Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
AGH University of Krakow

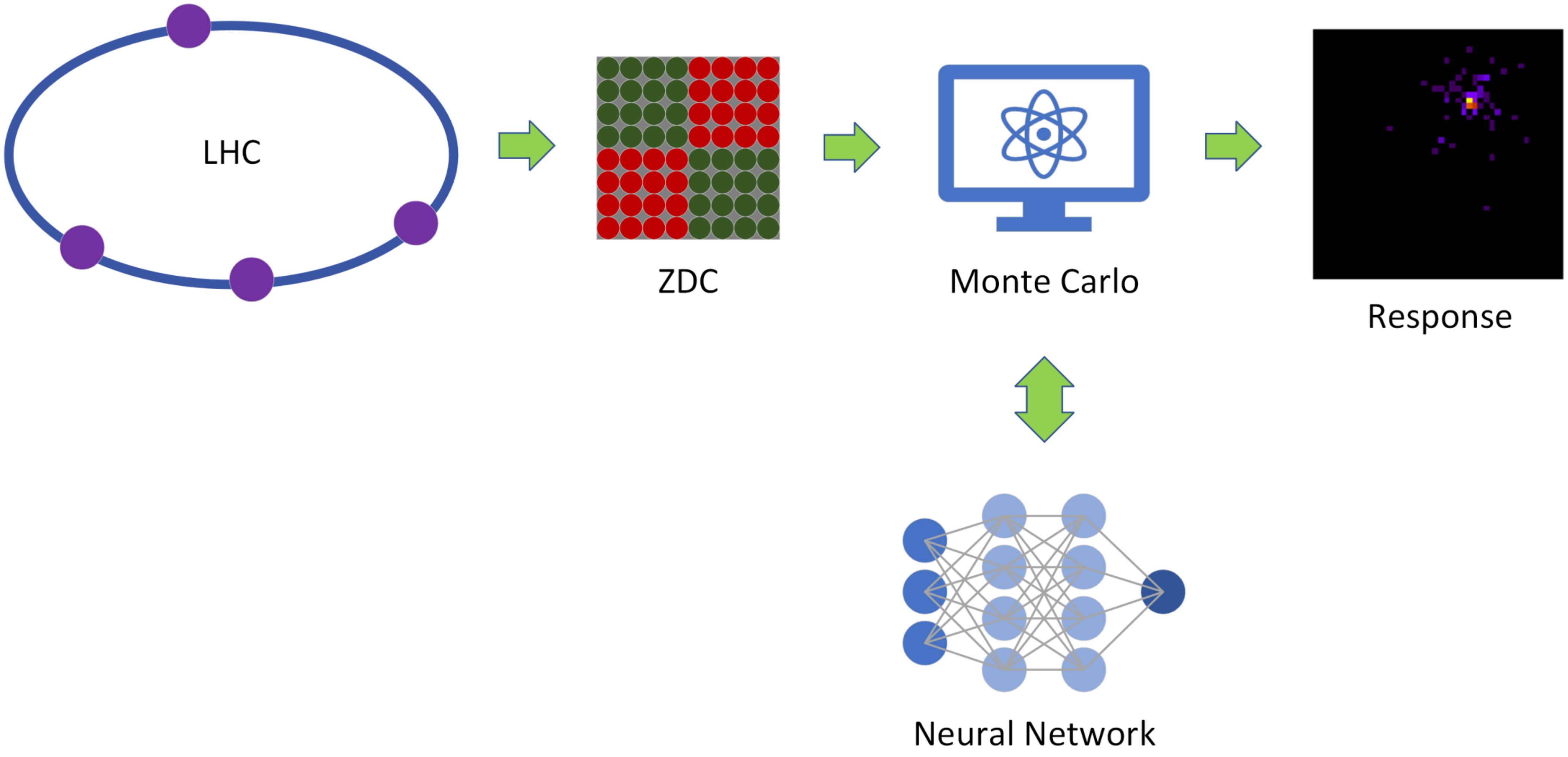
WMLQ 2024
4-7 June 2024, Warsaw Poland

Fast simulation of the Zero Degree Calorimeter responses with generative neural networks

Maksymilian Wojnar, **Emilia Majerz**

2024-06-04

What do we do?

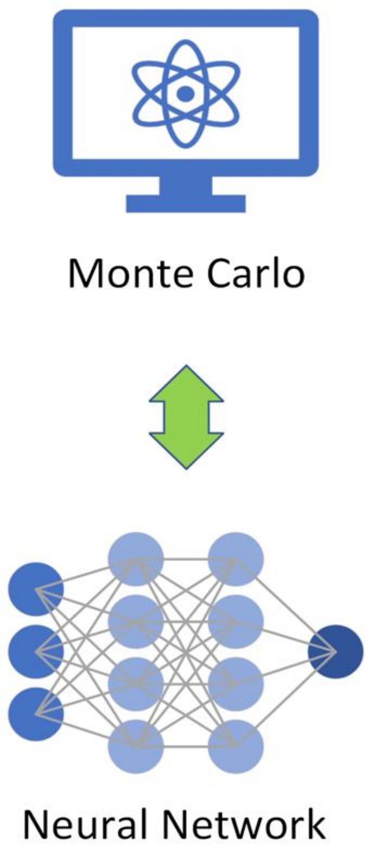


Zero Degree Calorimeter (ZDC)

- ALICE experiment.
- Detects the energy of the spectator nucleons in order to determine the overlap region in nucleus-nucleus collisions.
- Simulated using Monte Carlo approach.
  - Computationally expensive method!
- Neutron detector: 44x44 silica optical fibers grid.
  - Detection of Cherenkov light produced by charged particles in the fibers.
- Detector responses have 2-dimensional structure...
- ...making their simulation a perfect task for generative neural networks!

