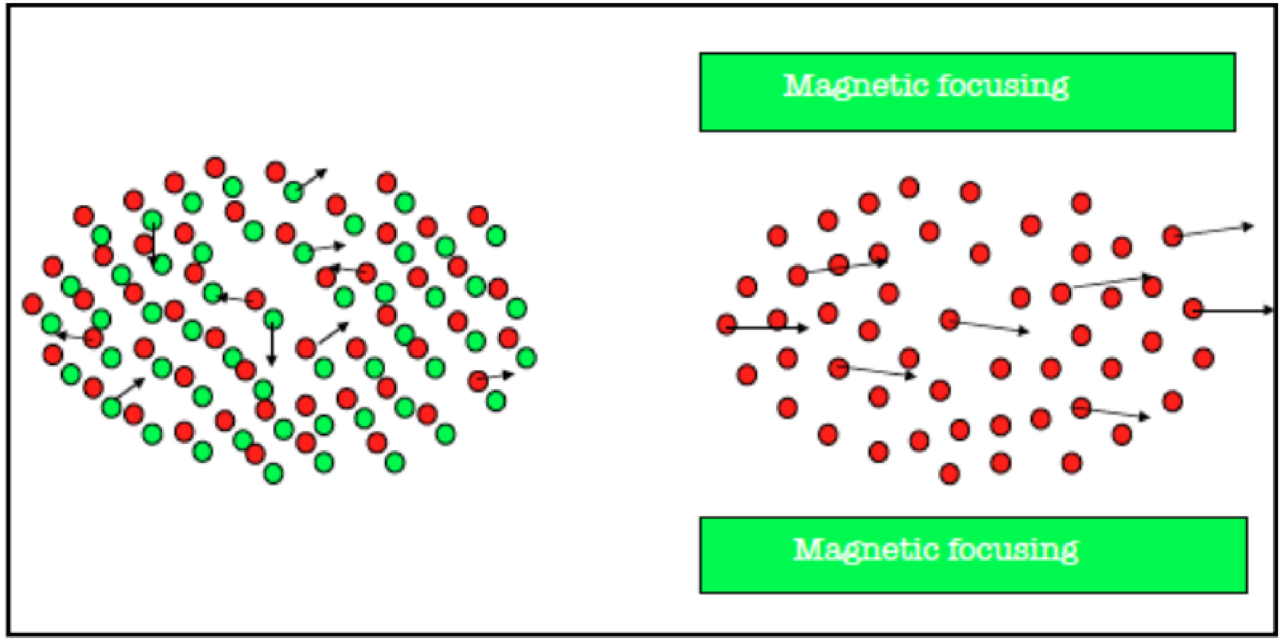
ADVANCING NON-LINEAR SPACE-CHARGE

Isabella Vojskovic

SOME BACKGROUND



Space-Charge

External Fields

Forces on Particles

$$\vec{F} = \frac{d\vec{p}}{dt} = m\vec{a} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

Propagating Particles

SC-kick, Magnet-kick, **SC-kick,** Magnet-kick etc...

Easy if distribution is such that the Space - charge force is assumed to be linear.

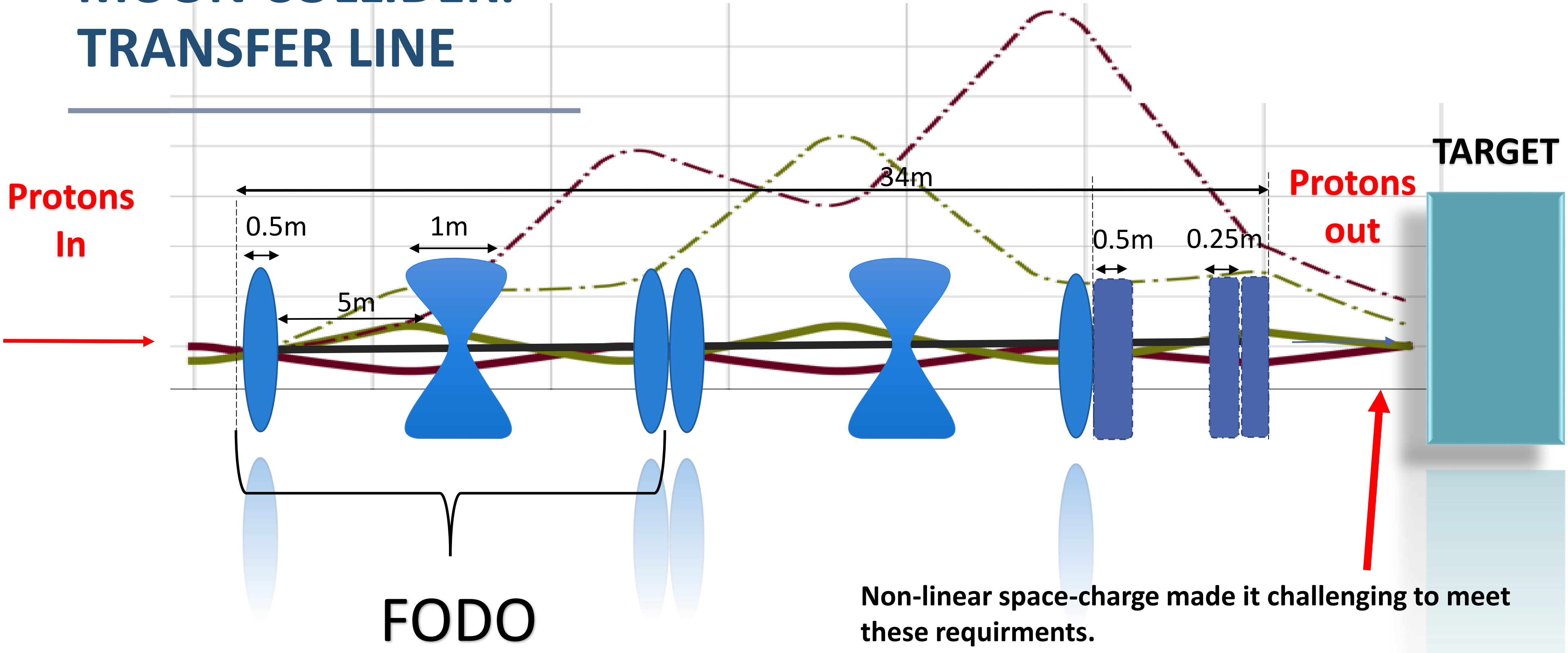
Otherwise.. It is computationally expensive

