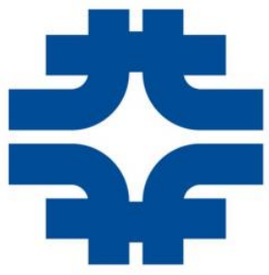
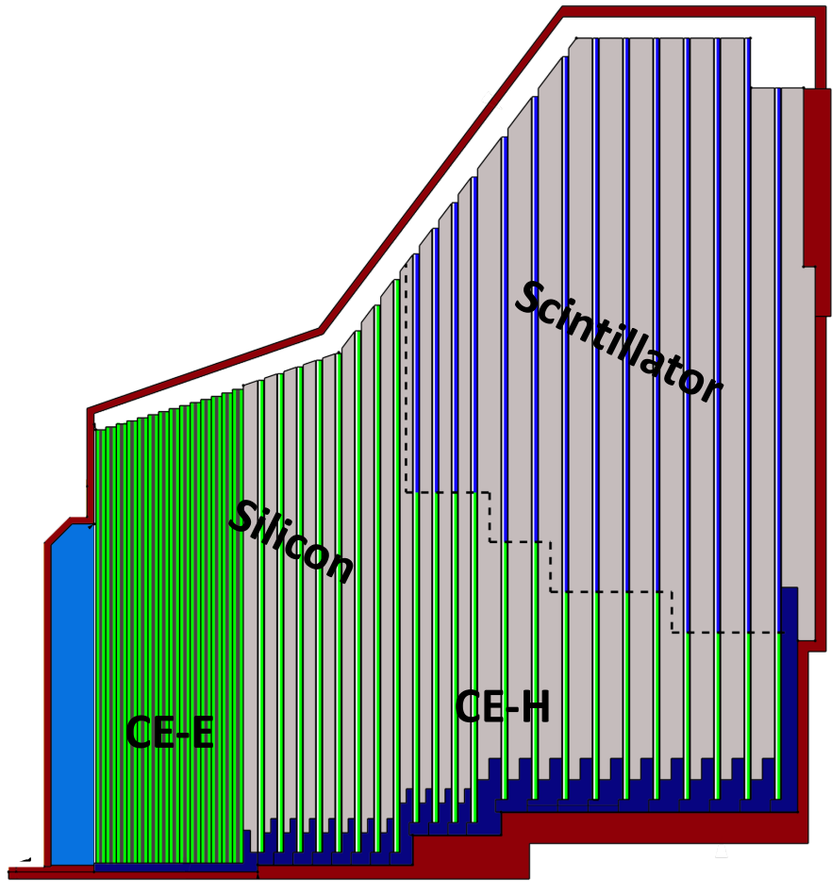
Design of a reconfigurable autoencoder neural network for detector front-end ASICs

INFIERI 2021 - August 31, 2021

Columbia University : Giuseppe Di Guglielmo, Luca Carloni
Fermilab : Farah Fahim, Cristian Gingu, Christian Herwig, Jim Hirschauer,
Martin Kwok, Nhan Tran
Florida Tech : Danny Noonan
Northwestern University : Manuel Valentin, Yingyi Luo, Seda Memik



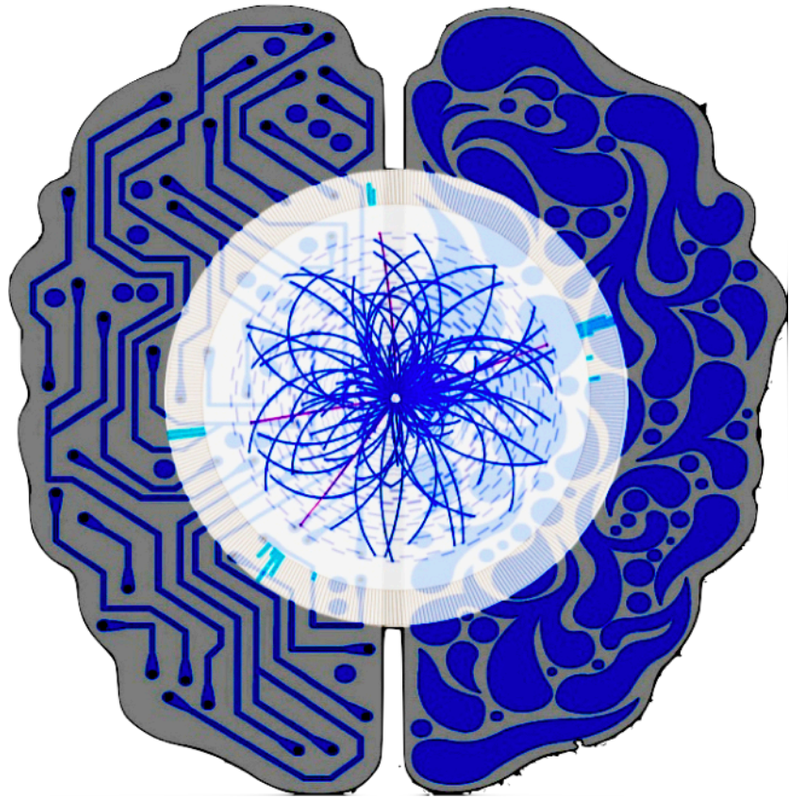
With thanks to the CMS Collaboration,
and in particular,
the CMS High-Granularity Calorimeter Group



Thanks also to

FAST MACHINE LEARNING LAB

https://fastmachinelearning.org/



2020 Fast ML for Science workshop:
https://indico.cern.ch/event/924283/

Please join the next workshop :
tentatively end-of-2021 / early-2022

HEP data challenge

HEP aims to discover increasingly **more massive particles**, probe **smaller distances**, and study **more rare processes**.

This requires a series of colliders with continually increasing **energy** and **luminosity**

➔ increasing **detector occupancy**

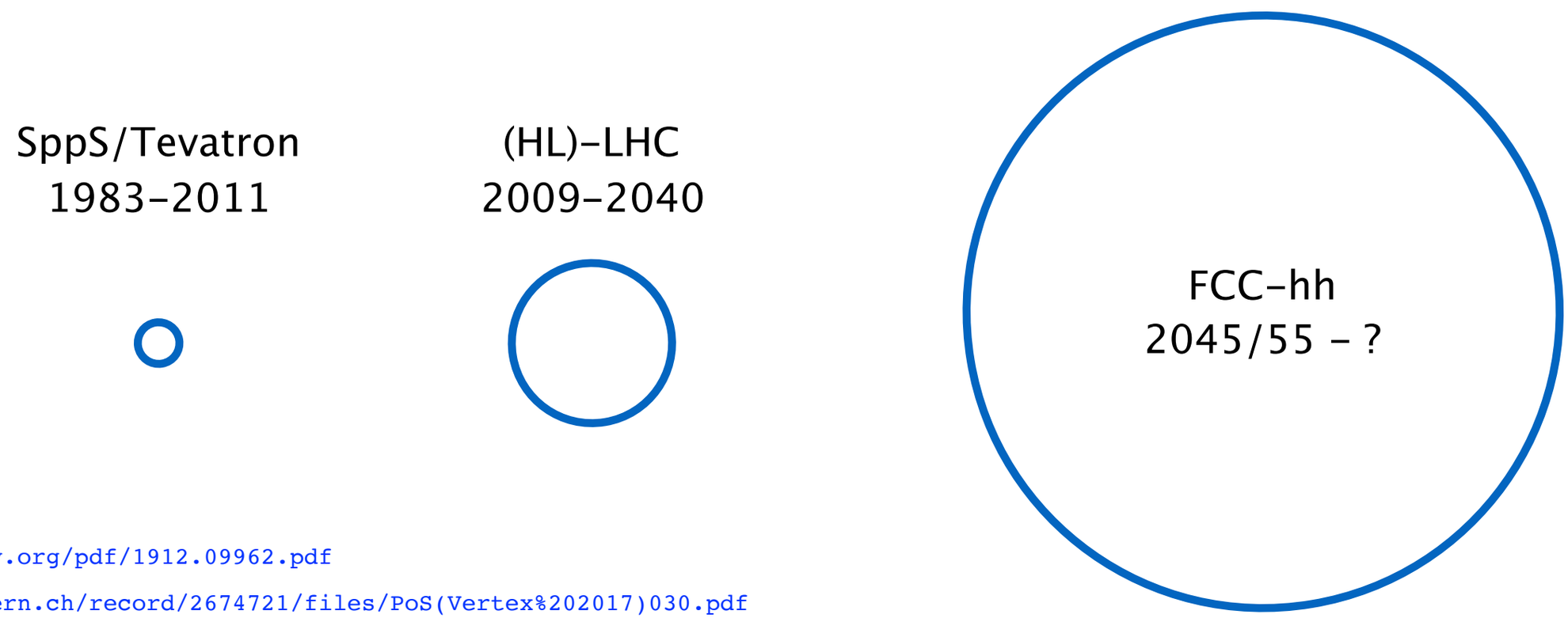
➔ increasing **detector granularity and precision**

➔ increasing **data volume** produced by detector

"The solution to every problem is another problem."
Johann Wolfgang von Goethe

HEP data challenge

Collider	Tevatron	LHC	HL-LHC	FCC-hh
Luminosity [$cm^{-2}\ s^{-1}$]	3.7×10^{33}	21×10^{33}	50×10^{33} with leveling	300×10^{33}
Pileup	1-2	50	200	1000
Typical number of tracker channels	<1M	>100M	>1B	17B **
Typical number of calorimeter channels	<100k	>100k	6M	100M ***
Inner detector TID	10 Mrad	100 Mrad	500 Mrad	30 Grad *



***https://arxiv.org/pdf/1912.09962.pdf

** http://cds.cern.ch/record/2674721/files/PoS(Vertex%202017)030.pdf

* https://cds.cern.ch/record/2651300/files/CERN-ACC-2018-0058.pdf

Data challenge solutions ➔ new problems

Increasing detector **data volume**

➔ **move more data processing to on-detector electronics**

➔ increasing **complexity, power consumption,** and **radiation tolerance**

What data processing should move on-detector?

- **data compression**
- reconstruction of low-level objects (hits, clusters)
- reconstruction of high-level objects (tracks, jets)