Fast simulations

- Using a surrogate of the whole mathematical model or its part.
  - The most computationally-intensive part -> a faster surrogate.
- Fast simulations at CERN.
  - Existing approaches at different experiments.
  - VAEs, GANs, NFs, Diffusion.
- There's still a gap to fill!
  - Research done mostly on other experiment's calorimeters.

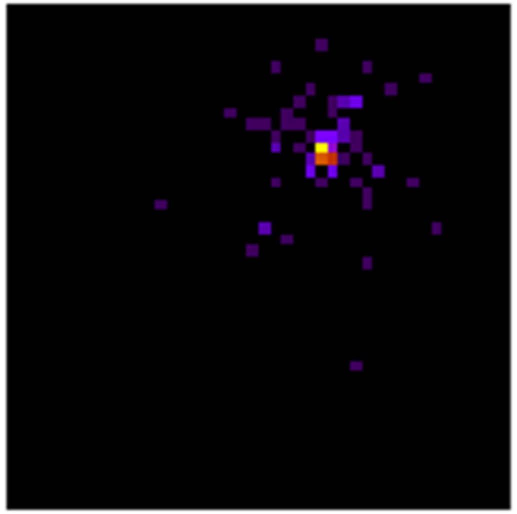


AGH RESEARCH UNIVERSITY EXCELLENCE INITIATIVE

Dataset

- Tabular data + images.
  - 306780 samples.
- Simulation input – primary particle.
  - 9 features: energy, momenta (3d), primary vertex position (3d), mass, charge.
- Simulation output – detector response.
  - Only neutron detector.
  - Detector response treated as an image.
  - Responses with 10 and more photons.
- Diversity of detector responses.

10	1.5	2.3	4.1	0.1	1.4	5.6	1.0	0.0



Diverse results from the same parameters

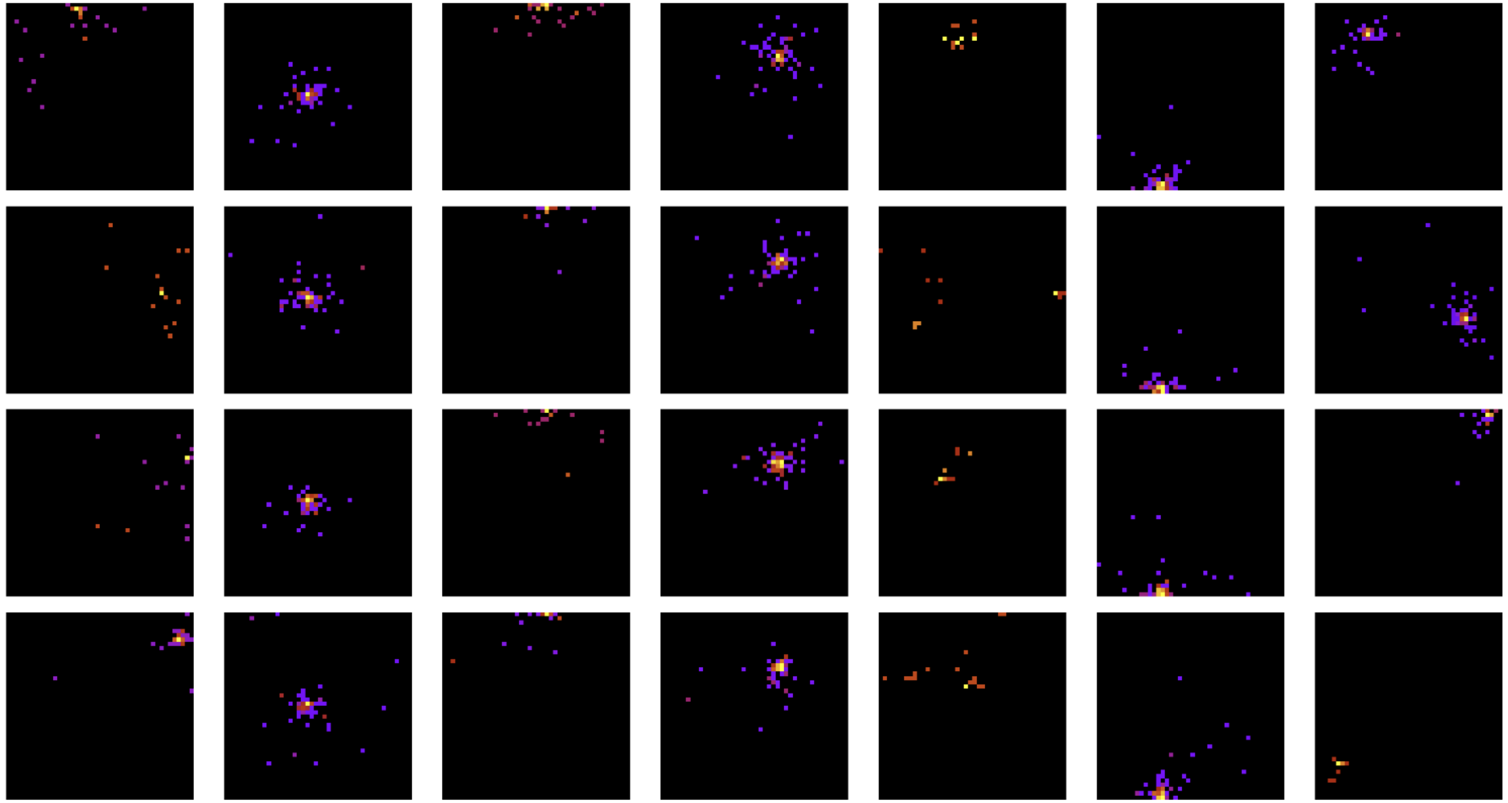
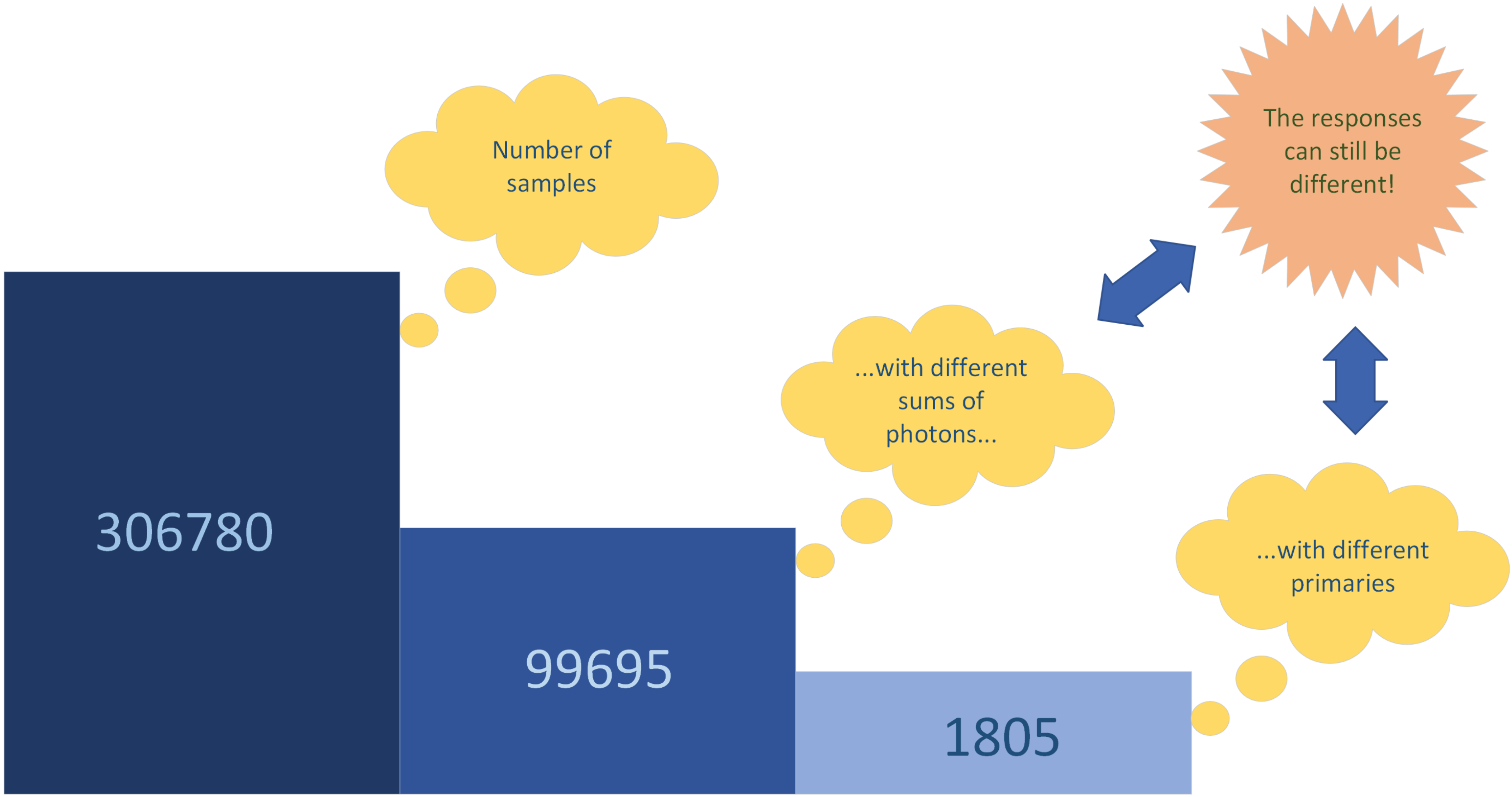


