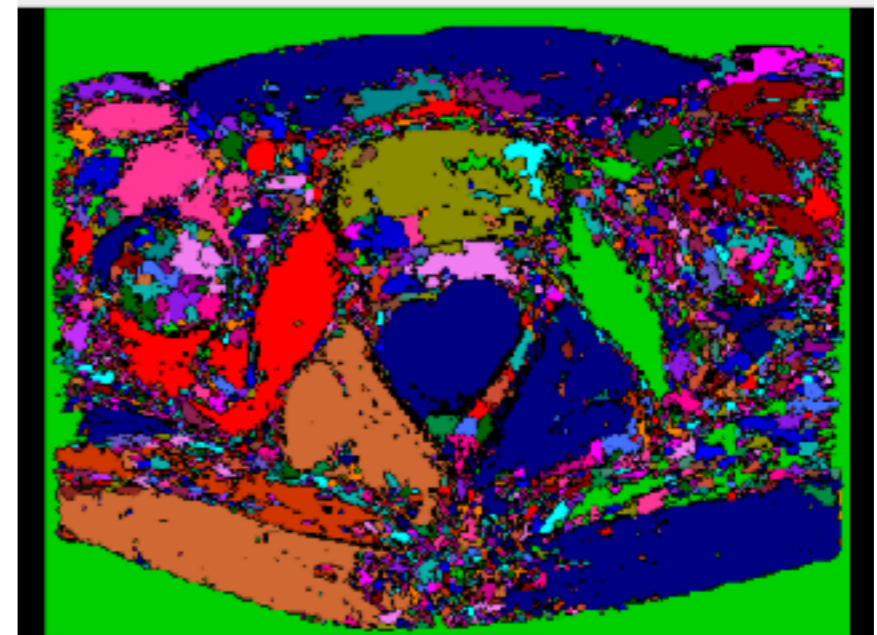
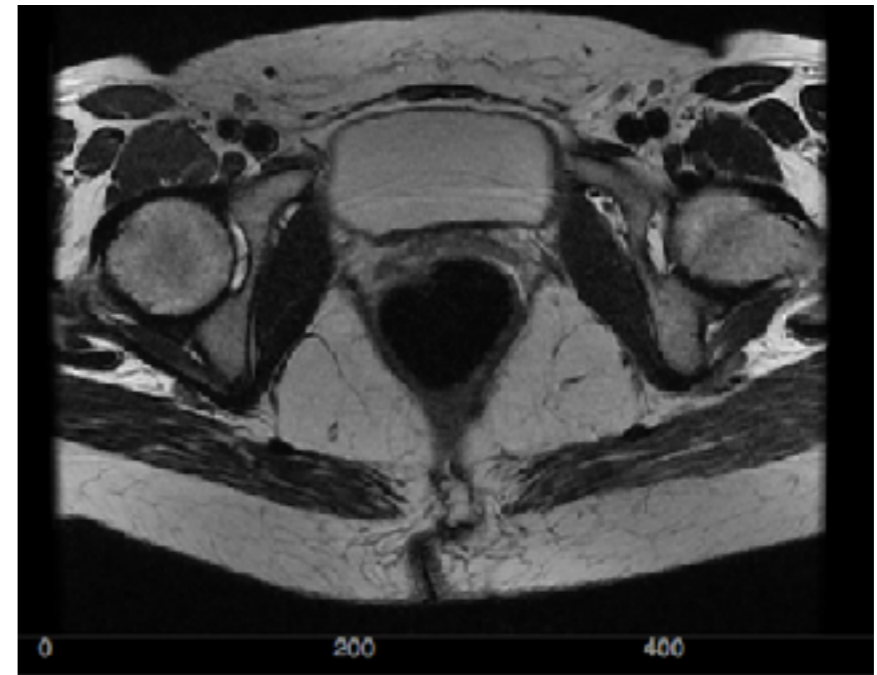


- Crucial to improve image segmentation and calibration

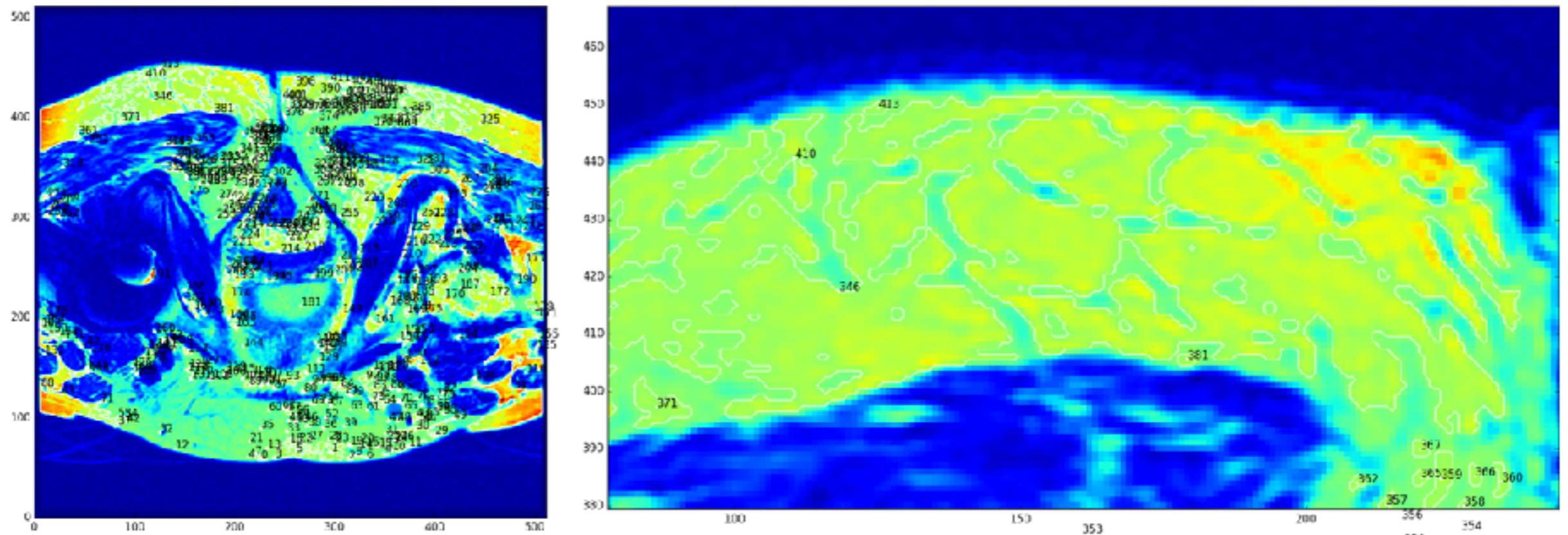
# Morphological Watersheds

---

- Preliminary results are encouraging
- We have to optimize all the parameters
- 3D segmentation



# Segmentation with Asterism

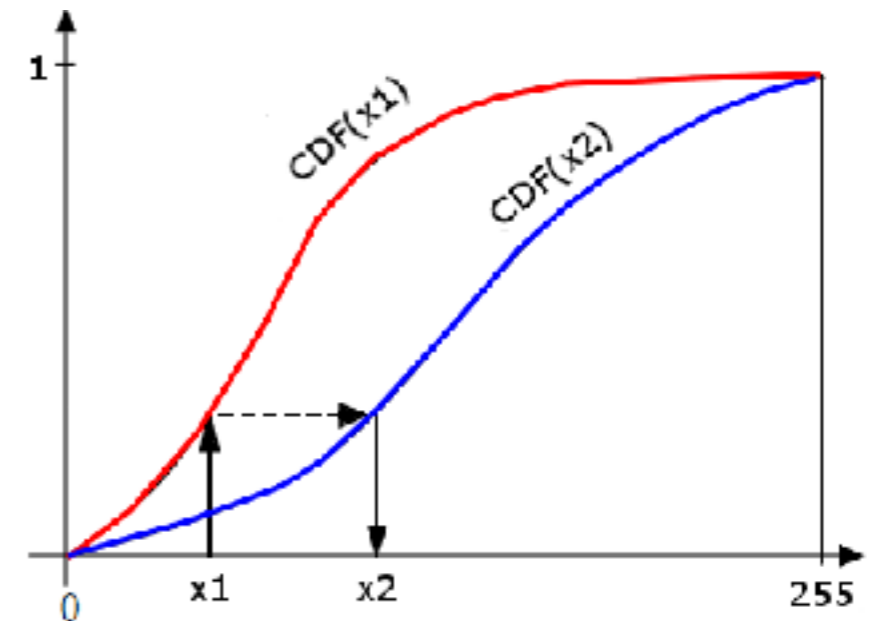


- Asterism is a python tool developed by Andrea Tramacere (Observatory of Geneva)
- Made for finding clusters in astrophysical images (Fermi)

# Histogram matching

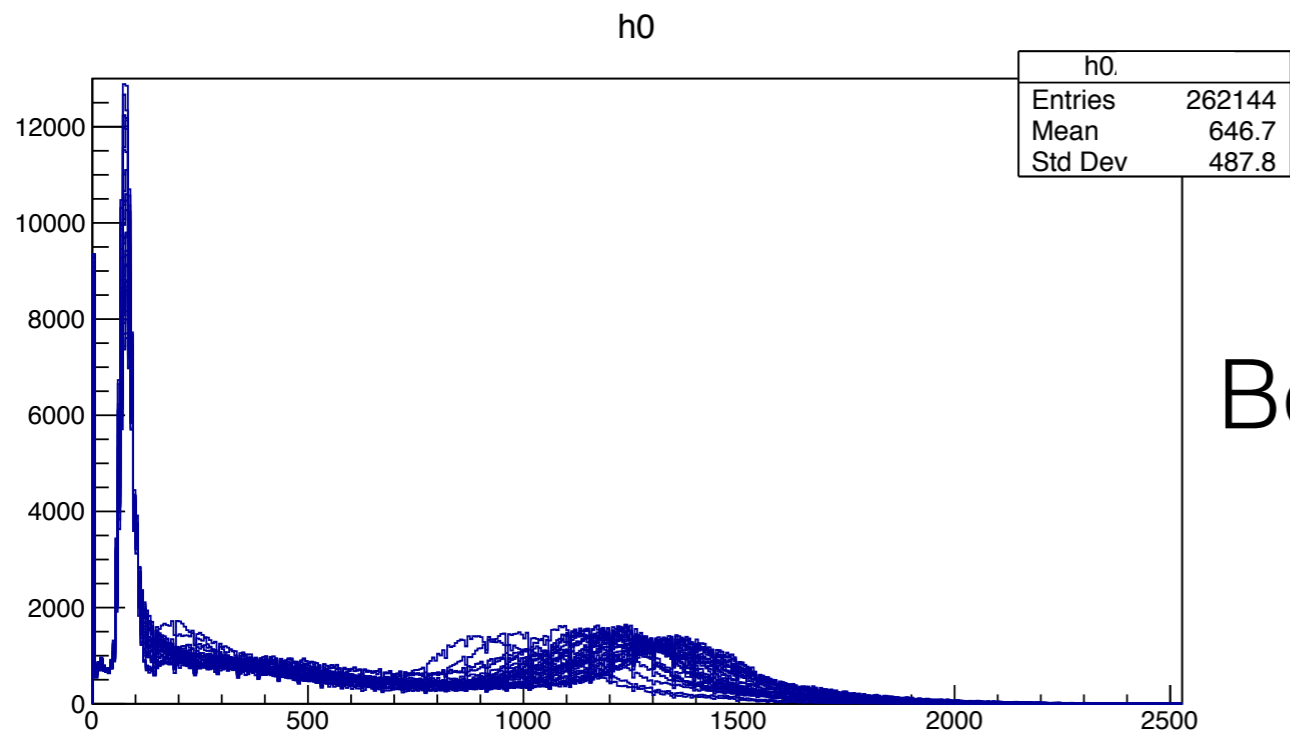
---

- One image is used as template
- It assumes the same amount of pixel per region in the image and in the template
- Recognizes regions using the cumulative distribution of the image and the template
- Assigns the same region color to the image pixels as the template

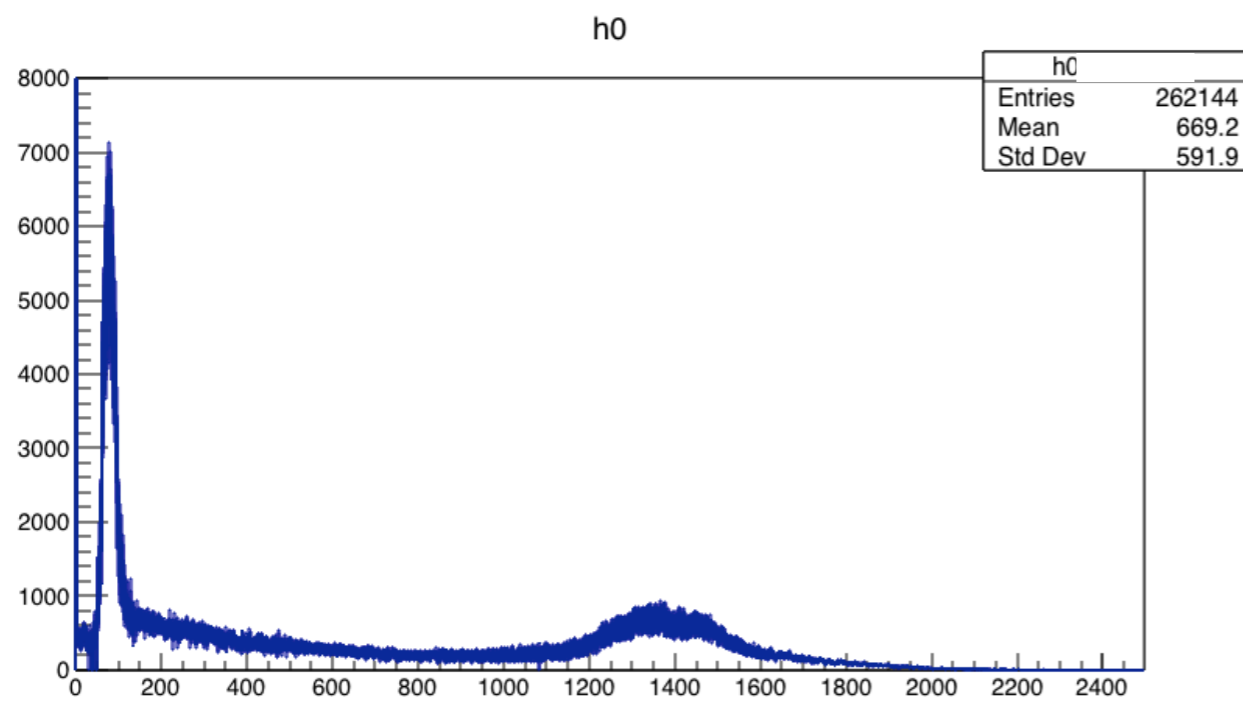
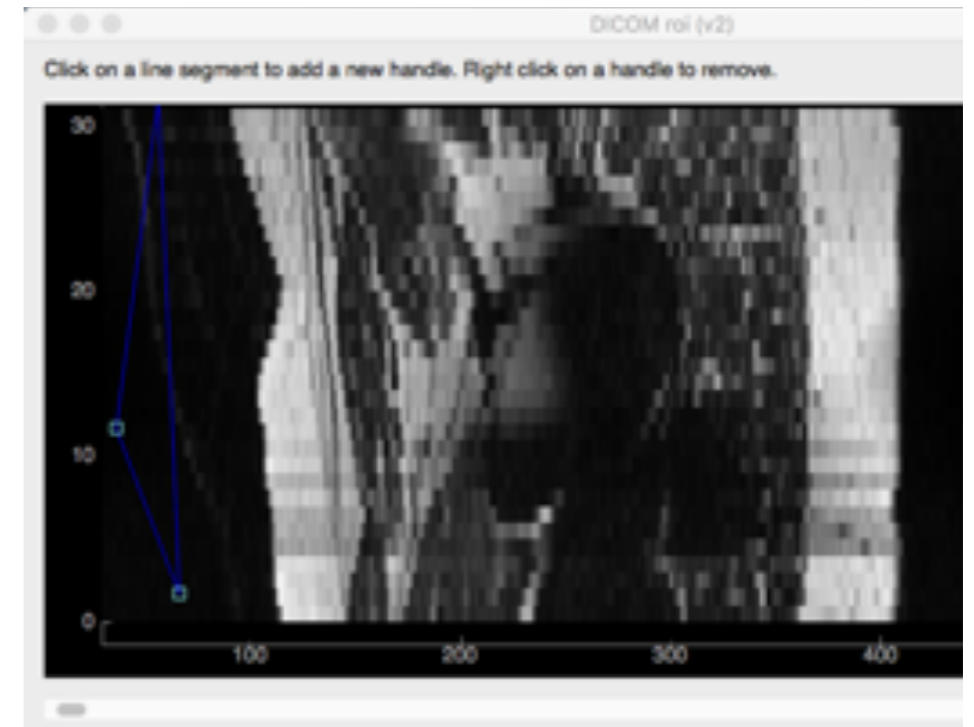




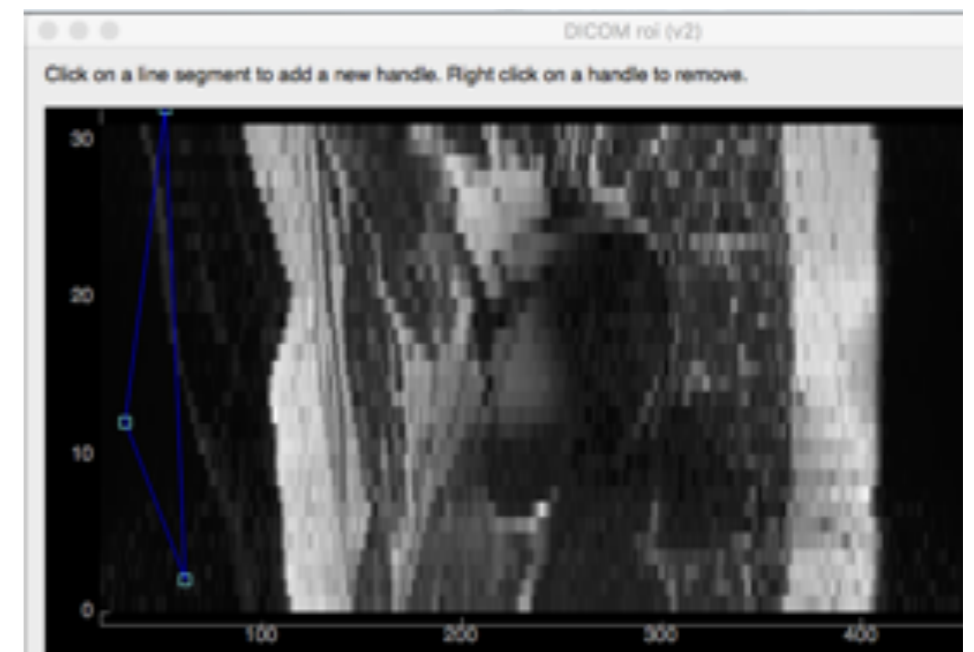
# Results of Histogram Matching



Before

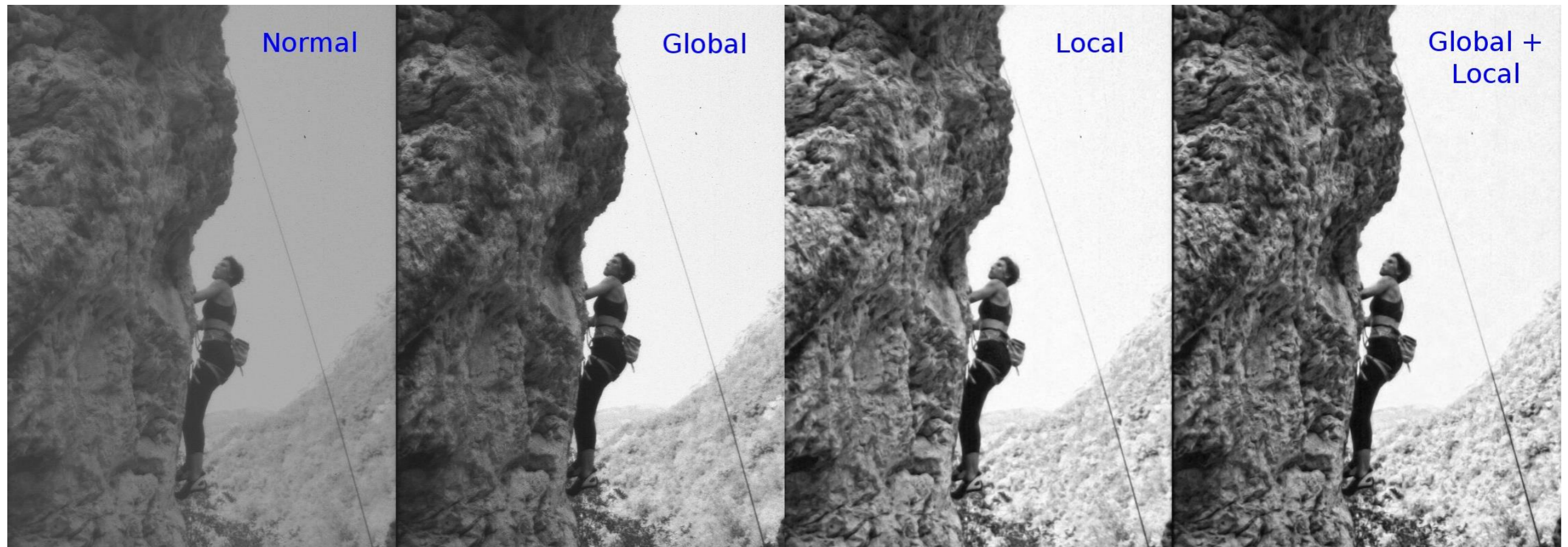


After



# Contrast Limited Adaptive Histogram Equalization

---



- Histogram equalization for each section of the image
- Contrast amplification is limited to avoid noise amplification