On-detector data compression

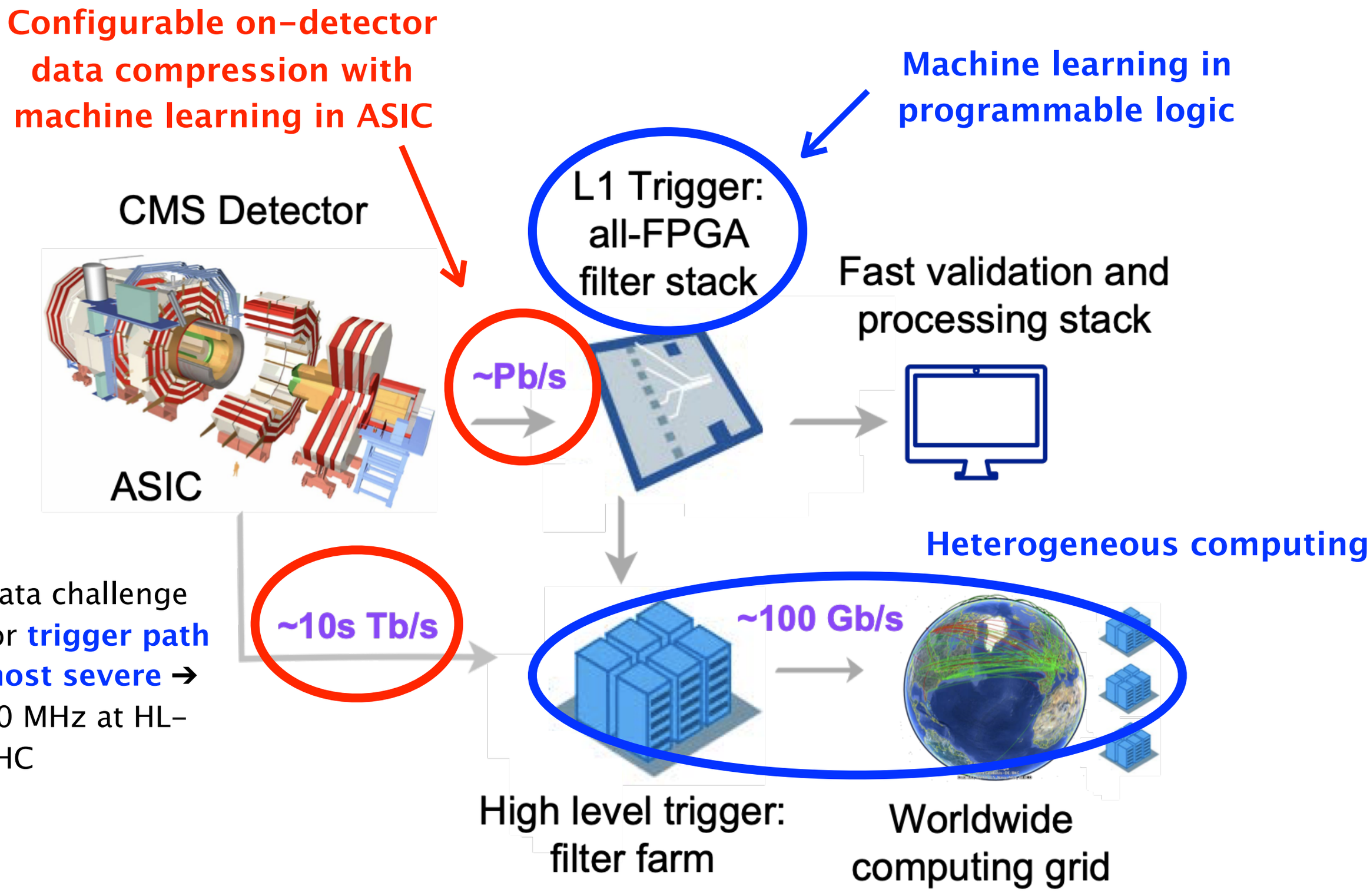
- This talk: **Neural Network (NN) autoencoder** in **ASIC** for **on-detector data compression**.

- General requirements for on-detector electronics:
  - **Low power consumption → well suited to ASIC**
  - **Radiation tolerant → well suited to ASIC**
  - **Complexity:** design must be **re-configurable → challenging for ASIC**

- Specific requirements for the CMS High-Granularity Calorimeter (HGCAL).

Illustration: Lisa Hornung/iStockPhoto

Context within HL-LHC Data Challenge

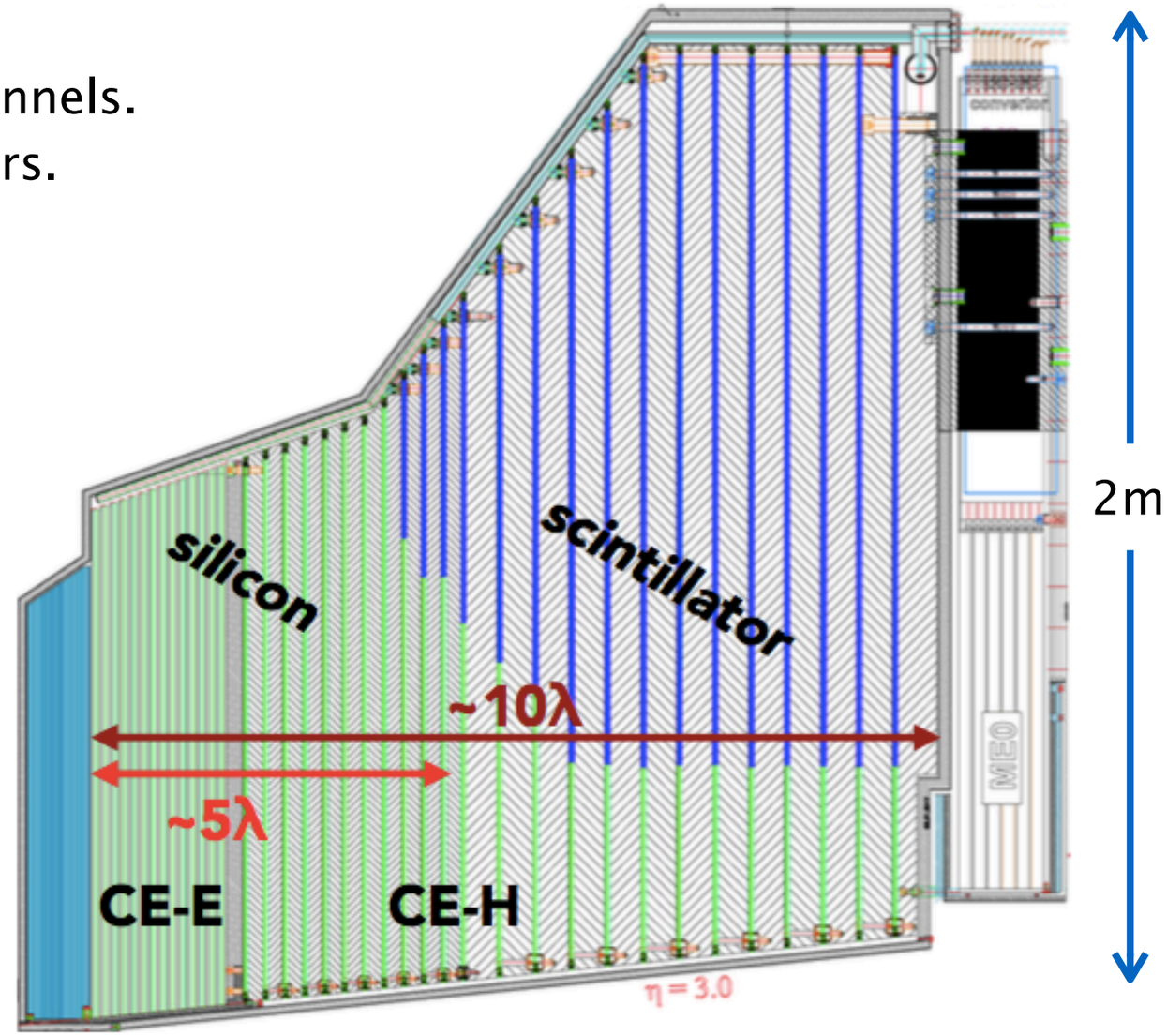


- Data challenge for **trigger path most severe** ➔ 40 MHz at HL-LHC

CMS High Granularity Calorimeter (HGCAL)

- "**Imaging calorimeter**" with ~6M readout channels.
  - 60× increase from current LHC calorimeters.

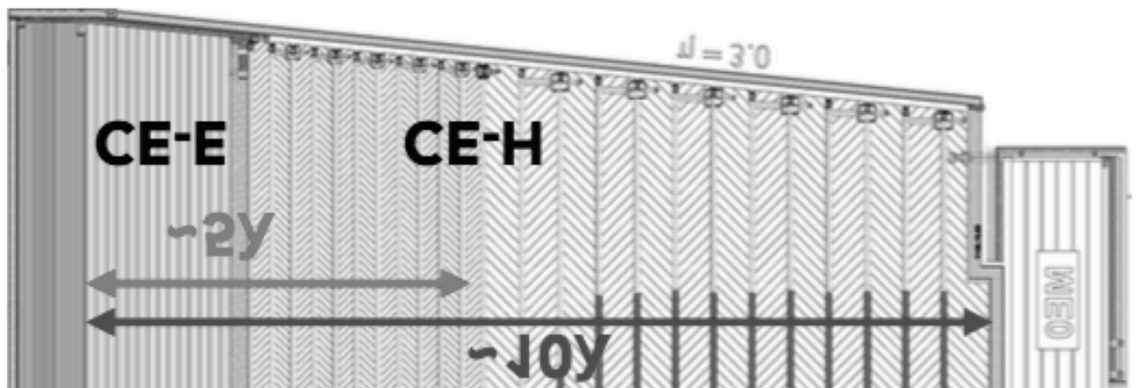
- ~50 layers of active material + absorber.
  - silicon sensors in front layers
  - scintillator + silicon in back layers



proton beam

proton beam

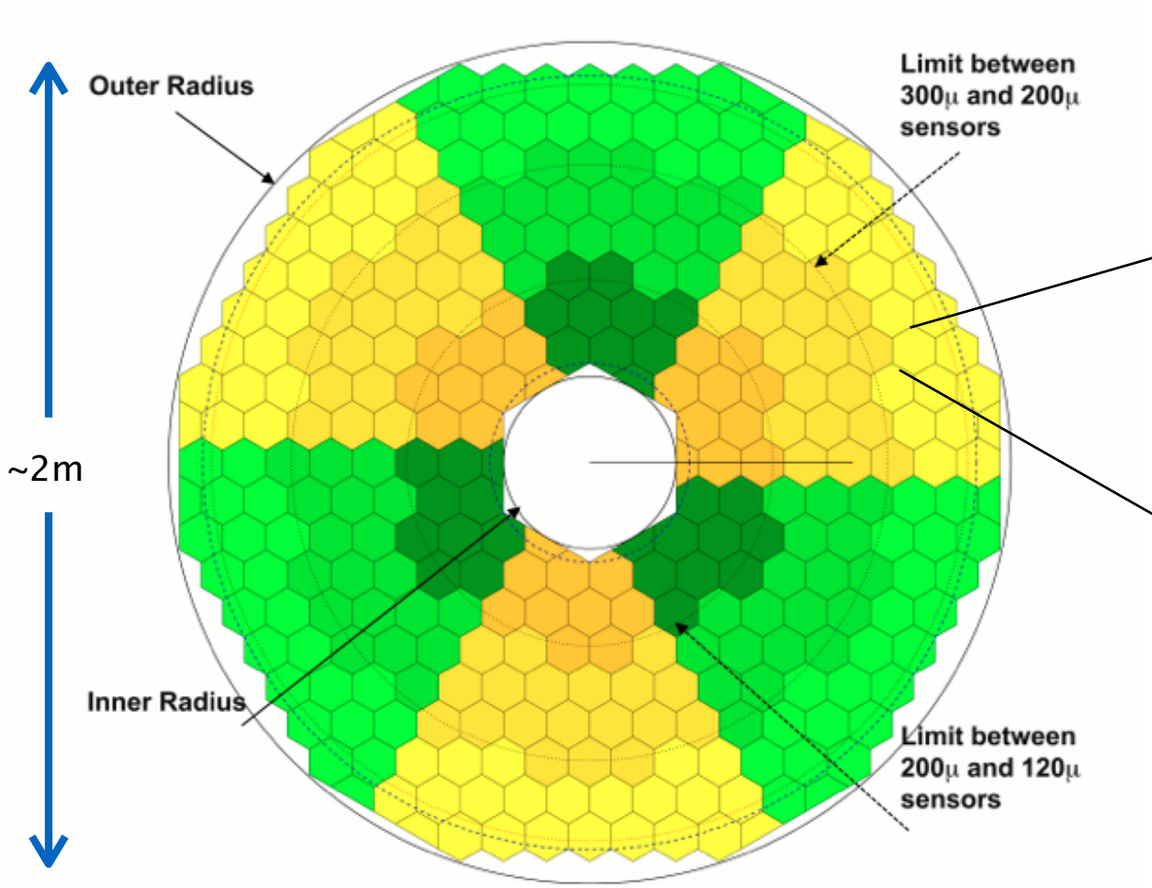
collision point

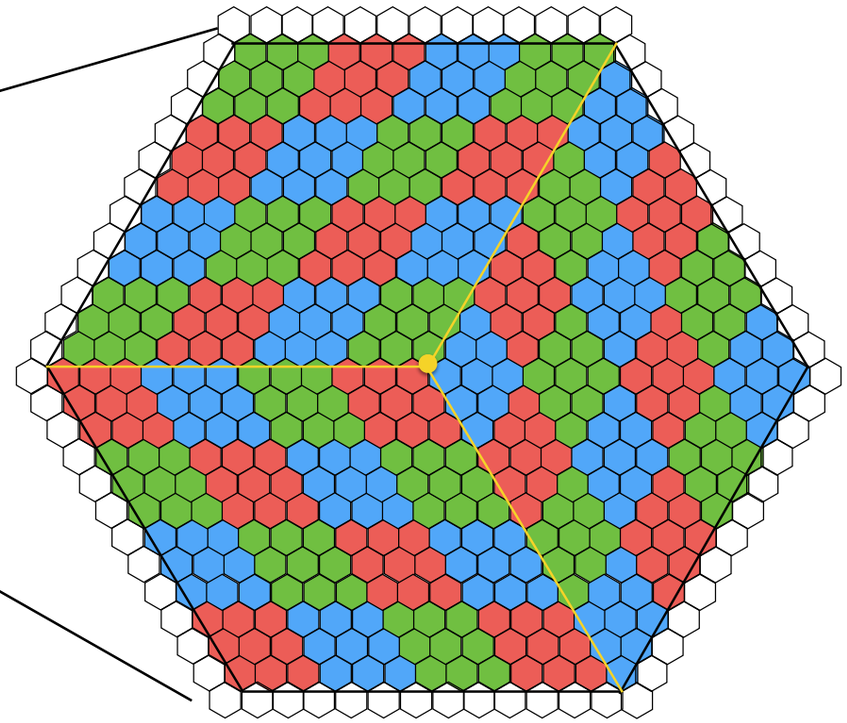


CMS High Granularity Calorimeter (HGCAL)

Each layer tiled with 300-500 8" hexagonal silicon modules.