6D – state vector

$$\vec{x} = (x, p_x, y, p_y, z, \delta)$$

Phase-Space

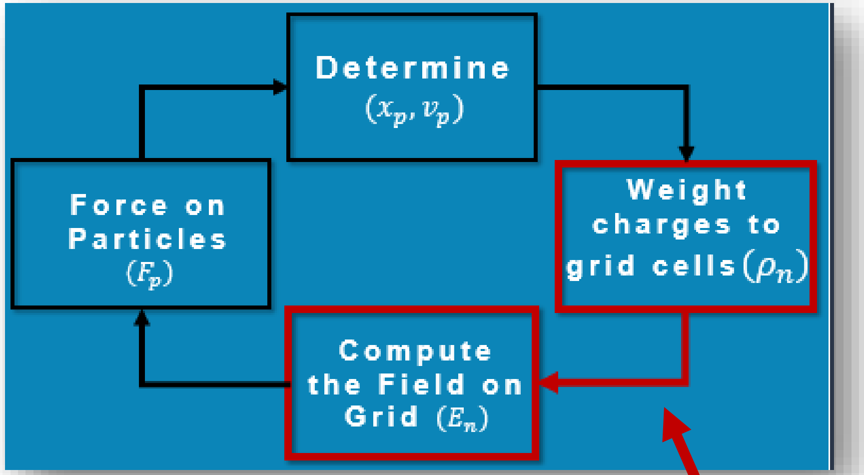
MUON COLLIDER: TRANSFER LINE



Non-linear space-charge made it challenging to meet these requirments.
Global optimization algorithms took forever to run!

HOW IS NON-LINEAR SPACE-CHARGE MODELLED?

PIC



The slow parts!

The most common way is to use:
Particle In Cell (PIC).

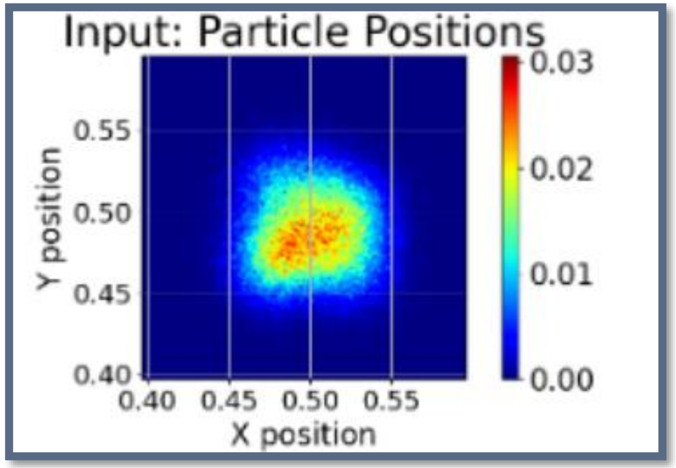
A Numerical way to Solve the Poisson Equation – Second-Order PDE.

