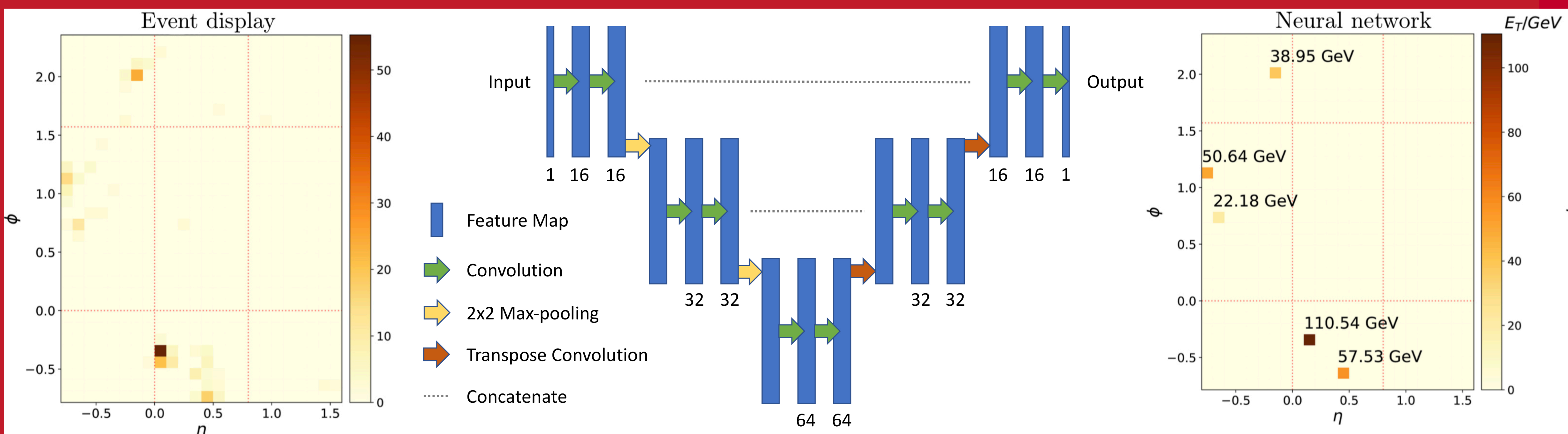
Development and implementation of deep neural networks close to sensors for object reconstruction and identification

Christian Schmitt (Mainz)



Cluster of Excellence PRISMA+ Precision Physics, Fundamental Interactions and Structure of Matter

