Object Condensation with Graph Neural Networks for the ECL Trigger - Status and Implementation

Belle II Germany Meeting

Isabel Haide, Marc Neu, Timo Justinger, Torben Ferber | Tuesday 1st October, 2024

AI Trigger Group at Belle II

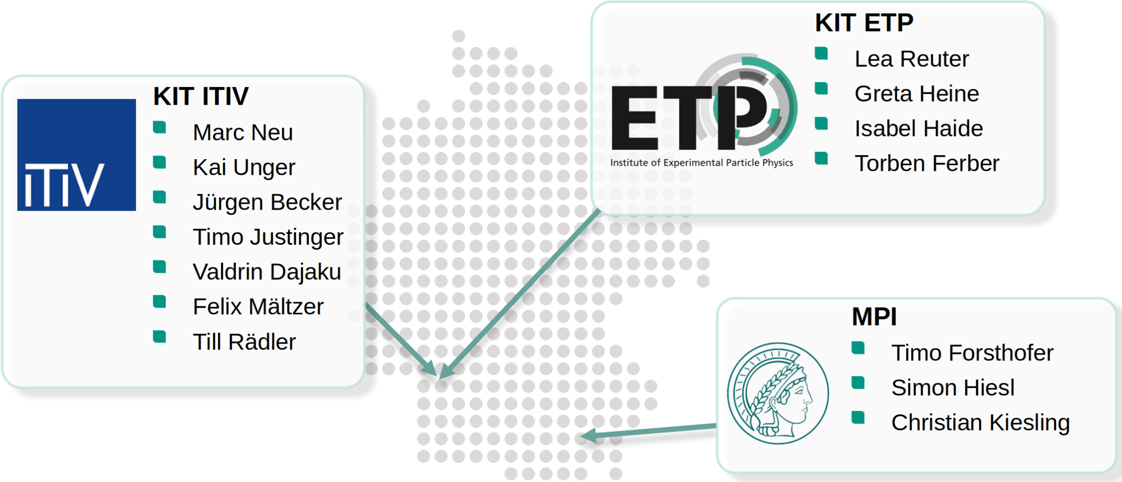
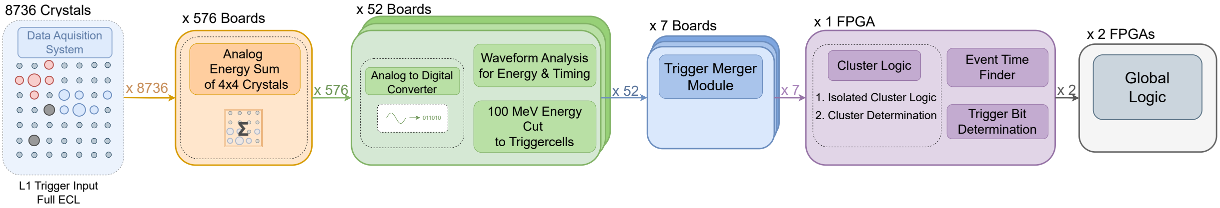


Figure: Greta Heine

Current Electromagnetic Calorimeter (ECL) Trigger



1.) All 8736 crystals are read out

2.) The energy of 16 crystals is analogously summed into one trigger cell (TC)

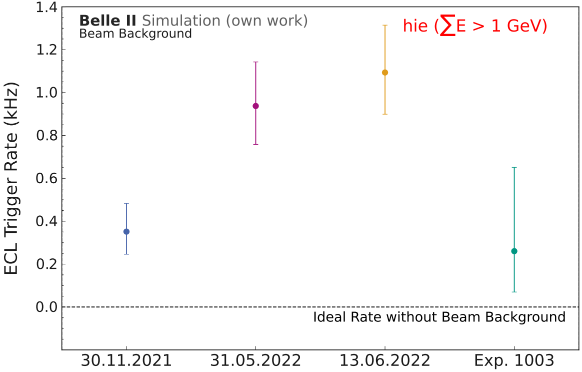
3.) A 100 MeV energy cut is applied to each TC to reduce input size