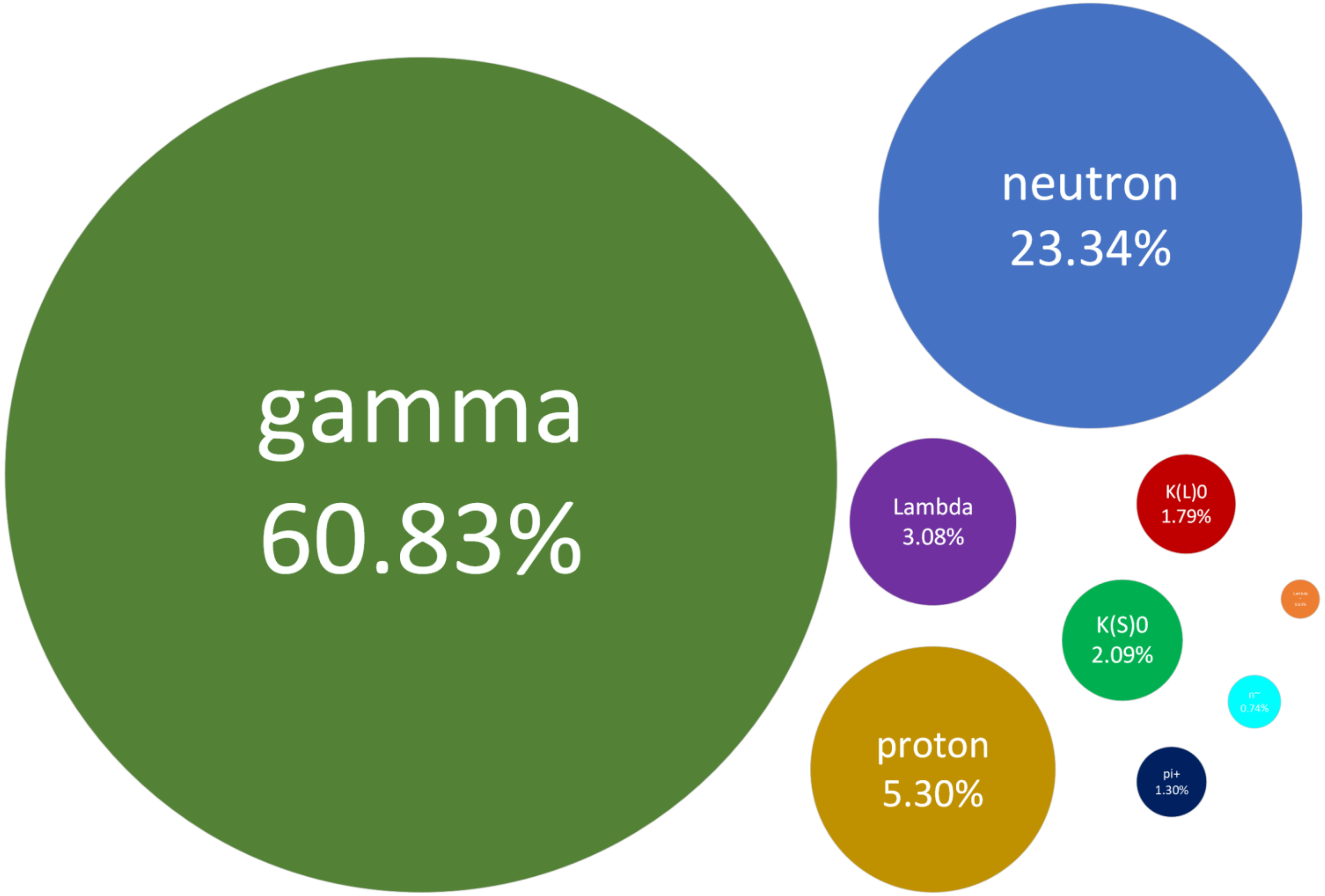
Figure 1: Example ZDC neuron detector simulations generated with GEANT software. The following columns are the responses to different particles, the first from the left is $\pi+$, followed by γ particles (the most common in this dataset), and the last is K_S^0. The rows show independent runs for the same particles.