Each 8" module includes either 192 or 432 ~1 cm sensor channels



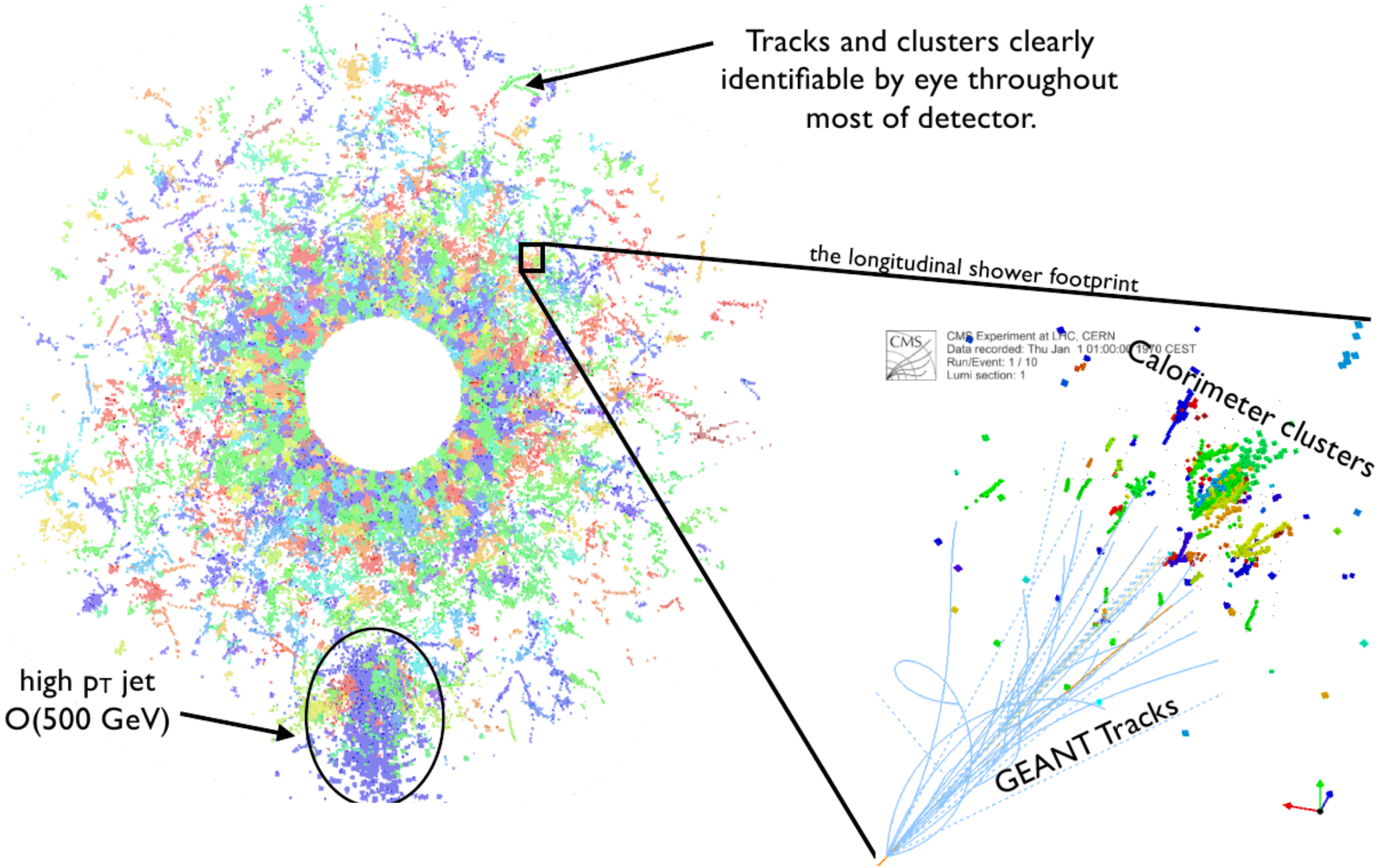


Front End electronics: each 8" module includes

- **≤6 HGCROC ASIC** : digitizes charge and arrival time and provides charge data for trigger path.
- **1 ECON-T ASIC** : selects/compresses digital trigger data for transmission off-detector.
  - **On-detector data compression with machine learning in ECON-T** .

