Ultra fast simulation algorithmics for industrial applications in Muon Tomography using Generative Adversarial Networks

Rubén López Ruiz,
Pablo Martínez Ruiz del Árbol, Celia Fernández Madrazo

Instituto de Física de Cantabria - Universidad de Cantabria

26/07/2023

3rd MODE Workshop
(Princeton, NJ)

Introduction: cosmic muons

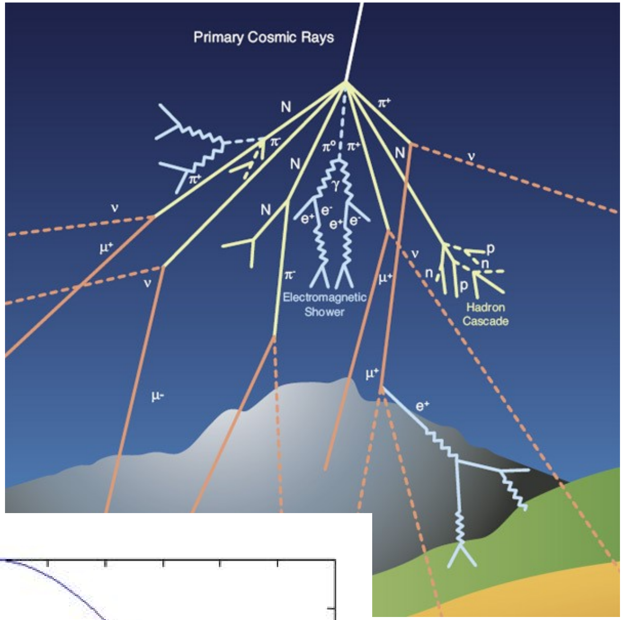
→ The Earth is constantly being hit by high energy particles (cosmic rays)

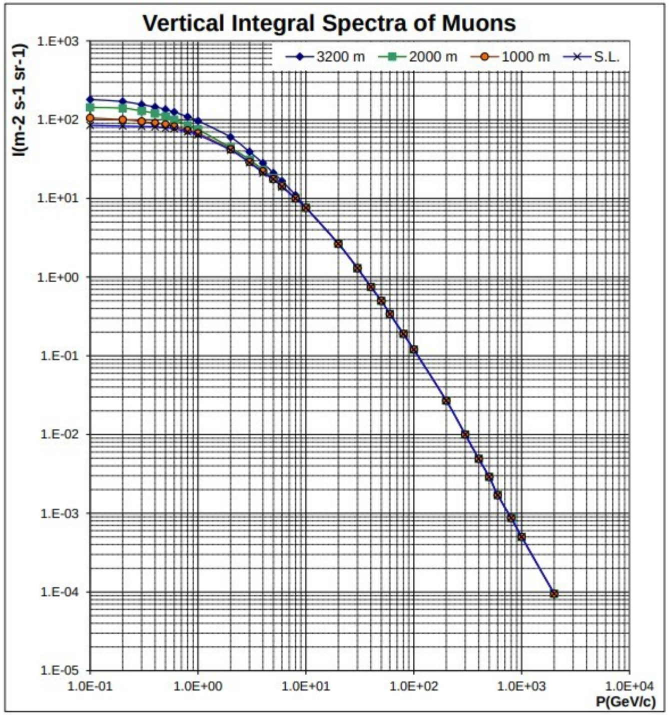
→ 98% protons, 1.8% alpha, 0.2% other

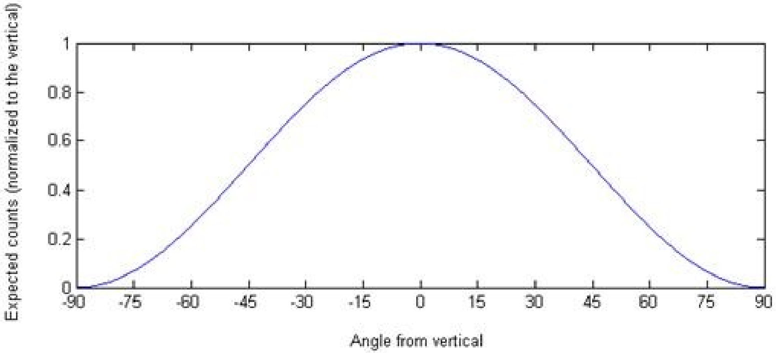
→ **Cosmic muons** are a product of the interaction of cosmic rays with the nuclei of the atmosphere

→ Surface rate: 10000 muons/m^2min

→ Flux proportional to $\cos^2(\theta)$ (angle with vertical)







Muon tomography

→ As they travel, cosmic muons interact with matter in two ways:

Ionization

$$-\left\langle \frac{dE}{dx}\right\rangle = \frac{4\pi}{m_e c^2}\cdot\frac{\boxed{n}\, z^2}{\beta^2}\cdot\left(\frac{e^2}{4\pi\varepsilon_0}\right)^2\cdot\left[\ln\left(\frac{2m_e c^2\beta^2}{\boxed{I}\cdot(1-\beta^2)}\right)-\beta^2\right]$$

$$\boxed{n = \frac{N_A\cdot Z\cdot\rho}{A\cdot M_u}}$$

Mean excitation potential

Multiple Coulomb scattering

Momentum dependence

$$\theta_0 = \frac{13.6}{\beta \boxed{cp}}\, z\sqrt{x/\boxed{X_0}}\,[1+0.038\ln(x/\boxed{X_0})]$$

$$\boxed{X_0 = 716.4\,\mathrm{g\,cm^{-2}}\,\frac{A}{Z(Z+1)\ln\frac{287}{\sqrt{Z}}}}$$

Energy loss in these processes depend on density, composition and size of objects

→ These are statistical processes driven by nature.