4.) Clusters are found through isolating cluster logic

5.) Event Timing for all triggers is defined through highest energetic cluster

6.) Trigger lines for Bhabha rejection and low multiplicity events are calculated

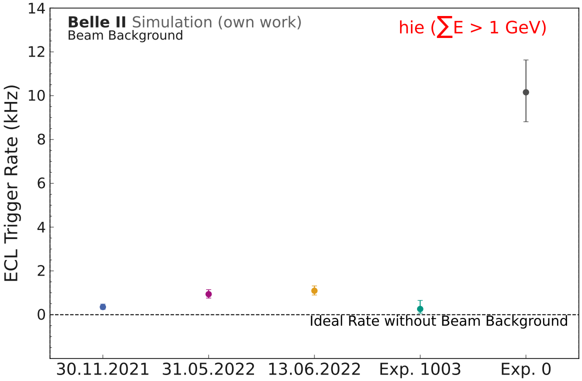
Beam Background in the ECL



- hie: Total energy sum in inner ECL > 1 GeV
- Simulation containing only beam background taken from data already increases trigger rate of single line to > 1 kHz
- Maximum possible total trigger rate = 30 kHz

$\Rightarrow$ **Trigger algorithm has to be adapted to rising background conditions**

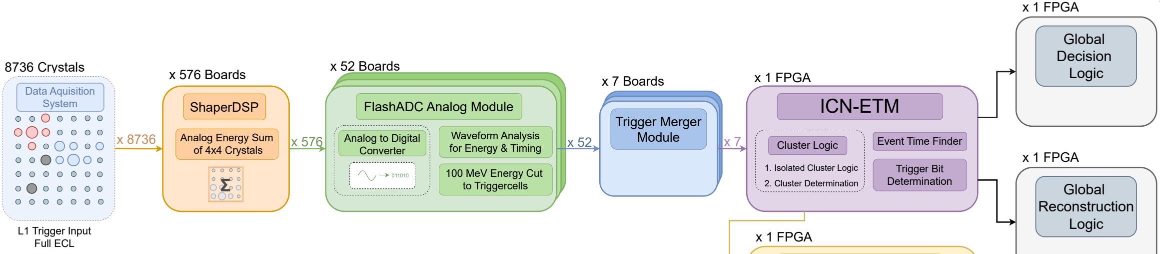
Beam Background in the ECL

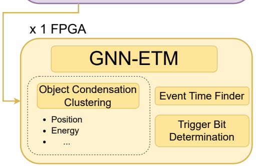


- hie: Total energy sum in inner ECL > 1 GeV
- Simulation containing only beam background taken from data already increases trigger rate of single line to > 1 kHz
- Maximum possible total trigger rate = 30 kHz

$\Rightarrow$ **Trigger algorithm has to be adapted to rising background conditions**