Bundesministerium für Bildung und Forschung

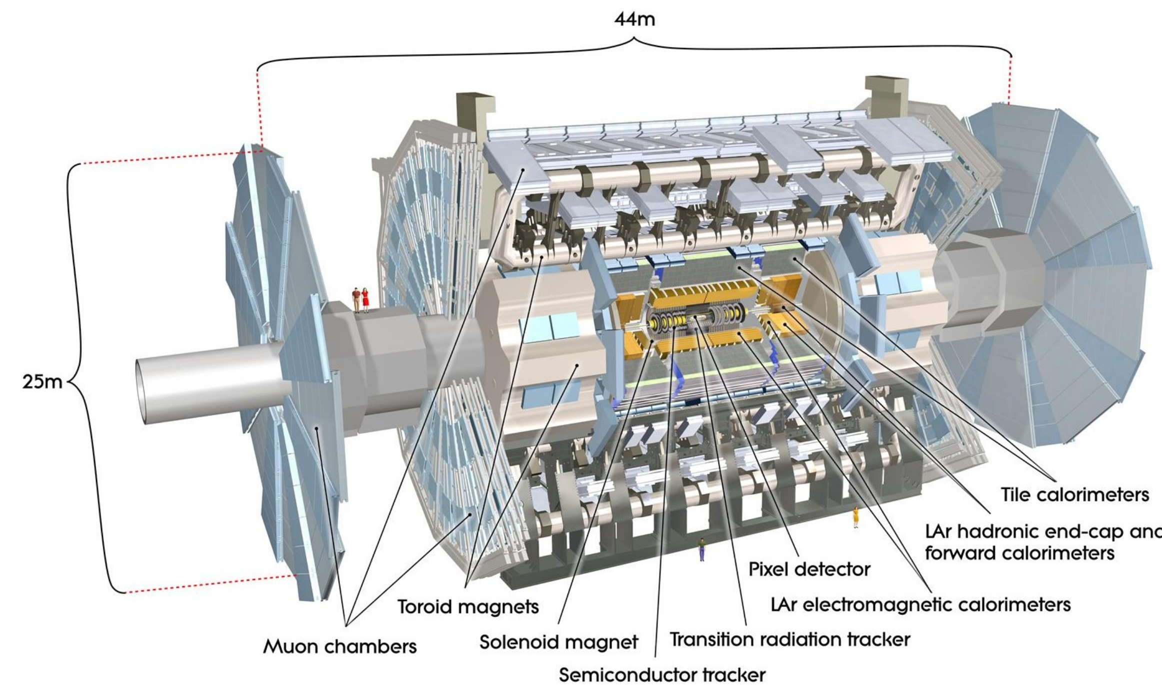
JG|U

Aim of the project in Mainz

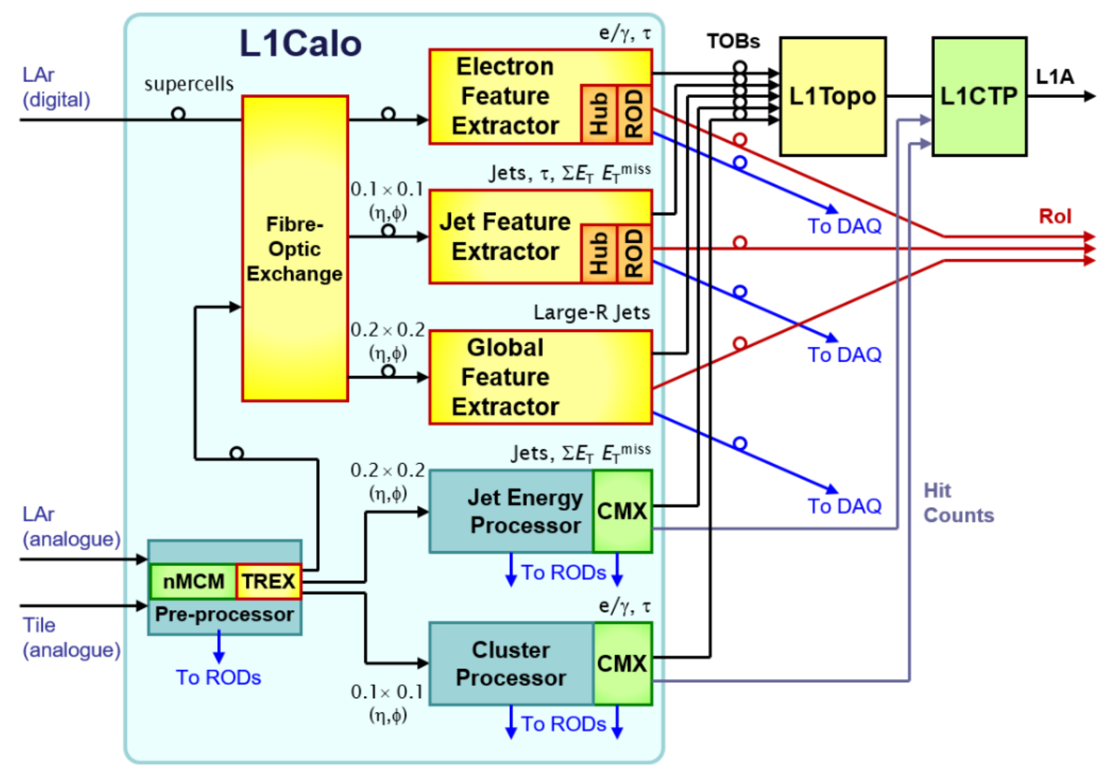
- **Processing of detector data at extremely high rates**
  - Not possible to store data due to its size
  - Usage of GPUs not possible due to their too high latency
  - Data has to be processed and filtered locally, maybe directly at the corresponding sensors
- **Solution: deep neural networks** as replacement for iterative algorithms, that can be efficiently evaluated on **FPGAs**
- **Test environment: ATLAS L1 Trigger** (40 MHz rate)

Mainz

- Participation in ATLAS L1-Trigger
  - Expertise on the development and implementation of algorithms on FPGAs



- ATLAS physics analysis
  - Expertise on machine learning techniques



- Close collaboration with the Computer Science Institute

Example: jet reconstruction

- Optimal reconstruction of jets: anti-kT algorithm
  - Iterative algorithm
  - Direct implementation on FPGAs possible, but latency too high (O(μs))
    - Bachelor thesis N. Nottbeck (Mainz, 2016)
  - Not possible to perform in real time, only simplified algorithms are currently in use in the ATLAS L1 Trigger ("sliding window")

