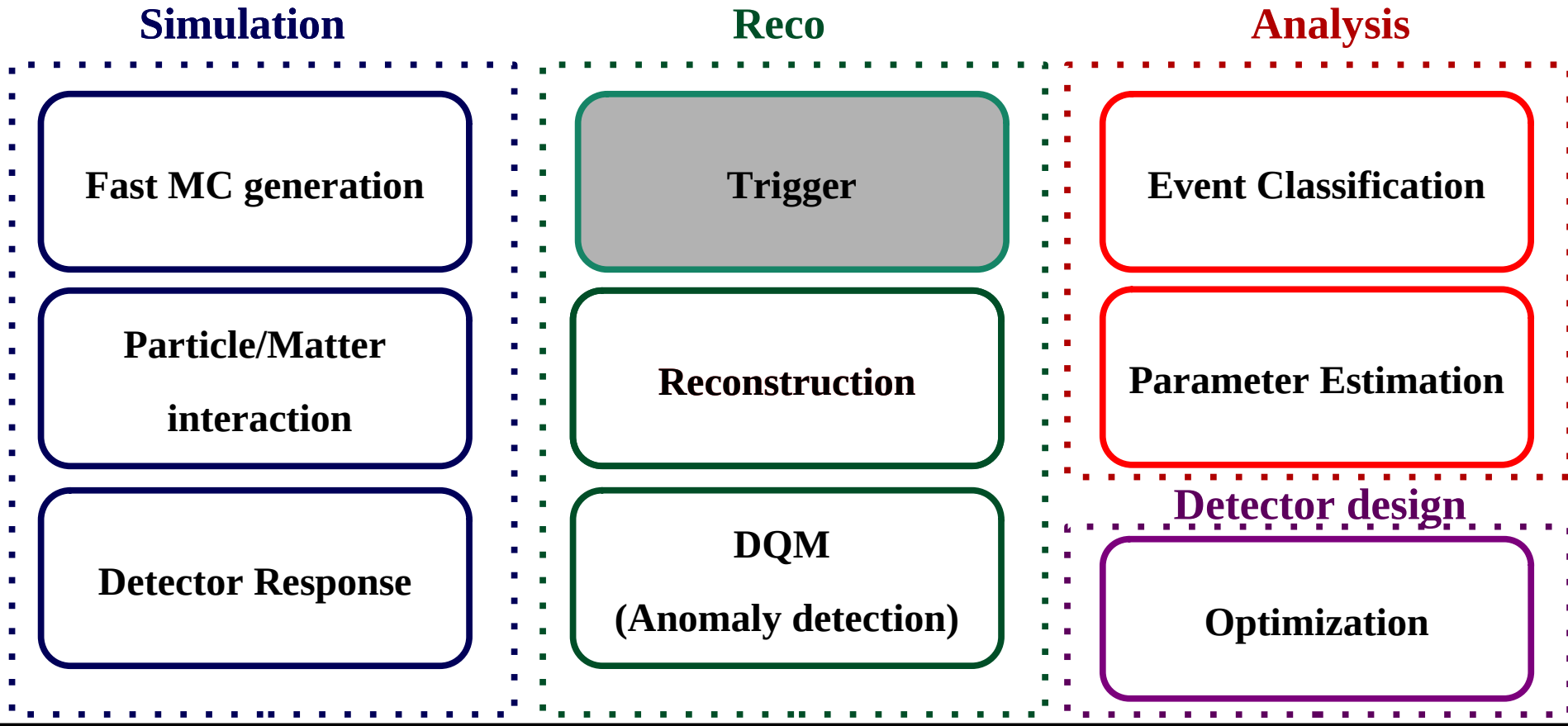
IFCA activities
COMCHA

CPAN meeting, 19-21 Nov, 2024

Pablo Martínez Ruiz del Árbol

Use of new computing technologies at IFCA

- Touching/Testing/Improving in almost all places where new technologies have an impact
- Close collaboration with IFCA Advanced Computing Group on many projects
- Also part of MODE (Machine-Learning Optimized Design of Experiments) collaboration
- In this talk showing just a bunch of developments of the last 2 years

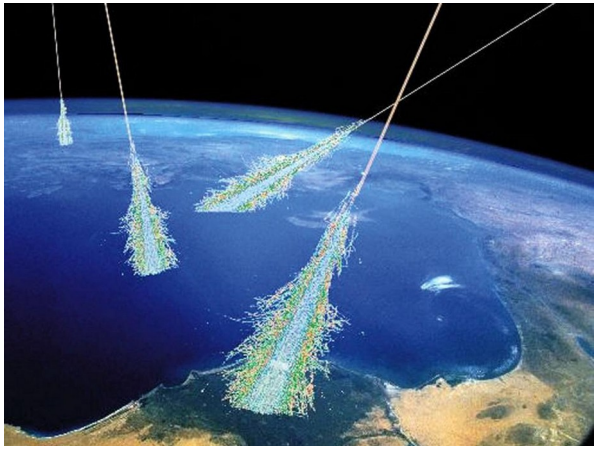


Fast simulation

Simulation for scattering muography

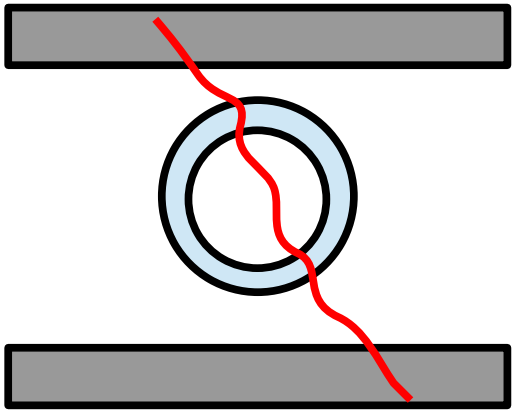
- Simulation for scattering muography has three different components

Muon flux generation



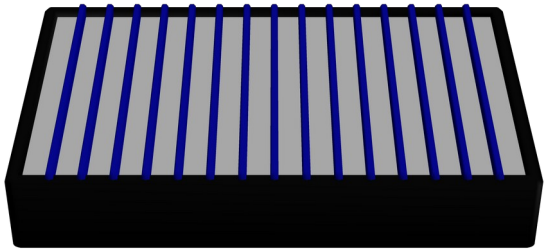
- Most generators parametrize muon flux as a function of altitude/latitude etc
- This part is usually pretty fast
- CRY is a good example

Muon propagation through matter



- Implementation of energy loss and multiple scattering at least
- Can be very time consuming specially for complex geometries
- GEANT4 very precise on this

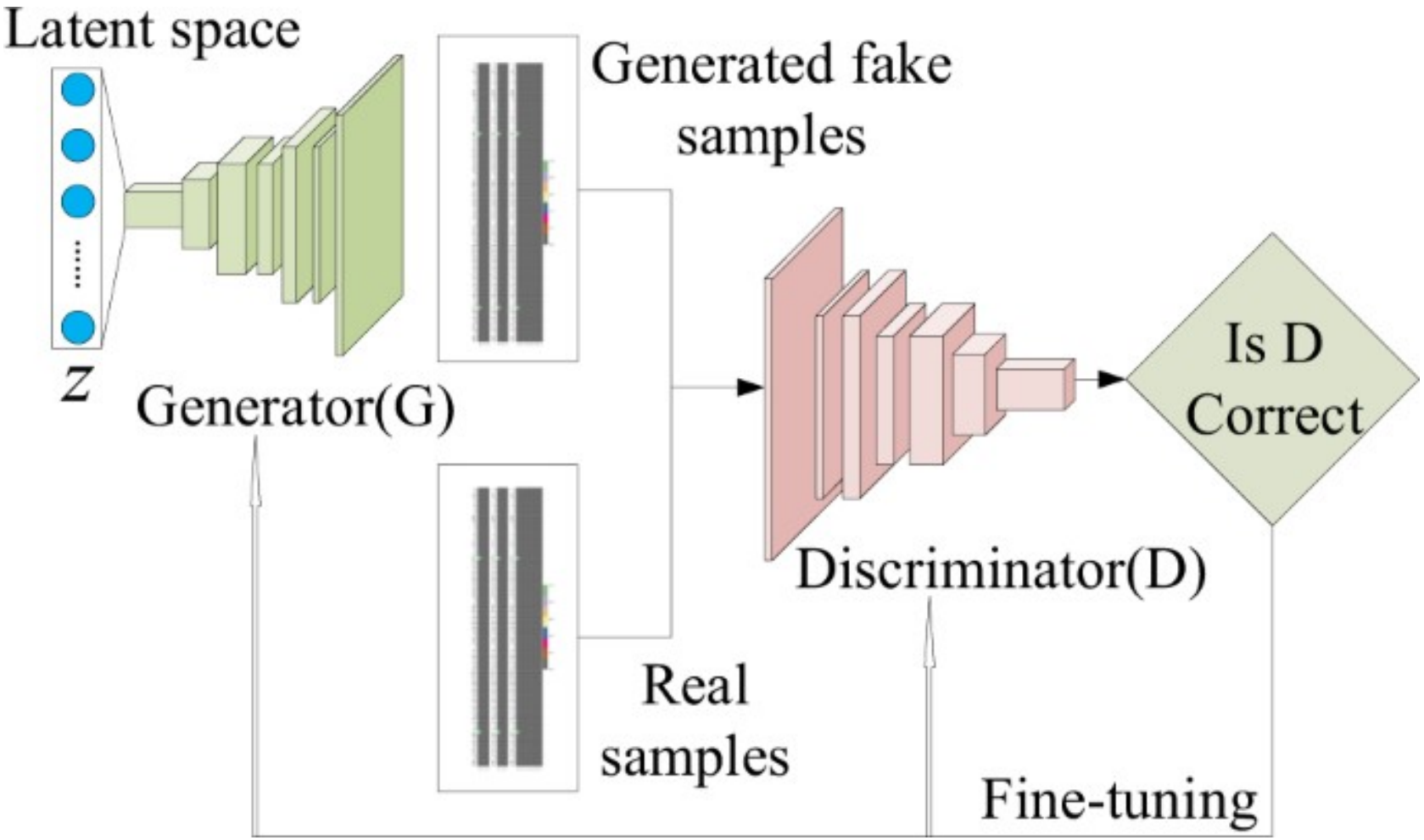
Detector simulation



- A model of the detector response has to be considered for precise MC simulation
- This part can be critical and it is typically difficult to implement
- No general recipes every detector needs its own model

Generative Adversarial Neural Networks

- We propose to use Generative Adversarial Neural Networks to reproduce muon propagation