Many samples... yet sparse data?



agh.edu.pl

Oh no... imbalanced data!



21 different particles in total

And... we have our challenge!

Chosen generative frameworks

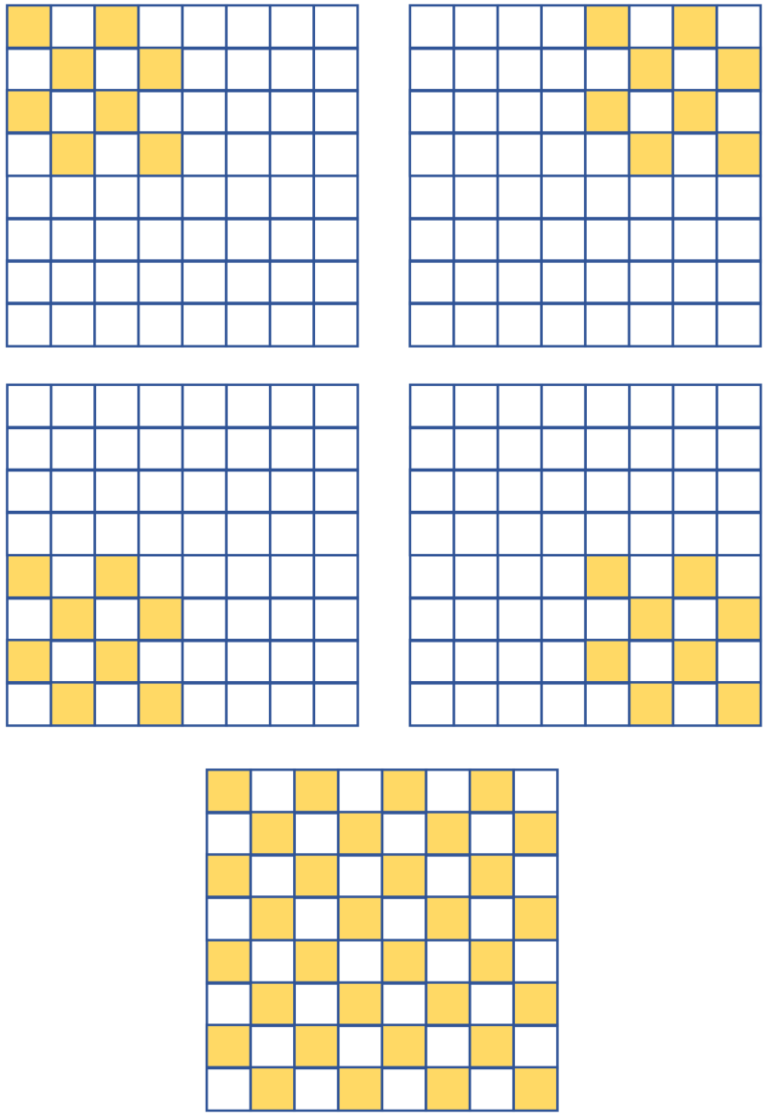
- Autoencoders
- GANs
- Vector Quantization
- Diffusion
- Normalizing Flows

Network architectures:

- CNN
- ViT
- MLP-Mixer

How do we measure the performance?

- Pixel-to-pixel comparison.
  - Can help... but not as a primary metric.
- Comparison of distributions of results.
  - ...but which distributions?
- One image -> 5 channels.
  - Wasserstein distance.
  - MAE.
- RMSE.
- Generation time.



Autoencoders

- Conditional Variational Autoencoders.
- Autoencoders with noise generator.
- Supervised autoencoders.

Architecture	CNN			ViT			MLP-Mixer		
Metric	Wasserstein	MAE	RMSE	Wasserstein	MAE	RMSE	Wasserstein	MAE	RMSE
VAE	**11.52**	17.76	50.38	11.90	18.05	49.48	12.22	18.00	49.51
Supervised AE	23.71	31.90	72.32	20.43	30.60	74.64	17.08	26.90	104.83
AE + Sinkhorn NG	26.53	29.07	66.16	**11.34**	15.88	44.17	x	x	x
AE + MSE NG	37.56	39.32	92.28	**11.19**	15.47	43.49	x	x	x

Generative Adversarial Networks (GANs)

- Conditional GANs.
- SDI-GAN.
- Postprocessing + additional loss.
- Comparison of different GAN models:

Model	Wasserstein	MAE	RMSE
GAN	7.09	25.65	104.60
GAN + postprocessing	**5.70**	24.71	100.98
GAN + l2 loss	6.44	27.37	109.24
GAN + l2 loss + postprocessing	6.07	26.78	107.07
SDI-GAN	6.57	27.01	107.82
SDI-GAN + postprocessing	6.36	26.58	105.94

Vector Quantization – the more, not always the better!