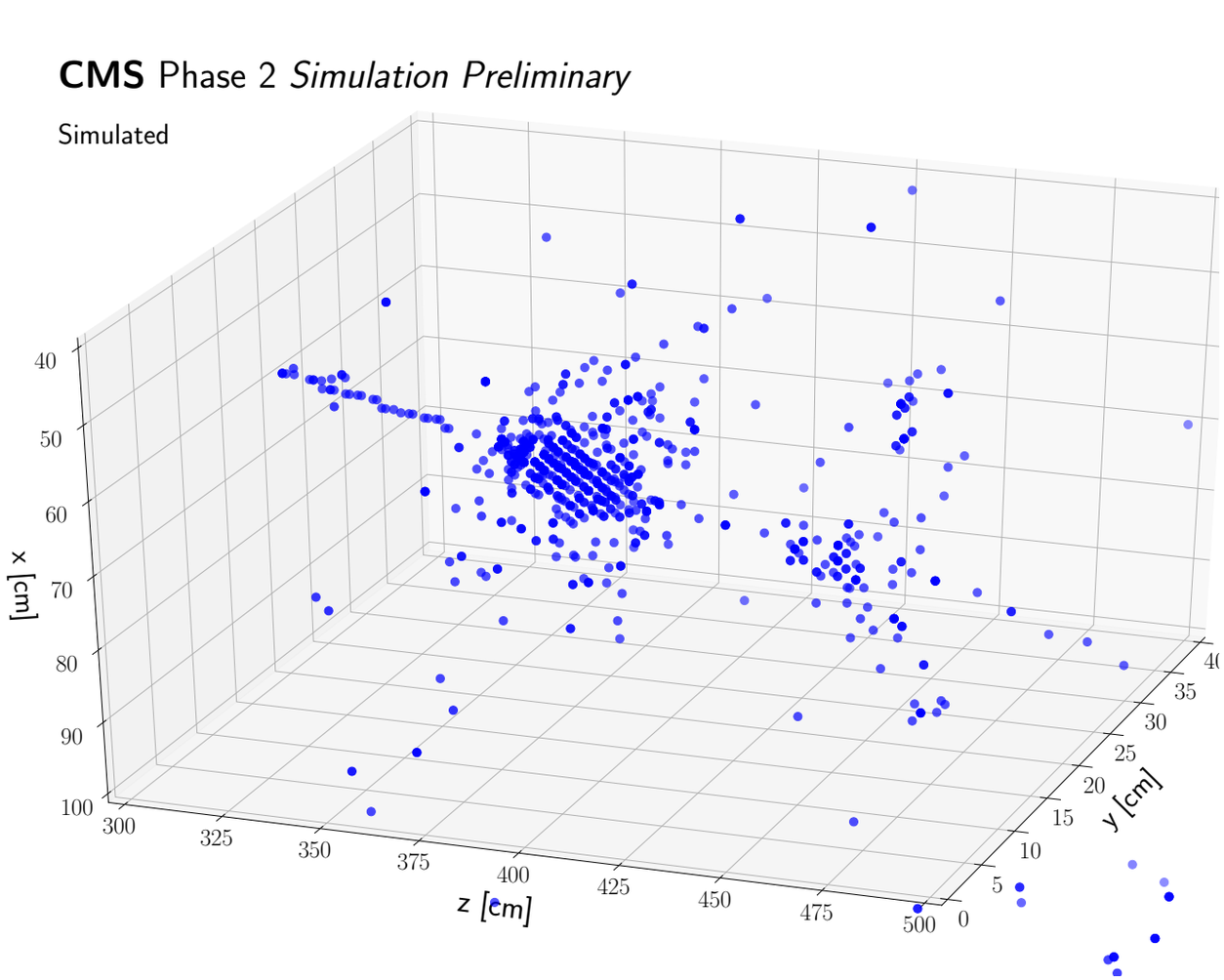
Imaging calorimeter

500 GeV jet in 140 pileup

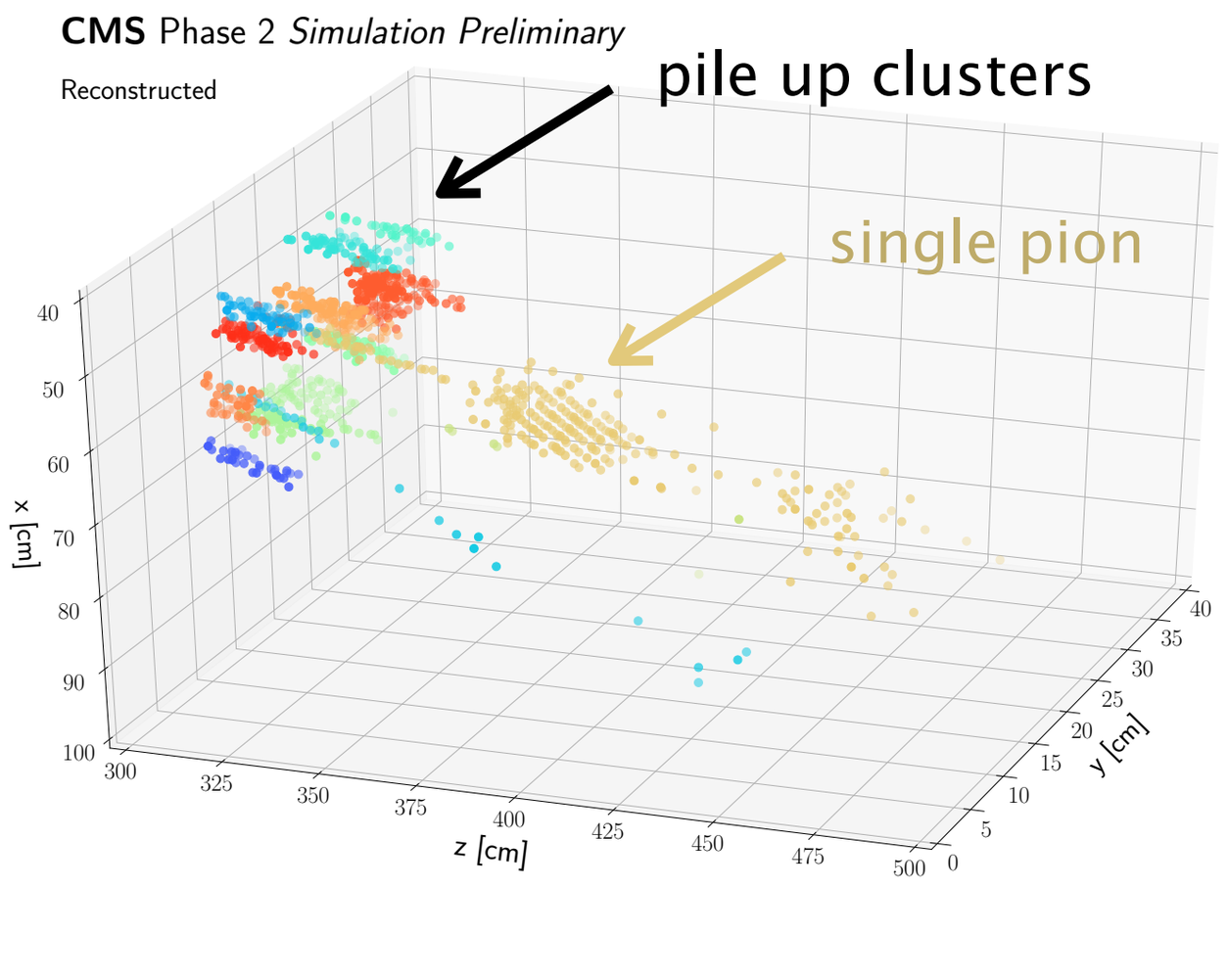


Lindsey Gray (FNAL)
https://cds.cern.ch/record/2239183/files/CR2016_440.pdf

Imaging calorimeter



Simulated hits for single ~50 GeV pion interacting with HGCAL



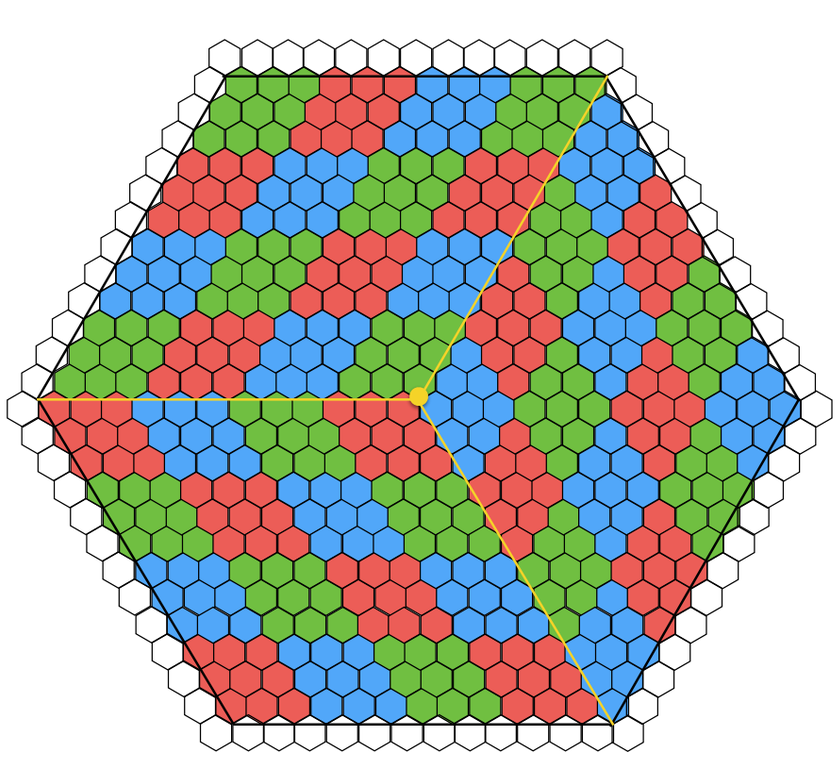
Reconstruction of clusters with 200 PU overlaid on single pion

Jingyu Zhang
ICHEP 2020

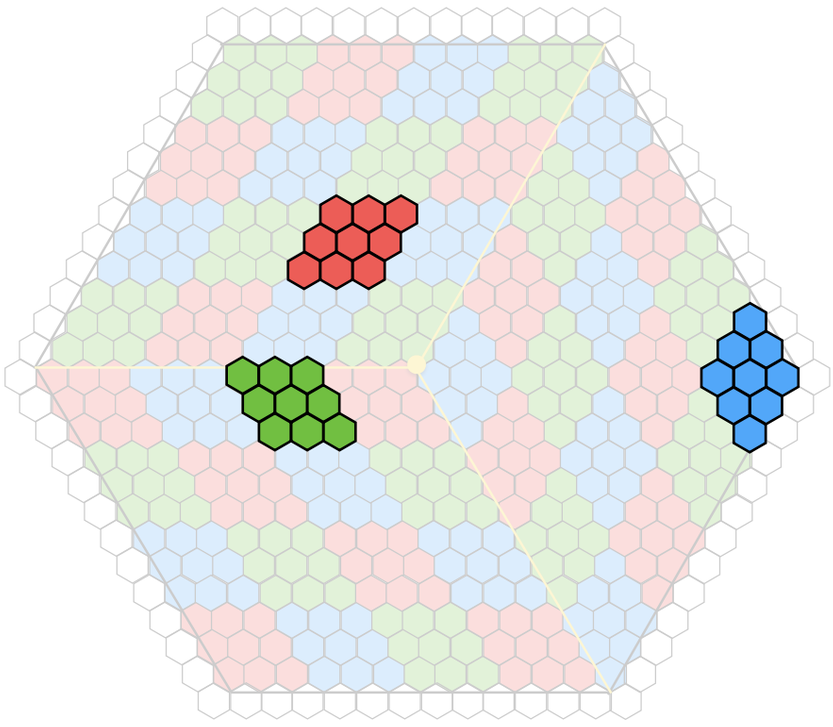
HGCAL trigger data challenge

Trigger path stage	Number channels	bits/ channel	Average Compression factor	Data rate*	# links* (10.24 Gbps)
Raw data	6M	20	1	5 Pb/s	1M
Hardware reduction	1M	7	1	300 Tb/s	60k
Threshold selection	1M	7	7	40 Tb/s	9k

* Assumes 40 MHz rate and 50% link packing efficiency



432 silicon sensors ➔ 48 trigger cells (TC) @ 7b per TC



Traditional threshold algorithm : 3 of 48 TC readout for most of detector (2 × 1.28G elink per module)

Specific challenges and requirements for the on-detector ASIC

Occupancy and pileup:

- Varies by 2-3 orders of magnitude over pseudorapidity/depth and in time.
- **Compression neural network must be configurable** to handle different detector locations and changing detector/beam conditions

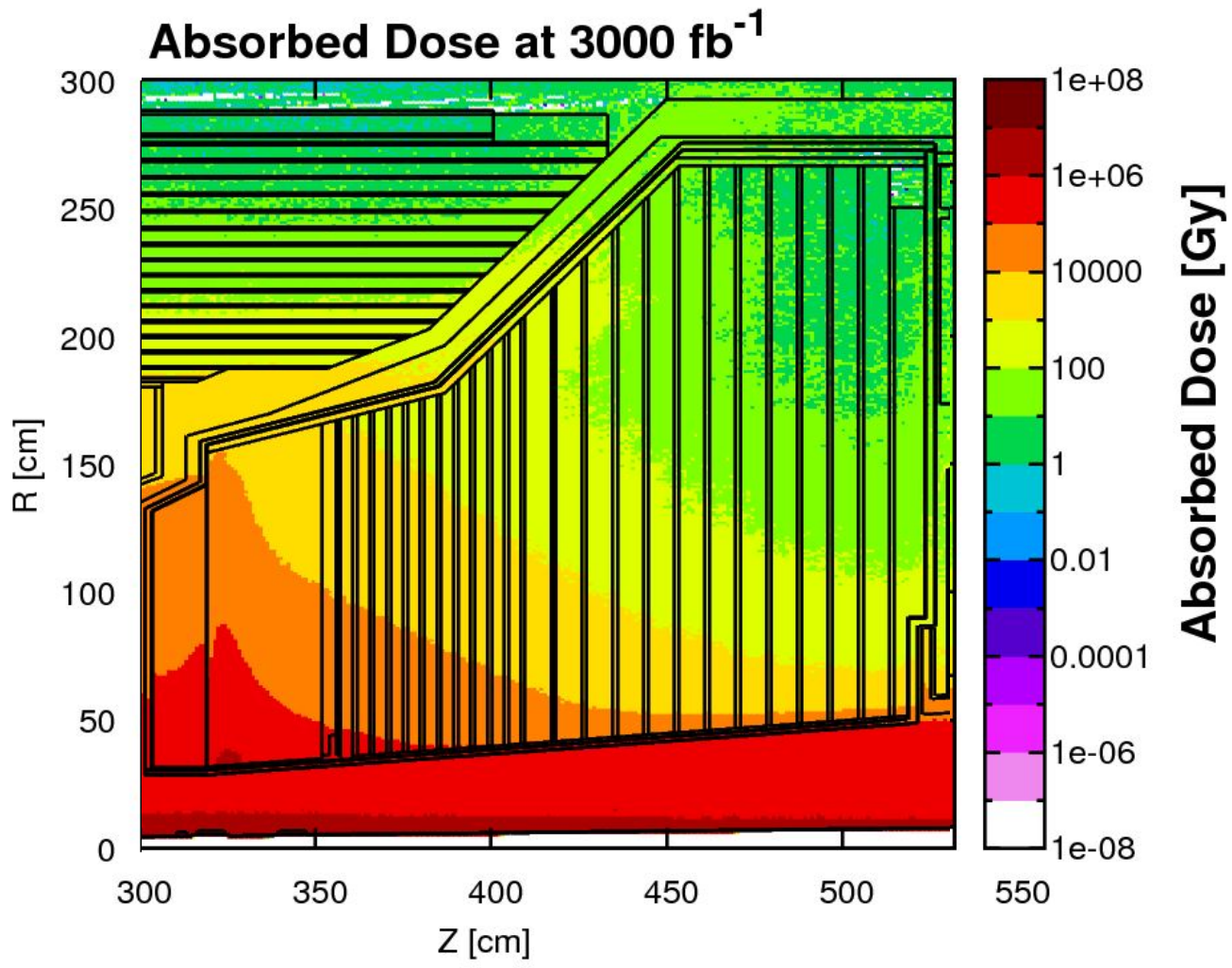
Latency:

- ~On trigger path latency is precious ➔ must be **< 100 ns**

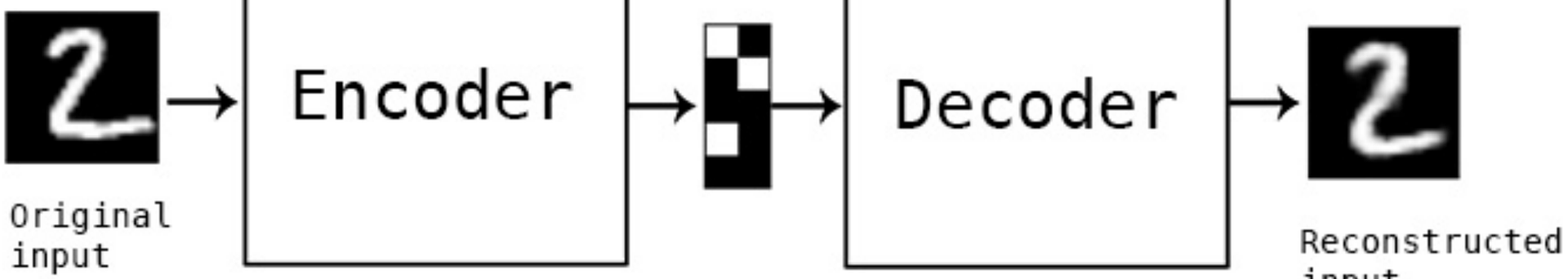
Power :

- ~30k encoder networks on the entire detector.
- Power budget is **100 mW** network, or around 1 nJ per inference.