# Segmentation with Deep Learning

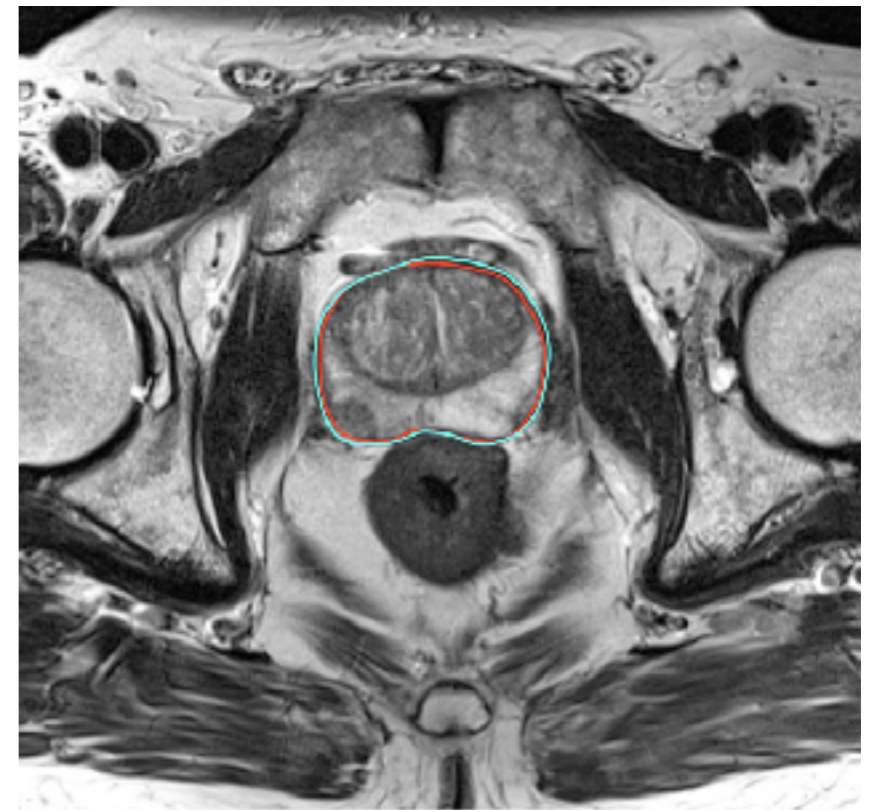
---

- Identify a portion of an image given the context
- Many deep learning applications are about object recognition
- Simple task for trained humans but hard for algorithms
- Challenge:
  - Small dataset

# ProMISe Challenge

---

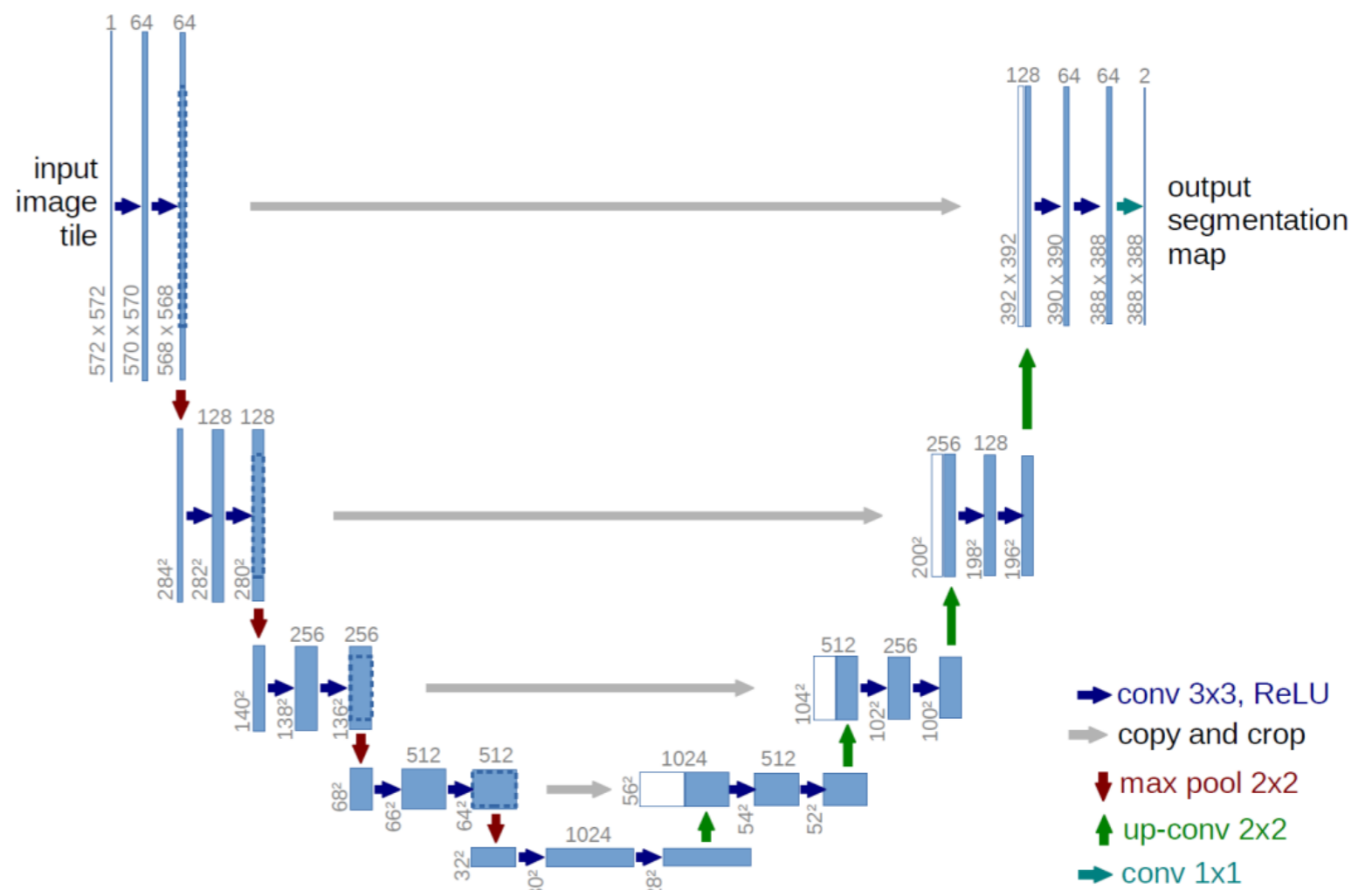
- Prostate MR Image Segmentation 2012
- The data includes both patients with benign prostatic hyperplasia and prostate cancer.
- The data includes both patients with benign prostatic hyperplasia and prostate cancer.
- 50 cases (about 500 usable slices)
- Reference segmentation included



<https://promise12.grand-challenge.org>

# U-net

- developed for biomedical image segmentation
- fully convolutional network
- consists of a contracting path and an expansive path



[img. from: Ronneberger et. al. arXiv:1505.04597v1]

# Data augmentation

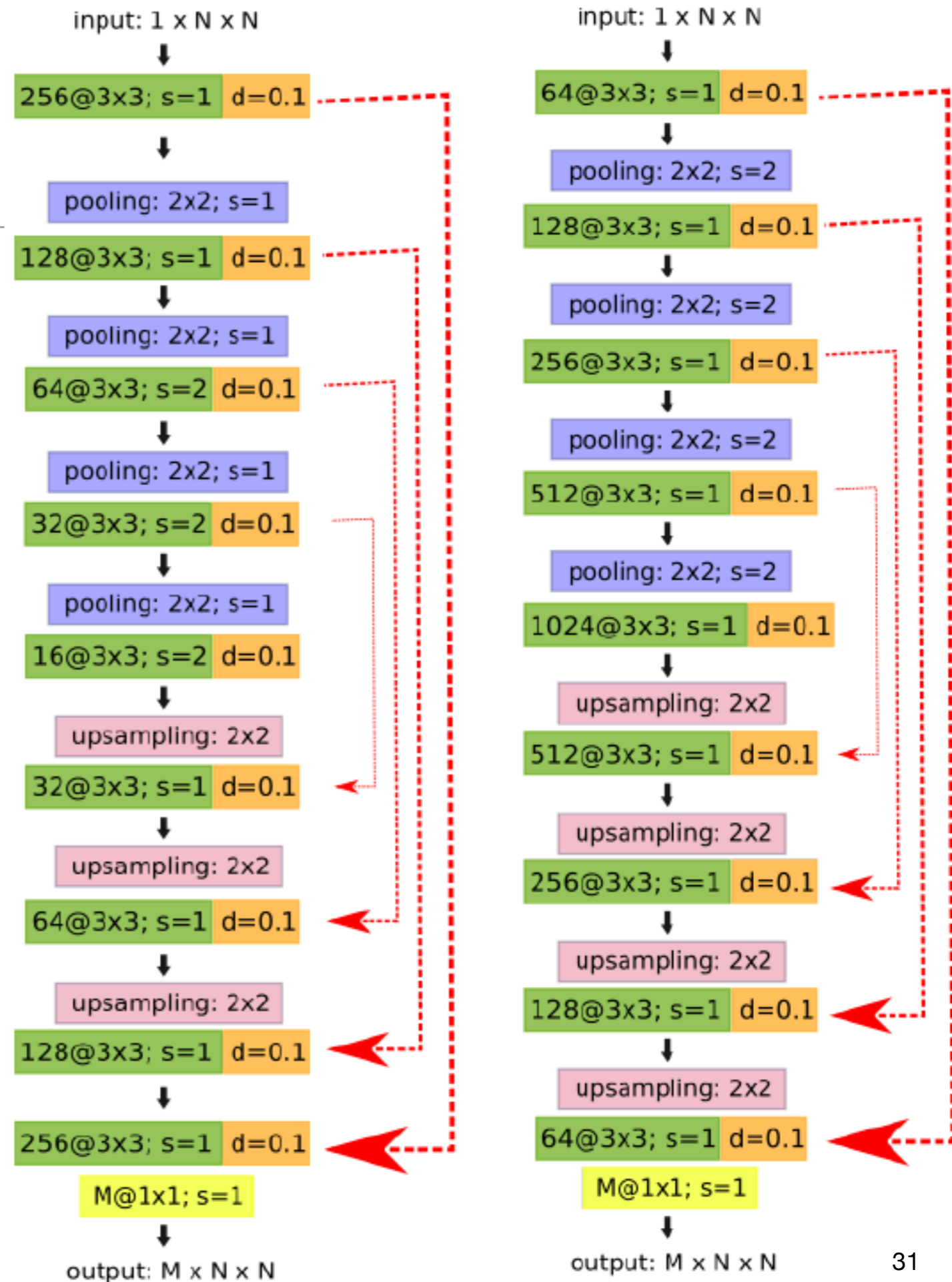
---

- Essential to teach the network the desired invariance and robustness properties
- Especially when few training samples are available
- Shift and rotation
- Deformations and intensity variations
- Random elastic deformations

# Modifications of U-net

- All Drop-Out
- 30 M parameters
- dropout after every convolutional
- InvertedNet
- less parameters
- 1.4 M parameters

[img. from: A.Novikov et al.  
arXiv:1701.08816v4]



# Cost function and its optimisation

---

- Dice's coefficient (or Sørensen index, or also similarity coefficient) is for estimating the similarity of two samples
- It equals twice the number of elements common to both sets divided by the sum of the number of elements in each set:

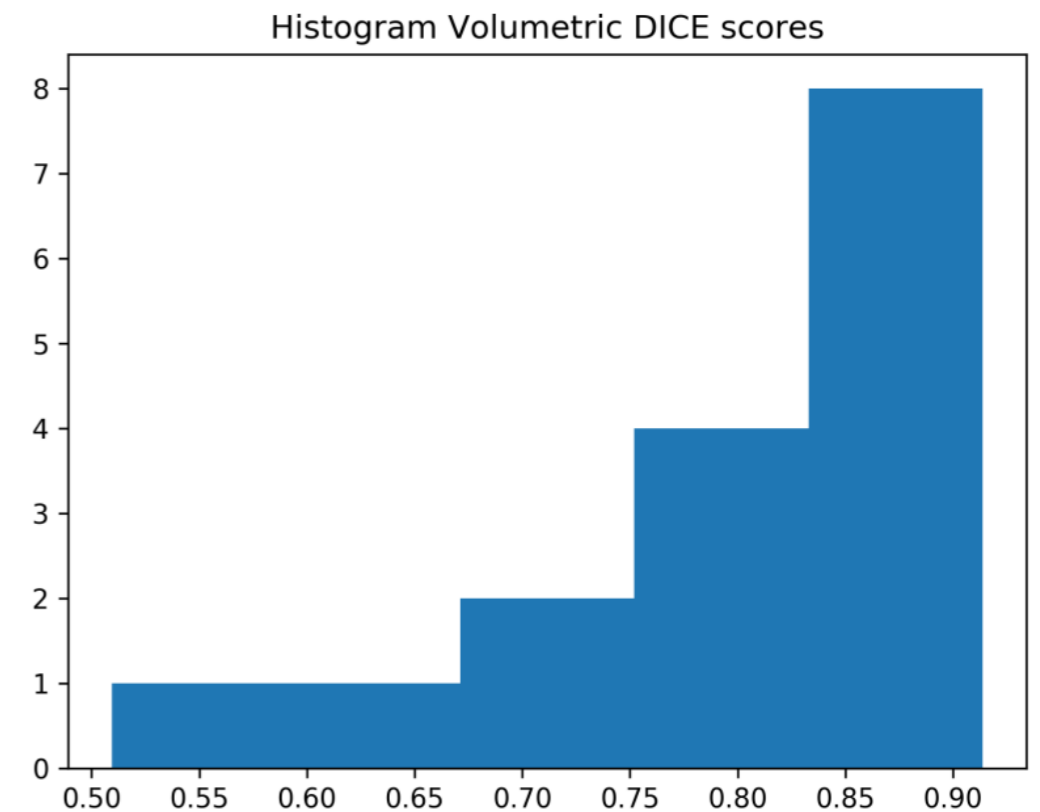
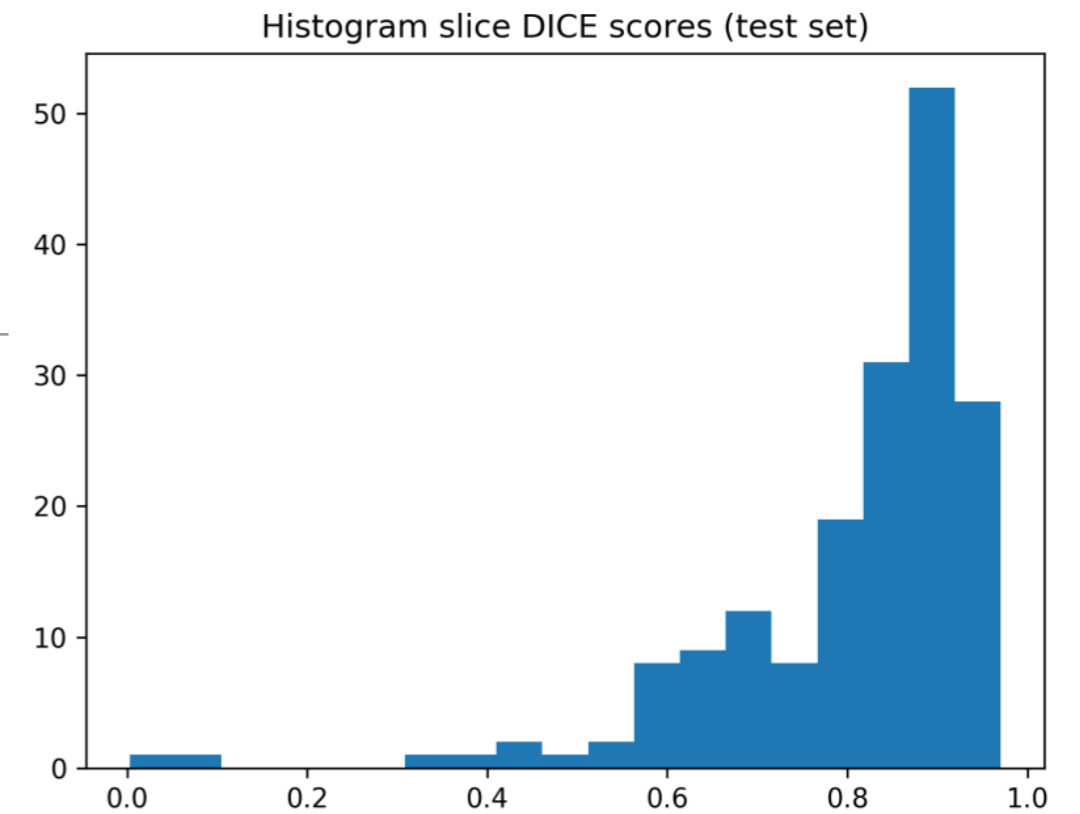
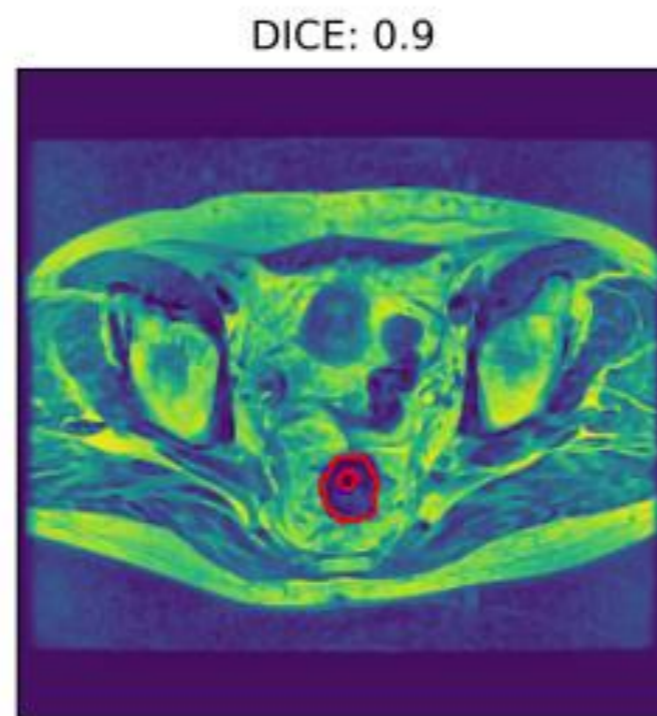
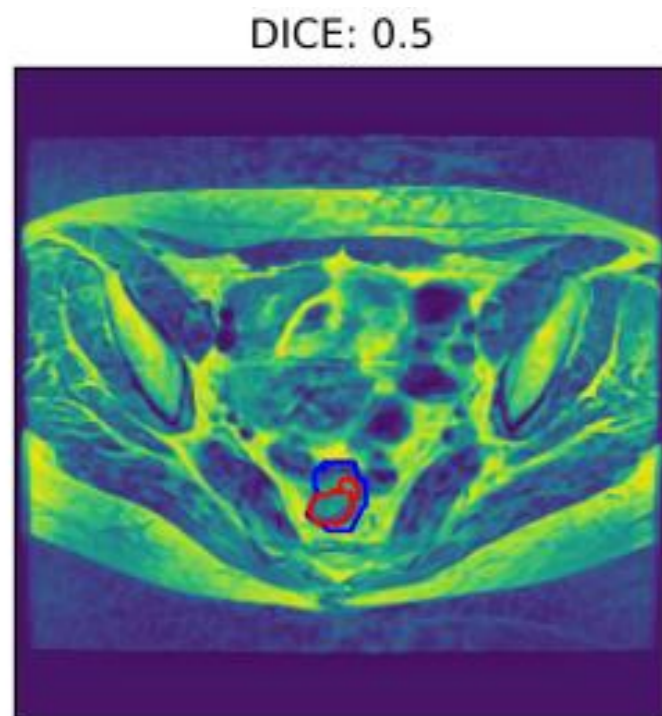
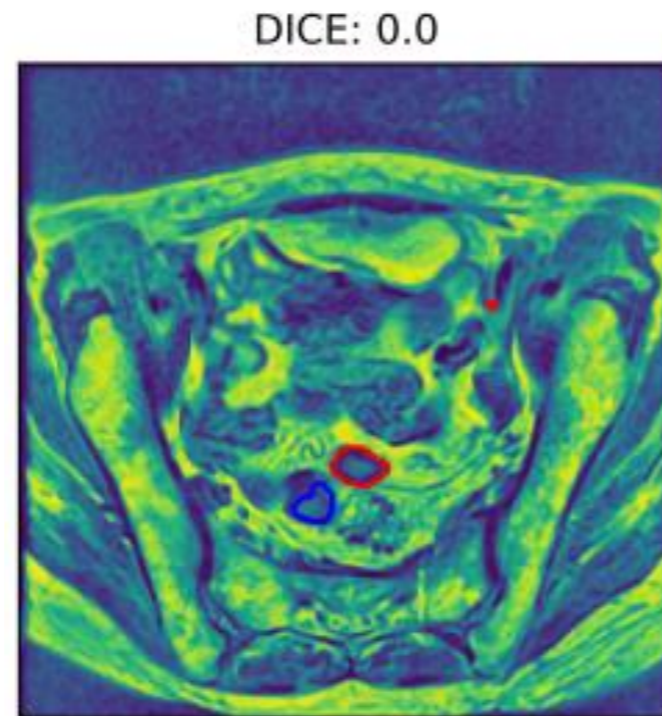
$$C_{Dice} = 2 \frac{\sum_i (x_i \cdot \chi_i) + \epsilon}{\sum_i (x_i + \chi_i) + \epsilon}$$