- VQ-VAE - discrete latent representations.
  - Choosing a nearest neighbor from the codebook.
- VQ-GAN - VQ-based generator.
- Medium-sized VQ-VAE achieves the best results in terms of reconstruction:

Model size	Wasserstein	MAE	RMSE
0.25M	11.54	12.96	38.46
1M	**9.86**	11.84	37.22
4M	11.73	13.78	43.54
13M	11.40	12.87	37.90
52M	12.12	13.73	39.74

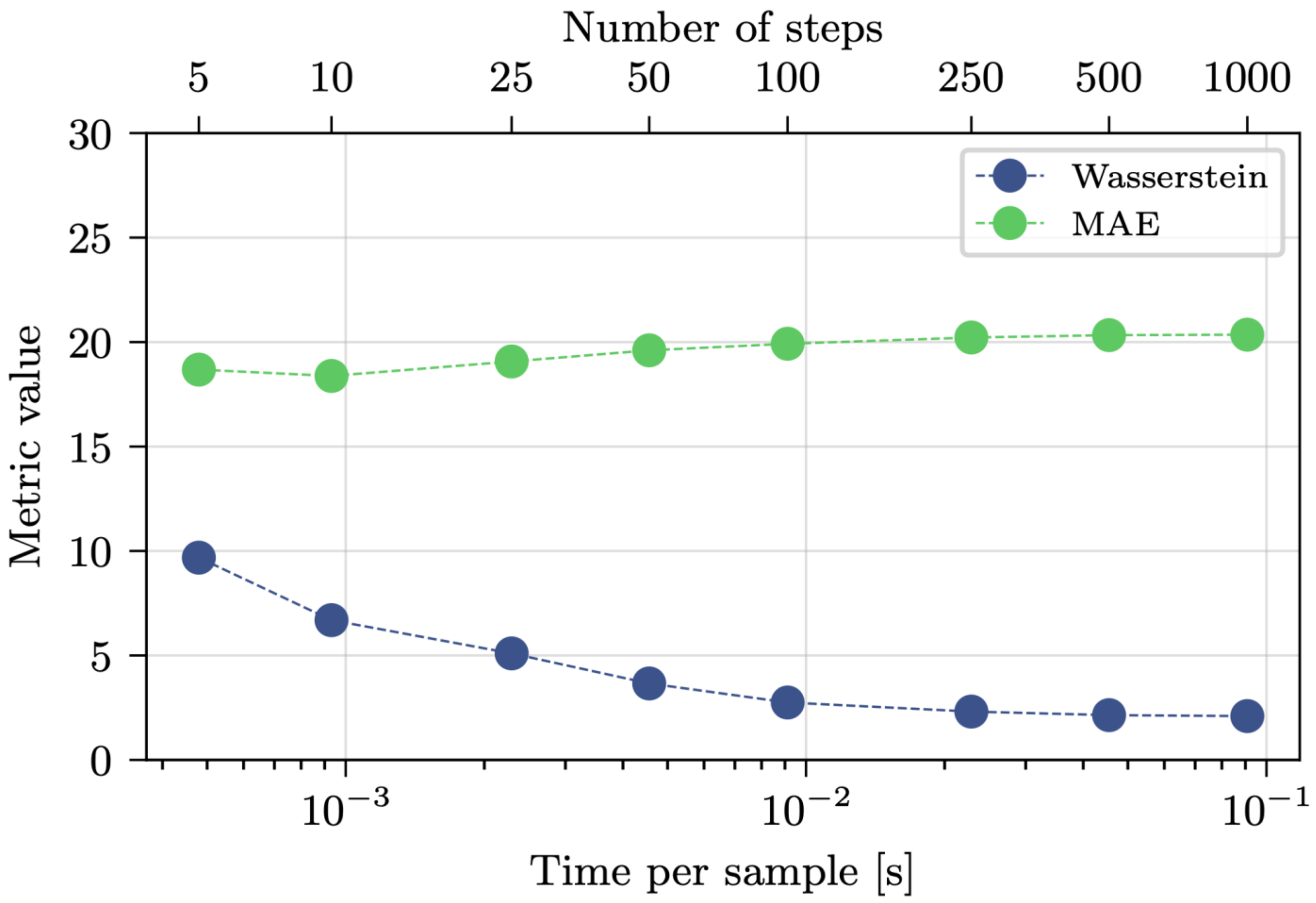
Normalizing flows

- CaloFlow-like approach.
  - Two models: one for the number of photons and one for the final result.
  - Trained independently.
- Bayesian network for the number of photons, flow as the main model.
- Adding noise to pixel values for training, removing the noise after computations.
  - Big influence on the results!
  - For one of the models:

Noise range: [0; val)	Wasserstein
1.0	12.57
0.75	6.67
0.5	**4.58**
0.1	7.10
0.01	8.10
0.001	7.39

