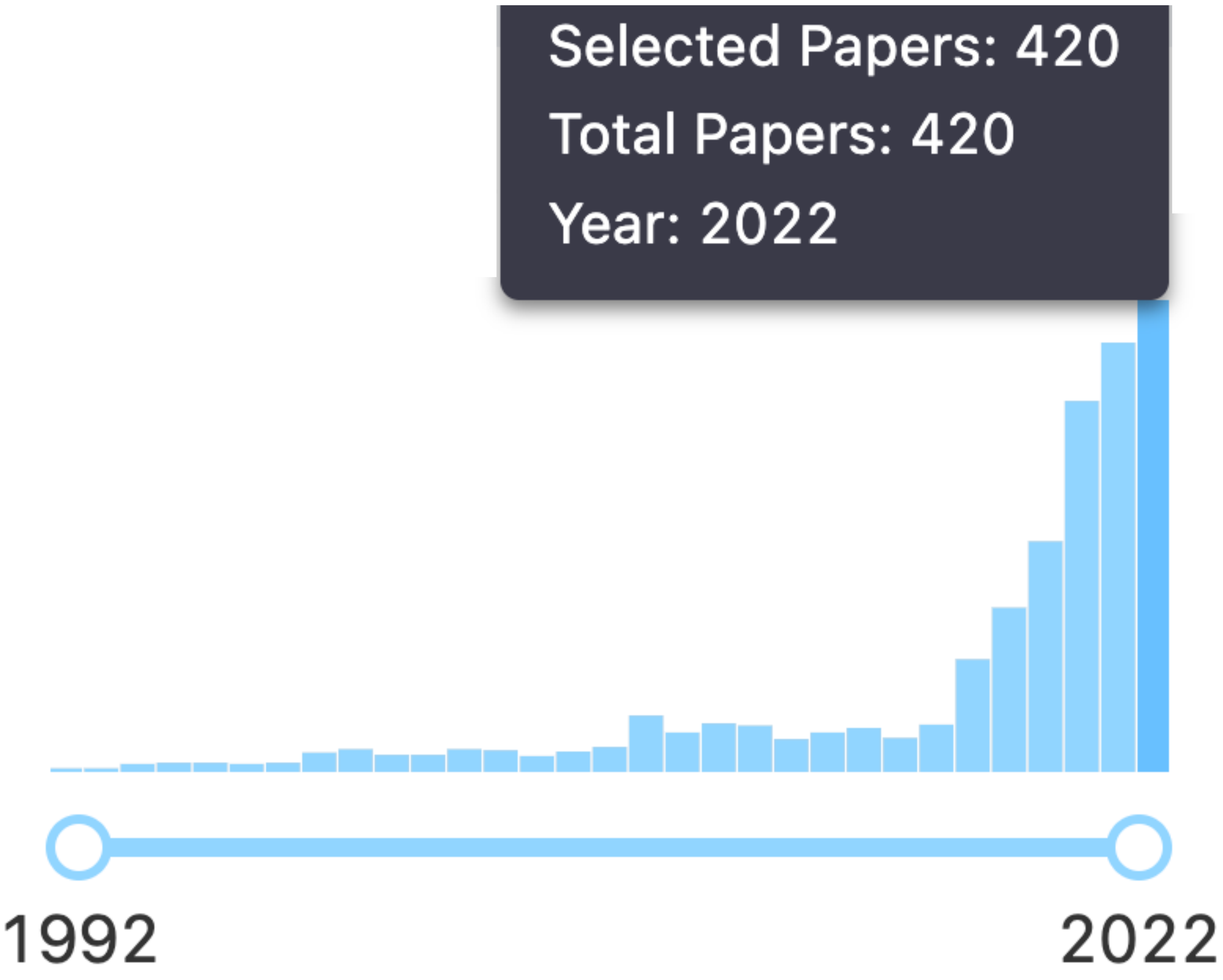
ETH zürich

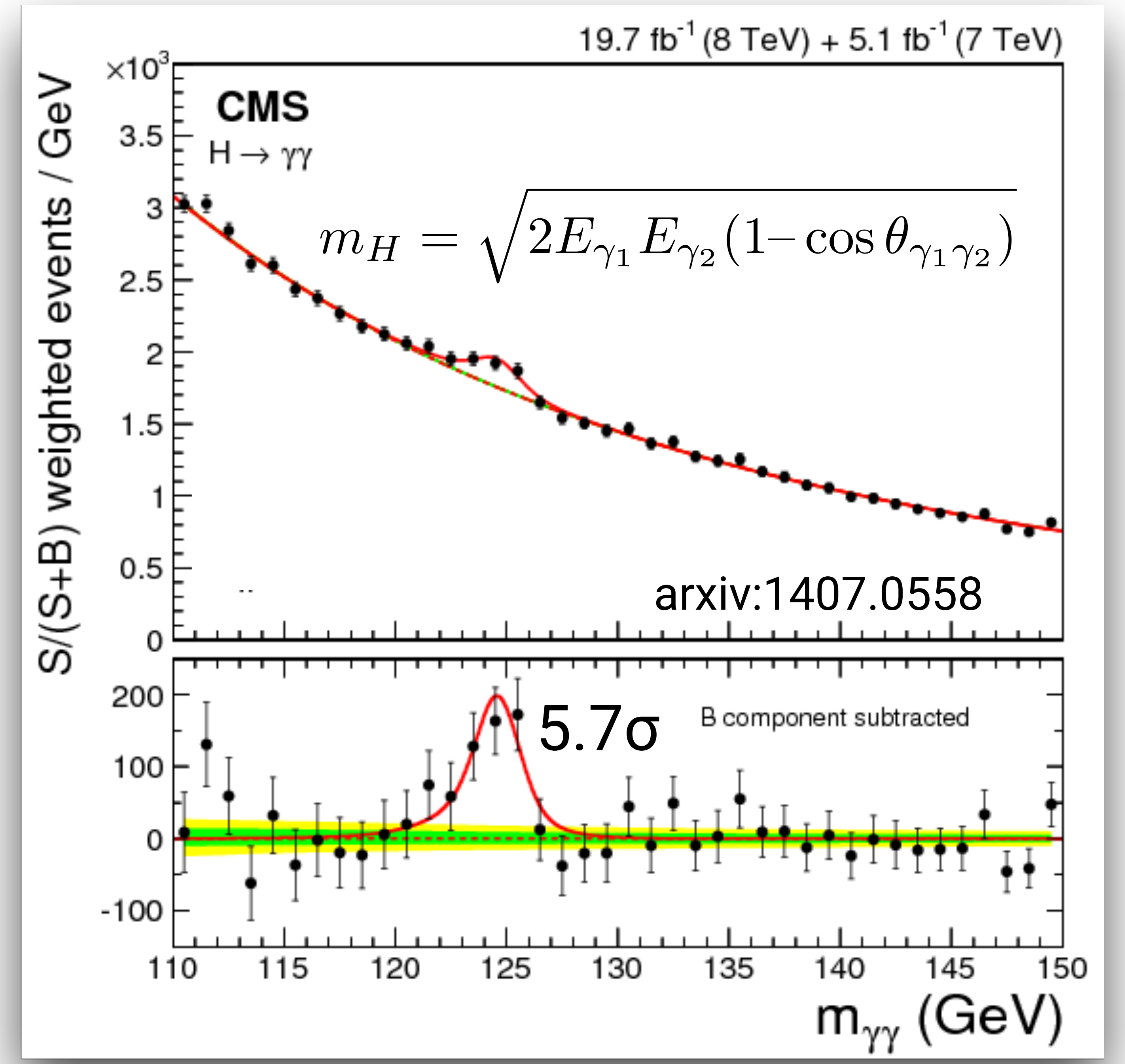
New developments in Machine Learning in particle physics

Thea Klæboe Årrestad (ETH Zürich)
thea.aarrestad@cern.ch

Spåtind 2023

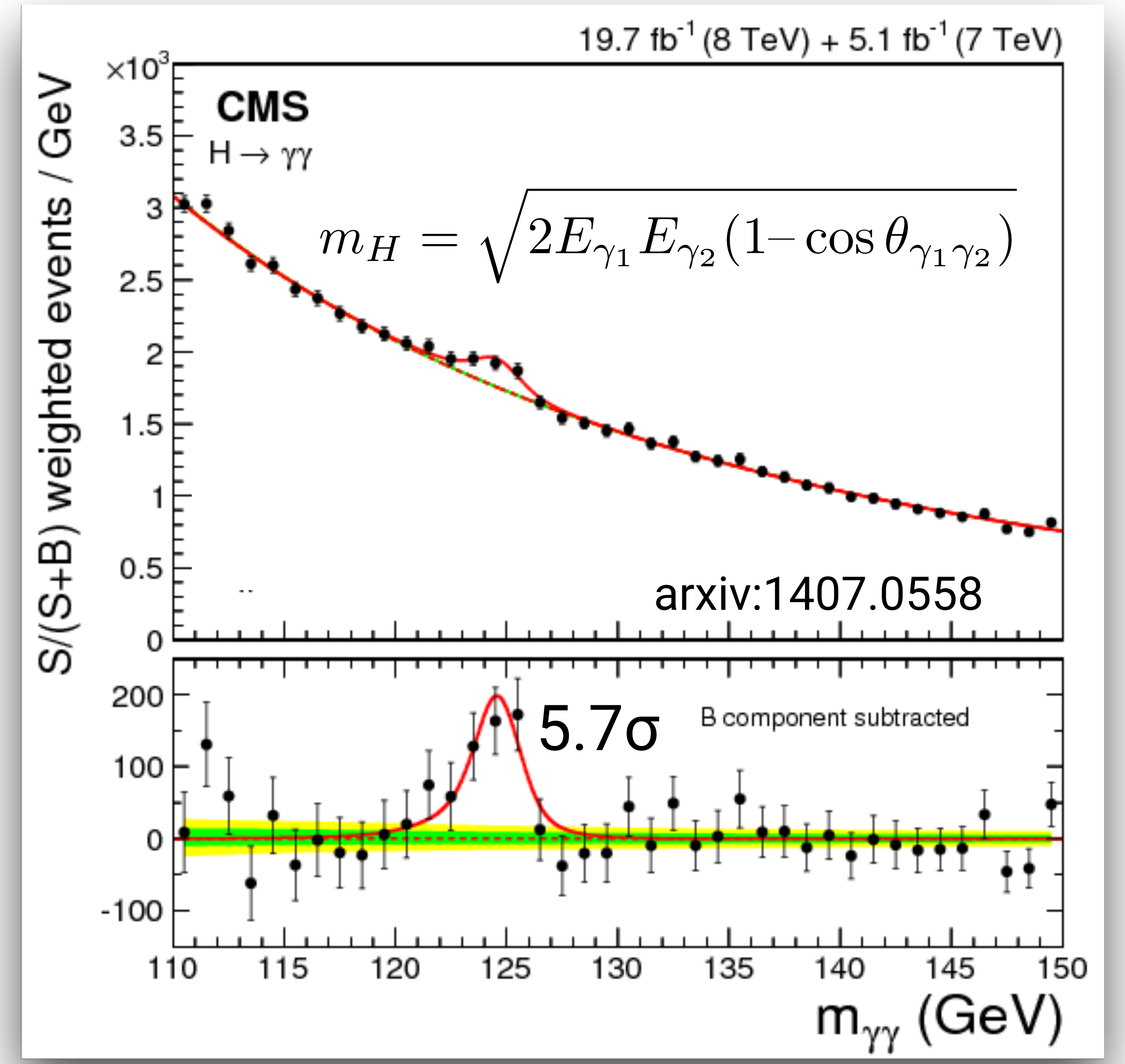
Inspire:
("machine learning" or "deep learning" or neural) and (hep-ex or hep-ph or hep-th)

