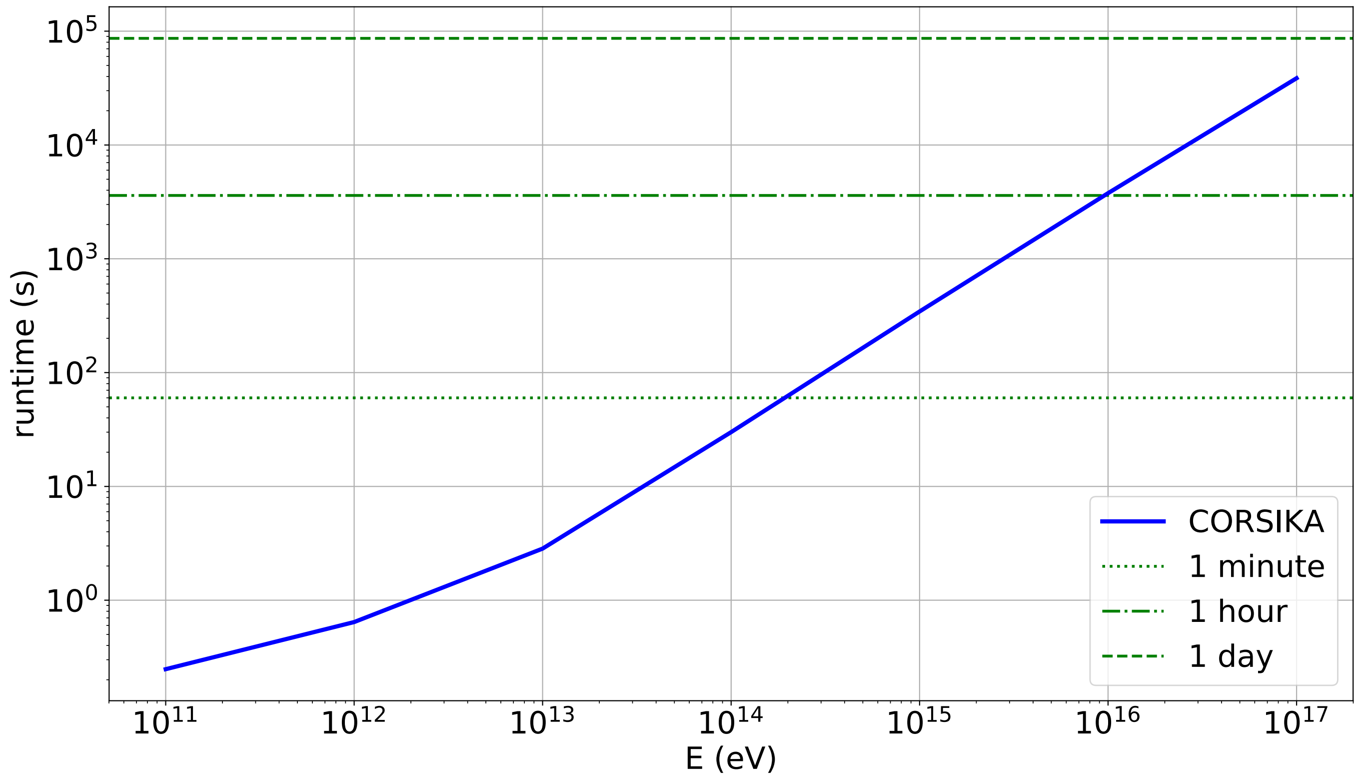
Speedup of Extensive Air Shower Simulations with Deep Neural Networks

Marcel Köpke, Dr. Markus Roth

CORSIKA Cosmic Ray Simulation Workshop Karlsruhe (2019)