GNN for the ECL Trigger - Implementation

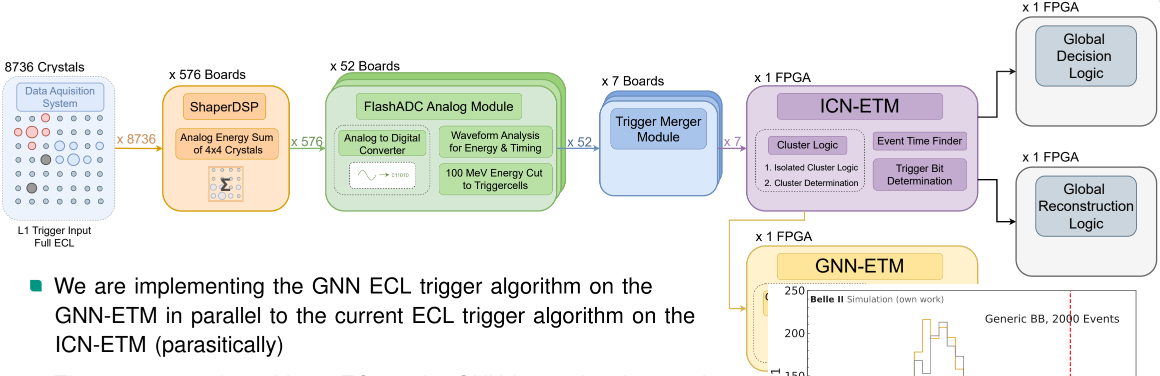


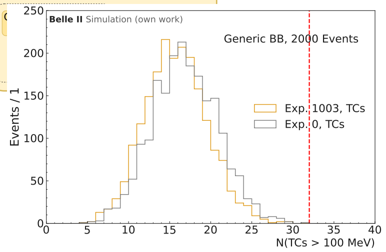


- We are implementing the GNN ECL trigger algorithm on the GNN-ETM in parallel to the current ECL trigger algorithm on the ICN-ETM (parasitically)
- The current number of input TCs to the GNN is restricted to 32 due to hardware restrictions
- Algorithm will store predicted clusters and their properties in raw data but will not be involved in trigger decisions

$\Rightarrow$ **Starting to run this in two weeks!**

GNN for the ECL Trigger - Implementation





- We are implementing the GNN ECL trigger algorithm on the GNN-ETM in parallel to the current ECL trigger algorithm on the ICN-ETM (parasitically)
- The current number of input TCs to the GNN is restricted to 32 due to hardware restrictions
- Algorithm will store predicted clusters and their properties in raw data but will not be involved in trigger decisions

$\Rightarrow$ **Starting to run this in two weeks!**

GNN Algorithm for the ECL Trigger

Current Situation:

- The current trigger uses a fast and easy clustering algorithm which returns a cluster as soon as a TC has a reconstructed energy above 100 MeV
  $\rightarrow$ Works very well in current conditions, with low(er) beam background levels and big (16 crystal) TCs
- For future beam background levels, trigger rates might rise drastically for only beam background
  $\rightarrow$ Trigger lines have to be prescaled and analyses could lose efficiency
- Current trigger algorithm additionally is by design not able to separate overlapping clusters, which can limit the efficiency for analyses such as $e^+e^- \rightarrow a\ (\rightarrow \gamma\gamma)\ \gamma$

GNN Trigger for the ECL:

- Including state-of-the-art algorithms, such as Graph Neural Networks (GNN), in the development of new triggers can increase trigger efficiency greatly
- Implementing the GNN Trigger algorithm now (for Run Period 2024c, parasitically) will help us understand the opportunities and challenges without changing the current trigger algorithm

