WMLQ2024 04/06/2024

Deep Generative Models for Particle Simulations at ALICE, CERN

Patryk Będkowski*, Karol Rogoziński*, Mikołaj Kita*

Jan Dubiński, Tomasz Trzciński, Przemysław Rokita, Kamil Deja

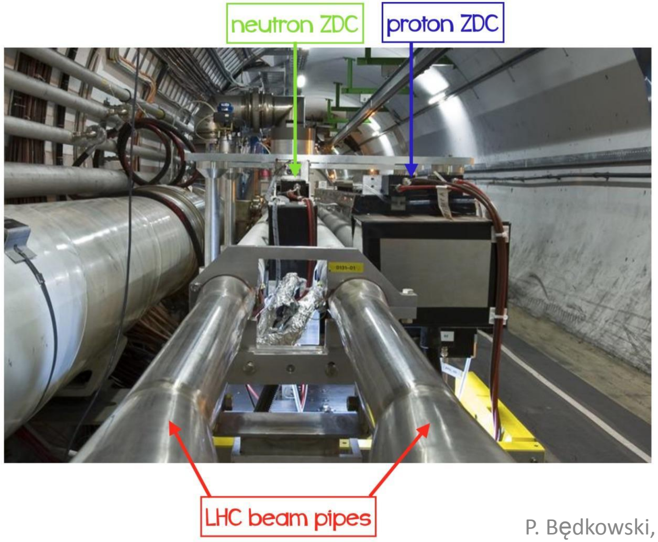
Warsaw University of Technology

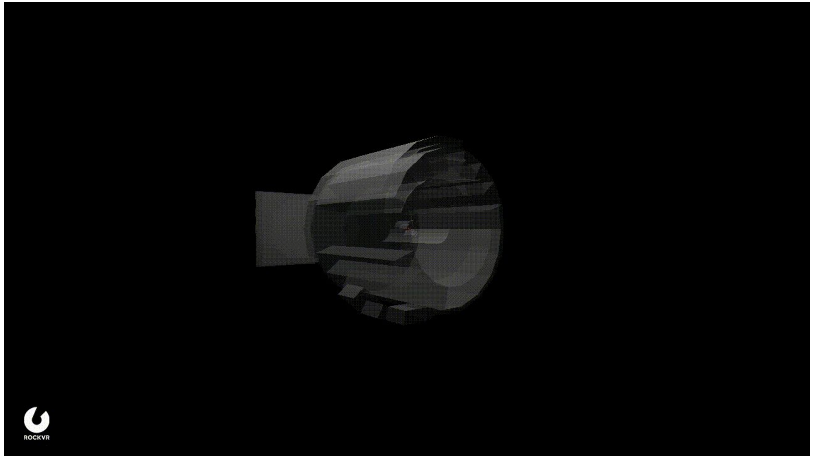
Content

Introduction to Zero Degree Calorimeter (1/2)

The Zero Degree Calorimeter is one of the components of the A Large Ion Collider Experiment (ALICE) detector, part of the Large Hadron Collider.

The ZDC calorimeter records the energy of the 'observer' particles through the phenomenon of Cherenkov radiation.





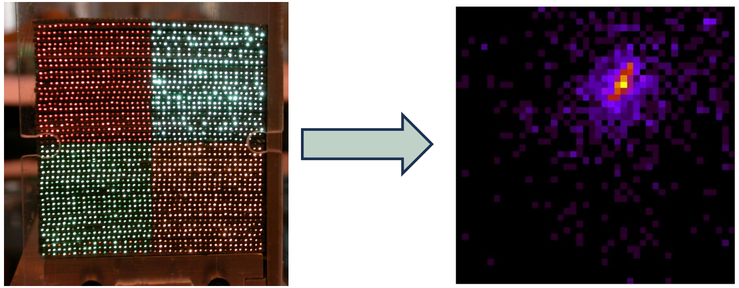
Introduction to Zero Degree Calorimeter (2/2)

The ZDC calorimeter consists of two devices named Neutron and Proton,

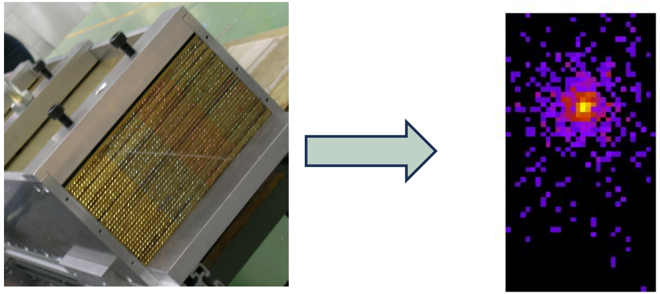
The principle of operation is based on the detection of photons produced by particles falling into optical fibres,

Photodetectors record the energy of photons in the form of one-dimensional images,

The recorded image is produced in response to a particle described by 9 conditional attributes (Mass, energy, charge, 3 position components and 3 momentum components of the particle during the collision).



ZDC Neutron



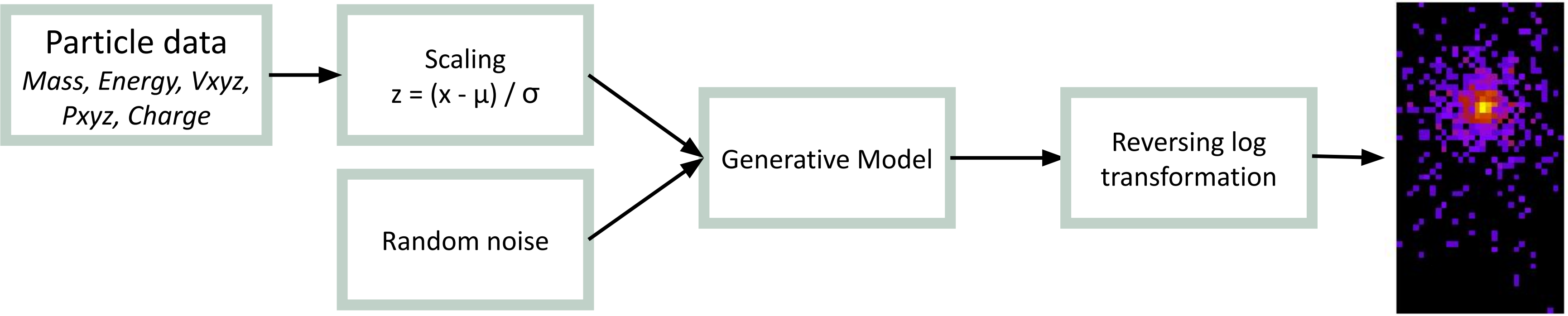
ZDC Proton

How we apply Deep Generative models in ZDC

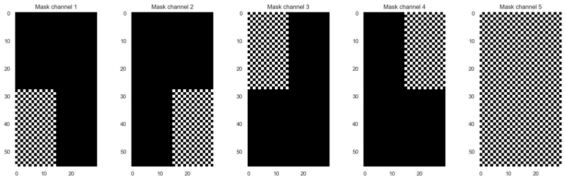
- Current simulations based on Monte Carlo techniques represent a high computational effort,

- Generative machine learning models allow the simulation of images formed by collisions of particles omitting the complex physical interactions between them.

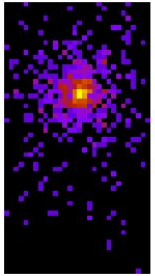
Purpose of work: Extend standard GAN architecture with regularization to perform simulations of the ZDC Proton calorimeter.



Validation - calculation of channels



True

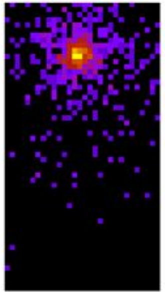


Sum pixels across channels

48, 21, 211, 92, 396

Gather distribution

Generated



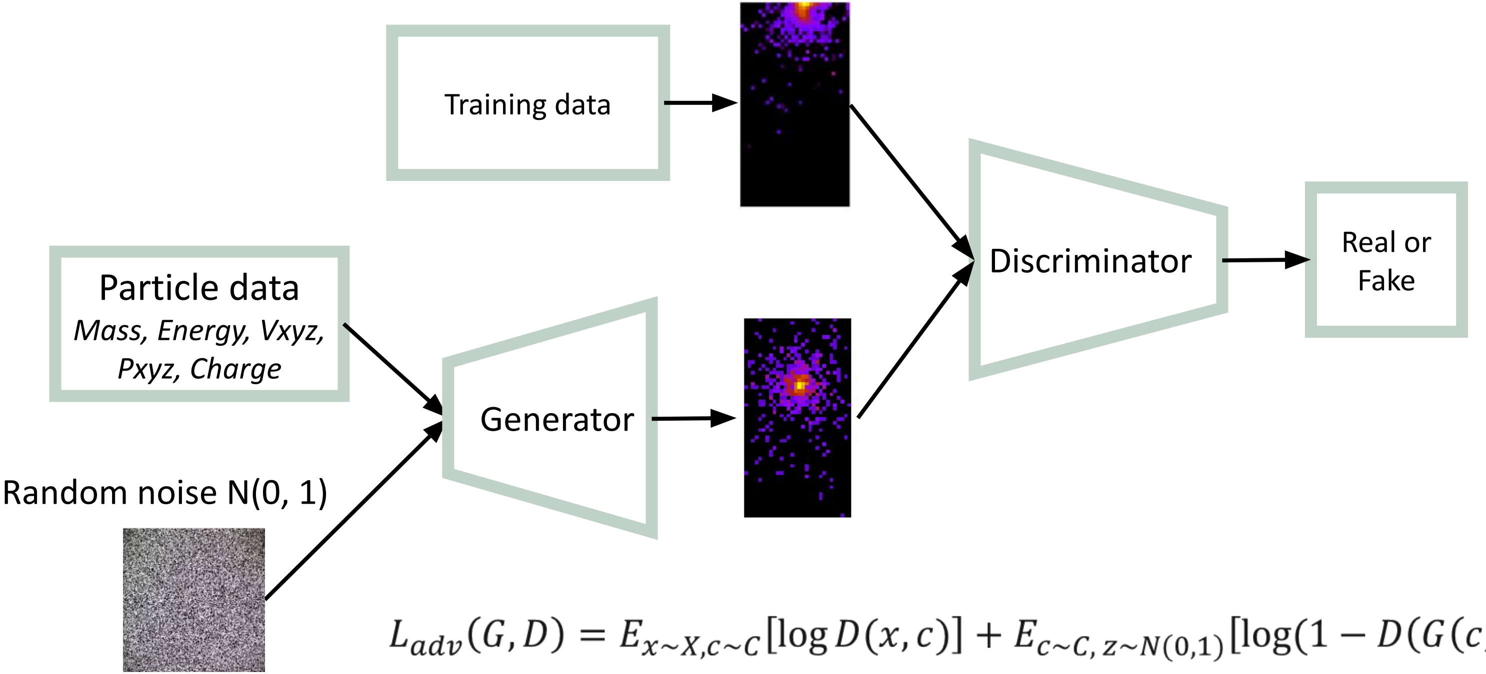
Sum pixels across channels

54, 16, 251, 94, 371

Compare channels distribution with Wasserstein distance

Improving
Generative Adversarial Networks

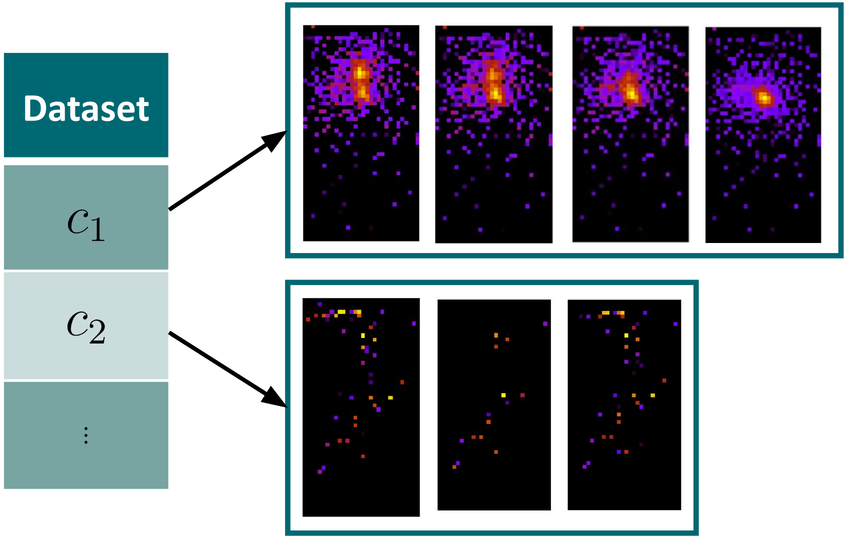
Generative Adversarial Network



$$L_{adv}(G,D) = E_{x\sim X, c\sim C}[\log D(x,c)] + E_{c\sim C,\, z\sim N(0,1)}[\log(1 - D(G(c,z),c))]$$