$$\nabla \cdot E = -\nabla^2 \Phi = \frac{\rho}{\epsilon_0}$$

$$\nabla^2 \Phi = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2}$$

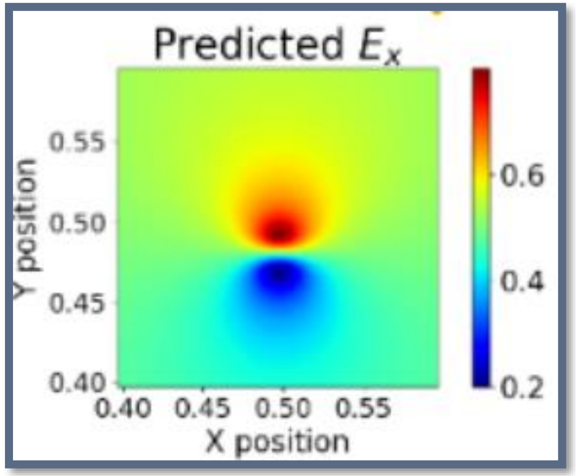
SUBSTITUTE COMPUTATIONALLY EXPENSIVE PARTS WITH NEURAL NETWORKS

1. Bin the particle position and use as input to the network.



Input

2. Calculate the corresponding the electric field of the distribution using PIC



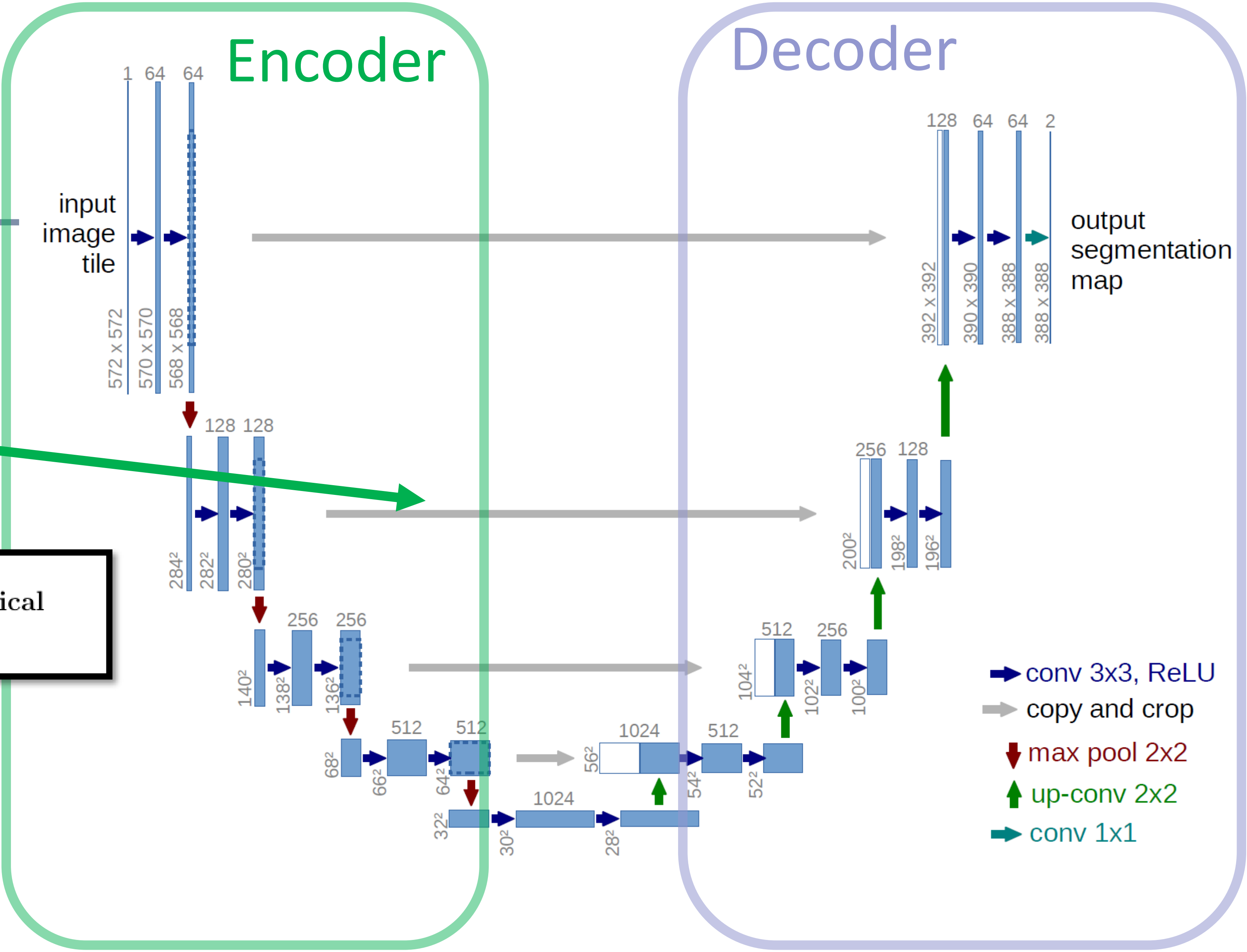
Output

U-NET

Low level details are passed to high level representation.

U-Net: Convolutional Networks for Biomedical Image Segmentation

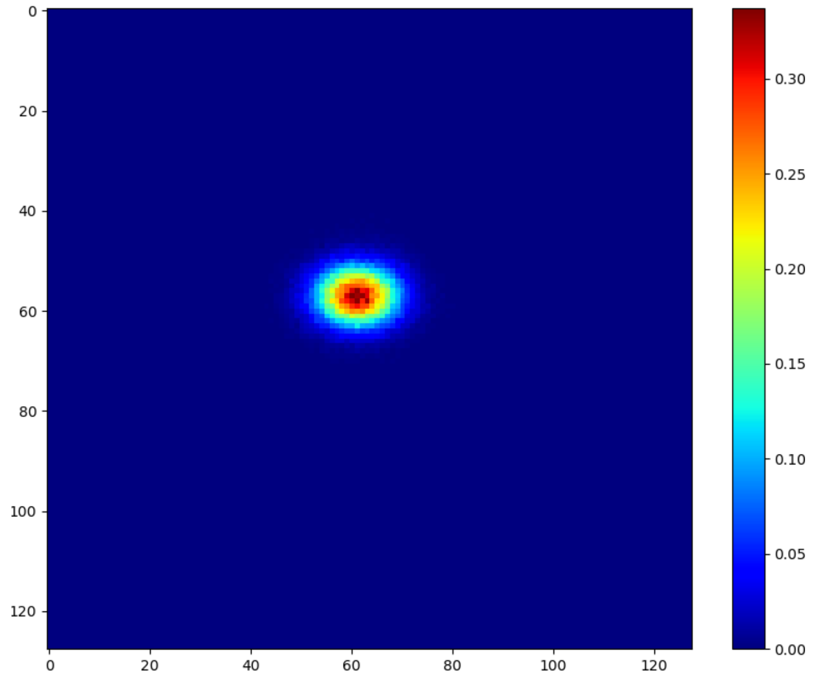
av O Ronneberger · 2015 · Citerat av 85638