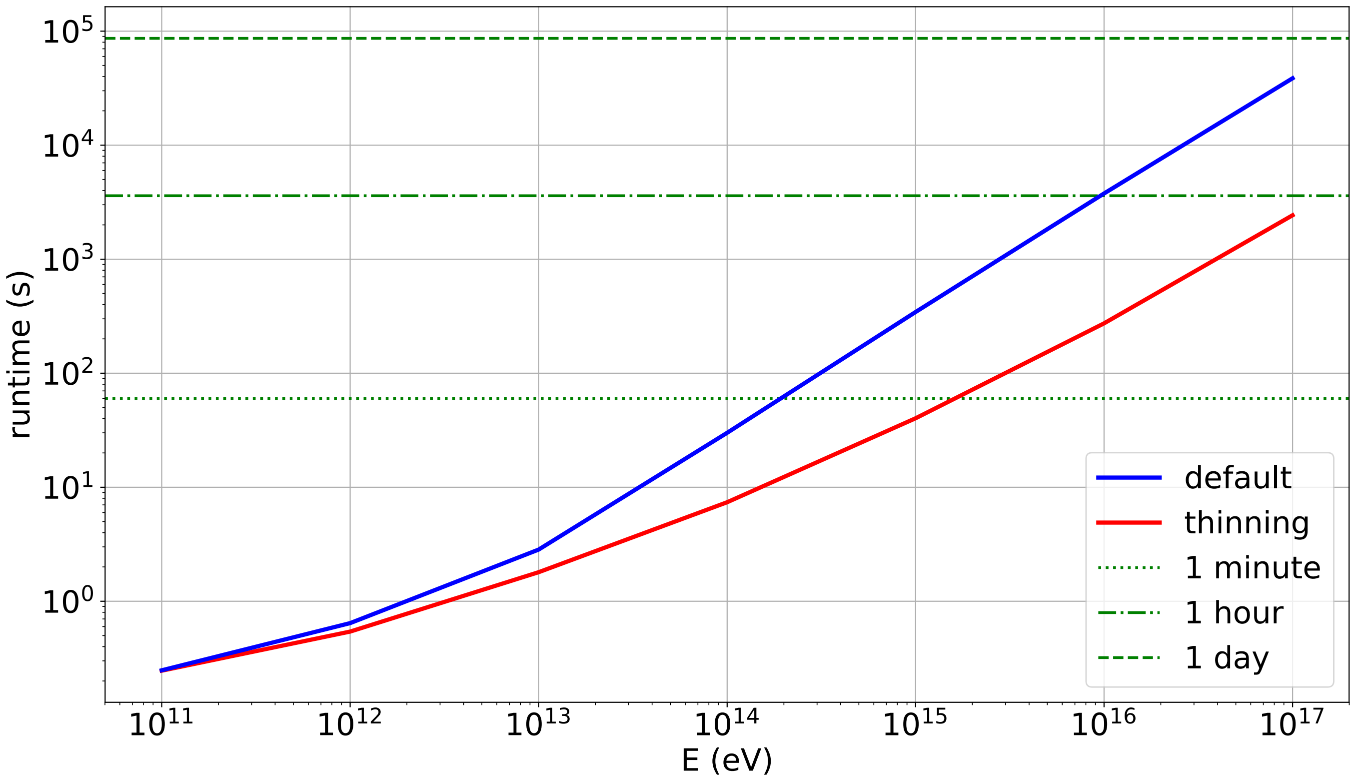
INSTITUTE FOR NUCLEAR PHYSICS (IKP), FACULTY OF PHYSICS
KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT)

Motivation

- 

- The time complexity of CORSIKA 7 simulations rises approximately linearly with the primary particle energy

Thinning



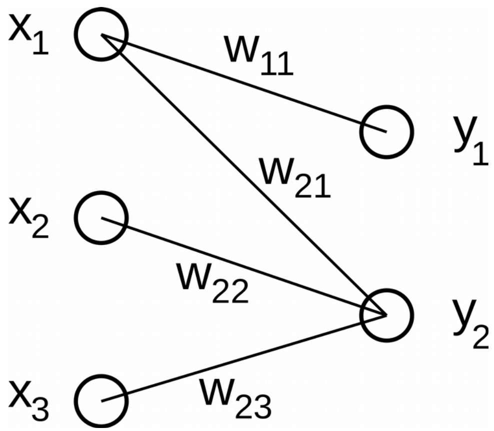
- Reduces (effective) particle content by particle-aggregation
- Preserves shower properties to leading order
- Reduces shower-to-shower fluctuations

Why Neural Networks?