Our Muography setup

- Our GAN simulation is running for a muography setup of large interest in the industry
  - Estimation of the corrosion level of a the inner walls of a pipe
- Simulator target: predict lower segment having the upper segment as input
  - This means that we rely on CRY for the simulation of the upper segment
  - All tests performed on MC samples where detectors are assumed to be perfect

Upper detector

x_1, y_1, $v_{x1} = \text{atan}(\theta_{x1})$, $v_{1y} = \text{atan}(\theta_{y1})$

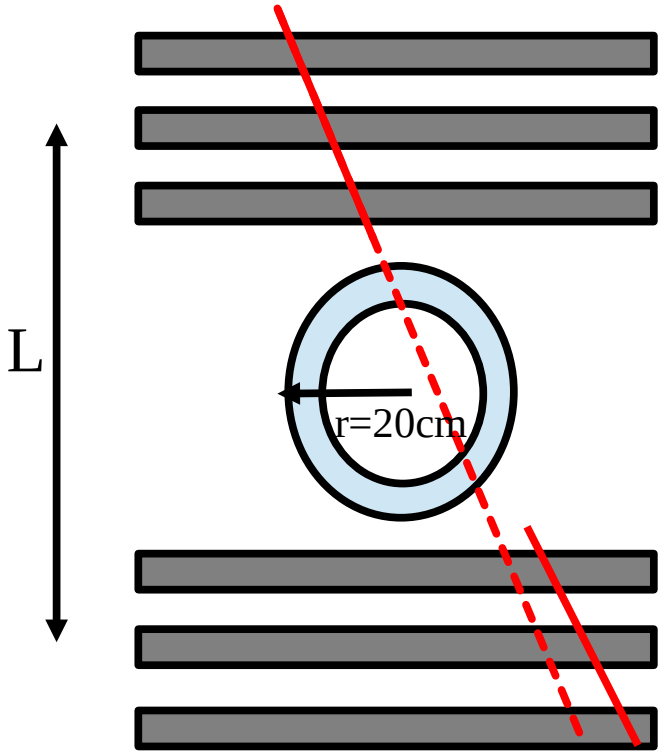
Lower detector

x_2, y_2, $v_{x2} = \text{atan}(\theta_{x2})$, $v_{2y} = \text{atan}(\theta_{y2})$

Target variables

$\Delta x = x_2 - L\, v_{x1} - x_1$ $\quad$ $\Delta y = y_2 - L v_{y1} - y_1$

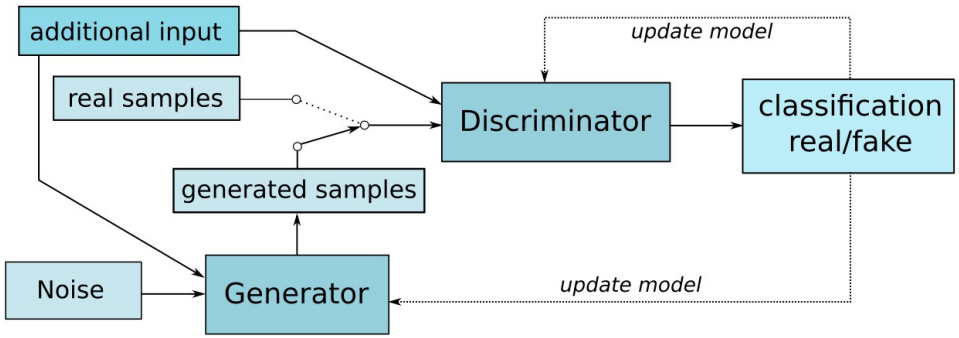
$\Delta v_x = v_{x2} - v_{x1}$ $\quad$ $\Delta v_y = v_{y2} - v_{y1}$



Wasserstein conditional GAN

- We devise a Wassersteing GAN with the following characteristics
  - Keras + TensorFlow
  - The variables of the segment in the first detector are given as input to the generator
  - **The thickness of the pipe to be generated is also provided as input**
  - Critic + Loss function → more stability
  - Architecture: 32, 64, 128 LeakyReLU
  - Latent space dimension: 64
  - Optimizer: Adam, 0.001 (halves every 50 epochs)
  - Trained for 1000 epochs (Total training time ~ 2-3 hours, GeForce RTX 3090)

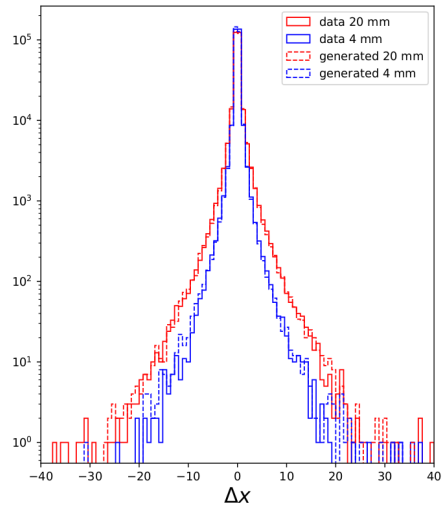
DOI:10.31526/JAIS.2024.487

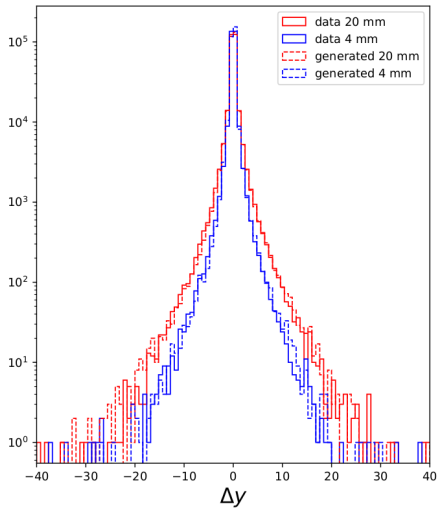


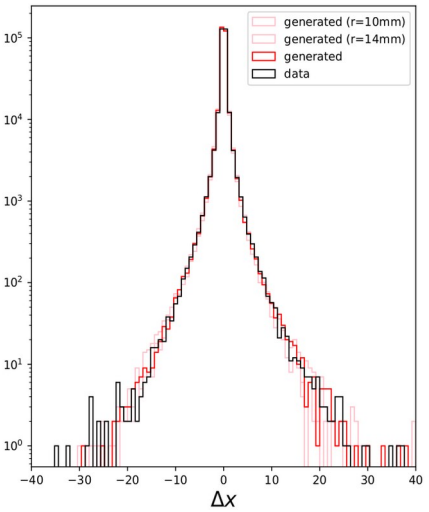
Simple GAN		
Pipe thickness (mm)	Number of training samples	Number of evaluation samples
16	306707	307352
Conditional GAN		
Pipe thickness (mm)	Number of training samples	Number of evaluation samples
4	619605	300000
6	618798	300000
8	617951	300000
10	616700	300000
14	614944	300000
16	615216	300000
18	614109	300000
20	613692	300000
12*	-	300000

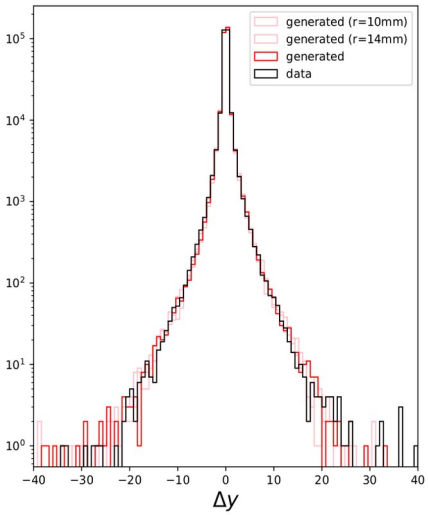
Results using a conditional WGAN (I)

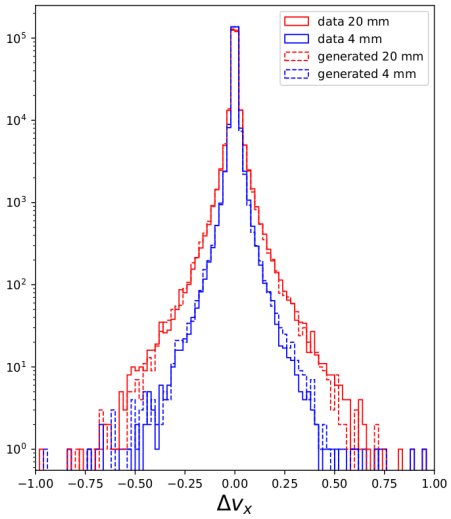
- The WGAN also provides very good results in describing the 1D distributions (and correlations)
- Also exhibits nice interpolation capabilities (good distributions for cases absent in the training)