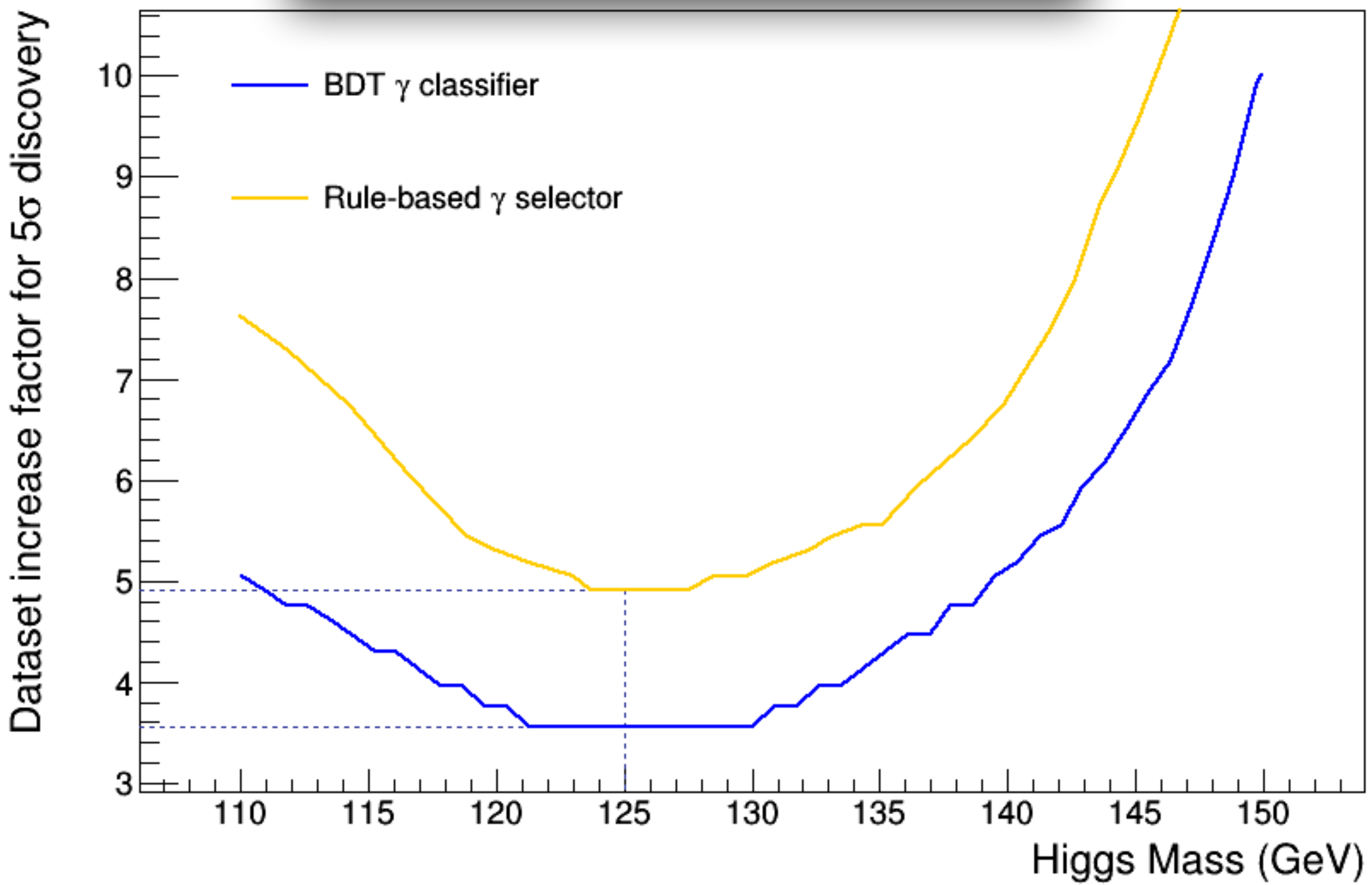




19.7 fb^{-1} (8 TeV) + 5.1 fb^{-1} (7 TeV)
CMS
H → γγ
$m_H = \sqrt{2E_{\gamma_1}E_{\gamma_2}(1-\cos\theta_{\gamma_1\gamma_2})}$
arxiv:1407.0558
5.7σ
B component subtracted
S/(S+B) weighted events / GeV
$m_{\gamma\gamma}$ (GeV)

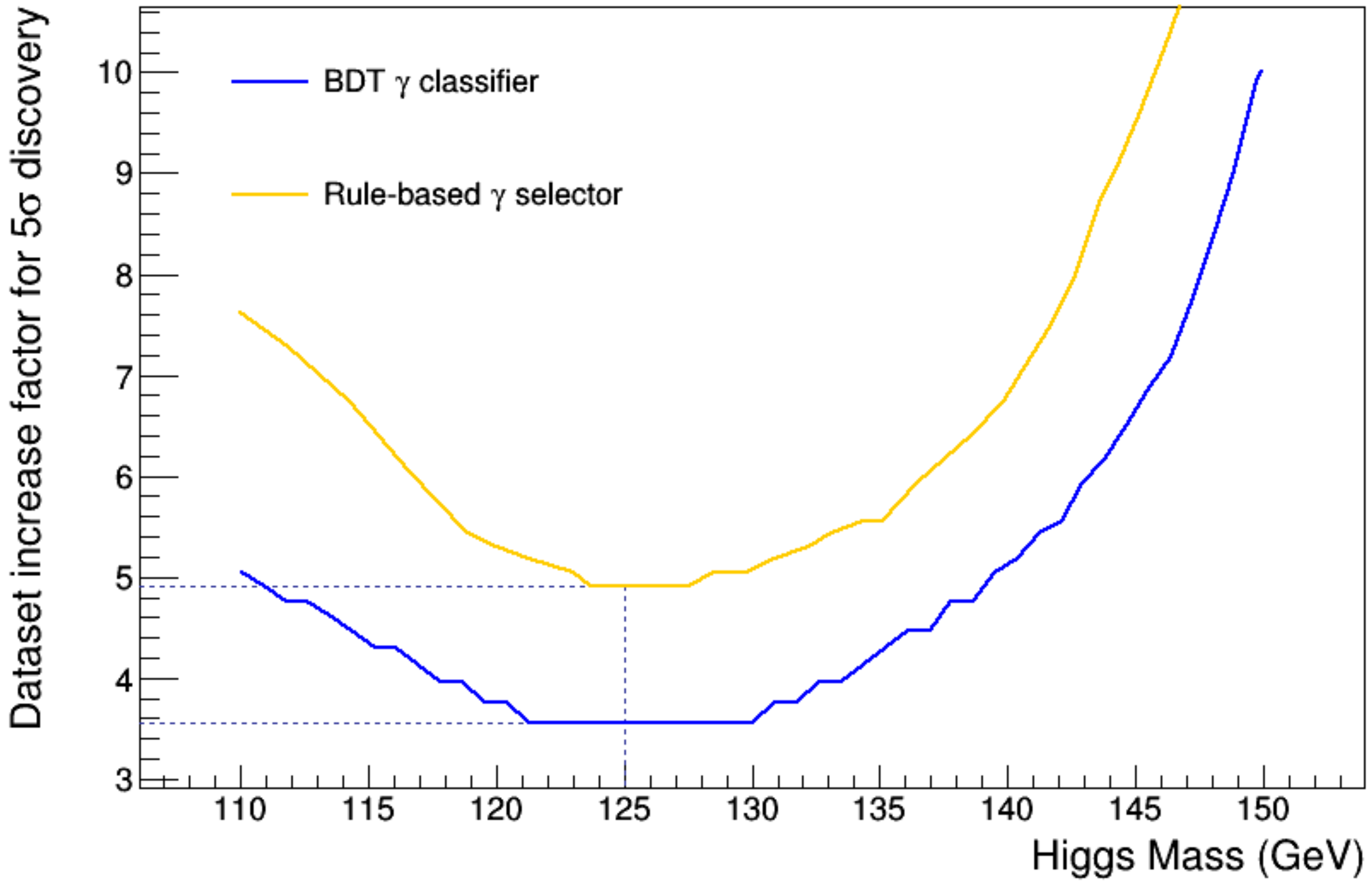
Nature Review





Nature Review

Analysis	Years of data collection	Sensitivity without machine learning	Sensitivity with machine learning	Ratio of P values	Additional data required
CMS[24] $H \rightarrow \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \rightarrow \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \rightarrow bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \rightarrow bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \rightarrow bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%



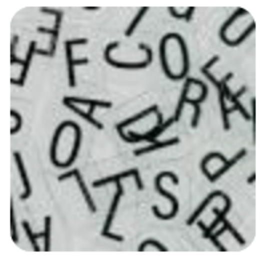
Google

chat gpt

All | News | Images | Videos | Books | More | Tools

About 10,700 results (0.31 seconds)

The New York Times



Opinion | ChatGPT Has a Devastating Sense of Humor

The chat bot makes a lot of mistakes. But it's fun to talk to, and it knows its limitations.

3 weeks ago

The New York Times

Can ChatGPT Make This Podcast?

It's writing podcast scripts, finishing students' homework and correcting mistakes in computer code: ChatGPT, the A.I. chatbot from OpenAI,...

4 weeks ago

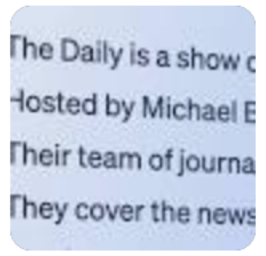
The New York Times

How to Use ChatGPT and Still Be a Good Person

It's a turning point for artificial intelligence, and we need to take advantage of these tools without causing harm to ourselves or others.

2 weeks ago

The New York Times



Did Artificial Intelligence Just Get Too Smart?

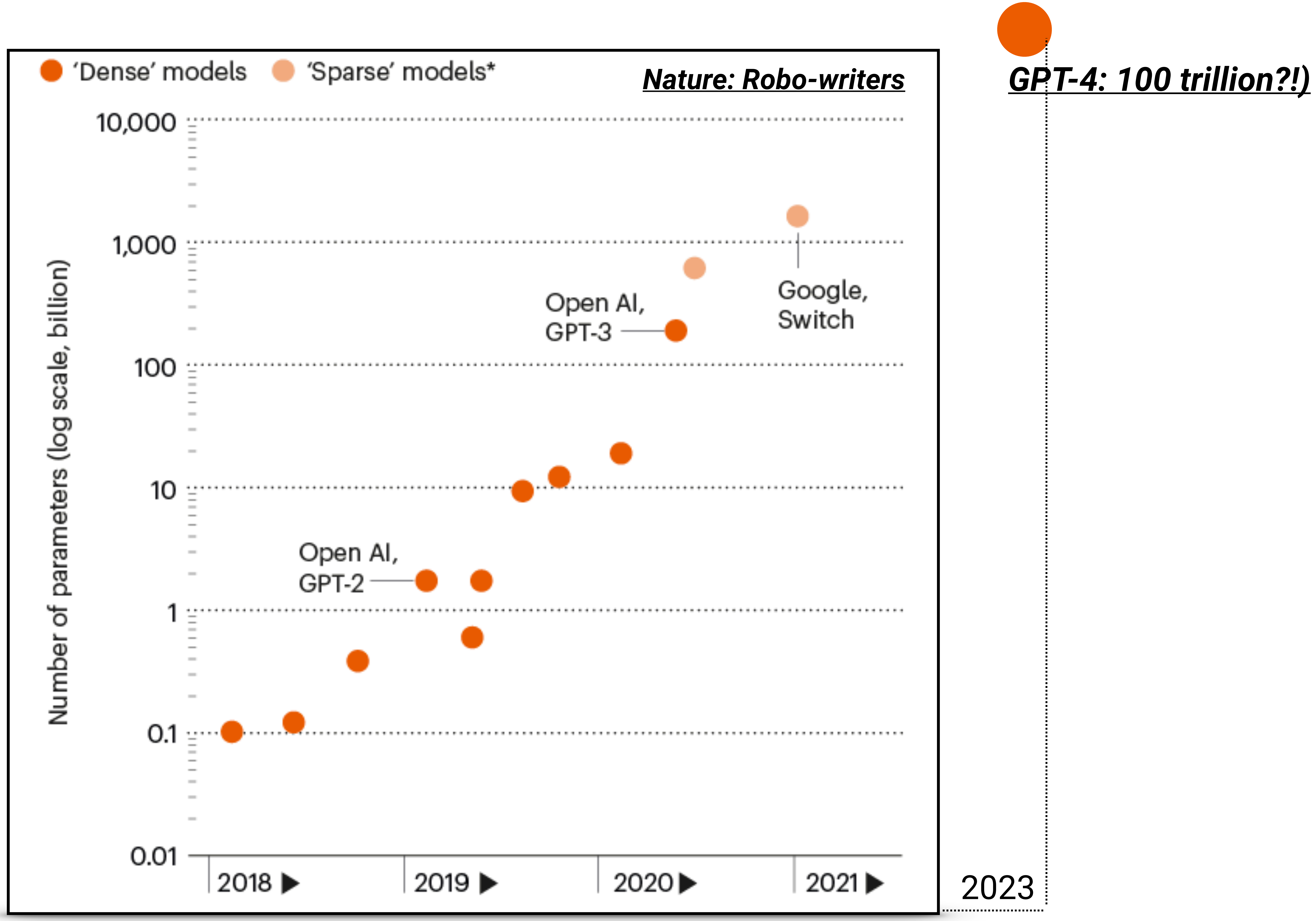
The power and potential of a technology called ChatGPT have led some to claim it heralds a new era in computing.