Karlsruhe Institute of Technology

- Can run on specialized **hardware** (GPU / TPU)
- Automatic **parallelization** (TensorFlow)
- Reduction to **essential features**
- Training can **fix meta-parameters**
- **Adjustable accuracy** possible

Neural Networks

- 

$$y_1 = w_{11} \cdot x_1 + w_{12} \cdot x_2 + w_{13} \cdot x_3$$

$$y_2 = w_{21} \cdot x_1 + w_{22} \cdot x_2 + w_{23} \cdot x_3$$

$$\vec{y} = w \cdot \vec{x} \ (+\vec{b})$$

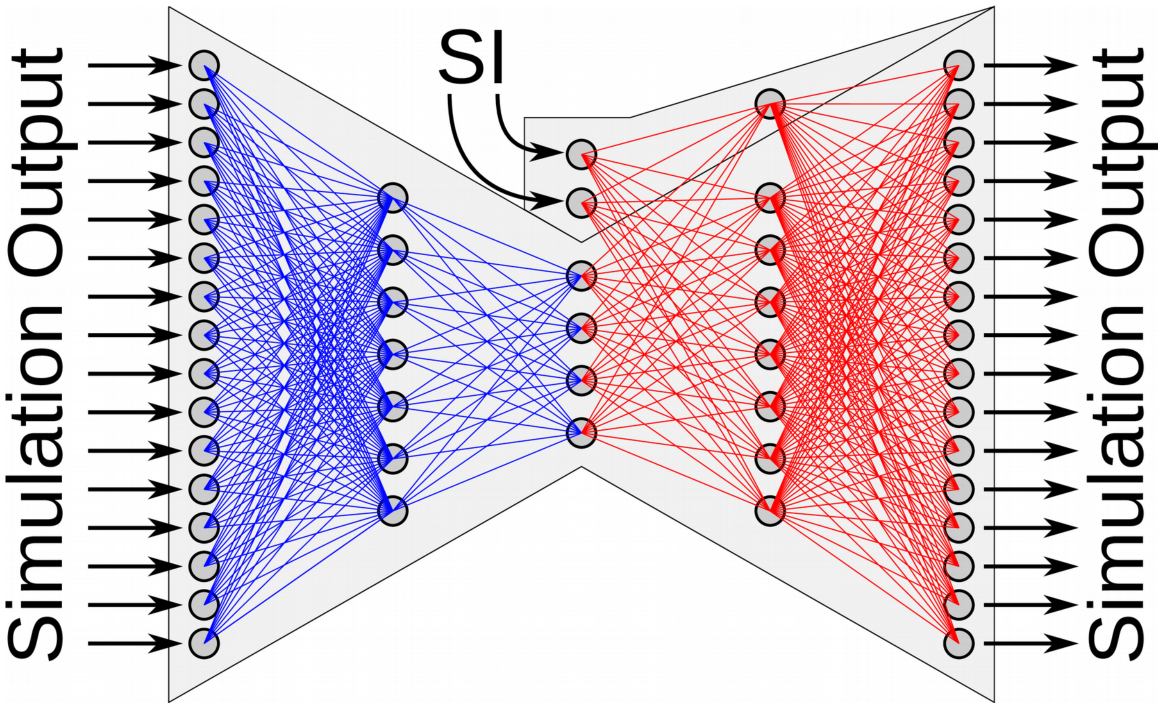
$$\vec{f} = a(\vec{y})$$

- Combination of linear and non-linear functions
- Training via loss function / metric on data pairs $(\vec{x}, \vec{t})$
- $L = L(\vec{f}(\vec{x}), \vec{t}) \quad \Rightarrow \quad w' = w - \alpha \cdot \nabla_w L$

Institute for Nuclear Physics (IKP), Faculty of Physics
Karlsruhe Institute of Technology (KIT)