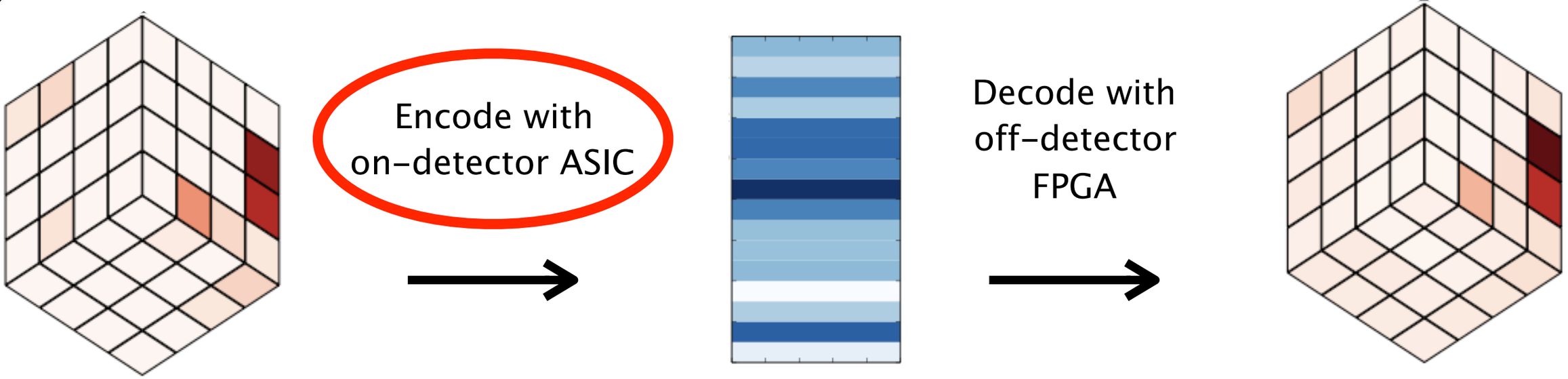
Radiation tolerance : up to **500 Mrad**



Autoencoder concept for data compression



https://blog.keras.io/building-autoencoders-in-keras.html

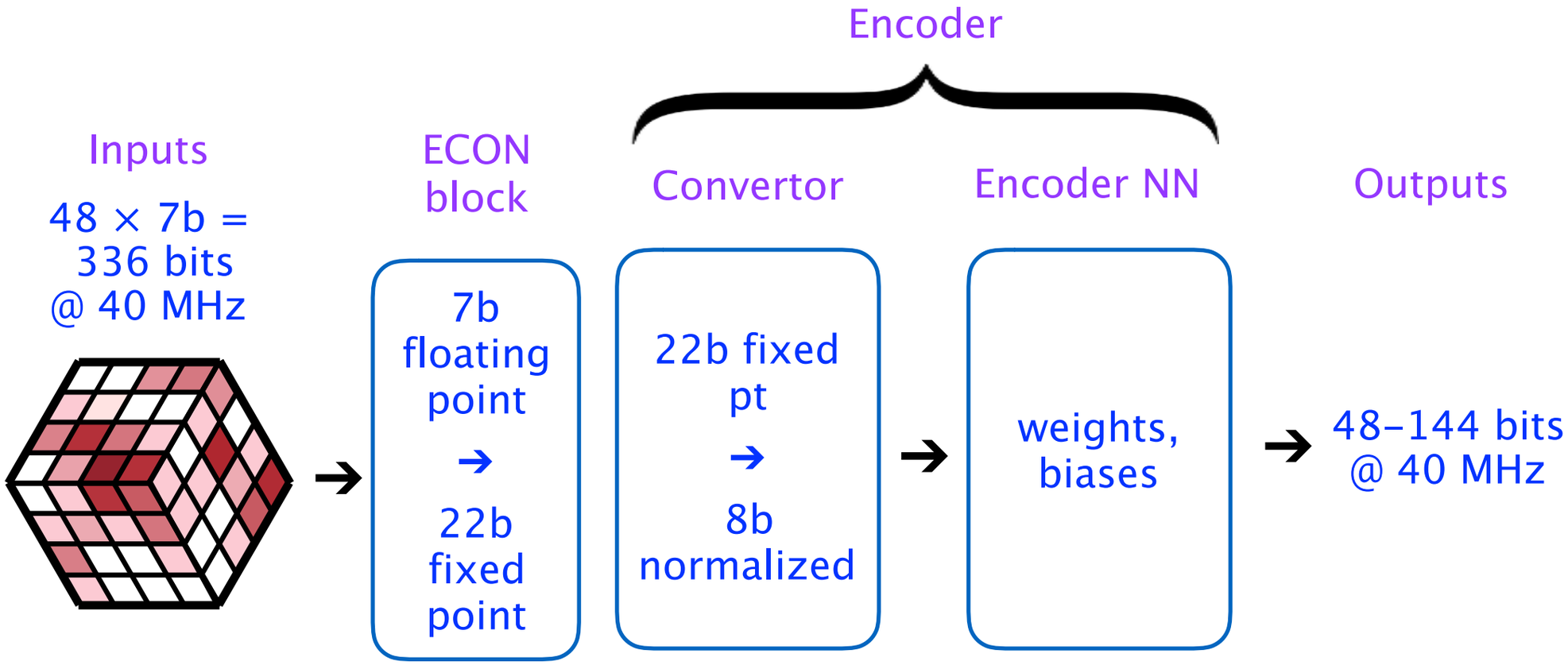


Encoder NN design considerations

- **Minimize** : **power** + **area** + **latency**
- **Maximize** : **physics performance** + **configurability** + **radiation tolerance**

- **Network architecture** and precision of weights and biases: fixed in design

- **Fully re-configurable** : **all network weights and biases** + dimensionality of output



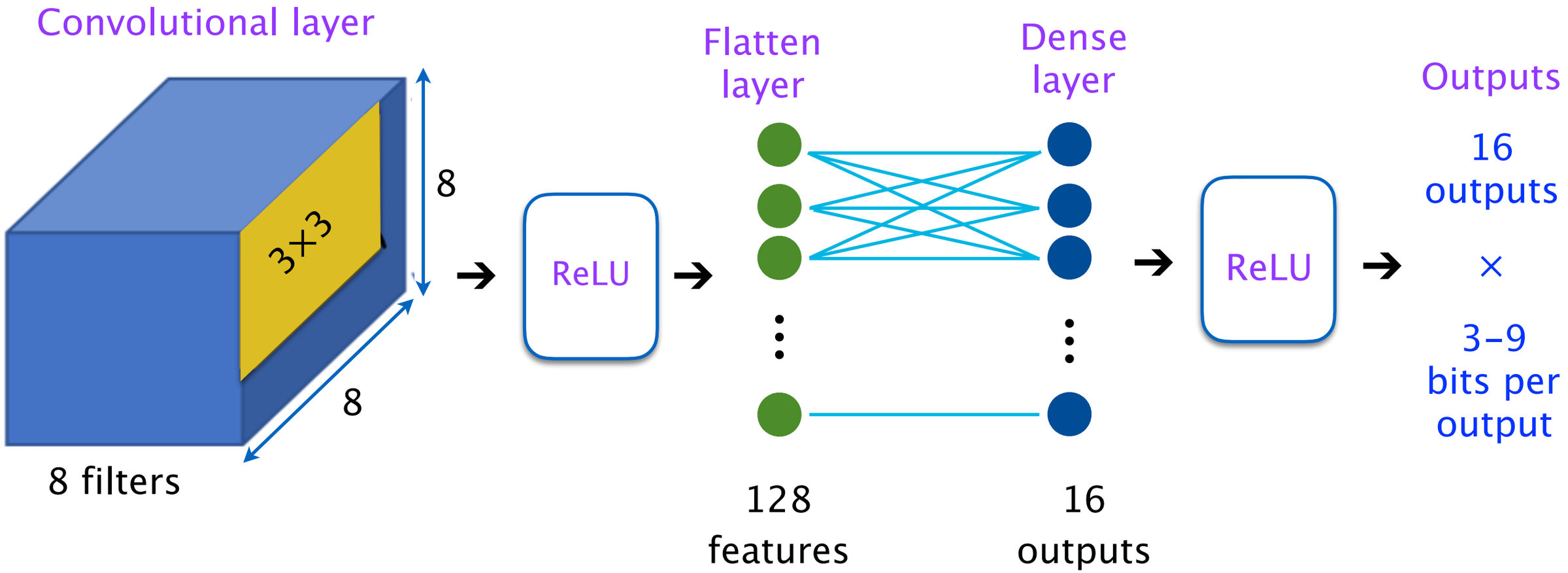
Encoder NN design considerations

Encoder NN components

- **Convolutional layer** (conv2D): extract geometric features
- **Flatten layer** : vectorizes 2D image from conv2D ($128 = 8 \times 4 \times 4$)
- **Dense layer :** decide which geometric features are important
- **ReLU :** activation function

Encoder NN

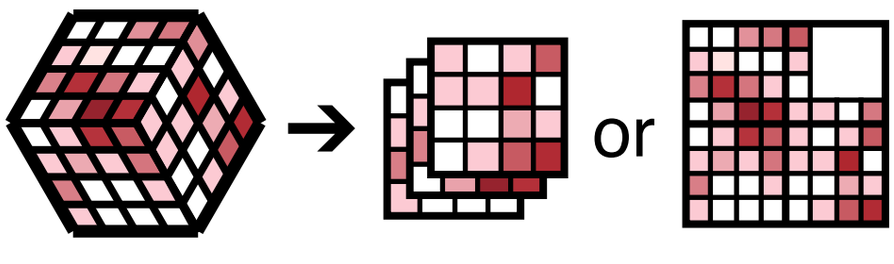
Optimization of dimensions shown next



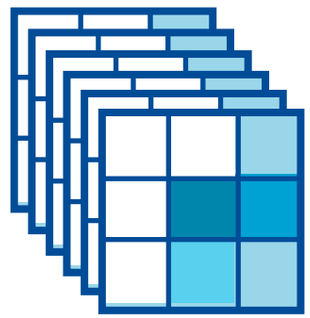
Encoder NN architecture optimization

- Optimize encoder network architecture choices including :