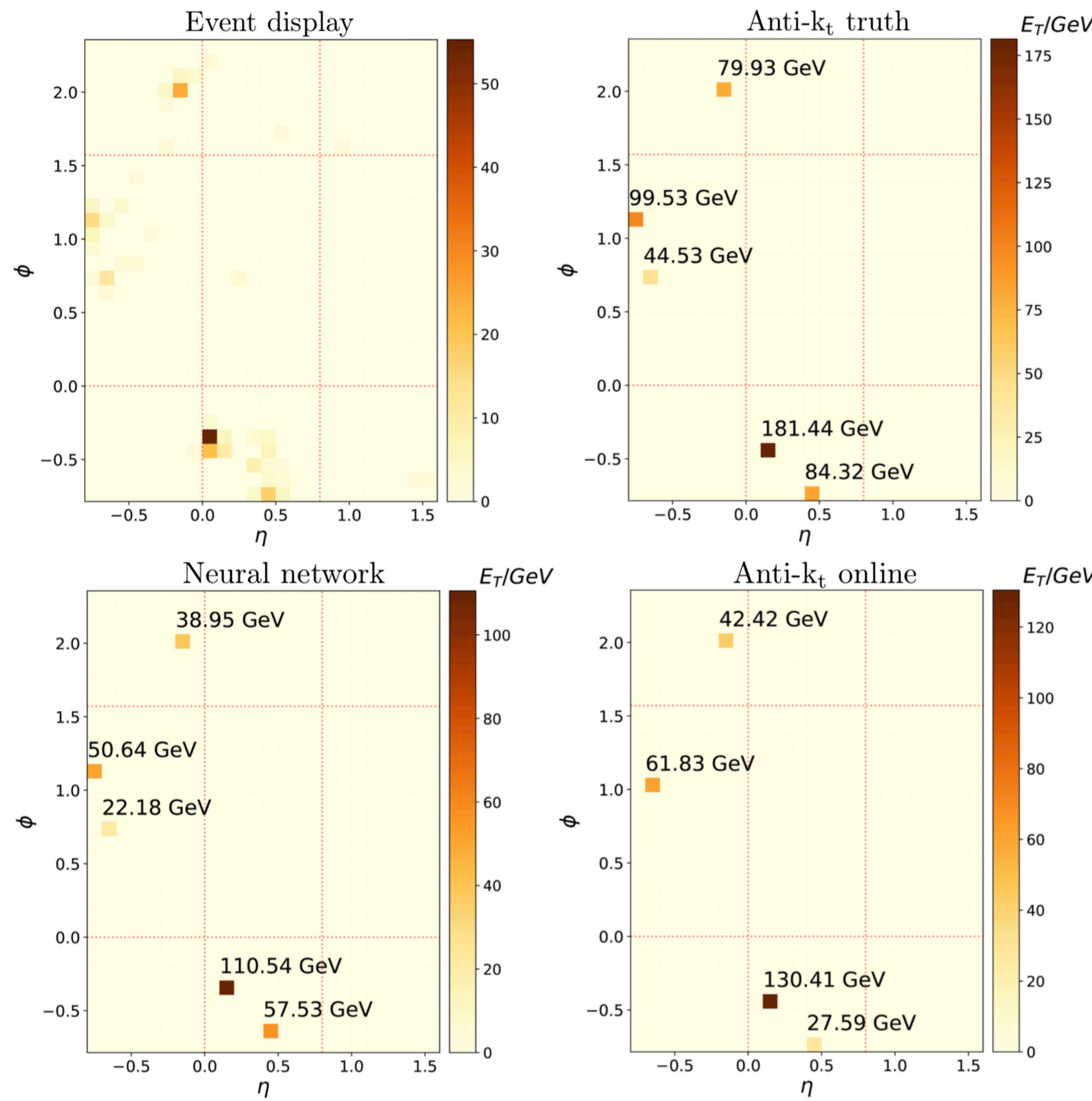
$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2} ,$$