Muon tomography (or muography) is a Non-Destructive Testing technique (NDT) that makes use of cosmic muons to obtain images of inaccessible places

Muon tomography

→ In practice there are two types of muon tomography:

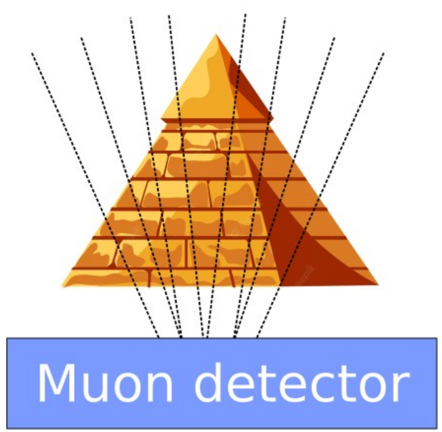
Absorption muography

→ Measures incident flux as a function of direction → Transmittance

→ Long exposure times

→ One detector

→ Large scale objects (pyramids, civil structures...)



Scattering muography

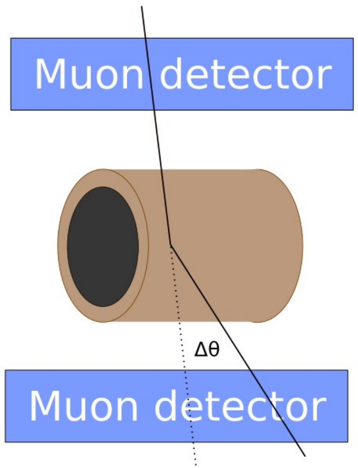
→ Measures position and angle shift → Change in muon trajectories

→ Shorter exposure times

→ Two detectors

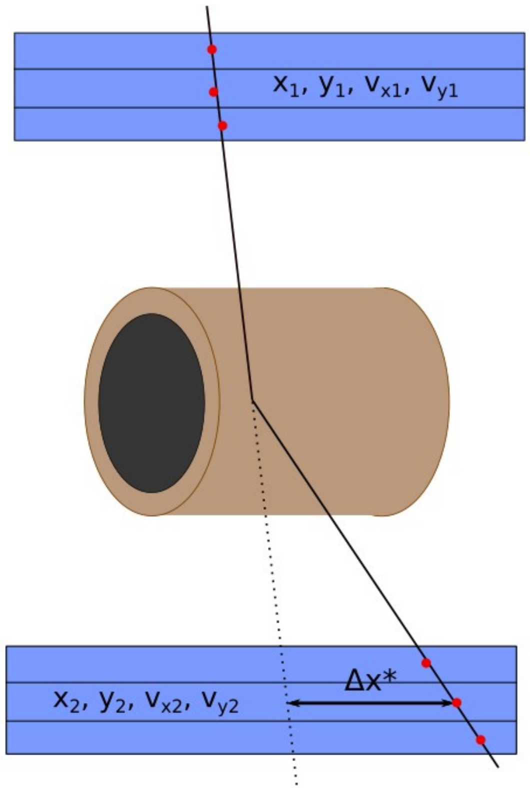
→ Small-medium scale objects

→ Industrial applications



Industrial applications

→ Scattering muography → preventive maintenance, quality control...



Reconstructed variables (at each detector):
- Muon position: **x, y**
- Muon trajectory vector:
  - $\mathbf{v_x = \tan(\theta_x)}$
  - $\mathbf{v_y = \tan(\theta_y)}$

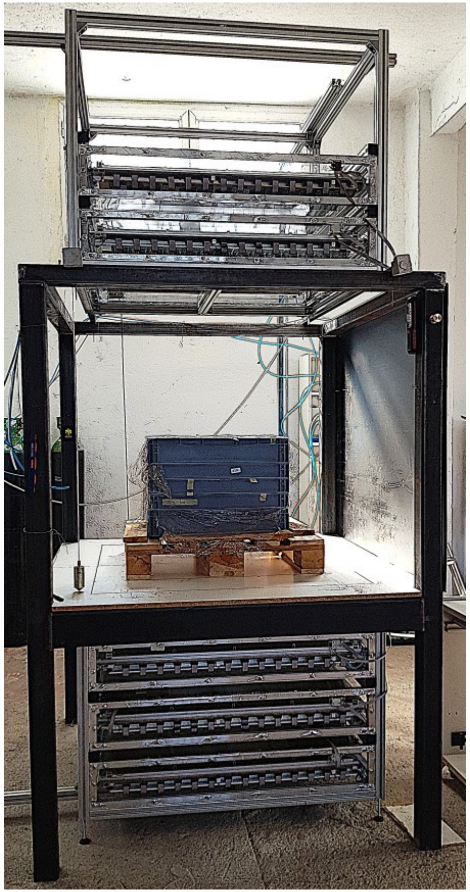
↓

Derived variables: $\mathbf{\Delta x^*, \Delta y^*, \Delta v_x, \Delta v_y}$

$$\Delta x^* = x_2 - x_1 + Lv_{x_1}$$
$$\Delta y^* = y_2 - y_1 + Lv_{y_1}$$
$$\Delta v_x = v_{x_2} - v_{x_1}$$
$$\Delta v_y = v_{y_2} - v_{y_1}$$