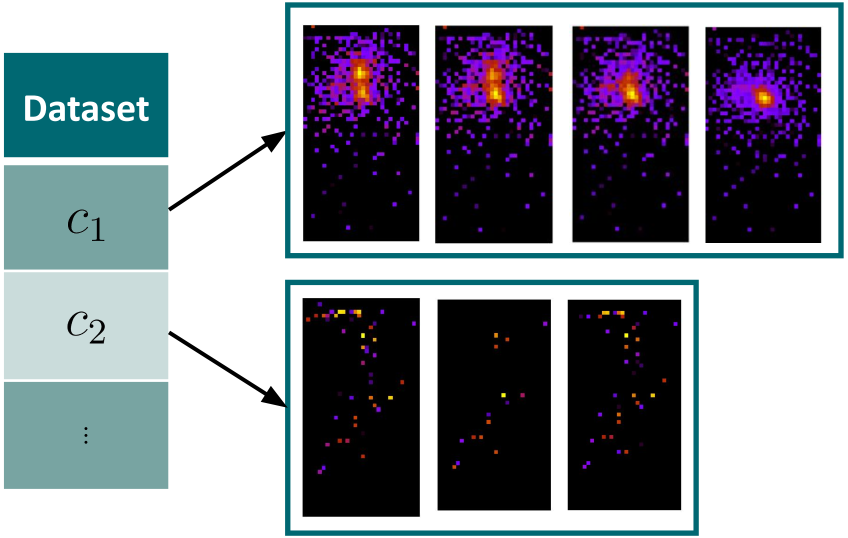
Diversity regularization - observation

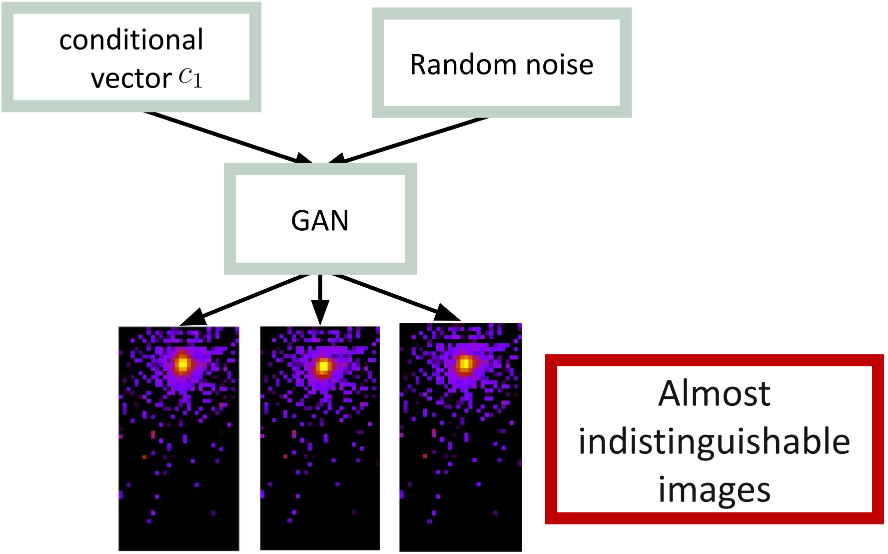
Problem: Vector c corresponds to diverse set of images in the dataset.



Diversity regularization

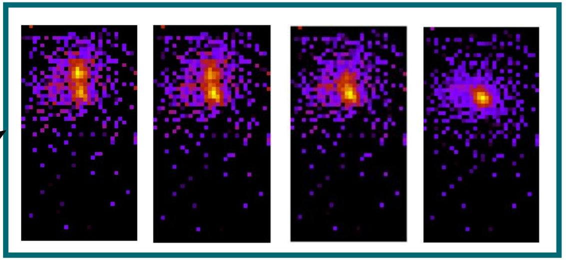
Problem: Vector c corresponds to diverse set of images in the dataset.
Trained standard GAN generates consistent images!

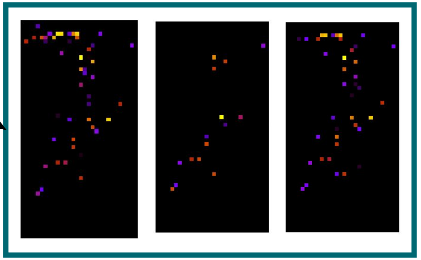




SDI-GAN – introduces diversity measure

Dataset
c_1
c_2
⋮



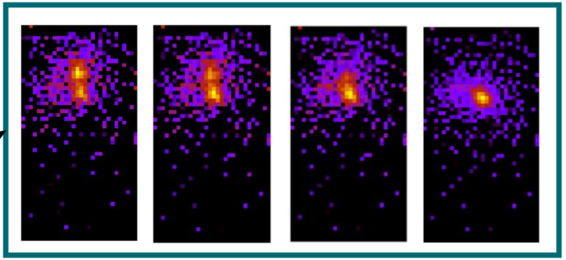


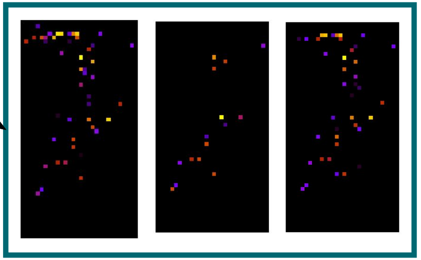
$f_{div}(c)$ → *normalize* → c_1, c_2, …

Diversity measure
0.64201
0.24086
⋮

SDI-GAN – introduces diversity measure

Dataset
c_1
c_2
⋮





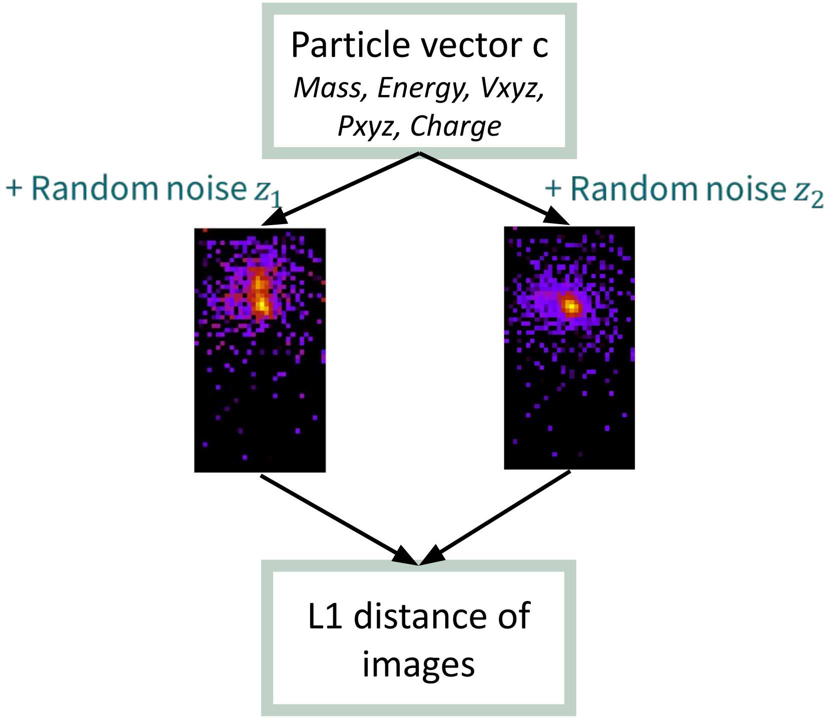
Variance across images for c:

$$f_{div}(c) = \sum_{i,j} \sqrt{\frac{\sum_t \left(x_{ij}^t - \mu_{ij}\right)^2}{|X|}}$$

$f_{div}(c)$ → *normalize* → (c_1, c_2, …)

Diversity measure
0.64201
0.24086
⋮

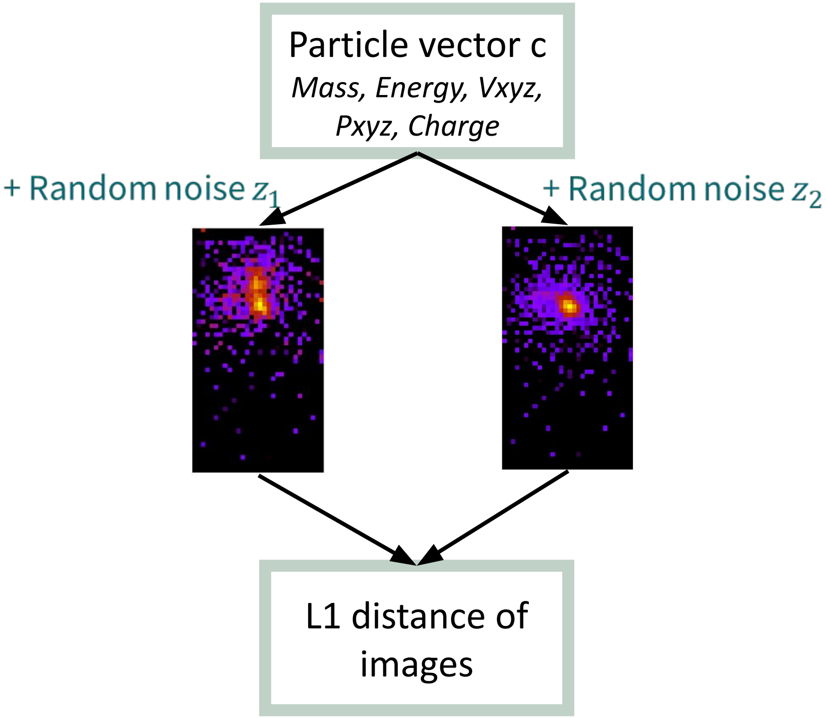
Diversity regularization – training



$$L(G,D) = L_{adv}(G,D) + \lambda_{div} L_{div}(G)$$

$$L_{div}(G) = f_{div}(c) * \left(\frac{d_I\big(G(c,z_1), G(c,z_2)\big)}{d_z(z_1,z_2)} \right)^{-1}$$

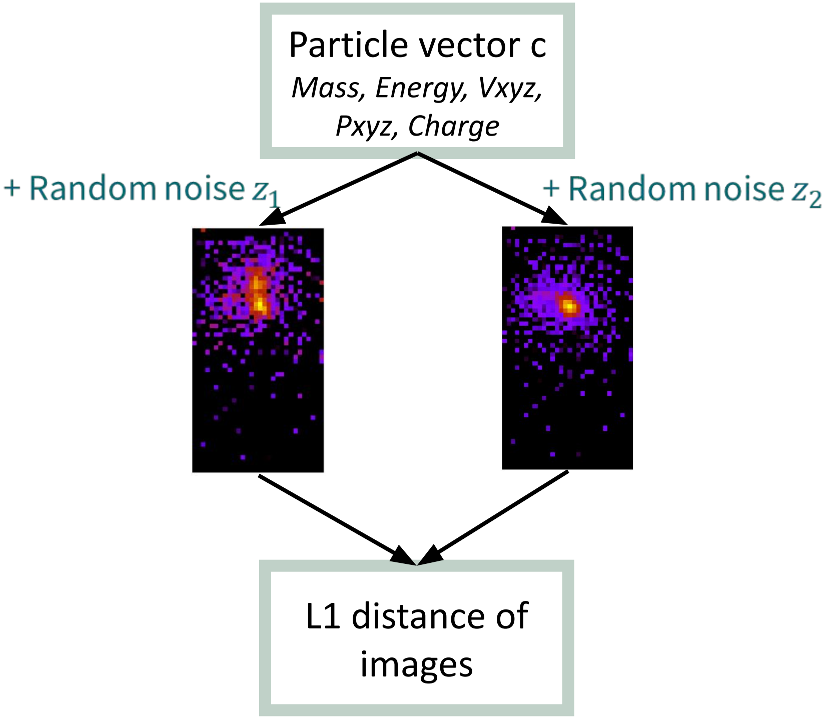
Diversity regularization – training



$$L(G,D) = L_{adv}(G,D) + \lambda_{div} L_{div}(G)$$

$$L_{div}(G) = f_{div}(c) * \left(\frac{d_I\big(G(c,z_1), G(c,z_2)\big)}{d_z(z_1,z_2)} \right)^{-1}$$