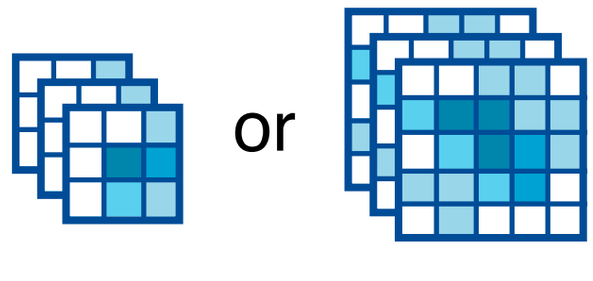
Geometry mapping



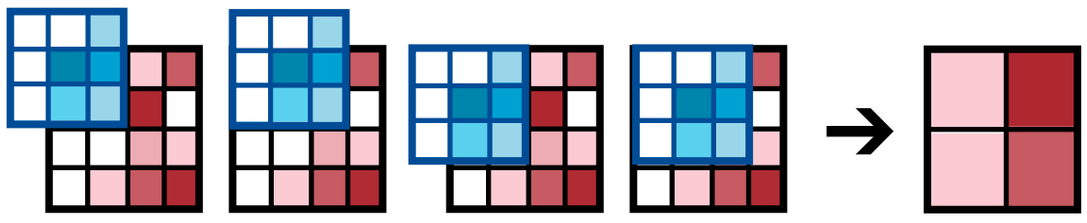
of conv2D filters



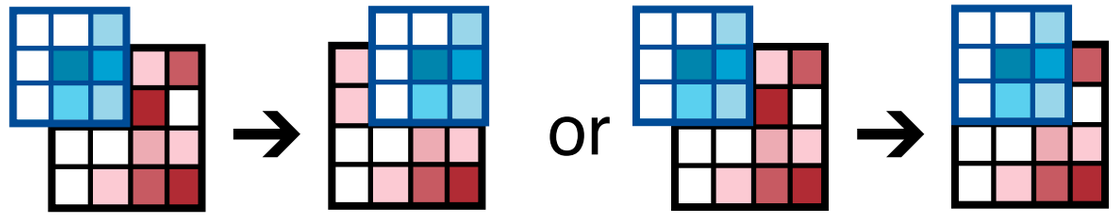
conv2D kernel size



Max pooling conv2D outputs



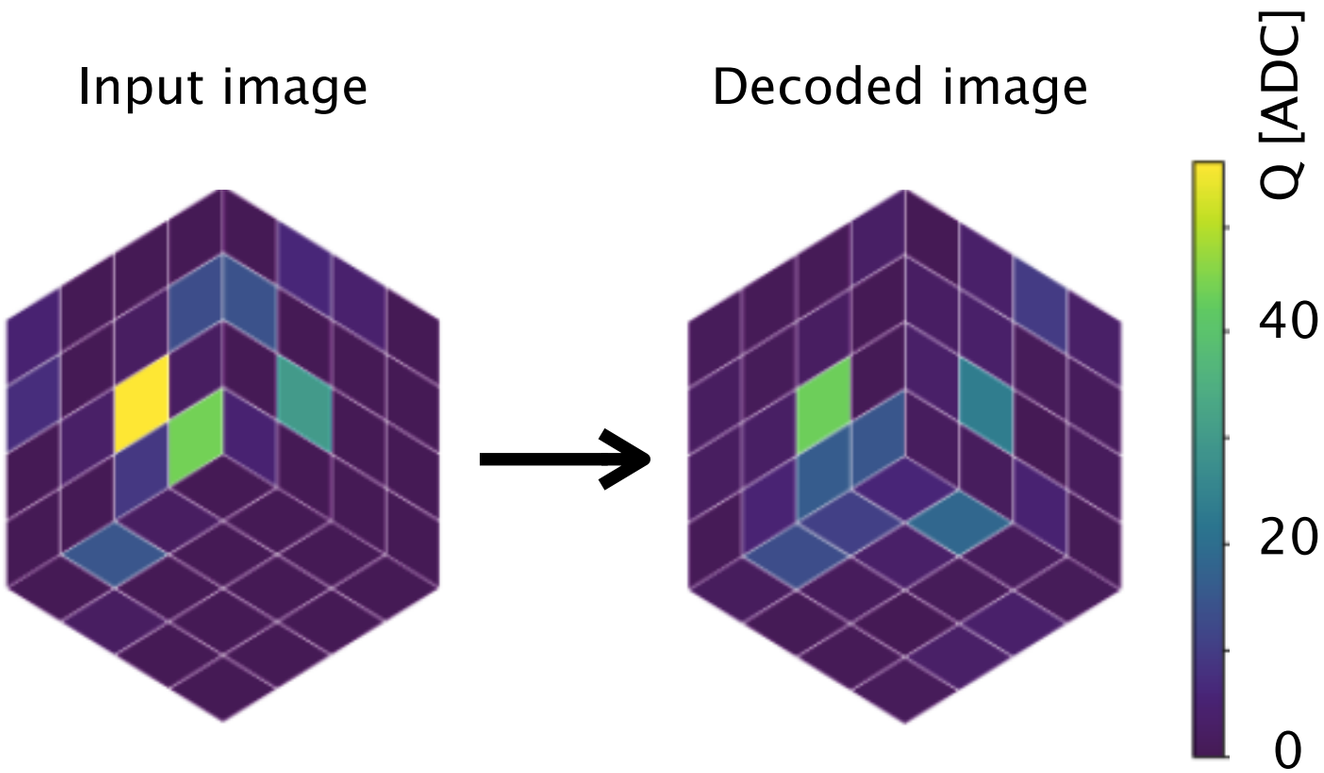
conv2D kernel stride



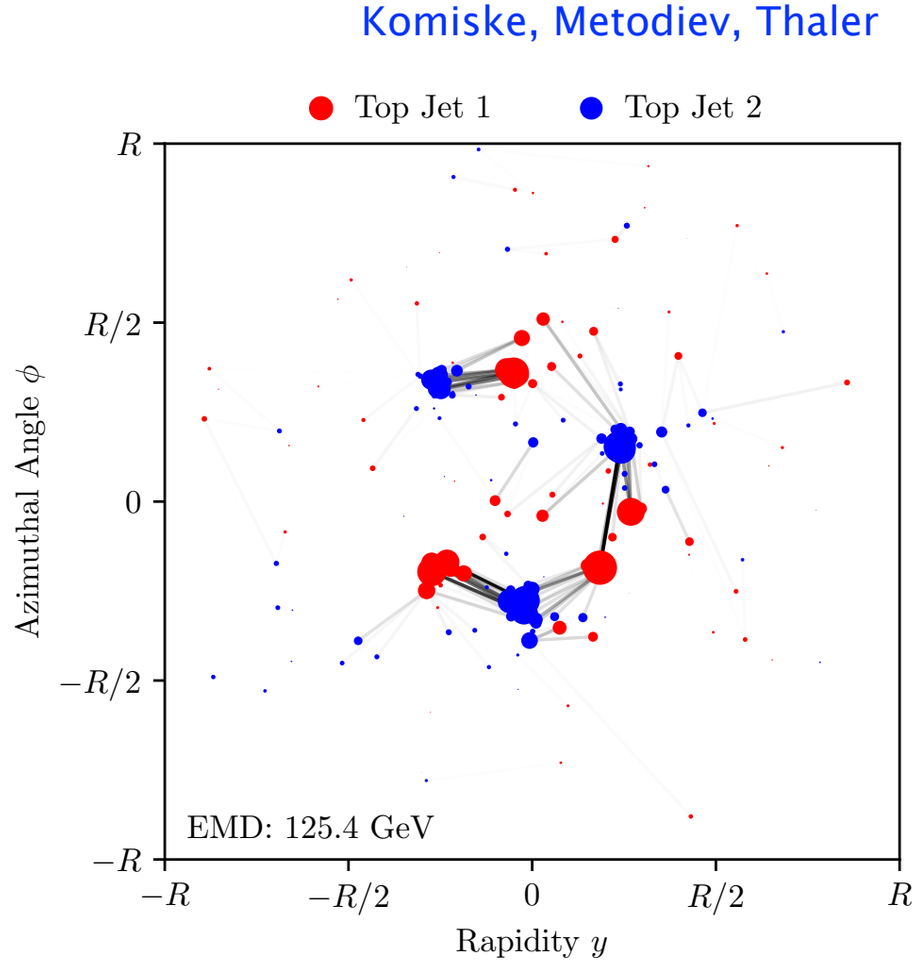
Performance metric : EMD

- For rapid prototyping evaluate network performance according to **image similarity**.
- **Energy Mover's Distance** :
  - the "work" required to rearrange one radiation pattern into another
  - first associated with "optimal transport" problem
- For each NN variation : train network and evaluate EMD with simulated physics events including top quarks (jets, leptons) and **200 pileup**.

arXiv:1902.02346
Komiske, Metodiev, Thaler



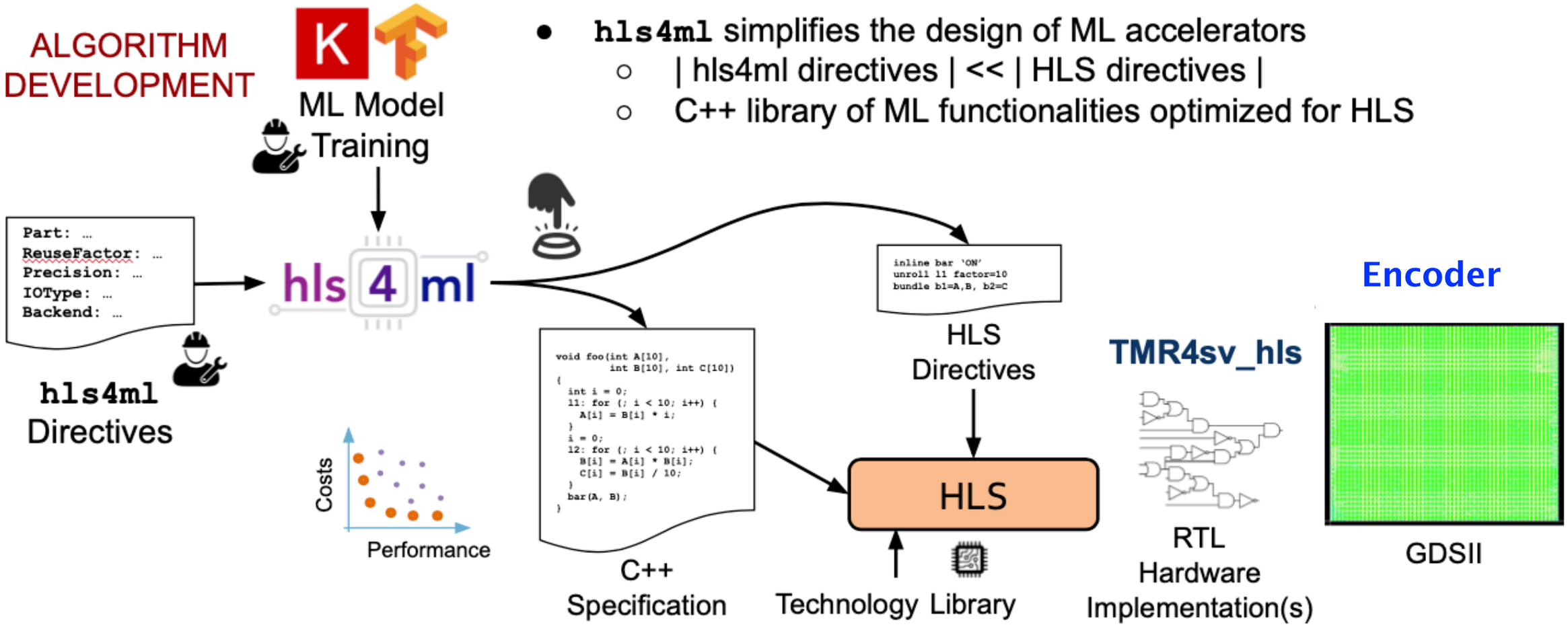
Quantify encoding performance as EMD to transform decoded image → input image.



Physics driven hardware co-design

Rapid prototyping and optimization of network achieved through

- **QKeras** : network development with **quantization-aware training** and physics simulation
- **hls4ml** : neural network description (h5 file e.g.) ➔ HLS-compliant C++ format
- **Catapult HLS** : C++ ➔ RTL
- **TMR4sv_hls** : Automated TMR for System Verilog



Rapid design optimization

- **Performance** : **EMD mean** and **RMS** are both important
- **Power and area** : scale with number of model **operations** and **parameters**

Lower EMD is better

	Network Architecture					Relative Power & Area		Relative Performance	
Test feature	**Geometry**	**# filter**	**kernel**	**stride**	**pooling**	**# params**	**# operations**	**EMD Mean**	**EMD RMS**
Reference	4x4x3	8	3x3	1	none	1.00	1.00	1.00	1.00
4x4x3 -> 8x8	**8x8**	8	3x3	1	none	2.73	1.76*	0.64	0.41
max pooling	8x8	8	3x3	1	**2x2**	0.71	0.97*	0.59	0.33
3x3 -> 5x5 kernel	8x8	8	**5x5**	1	2x2	0.99	2.76	0.64	0.35
pooling -> stride=2	8x8	8	3x3	**2**	**none**	**0.94**	**0.59**	**0.76**	**0.46**
8 -> 10 filters	8x8	**10**	3x3	2	none	1.17	0.73	0.73	0.43
8 -> 6 filters	8x8	**6**	3x3	2	none	0.70	0.44	0.85	0.57

* zero operations removed

- **Reference design** : presented in Fall 2020**

- **Final design** : 8×8 geometry + 8 filters + 3×3 kernel + stride =2
  - **50% power** and 80% area of reference (from simulation)
  - **2× better performance** (EMD RMS) than reference

** https://indico.cern.ch/event/924283/contributions/4105329/attachments/2152250/3630590/encoder_asic_fastml2020.pdf
https://www.eventclass.org/contxt_ieee2020/online-program/session?s=N-34#e280
https://www.eventclass.org/contxt_ieee2020/online-program/session?s=N-24#e189