↓

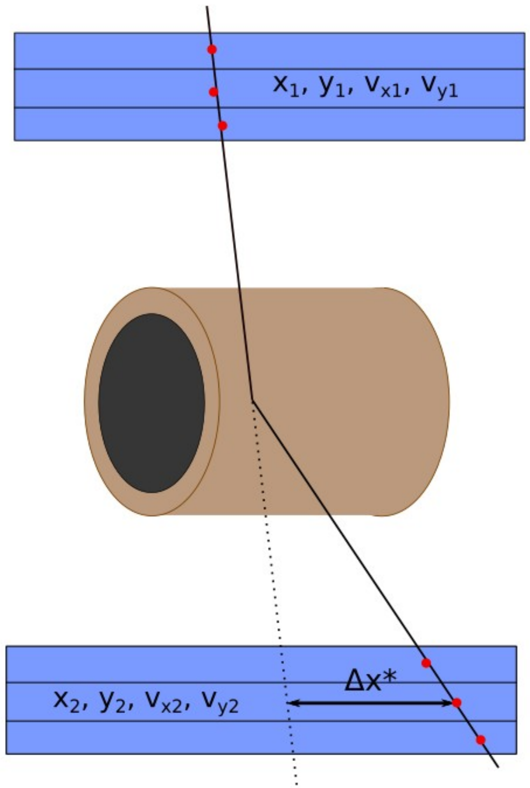
Information about intermediate object (composition, defects...)

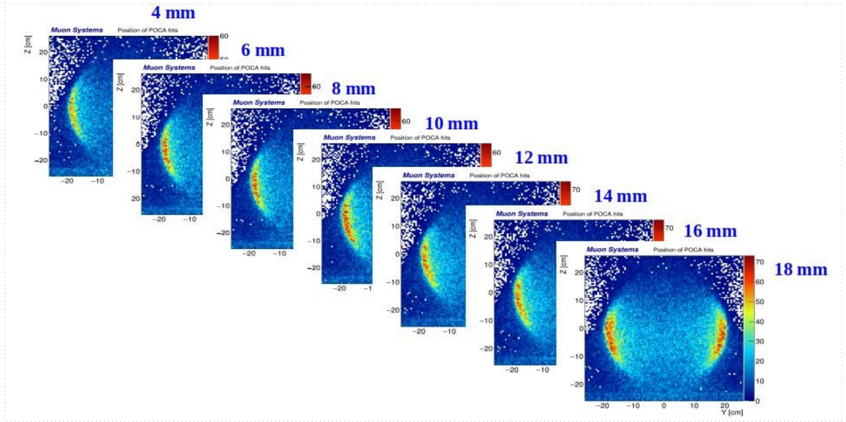


Industrial applications

→ **Particular application**: monitor the wear of steel pipes

1. Take muon data and reconstruct images of the pipes.
2. Feed images to a ML regression model
   → Infer the thickness of the pipes



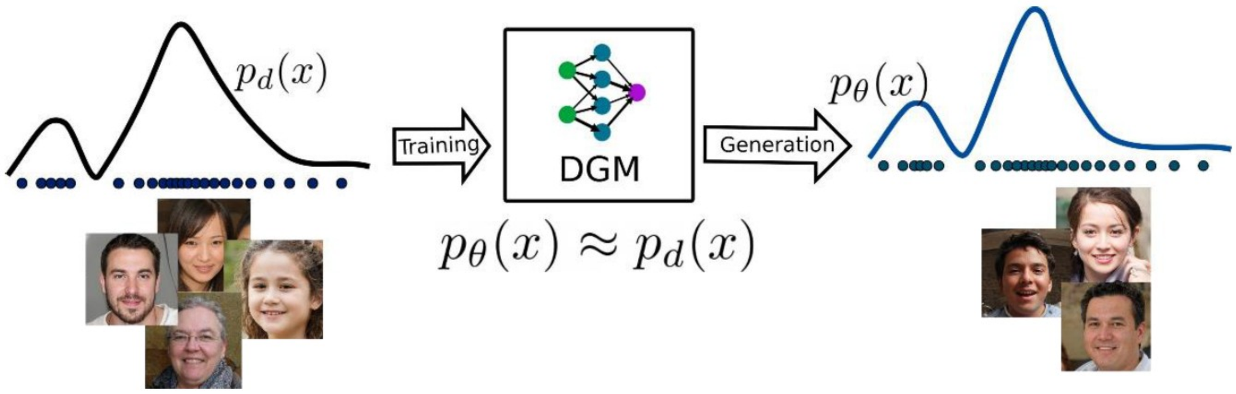


ML models require **lots of simulation data** for training.
→ CRY (Cosmic Ray Shower Library) and Geant4 (passage through material) → Slow and computationally expensive.

<u>Possible alternative</u>: **generative models (GAN)**

Generative Adversarial Networks (GAN) are a class of machine learning models based on deep neural networks that are capable (after proper training) of generating new synthetic data with the same characteristics as the training data.