- Optimised using Adam (Adaptive Moment estimation), a stochastic gradient descent algorithm
- The speed is function of the running averages of the gradients and the second moments of the gradient



# Results - Inverted Net

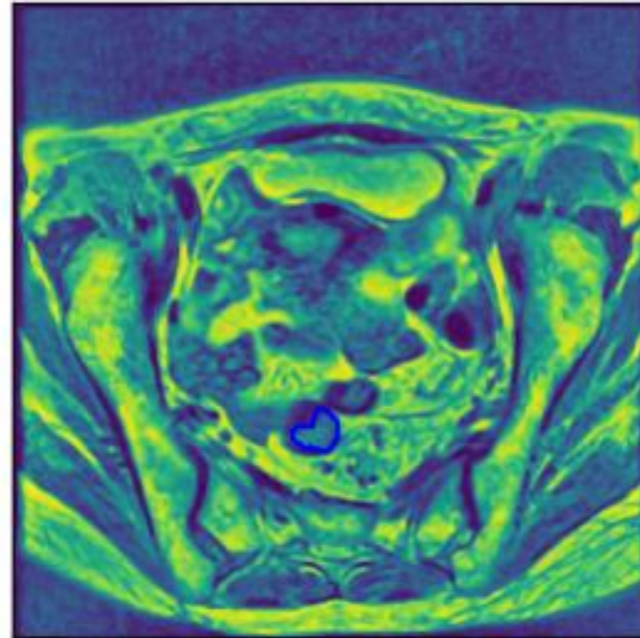
- 1.4 M parameters
- 38 patients for training
- 16 for test



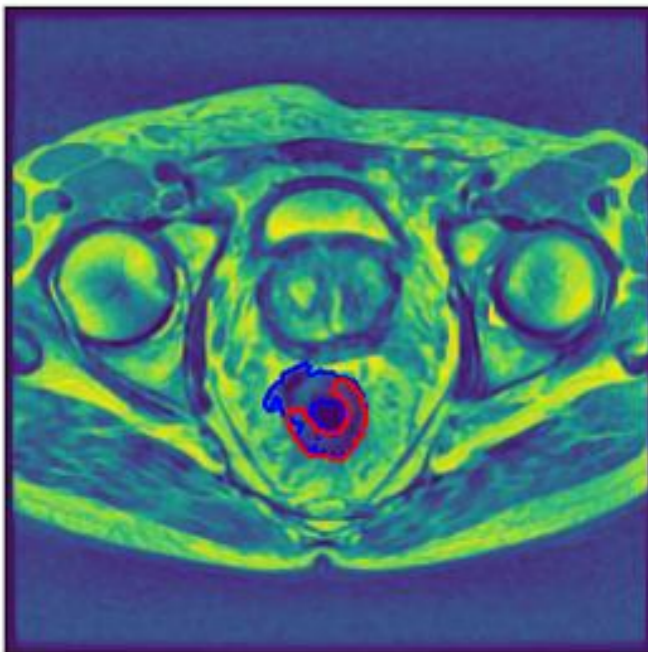
# Results - All Drop-Out

- 30 M parameters
- 4 times slower than InvertedNet

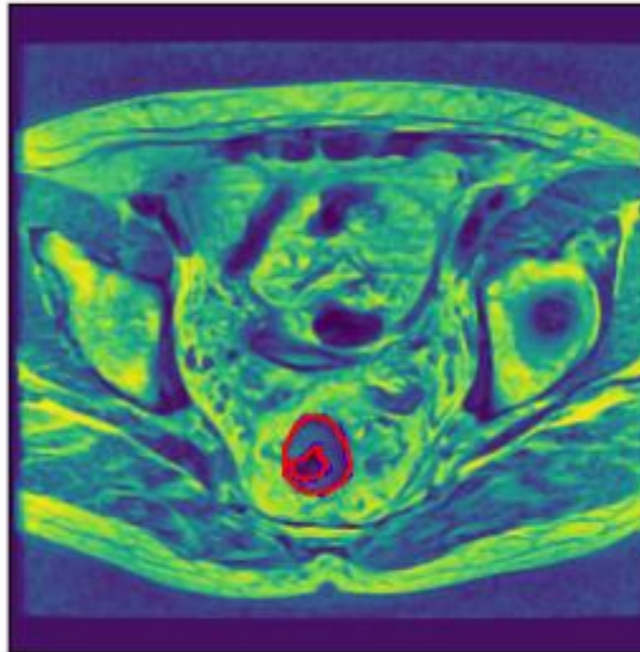
DICE: 0.0



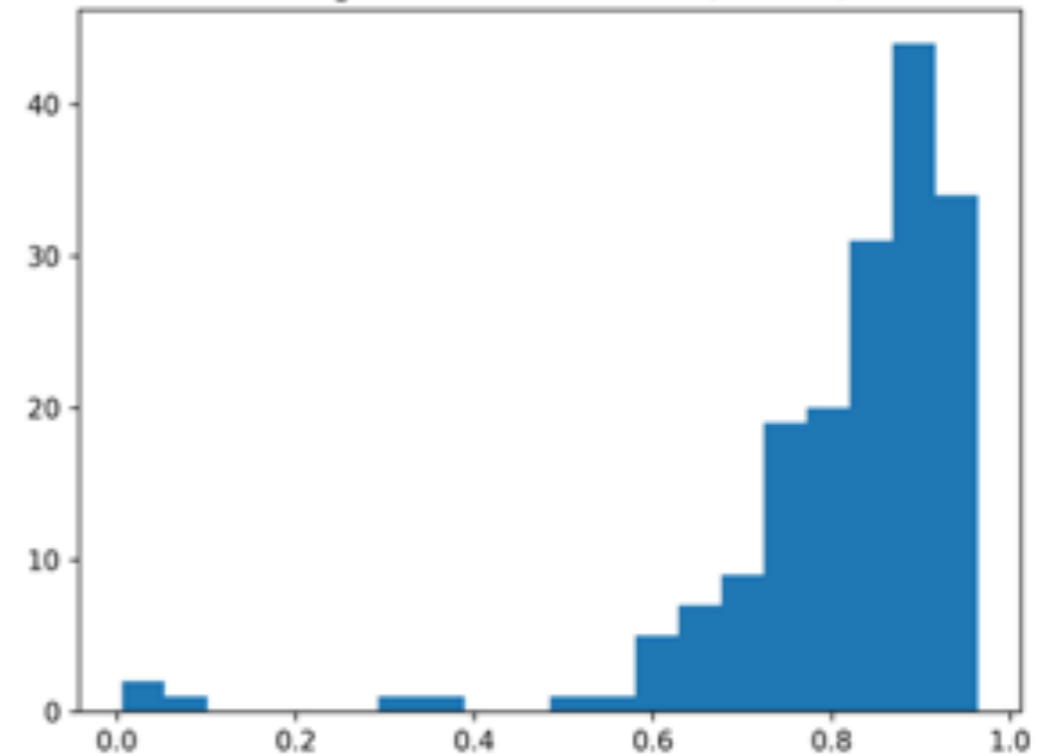
DICE: 0.5



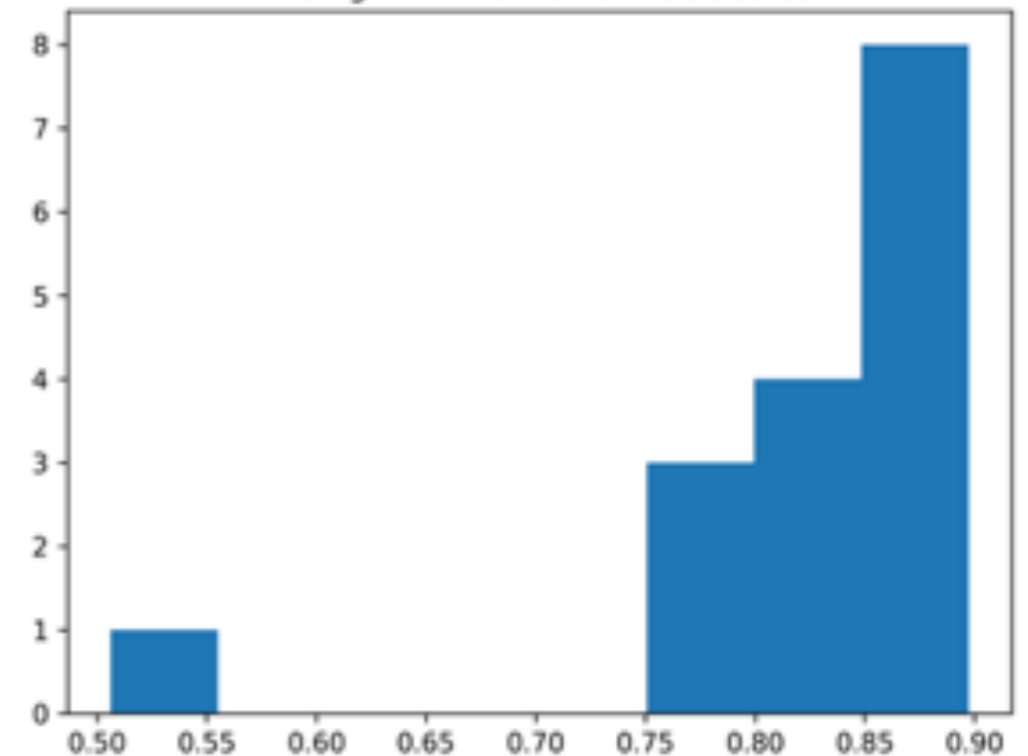
DICE: 0.9



Histogram slice DICE scores (test set)



Histogram Volumetric DICE scores



# Summary

---

- MR Images could be analysed to find features not-visible even to trained Medical Doctors
- We developed a tool to import images converting them to 3D numerical tensors
- It allows to pre-process the data and compute texture parameters
- We started to apply Deep Learning algorithms

Conditional Variational Auto Encoder (VAE)  
to simulate accurately low energy nuclear interactions

# Geant4 (GEometry ANd Traking)

---

- Developed by an International Collaboration
  - Established in 1998
  - Approximately 100 members, from Europe, US and Japan
  - <http://geant4.org>
- Open source
- Written in C++ language
  - Takes advantage from the Object Oriented software technology

[Geant4, a simulation toolkit Nucl. Inst. and Methods Phys. Res. A, 506 250-303

Geant4 developments and applications Transaction on Nuclear Science 53, 270-278]



# Geant4 applications

---

- Nuclear Physics experiments
- High Energy Physics
- but also:
  - Hadrontherapy
  - Radiobiology
  - and many others...

# Geant4 applications: hadrontherapy

---

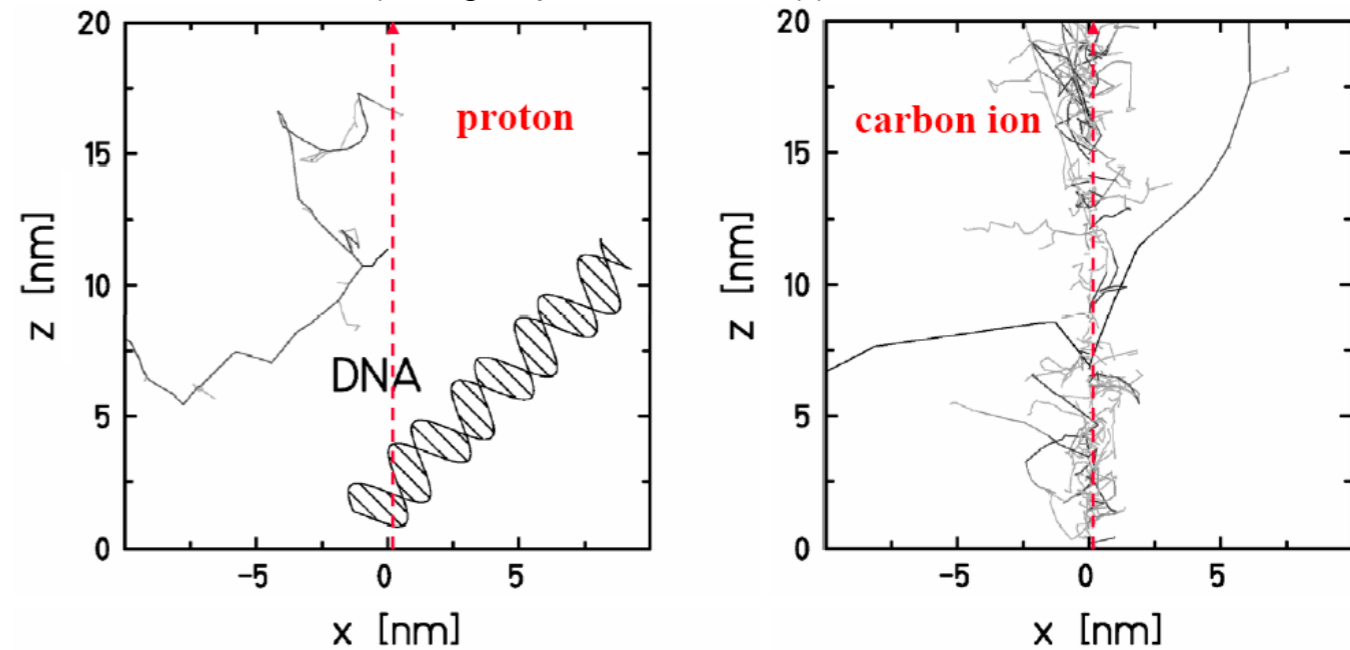
- Hadrontherapy:
  - External radiation therapy using strongly interacting particles to treat mainly tumour
  - Mainly with p and C ions
- MC codes are used to:
  - Generate input parameters of the treatment planning algorithms
  - Validate the dose calculation of such algorithms
  - Estimate the production of  $\beta$  emitters, such as  $^{11}\text{C}$  and  $^{15}\text{O}$
  - Link the production of prompt  $\gamma$  with the dose distribution



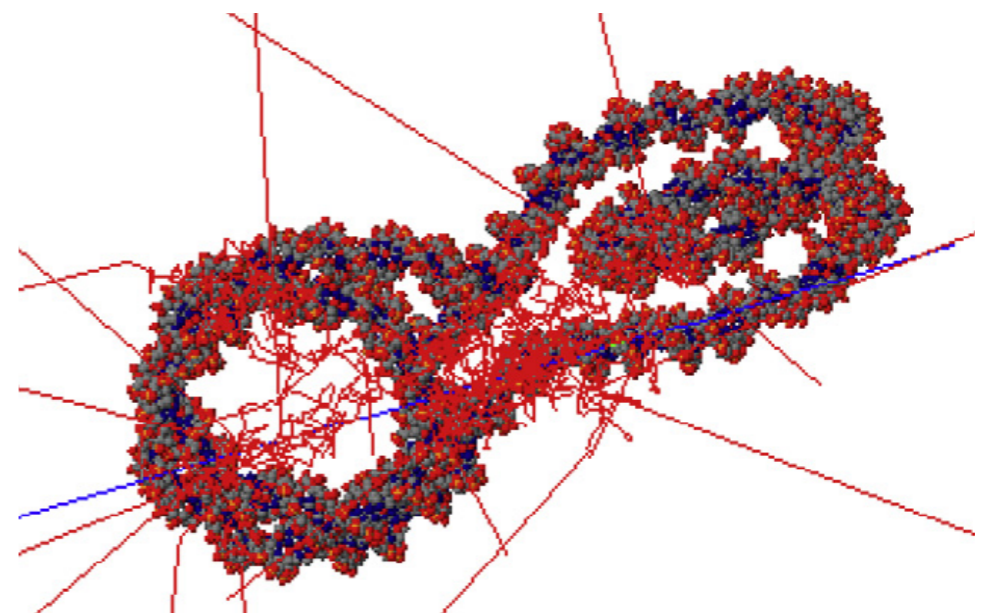
# Geant4 applications: radiobiology

- To link the physical dose deposited to the biological effectiveness
- Geant4 has a dedicated package for modelling early biological damage induced by ionising radiation at the DNA scale (Geant4-DNA)

image from: U. Amaldi and G. Kraft,  
Rep. Prog. Phys., vol. 68, no. 8, pp. 1861–1882, Jul. 2005.



atomistic view of a dinucleosome  
irradiated by a single 100 keV proton  
Image from M. A. Bernal et al Physica Medica, vol. 31, no. 8, pp.  
861–874, Dec. 2015.





# Geant4, further applications

- Radio-protection in space mission
- Shielding for satellites
- Single event upset and radiation damages to electronics
- Simulations for nuclear spallation sources
- Radioactive waste



First slide of the talk “ESA Geant4 R&D Activities from the Geant4 Space User Workshop Hiroshima, 26 August 2015

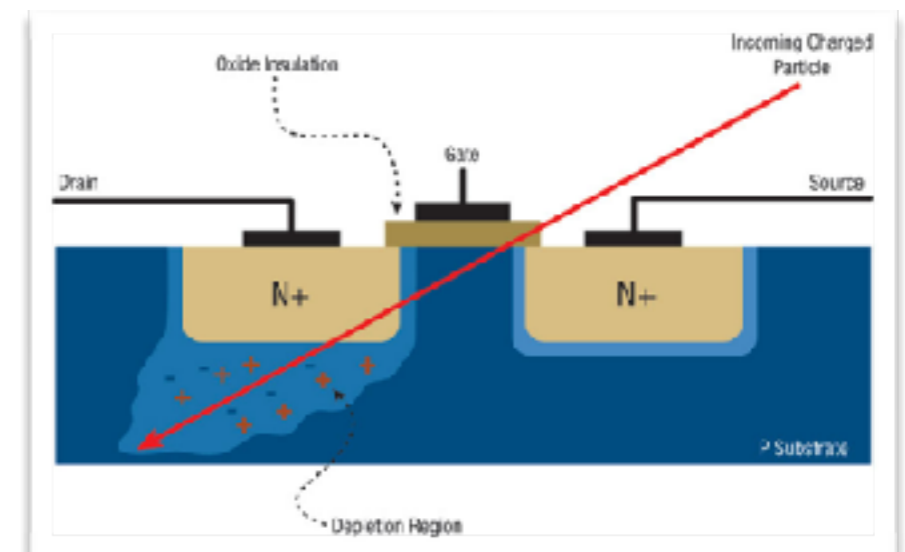


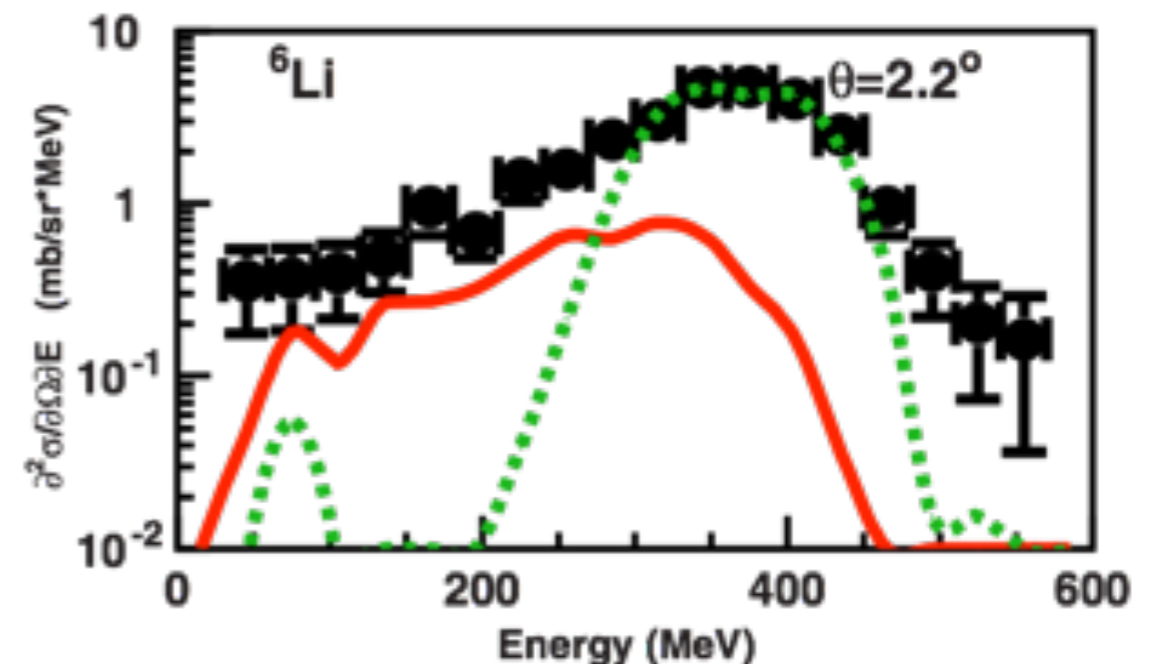
Figure from M. Sawant, COTS Journal Jan. 2012