Network Design of GNN ECL Trigger Algorithm

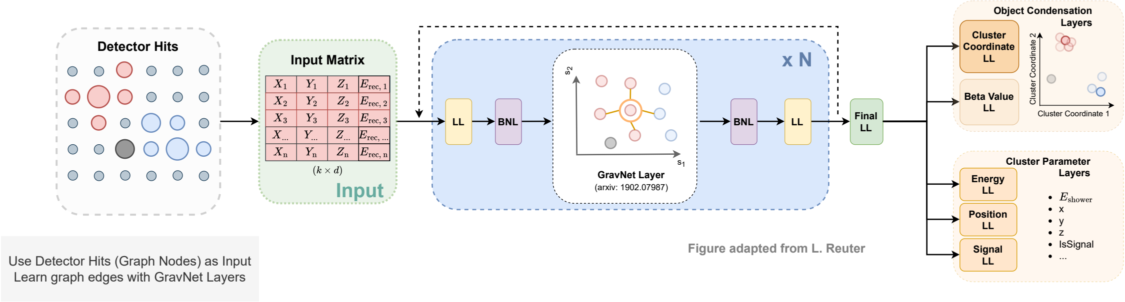


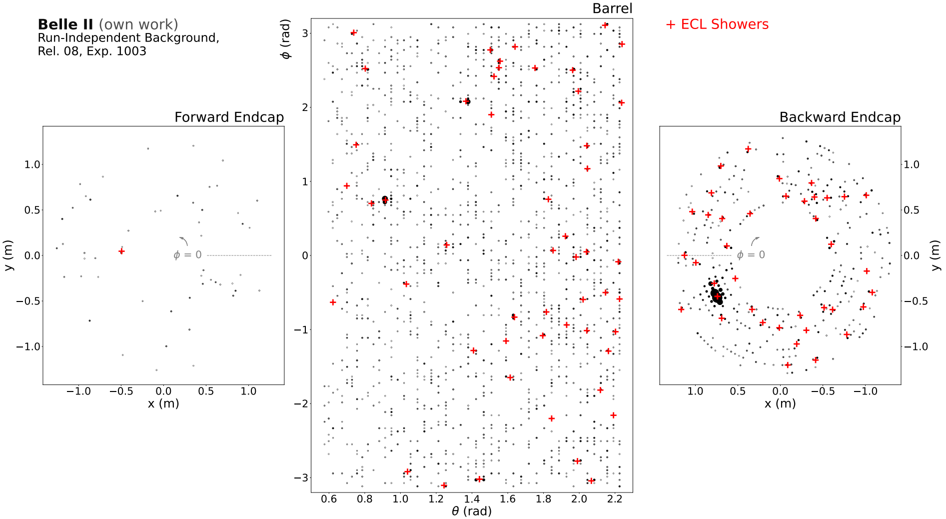
Figure adapted from L. Reuter

- Object Condensation (OC): One-shot algorithm for both detection and reconstruction of clusters (arXiv:2002.03605)
- Irregular geometry and varying input sizes in the ECL $\rightarrow$ **Graph Neural Networks (GNN)**
- OC algorithm is adaptable to **different beam backgrounds** and can be (for future upgrades) adapted to different inputs
- Implementation on FPGA requires max. 2 GravNet blocks and reduction of linear layers
- Replacement of Euclidean distance for k-Nearest-Neighbour algorithm with Manhattan distance to reduce needed resources
- Inputs and weights additionally need to be quantized for actual implementation