Generative Neural Networks

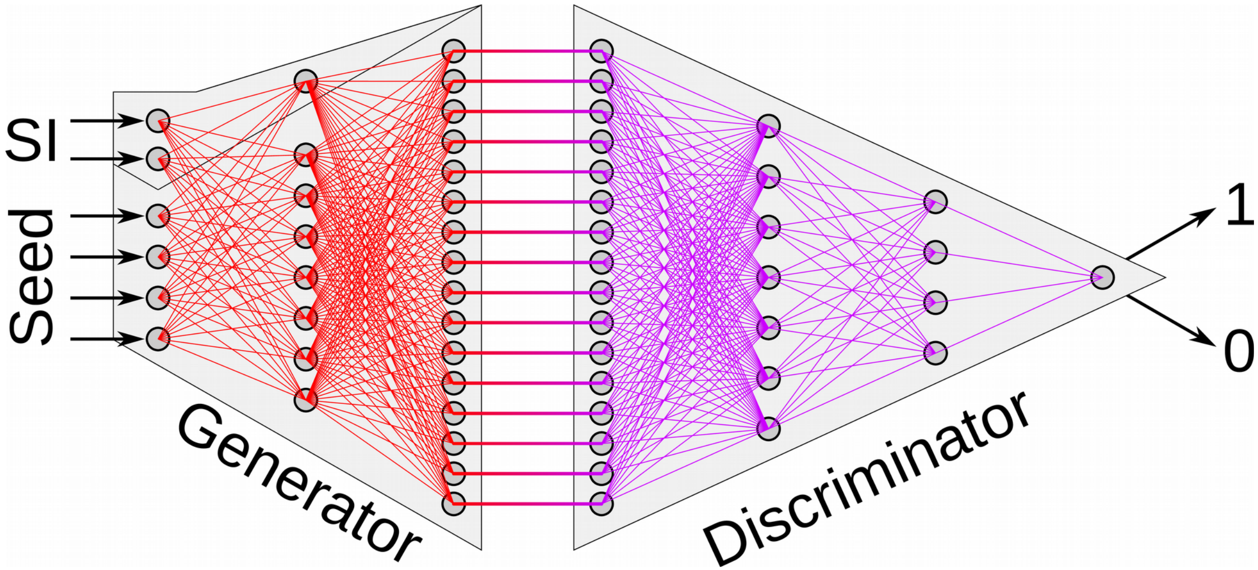
- Autoencoder



- Problem: Explicit definition of metric

Generative Neural Networks

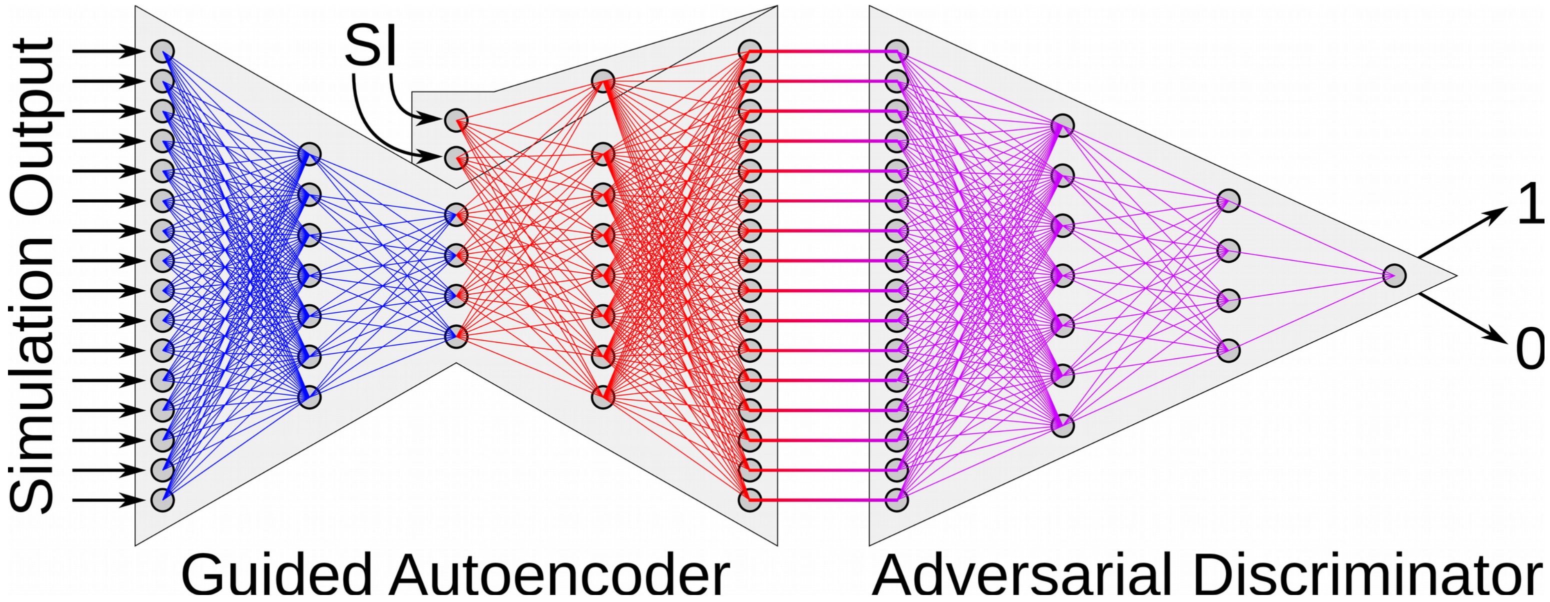
- Adversarial Neural Network



- Train discriminator on real (1) and generated (0) data
- Train generator to outsmart the discriminator

Fast Implicit Simulation Heuristic

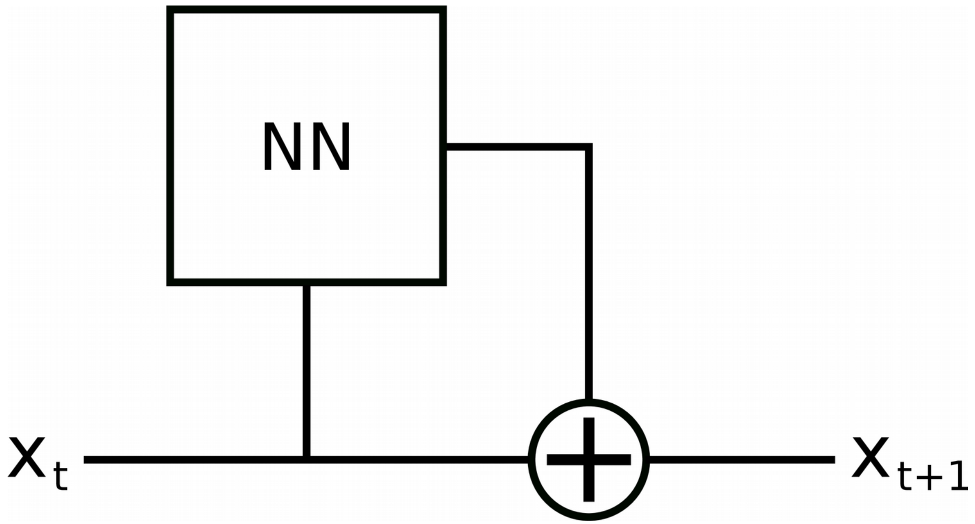
- Autoencoder with Adversarial Metric



- Simulation Input (SI) can be extended with meta-parameters
- Discriminator can be refined with real measurements

Adjustable Accuracy [1]

- ResNet



- Translate to ordinary differential equation (ODE)

$$x_{t+1} = x_t + f(x_t, \theta_t) \quad \Rightarrow \quad \frac{dx(t)}{dt} = f(x(t), t, \theta)$$

- Solve with standard ODE solver

- Adapt solver accuracy on the fly (training: high, inference: low)

Working Example [2]