3 weeks ago

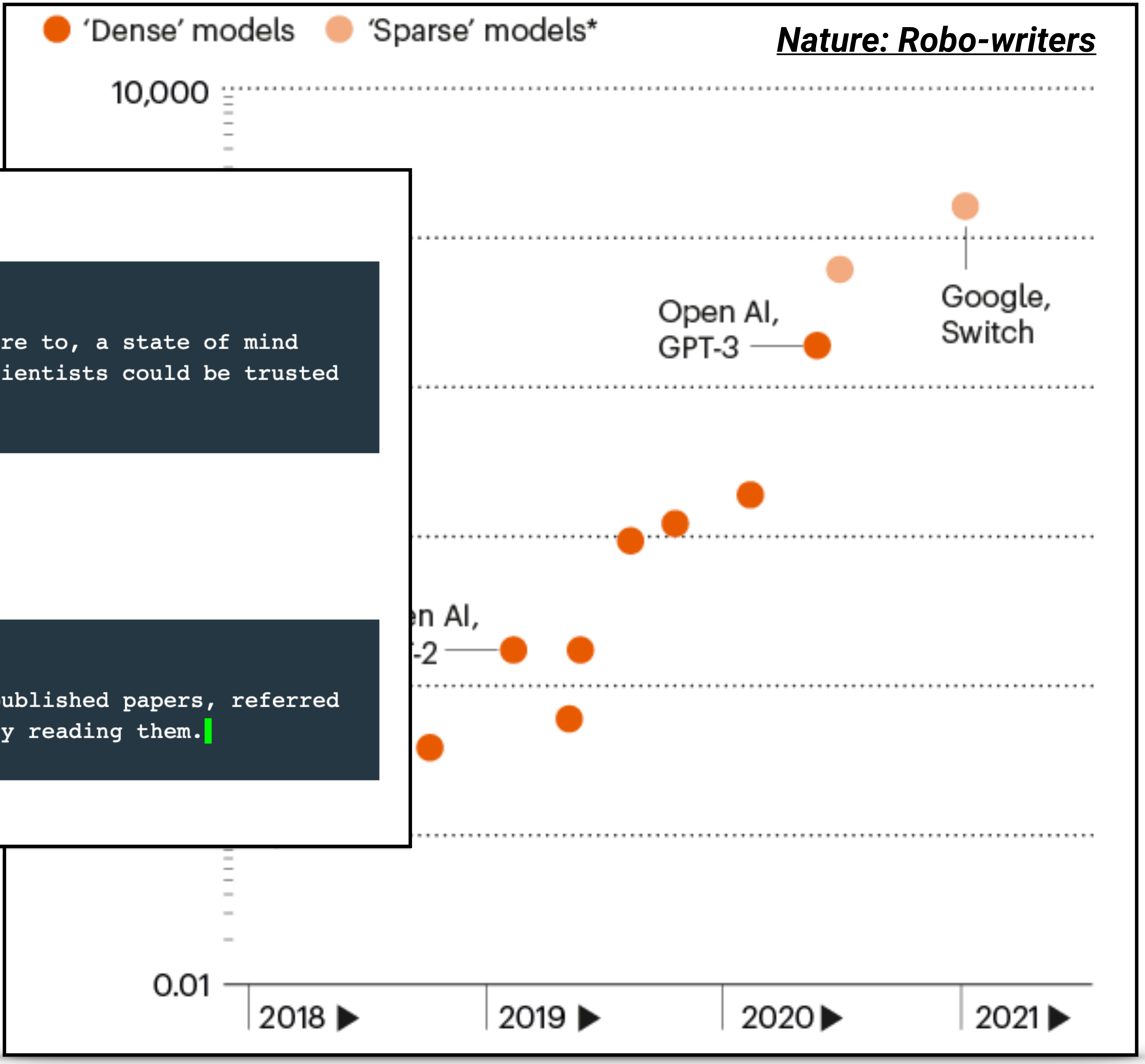
The New York Times

ChatGPT is Social Media's Newest Star

Social media's newest star is a robot: a program called ChatGPT that tries to answer questions like a person. Since its debut last week,...



GPT-3: 175 billion parameters (0.16% of the human brain)



Example prompt

```
Rigor [adj.]
Something for scientists to aspire to, a state of mind
that would not be required if scientists could be trusted
to do their job.
```

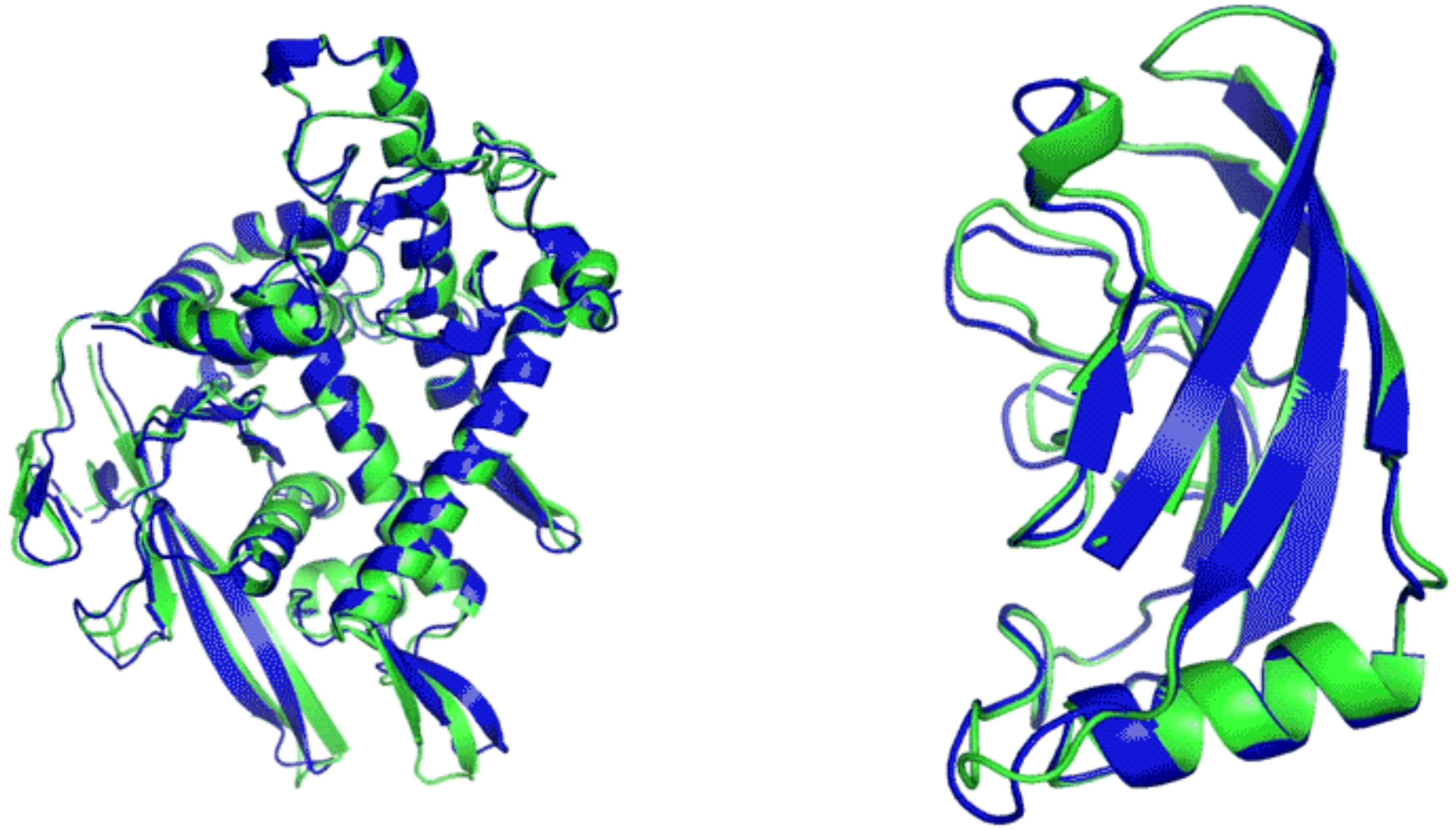
View next definition

GPT-3's output: 1 of 10

```
The Literature [noun]
A name given to other people's published papers, referred
to by scientists without actually reading them.
```

Gwern.net

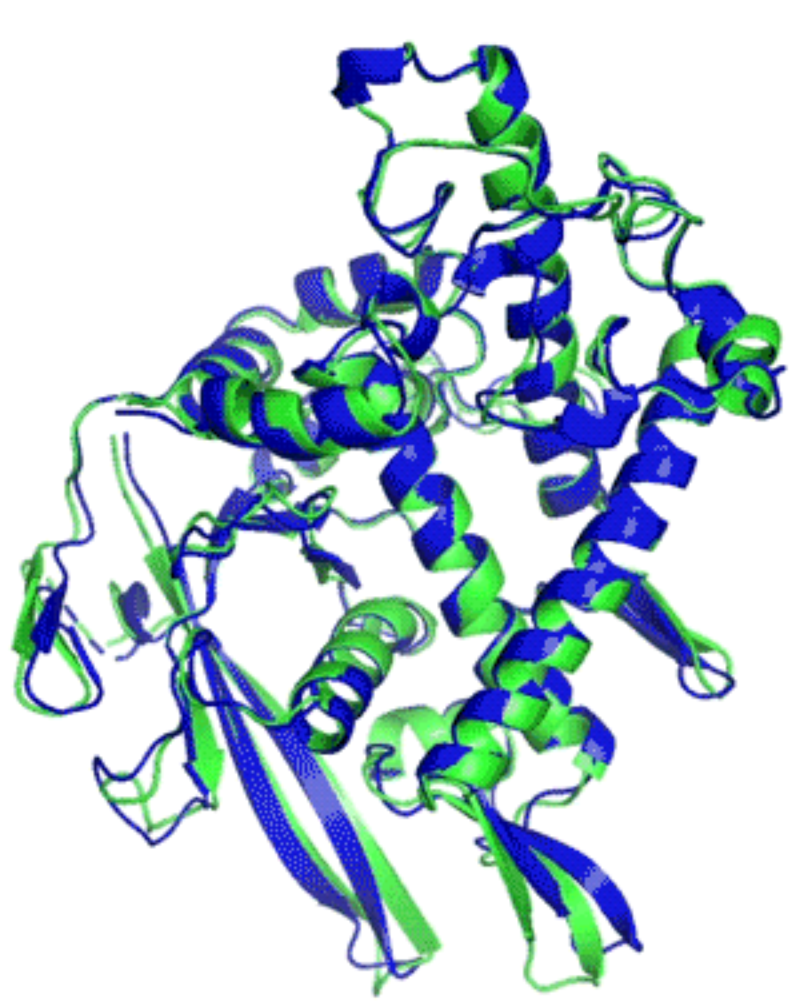
GPT-3: 175 billion parameters (0.16% of the human brain)



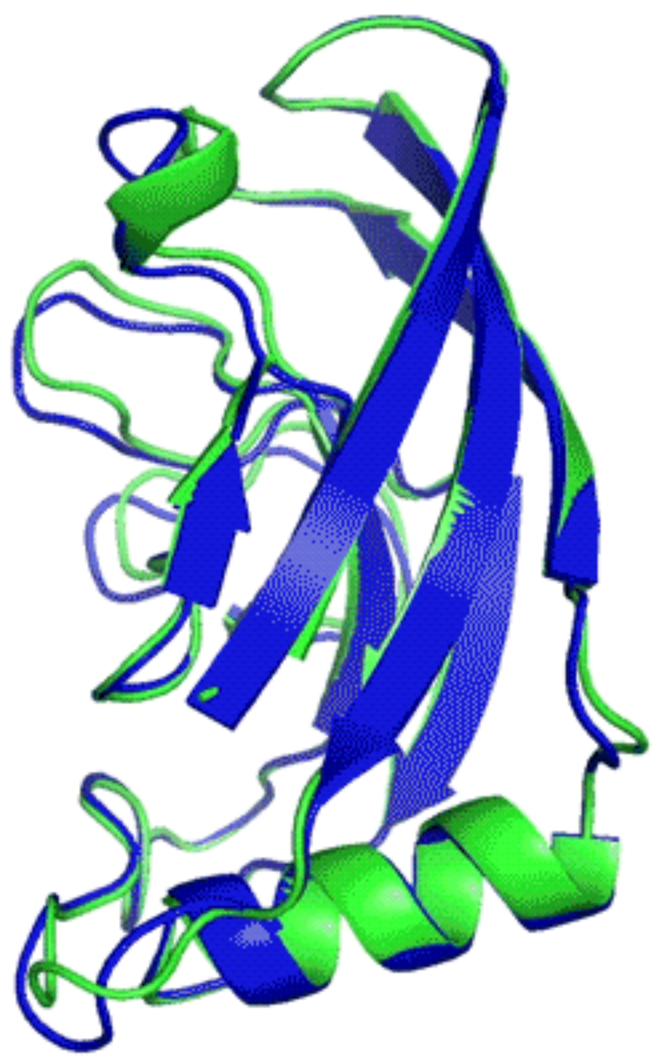
T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

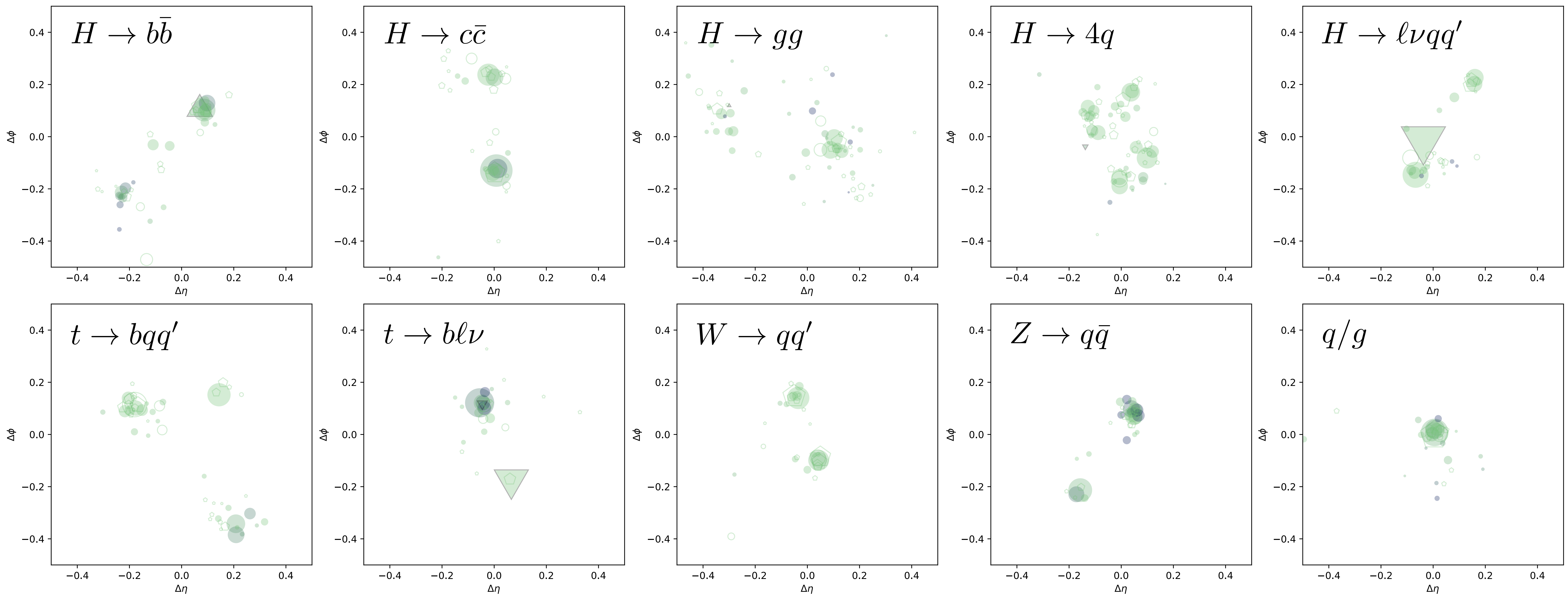


T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction

100 million jets for training

	Accuracy	# params
PFN	0.772	86.1 k
P-CNN	0.809	354 k
ParticleNet	0.844	370 k
ParT	**0.861**	2.14 M
ParT (plain)	0.849	2.13 M



"Particle Transformer For Jet Tagging" H. Qu, C. Li, S. Qian

T

I'm a bit late with my presentation, could you summarise the latest progress in Machine Learning for Particle Physics?