Diffusion models

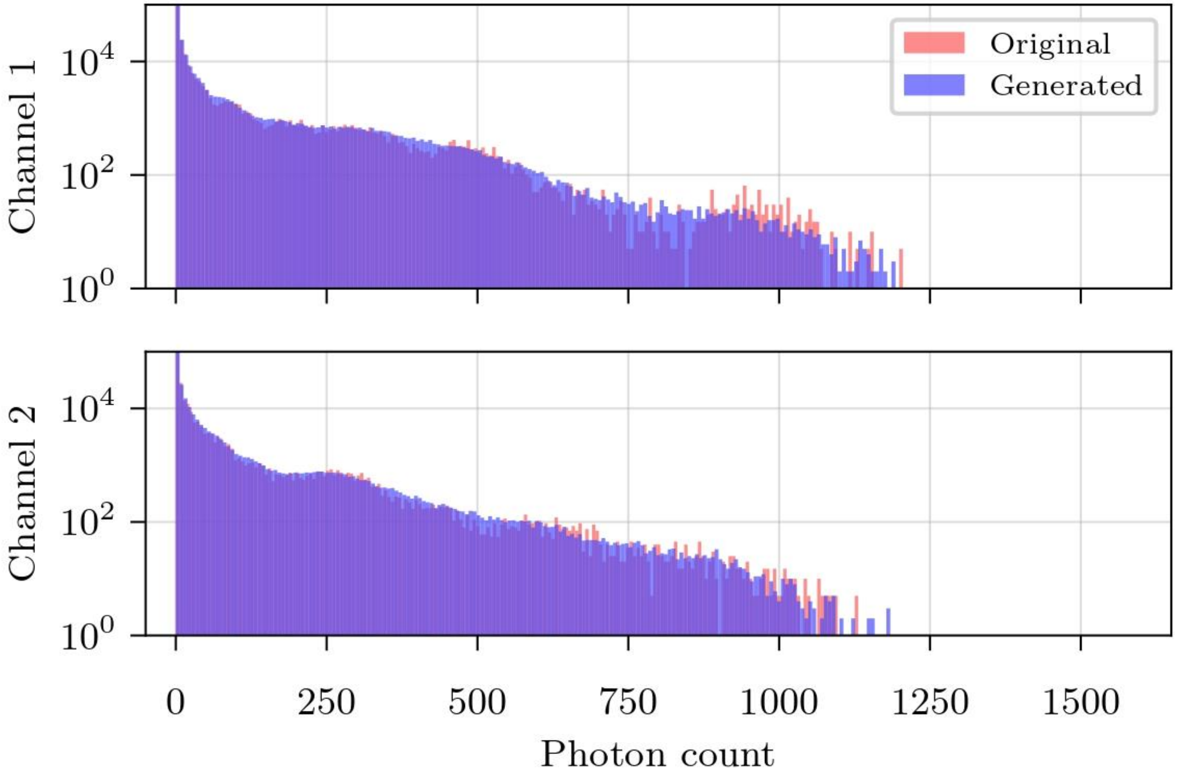
- Training a Denoising Diffusion Probabilistic Model (DDPM).
  - U-Net with convolutions and attentions.
- Sampling with a Denoising Diffusion Implicit Model (DDIM).
  - Allows for fewer steps.
- Tradeoff between generation time and the quality of the samples.