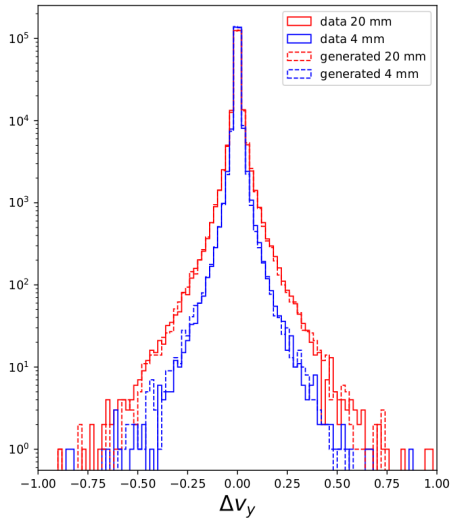


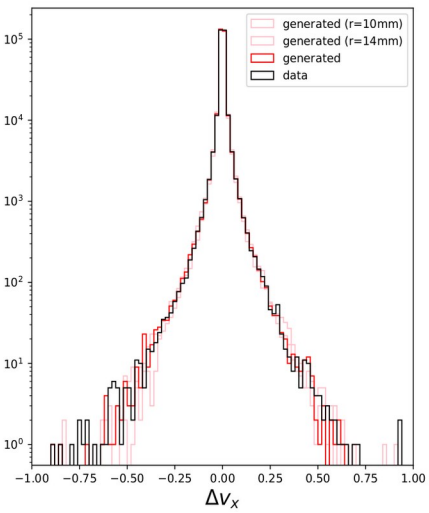


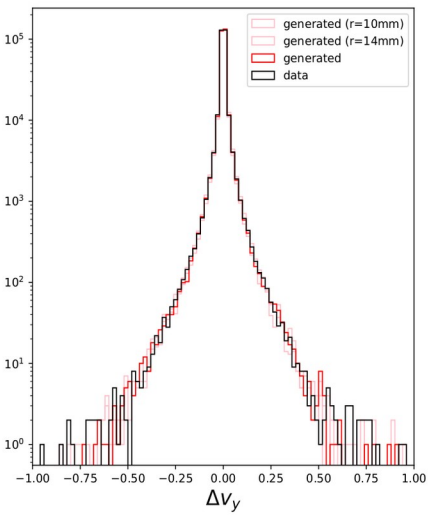








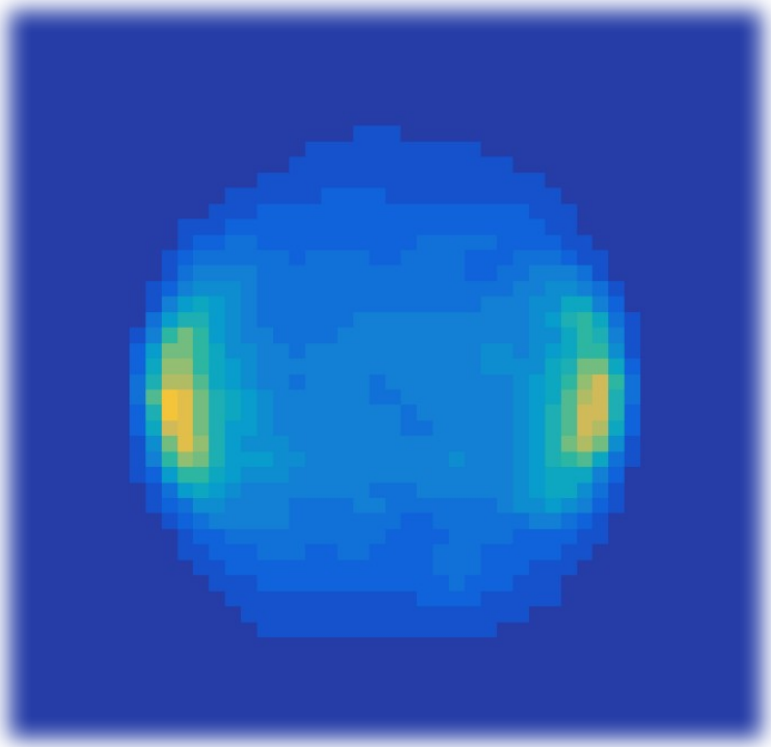




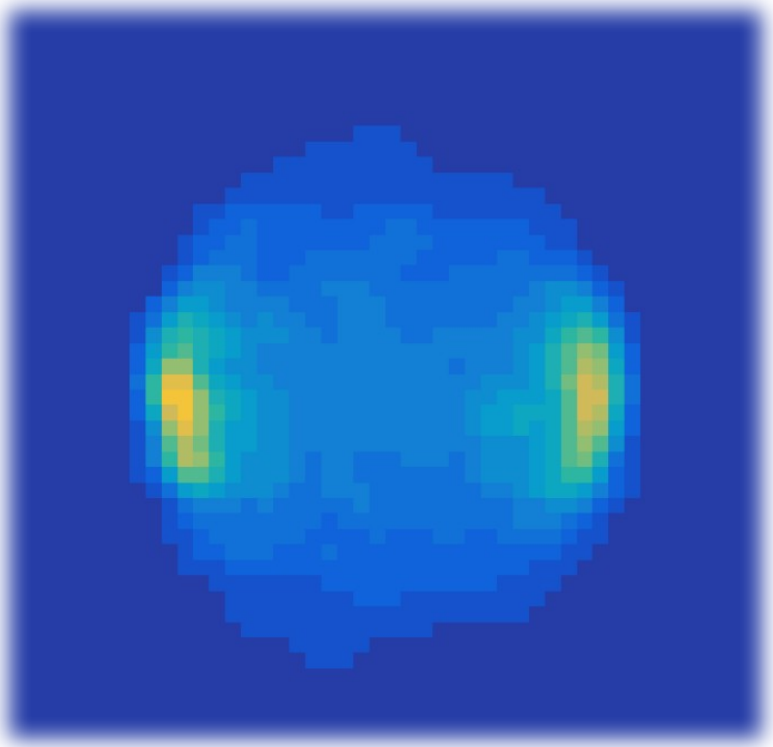
Results using a conditional WGAN (I)

- The purpose of this system is to produce simulations quickly to be used with a ML method
- To proof that, we really need to see whether the GAN has “absorbed” the geometry
- First thing we can do is to see how the POCA estimation looks for this simulation

Geant4 generated - 16 mm thickness pipe



GAN generated - 16 mm thickness pipe

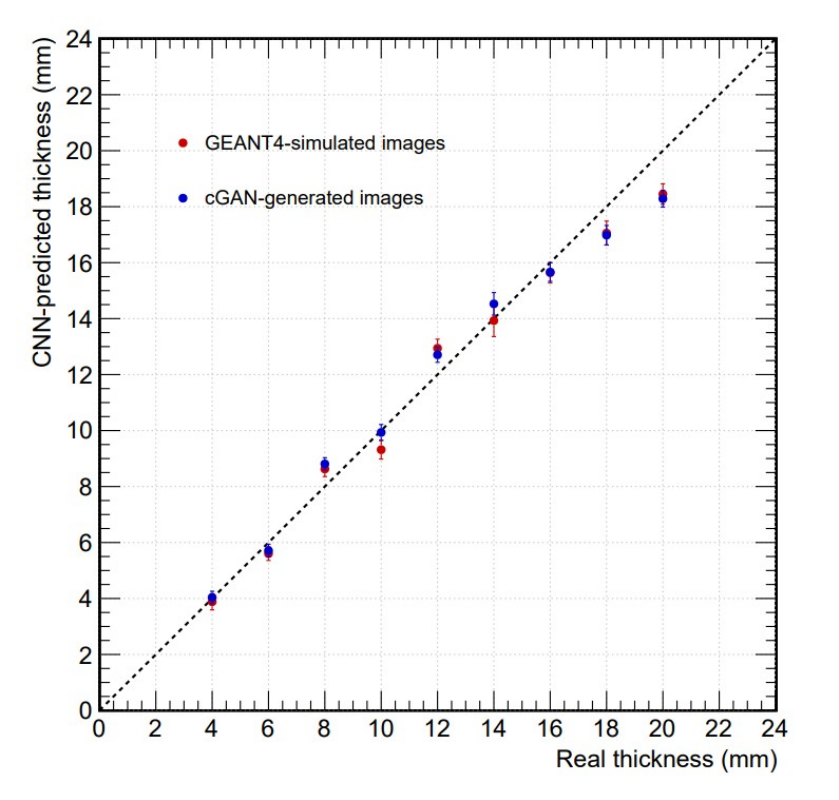


Results using a conditional WGAN (II)

- A convolutional neural network has been trained to predict the pipe thickness based on POCA
- Training dataset based on 90 POCA images per thickness (4, 6, 8, …, 18 mm thickness)
- Each POCA image is made with 300K muons based on CRY + GEANT4 simulation
- The CNN uses RestNET50 with 4 additional dense layers with 1024, 512, 512 and 256 nodes

The CNN scores similarly on both GEANT4 and GAN samples indicating the validity of the method.

GAN generation is 80 times faster.

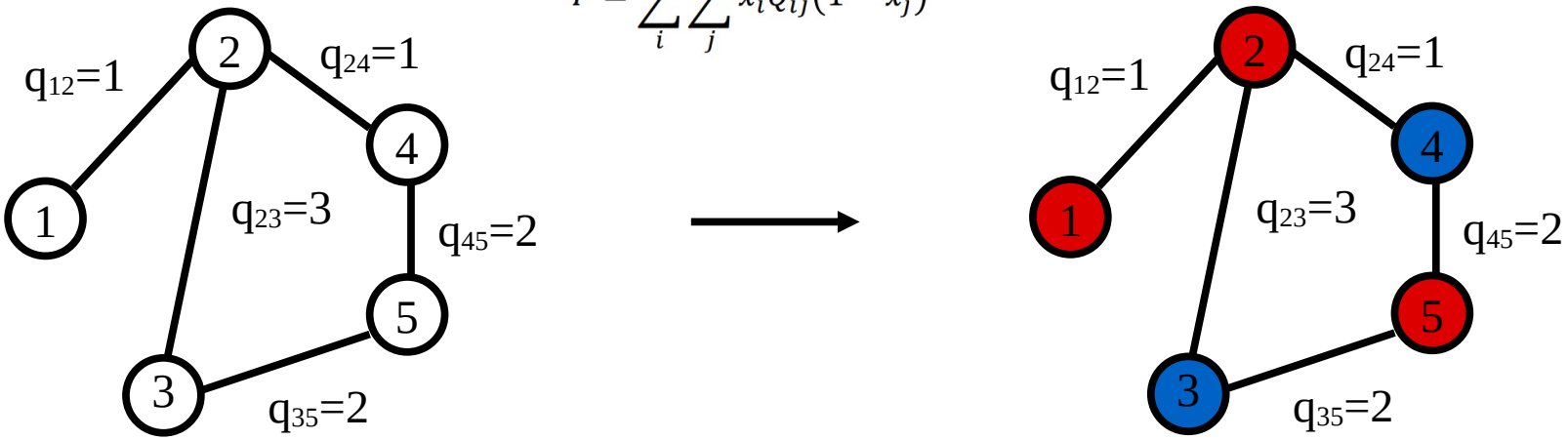


Vertex Reconstruction

Vertex reconstruction in Quantum Computing

- Vertex Reconstruction is the process of clustering tracks into a set of vertices
- This problem is combinatorial in nature: consider a problem with 2 true vertices.
  - Which track combinations minimize their relative distances and maximize to the others?
  - Actually this problem can be seen as a well-known problem in Graph Theory: Max-Cut
- Given a graph with nodes and weighted edges → assign labels (red or blue) to the nodes in such a way that the sum of the weights crossing from one group to the other is maximal
- Encoding the solutions as vectors x = [0, 1, 0, 1, 0] then need to maximize:

$$F = \sum_i \sum_j x_i Q_{ij} (1 - x_j)$$



https://doi.org/10.1051/epjconf/202227409002

Vertex reconstruction in Quantum Computing

- Consider tracks 3D points as a fully connected graph with weights equal to their distance
- Finding the assignation of tracks to vertices equals to finding two groups that maximize their mutual distance to each other → which is precisely a Max-Cut problem
- Most Quantum Computers can implement an Ising Hamiltonian of the form

