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

B2GM TRG Parallel Meeting

Isabel Haide, Marc Neu, Timo Justinger, Torben Ferber | Friday 4th October, 2024

AI Trigger Group at Belle II

KIT ITIV

- Marc Neu
- Kai Unger
- Jürgen Becker
- Timo Justinger
- Valdrin Dajaku
- Felix Mältzer
- Till Rädler

KIT ETP

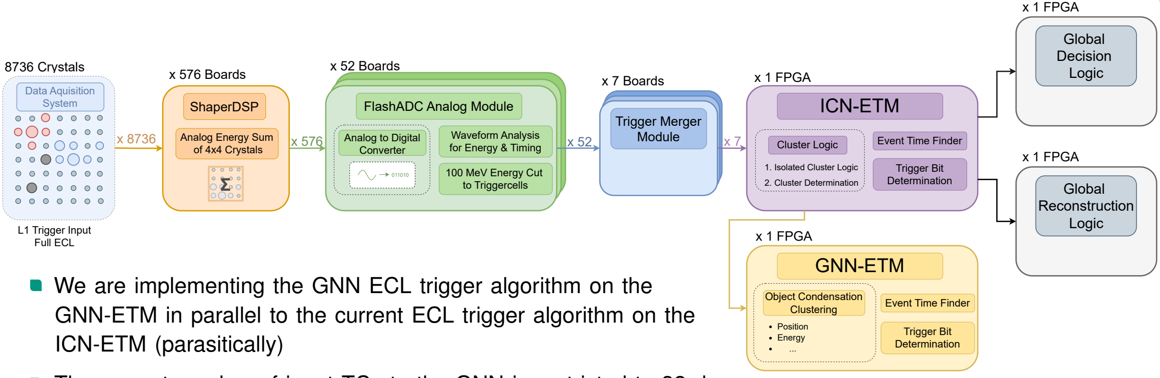
- Lea Reuter
- Greta Heine
- Isabel Haide
- Torben Ferber

MPI

- Timo Forsthofer
- Simon Hiesl
- Christian Kiesling

Figure: Greta Heine

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

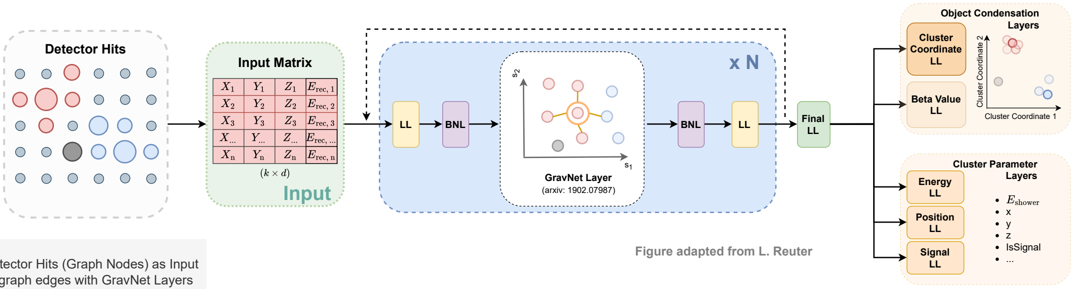
Current Situation:

- 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



Use Detector Hits (Graph Nodes) as Input
Learn graph edges with GravNet Layers

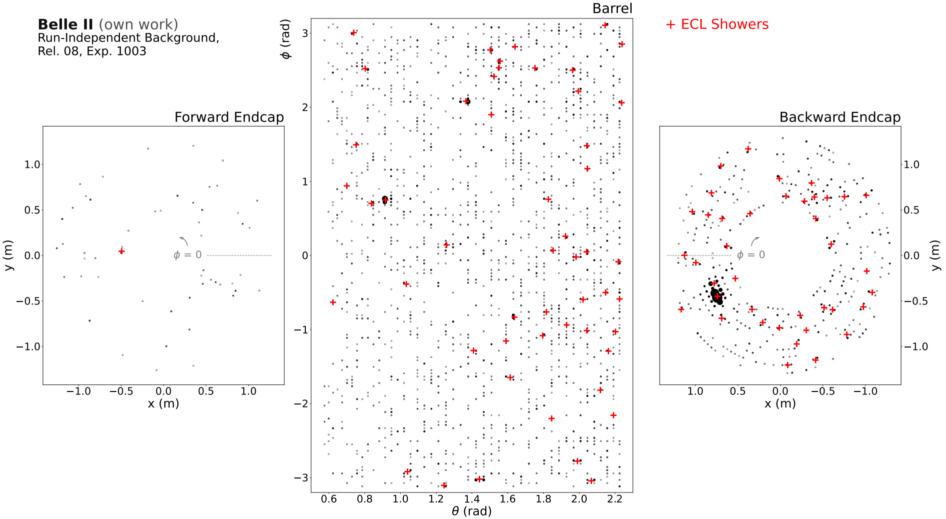
Figure adapted from L. Reuter

- 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

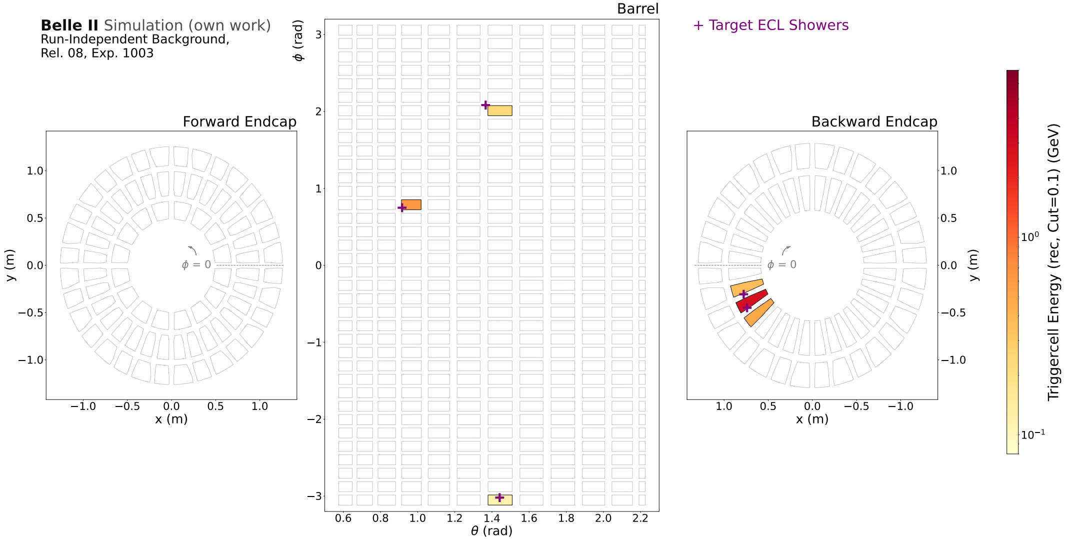
Training Samples and Training Target

- 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



Optimization for Hardware Implementation

Karlsruhe Institute of Technology

- 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

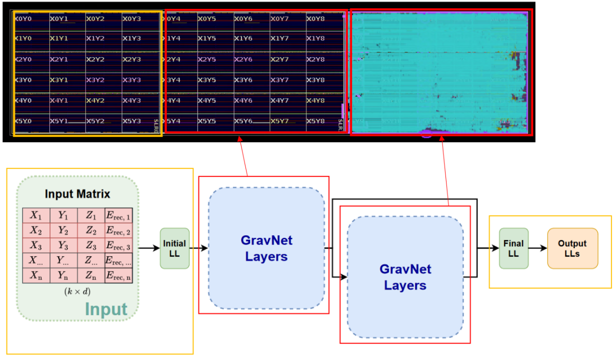


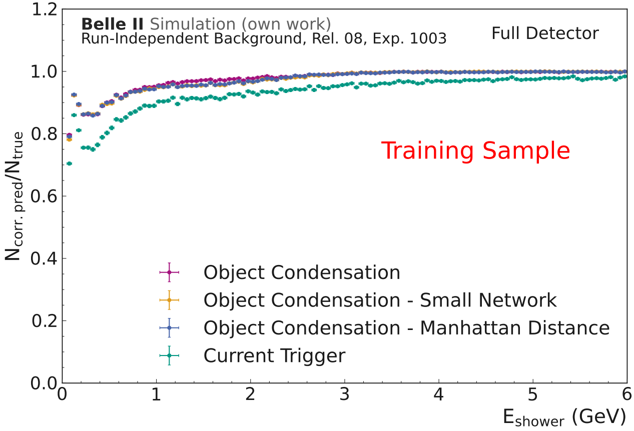
Figure: Marc Neu

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

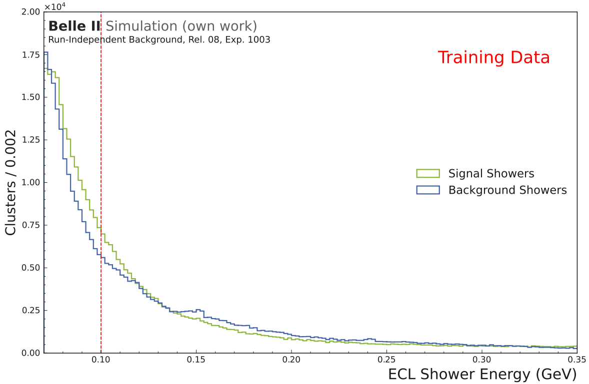
- Original network has been designed by broad hyperparameter optimisation (number of GravNet blocks was set to 2)
- The evaluation shown here is still on CPU and with non-quantized weights and inputs