Diversity regularization – training



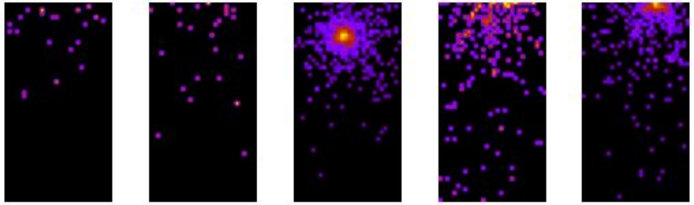
$$L(G,D) = L_{adv}(G,D) + \lambda_{div} L_{div}(G)$$

$$L_{div}(G) = f_{div}(c) * \left(\frac{d_I(G(c,z_1), G(c,z_2))}{d_z(z_1,z_2)} \right)^{-1}$$

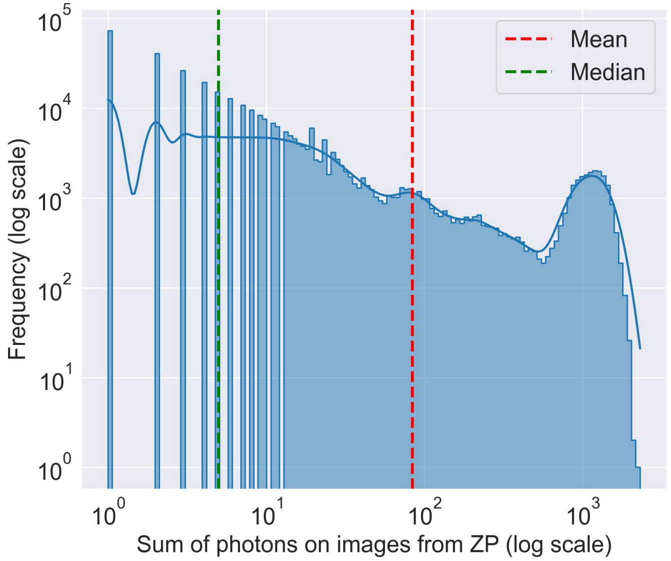
$f_{div}(c)$	**0.64201**	**0.24086**	**0.002**

Intensity regularization

Problem: Wide sum of pixel values across images within the dataset.



$$f_{in}(x) = \sum_{i,j} x_{ij}$$



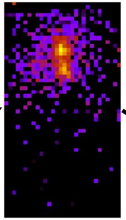
Intensity: $L_{in} = |f_{in}(x_c) - f_{in}(\hat{x}_c)|$

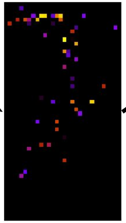
Modified loss function: $L = L_{adv}(G, D) + \lambda_{div} L_{div}(G) + \lambda_{in} L_{in}(G)$

Auxiliary regressor

Idea: We can control geometric properties of the generated image.

Dataset
c_1
c_2
⋮



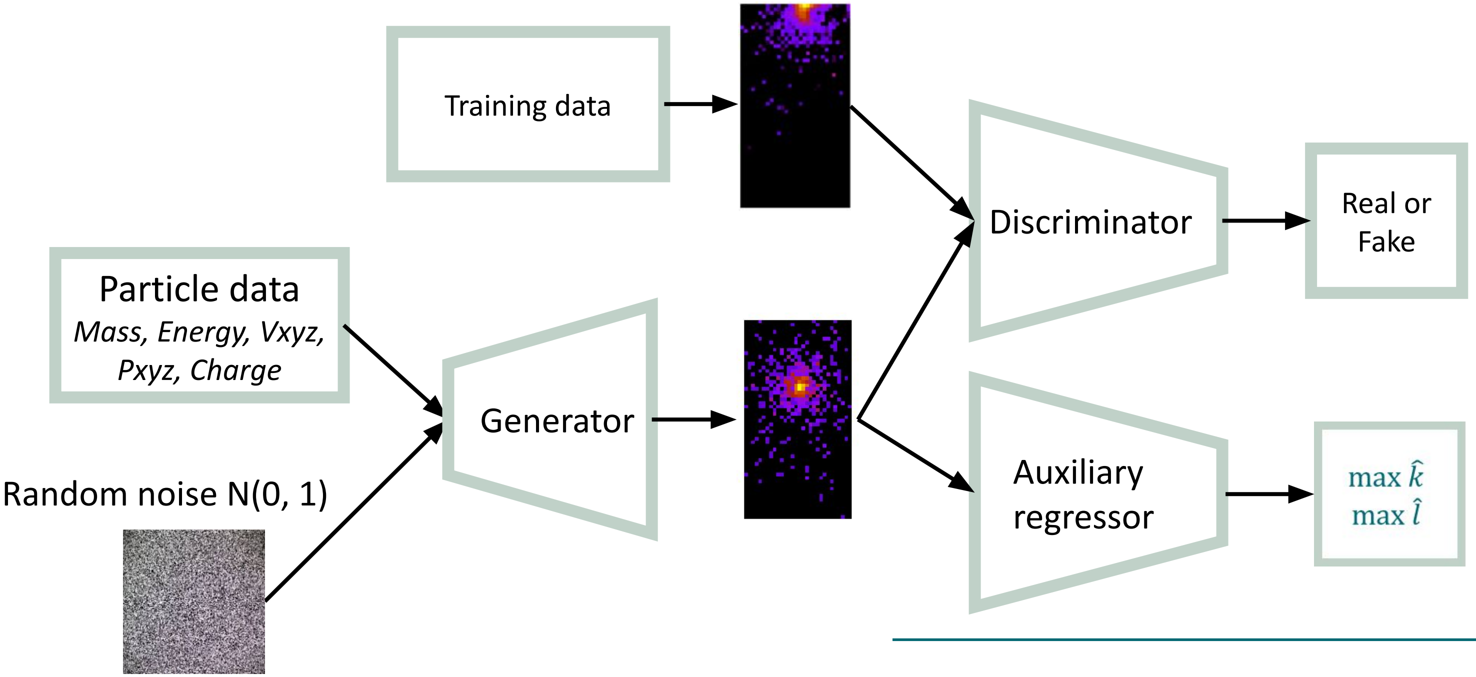


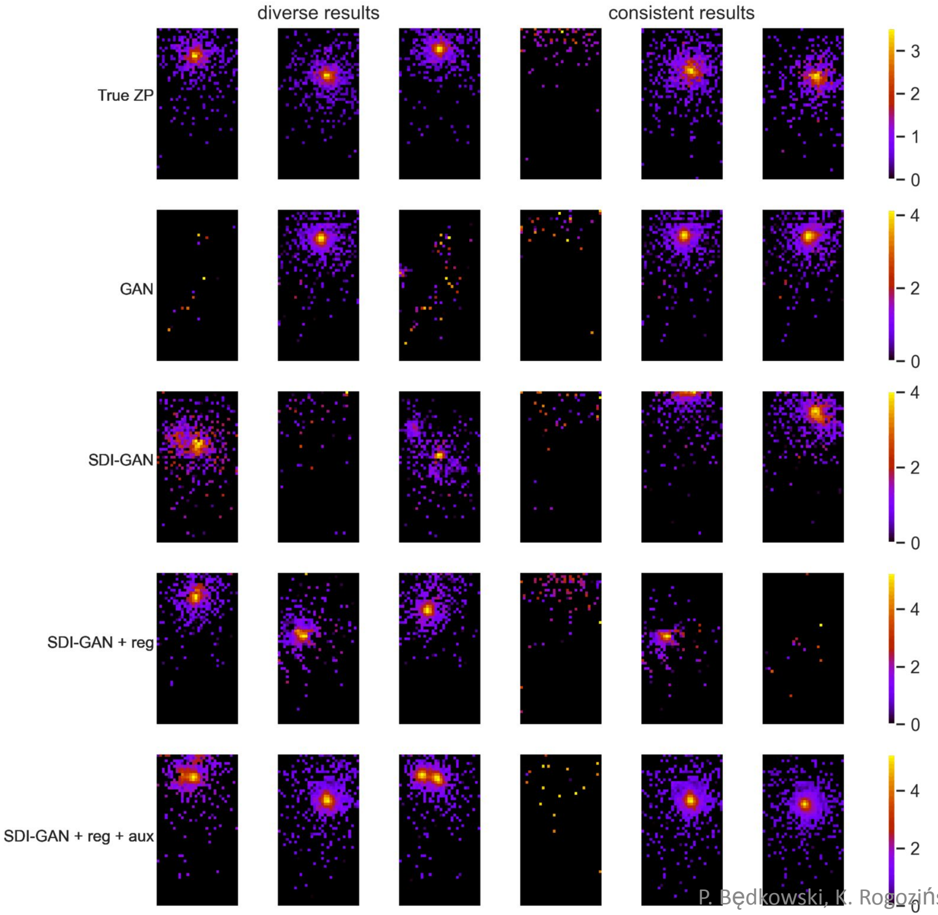
k	l
18	14
3	9
⋮	⋮

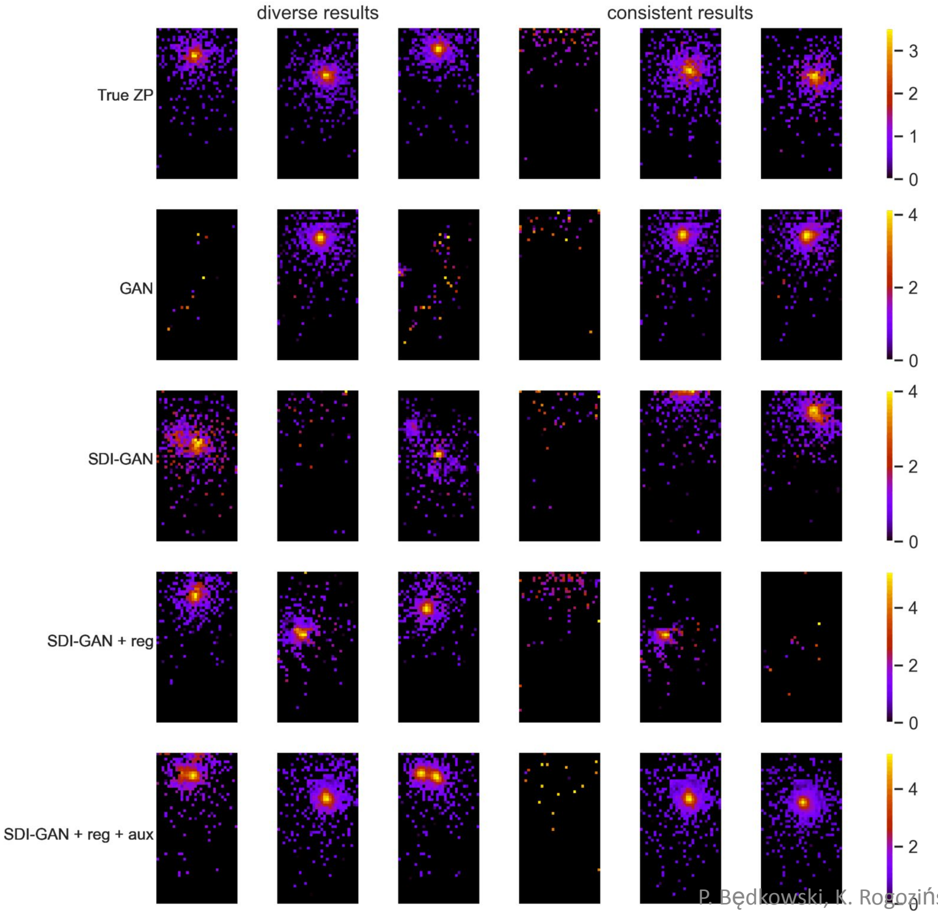
$$L_{aux}(G) = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\hat{k}_i - k_i\right)^2 + \left(\hat{l}_i - l_i\right)^2\right]$$