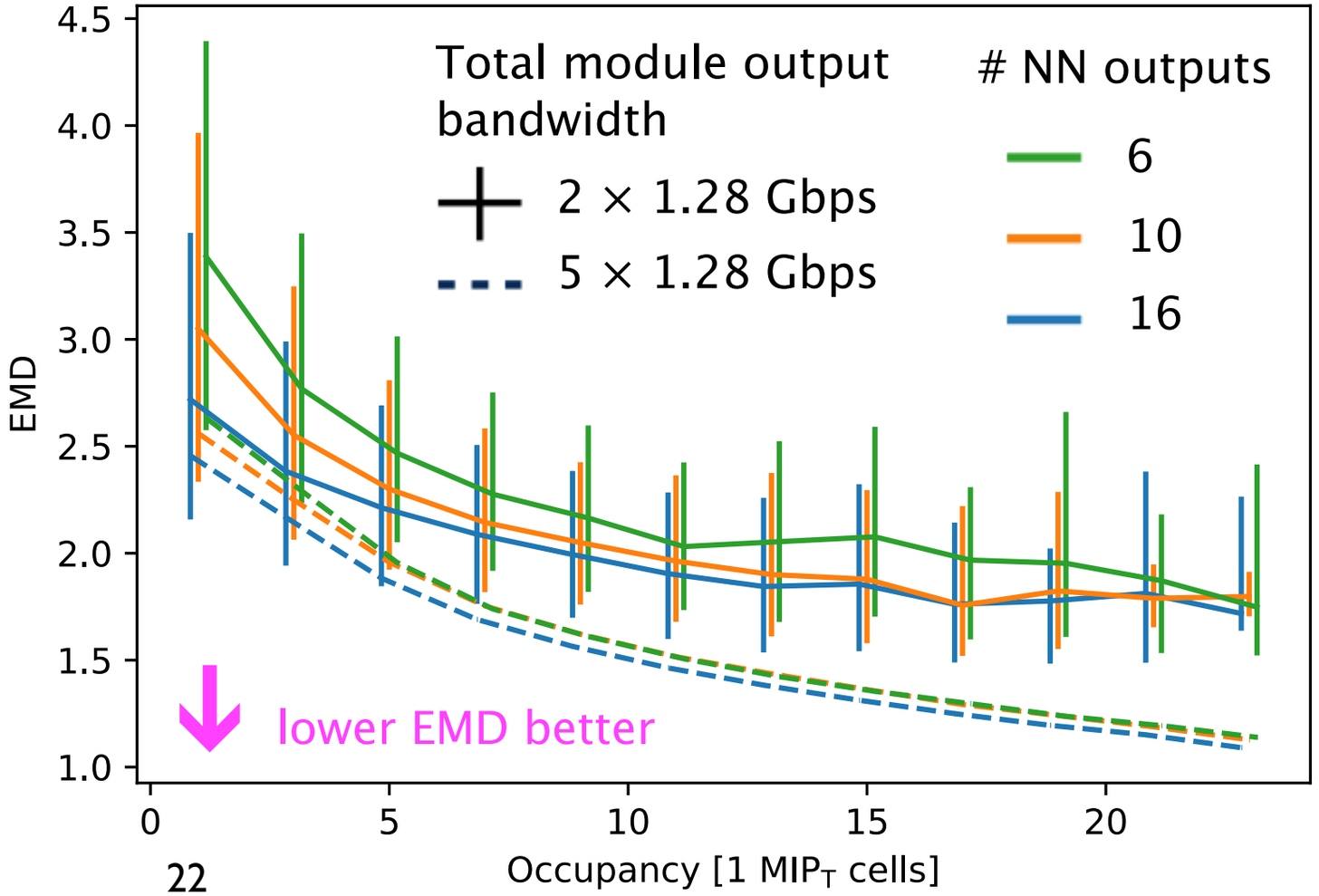
Optimization of NN output

- Better to use **many low-precision** or **fewer high-precision outputs**?
- Compare EMD performance keeping power and area fixed.
- Conclusion : **more lower-precision outputs is better**
    - for both high- and low-bandwidth scenarios
    - for full range of module occupancy

ECON ASIC allows user to **select any of 16×9 output bits** for transmission

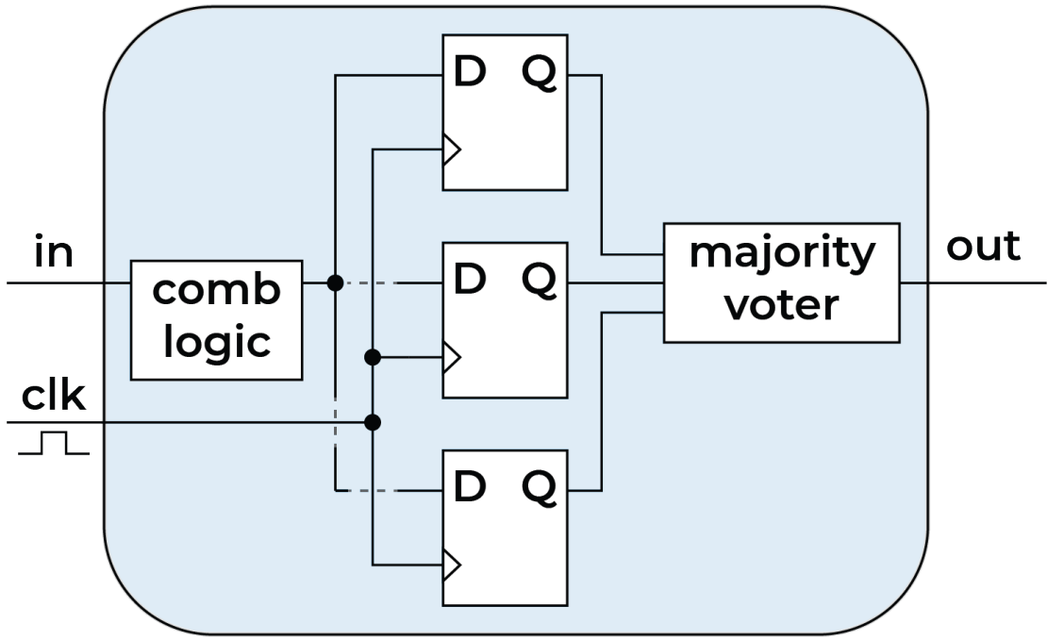
- Expect to use 16 × 3 (9) bits for low (high) occupancy zones.

- Corresponding precision used in **QKeras quantization-aware training optimizes network** for programmed output configuration.



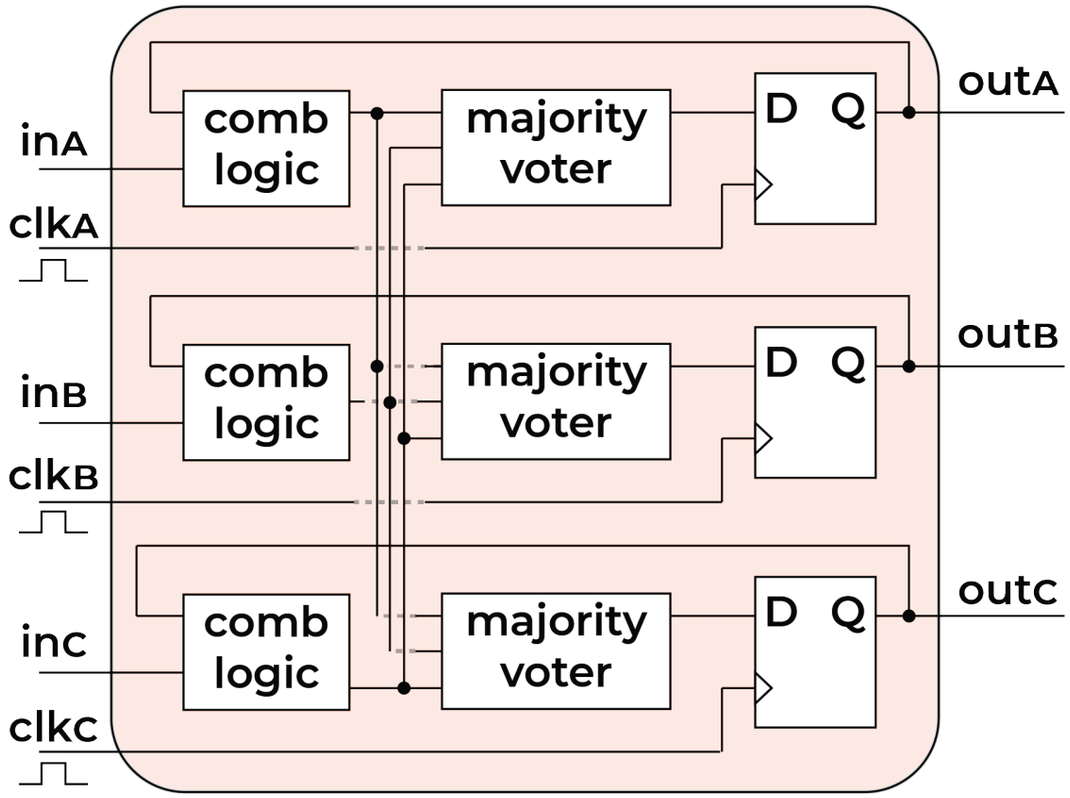
Single event effect mitigation

Data path : Encoder & Convertor



- New data every 25ns
- Triplicate registers without auto-correction

Configuration : I²C secondary

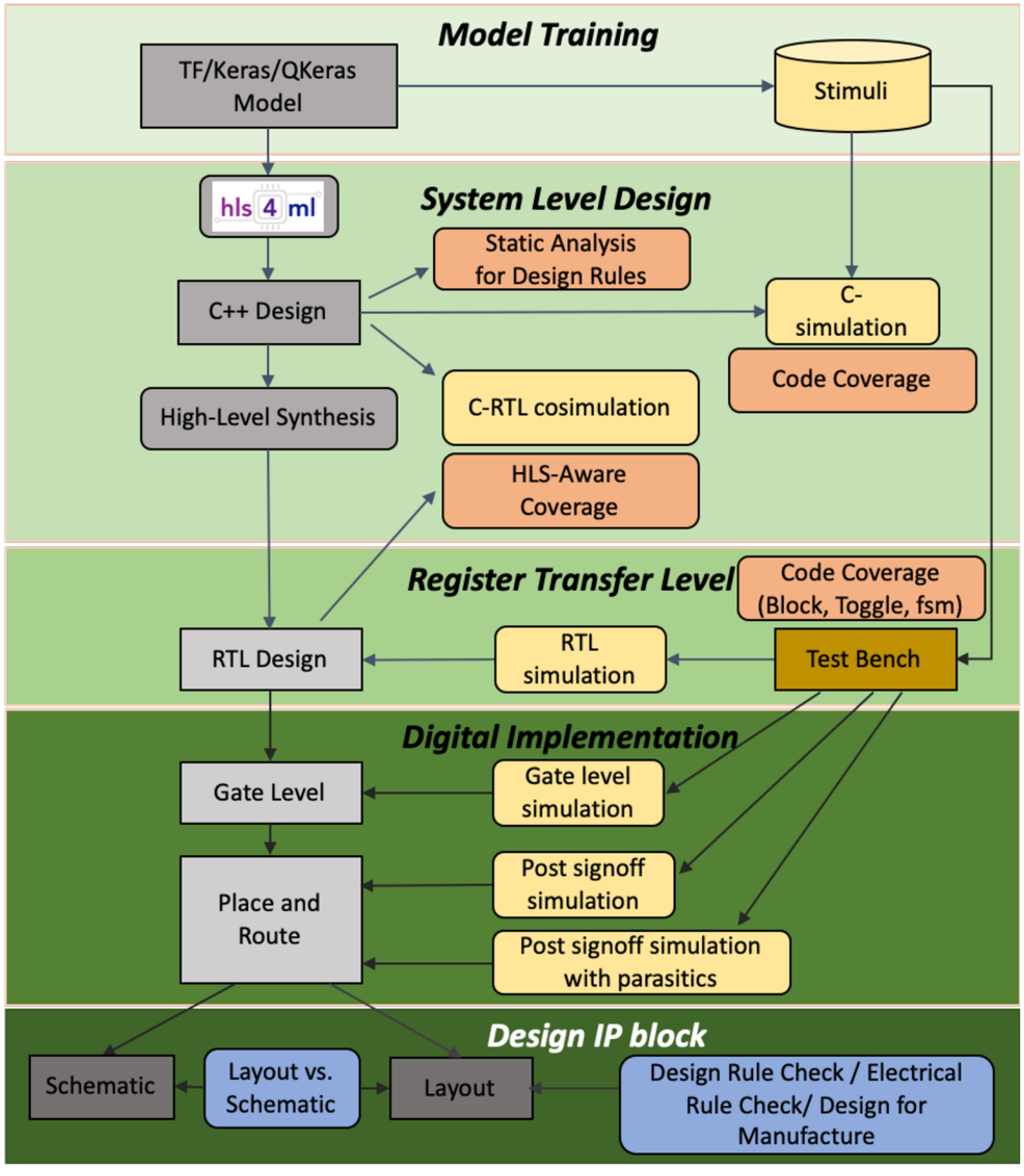


- Long term weights storage
- Triplicate registers, logic, and clocks
- Auto-correction included