→ Generative modeling: learn patterns in input data → produce new samples
→ Unsupervised learning task



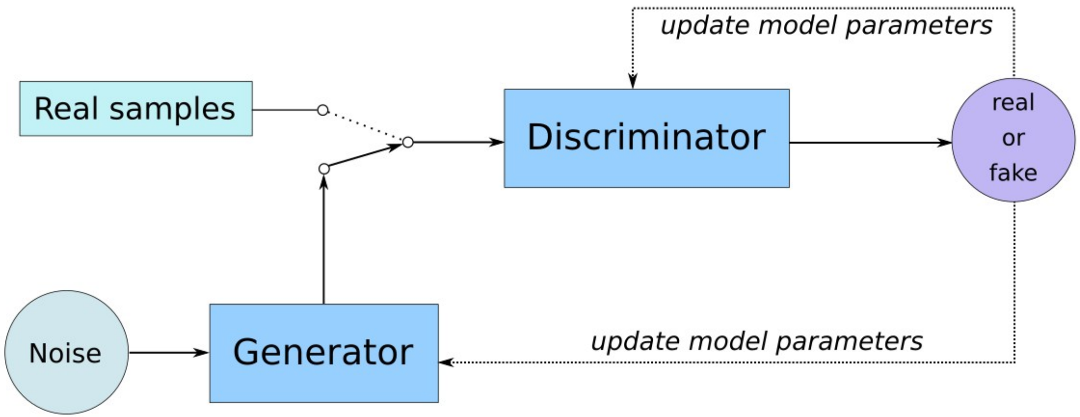
Gan framework. Ian J. Goodfellow (2014)
Idea: frame generative modeling as a supervised learning task.

Use of two submodels: → **Discriminator**: receives sample in the domain → classifies as real or fake.
(neural networks) → **Generator**: receives input noise → generates sample in the domain.

→ **Adversarial training**: both models trained together → shared loss function

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))].$$

Zero sum game:
- → Discriminator learns to identify generated samples
- → Generator learns to produce plausible samples

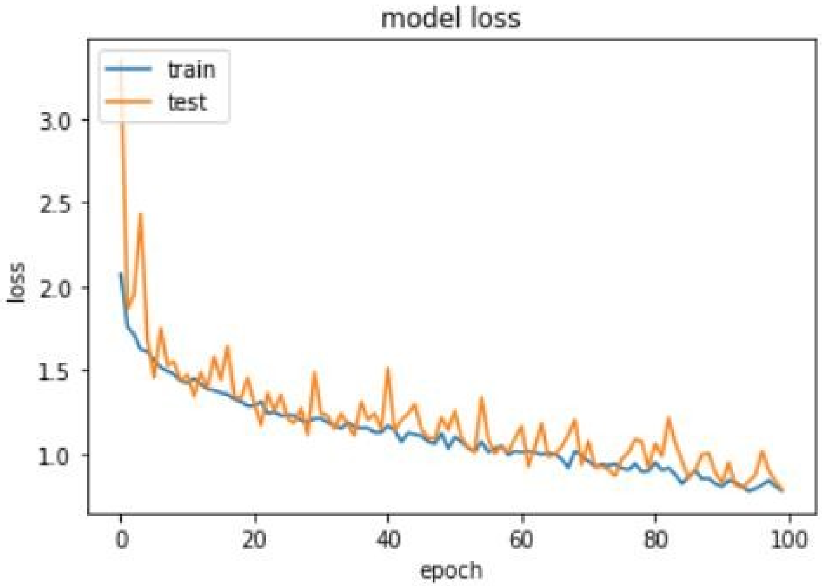


Adversarial training

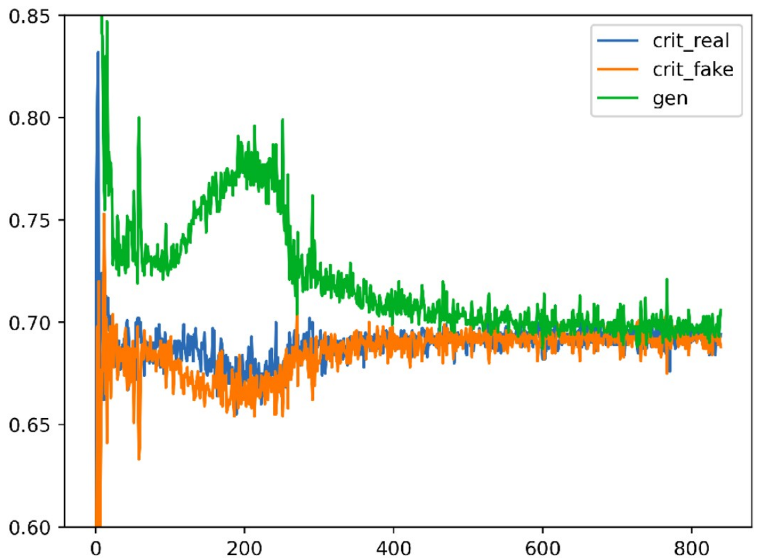
Ultimately both models reach an equilibrium where the generator produces good samples that fool the discriminator about half of the times.