- The network with reduced amount of linear layers (=small network) only has around 8000 tunable parameters, 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
- Implementation still has 200% DSP usage with a quantization of 16bit, higher quantization or further reduction is necessary



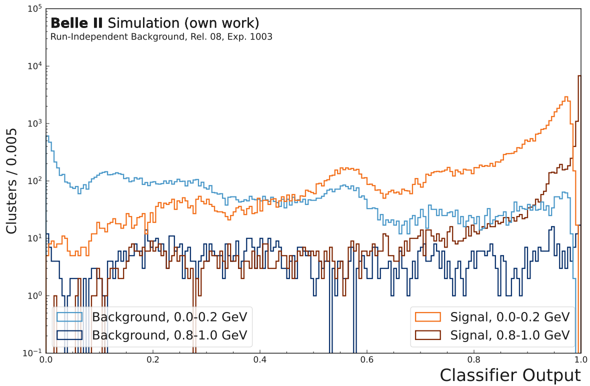
- Efficiency: N(corr. pred.) / N(true)
  - True are all ECL showers visible on TRG level, correctly predicted are all GNN/TRG clusters truthmatched to a shower

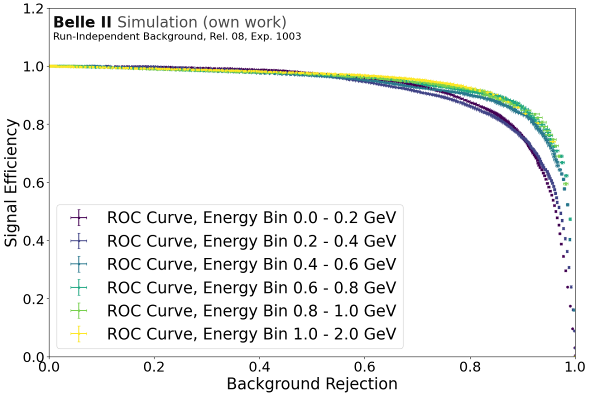
Signal/Background Showers

- 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

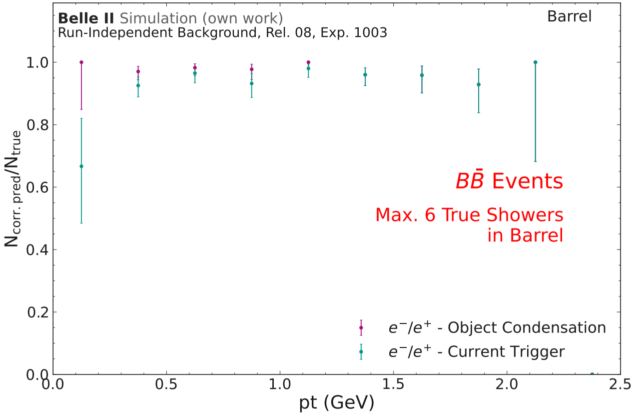




- Network can classify signal/background clusters on training data with high efficiency and high beam background rejection
- Signal definition has to be carefully defined, especially for charged particles (work in progress)