Auxiliary regressor

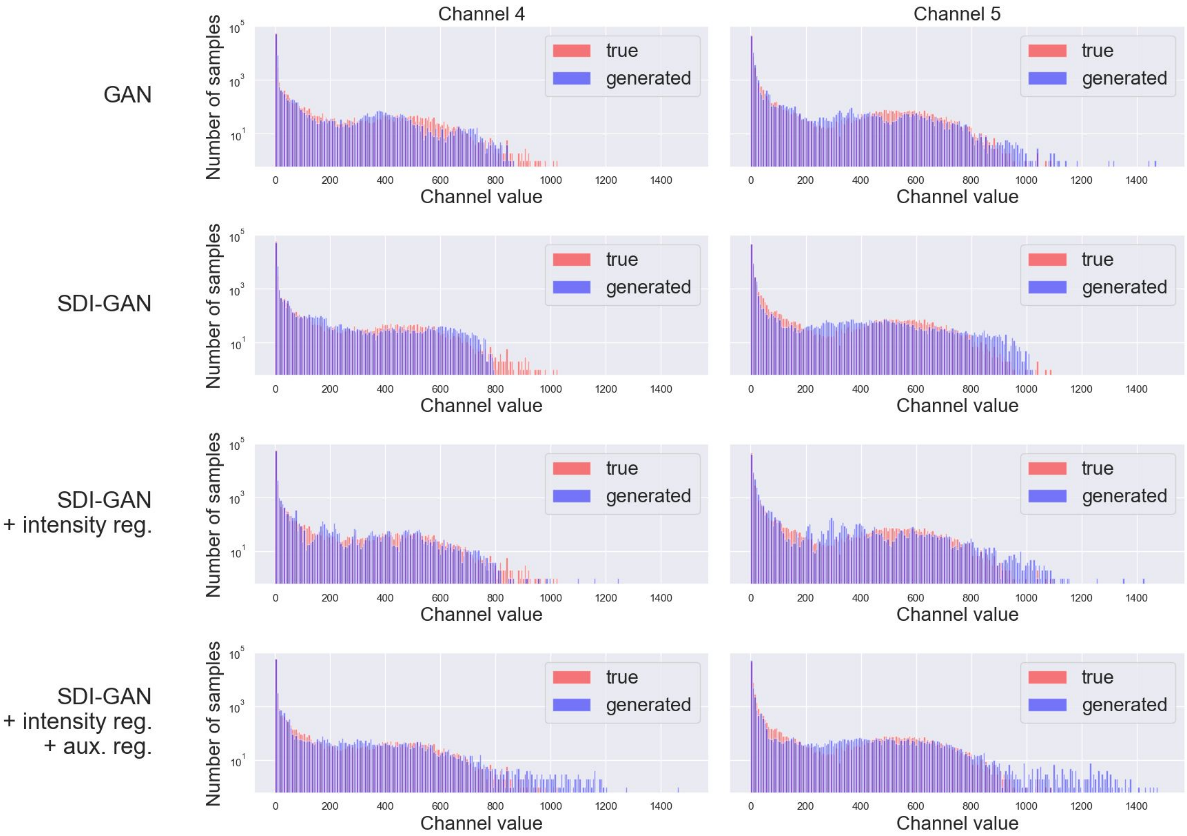
Idea: We can control geometric properties of the generated image.




diverse results
consistent results
True ZP
GAN
SDI-GAN
SDI-GAN + reg
SDI-GAN + reg + aux
3
2
1
0
4
2
0
4
2
0
4
2
0
4
2
0



- The GAN repeatedly generates consistent images,
- SDI-GAN improves on delivering diverse results where needed,
- Intensity regularization helps to produce images with sum of pixel close to original,
- Applying auxiliary regressor ensures the preservation of geometric properties.



- The GAN has visible problems with underproducing high-energy responses.
- Implementation of additional regularizations and auxiliary regressor positively influence better alignment to true distribution, but tends to oversample high-energy responses.

Summary

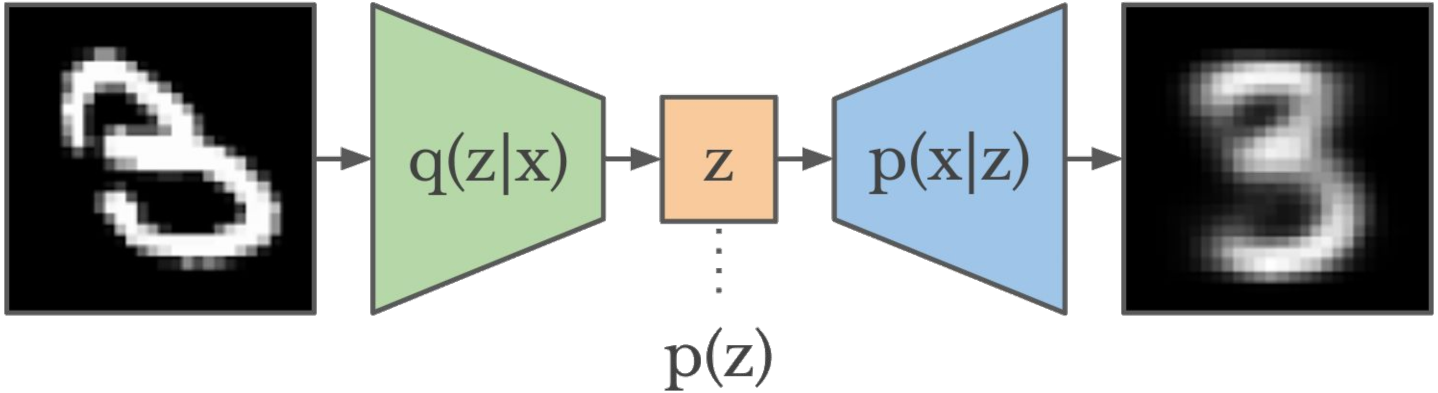
Model	WS MEAN↓	Std dev.
GAN	2.4752	1.6843
SDI-GAN	2.3571	1.6000
SDI-GAN + reg.	2.2916	1.8210
SDI-GAN + reg. + aux. reg.	**2.0777**	**1.6381**

Main takeways:

- Generative machine learning models offer cost-effective alternative to Monte-Carlo based simulations,
- By applying multiple regularizations we can tune the model to the specific data characteristics.

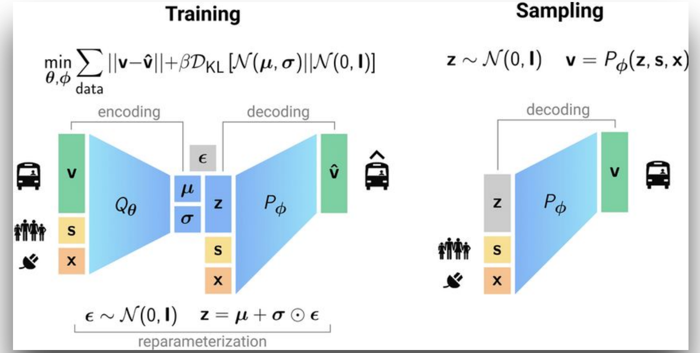
Improving
Variational Autoencoders

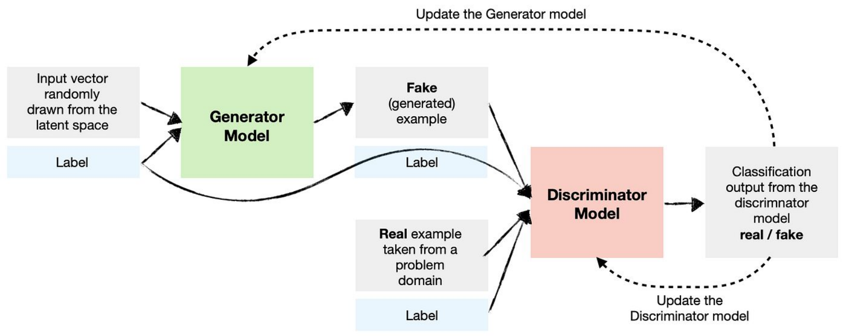
VAE in nutshell



But which generative AI should we choose?

CVAE

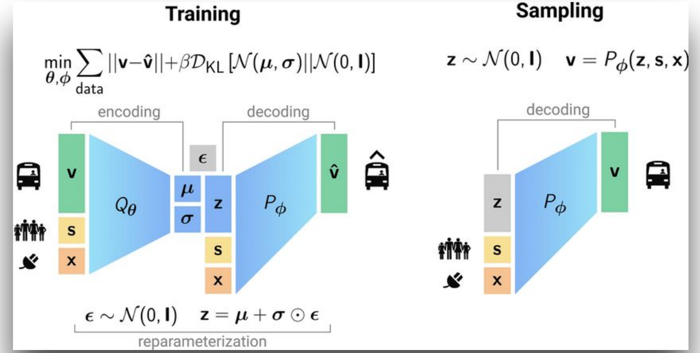


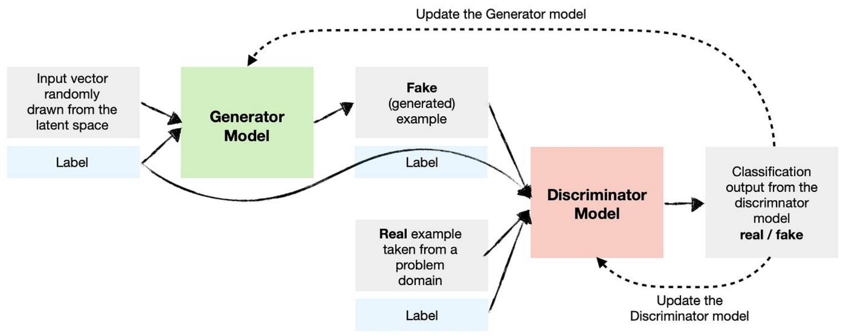


CGAN

But which generative AI should we choose?