$$d_{iB} = k_{ti}^{2p} ,$$

⇒DNN as replacement for anti-kT algorithm

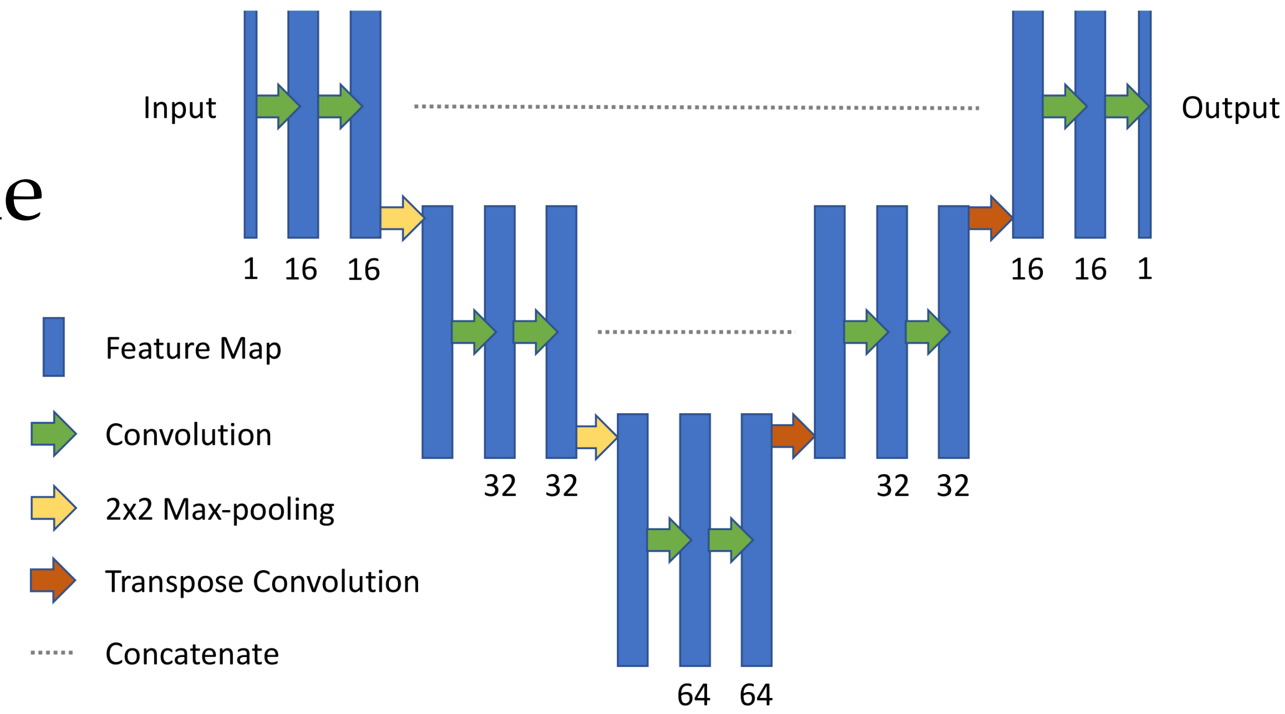
DNN instead of Anti-kT

- Current status:
  - Jet reconstruction via DNN
  - Image recognition based on calorimeter images (Trigger towers)
  - DNN is able to reconstruct overlapping jets better than anti-kT



DNN Details

- DNN has to reconstruct **Position** as well as **Energy** of **arbitrary number** of jets!
  - Much more complex than simple image classification tasks