Performance on Clusters from Charged Particles

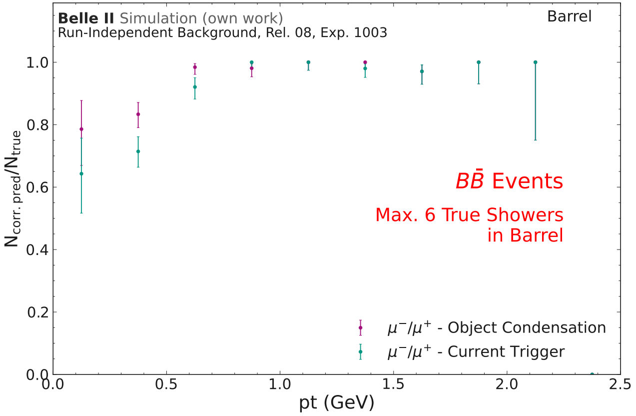
- Efficiency on charged particles might differ due to different signatures
- Preliminary tests on 10k mixed $B\bar{B}$ events
- Due to the restriction of max. 6 TRGECLClusters in the order Barrel $\rightarrow$ Forward Endcap $\rightarrow$ Backward Endcap, for both algorithms I compare only events that have $\leq$ 6 Target ECL Showers in the barrel
- If a Target ECL Shower has $\geq$ 30% energy deposition by an MC particle, I match it to that MC particle
- Efficiency for OC is very high over all pt bins
- Efficiency for OC is also consistently high when including all detector parts and all Target ECL Showers



- 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

Performance on Clusters from Charged Particles

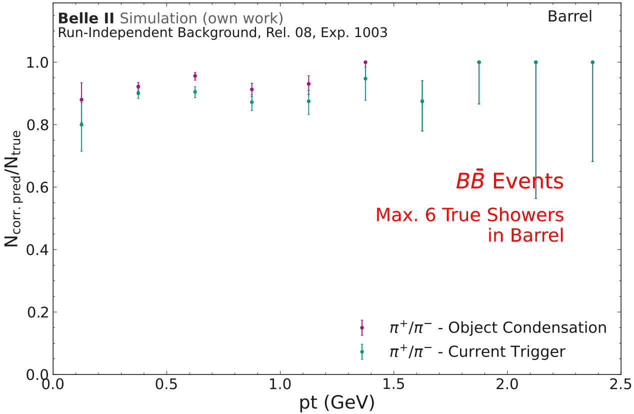
- Efficiency on charged particles might differ due to different signatures
- Preliminary tests on 10k mixed $B\bar{B}$ events
- Due to the restriction of max. 6 TRGECLClusters in the order Barrel $\rightarrow$ Forward Endcap $\rightarrow$ Backward Endcap, for both algorithms I compare only events that have $\leq$ 6 Target ECL Showers in the barrel
- If a Target ECL Shower has $\geq$ 30% energy deposition by an MC particle, I match it to that MC particle
- Efficiency for OC is very high over all pt bins
- Efficiency for OC is also consistently high when including all detector parts and all Target ECL Showers



- 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

Performance on Clusters from Charged Particles

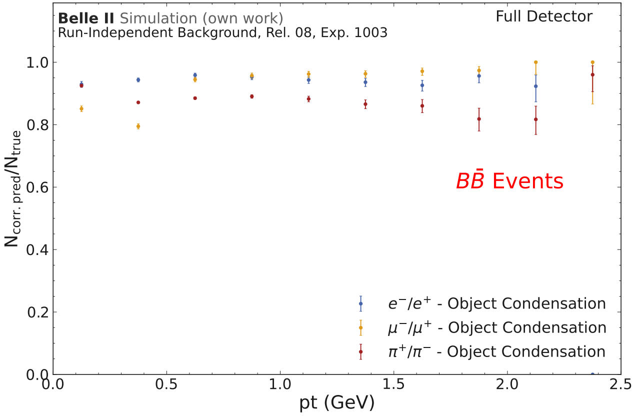
- Efficiency on charged particles might differ due to different signatures
- Preliminary tests on 10k mixed $B\bar{B}$ events
- Due to the restriction of max. 6 TRGECLClusters in the order Barrel $\rightarrow$ Forward Endcap $\rightarrow$ Backward Endcap, for both algorithms I compare only events that have $\leq$ 6 Target ECL Showers in the barrel
- If a Target ECL Shower has $\geq$ 30% energy deposition by an MC particle, I match it to that MC particle
- Efficiency for OC is very high over all pt bins
- Efficiency for OC is also consistently high when including all detector parts and all Target ECL Showers