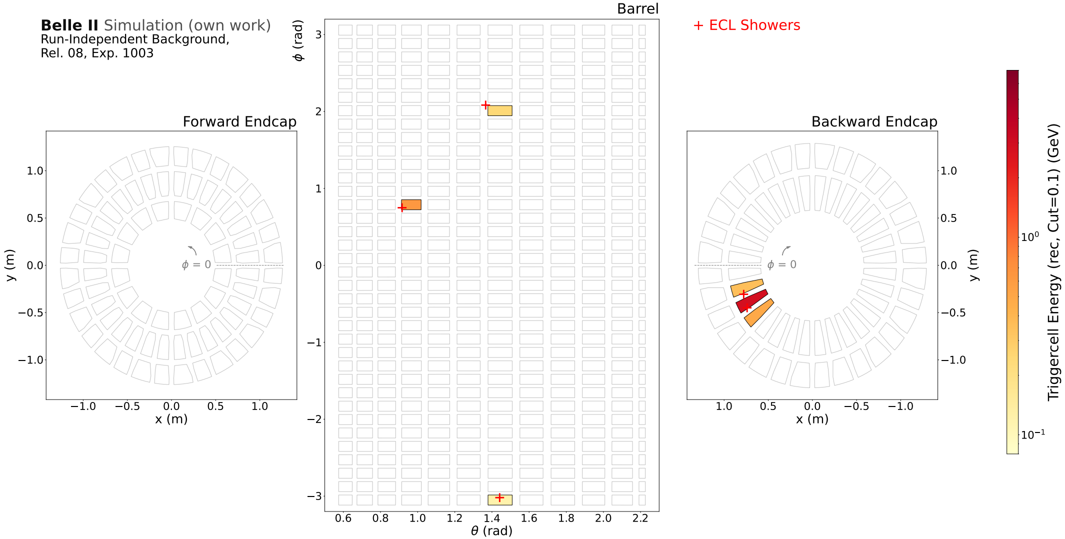
Training Samples and Training Target

- As training inputs I am using only TSIM TCs with timing within the TRG timing window (256ns) given by basf2
- The TC energy and timing is taken from TSIM, while MC truth information is taken from ECL simulation
- The target for training is offline ECL showers due to two reasons:
  - Anything we cannot reconstruct offline will also be not a target for online reconstruction
  - Eventually, ECL showers can allow us to train on data to reduce MC effects in the training sample
- The current training is done on a 50/50 mixture of two categories of simulated events to create an unbiased training sample:
  1.) 1-6 photons with energy 0.05-7 GeV, additionally adding low energetic photons following the background distribution (more detail next slides)
  2.) 1-6 photons with energy 0.05-7 GeV, additionally adding two photons that have a small opening angle
- Input Features: Reconstructed energy per TC, timing (calibrated to highest-energetic TC per event), x, y, z position of TC from lookup table

Event Display - Crystals with ECL Showers

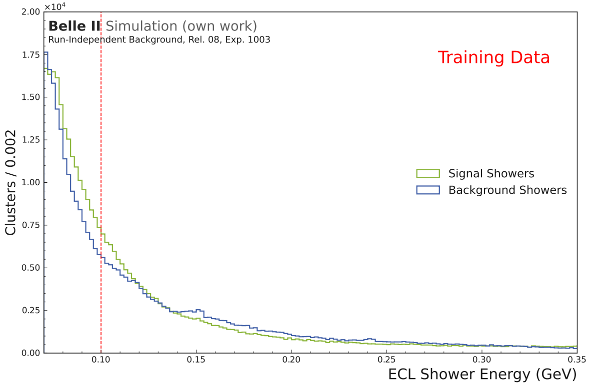


Event Display - Triggercells with ECL Showers



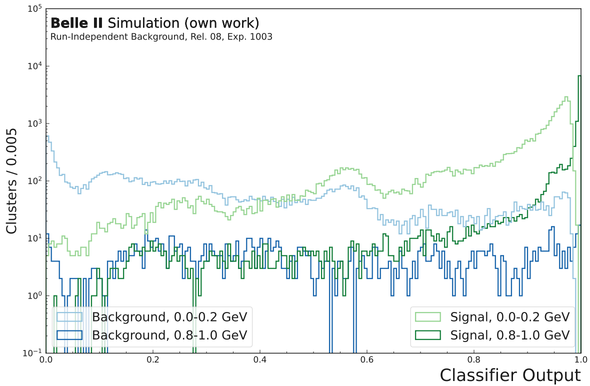
Signal/Background Showers

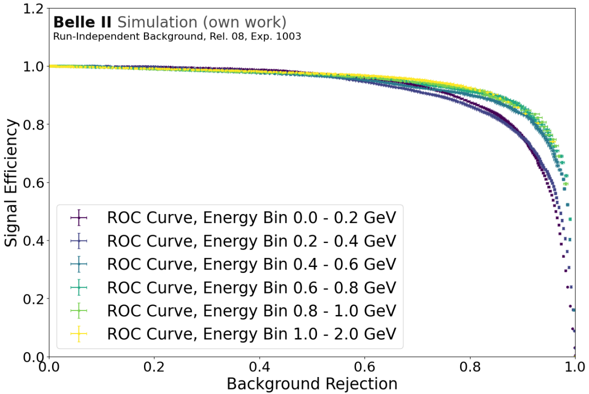
- Beam background already contains a lot of low-energetic showers
- If I use simulated samples for training where the energy distribution for the signal photons is uniform between 0.05 and 7 GeV, the low-energetic ECL showers are dominantly originating from beam background
  $\rightarrow$ Bias for network performance for signal/background classification
- Therefore I model beam background energy and number of showers to generate a (mostly) unbiased training sample
- A signal shower is defined analogous to the basf2 MC matching definition for clusters