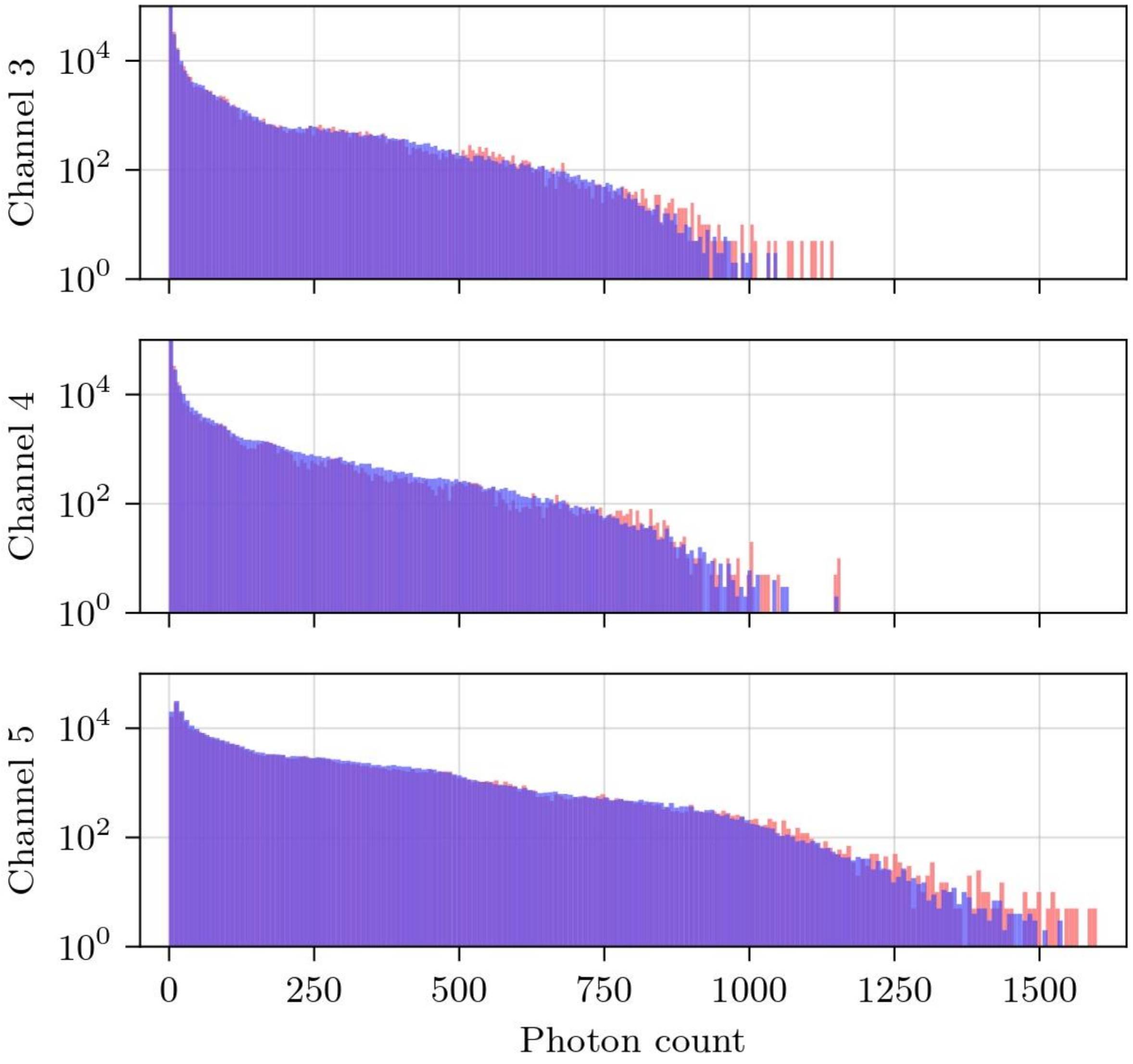


Final performance and generation time comparison

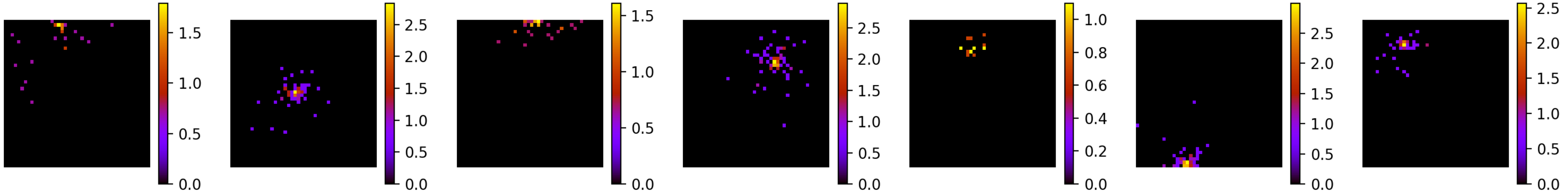
Model	Wasserstein	MAE	RMSE	Generation time [ms]
GEANT (original data)	0.53	16.41	59.87	–
Autoencoder	11.19	15.47	43.49	**0.015**
GAN	5.70	24.71	100.98	**0.023**
VQ-VAE	9.61	21.95	65.82	0.091
VQ-GAN	4.58	22.90	85.45	0.091
NF	**4.11**	19.36	127.22	160.0
Diffusion	**3.15**	20.10	73.58	5.360