→ **Training is hard and can be unstable**: fine tuning of parameters → equilibrium

Classification NN loss function



GAN loss function



Ultra fast simulation for muon tomography

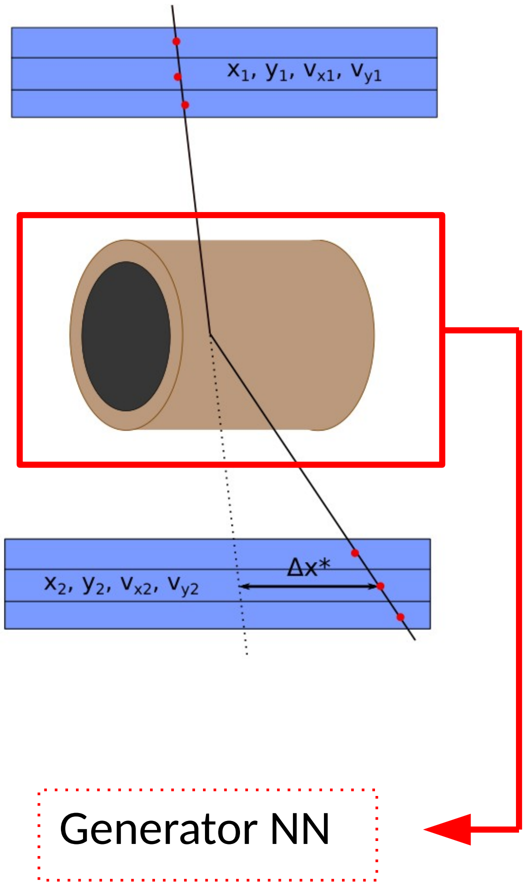
Set up: muon scattering tomography through metal pipes of different thickness

2 detectors measuring x, y, v_x, v_y → 8 variables

→ First detector measures original cosmic muon flux → cheap (CRY)

→ So we are interested in simulating 2nd detector info (propagation part)

We will use first detector info as additional input.



Goal: train a GAN to generate the variables that characterize the muon scattering through metal pipes.
→ Replace muon propagation part by a generative ML model.

$$\Delta x^* = x_2 - x_1 + L v_{x_1}$$
$$\Delta y^* = y_2 - y_1 + L v_{y_1}$$
$$\Delta v_x = v_{x_2} - v_{x_1}$$
$$\Delta v_y = v_{y_2} - v_{y_1}$$

Ultra fast simulation for muon tomography

Training data: simulation events of cosmic muons (CRY) and their passage through metal pipes of different thickness (Geant4).

Muon samples:

$$x_1, y_1, v_{x_1}, v_{y_1}, \Delta x^*, \Delta y^*, \Delta v_x, \Delta v_y, r$$

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

<u>Two experiments</u>:

1. **Simple GAN**: trained on data with only one thickness value.
   → Check if model produces plausible events.

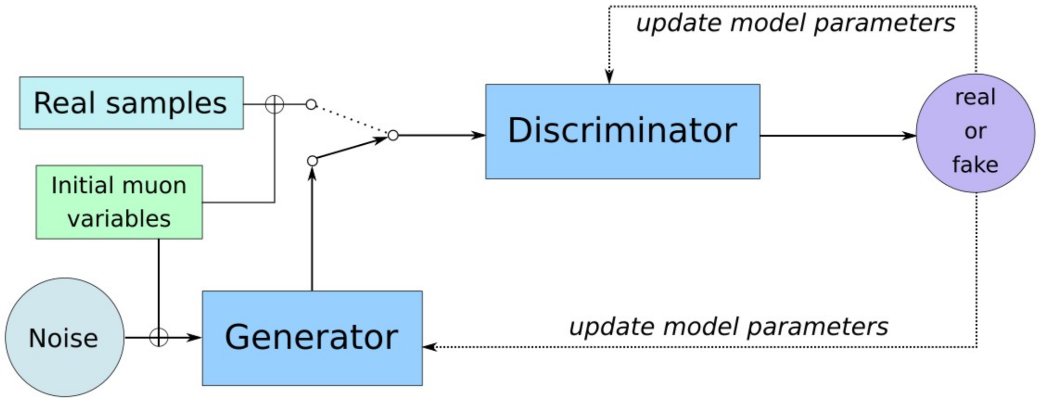
2. **Conditional GAN**: trained with multiple thickness values.
   → Check tuned generation and interpolation capabilities.

Results: Simple GAN

Simple GAN:
Trained with samples from thickness 16 mm pipes.

→ **Input** of generator:
Latent noise.
First detector variables (x, y, vx, vy).

→ **Output** of generator:
Shift in position and direction of muon trajectory (i.e. $\Delta x^*, \Delta y^*, \Delta v_x, \Delta v_y$).



<u>Result of the optimization</u>

- Loss function: Mean Squared Error → more info
- Architecture (G): 512, 256, 256, 128, 64, 16 LeakyReLU
- Latent space dimension: 64
- Optimizer: Adam, 0.001 (halves every 50 epochs)
- Trained for 200 epochs
- No use of Dropout layers → instability

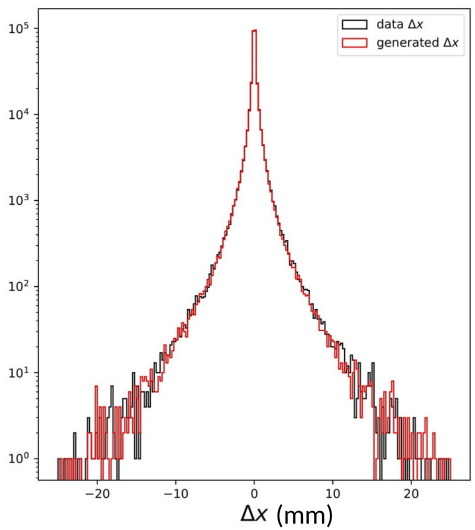
Results: Simple GAN

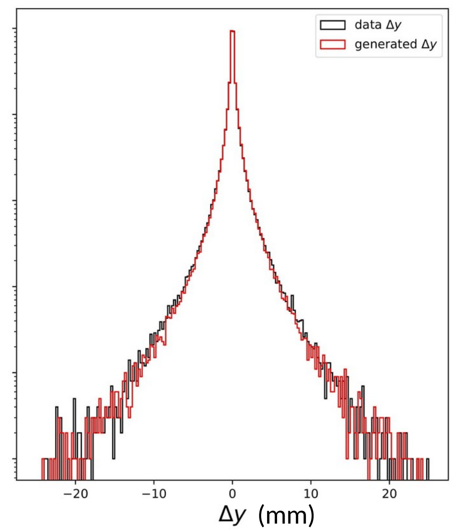
Results of **real** and **generated** variable distributions.