# Problems below 100MeV/A

- Despite the numerous and relevant application would use it, there is no dedicated model to nuclear interaction below 100 MeV/A in Geant4
- Many papers showed the difficulties of Geant4 in this energy domain:
  - Braunn et al. have shown discrepancies up to one order of magnitude in  $^{12}\text{C}$  fragmentation at 95 MeV/A on thick PMMA target
  - De Napoli et al. showed discrepancy specially on angular distribution of the secondaries emitted in the interaction of 62 MeV/A  $^{12}\text{C}$  on thin carbon target
  - Dudouet et al. found similar results with a 95 MeV/A  $^{12}\text{C}$  beam on H, C, O, Al and Ti targets

- **Exp. data**
- **G4-BIC**
- **G4-QMD**

[Plot from De Napoli et al. Phys. Med. Biol., vol. 57, no. 22, pp. 7651–7671, Nov. 2012]

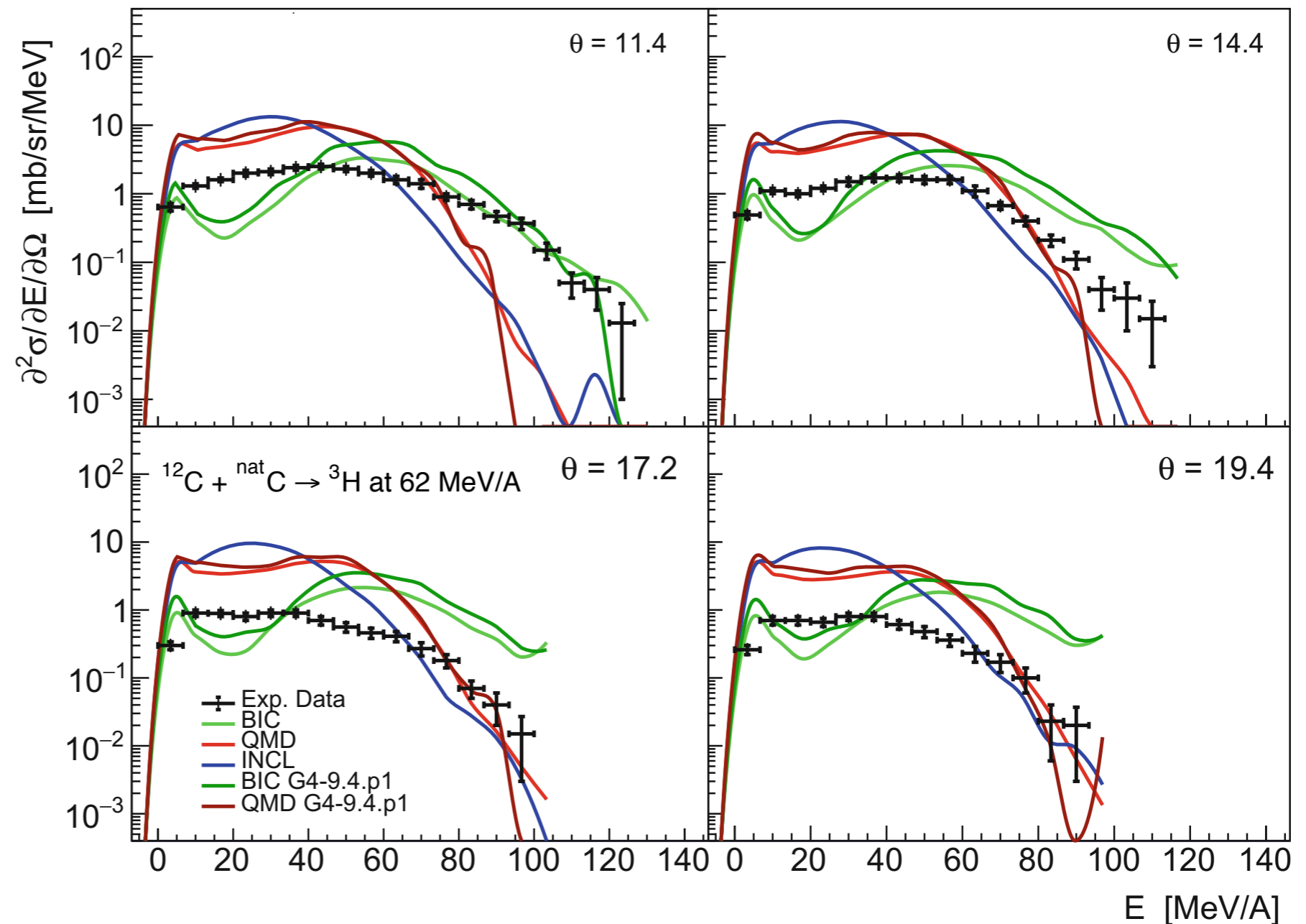


Cross section of the  $^6\text{Li}$  production at 2.2 degree in a  $^{12}\text{C}$  on  $^{nat}\text{C}$  reaction at 62 MeV/A.

# Update of a $^{12}\text{C}$ fragmentation benchmark

[Update of the benchmark originally published on De Napoli et al. Phys. Med. Biol., vol. 57, no. 22, pp. 7651–7671, Nov. 2012]

- 62 MeV/A  $^{12}\text{C}$  on thin carbon target
- doubly differential cross sections
- INCL was not available at the time of the original publication



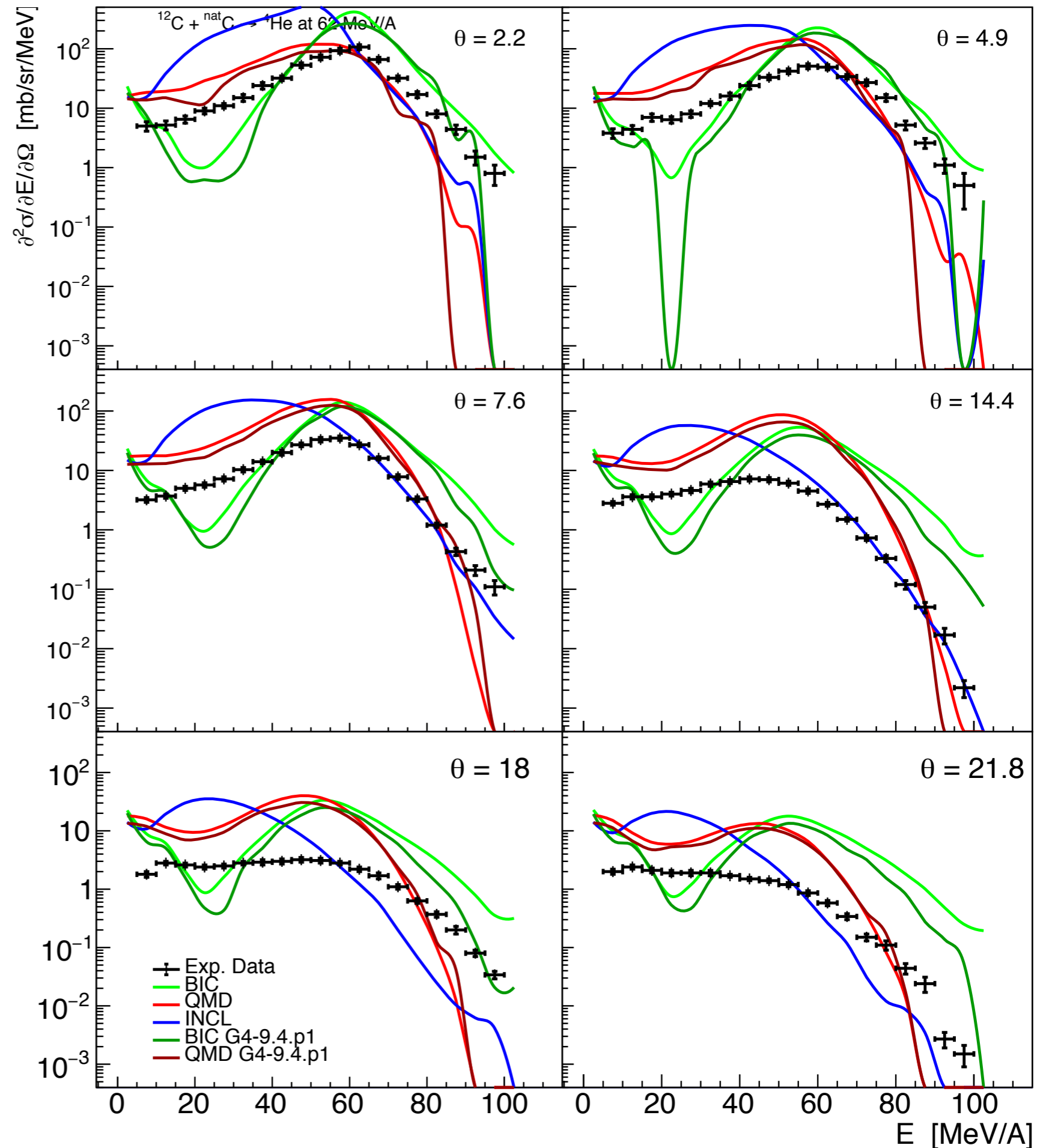
[C. Mancini-Terracciano et al. IFMBE Proceedings Series 68/1 (2018), pp. 675–685. doi: 10.1007/978-981-10-9035-6\_126]

# Update of a $^{12}\text{C}$ fragmentation benchmark

[Update of the benchmark originally published on De Napoli et al. Phys. Med. Biol., vol. 57, no. 22, pp. 7651–7671, Nov. 2012]

- alpha

[C. Mancini-Terracciano et al. IFMBE Proceedings Series 68/1 (2018), pp. 675–685. doi: 10.1007/978-981-10-9035-6\_126]



# About GeNIALE (Geant Nuclear Interaction At Low Energy)

---

- Aims at improving the capacity of Geant4 to simulate low energy nuclear reactions
- Granted by the INFN National Scientific Committee 5 (CSN5)
- CSN5 is devoted to technological and inter-disciplinary research
- The core of GeNIALE is the implementation in Geant4 of a new model for the first stage of the interaction between a hadron -or a nucleus- and a target nucleus
- Such a model will be coupled with the models already implemented in Geant4 for the second stage, and with the Geant4 framework in general



# Suitable models

---

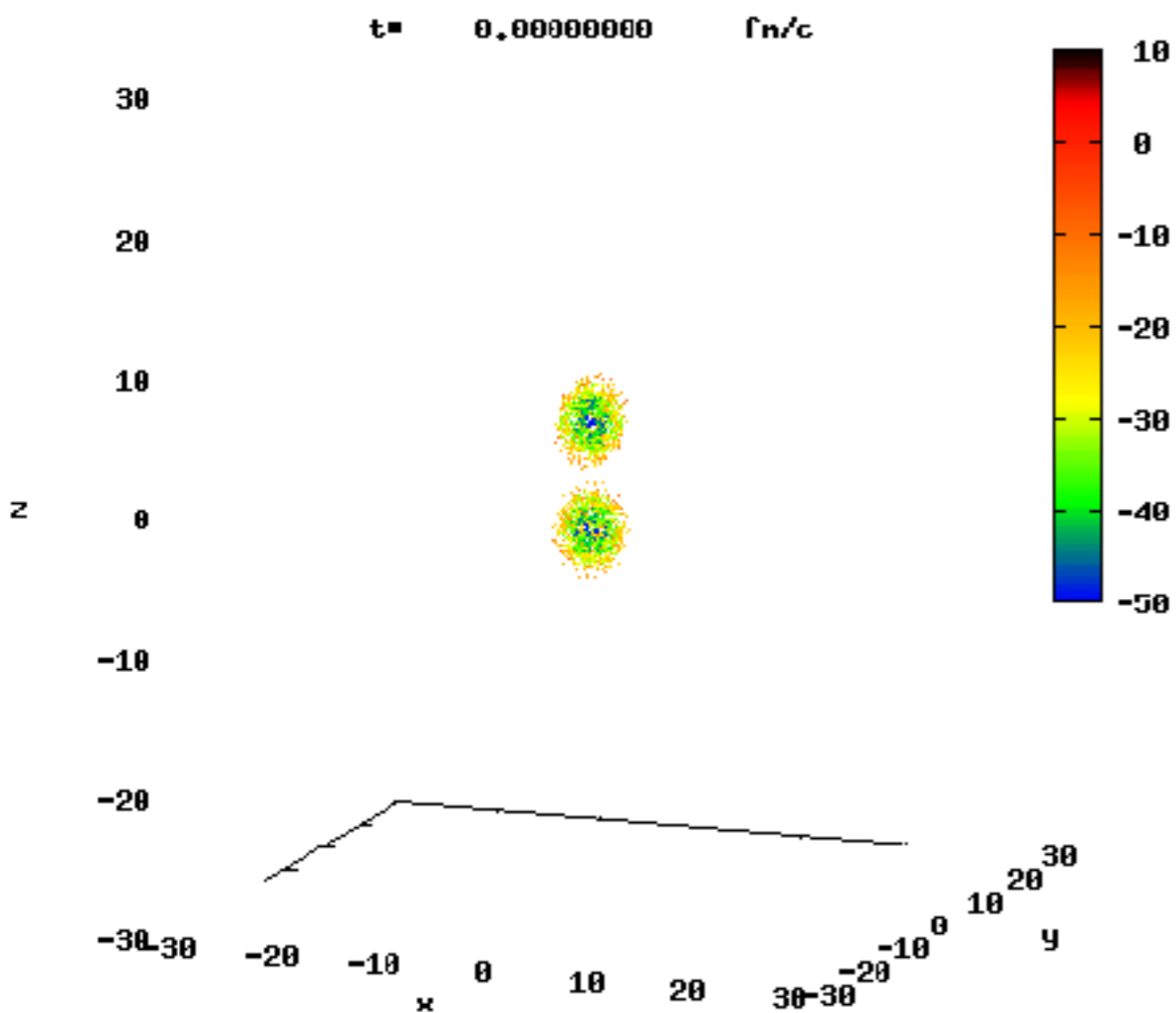
## SMF (Stochastic Mean Field)

- Developed by Maria Colonna (INFN LNS, Catania)
- describes the time evolution of the density distribution
- involves the implementation of an effective attractive mean-field nuclear interaction
- mean-field is self-consistent, depends on the density
- includes two-bodies correlations through nucleon-nucleon collisions

## BLOB (Boltzmann-Langevin One Body)

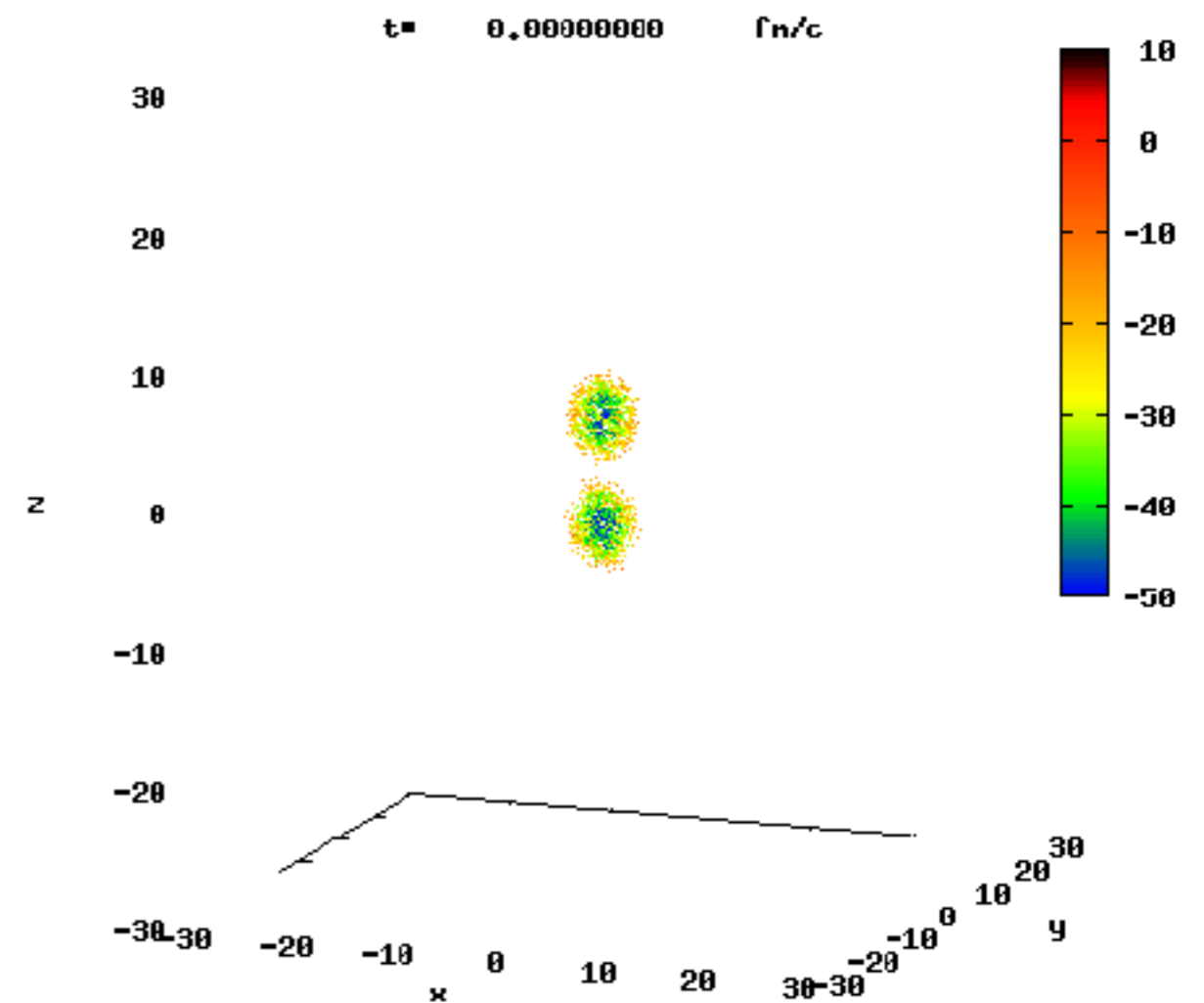
- Implemented by Paolo Napolitani (IPN, Orsay)
- Derived from SMF
- Adds fluctuations in the dynamics treating the nucleon-nucleon collisions as a stochastic process

# SMF and BLOB



SMF

100 test particles per nucleon  
 $^{12}\text{C}$  on  $^{12}\text{C}$  at 62 MeV/n



BLOB

# Geant4 interface to SMF and BLOB

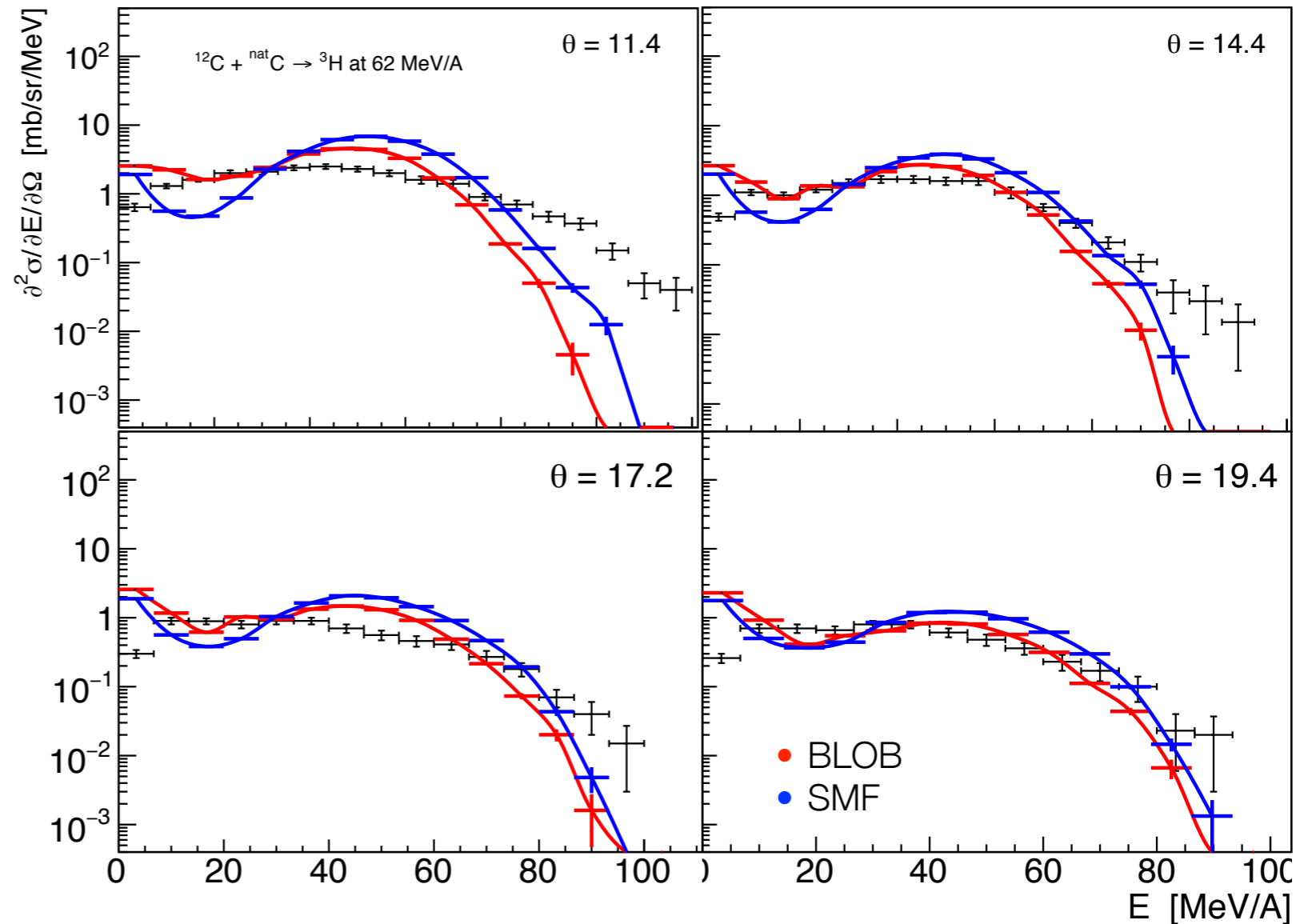
---

- Dummy G4-model, loads the output from SMF/BLOB
- Sample the final state
  - Fragments mass and charge
  - Gas particles emitted
- Applies Geant4 de-excitation to excited fragments