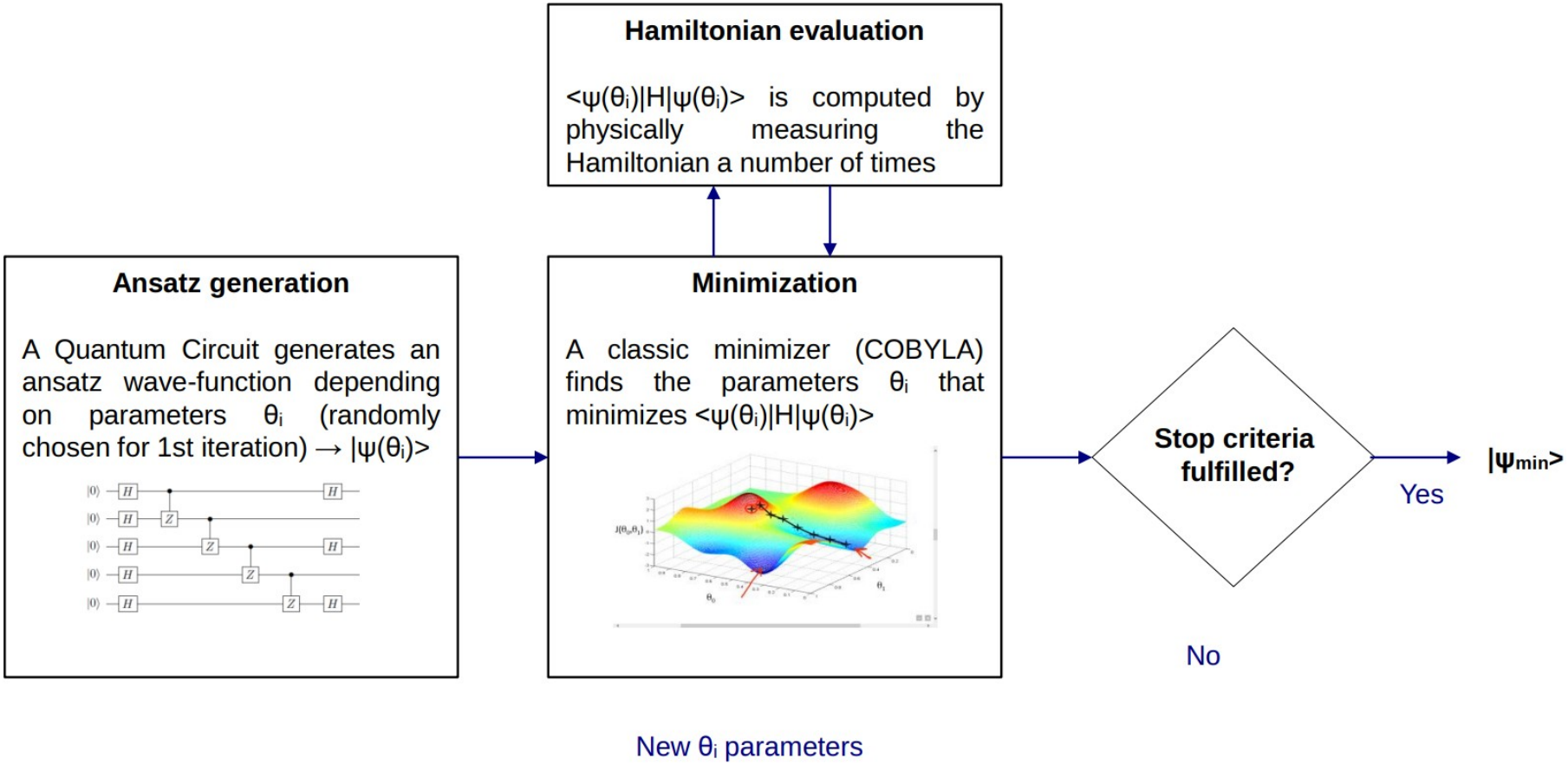
$$H(\sigma) = -\sum_{\langle i\ j\rangle} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j \longrightarrow F = -\sum_i\sum_j x_i Q_{ij}(1-x_j)$$

- The group assignment can be encoded in a set of quantum bits $A = [q_1, q_2, q_3, \ldots, q_N]$
- The x_j operator is defined to be $1/2(1 + \sigma_j)$ with value 0 or 1 when applied to q_j
- The quantum state for which this Hamiltonian is minimum is the solution to the problem

https://doi.org/10.1051/epjconf/202227409002

The VQE algorithm

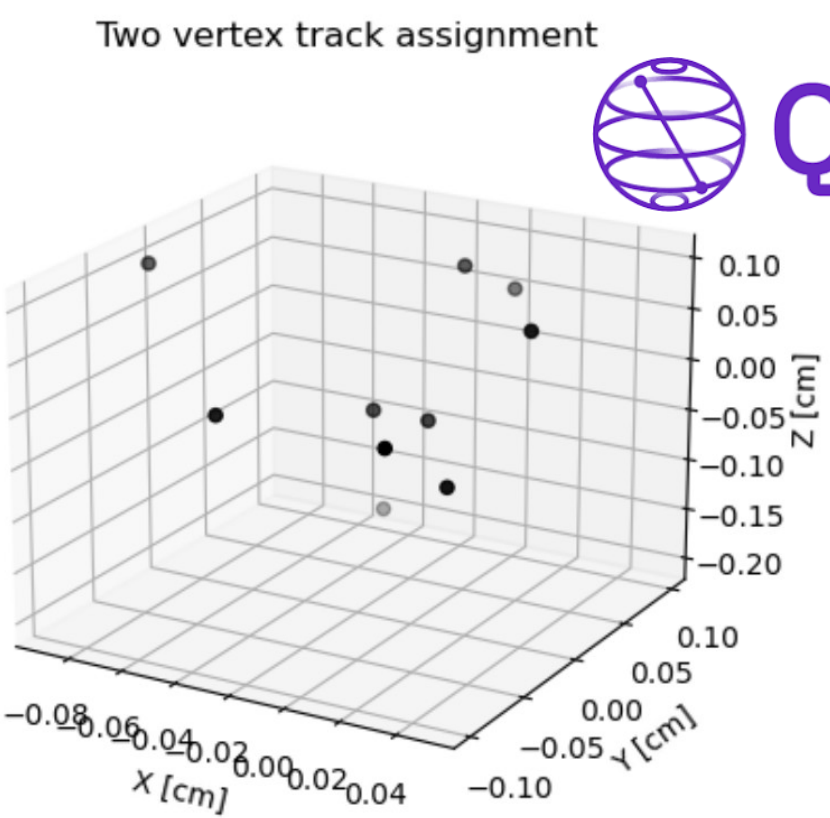
- The Variational Quantum Eigensolver is a hybrid classic-quantum algorithm
  - Aiming at finding the multi-qbit state that minimiez a given Hamiltonian

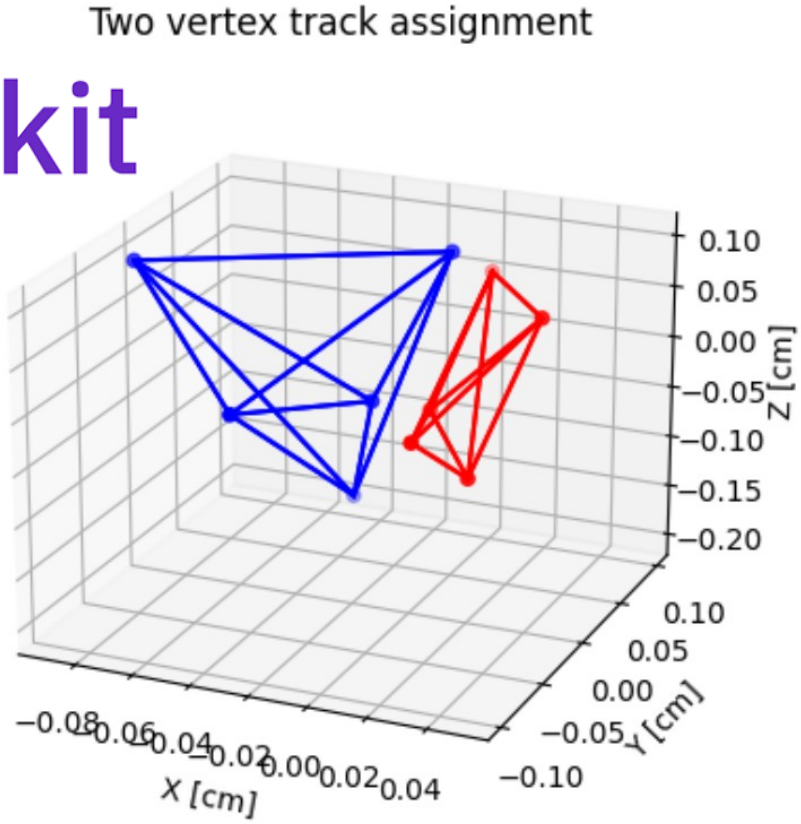


https://doi.org/10.1051/epjconf/202227409002

Vertex Reconstruction: results

- Algorithm implemented in the simulation framework of IBM Qiskit
- Reconstruction efficiency above 90% for vertices statistically separated below 1 mm





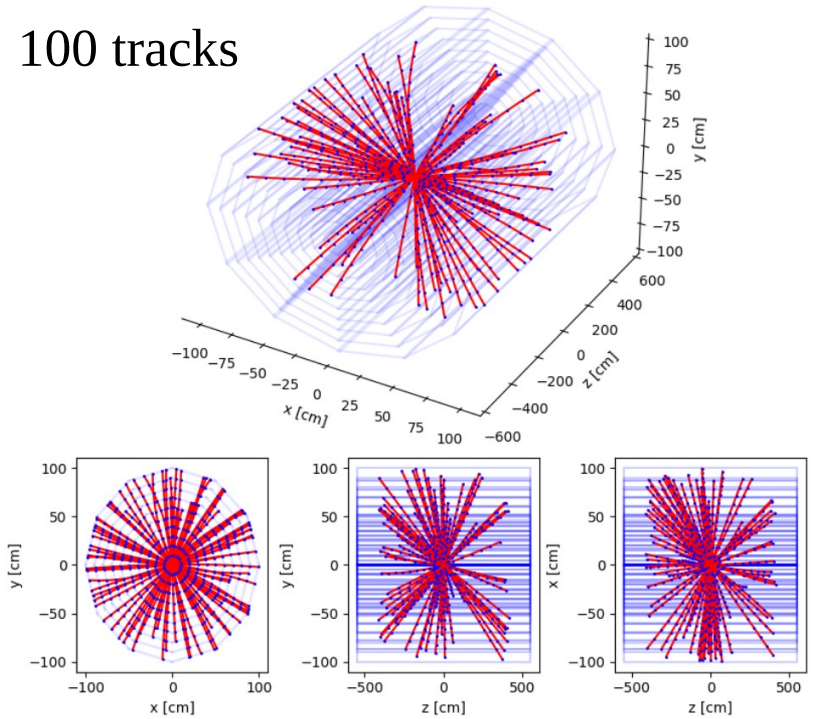
https://doi.org/10.1051/epjconf/202227409002