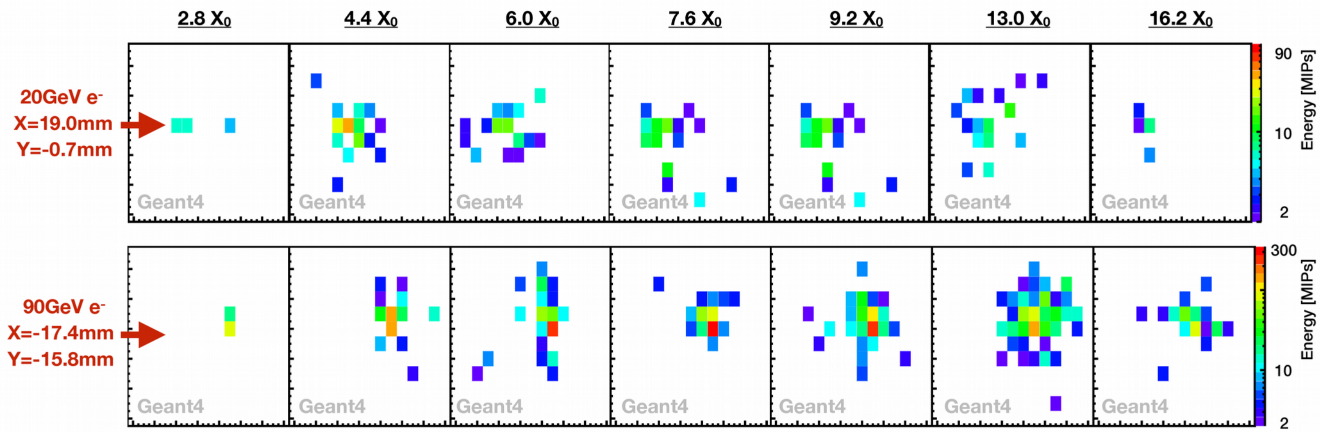
- „Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network“ - Martin Erdmann, Jonas Glombitza, Thorben Quast – arXiv: 1807.01954

  Some Highlights:

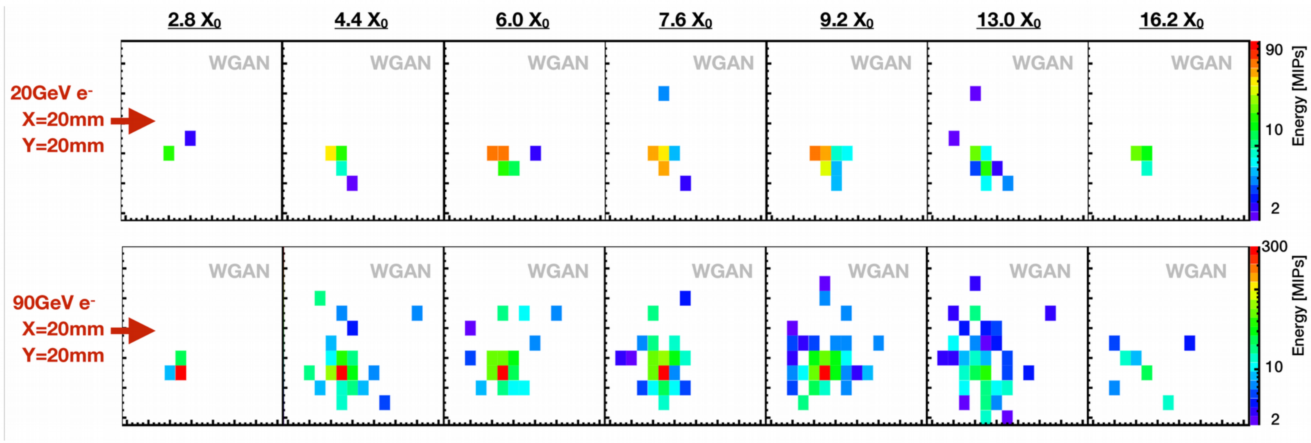
- **Wasserstein GAN** for HEP Geant4 simulations
- **Interpolation** between training energies
- Up to **x6660 faster** than Geant4
- Runtime is **independent of particle energy**

Geant4 vs. Wasserstein GAN [2]