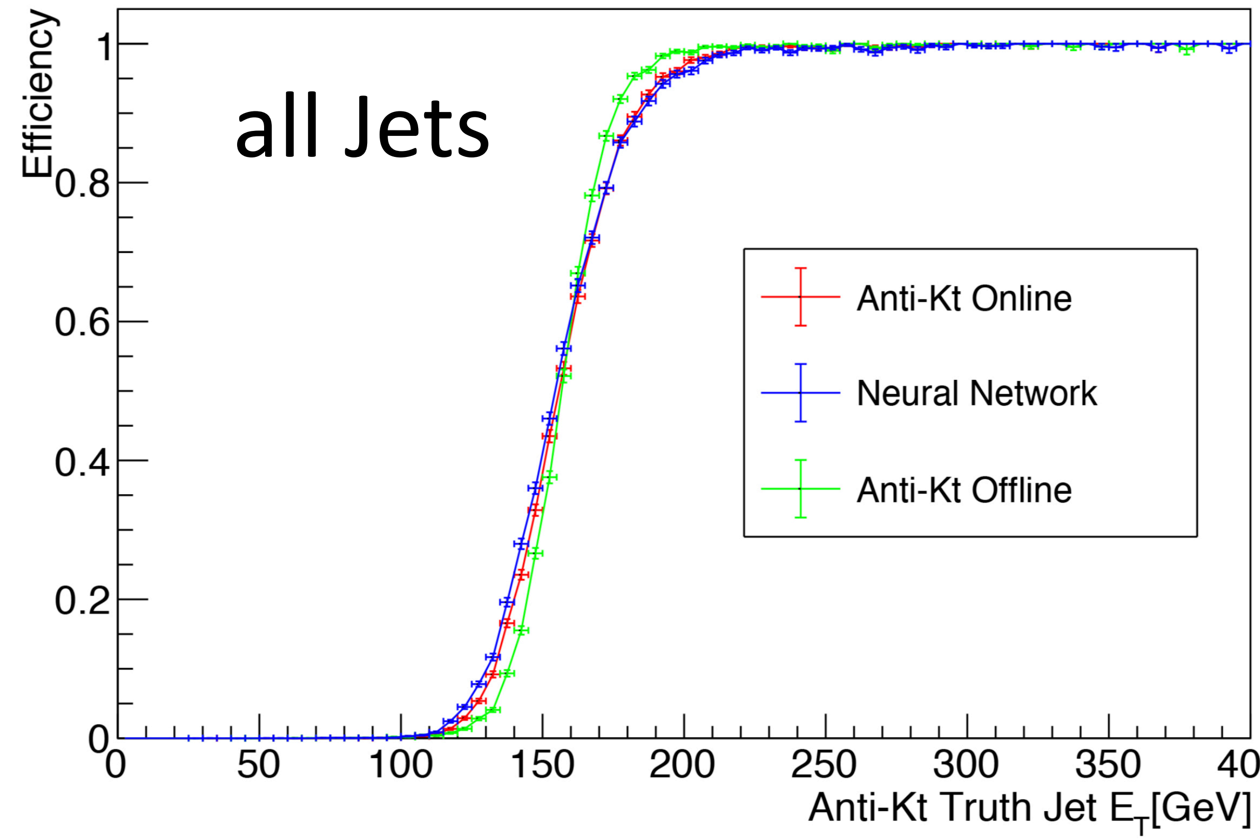


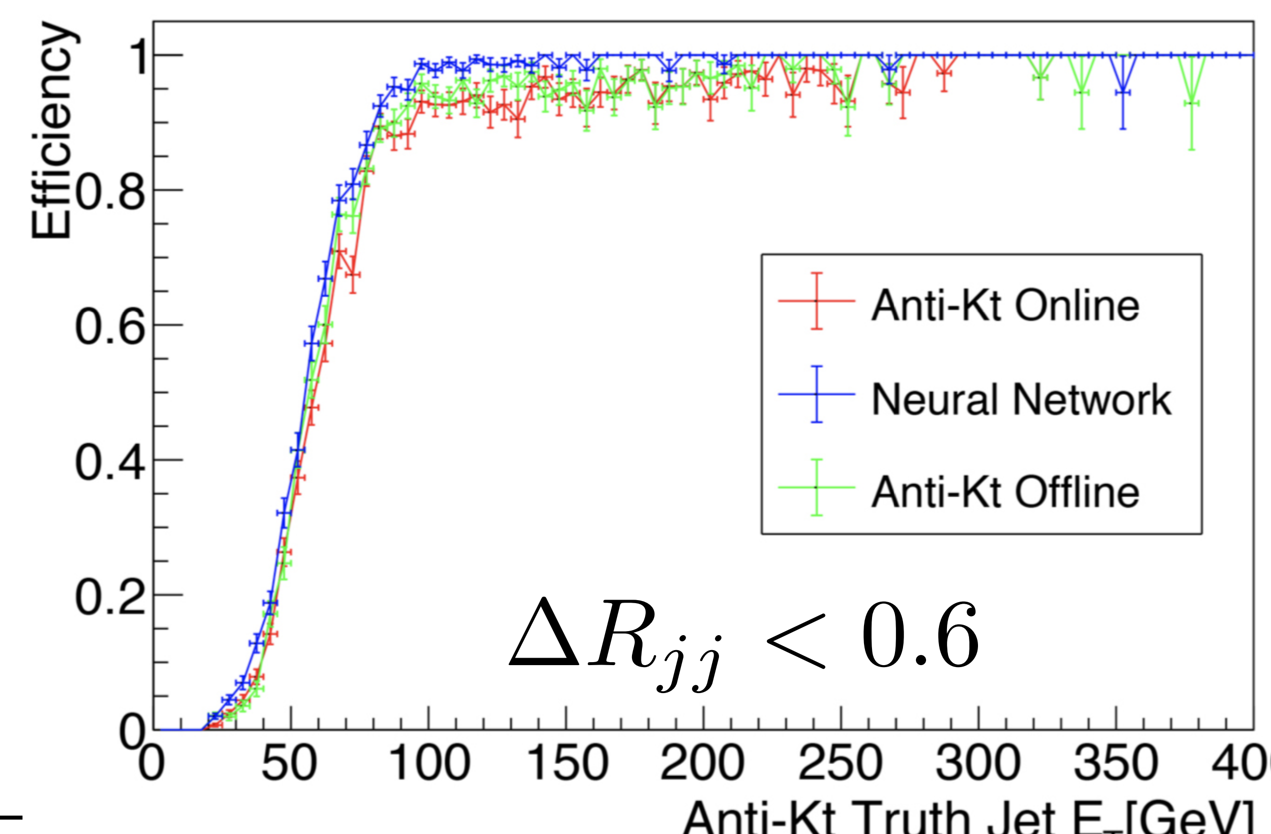
- Ansatz: create new image with jet information from calorimeter image (2D-conv with **U-net** architecture)
- **Additional connections between the layers avoids information loss by down-sampling**
- **ReLU as activation function (performant and ideal for FPGA!)**
- 3x3 kernel size for all layers $\Rightarrow$ total of 116881 weights