2D –CASE INPUT DATA

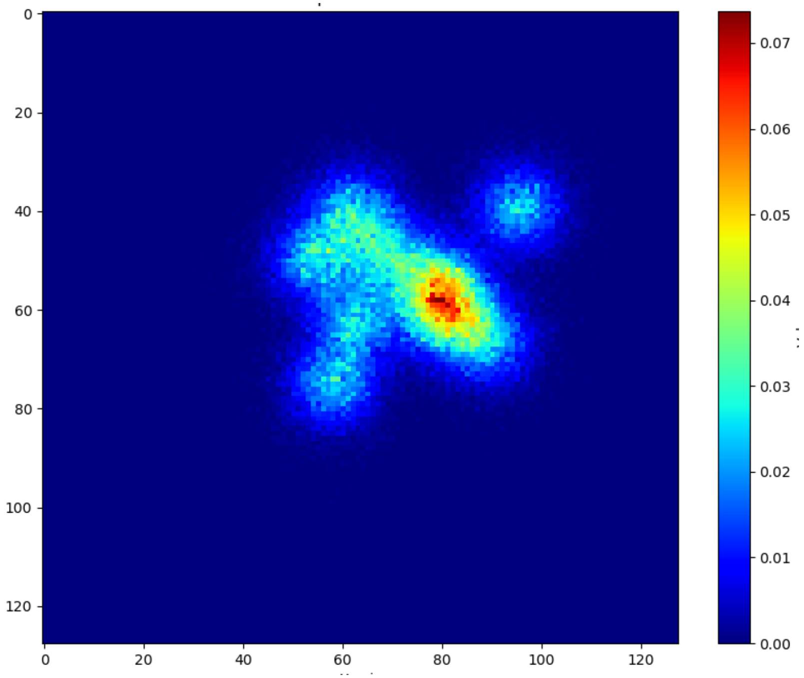
100.000 particles

y



x

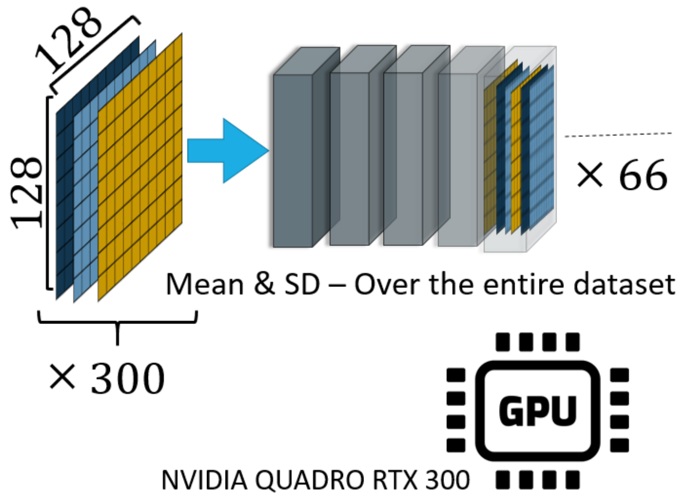
+



Grid size: 128x128

DATA &TRAINING

INPUT +OUTPUT



Library: TensorFlow

Activation Function: Leaky ReLU

Final Activation Function: Sigmoid Function

Loss Function: MSE or Logcosh

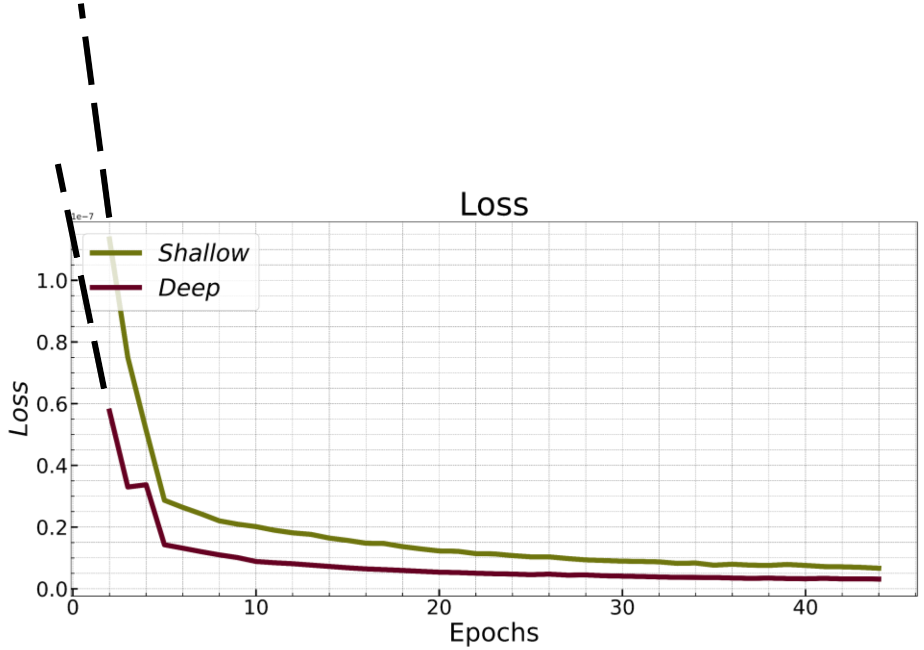
Initial Learning Rate: 0.001

Optimization: Adaptive stochastic gradient descent

Dataset Size: Split into multiple sets:
1500 matrices per training

Batch Size: Normally 4 or 16 (2^2, 4^2 for optimal GPU)