Track Reconstruction

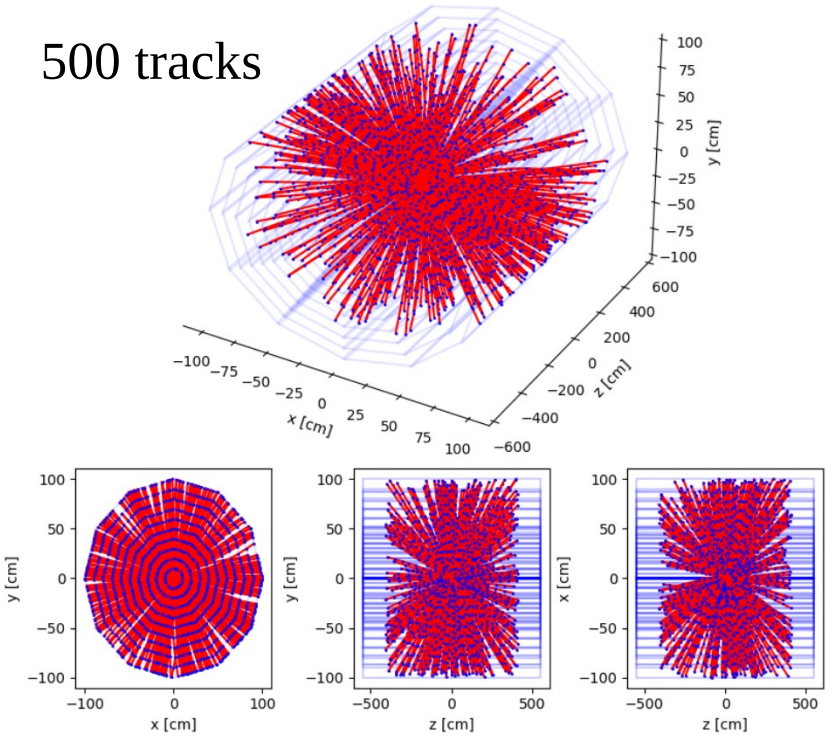
Track reconstruction using GNN (I)

- Graph Neural Networks have proven to be extremely succesful for track reconstruction
- We tested a SAGE (Sample and AggregatE) architecture based on PyTorch Geometric
  - It predicts the value of the edges between hits: 1 if they belong to the same track 0 if not
- Tested on a fake detector with curved tracks and perfect resolution

100 tracks

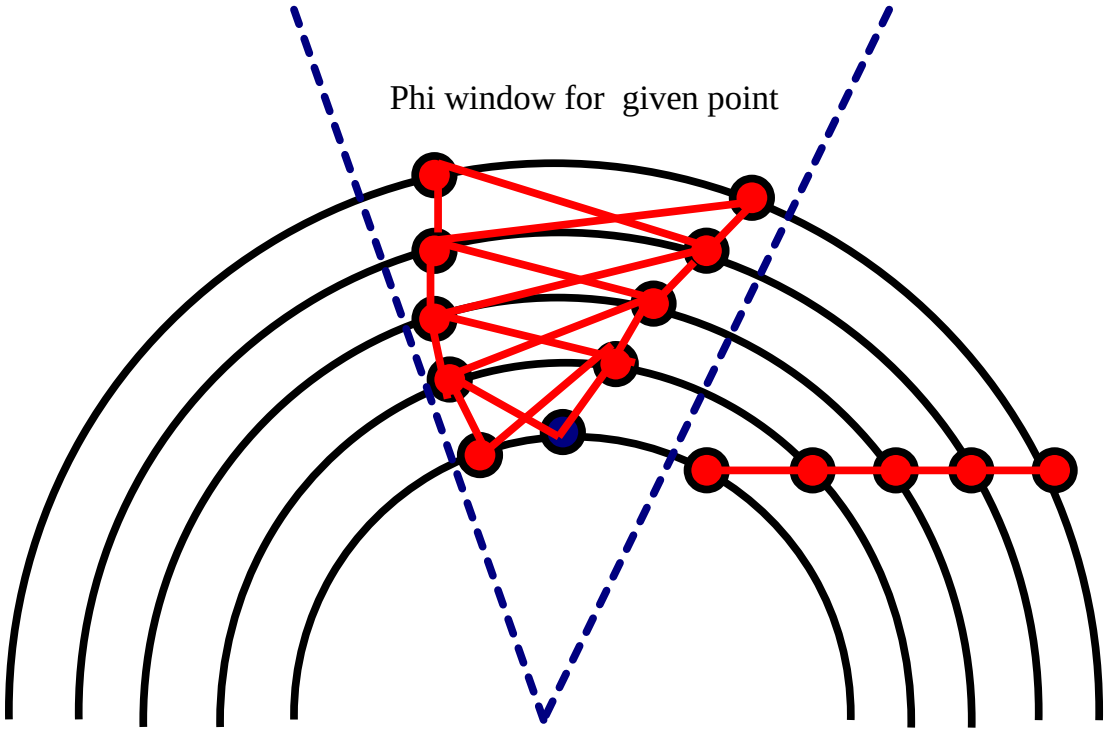


500 tracks



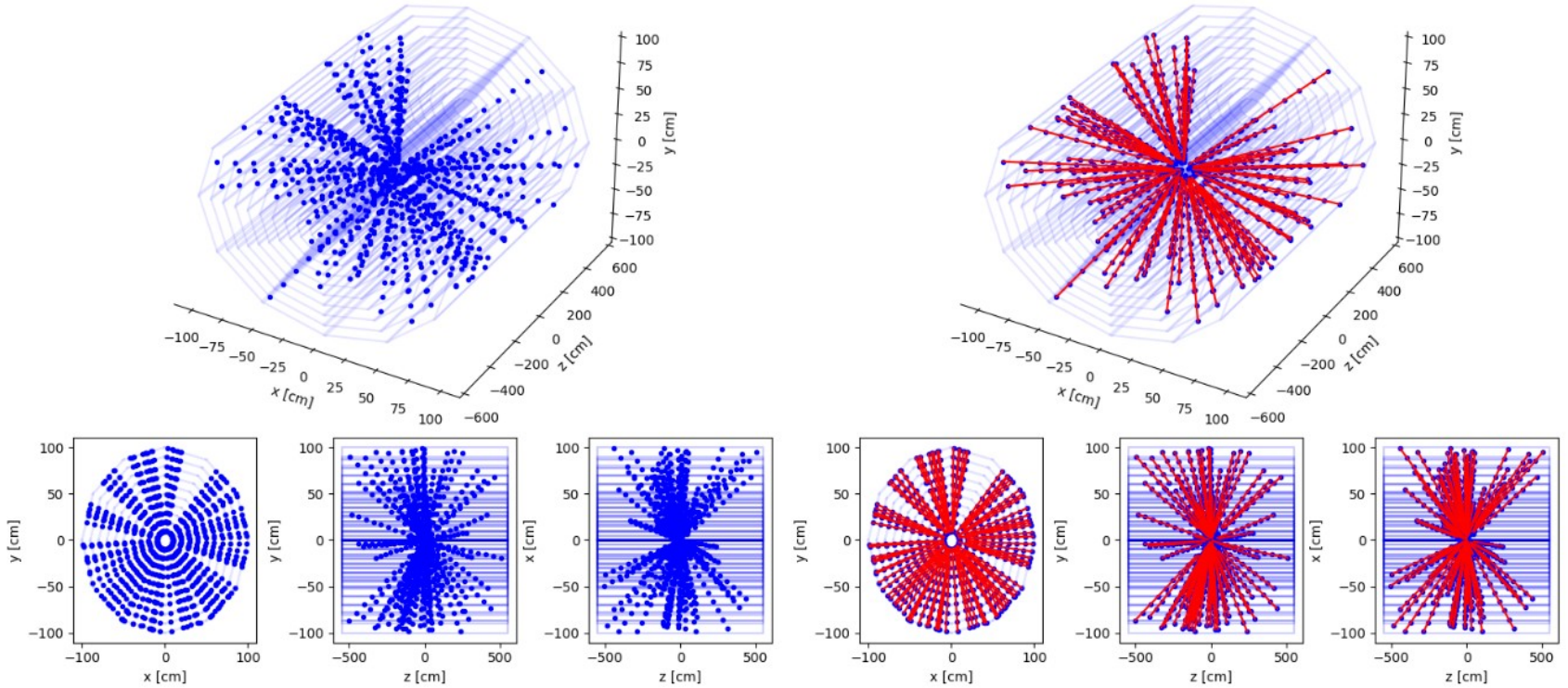
Track reconstruction using GNN (II)

- The complexity of the problem is reduced by introducing two constraings:
  - Only hits in consecutive tracker layers are susceptible to be connected
  - A phi window is introduced for every hit in order for others to be connected



Track reconstruction using GNN (III)

- Example for an event with 100 tracks (left hit cloud, righ GNN reconstructed tracks)



Track reconstruction using GNN (III)

- Very high track reconstruction efficiency for a phi window of 20º but degrades rapidly
- Training performed using only a set of 1000 tracks → higher numbers needed for higher angles

Detector design

Differential programming (quick definition)

- A new paradigm in which a computer program/function can be differentiated
- This is achieved by using automatic differentiation usually exploiting the chain rule

```
def myTargetFunction(x):

    x1, dx1dx = funcA(x)
    y, dydx1 = funcB(x1)
    return y, dydx1*dx1dx
```

```
def funcA(x):

    x1 = x*x
    dx1dx = 2*x
    return x1, dx1dx
```