T

I'm a bit late with my presentation, could you summarise the latest progress in Machine Learning for Particle Physics?

Sure! In recent years, there has been significant progress in the application of machine learning techniques to particle physics. Some key areas where machine learning has made significant contributions include:

1. Classification of particles: Machine learning algorithms, particularly deep learning algorithms, have been successfully used to classify different types of particles based on their properties and characteristics. For example, they can be used to identify and distinguish between different types of hadrons (such as protons and neutrons) or leptons (such as electrons and muons).
2. Event reconstruction: Machine learning algorithms have also been used to improve the accuracy of event reconstruction in particle physics experiments. Event reconstruction refers to the process of reconstructing the collision events that occurred in a particle accelerator from the raw data collected by detectors. Machine learning algorithms can be used to identify patterns and correlations in the data that can help scientists more accurately reconstruct the events that occurred.
3. Particle identification: Machine learning algorithms have been used to identify and classify particles based on their properties and characteristics. This can be particularly useful in the context of high-energy particle colliders, where large numbers of particles are produced in each collision event.
4. Simulation and modeling: Machine learning algorithms have also been used to improve the accuracy of simulations and models in particle physics. For example, they have been used to model the interactions between particles and the underlying physical processes that take place during collisions.
5. Data analysis and interpretation: Machine learning algorithms have been used to analyze and interpret large datasets generated by particle physics experiments. For example, they have been used to identify patterns and correlations in the data that can help scientists better understand the underlying physical processes at play.

Overall, machine learning has made significant contributions to particle physics and is expected to continue to play a major role in this field in the future.

T

I'm a bit late with my presentation, could you summarise the latest progress in Machine Learning for Particle Physics?

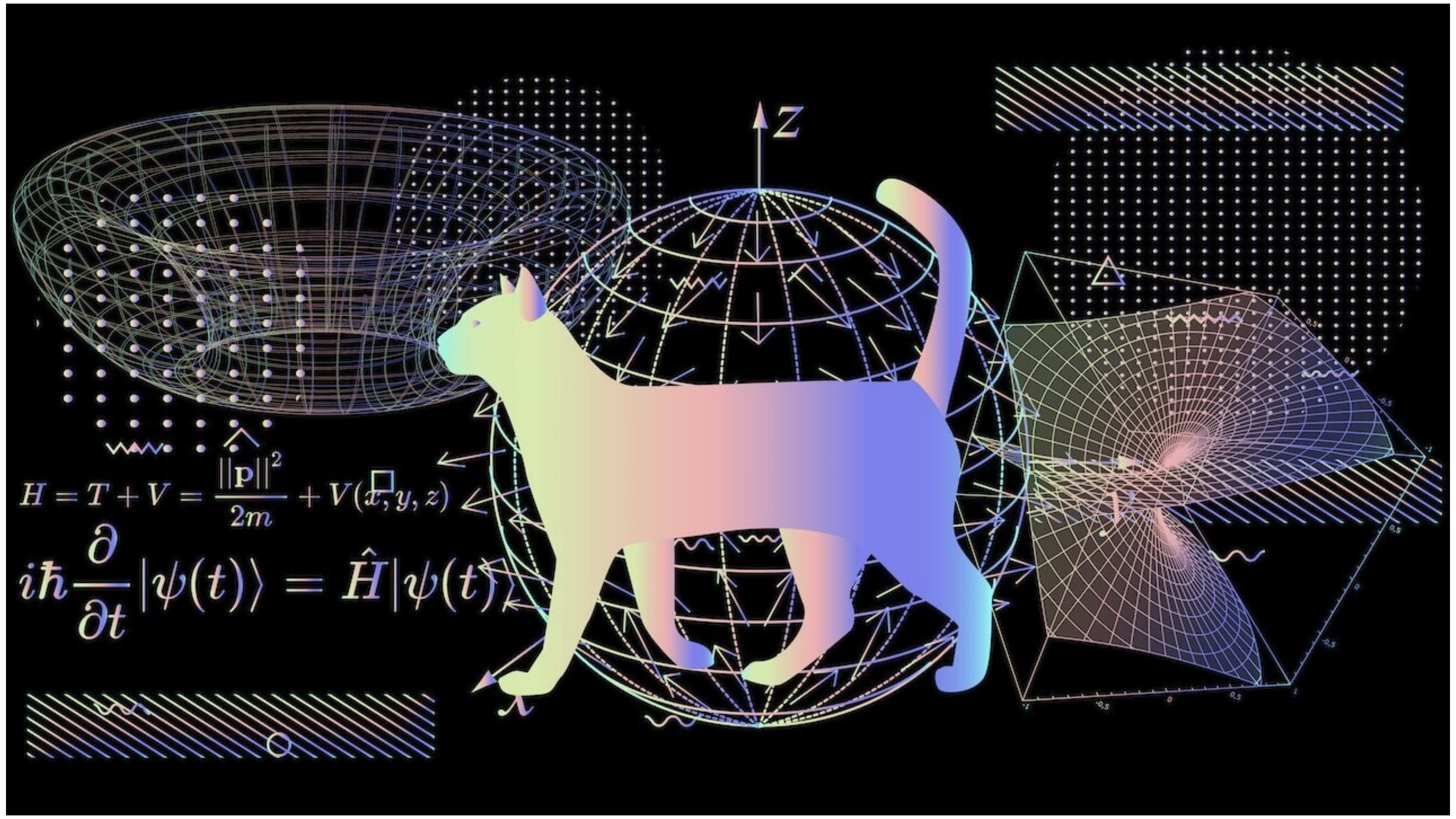
Sure! In recent years, there has been significant progress in the application of machine learning techniques to particle physics. Some key areas where machine learning has made significant contributions include:

1. Classification of particles: Machine learning algorithms, particularly deep learning algorithms, have been successfully used to classify different types of particles based on their properties and characteristics. For example, they can be used to identify and distinguish between different types of hadrons (such as protons and neutrons) or leptons (such as electrons and muons).
2. Event reconstruction: Machine learning algorithms have also been used to improve the accuracy of event reconstruction in particle physics experiments. Event reconstruction refers to the process of reconstructing the collision events that occurred in a particle accelerator from the raw data collected by detectors. Machine learning algorithms can be used to identify patterns and correlations in the data that can help scientists more accurately reconstruct the events that occurred.
3. Particle identification: Machine learning algorithms have been used to identify and classify particles based on their properties and characteristics. This can be particularly useful in the context of high-energy particle colliders, where large numbers of particles are produced in each collision event.
4. Simulation and modeling: Machine learning algorithms have also been used to improve the accuracy of simulations and models in particle physics. For example, they have been used to model the interactions between particles and the underlying physical processes that take place during collisions.
5. Data analysis and interpretation: Machine learning algorithms have been used to analyze and interpret large datasets generated by particle physics experiments. For example, they have been used to identify patterns and correlations in the data that can help scientists better understand the underlying physical processes at play.

Overall, machine learning has made significant contributions to particle physics and is expected to continue to play a major role in this field in the future.

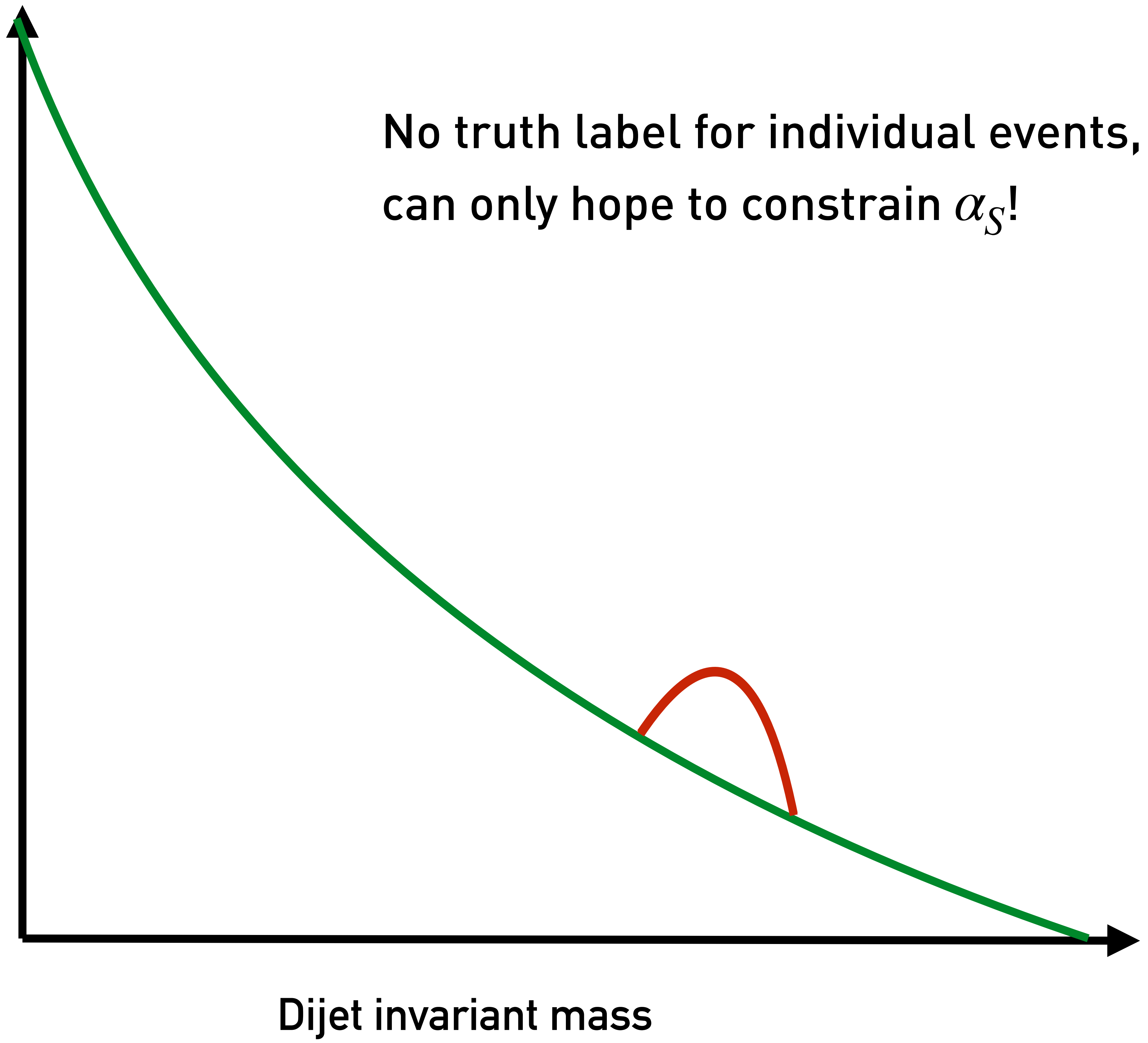
What makes particle physics special?