CVAE





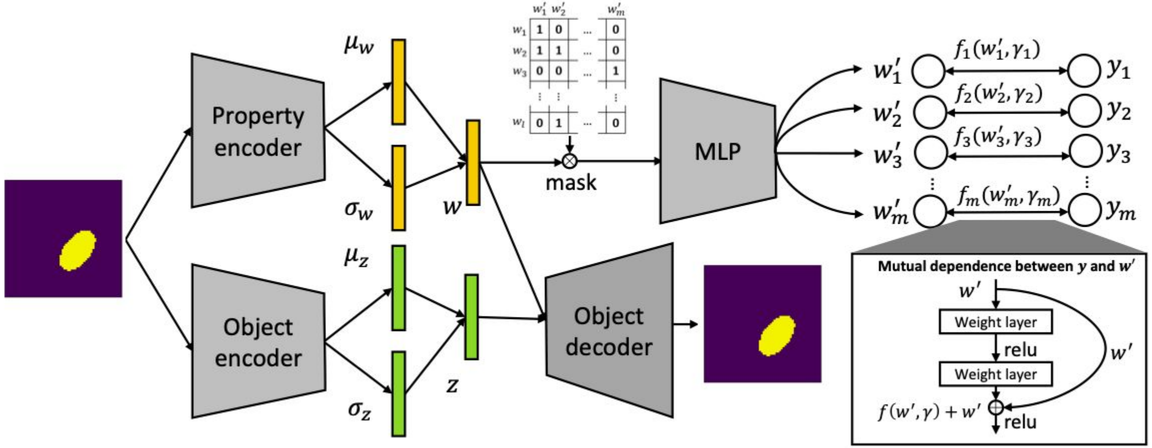
CGAN

BEST FRIEND -> auxiliary regressor 🎯

Control over generated data properties

Control over generated data properties

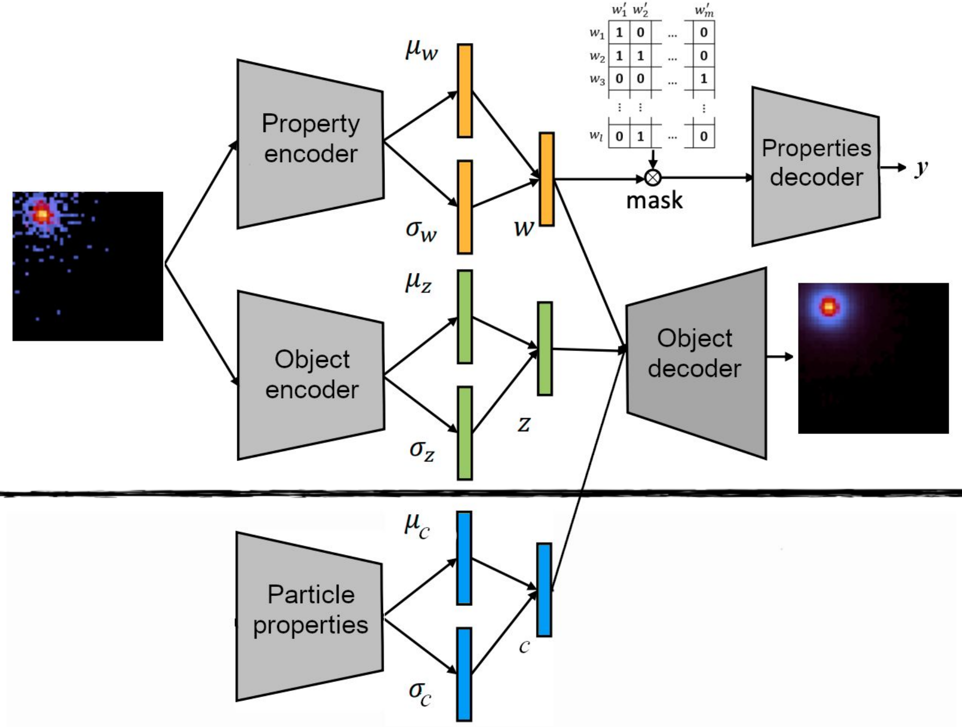
CorrVAE



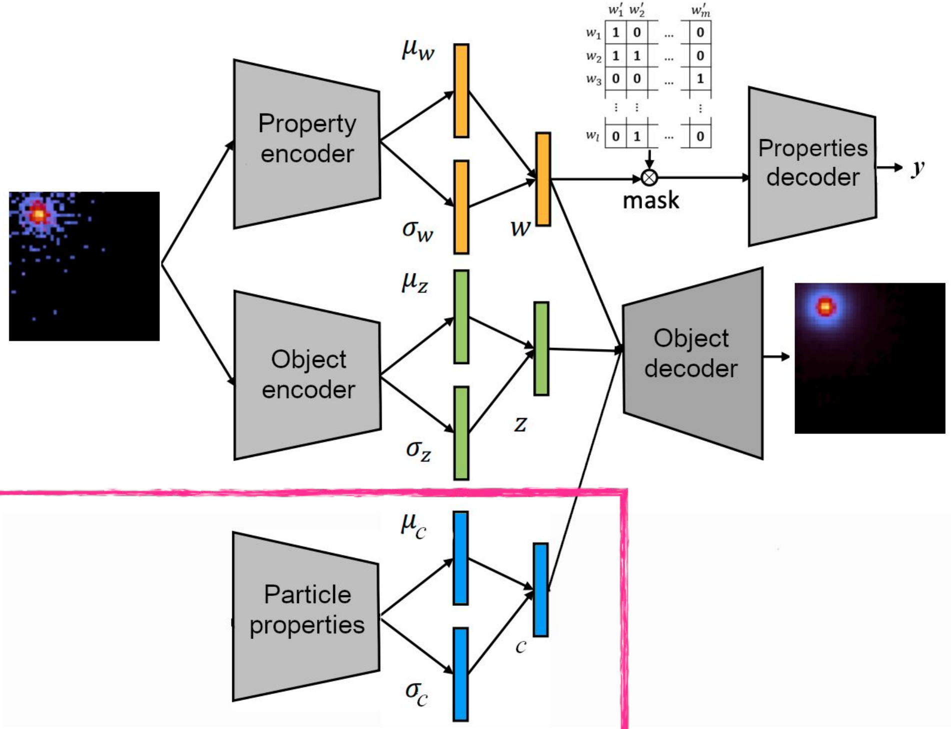
Our solution

Our solution

Adjusted CorrVAE



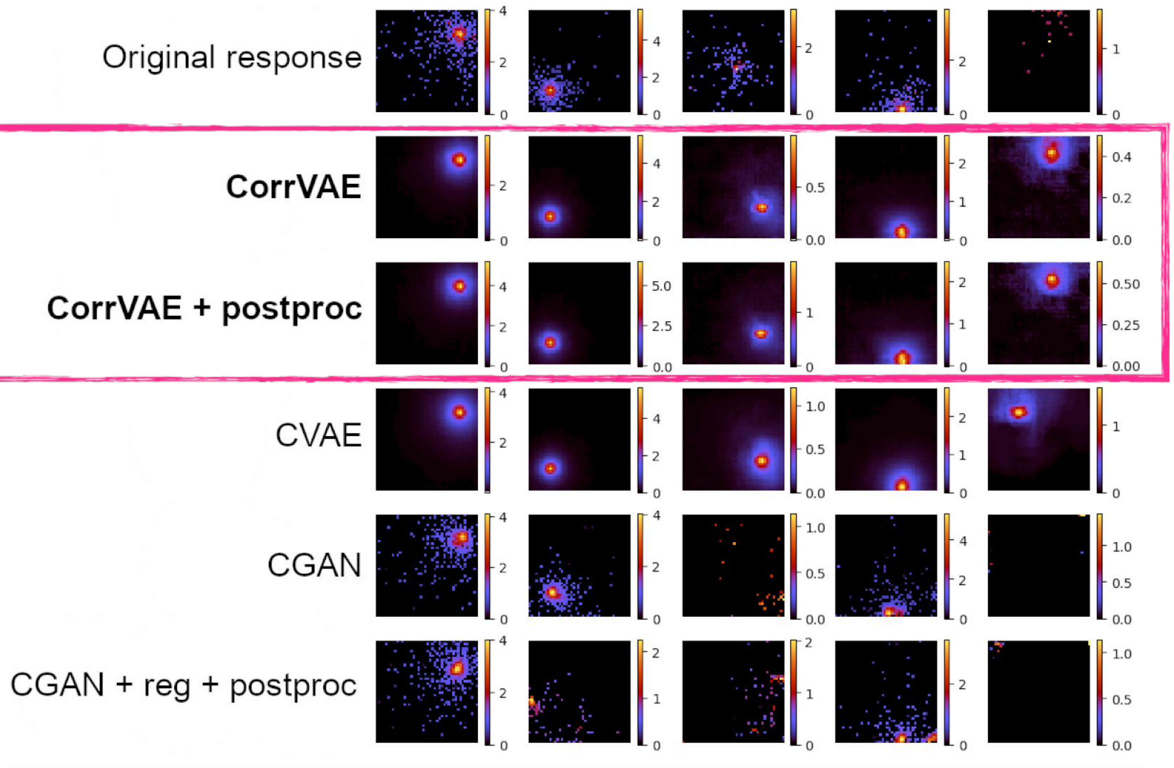
Our solution



Main changes

Generation comparison

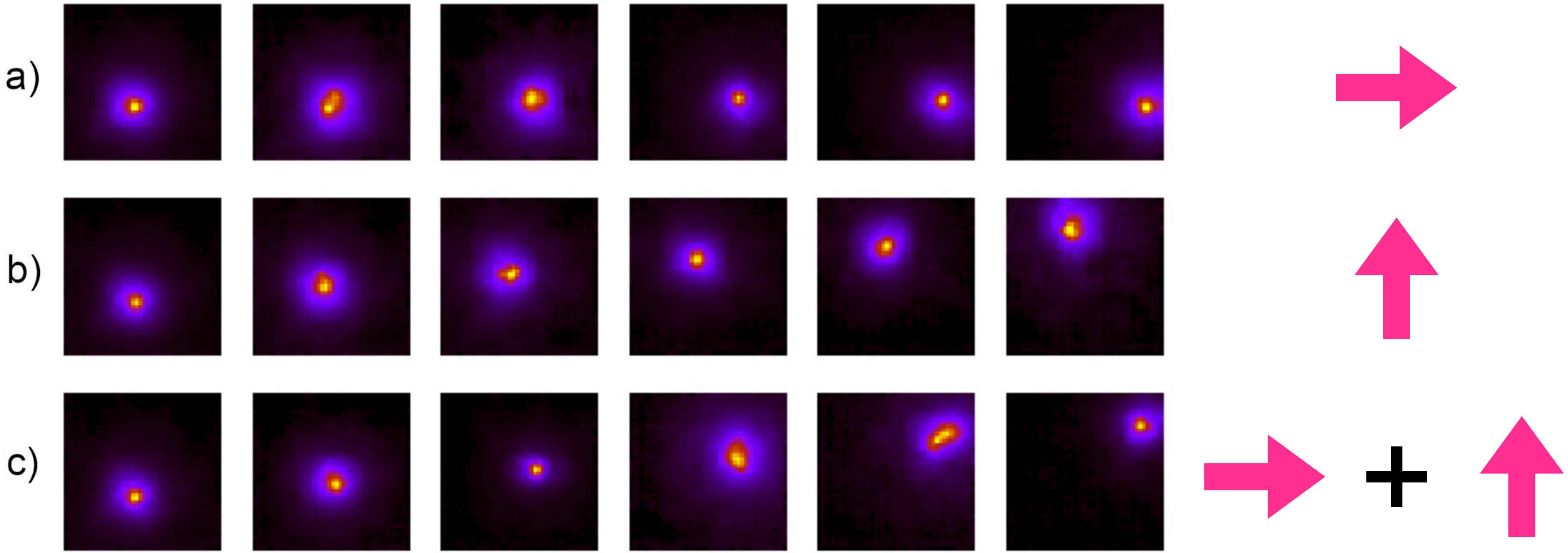
Ours



Comparison of metrics

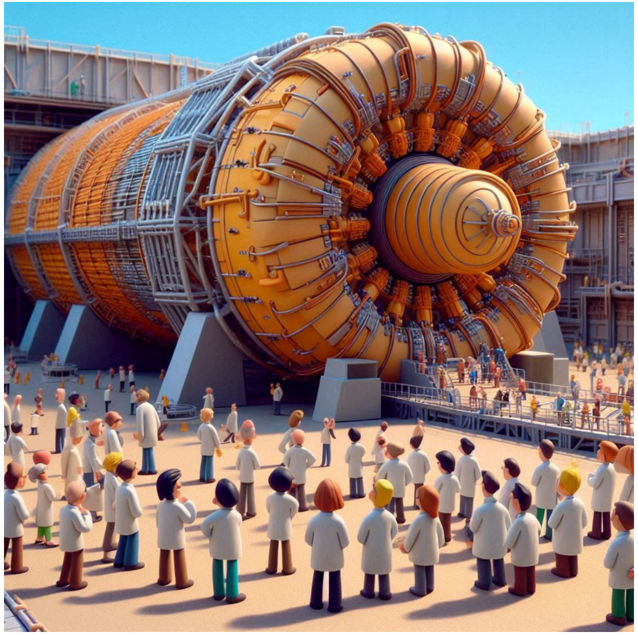
	MSE	Wasserstein
CorrVAE	**1.03**	**16.15**
CorrVAE + postproc	**1.18**	**3.83**
CVAE	1.02	6.35
CGAN	2.96	8.27
CGAN + reg + postproc	2.98	5.15

Traversing the latent space



Improving
Diffusion Models

Diffusion-based generative models



Scientists walk around Large Hadron Collider at CERN

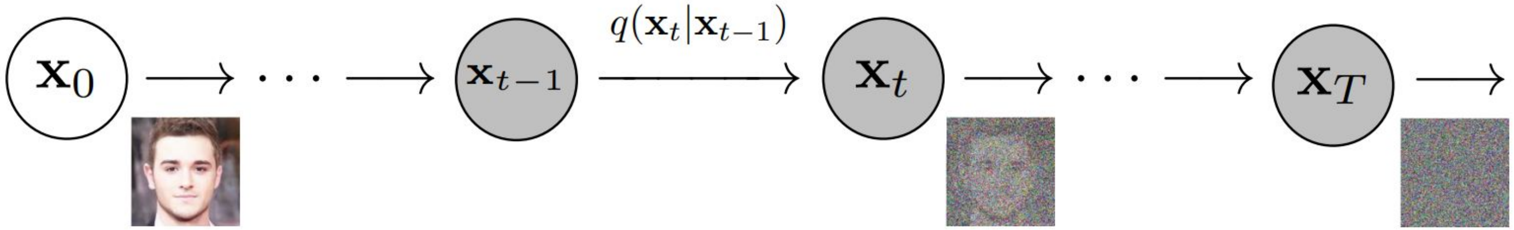
Disney pixar movie poster about a happy small particle flying around at CERN