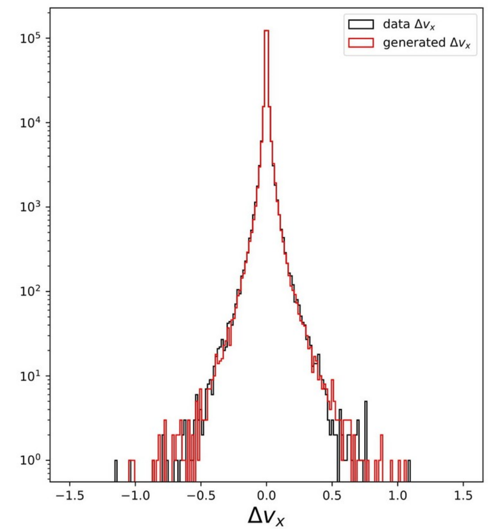
We observe that the generator is capable to produce a set of samples that resembles the original distributions of the independent variables.

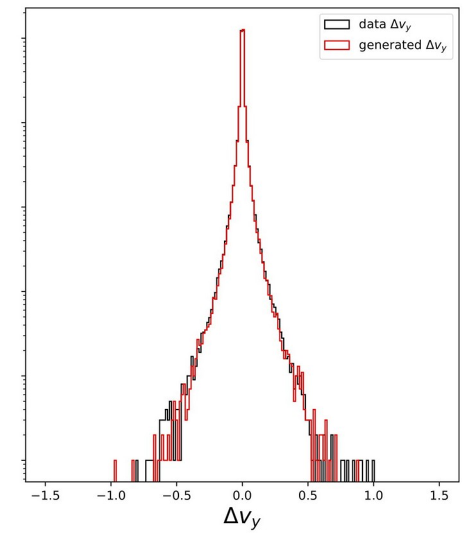
	Variable	Real samples	Generated samples
Mean	Δx^*	$5.9 \cdot 10^{-5}$	$-6.0 \cdot 10^{-3}$
	Δy^*	$-1.2 \cdot 10^{-3}$	$4.2 \cdot 10^{-3}$
	Δv_x	$3.4 \cdot 10^{-5}$	$2.3 \cdot 10^{-4}$
	Δv_y	$1.5 \cdot 10^{-5}$	$-8.0 \cdot 10^{-5}$
Skewness	Δx^*	1.10	0.55
	Δy^*	0.10	0.60
	Δv_x	-0.05	0.56
	Δv_y	0.24	-0.66

Table 5.1: Mean and skewness.









Results: Simple GAN

But we are interested in generating the global 4D distribution.

Qualitatively we observe that the model is able to **capture the correlations** between the variables:

→ Sample-to-sample generation is good
→ Properties of global 4D distribution are preserved.

Real samples

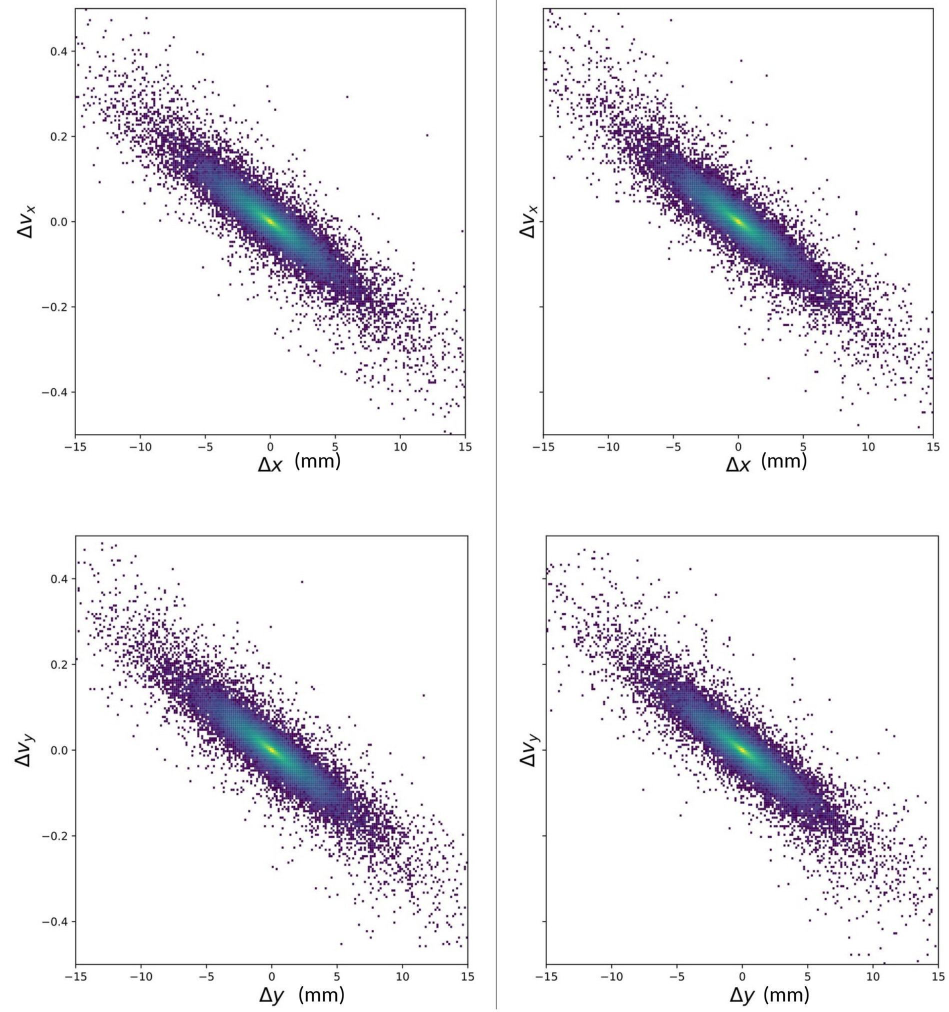
	Δx^*	Δy^*	Δv_x	Δv_y
Δx^*	1.71	$-6.98 \cdot 10^{-2}$	$-3.60 \cdot 10^{-2}$	$6.38 \cdot 10^{-4}$
Δy^*		1.70	$9.18 \cdot 10^{-4}$	$-3.43 \cdot 10^{-2}$
Δv_x			$9.82 \cdot 10^{-4}$	$-9.60 \cdot 10^{-6}$
Δv_y				$9.34 \cdot 10^{-4}$