$$dP^n_{data} = |M_S + M_B|^2 dp_1 dp_2 \ldots dp_n$$



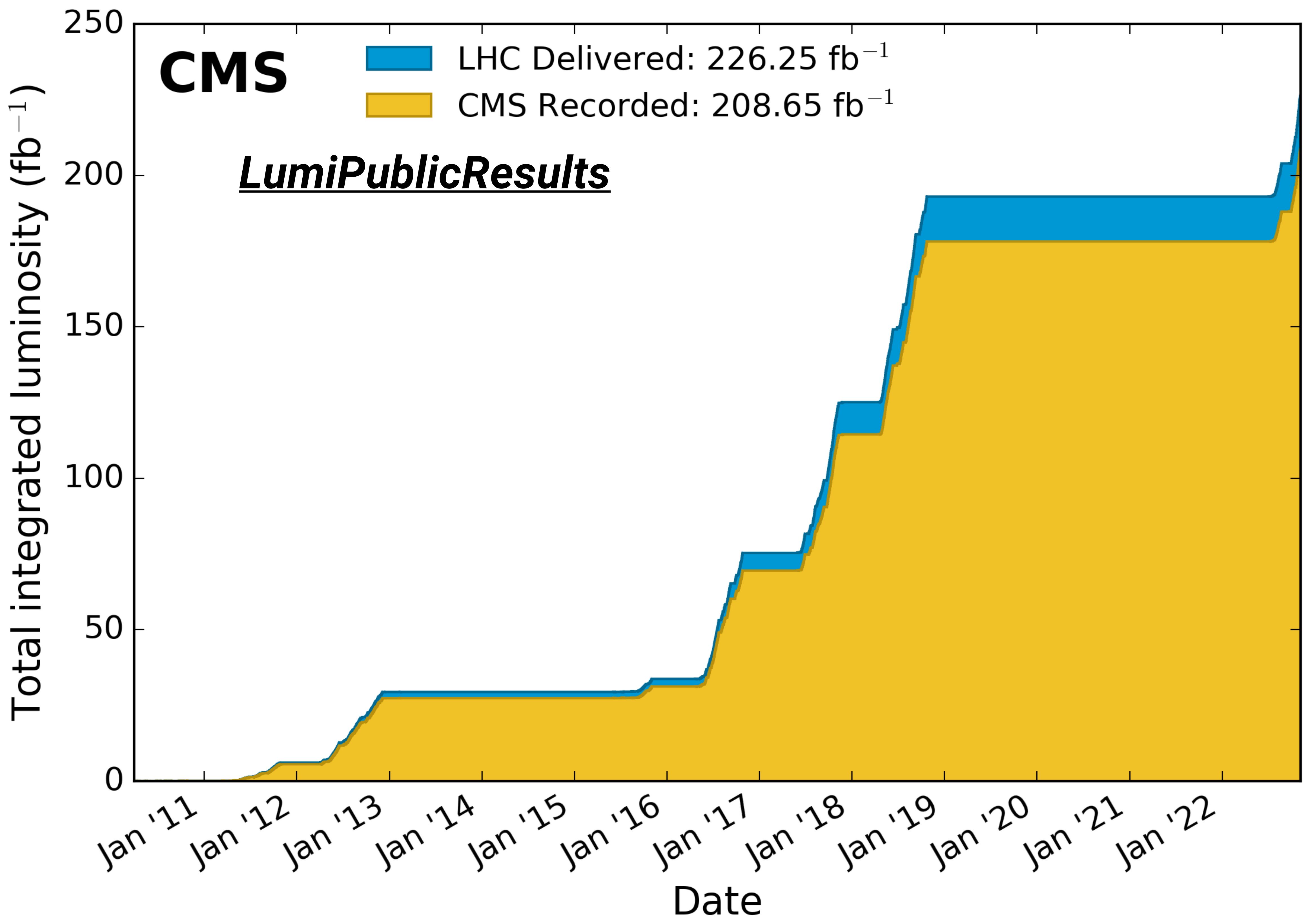
$$M_S M_B{}^* + M_B M_S{}^*$$

$$P_{data} = \alpha_S P_S + \alpha_B P_B$$

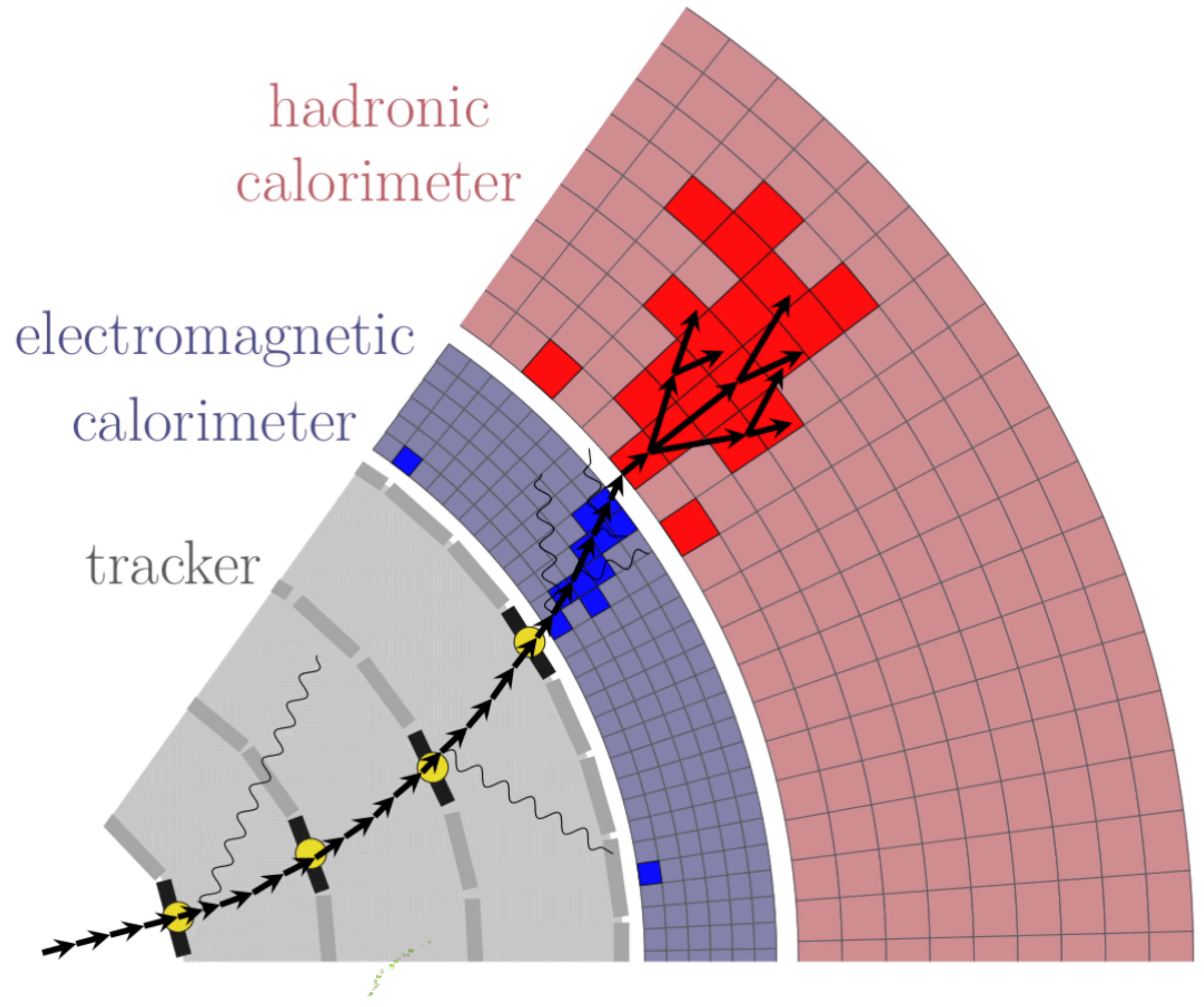


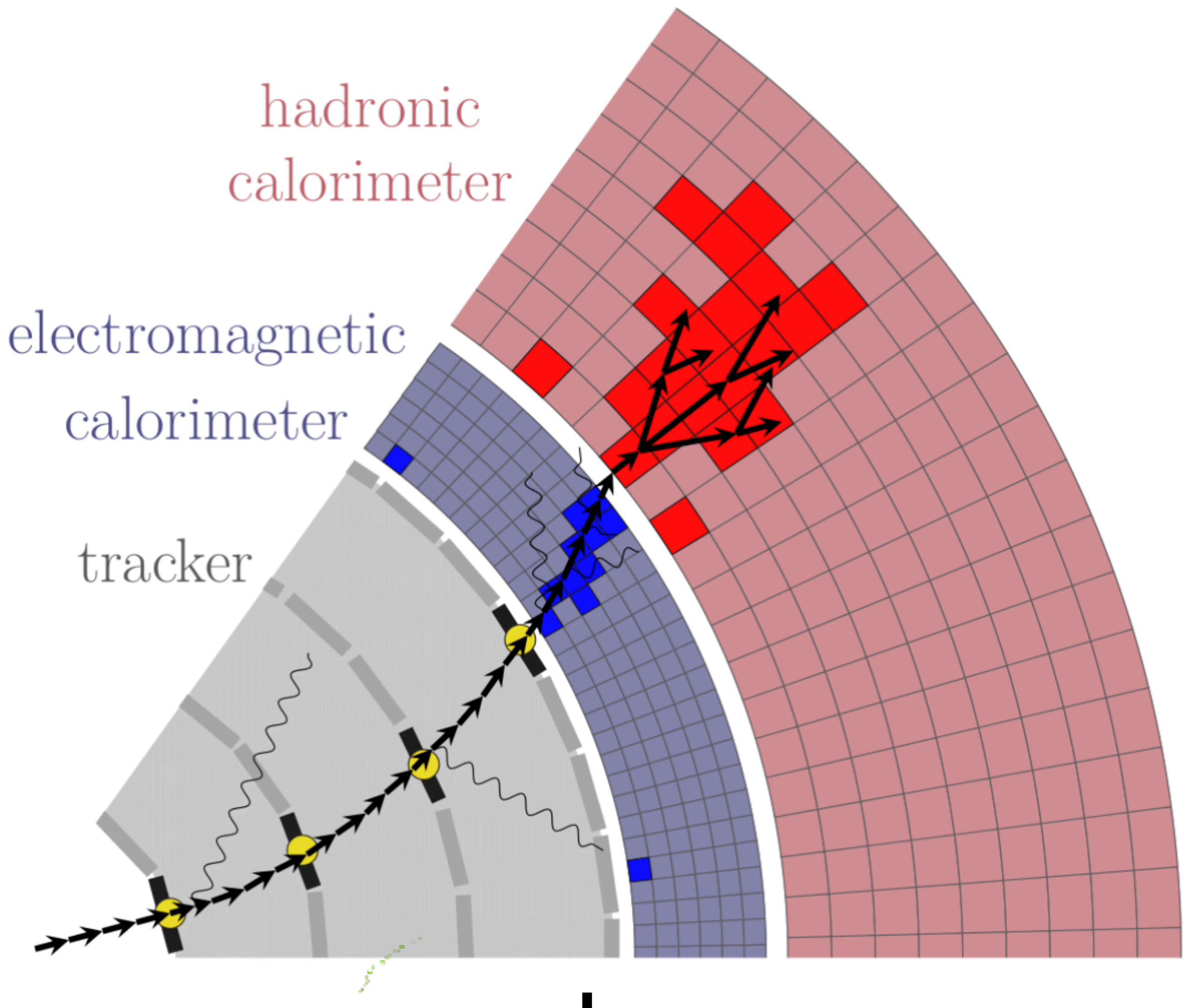
O(10)
O(10^3)
O(10^10)
10^−18m
10^−15m
10^−6m
100m
2T

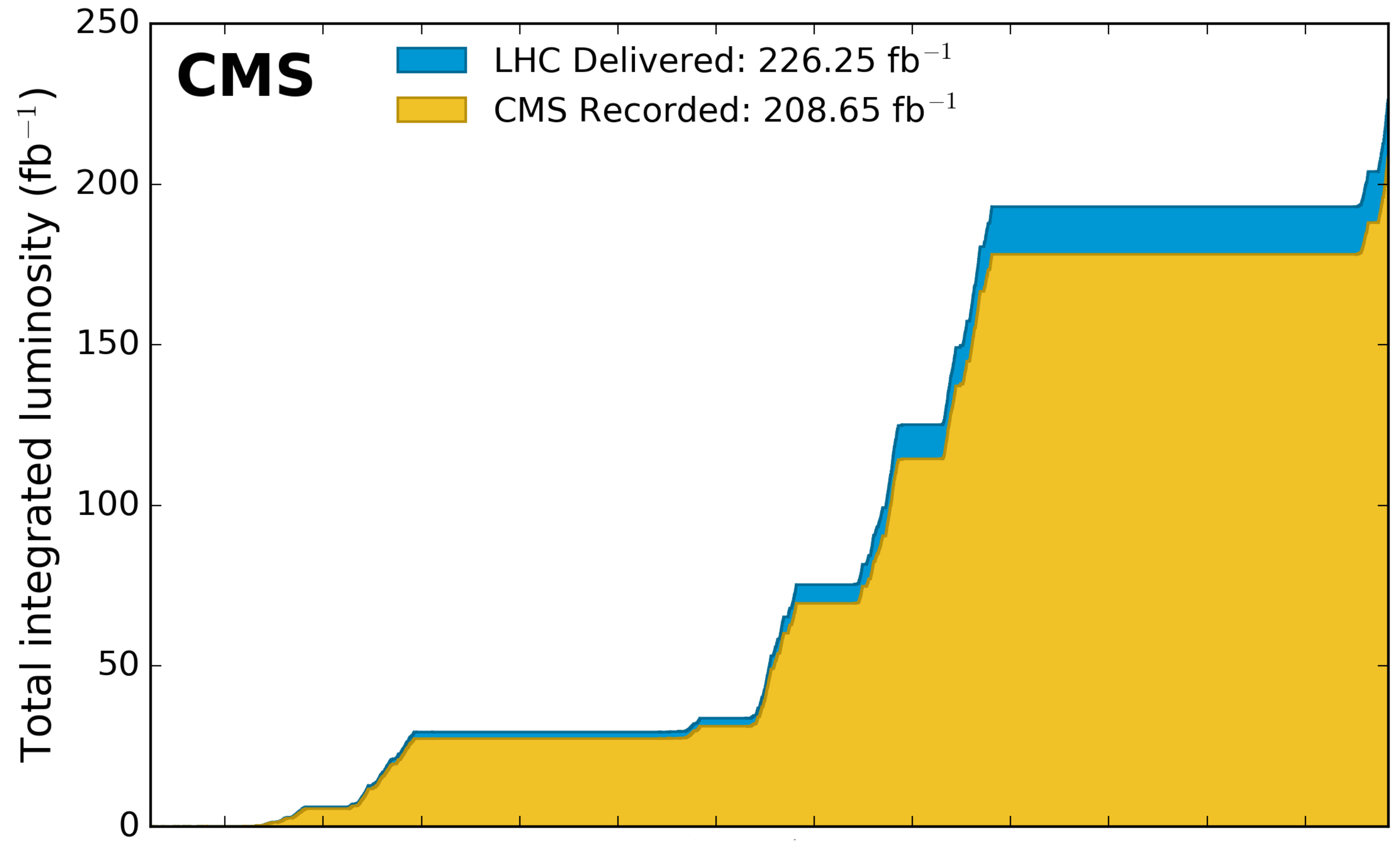
~40 quadrillion collisions recorded at LHC