Attempt to use diffusion models for ZDC simulations

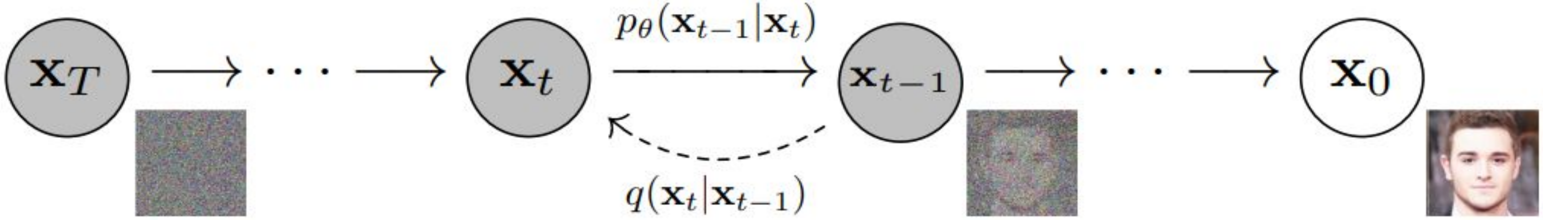
- Diffusion models have many advantages, such as the high quality of the content generated and a high degree of flexibility, allowing them to be used in many fields
- They also have disadvantages, such as the need for a large amount of training data and the need for high computing power
- Following diffusion models were tested in our study: unconditional, conditional and latent diffusion models

Diffusion models - diffusion process

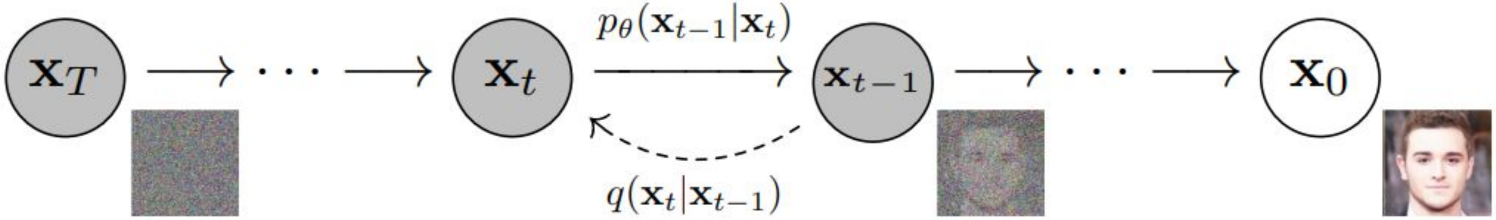
The diffusion process (**forward process**) involves adding Gaussian noise to the image at each step:

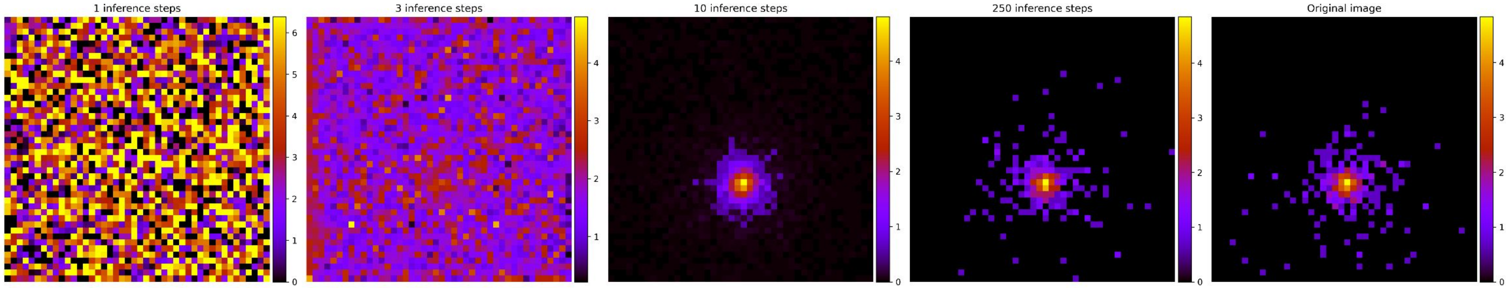


The **reverse diffusion process works** on the same principle in reverse:



DDPM sampling method

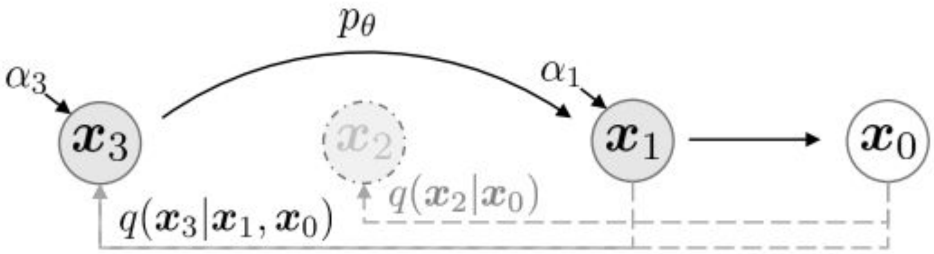




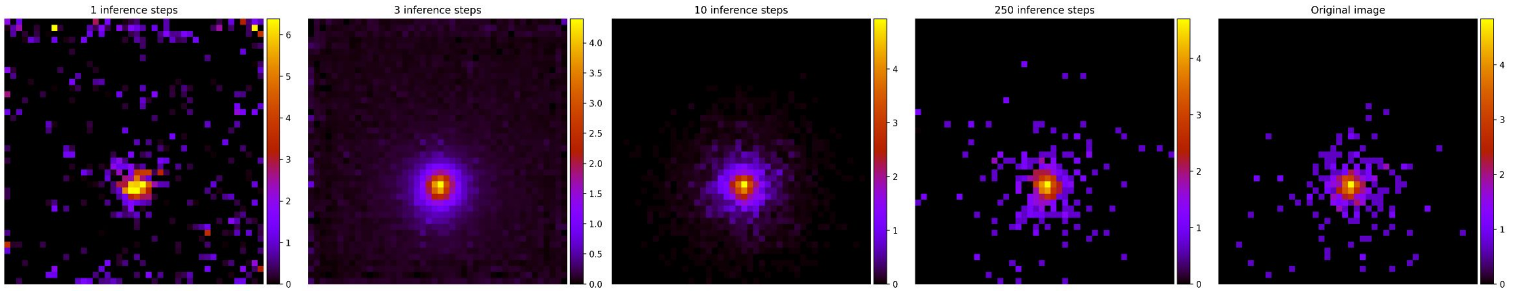
DDPM sampling method removes estimated noise at each step:

Source: *J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models"*

DDIM sampling method



DDIM allows for the omission of intermediate steps in the denoising process, enabling faster generation of samples:

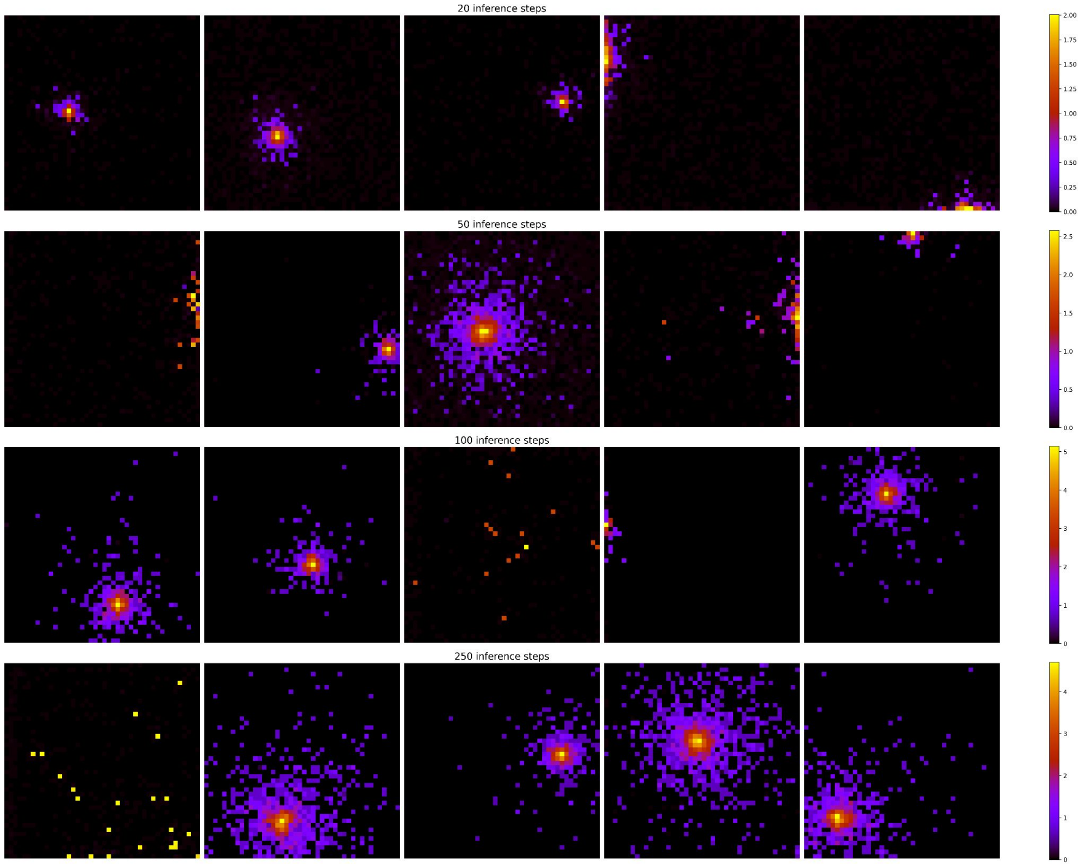


Source: J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models"

Unconditional diffusion model - results

Model	Method	# inference steps	Mean WS	ch1	ch2	ch3	ch4	ch5	Time [min]
Unconditional diffusion model	DDIM	10	20.7	11.3	13.8	12.8	13.6	52.0	3
	DDIM	**20**	**8.1**	**6.3**	**4.7**	**5.1**	**4.4**	**20.0**	**6**
	DDIM	50	50.5	32.7	27.5	34.5	31.4	126.4	12
Unconditional diffusion model	DDPM	20	32.4	19.5	21.1	20.7	21.7	79.3	3
	DDPM	**250**	**2.3**	**2.4**	**2.9**	**2.3**	**1.2**	**2.6**	**59**
	DDPM	500	3.9	4.1	1.2	2.6	2.5	9.1	119
GAN	-	-	8.3	4.4	5.5	7.3	9.1	15	<1
VAE	-	-	6.4	4.6	5.2	4.2	9.1	13.7	<1
Monte Carlo	-	-	-	-	-	-	-	-	90