DNN results

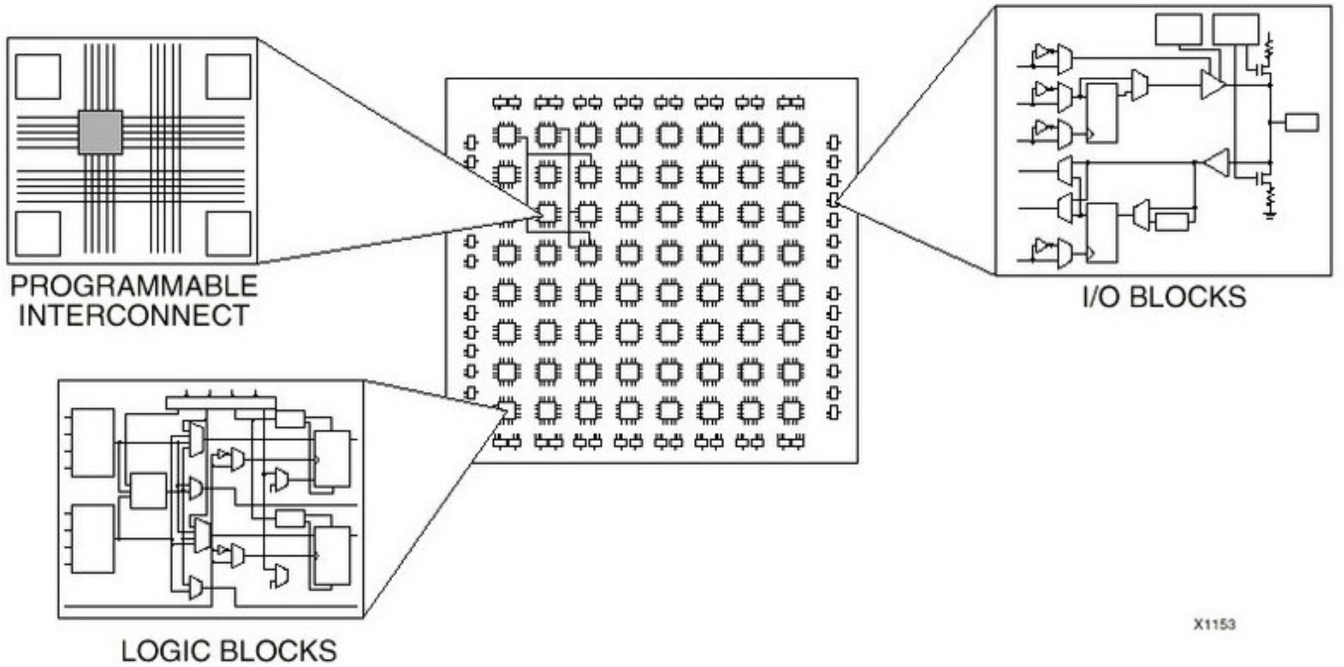
- Trained on ttbar events using own fast detector simulation und μ=0
- Results for fully simulated HH->4b events (μ=60)
  - Identical performance as anti-kT!
  - Overlapping Jets: better than anti-kT ans even **better than anti-kT running on full detector resolution!**