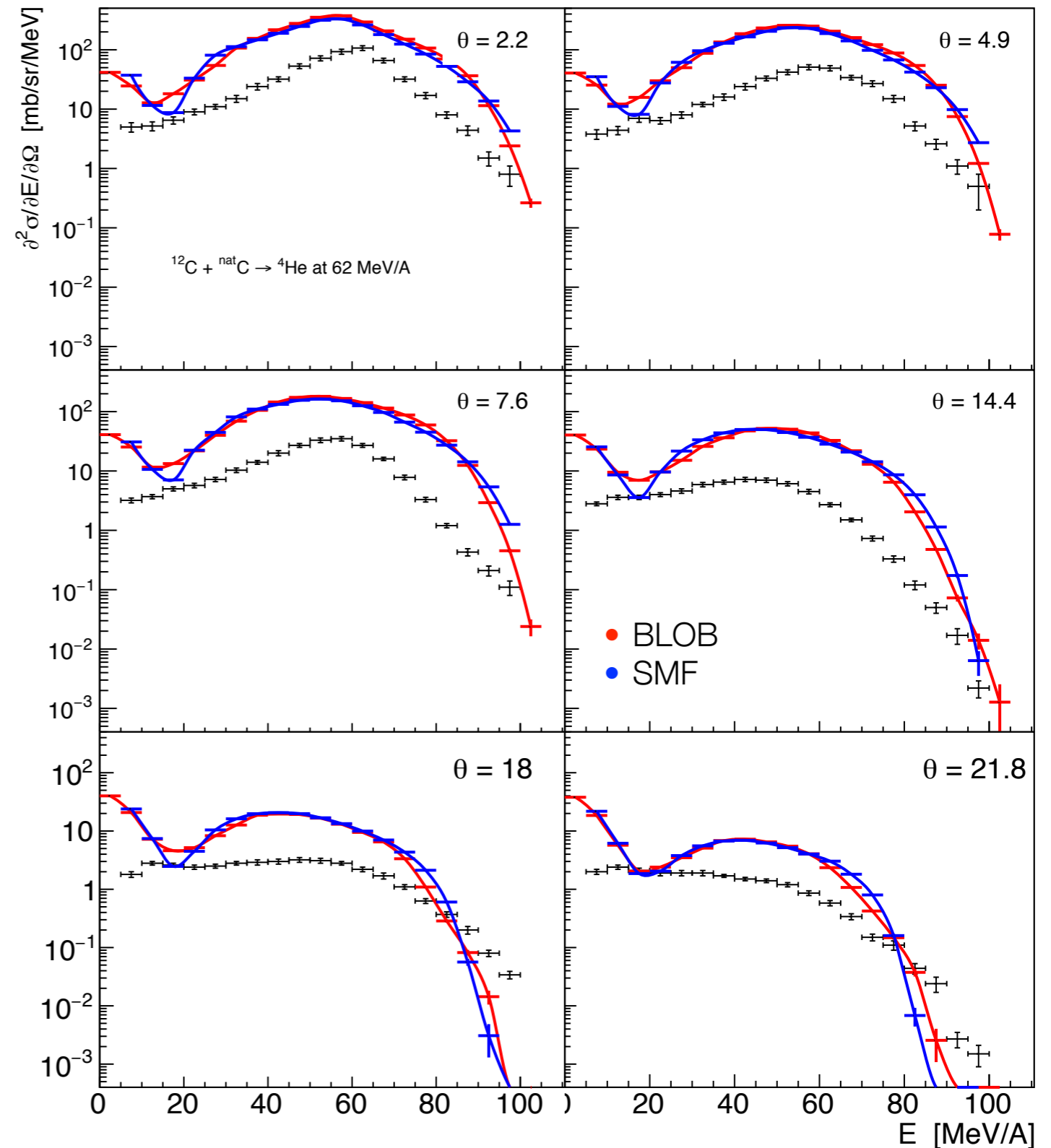
# Interfacing SMF and BLOB to Geant4

- SMF and BLOB had been interfaced with Geant4 and its de-excitation phase
- Similar results between SMF and BLOB



# Interfacing SMF and BLOB to Geant4

- SMF and BLOB had been interfaced with Geant4 and its de-excitation phase
- Similar results between SMF and BLOB



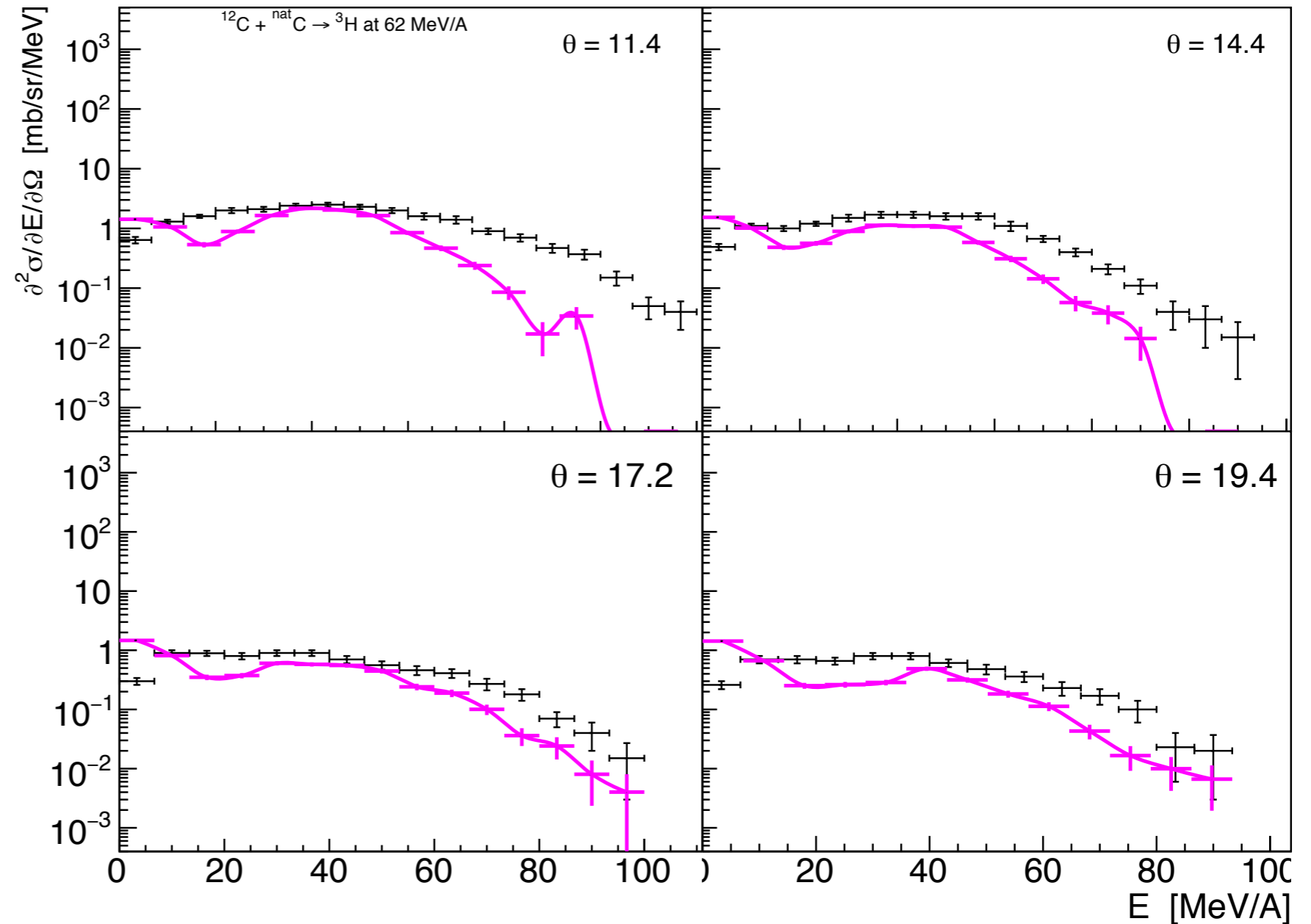
# Coalescence

---

- To insert more than two bodies correlation in an effective way
- Implemented between SMF/BLOB and the de-excitation phase
- Two small fragments are coalesced if  $\Delta x < 6$  fm
- Applied recursively

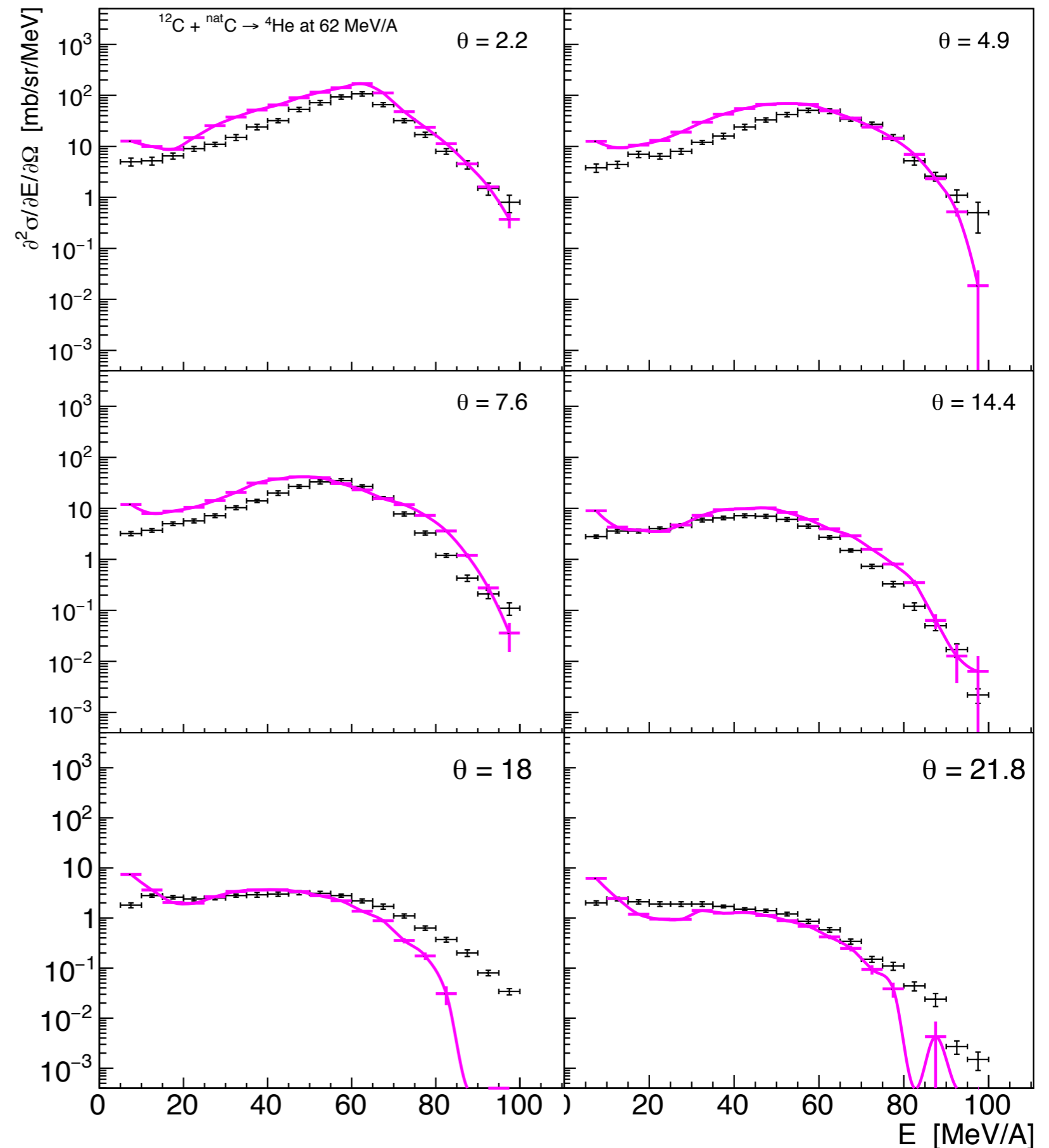
# Preliminary results with BLOB and Geant4

- tritium
- Increasing the test particles number (from 100 per nucleon to 500 per nucleon) the excitation energy is better estimated



# Preliminary results with BLOB and Geant4

- alpha
- It is not possible to increase the number of test particle also in SMF



# Code optimisations

- Optimisation of the function “lapla” without changing the code structure
- 68% speed-up in the function
- 52% speed-up overall

⌵ **Elapsed Time** <sup>?</sup>: **231.966s**

⌵ **CPU Time** <sup>?</sup>: **231.938s**  
Total Thread Count: 1  
Paused Time <sup>?</sup>: 0s

## ⌵ Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time <sup>?</sup>
lapla	run-orig	176.281s
erff	libm.so.6	17.201s
define_two_clouds_rp	run-orig	9.658s
sortrx	run-orig	7.018s
powf	libm.so.6	5.377s
[Others]		16.403s

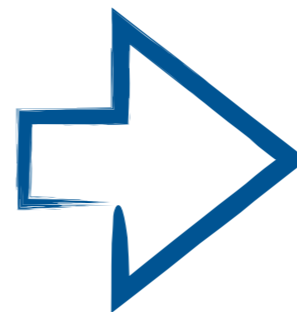
⌵ **Elapsed Time** <sup>?</sup>: **110.235s**

⌵ **CPU Time** <sup>?</sup>: **110.223s**  
Total Thread Count: 1  
Paused Time <sup>?</sup>: 0s

## ⌵ Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time <sup>?</sup>
lapla	run	56.086s
erff	libm.so.6	17.038s
define_two_clouds_rp	run	9.051s
sortrx	run	7.450s
powf	libm.so.6	5.184s
[Others]		15.414s



# Using OpenMP

- Distributing the main loop of the “lapla” on 24 cores
- Small gain overall
- A lot of time spent in distributing data

Elapsed Time <sup>?</sup> **103.925s**  
CPU Time <sup>?</sup>: 669.258s  
Total Thread Count: 24  
Paused Time <sup>?</sup>: 0s

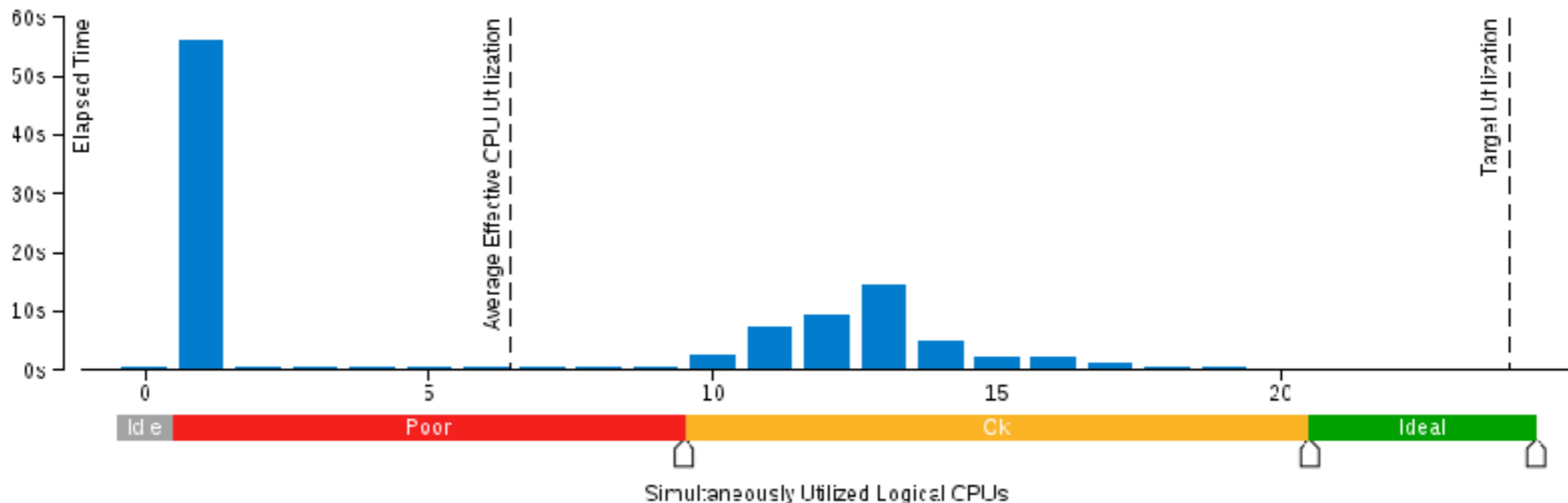
## Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time <sup>?</sup>
func@0x18a90	libgomp.so.1	379.998s
func@0x18bf0	libgomp.so.1	126.738s
lapla_._omp_fn.0	run-omp	44.980s
lapla	run-omp	37.877s
erff	libm.so.6	16.699s
[Others]		62.966s

## Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



# Using Variational Auto Encoder

---

- Despite the optimisation, BLOB is still too slow
- The idea:
  - Bin the PDF output of BLOB
  - Creating a 3D “image”
  - Train a Variational Auto Encoder to reproduce such “images”
  - Condition the VAE to impact parameter

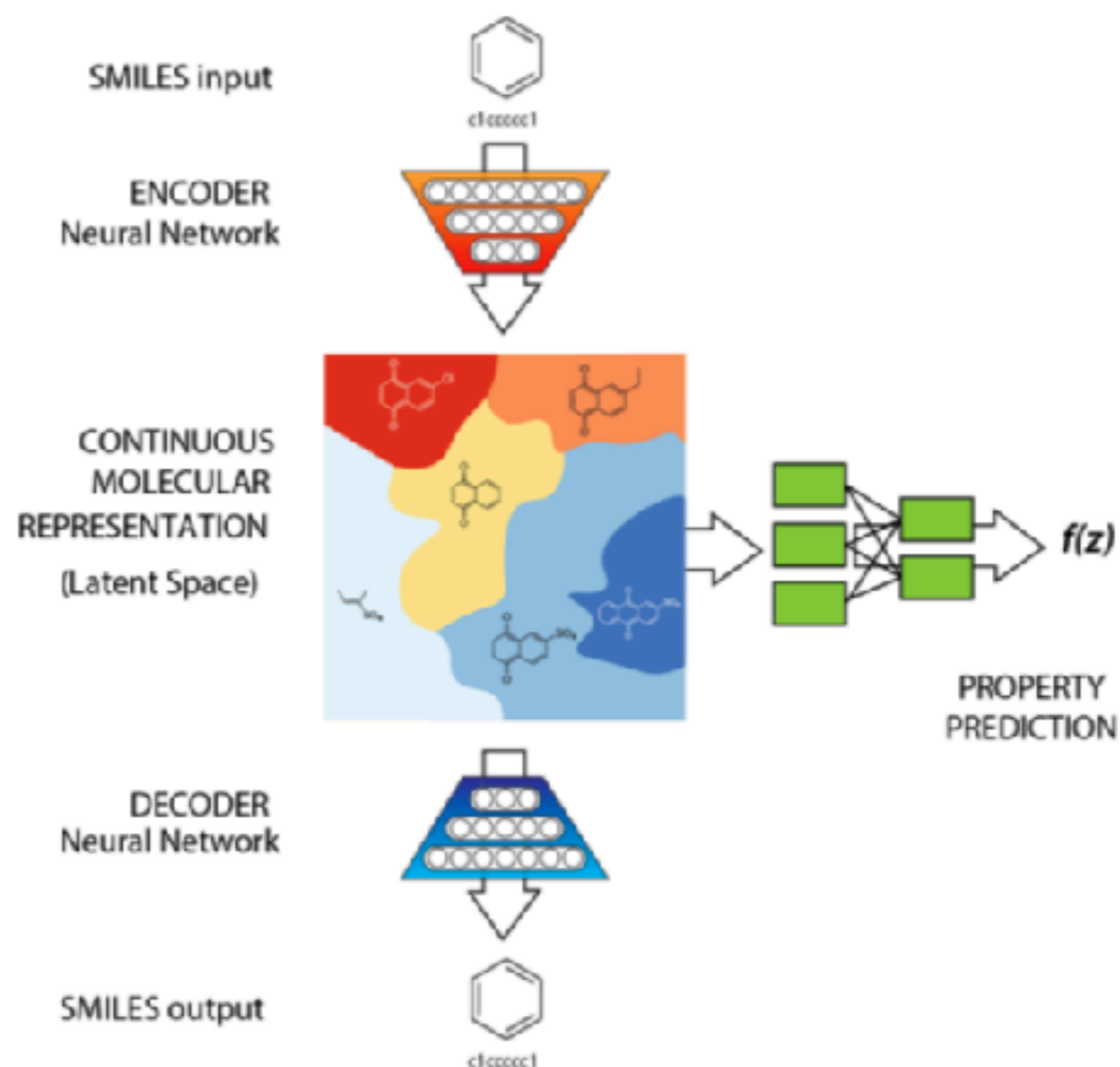


# Conditioning to $b$

- Taking inspiration from:

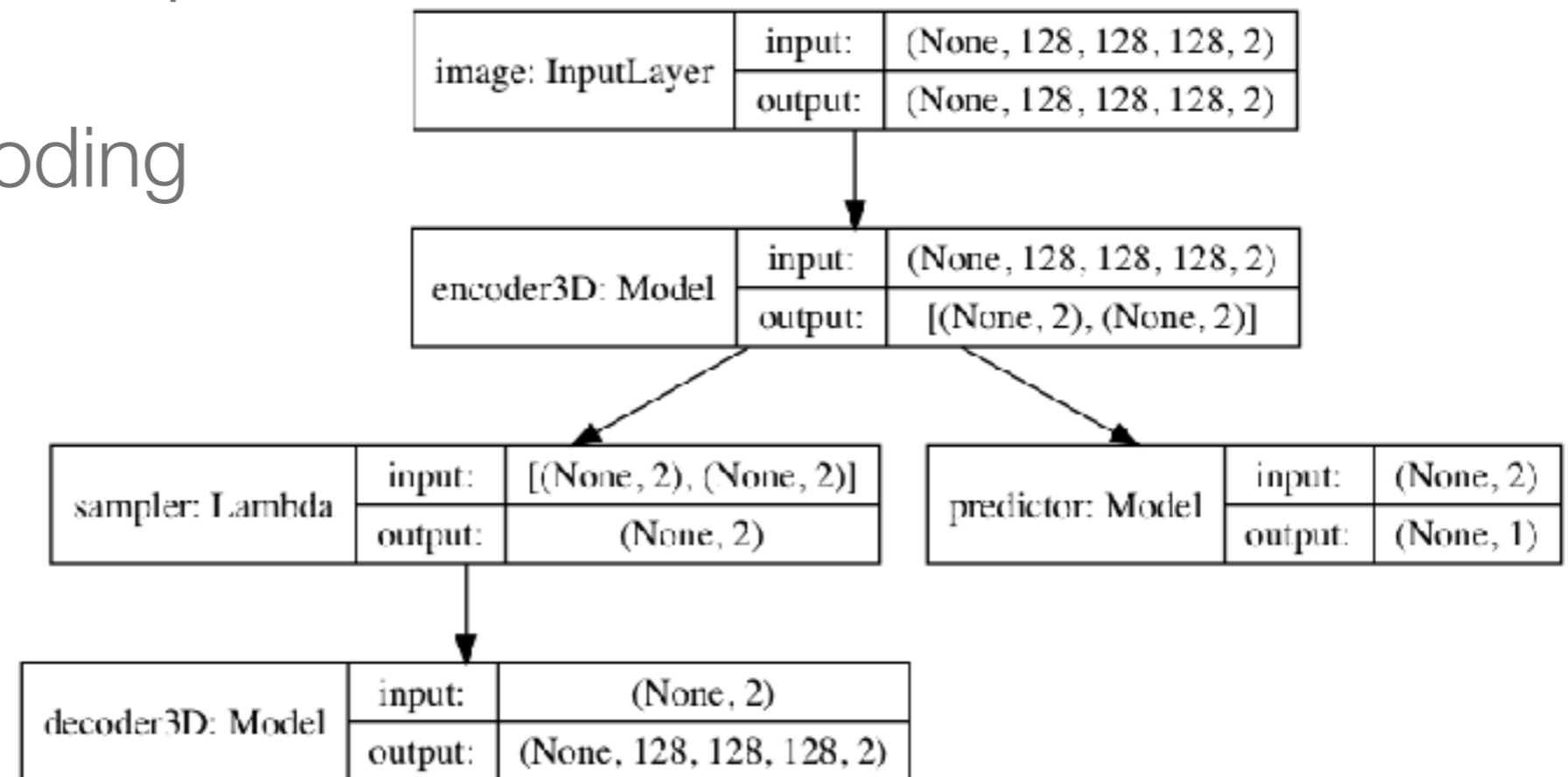
[Automatic chemical design using a data-driven continuous representation of molecules, Gómez-Bombarelli et al. arXiv:1610.02415]

- VAE for generating new chemical compounds with properties that are of interest for drug discovery
- To organize latent space w.r.t chemical properties they jointly trained the VAE with a predictor
- It predicts these properties from latent space representations



# Conditional VAE

- Convolutional 3D encoding
- Conditioned latent space
- Symmetric decoding



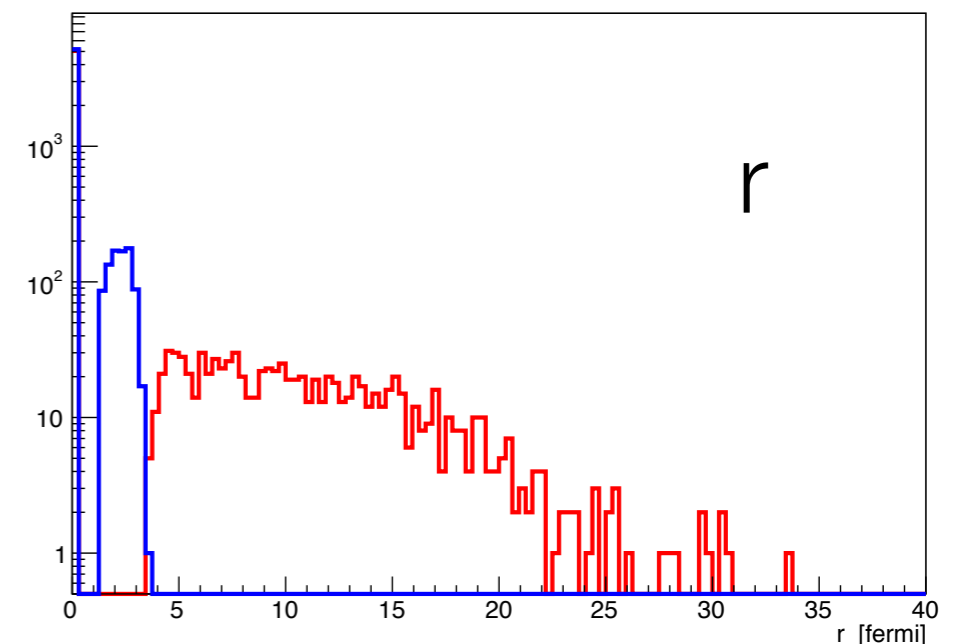
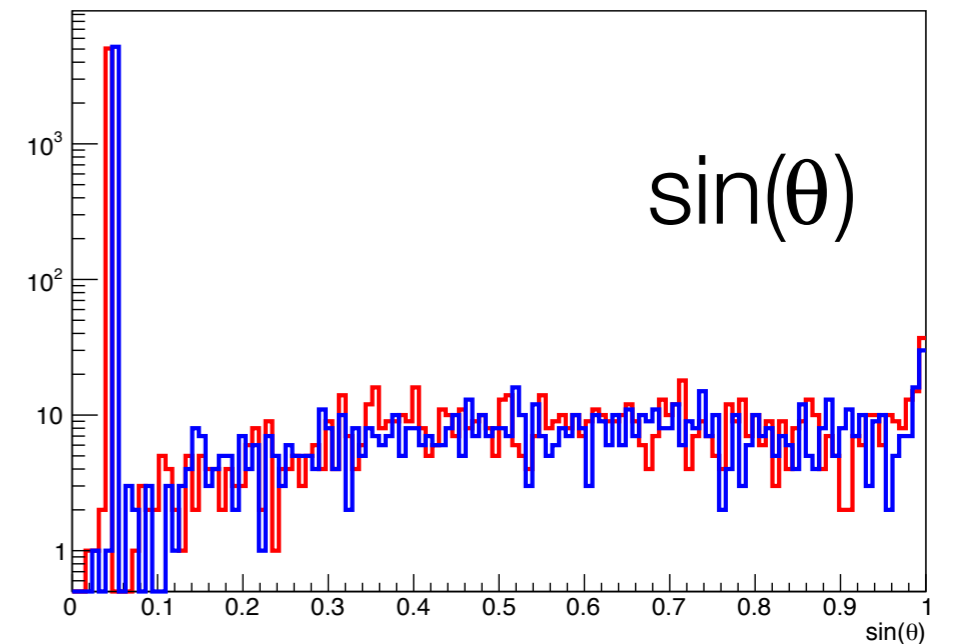
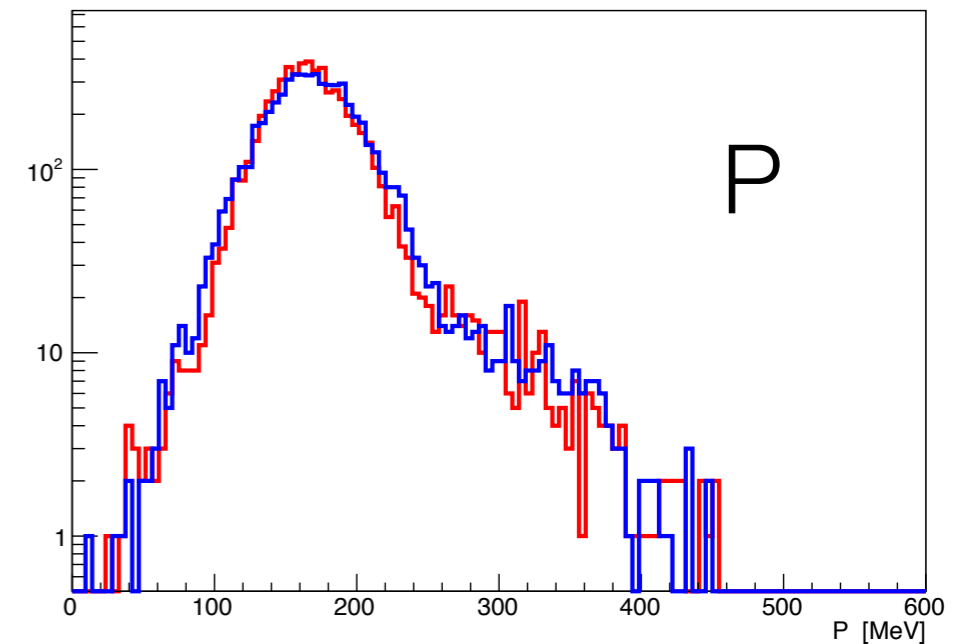
# Training dataset

---

- The BLOB final state is a list with the position in the phase space of fragments and gas particles
- Fragments: A and Z (real), P, Q and Excitation energy
- Gas particles: Z, P and Q. Each represent a 1/500 probability of having a nucleon in that position of phase space
- 1 000 events
- Generated with linear impact parameter
- 90% for training and 10% of them for test

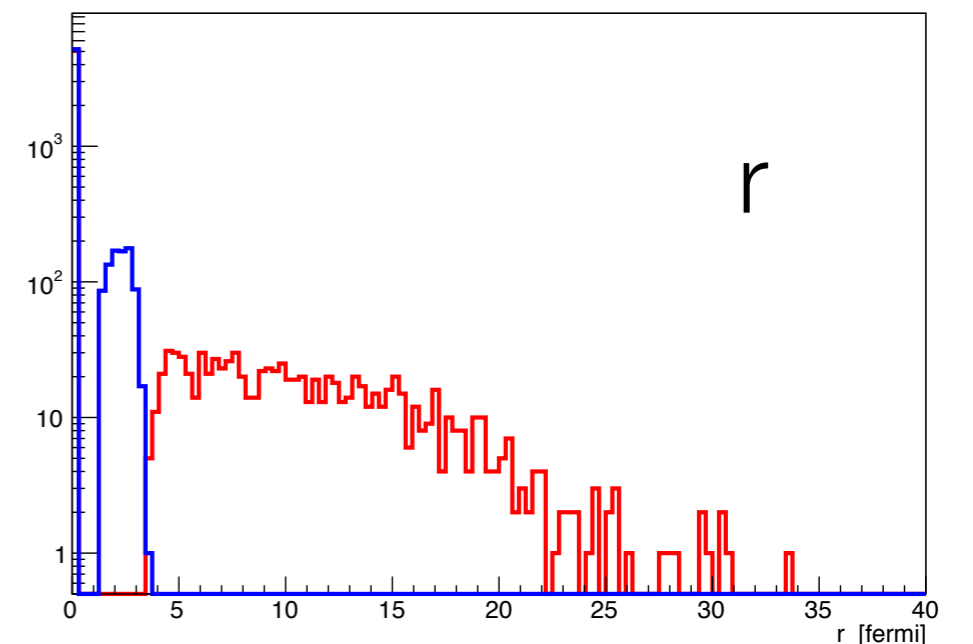
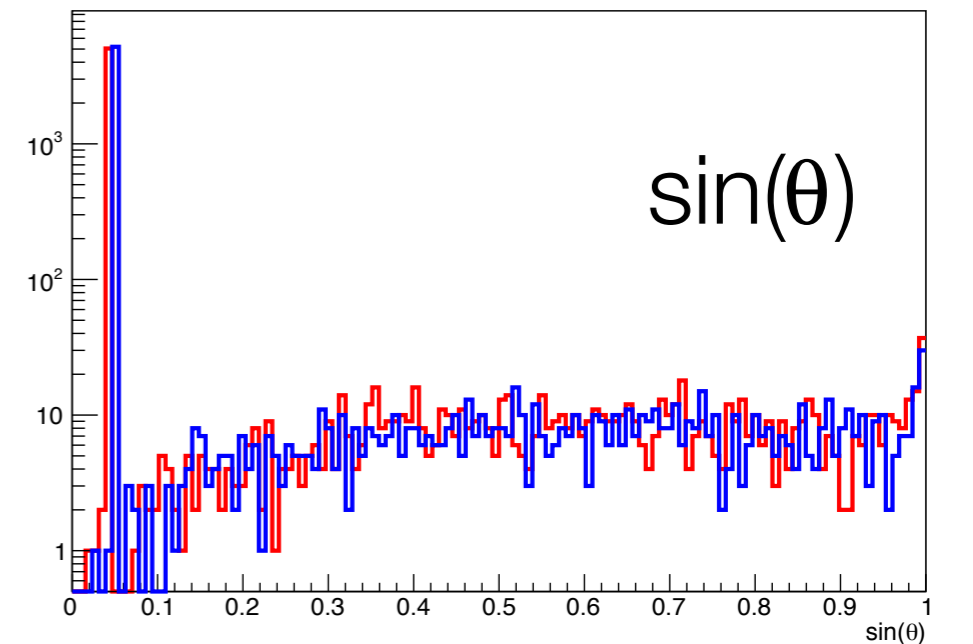
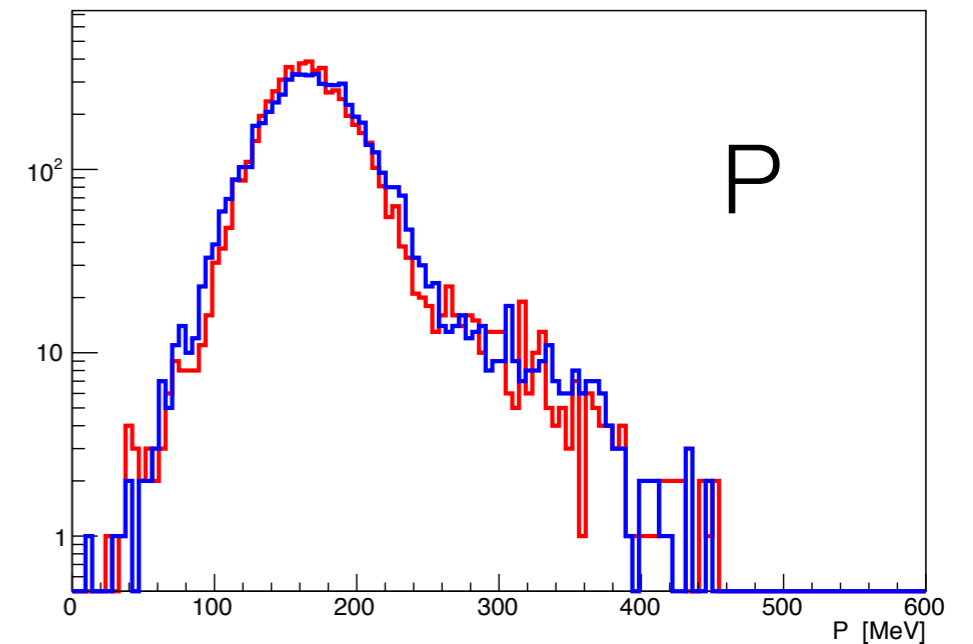
# Reducing dimensionality

- Only events with 2 fragments are considered
- We divided the test particles in two samples:
  - Projectile like (red)
  - Target like (blue)
- $\sin(\theta)$  instead of  $\theta$  to:
  - have same sign
  - enhance small angles



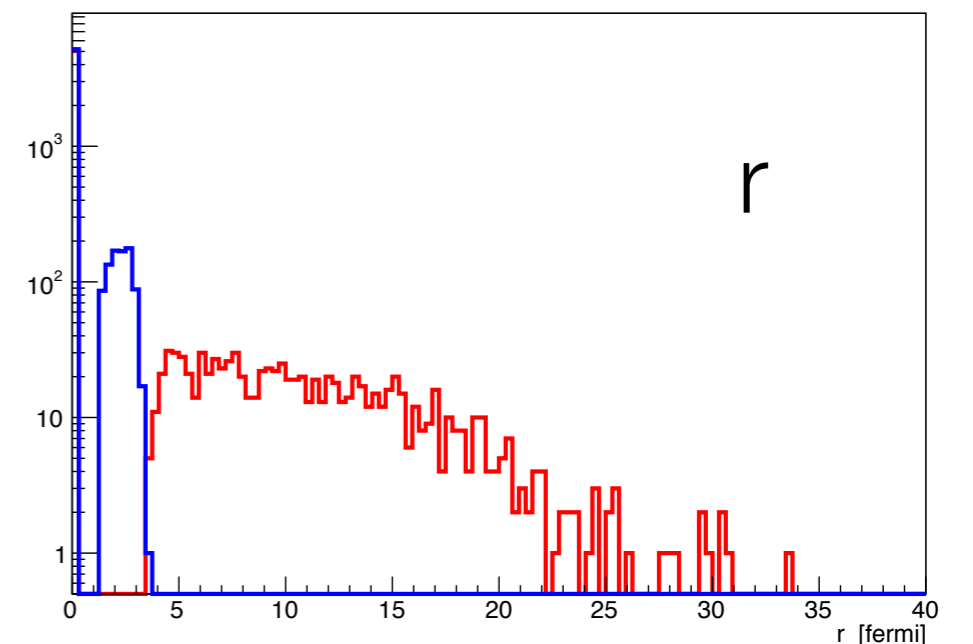
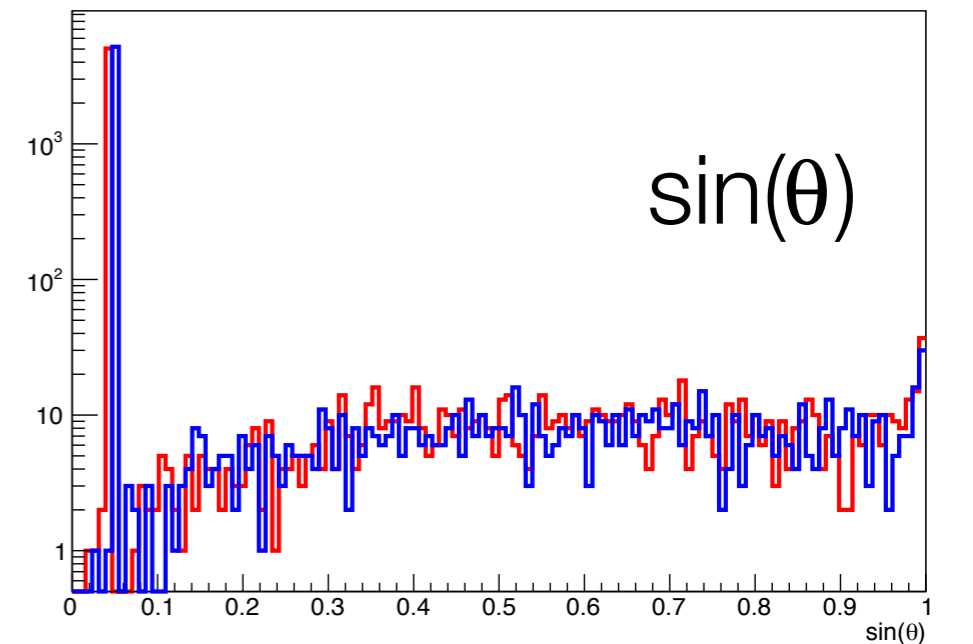
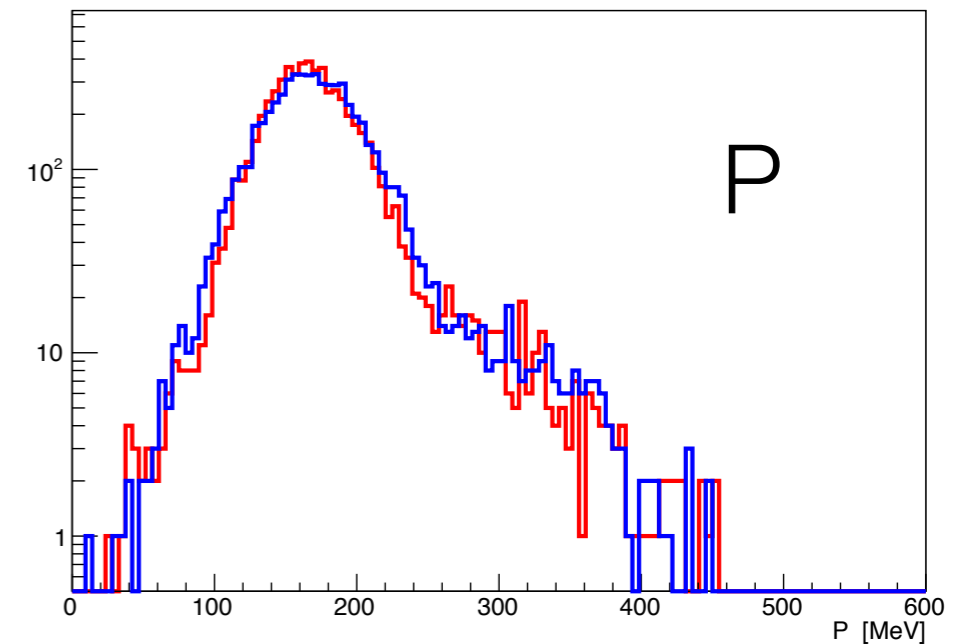
# Reducing dimensionality