Epochs: 8 or 12

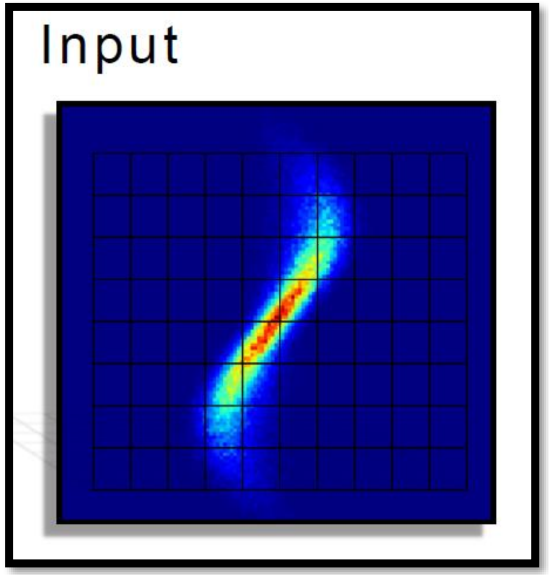


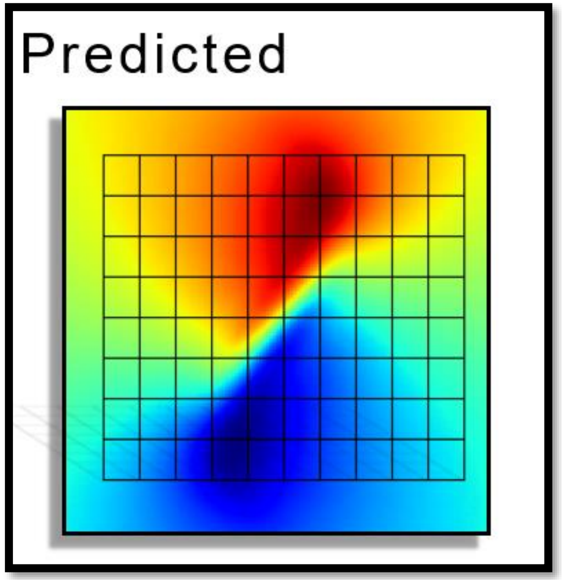
After a few minutes of training the model is capable to achieve a relative error of 0.5% on the validation set.