Performance of Signal Classifier

Karlsruhe Institute of Technology





- Network can classify signal/background clusters on training data with high efficiency and high beam background rejection
- Signal definition has to be carefully defined, especially for charged particles (work in progress)

Optimization for Hardware Implementation

- Current hardware restrictions on available UT4 board allow a maximum of two GravNet layers
- Additionally, we optimize the network further to make implementation on FPGA easier/possible:
  - Replacement of euclidean distance with Manhattan distance for k-Nearest-Neighbour algorithm
  - Removal of linear layers and unnecessary calculations, as far as possible
  - Quantization of inputs and training weights

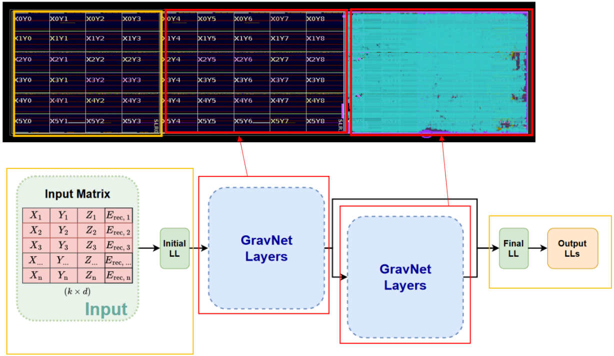


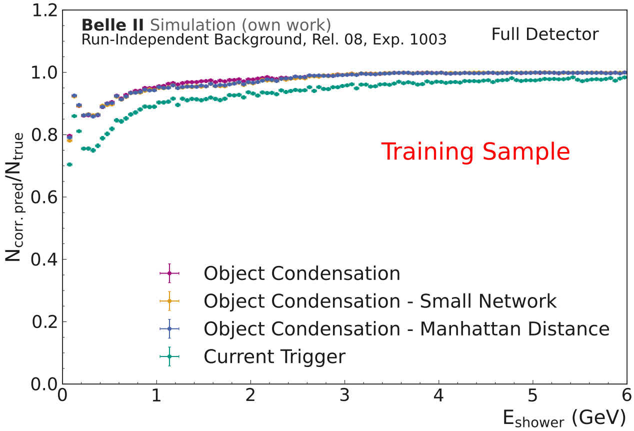
Figure: Marc Neu

Distance	LUTS	FF	BRAM	DSP	Total Power	Frequency
Euclidean	13.79%	9.24%	6.96%	30.64%	6.514W	256.4 MHz
Manhattan	14.87%	9.18%	6.96%	1.28%	4.468W	256.4 MHz

Table: Timo Justinger

Optimization for Hardware Implementation - Efficiency

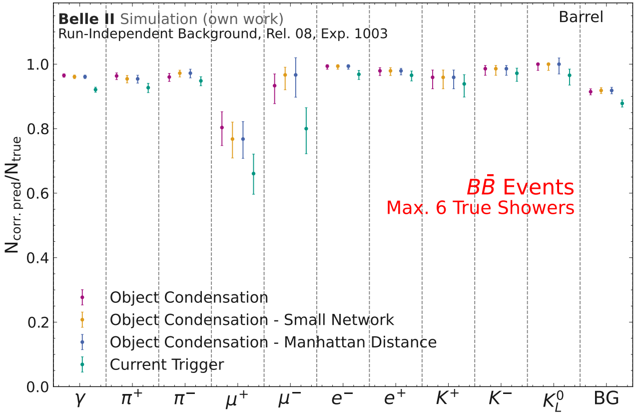
- Original network has been designed by broad hyperparameter optimisation (number of GravNet blocks was set to 2)
- The evaluation shown here is still on CPU and with non-quantized weights and inputs
- The network with reduced amount of linear layers (=small network) only has around 8000 tunable parameters, which is a reduction of factor 2 in free parameters
- Using Manhattan distance also does not impact network performance significantly but has great impact on FPGA implementation