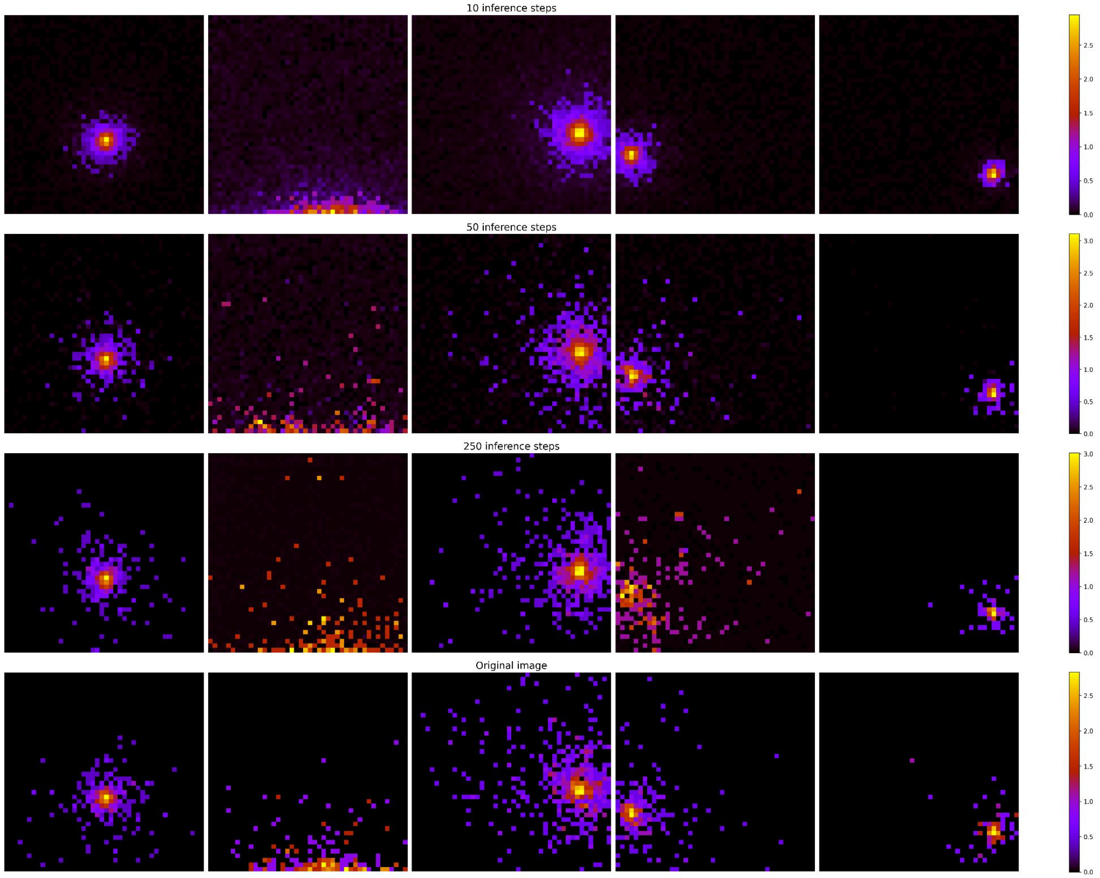
The unconditional diffusion model achieves significantly better results, but needs more time to generate a set of answers.


20 inference steps
50 inference steps
100 inference steps
250 inference steps

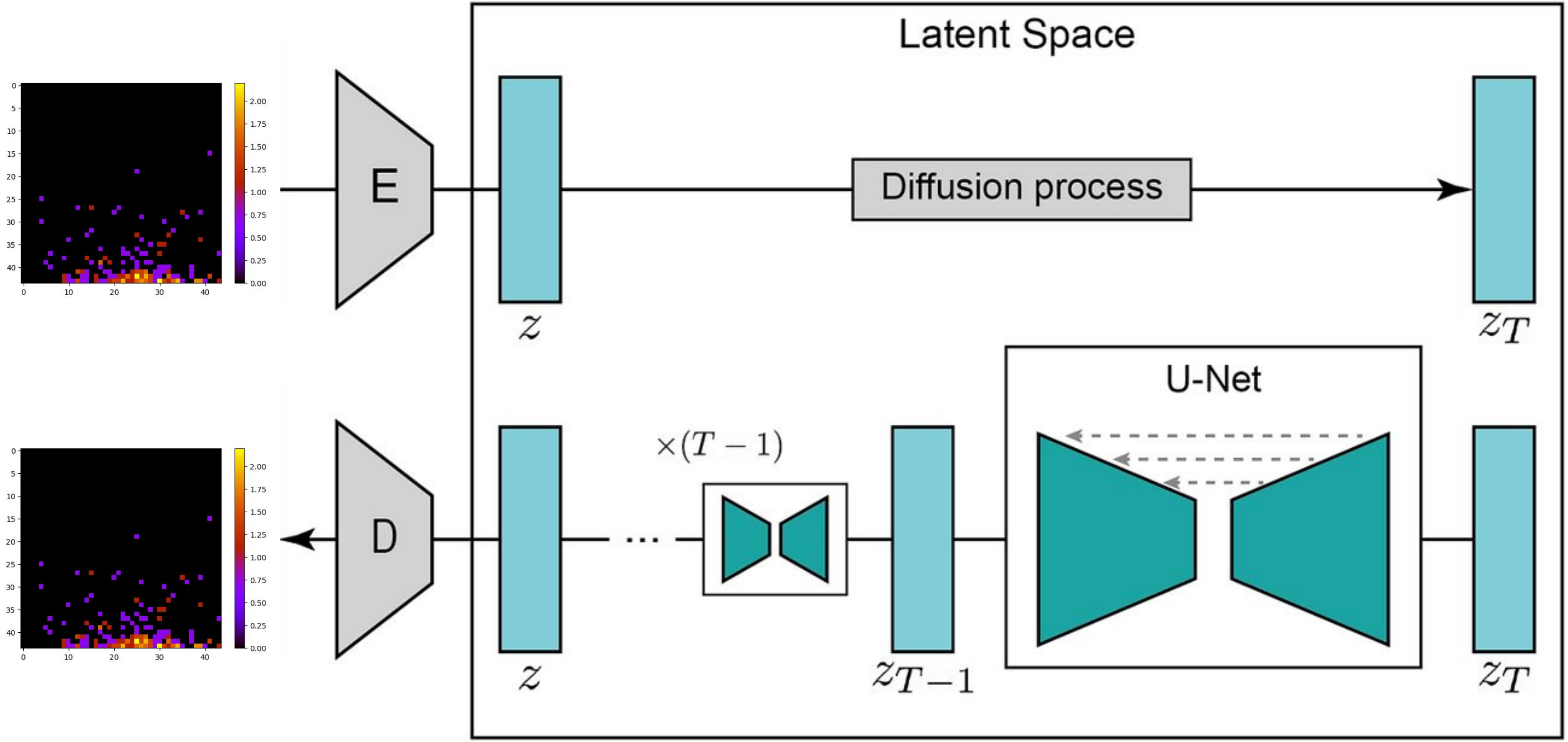
Conditional diffusion model - results

Model	Metgod	# inference steps	Mean WS	ch1	ch2	ch3	ch4	ch5	Time [min]
Conditional diffusion model	DDIM	20	8.1	6.3	4.7	5.1	4.4	20.0	7
	DDIM	**50**	**6.39**	**3.89**	**3.38**	**3.89**	**3.01**	**17.76**	**15**
	DDIM	100	21.6	14	13.1	14	12.7	54.3	30
Conditional diffusion model	DDPM	50	7.2	4.6	4.8	4.6	5.1	16.9	15
	DDPM	**500**	**1.2**	**0.9**	**1.1**	**0.8**	**1.0**	**2.1**	**109**
	DDPM	1000	1.8	1.7	1.8	1.9	1.0	2.4	218
GAN	-	-	8.3	4.4	5.5	7.3	9.1	15	<1
VAE	-	-	6.4	4.6	5.2	4.2	9.1	13.7	<1
Monte Carlo	-	-	-	-	-	-	-	-	90

The conditional diffusion model with cross-attention achieves the best results of the models tested.

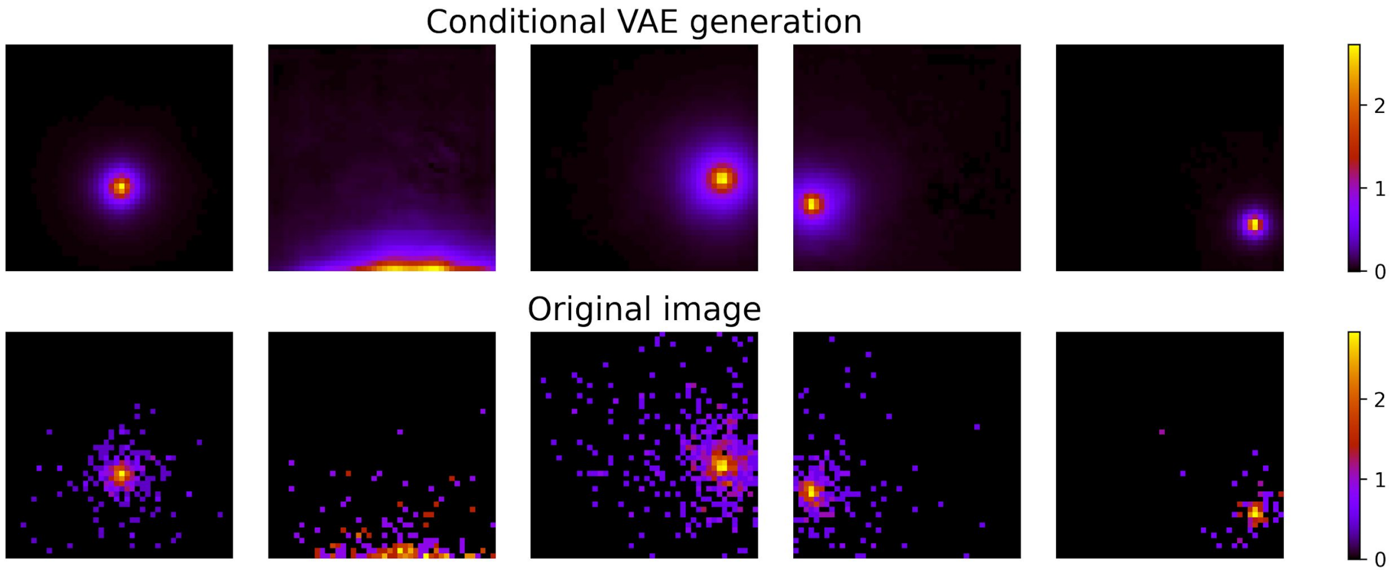

10 inference steps
50 inference steps
250 inference steps
Original image

Latent Diffusion Models

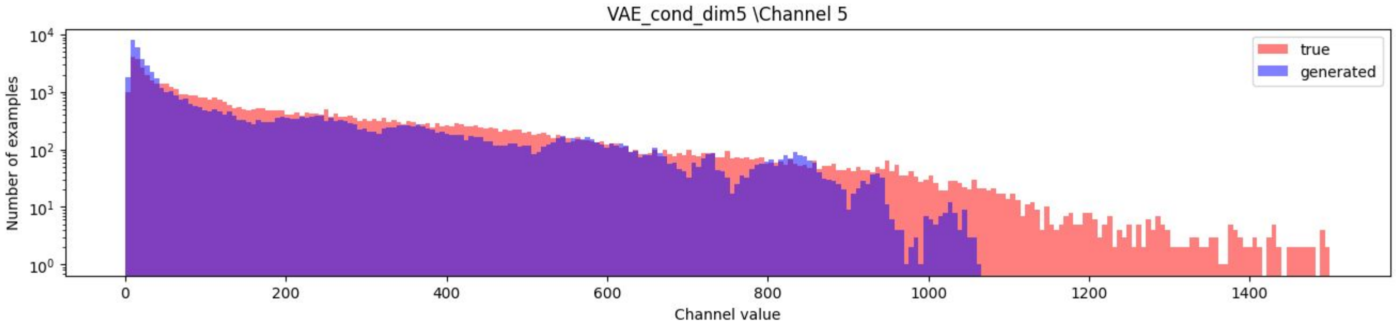


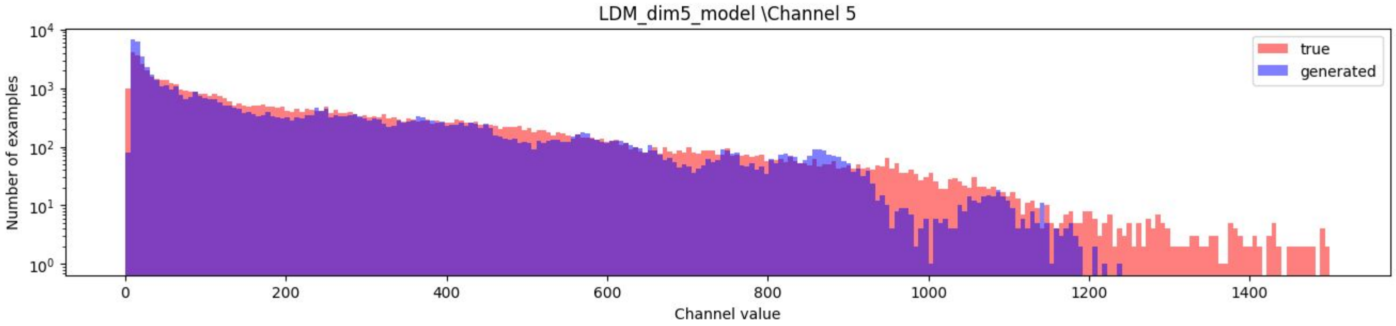
Conditional VAE - results

Model	Mean WS	ch1	ch2	ch3	ch4	ch5	Time [min]
Conditional VAE	23.92	13.1	16.8	13.4	18.5	18.5	<1