Original and diffusion-generated data channels

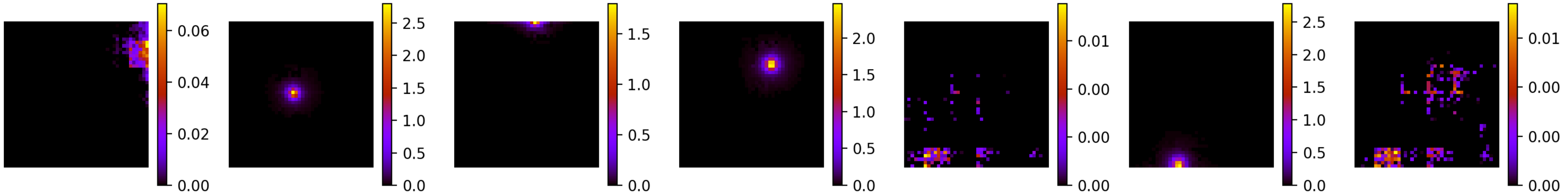




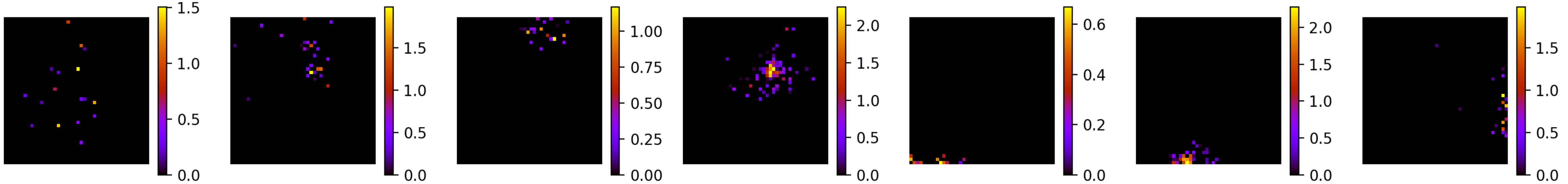
Example simulations



Reference

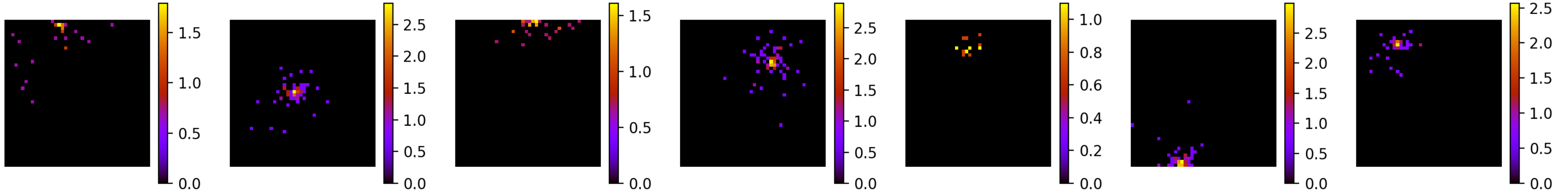


Autoencoder with a noise generator and MSE loss

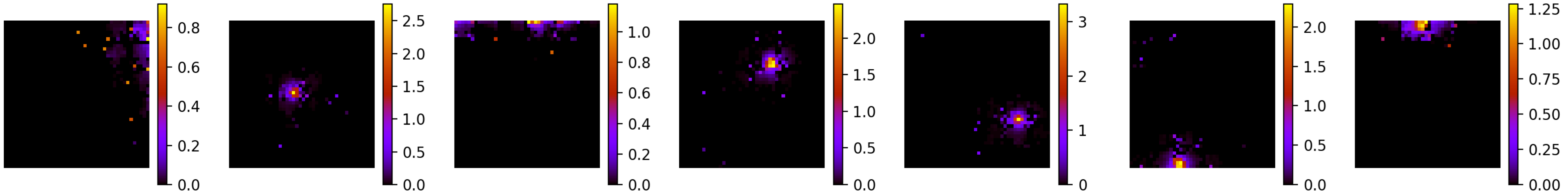


GAN with a postprocessing step

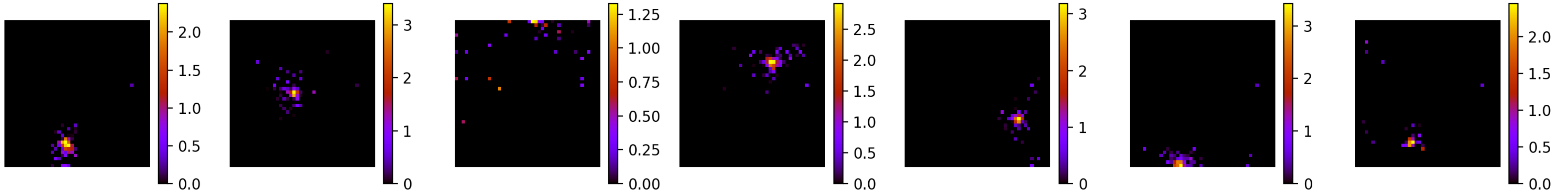
Example simulations



Reference

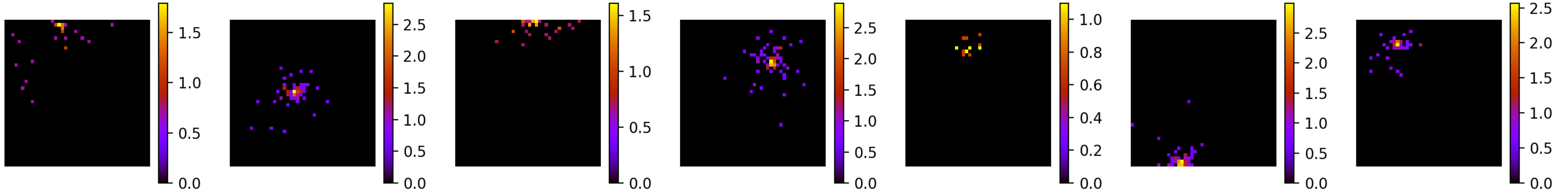


VQ-VAE with a transformer as a learnable prior and adjusted sampling temperature

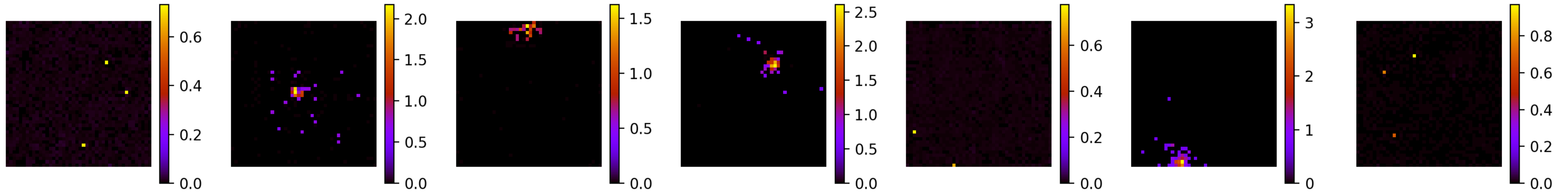


VQ-GAN with a transformer as a learnable prior

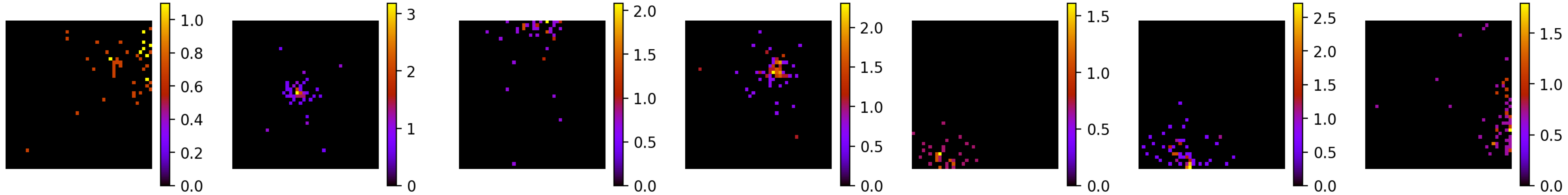
Example simulations



Reference



DDIM after 50 denoising steps and adjusted η parameter



NF with training noise set to 0.5

Pros & cons

	VAEs	GANs	VQs	NFs	DIFFs
Pros	• Fast generation	• Fast generation	• Pretty good metrics and a reasonable generation time	• Good metrics	• The best metrics
Cons	• Blurry outputs • Perturbated diversity	• Training stability issues	• Complicated framework	• Very slow generation*	• Slow generation

Conclusions and future directions

- ViT-based frameworks are the most effective.
- VQs, NFs and Diffusion models perform the best and are worth-developing.
    - VQs provide the best tradeoff between generation time and sample quality.
    - NFs have potential, though need to be faster.
    - Diffusion is great... but maybe can be even better?
- We want to further explore these three methods.

Code availability

https://github.com/m-wojnar/zdc

Acknowledgements

- We would like to thank Professors: Jacek Kitowski, Witold Dzwinel, Marcin Kurdziel and Jacek Otwinowski for their support in this work.
- We would also like to thank the Warsaw University of Technology's group: Jan Dubiński, Kamil Deja, PhD and Professor Tomasz Trzciński for the discussions during our meetings.
- This work is co-financed in part supported by the Ministry of Science and Higher Education (Agreement Nr 2023/WK/07) and by the program of the Ministry of Science and Higher Education entitled "PMW".
- We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016410.

Thank you!

You can contact us via
majerz@agh.edu.pl
mwojnar@agh.edu.pl