- To reduce the dimensionality and use the keras 3D kernels
- We consider only:
  - The modulus of the momentum
  - its angle with the collision axis
  - The distance of each test particle with the fragment center



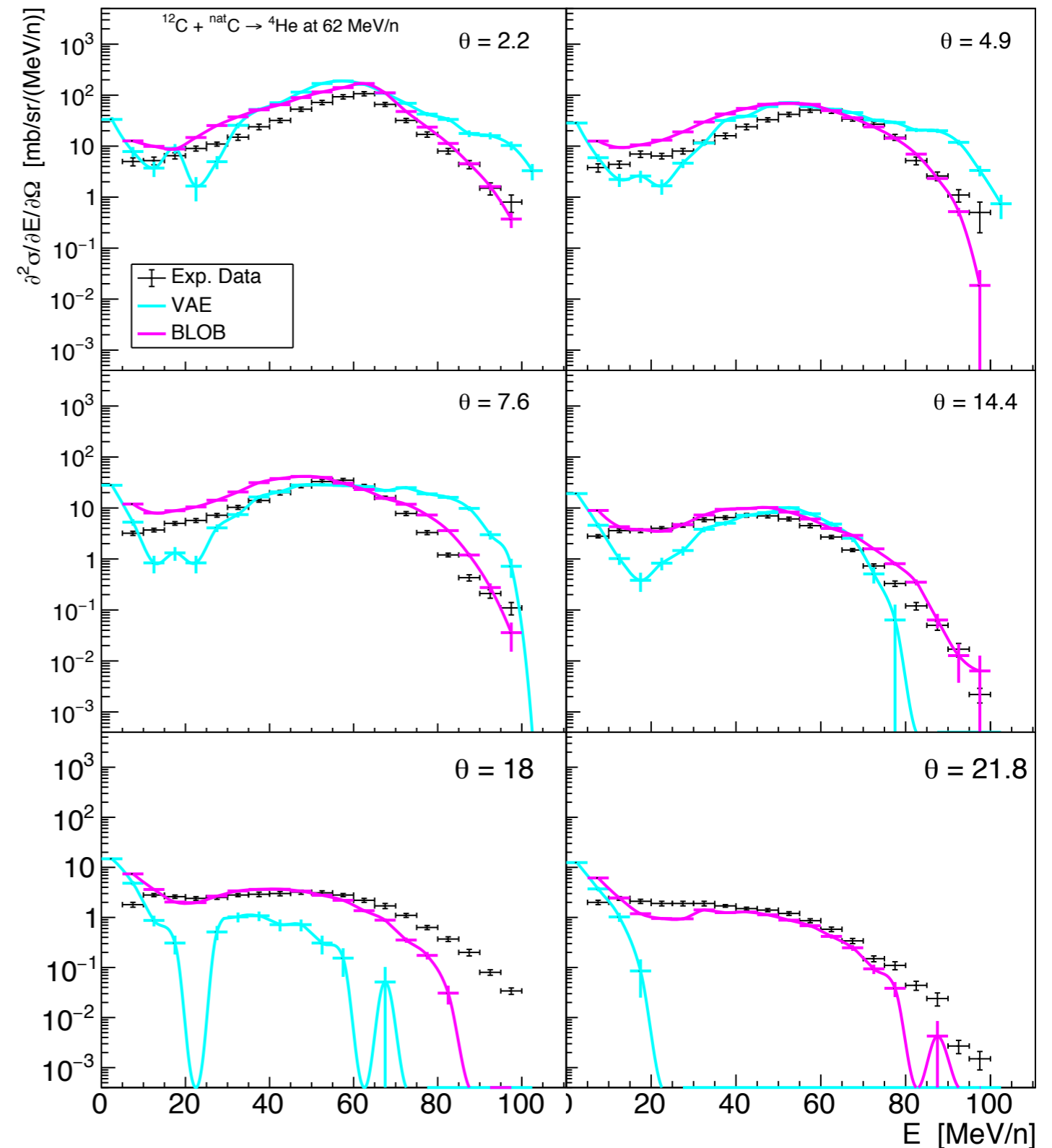
# Reducing dimensionality

- Fragments are represented by  $500 \cdot A$  particles
- $P$  is sampled with gaussian distribution:
  - mean =  $P_{frag}$
  - sigma = Excitation energy
- All with the same  $\theta$
- $r = 0$



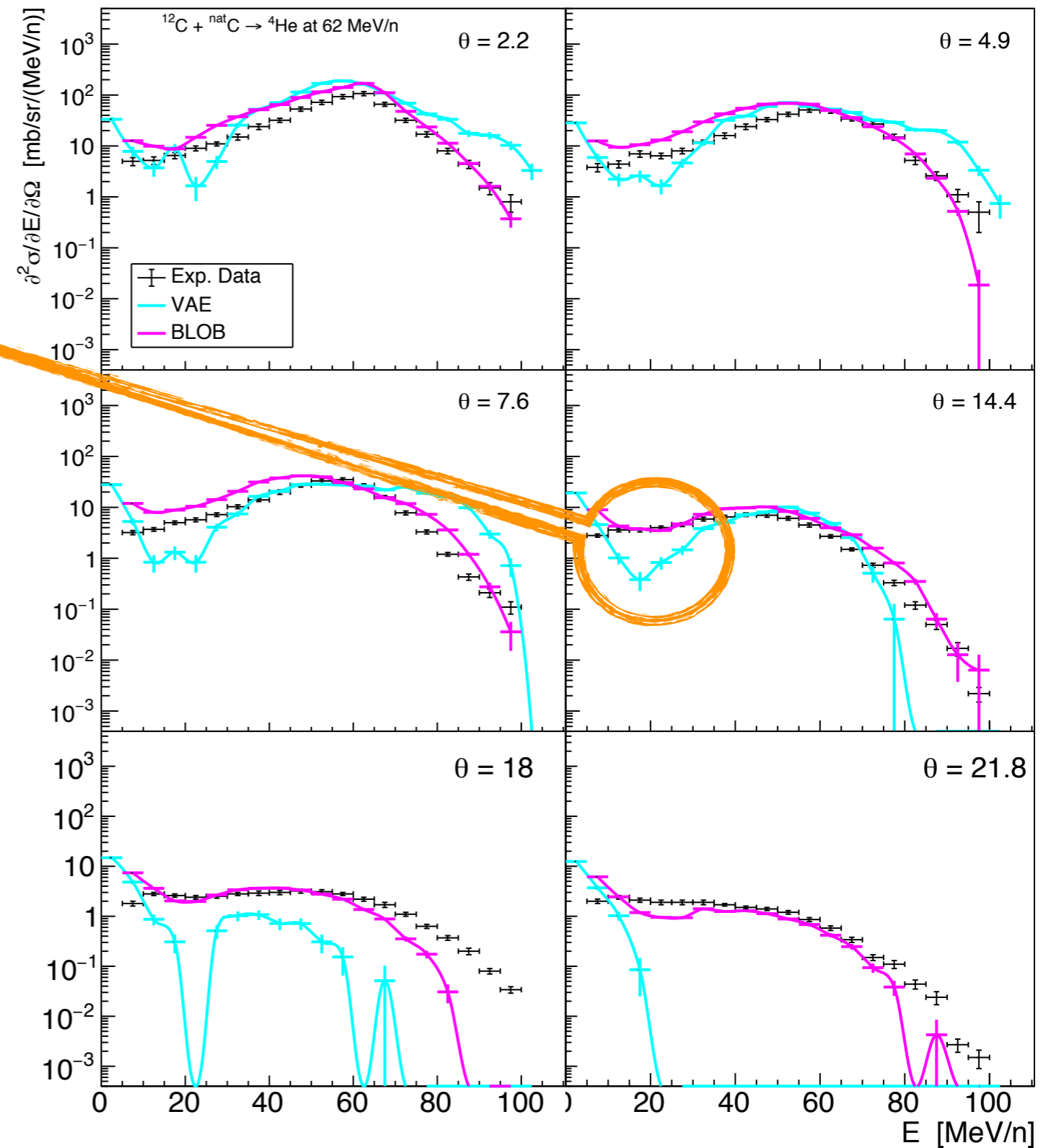
# Testing reconstruction

- Fragments are identified selecting  $r < 1$  fermi
- Momentum = average
- Excitation energy = variance
- $\theta$  = average



# Testing reconstruction

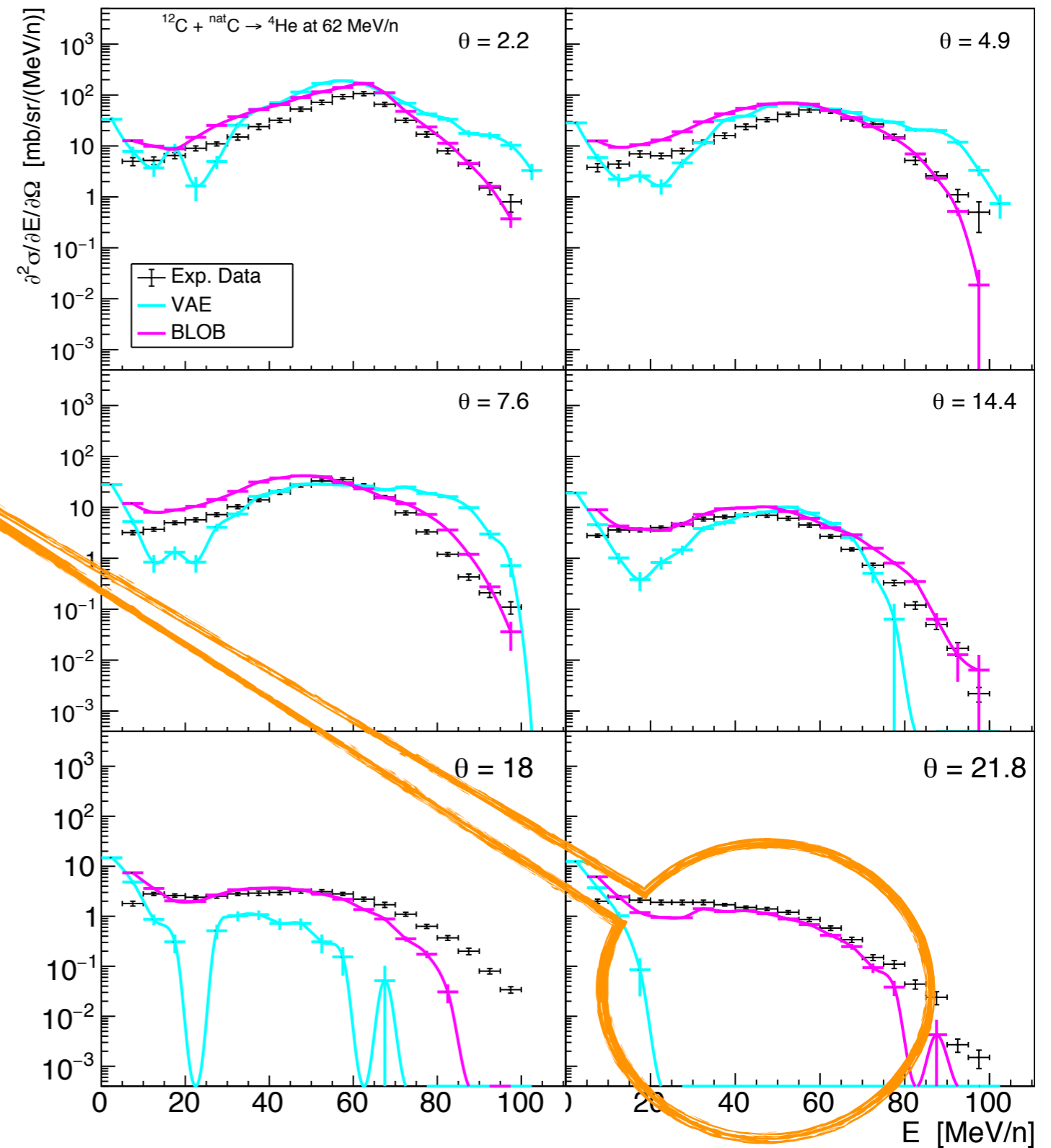
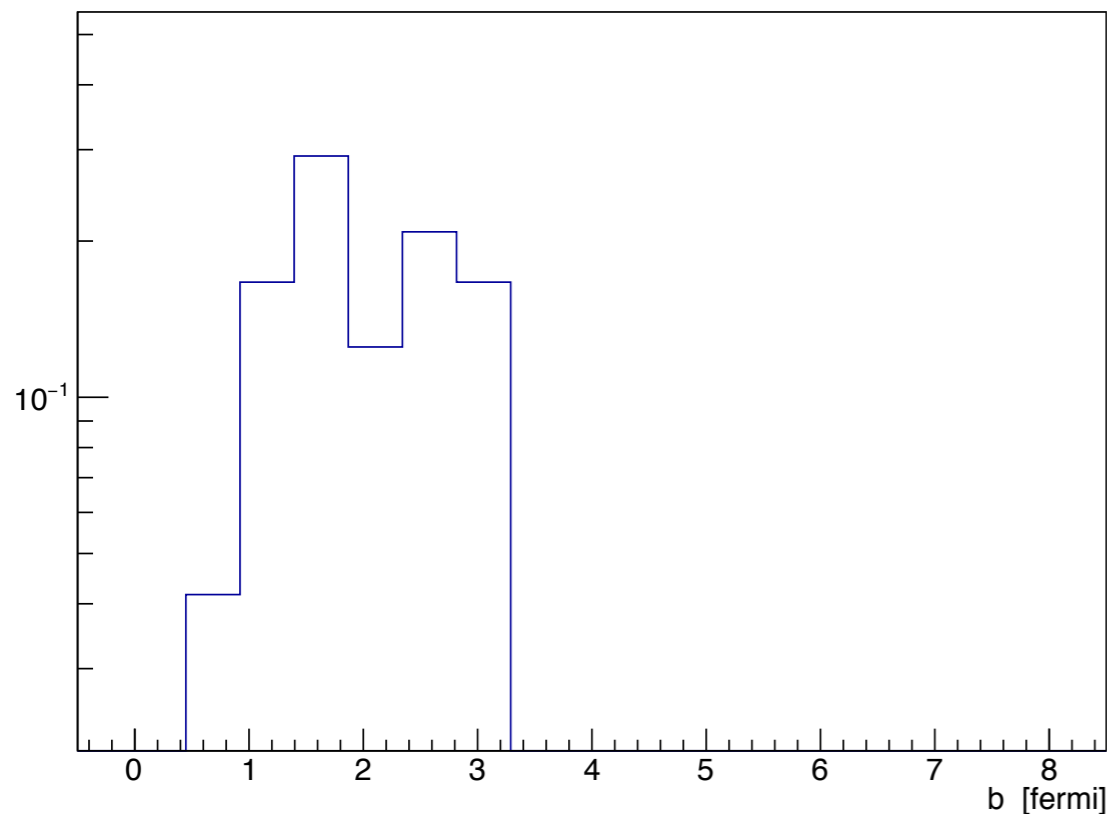
- Lack of particles at mid rapidity
- Underestimation of neck events
- Because of coalescence not active





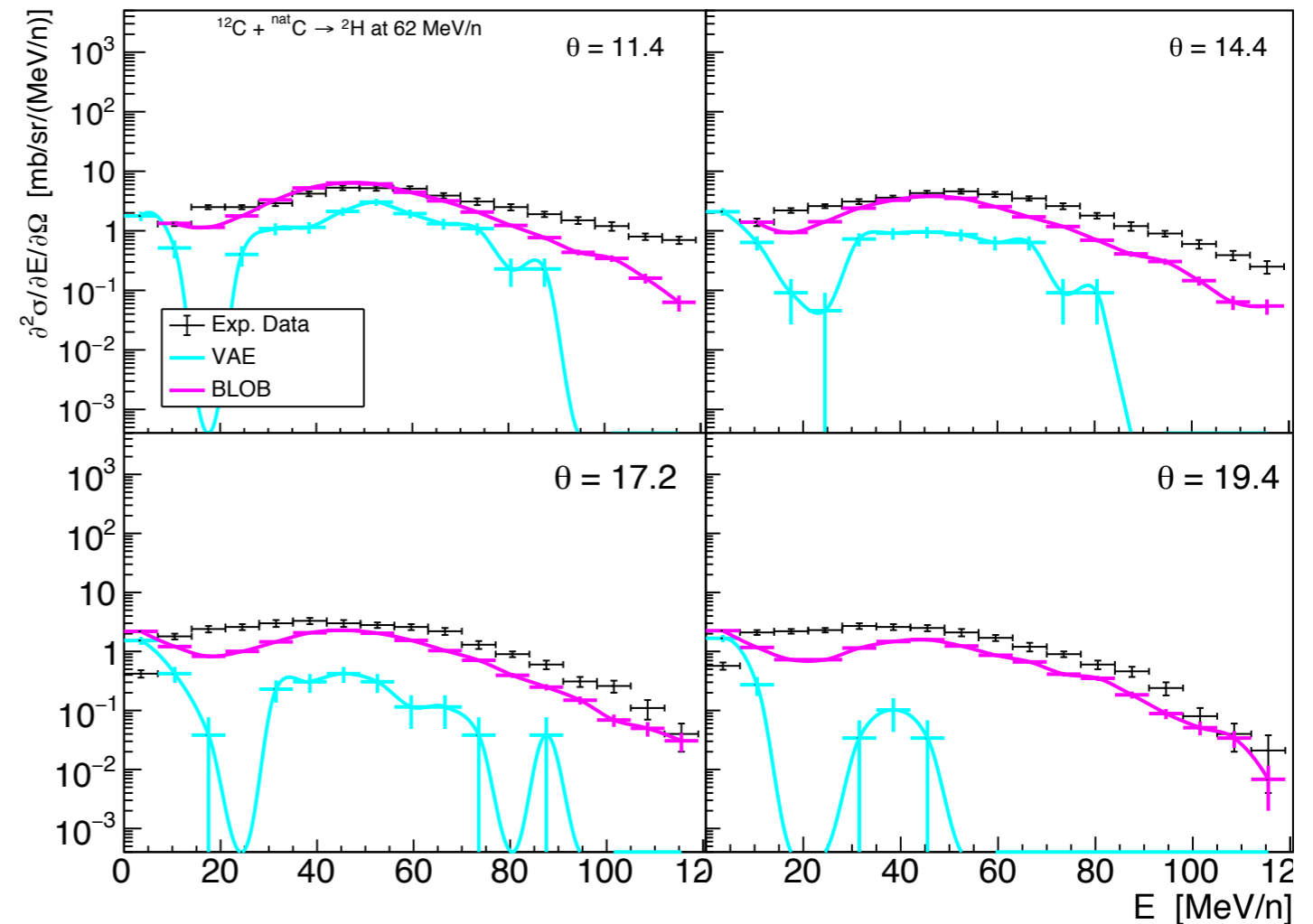
# Testing reconstruction

- Lack of particles at large angle
- Because of the request of 2 fragments



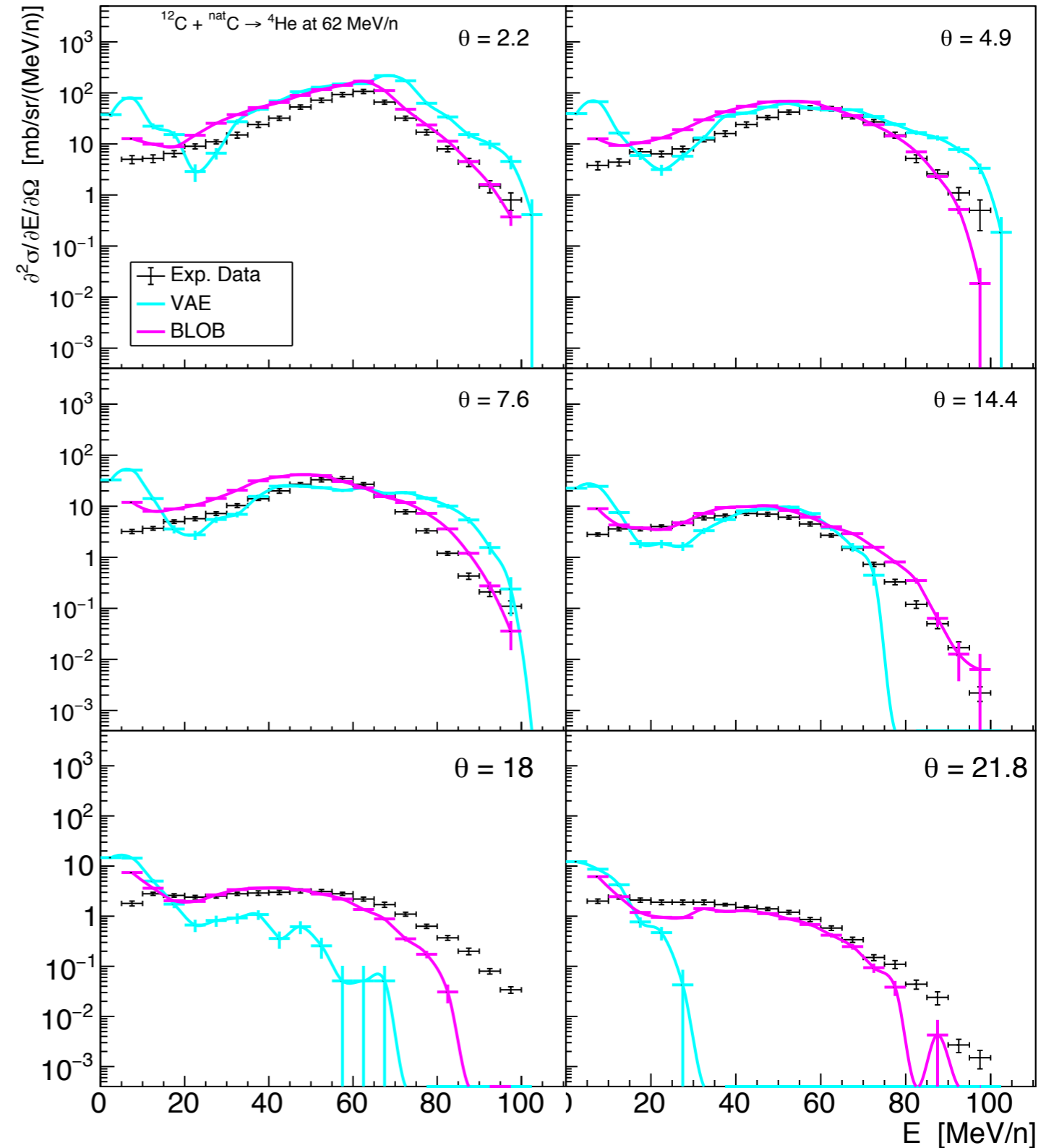
# Testing reconstruction

- Same on deuterium
- Next step will be add a 3rd channel for neck particles
- And clustering algorithm