Generated samples

	Δx^*	Δy^*	Δv_x	Δv_y
Δx^*	1.48	$-0.34 \cdot 10^{-2}$	$-3.47 \cdot 10^{-2}$	$1.28 \cdot 10^{-4}$
Δy^*		1.35	$0.63 \cdot 10^{-4}$	$-3.14 \cdot 10^{-2}$
Δv_x			$9.65 \cdot 10^{-4}$	$-3.90 \cdot 10^{-6}$
Δv_y				$8.74 \cdot 10^{-4}$

Table 5.2: Covariance matrices of real and generated samples.

Real events | Generated events



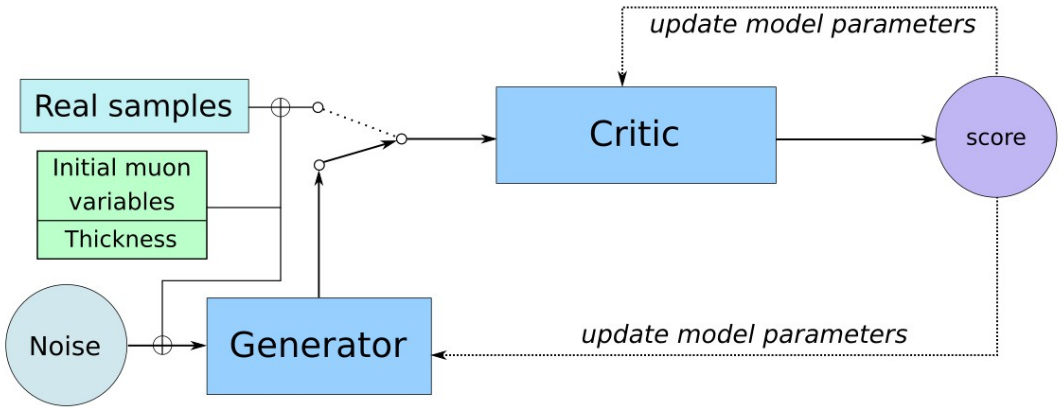
Results: Conditional GAN

Conditional GAN:
Trained with samples of various thickness of the pipes.

→ **Input** of generator:
- Latent noise.
- First detector variables (x, y, vx, vy).
- Thickness of the pipe (as a label)

→ **Output** of generator:
- Shift in position and direction of muon trajectory (i.e. $\Delta x^*, \Delta y^*, \Delta v_x, \Delta v_y$).

For training we have used 4, 6, 8, 10, 14, 16, 18, 20 mm labels (12 mm reserved to test interpolation)



Result of the optimization

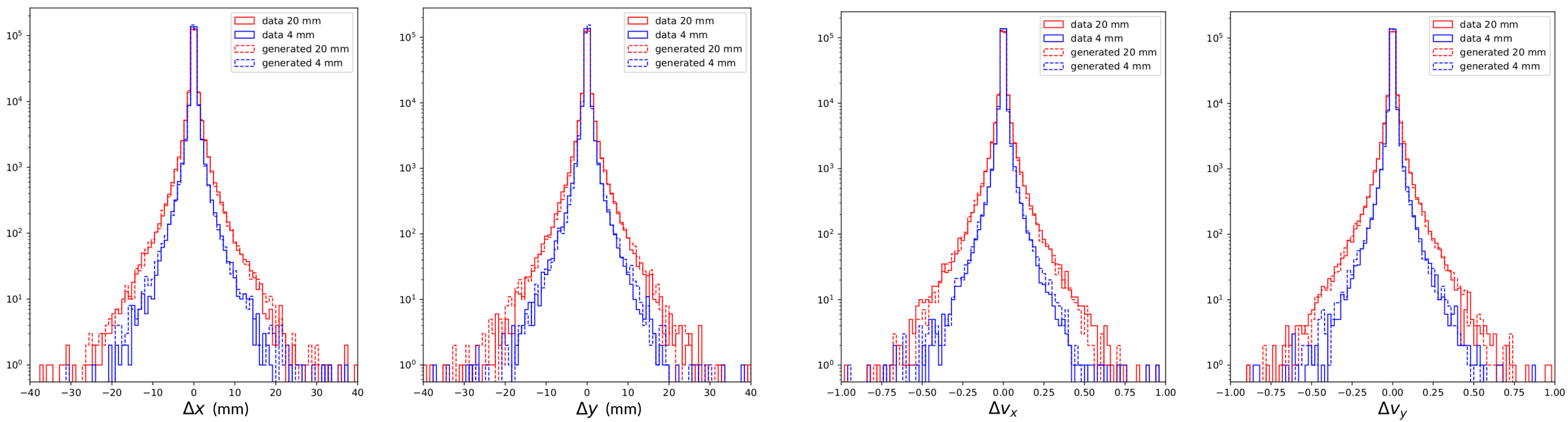
- WGAN-GP framework → critic + loss function
  → more stability
- Architecture (G): 32, 64, 128 LeakyReLU
- Latent space dimension: 16
- Optimizer: Adam, 0.0001
- Trained for 1000 epochs
- No use of Dropout layers → instability