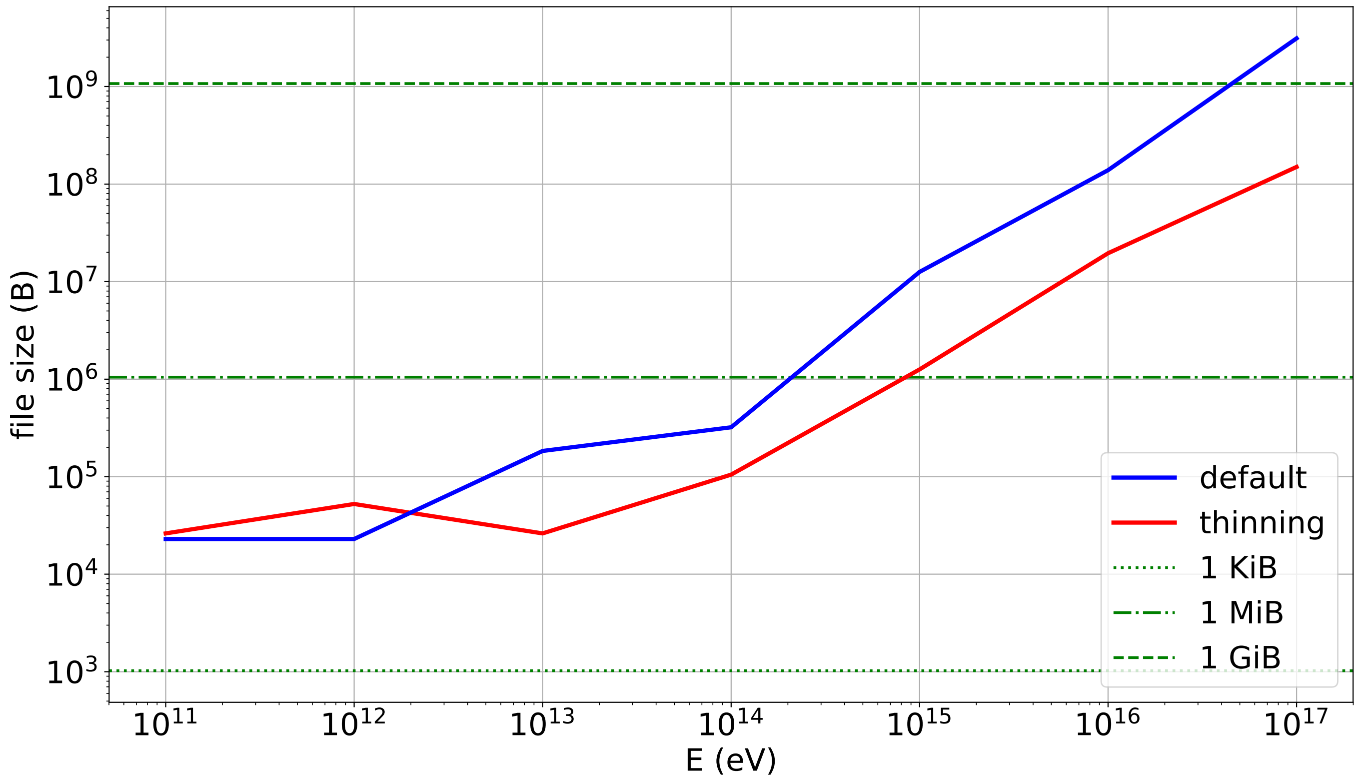
- Geant4:



- Wasserstein GAN:



Integration into CORSIKA 8



- Shower library needed for analyses and model training
- Trained model = effective compression of shower library
- Integrate training into CORSIKA 8? $\Rightarrow$ Transmit model updates?

Summary

- Deep Neural Networks can serve as an effective simulation heuristic
- Can improve runtime and (effective) data compression
- Can employ specialized hardware
- Trained models can be conditioned on meta-parameters
- Adjustable accuracy is possible
- ErUM: Innovative Digitale Technologien zur Erforschung von Universum und Materie

References

- Title picture:
  Karlsruhe Castle - Meph666 [CC BY-SA 3.0]
  https://commons.wikimedia.org/wiki/File:Karlsruhe-Schloss-meph666-2005-Apr-22.jpg

- [1] „Neural Ordinary Differential Equations“ - Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud – arXiv: 1806.07366

- [2] „Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network“ - Martin Erdmann, Jonas Glombitza, Thorben Quast – arXiv: 1807.01954