O(1) trillion simulated events





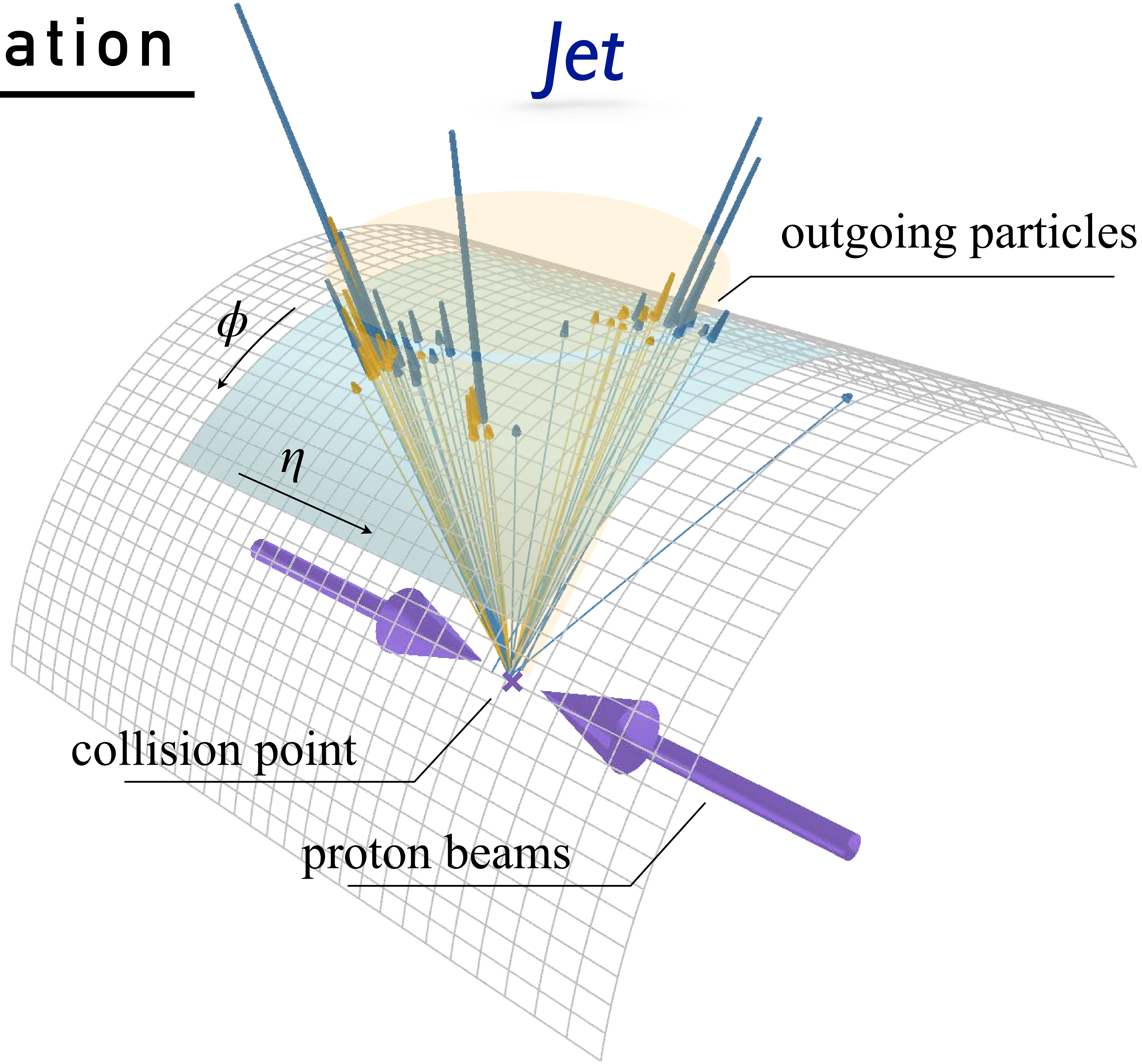


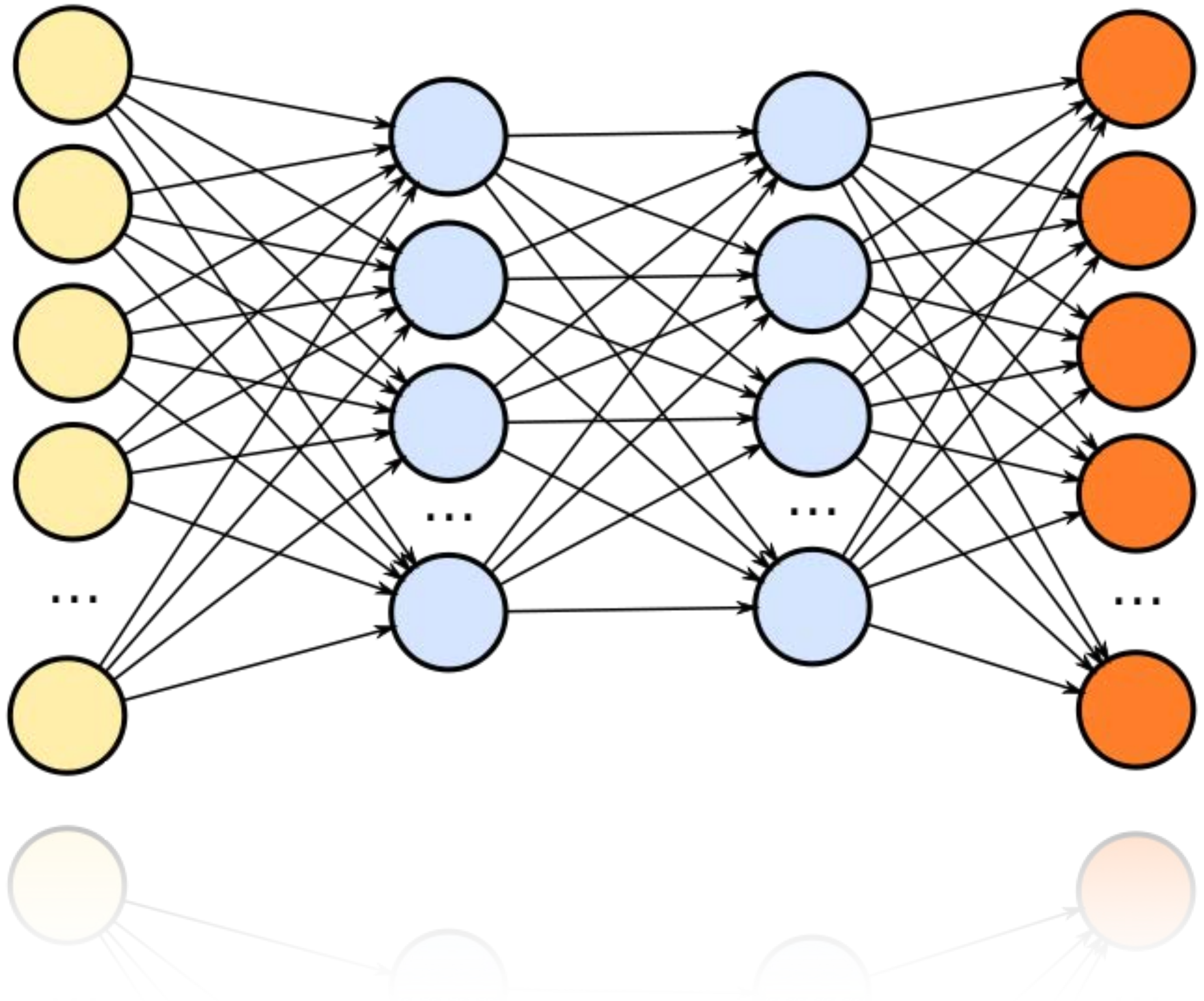
Fully supervised
- Requires truth labels
- Only possible using simulation

Weakly supervised
- Requires physics knowledge to define regions enriched in signal/background

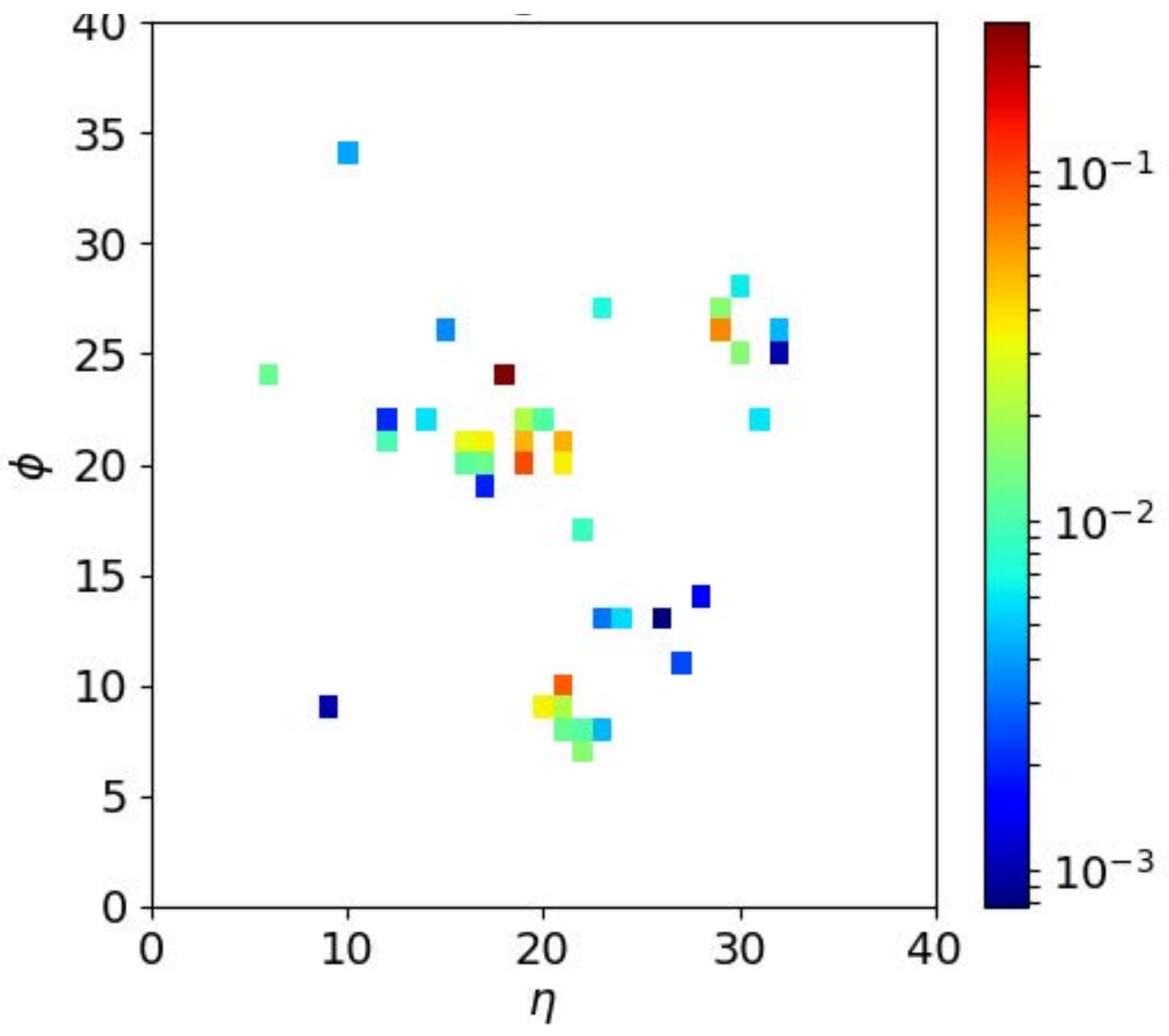
Unsupervised
- No signal prior, completely data driven