Results: Conditional GAN

Results of real (—) and generated (----) distributions conditioned on the thickness label.

2 labels represented: 4 mm, 20 mm

We observe the model is able to modulate the generation.
Also it reproduces well the correlation between variables (although not shown)



Results: Conditional GAN

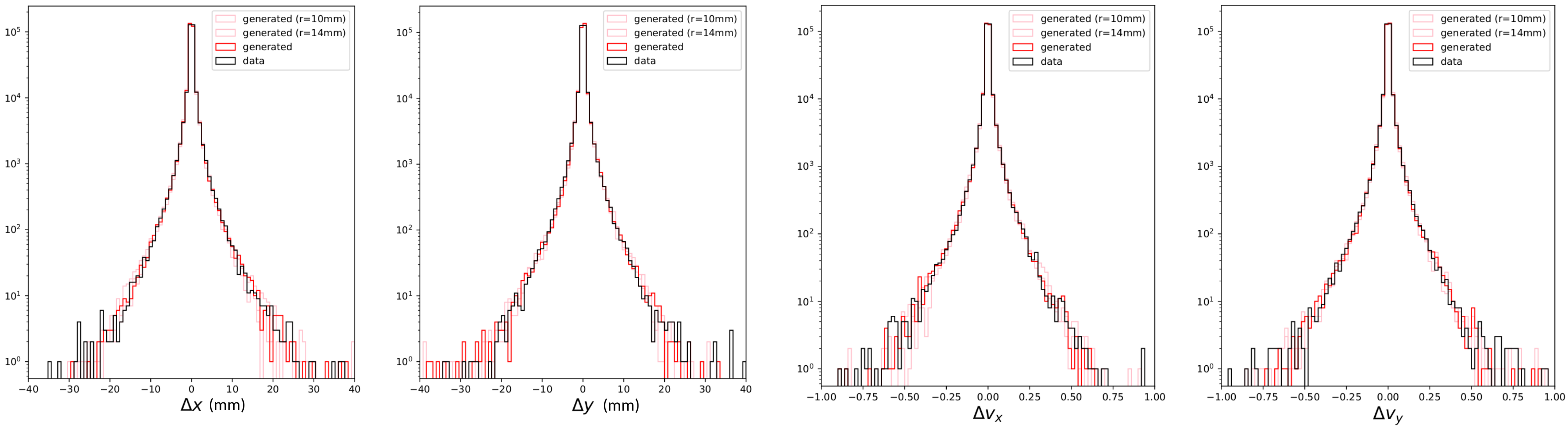
→ We ask the model to generate samples of a 12 mm pipe (never learned).

→ Using interpolated label between 10 and 14 mm.

Qualitatively, we see that the model is also able to produce muon samples that match the real distributions.

	Variable	Real samples	Generated samples
Mean	Δx^*	$-6.1 \cdot 10^{-4}$	$-1.4 \cdot 10^{-2}$
	Δy^*	$2.7 \cdot 10^{-3}$	$1.8 \cdot 10^{-2}$
	Δv_x	$2.5 \cdot 10^{-5}$	$-2.2 \cdot 10^{-4}$
	Δv_y	$-3.9 \cdot 10^{-5}$	$-7.0 \cdot 10^{-6}$
Skewness	Δx^*	-2.84	0.76
	Δy^*	4.82	-0.34
	Δv_x	0.01	-0.70
	Δv_y	0.13	0.02

Table 5.3: Mean and skewness results for $r = 12$ mm data.



Computational gain

→ Note on computational speed: time taken to produce 10000 events.

- → Geant4 (just the propagation simulation part): ~**37 seconds**
- → GAN model: ~**0.7 seconds**

With a properly trained GAN we can produce in the same time, about **50 times** more simulation data than with Geant4.

Drawbacks:

- → Fine tuning of the parameters.
- → Training on GPU: 2-3 hours.

Conclusions

→ GAN models can provide an excellent tool for producing simulation data in the context of muon tomography.

→ Very flexible solution → modulation and interpolation capabilities.

→ Computationally faster and less expensive solution (~50 times).

→ Further development:

  → Simulate real detector effects.

  → Explore new generative models: diffussion, normalizing flows…

Thank you for your attention !!

Rubén López Ruiz
lopezr@ifca.unican.es

IFCA

UNIVERSIDAD DE CANTABRIA

Back up

Wasserstein distance

Wasserstein Distance between real distributions for each pipe thickness value