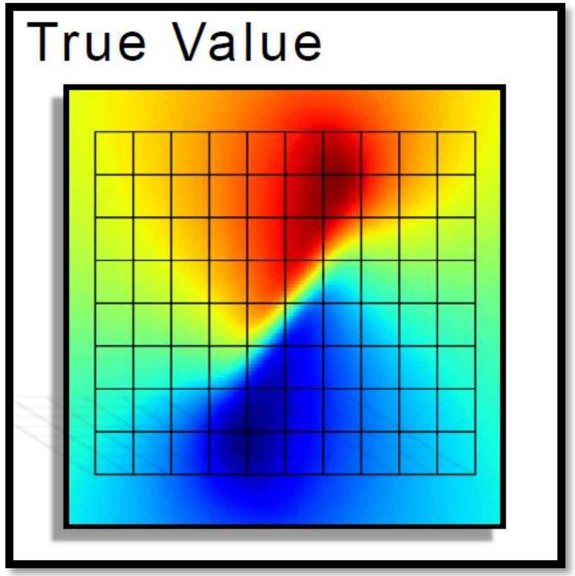
RESULTS

The model is capable to achieve a relative of 1% on a non-seen dataset

Time taken: 0.03s on CPU



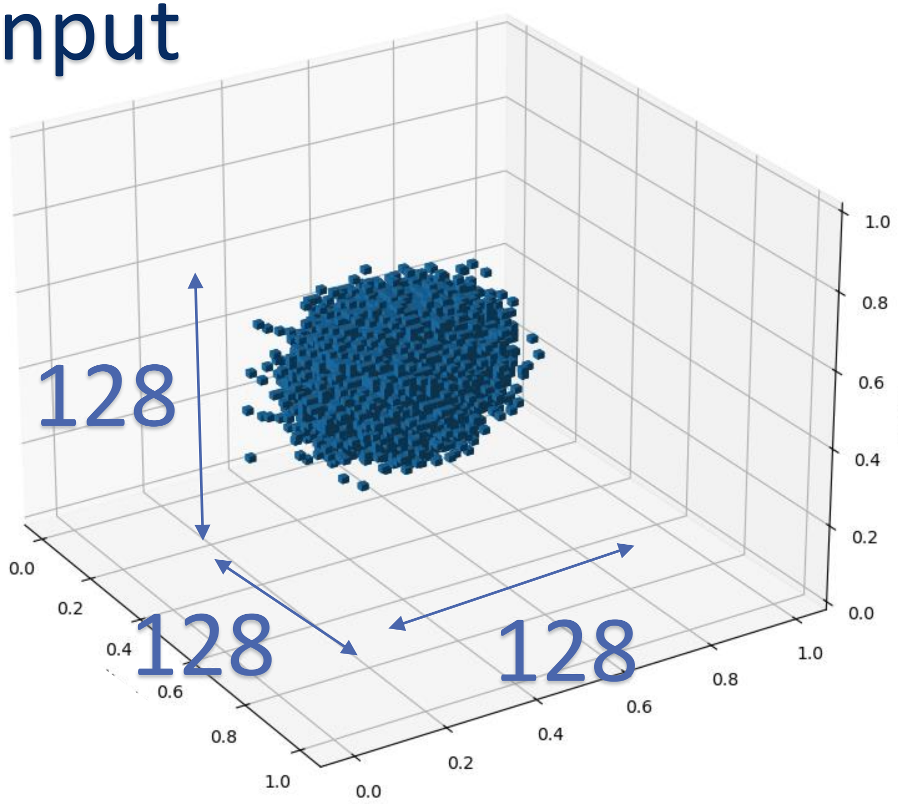




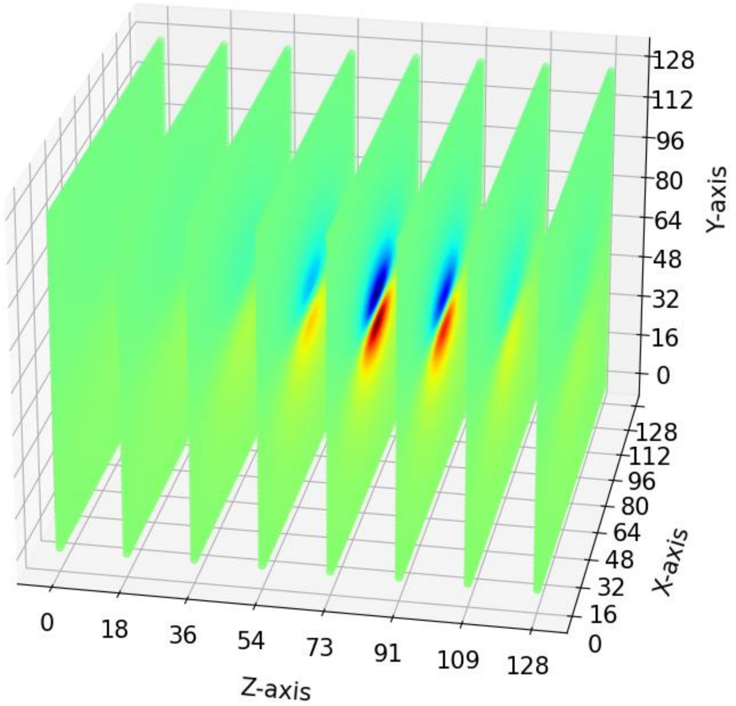
3D CONVOLUTIONS

Max 0.7% error across all slices in 10 randomly sampled batches

Input



Output (One Plane)



Nice results!
But more computationally expensive:

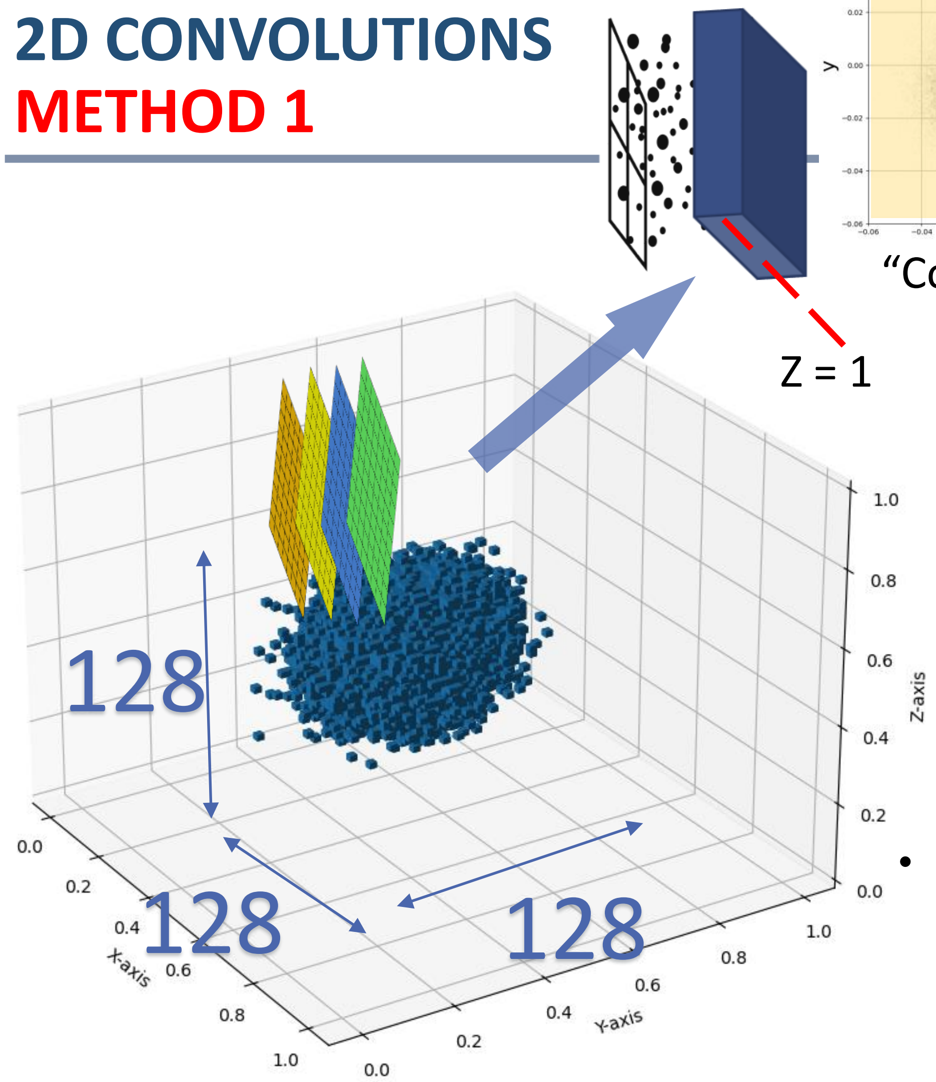
k^3 vs. k^2

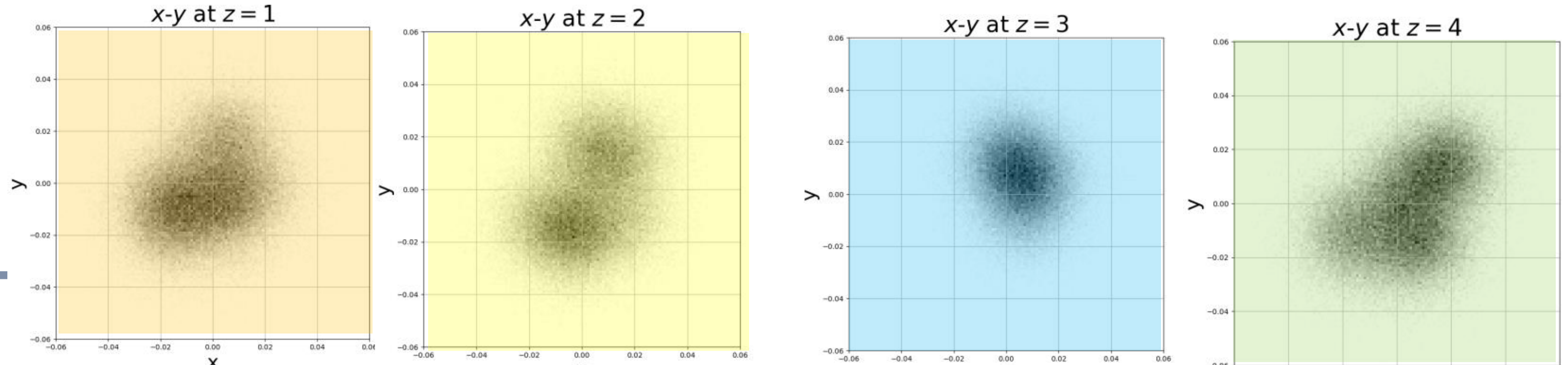
Kernel size

SPEED UP

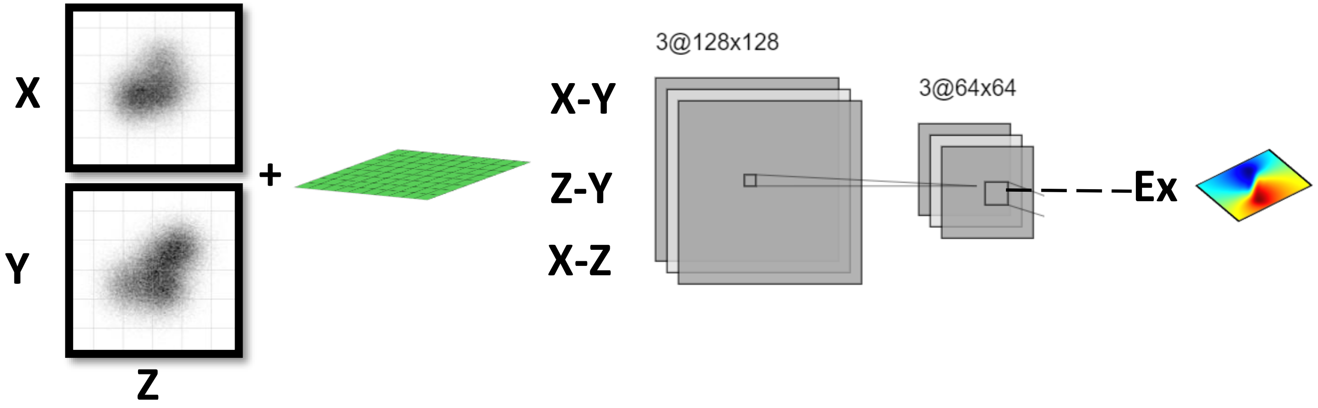
Model	Time
TraceWin (PIC simulation)	8s (3D) 2s (2D)
2D Conv	0.05s
3D Conv	0.5s
2D Conv 128 images	0.1s

3D MODELLING USING 2D CONVOLUTIONS
METHOD 1





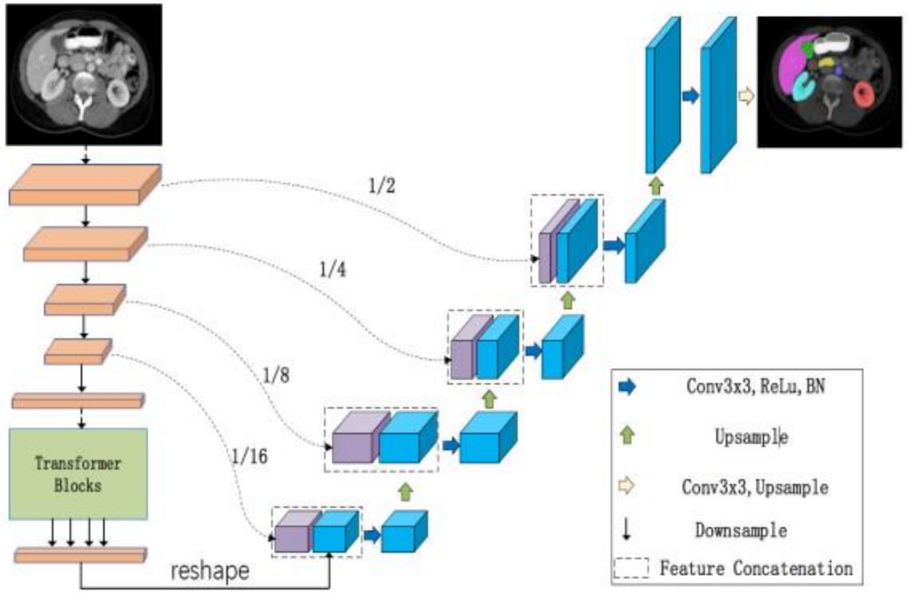
“Colors” gives information about the z position