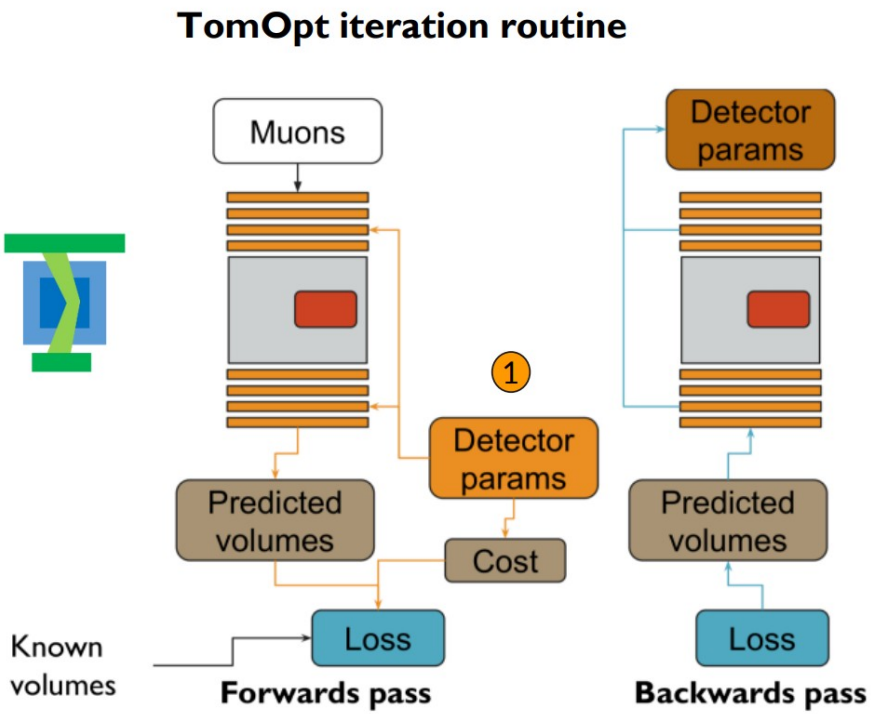
```
def funcB(x):

    y = 1.0 / (1.0 + x)
    dydx1 = -1.0/(1.0 + x)**2
    return y, dydx1
```

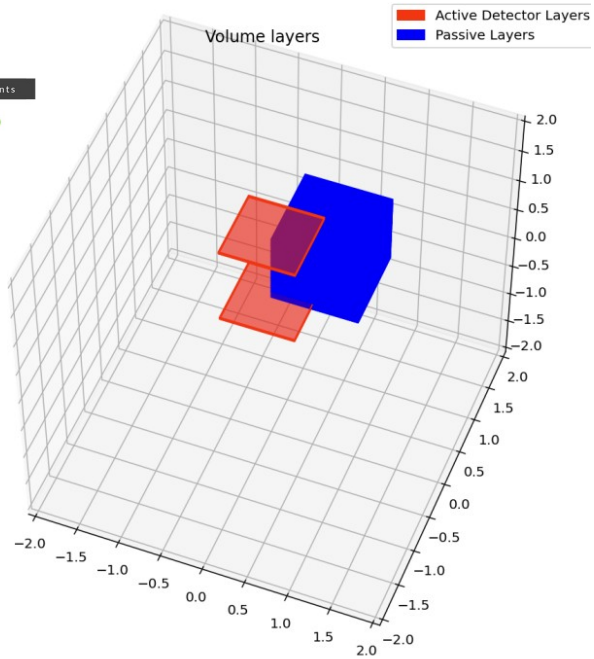
- This technique allows to quickly and efficienctly estimate gradients of complex functions
- It is possible to minimize complicated loss functions using the gradient and SGD
- In a very simplistic way you can see this as a generalization of the backpropagation method
  - But applied on generic functions and not on simple structures such as neurons

https://doi.org/10.48550/arXiv.2309.14027

Differential programming for detector design

- Optimal design/configuration of a particle detector can be estimated using DP
- The objective function should contain metrics about all important parameters in the design:
  - Performance (efficiencies, resolutions, etc), cost, constraints in the system
- These ideas are being exploited to produce optimal design for a muography experiment





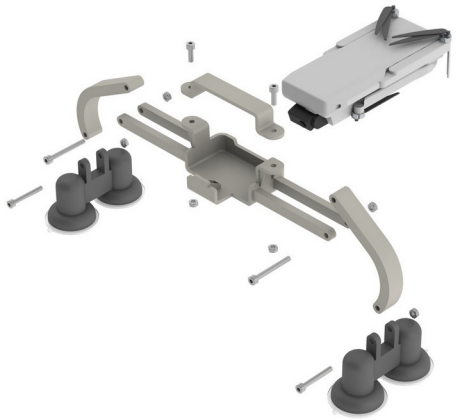
https://doi.org/10.48550/arXiv.2309.14027

Spin-offs and collaborations

MAPSIA: Automatic Road Defect detection

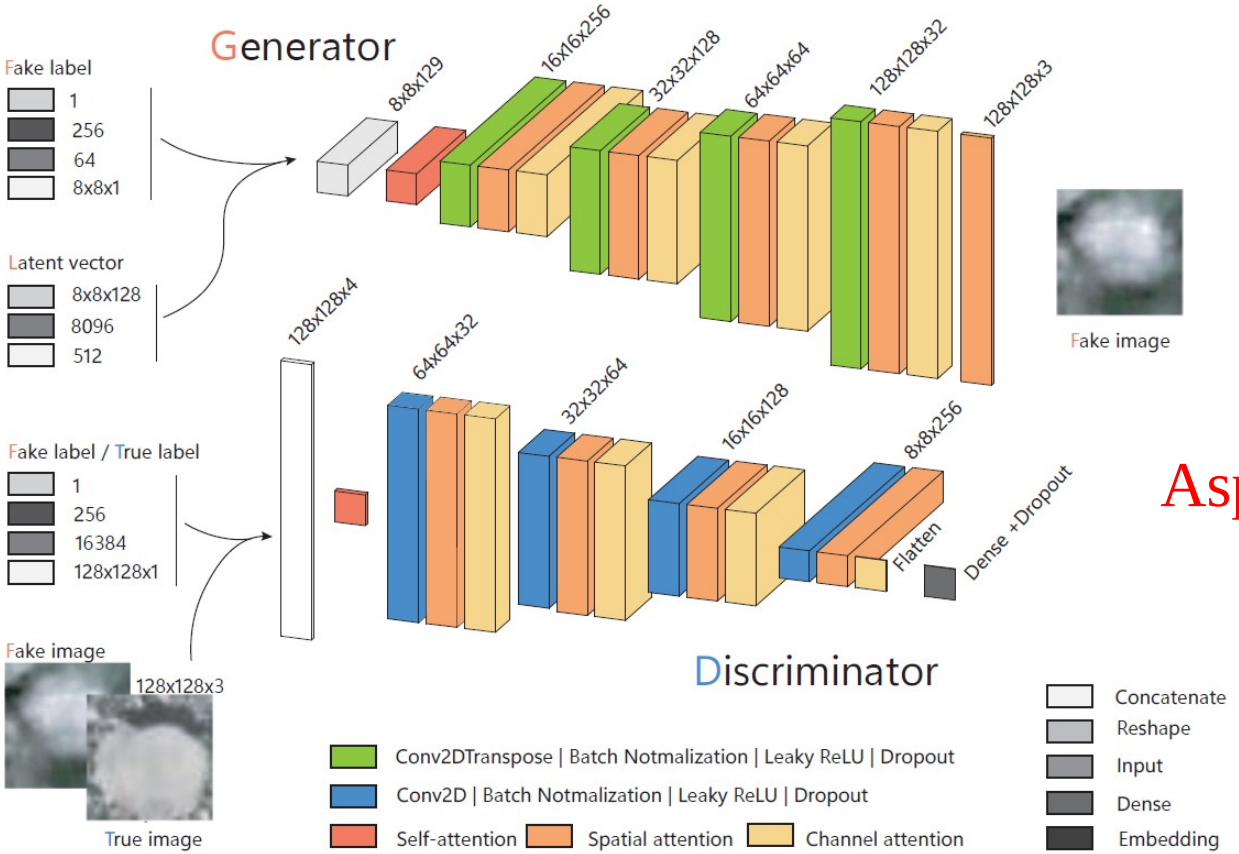
- We initiated a collaboration with our civil engineering colleagues on AI for road security
- Building and training a system to automatically detect and classify defects in the pavement
- As a side product we also developed:
  - GAN model with attention mechanisms to generate fake defect images
  - Diffusion model to generate fake defect models
  - Generative model with segmentation to generate fake defect images from drawings
- We actually built a system to collect the images: the Mosquito!

MAPSIA
MACHINE LEARNING ON THE ROAD



AsphaltGAN: Generating fake images

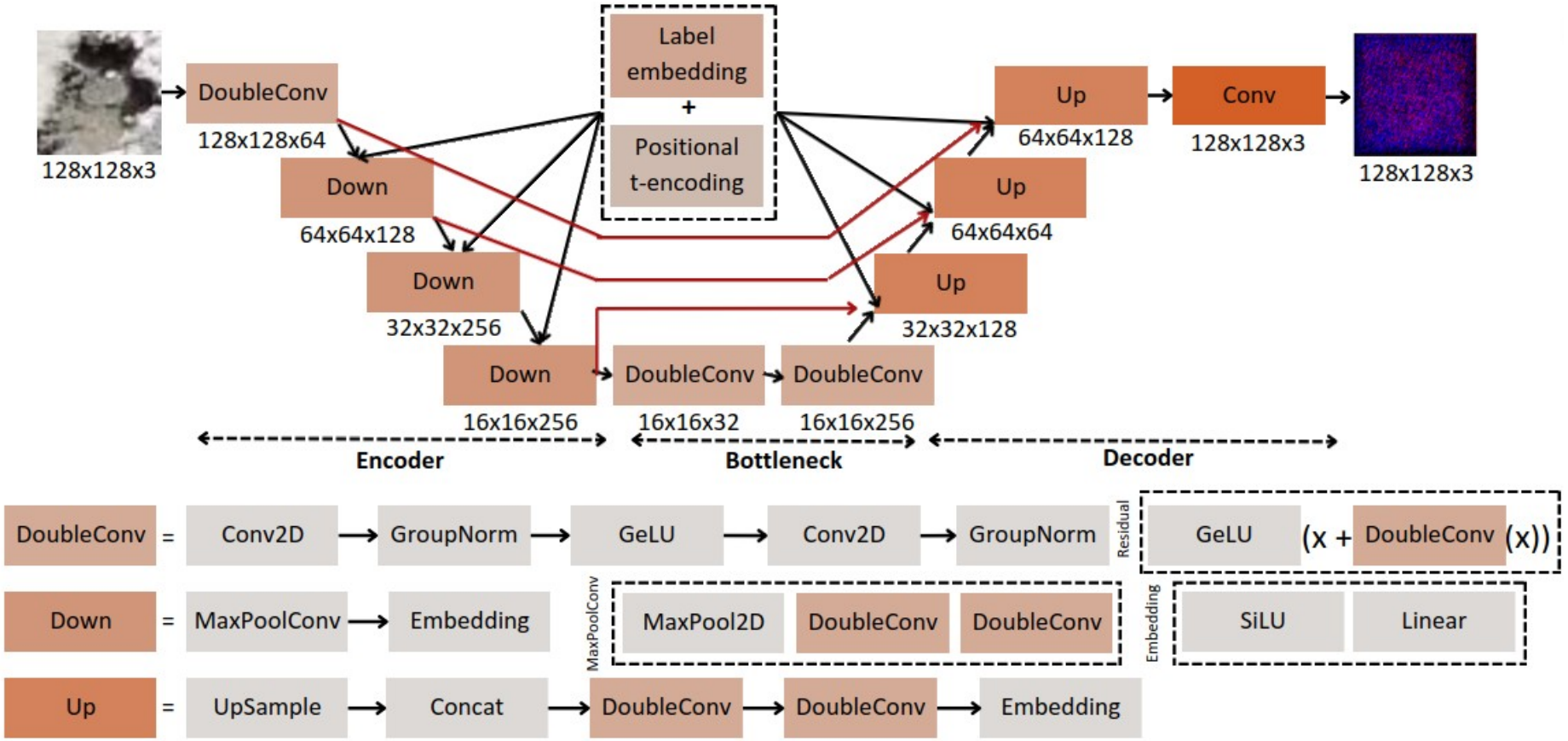
- A dataset of images was collected with Mosquito however some of the defect categories were missrepresented so we decided to make a system to generate those kind of defects