- **Shows huge potential of such DNNs (even outside the trigger environment)**





Implementation on FPGA

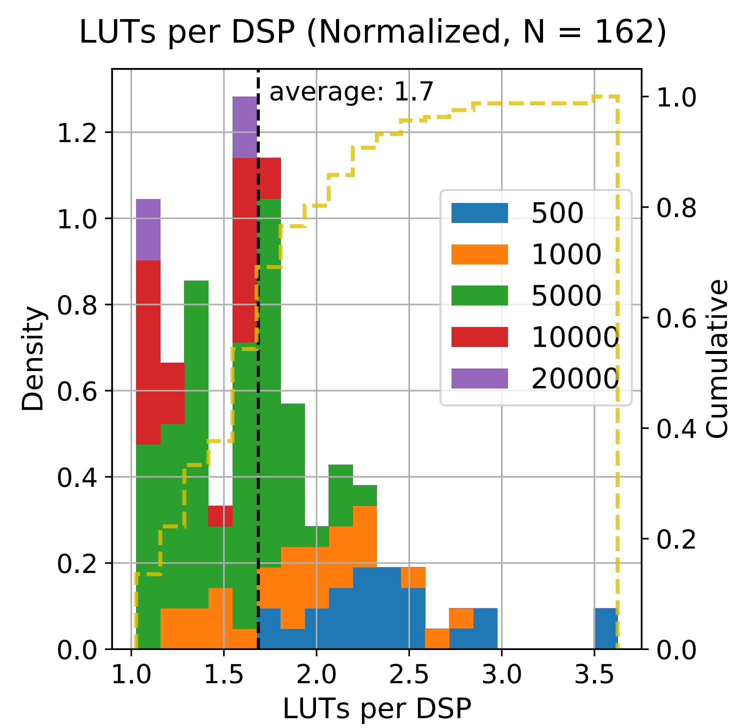


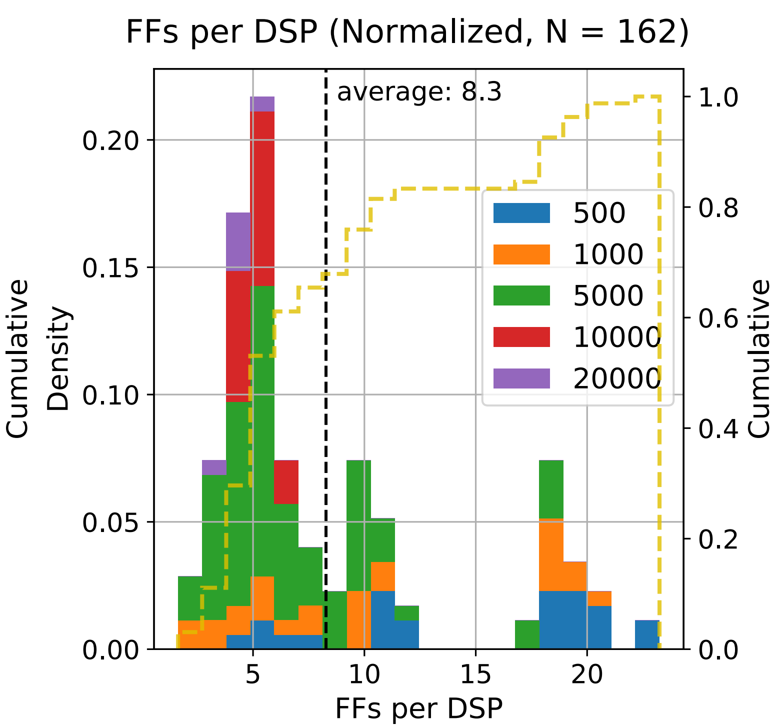
- FPGAs
  - Configurable integrated circuits with dedicated special units for e.g. multiplications (DPSs)
- Advantages for this project compared to CPU or GPU:

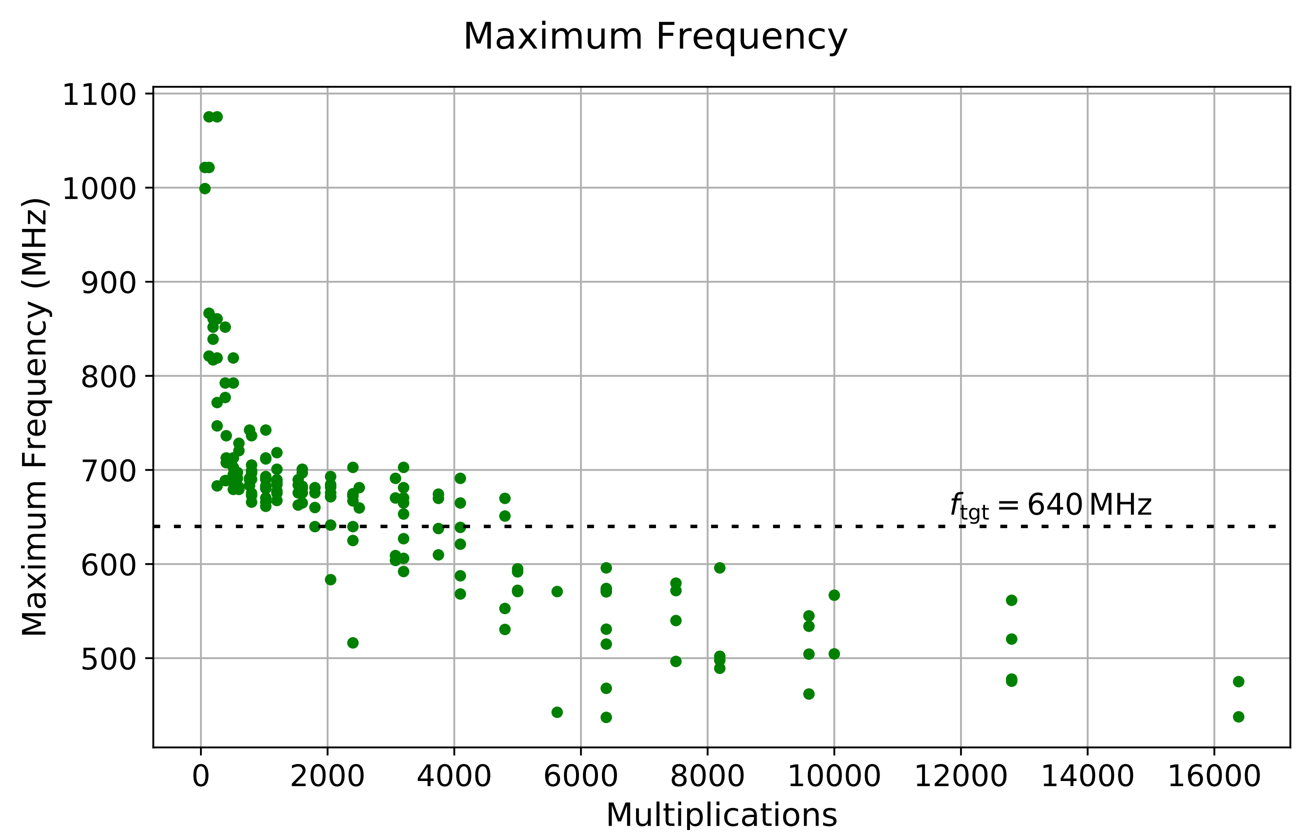
  very short latencies possible and **guaranteed timing**
- **Challenges:**
  - Adaptation of network architecture needed
  - Arithmetic precision, signal propagation delay, resource usage on the FPGA, …

First results for dense layers

- Dense layer successfully implemented on FPGAs
  - Calculations in DSPs
  - Only small additional resource overhead (LUT, FF)
  - High frequencies can be reached
- Target FPGA: Xilinx US+9P
  - 6840 DPSs; 2.4M FF, 1.2M LUT
  - ‘mid-range’ US+







Implementation on FPGAs

- Long term goal:
  - **Universal framework to implement DNN on FPGAs**
  - Starting point: pre-trained DNN (e.g. with Keras)
  - Framework creates VHDL code and all other files needed for implementation on FPGA
  - Other parameters: e.g. desired arithmetic precision for the calculations on the FPGA, chosen maximal latency / minimal frequency for the implementation

Plans for the next 2,5 years

- **Development of deep network architectures as replacement for iterative algorithms**
  - anti-kt jets, missing transverse energy, jet substructure
- **Adaptations and Improvements of existing deep neural network methods**
  - Optimal performance with limited resources (FPGAs)
- **Adaptations on FPGA resources and validation**