Design and verification methodology

Verification performed at each stage of design:

- Model training
- hls4ml
- Catapult HLS
- RTL
- Synthesis
- Place and route
- LVS and DRC



Design and verification methodology

Step	Type	Run Time	Iterations	Size
Model generation	D	1s	50–100	1.1k lines of C++
C Simulation	V	1s		
HLS	D	30 min	3–100	40k lines of verilog
RTL simulation	V	1 min		
Logic synthesis	D	6 hrs	6	750k gates
Gate-level sim	V	30 min		
Place and route	D	50 hrs		780k gates
Post-layout sim	V	1 hrs		
Post-layout parasitic sim	V	2 hrs		
SEE simulation	V	4 hrs		
Layout	D	20 min	1	7.6M transistors
LVS and DRC	V	1 hr		

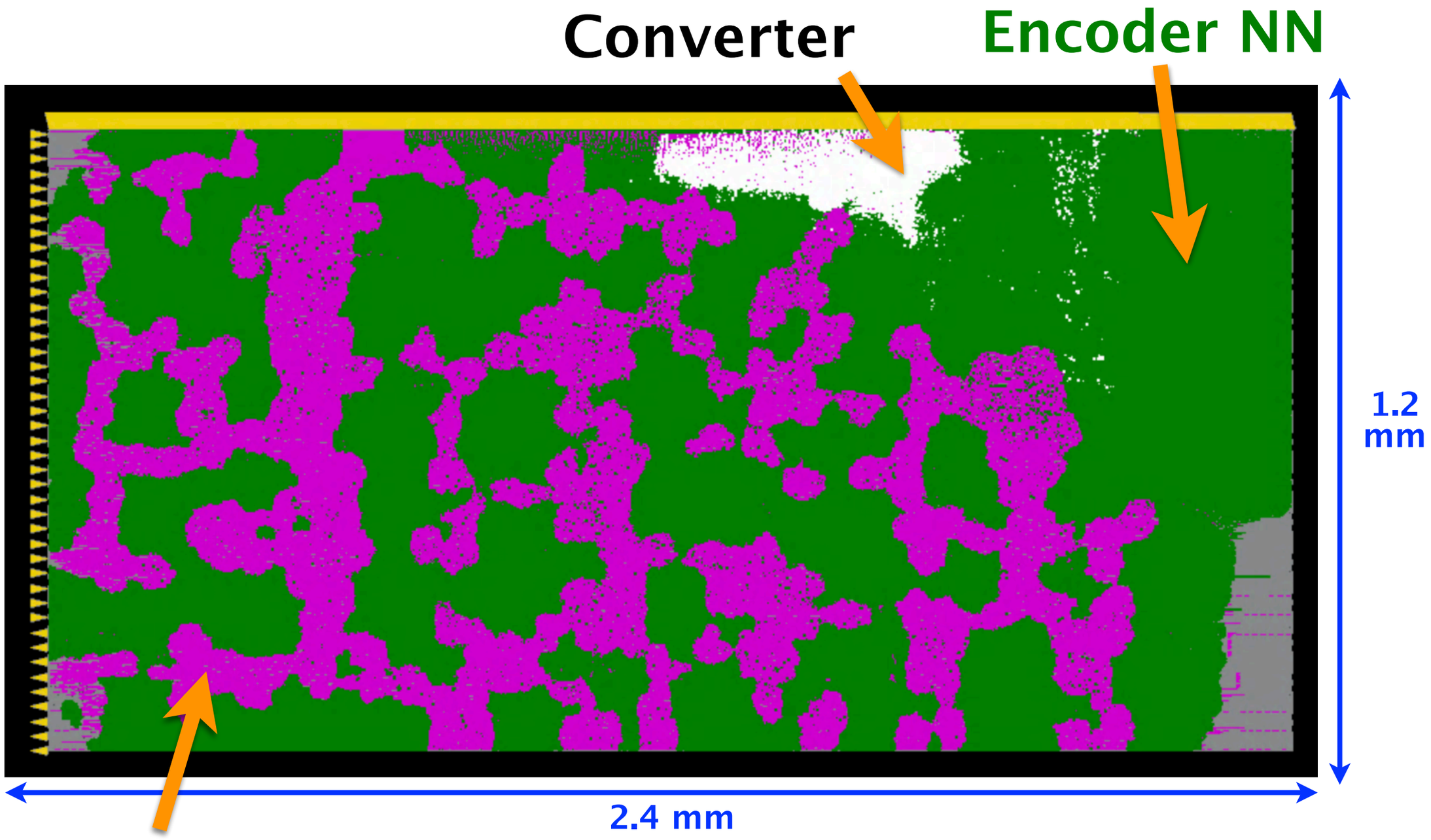
Network optimization

Design optimization

Increasing time and complexity

Place and route

- Integrated design to avoid routing congestion from 14k bits of weights (programmable via I^2C) connected from periphery.



Design Performance Metrics

Physics performance studies in progress → preliminary performance with non-optimized training **comparable to traditional threshold algorithm.**

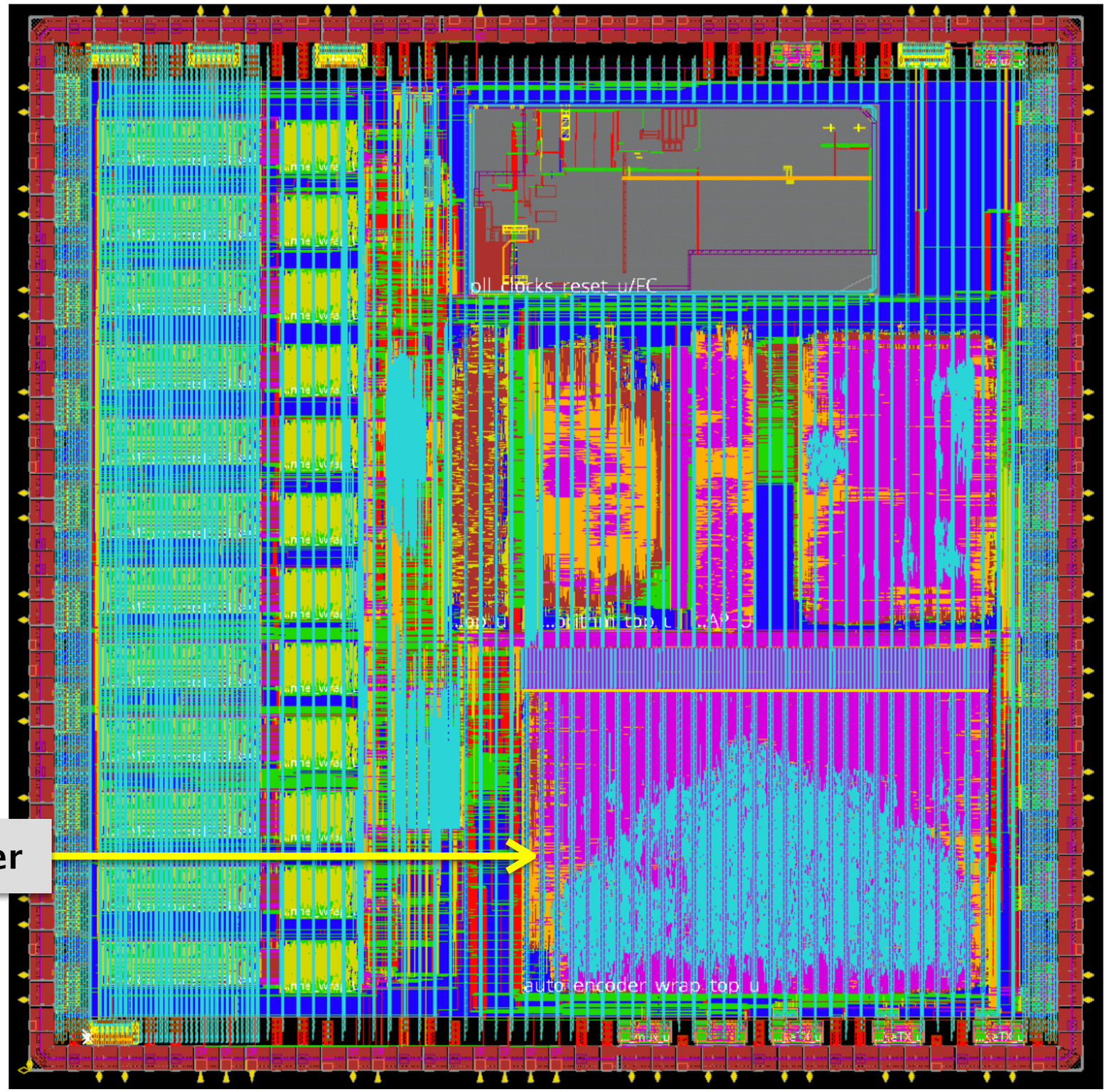
Requirements	
Rate	40 MHz
Total ionizing dose	200 Mrad
High energy hadron flux	1×10^7 cm^2/s

Metric	Simulation	Target
Power	48 mW	<100 mW
Energy / inference	1.2 nJ	N/A
Area	2.88 mm^2	<4 mm^2
Gates	780k	N/A
Latency	50 ns	<100 ns

* EMD RMS

ECON-T-P1 submitted

- **ECON-T-P1 submitted** for fabrication on June 28, 2021.
- Chips expected to reach Fermilab in early October 2021.
- We are ready and excited to test the chip and evaluate the performance of NN encoder



* EMD RMS

Summary

- **Autoencoder neural network for on-detector data compression**.
  - Low power, low latency, radiation tolerant, fully re-configurable
  - 65nm LP CMOS
  - Prototypes will be tested in Fall 2021

- **Established design and verification methodology** based on **hls4ml + Catapult HLS** allows rapid progression from algorithm development through circuit implementation.

- Optimized network provides **2× better performance** at **~50% power** of reference network.

Acknowledgements

- **ECON design team for inclusion in ECON ASIC :** Davide Braga, Mike Hammer, Jim Hoff, Paul Rubinov, Alpana Shenai, Cristina Mantilla Suarez, Chinar Syal, Xiaoran Wang, Ralph Wickwire

- **CMS HGCAL for simulated training images**
  - Jean-Baptiste Sauvan for simulation development
  - Andre Davide for useful discussion on network optimization

- **hls4ml developers** : Javier Duarte, Phil Harris, Vladimir Loncar, Jennifer Ngadiuba, Maurizio Pierini, Sioni Summers
  https://fastmachinelearning.org/hls4ml/

- **Mentor/Siemens Catapult HLS** : Sandeep Garg and Anoop Saha

- **Cadence Innovus and Incisive** : Bruce Cauble and Brent Carlson

Additional material

Precision of weights and variables

- Diagram is example for 4×4×3 reference network – same structure as final 8×8 network
- **Weights** are all 6b

For final 8×8 network:

- **hidden layer neurons:**
  - 8b fraction
  - sufficient integer bits to cover theoretical max value
- **output neurons**:
  - 9b total
  - 1b integer
  - covers maximum value from physics simulation