Data representation



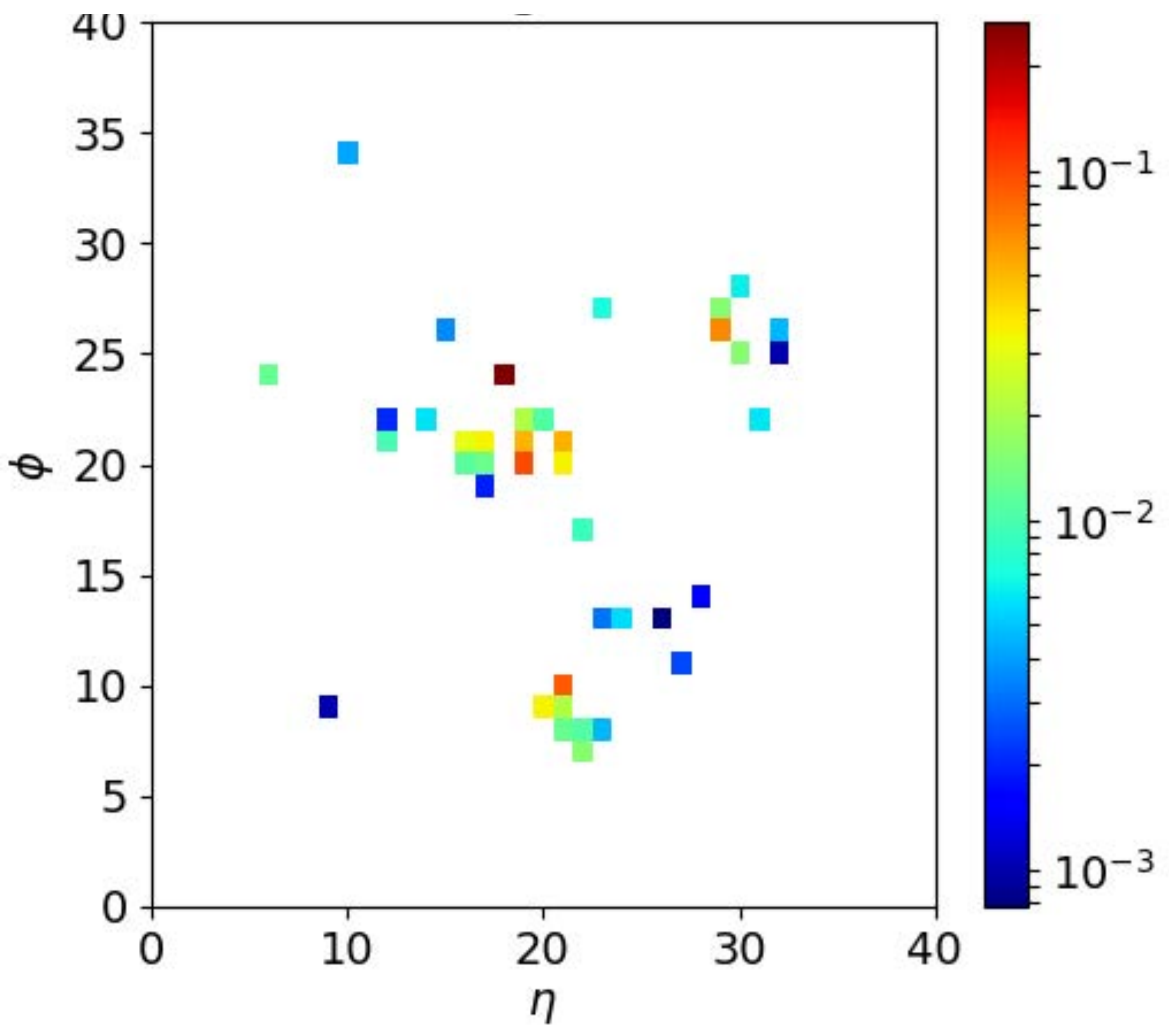


Image



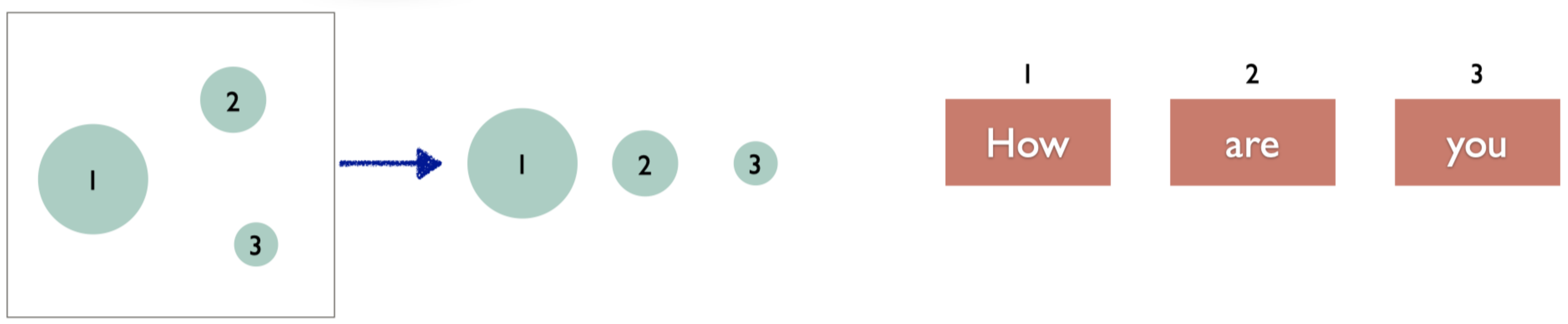
arXiv:1511.05190

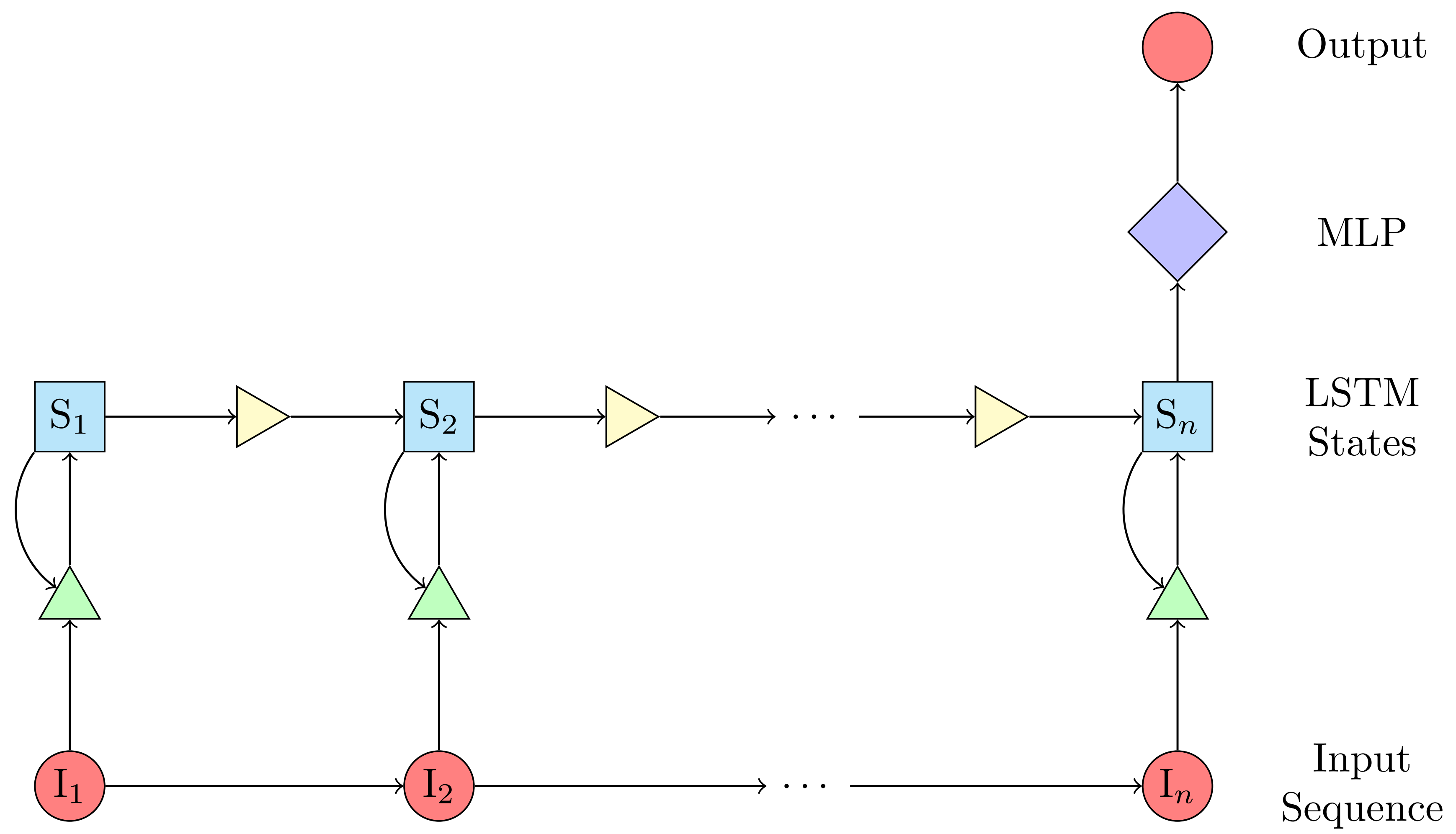
Image



arXiv:1511.05190

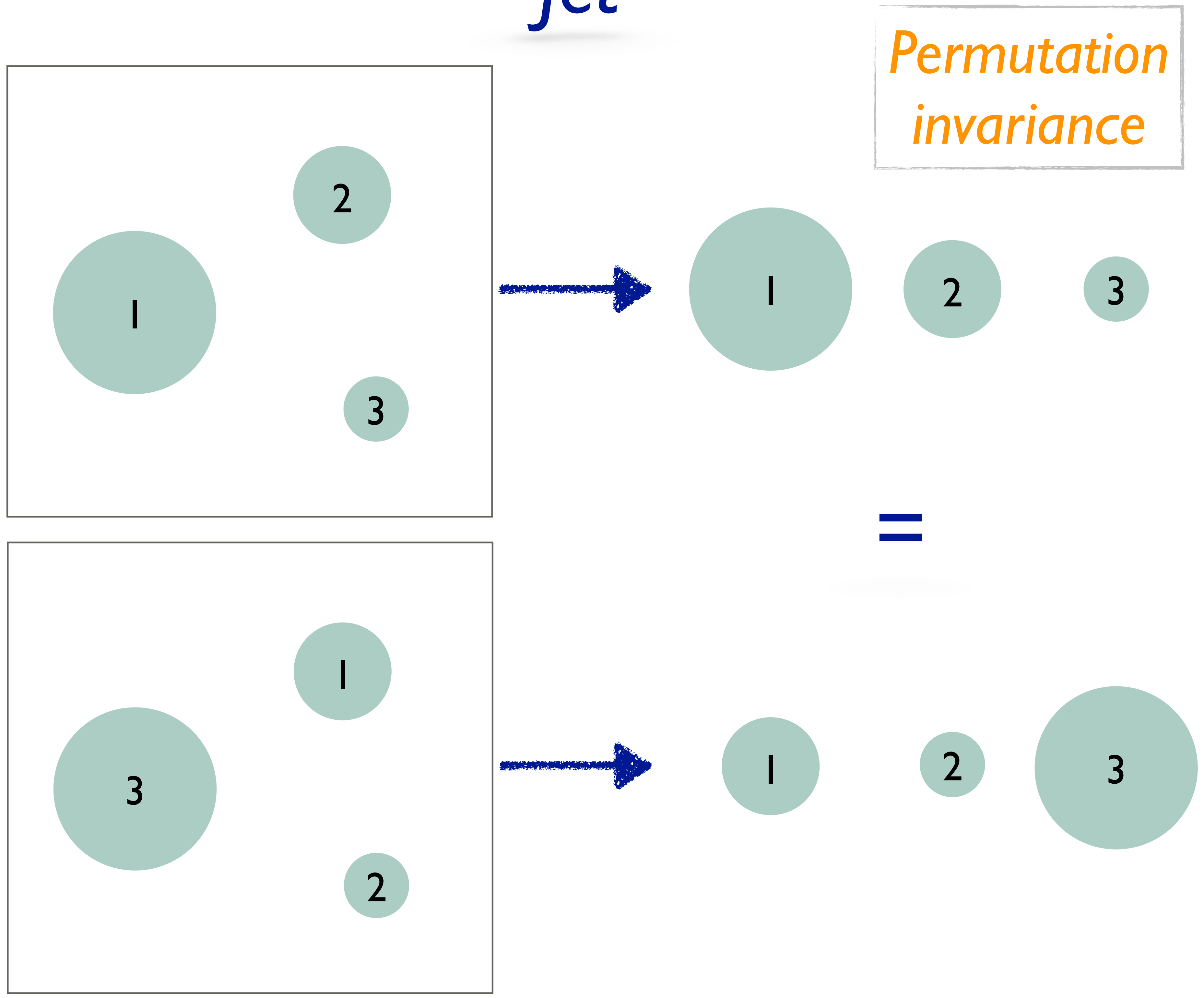