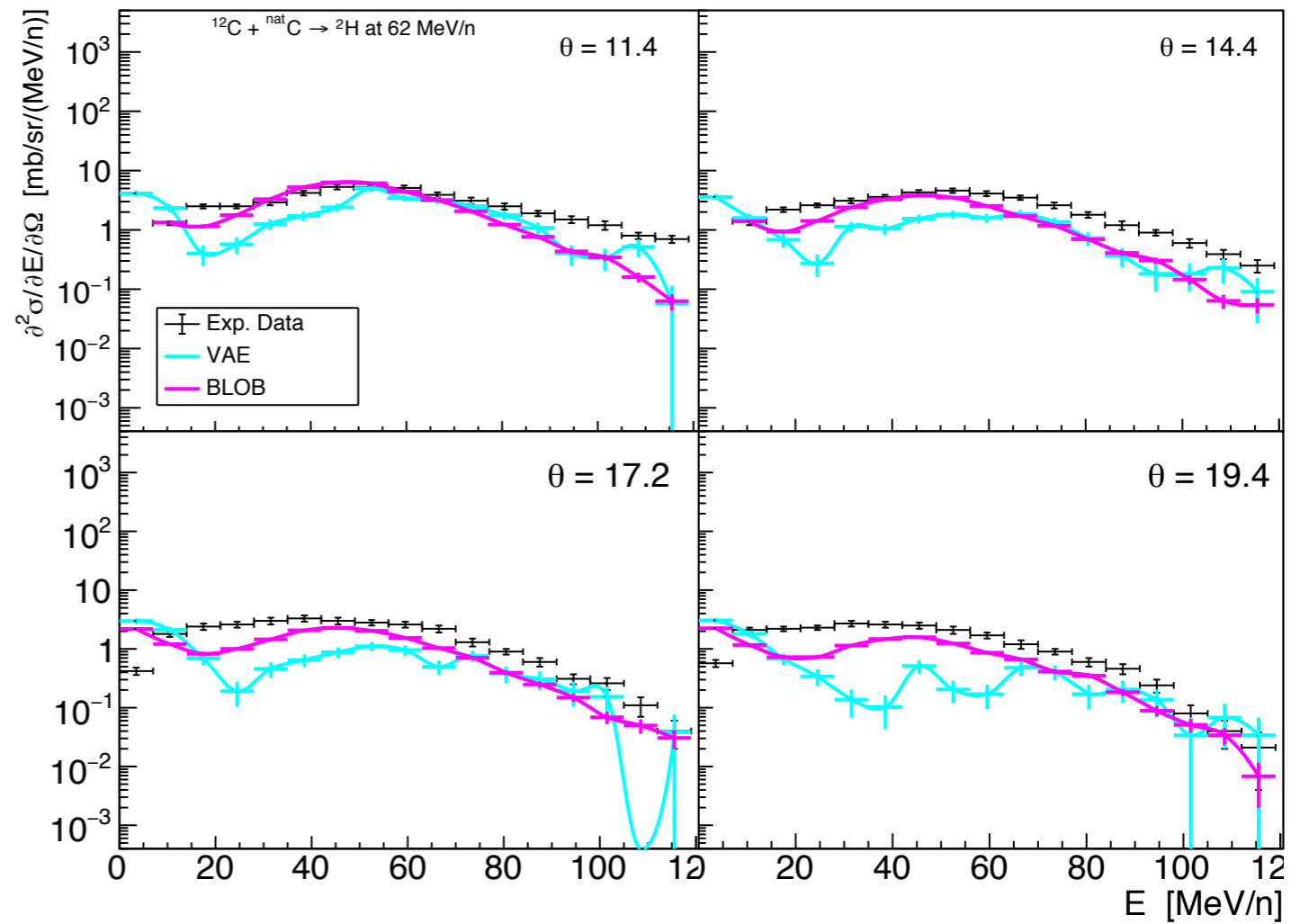
# With coalescence

- Using the coalescence
- The mid rapidity lack is mitigated



# With coalescence

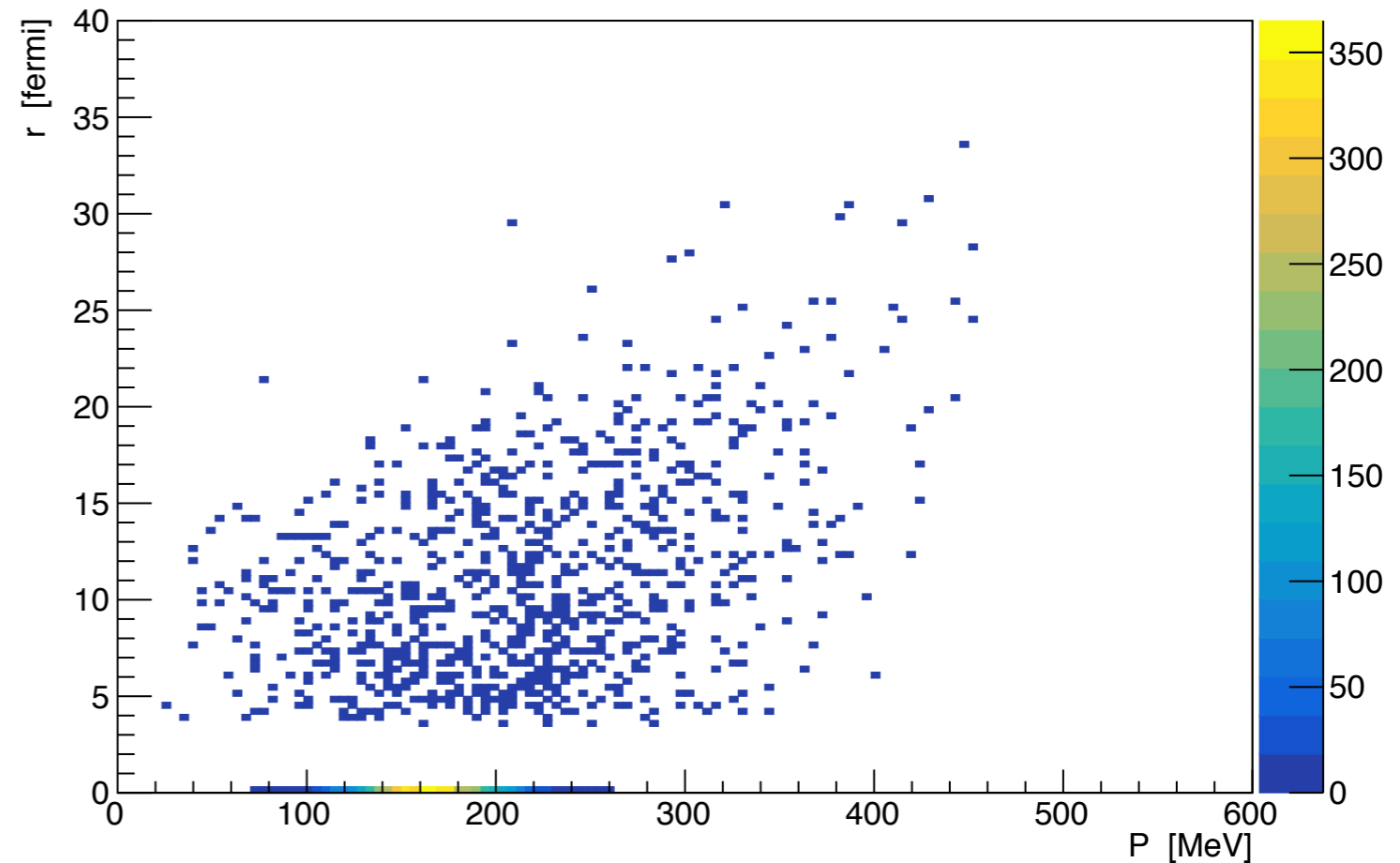
- Also for deuterium



# Challenges

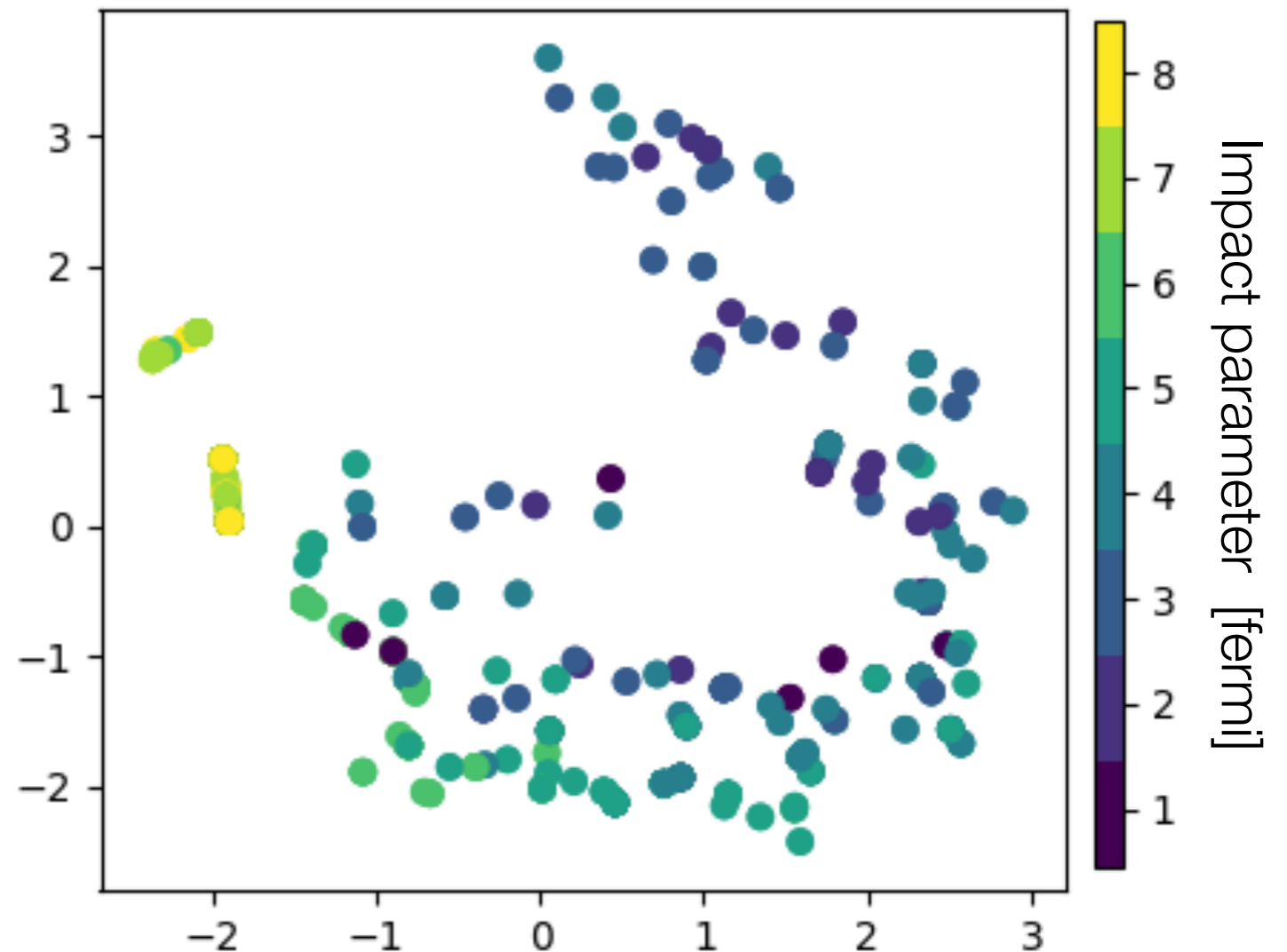
---

- Sparse
- Large input ( $128^3$  numbers)
- Small dataset (for the moment)
- Impact parameter distribution non uniform (for the moment)



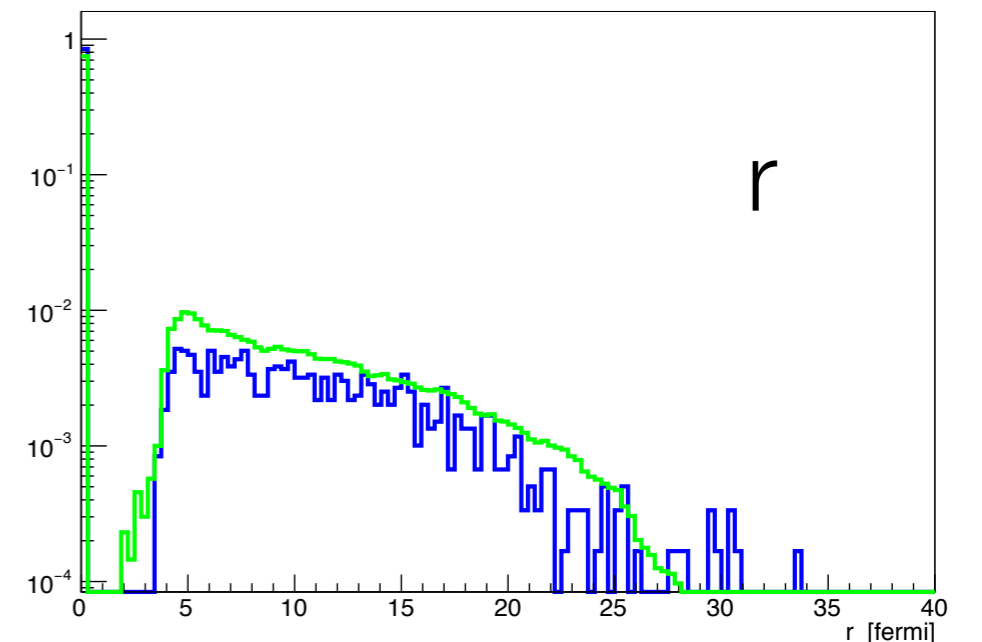
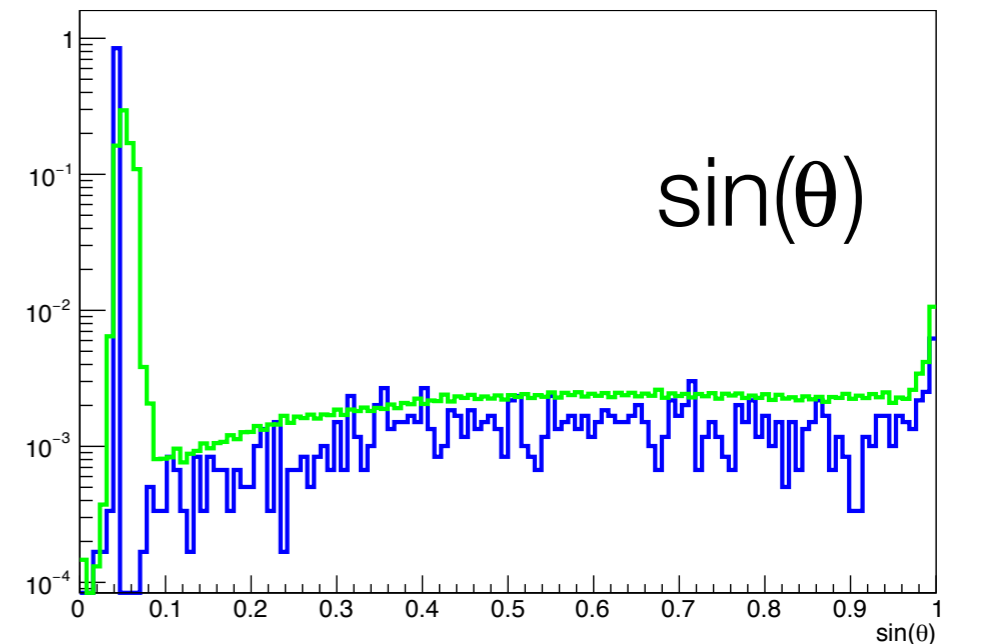
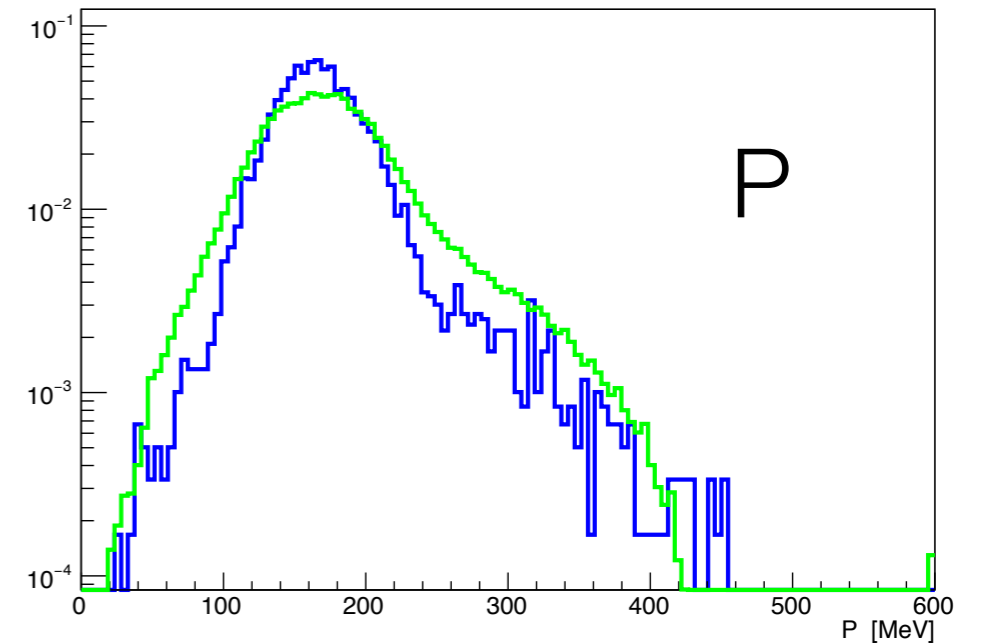
# Latent space

- 40 epochs of training
- Events with similar impact parameters are close in latent space
- Especially the events with very large impact parameters



# Output distributions

- The generated distributions (green) looks similar to the input (blue)
- The generated event has been generated sampling two gaussian in latent space with:
  - means = position of the input
  - sigmas = 0.1



# Next steps

---

- Generate dedicated input from BLOB
  - At least two order of magnitudes more events
  - Uniform  $b$
- Clustering test particles in 3 groups (Projectile like, Target like and neck)
  - I.e.: 3 channels images
- Train a classifier to identify  $b$  from 3D distributions



# Next steps

---

- Condition the VAE to two parameters:  $b$  and primary  $E$
- Train it with different energies
- Couple the generated model with Geant4 (in C++)
- Benchmark with data at different energies