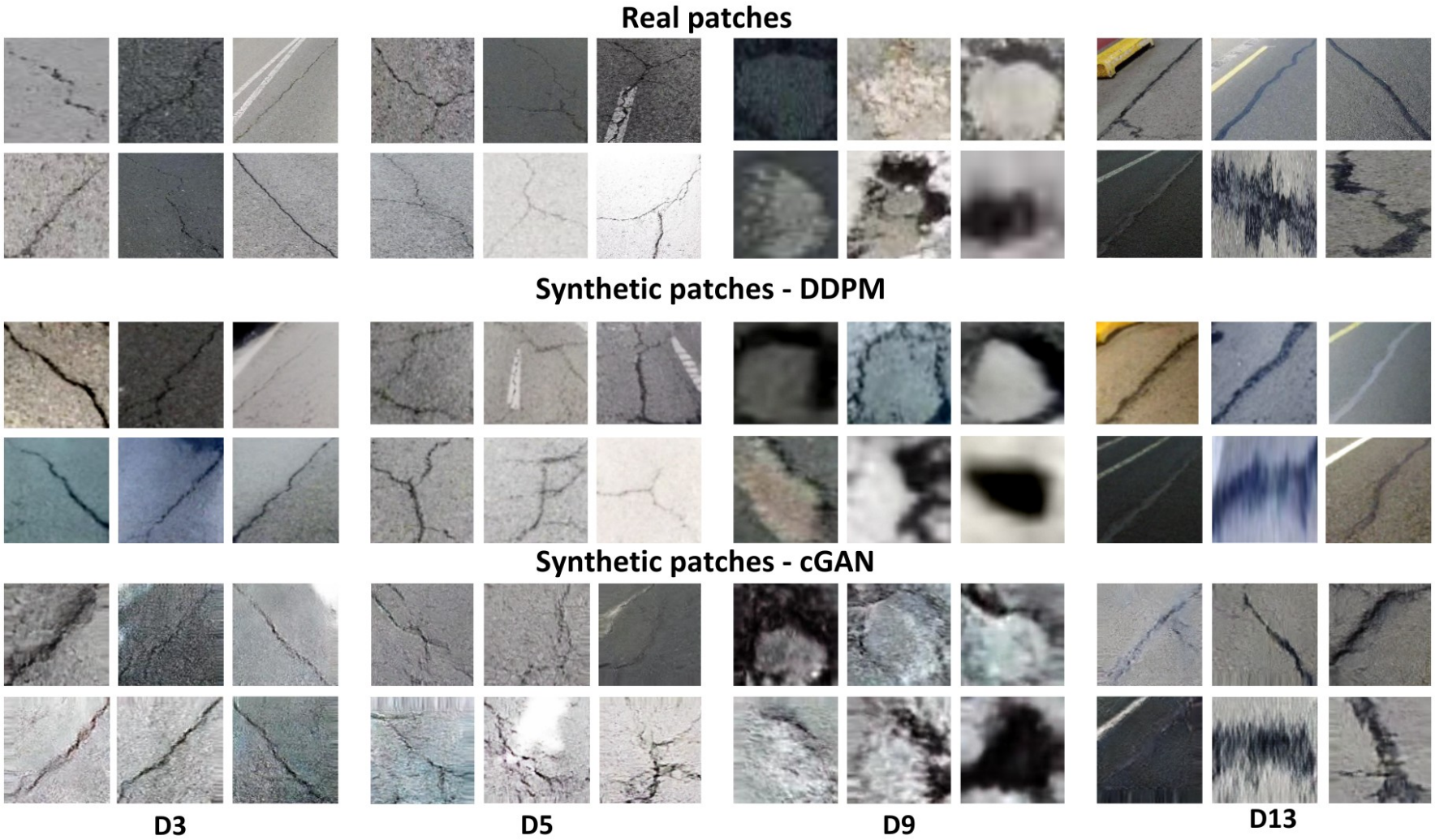
AsphaltGAN

cDDPM: Generating fake images (difussion)

- We also try a system based on difussion: cDDPM

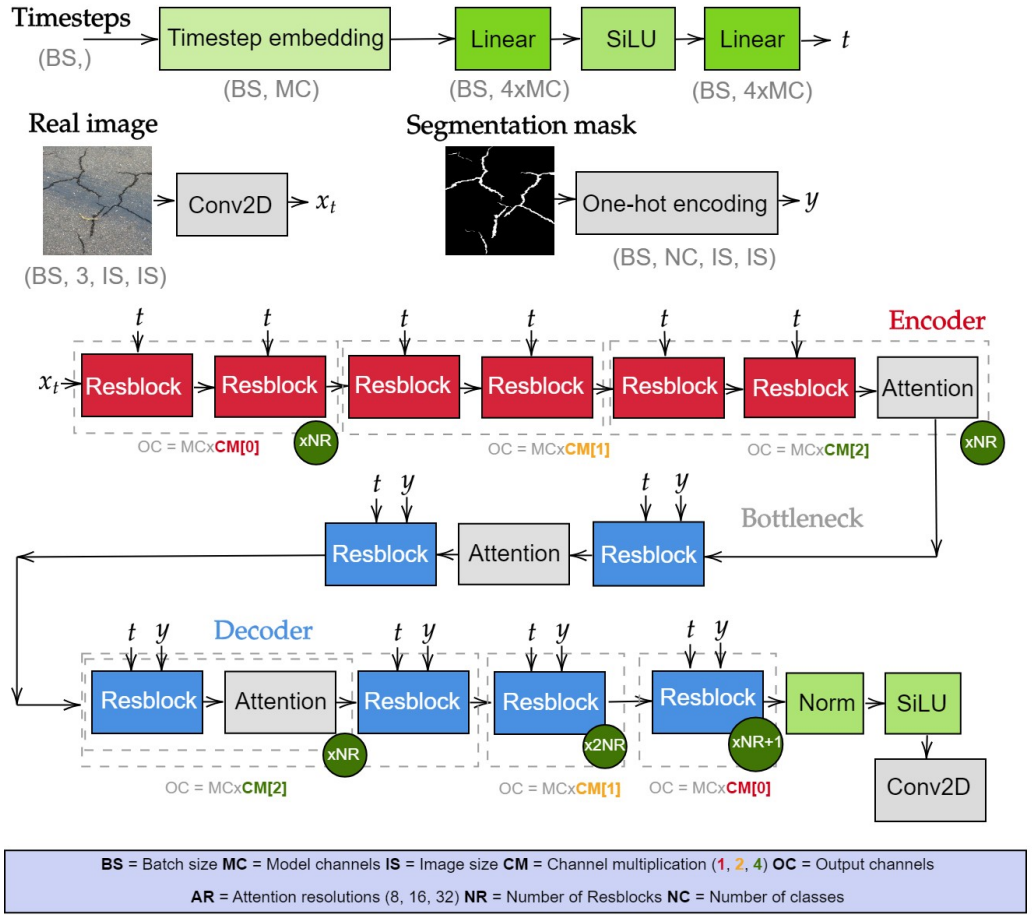


cDDPM: Generating fake images (difussion)

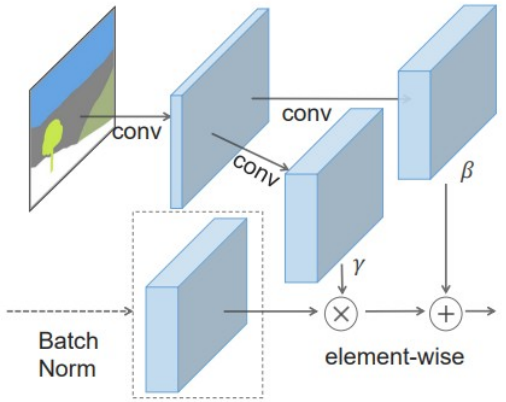


Road Painter: from drawing to image

- We finally also tried a SPADE architecture to convert a hand made mask into a road defect