- Efficiency: N(corr. pred.) / N(true)
  - True are all ECL showers visible on TRG level, correctly predicted are all GNN/TRG clusters truthmatched to a shower

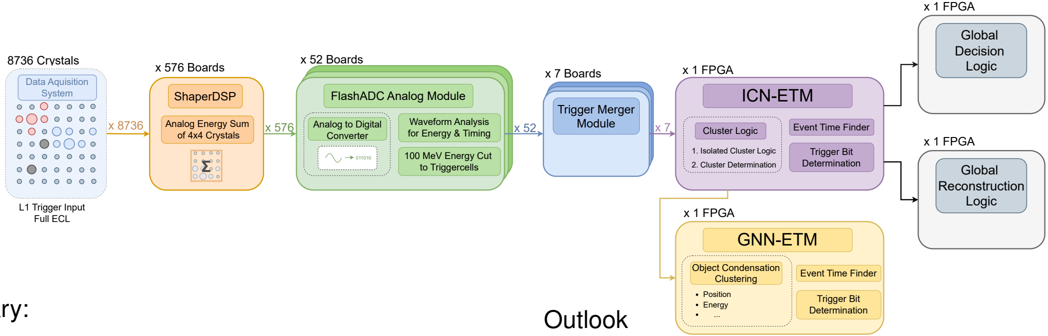
Optimization for Hardware Implementation - Efficiency

- Original network has been designed by broad hyperparameter optimisation (number of GravNet blocks was set to 2)
- The evaluation shown here is still on CPU and with non-quantized weights and inputs
- The network with reduced amount of linear layers (=small network) only has around 8000 tunable parameters, which is a reduction of factor 2 in free parameters
- Using Manhattan distance also does not impact network performance significantly but has great impact on FPGA implementation



- Efficiency: N(corr. pred.) / N(true)
  - True are all ECL showers visible on TRG level, correctly predicted are all GNN/TRG clusters truthmatched to a shower

Summary and Outlook



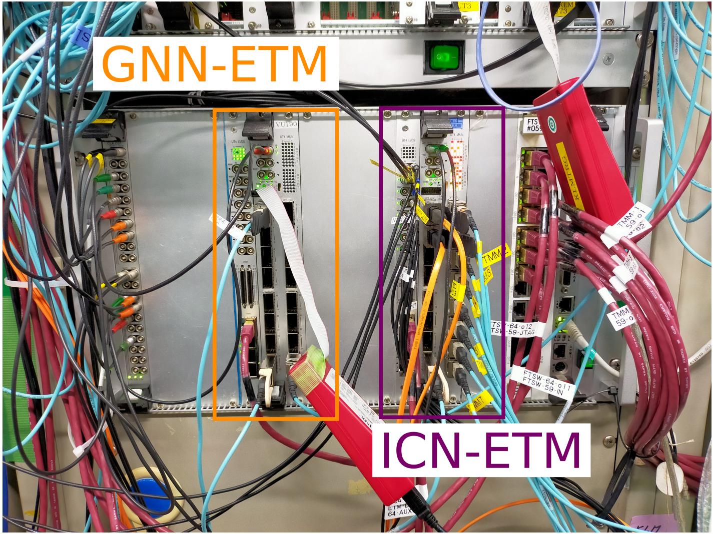
Summary:

- Implementation for new GNN ECL Trigger algorithm is planned for this run period, Marc is working on hardware realization
- Preprocessing, interface between ICN-ETM and GNN-ETM and Belle2Link DAQ is already done
- Optimization of network for hardware implementation by reducing network size and removing difficult calculations, such as euclidean distances

Outlook

- Network weights have to be quantized to 16bit, partially 8bit quantization
- First implemented model will hopefully take data within the first few runs of the next data-taking period
- Next step is then evaluating on the new raw data and comparing performance on physics data

Summary and Outlook



Backup

Backup

Full BB Efficiency for Different Particles