- 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

Performance on Clusters from Charged Particles

- Efficiency on charged particles might differ due to different signatures
- Preliminary tests on 10k mixed $B\bar{B}$ events
- Due to the restriction of max. 6 TRGECLClusters in the order Barrel $\rightarrow$ Forward Endcap $\rightarrow$ Backward Endcap, for both algorithms I compare only events that have $\leq$ 6 Target ECL Showers in the barrel
- If a Target ECL Shower has $\geq$ 30% energy deposition by an MC particle, I match it to that MC particle
- Efficiency for OC is very high over all pt bins
- Efficiency for OC is also consistently high when including all detector parts and all Target ECL Showers



- 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

Current Status of Implementation

- Implementation is done by Marc with great support from the TRG group, especially Unno-san
- Due to their support, we already have an interface between ICN-ETM and GNN-ETM as well as integration of the GNN-ETM channel in Belle2Link DAQ
- Currently debugging the interface between the GNN-ETM and Belle2Link in local run configuration
- The GNN Implementation is currently under development, we expect a first full implementation on hardware in approx. 2 weeks, first quantized training will be ready for implementation next week
- Preprocessing is implemented and tested, ready to be integrated into the GNN-ETM firmware in the next 2 weeks

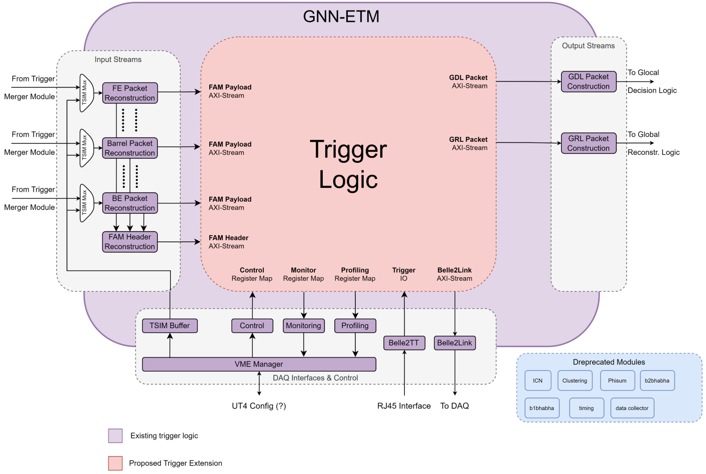
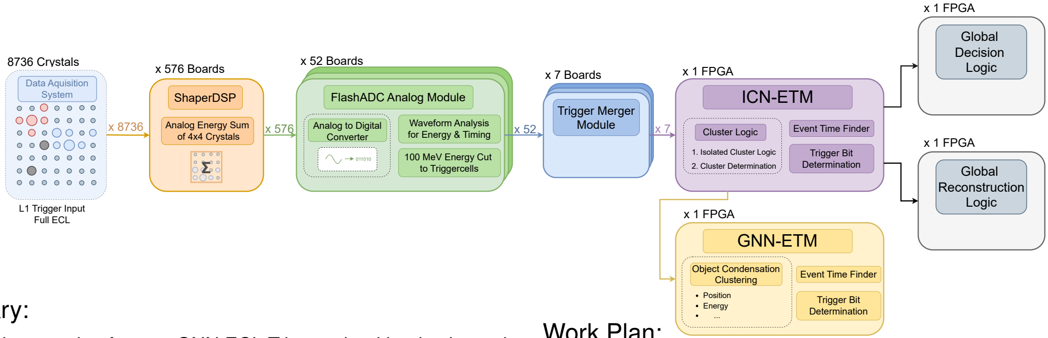


Figure: Marc Neu

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

Work Plan:

- Network weights have to be quantized to 16bit, partially 8bit quantization, finishing until Monday
- First full implemented model will be ready in approx. 2 weeks
- Next step is then evaluating on the new raw data and comparing performance on physics data

Backup

Full BB Efficiency for Different Particles