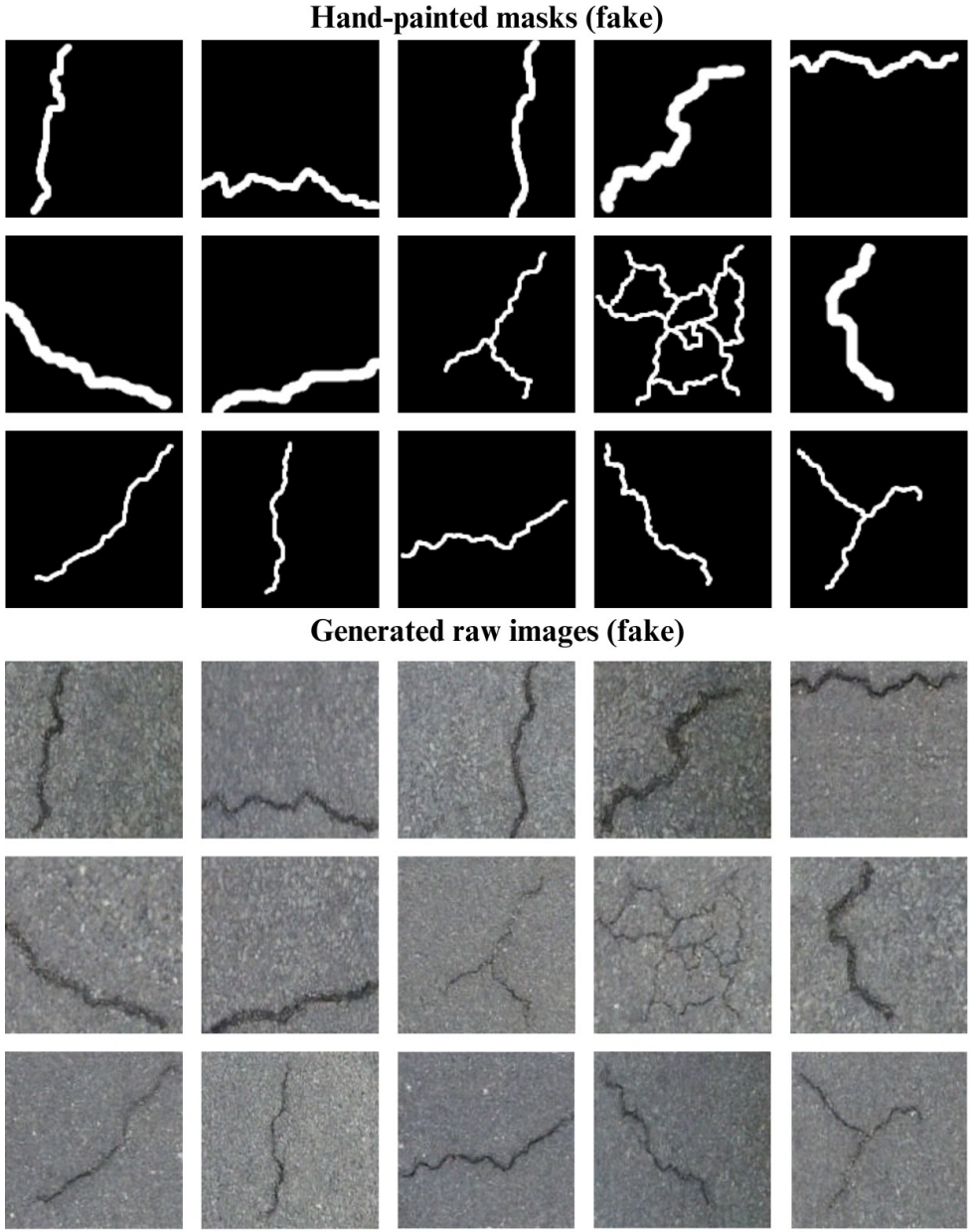


Space decoder



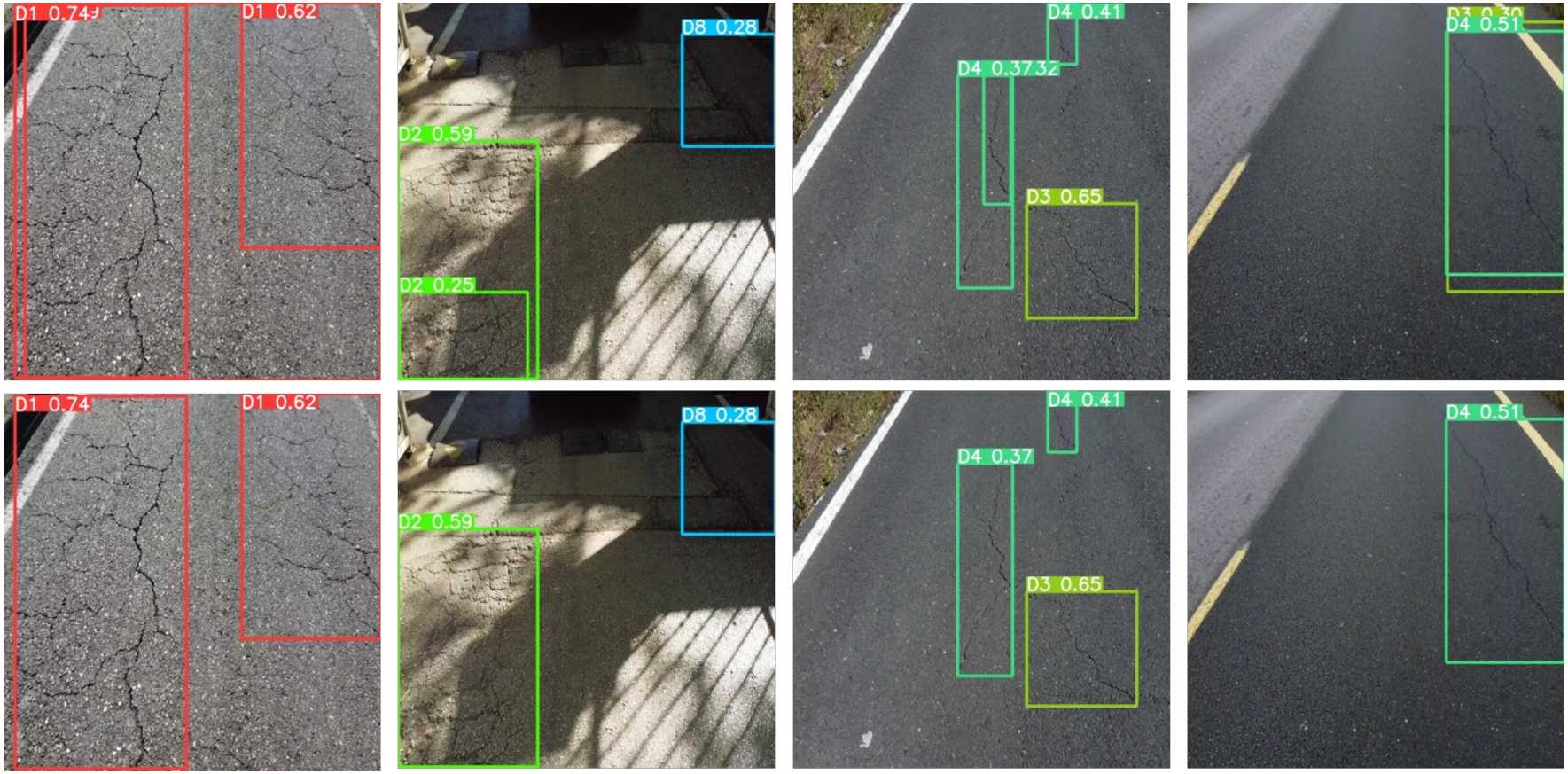
Road Painter: from drawing to image

- Realistic road images obtained from the hand-painted masks



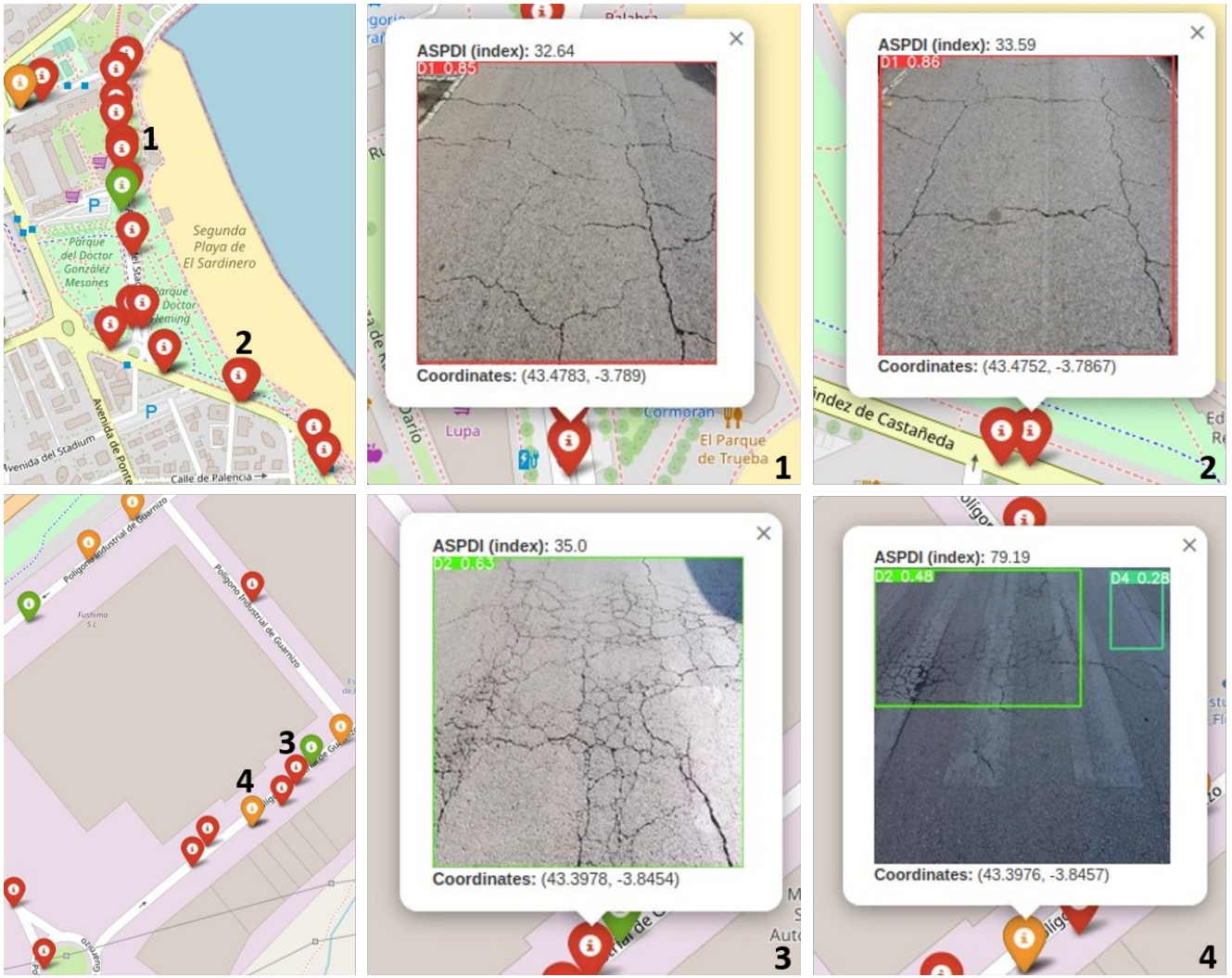
Road Painter: from drawing to image

- A detection/classification architecture based on YoloV5 is devised
  - The system is trained with both real and synthetic images
  - High efficiency (>80%) and low fake rate (< 10%) for all types of defects



Road Painter: from drawing to image

- Defects are also geolocalized and automatically updated on a web monitoring tool



Conclusions

- New computing paradigms are emerging, growing and changing the world as we know it
- At IFCA many different algorithms are being used and tested for many different tasks
- In this talk I have shown some examples concerning:
  - Fast simulation, vertex reconstruction, track reconstruction and experiment design
- Also an spin-off collaboration based on our knowledge on AI systems has been explained