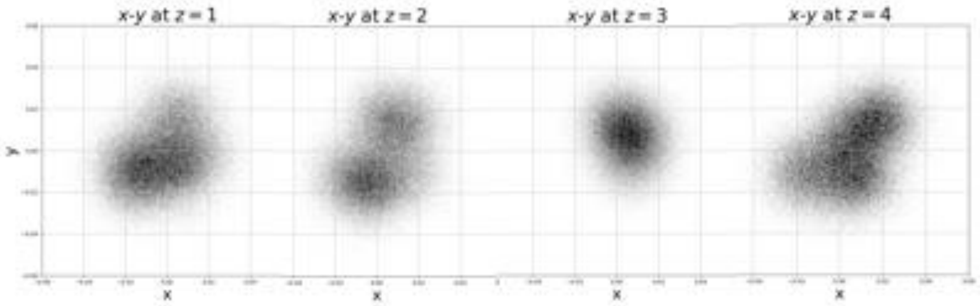
- Processing channels in groups, and then concatenating and processed together for better performance. (Used Pytorch for this)

3D MODELLING USING 2D CONVOLUTIONS
METHOD 2

LEVIT-UNET: MAKE FASTER ENCODERS WITH TRANSFORMER FOR MEDICAL IMAGE SEGMENTATION



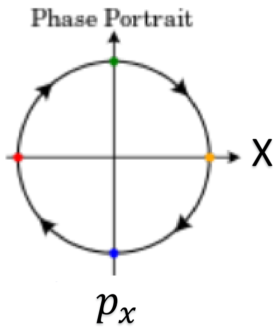
Vision Transformers



ENERGY CONSERVATION - SYMPLECTICITY

Energy Conserving Pendulum Give Rise to Conserved Phase-Space Area

A pendulum's dynamics in phase-space looks like:

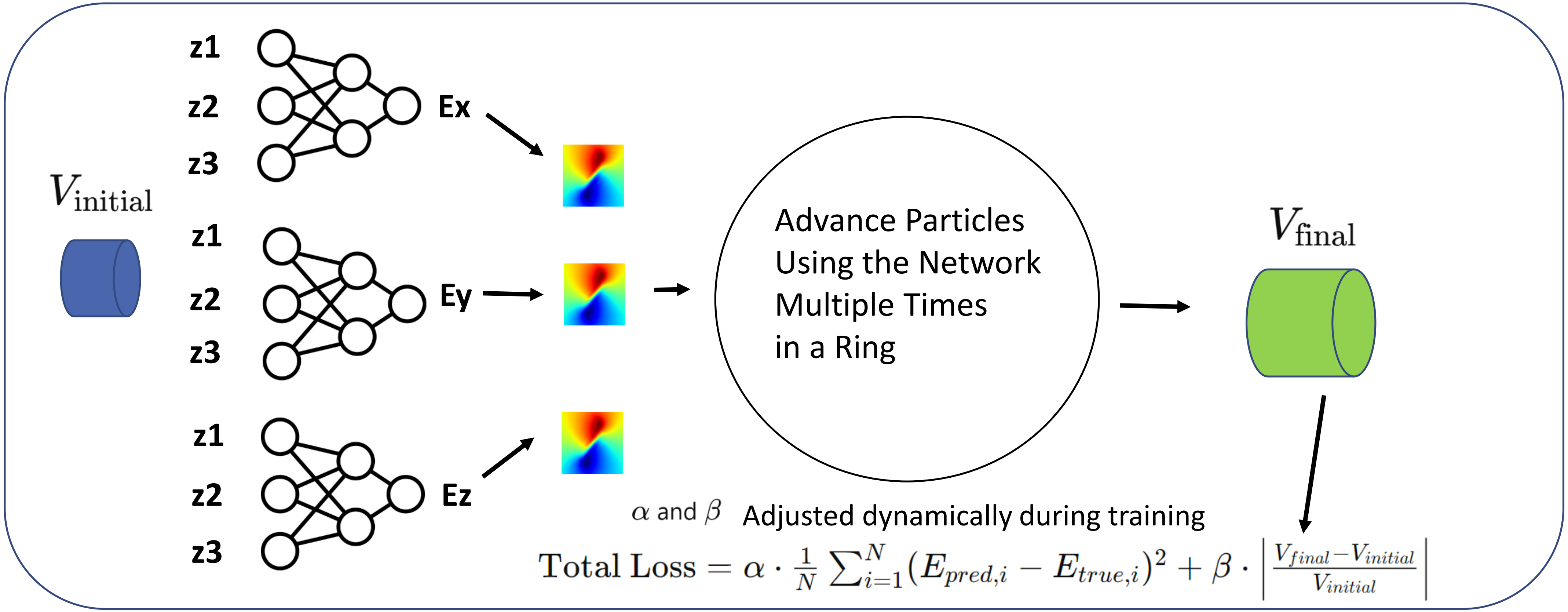


Energy-Conserving: An Inherent Property of Hamiltonian Dynamics

We use the Hamilton's canonical equations of motion.
Not Newton's Equations.

$$\frac{d\mathbf{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \qquad E_{tot} = \mathcal{H}(\mathbf{q}, \mathbf{p})$$

INCORPORATE PHYSICS



THANK YOU

Contact Info:

Isabella Vojskovic

+46 (76)166 7309

Isabella.vojskovic@ess.eu