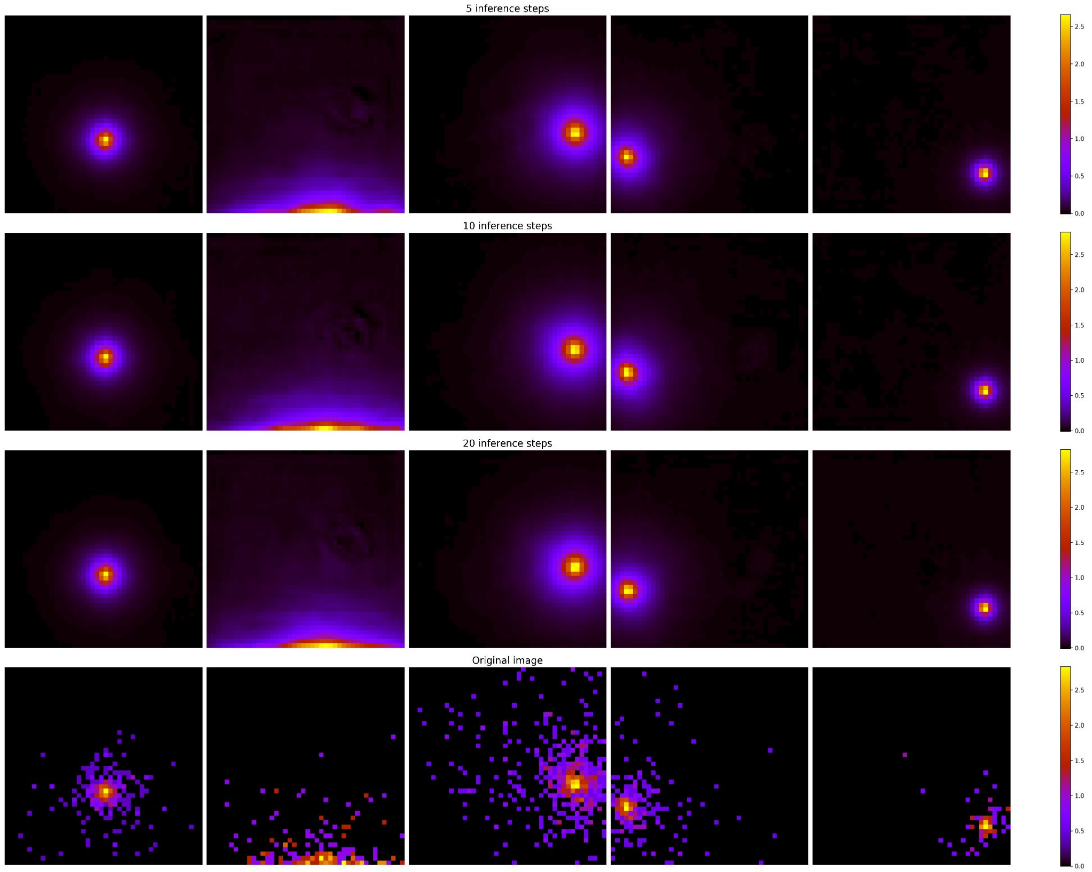
Latent Diffusion Model improves generation of high channel values in VAE





Latent Diffusion Model - results

Model	Method	# inference steps	Mean WS	ch1	ch2	ch3	ch4	ch5	Time [min]
CondVAE	-	-	23.92	13.1	16.8	13.4	18.5	18.5	<1
LDM	**DDIM**	**5**	**13.5**	**5.2**	**9.6**	**9.5**	**11.3**	**30.8**	**<1**
	DDIM	10	14.3	6.7	10.2	10.1	11.9	32.8	<1
LDM	**DDPM**	**5**	**12.6**	**5.8**	**8.8**	**9.1**	**10.5**	**28.6**	**<1**
	DDPM	10	13.5	6.4	9.5	9.5	11.3	30.8	<1
GAN	-	-	8.3	4.4	5.5	7.3	9.1	15	<1
VAE	-	-	6.4	4.6	5.2	4.2	9.1	13.7	<1
Monte Carlo	-	-	-	-	-	-	-	-	90

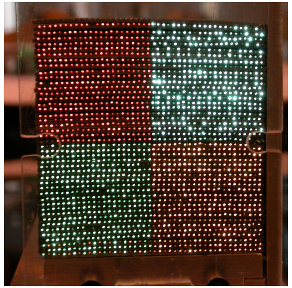

5 inference steps
10 inference steps
20 inference steps
Original image
0.0
0.5
1.0
1.5
2.0
2.5

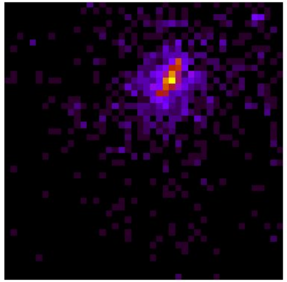
Best results comparison

Model	Mean WS	ch1	ch2	ch3	ch4	ch5	Time [min]
LDM	**12.6**	**5.8**	**8.8**	**9.1**	**10.5**	**28.6**	**<1**
GAN	8.3	4.4	5.5	7.3	9.1	15	<1
VAE	6.4	4.6	5.2	4.2	9.1	13.7	<1
Unconditional diffusion model	**2.3**	**2.4**	**2.9**	**2.3**	**1.2**	**2.6**	**59**
Conditional diffusion model	**1.2**	**0.9**	**1.1**	**0.8**	**1.0**	**2.1**	**109**

Latent Diffusion Model shows significant potential due to its rapid generation time. However, it still requires more development work to reach the performance benchmarks set by other models.

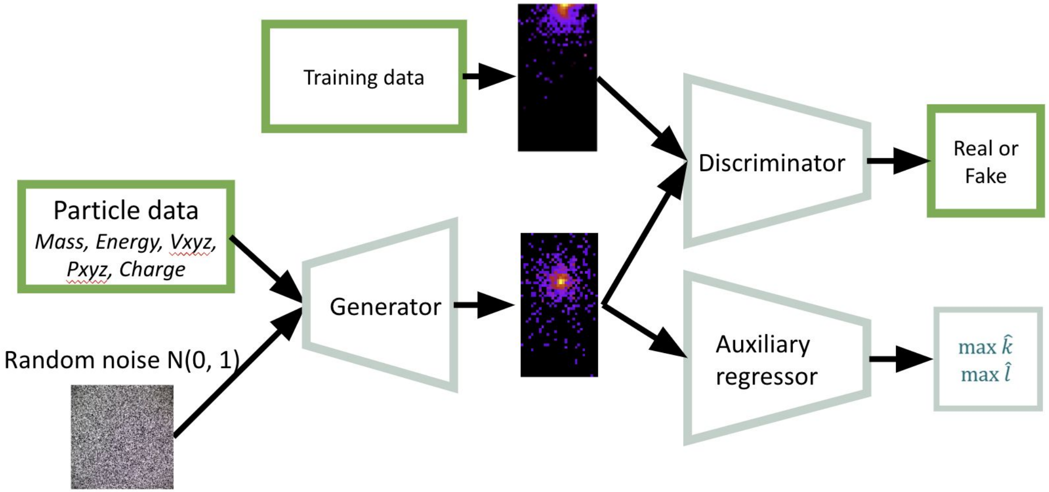
Summary





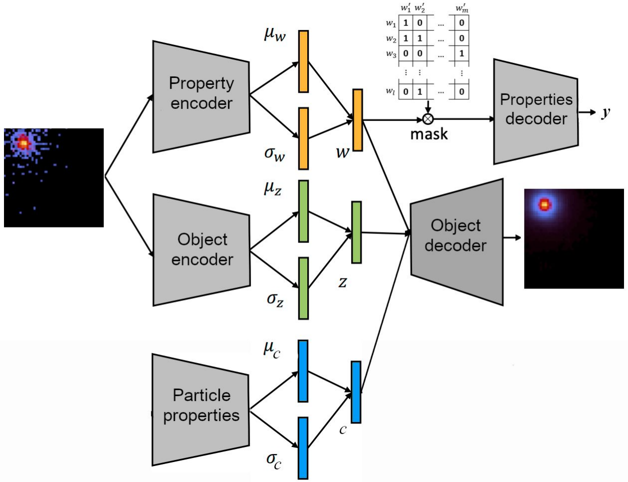
We create fast simulation of the Zero Degree Calorimeter with generative ML

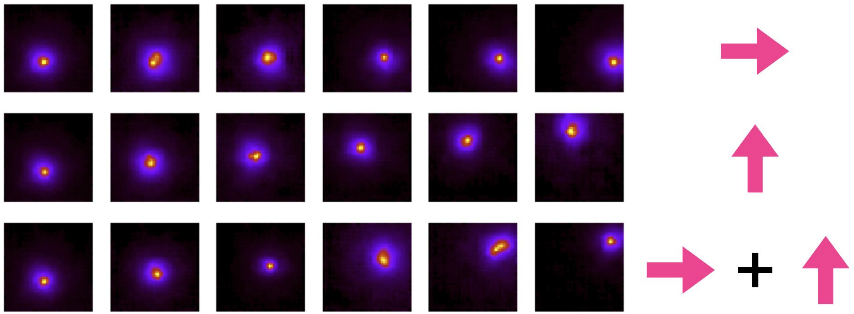
Summary



We increase the fidelity of GAN-based simulation with loss regularization for output diversity and intensity

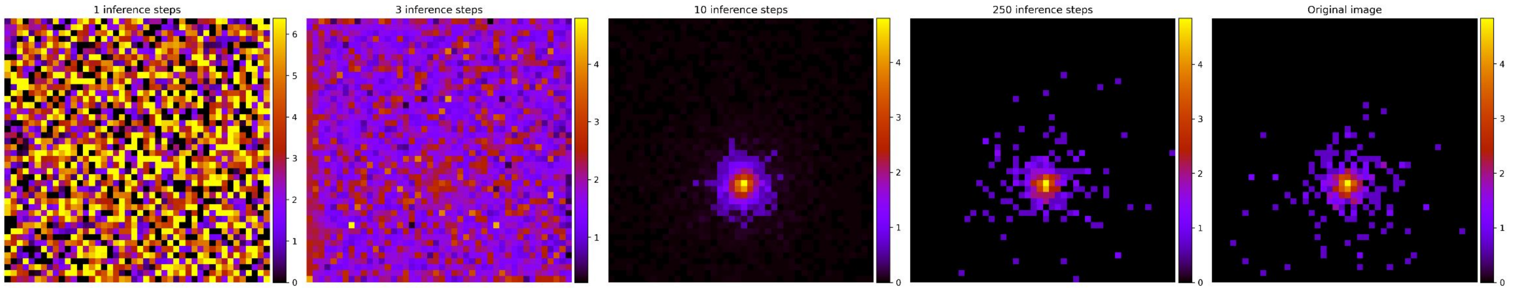
Summary





For VAE models we achieve precise control over the properties of the generated output thanks to separate latent spaces

Summary



We explore the trade-offs between the simulation quality and efficiency for various diffusion models:
Latent Diffusion Models match the speed of VAEs and GANs

WMLQ2024 04/06/2024

Deep Generative Models for Particle Simulations at ALICE, CERN

Patryk Będkowski*, Karol Rogoziński*, Mikołaj Kita*

Jan Dubiński, Tomasz Trzciński, Przemysław Rokita, Kamil Deja

Acknowledgments:

This research was funded by National Science Centre, Poland grants: 2020/39/O/ST6/01478 and 2022/45/B/ST6/02817. This research was supported in part by PLGrid Infrastructure grants: PLG/2023/016393, PLG/2023/016361, PLG/2023/016278.