Sequence



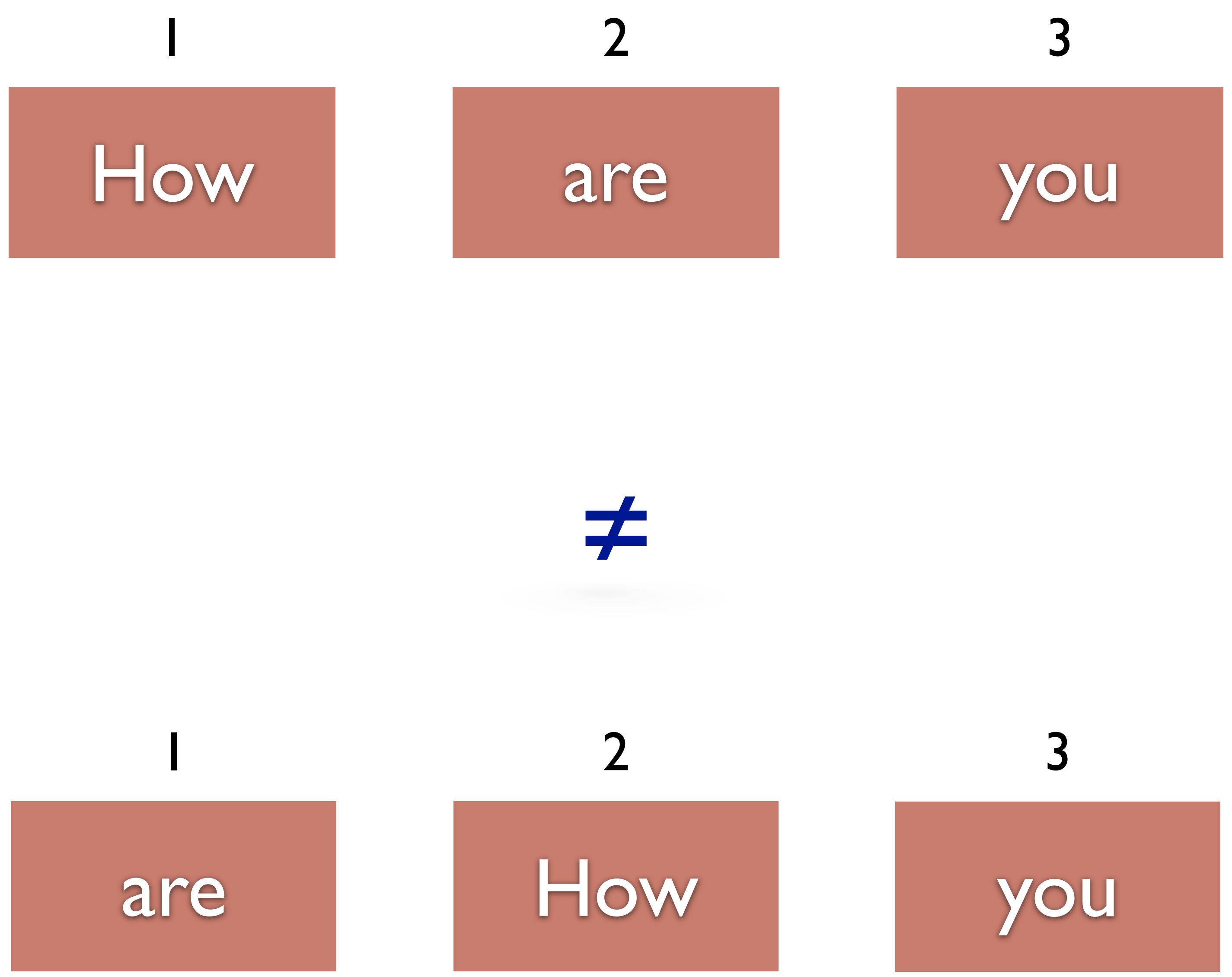


arXiv:1607.08633

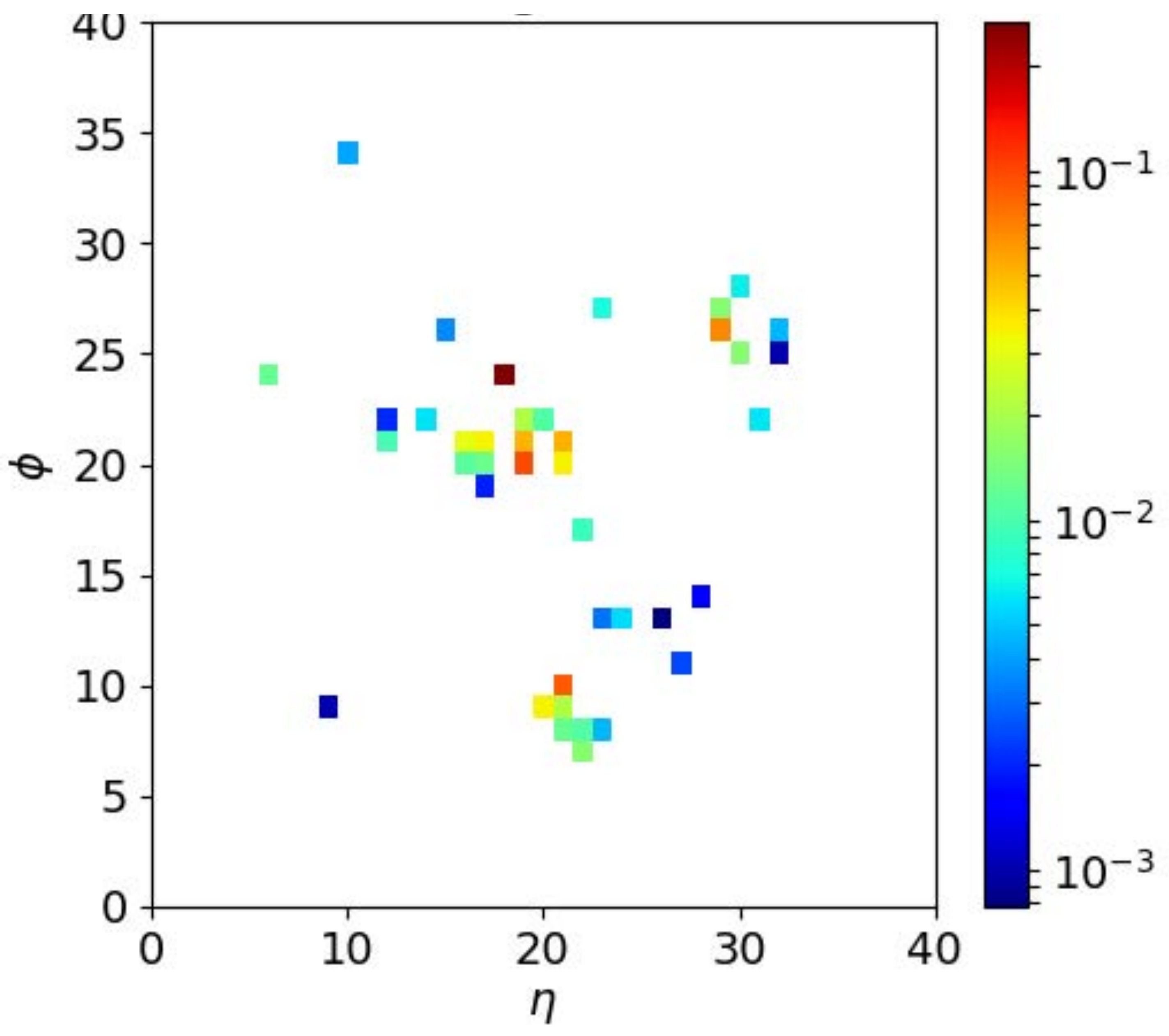
Jet



Sequence

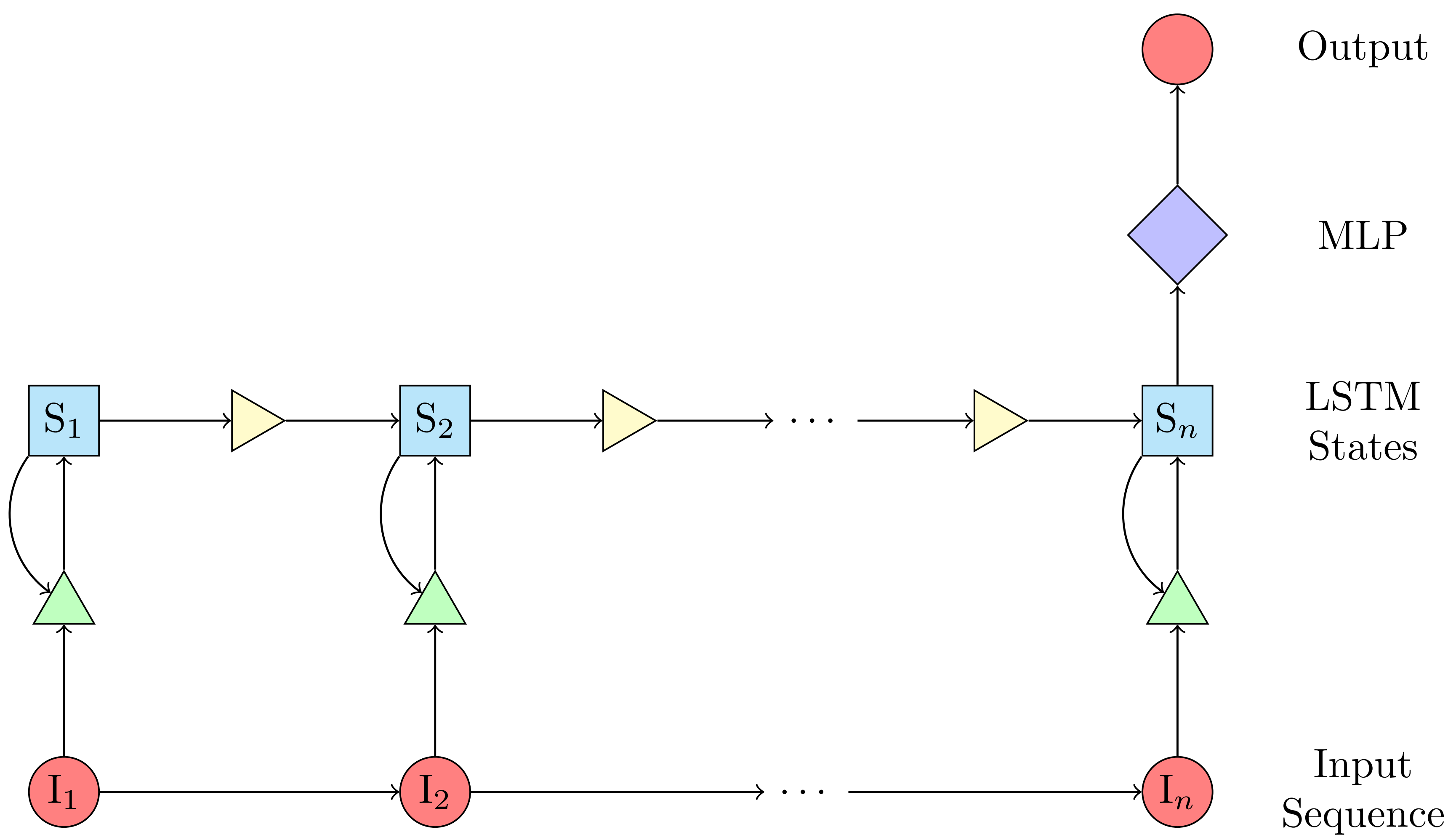


Image



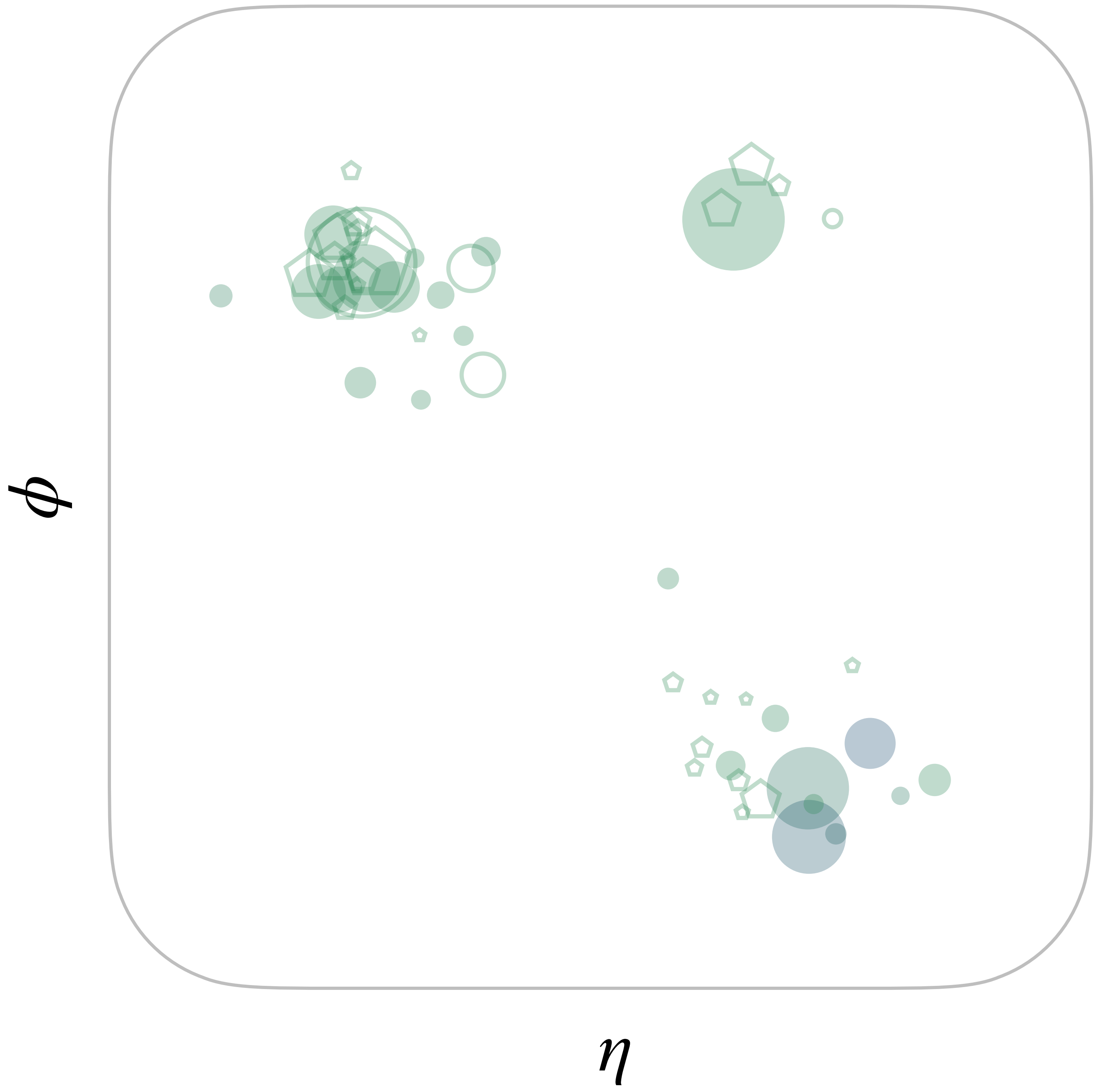
arXiv:1511.05190

Sequence

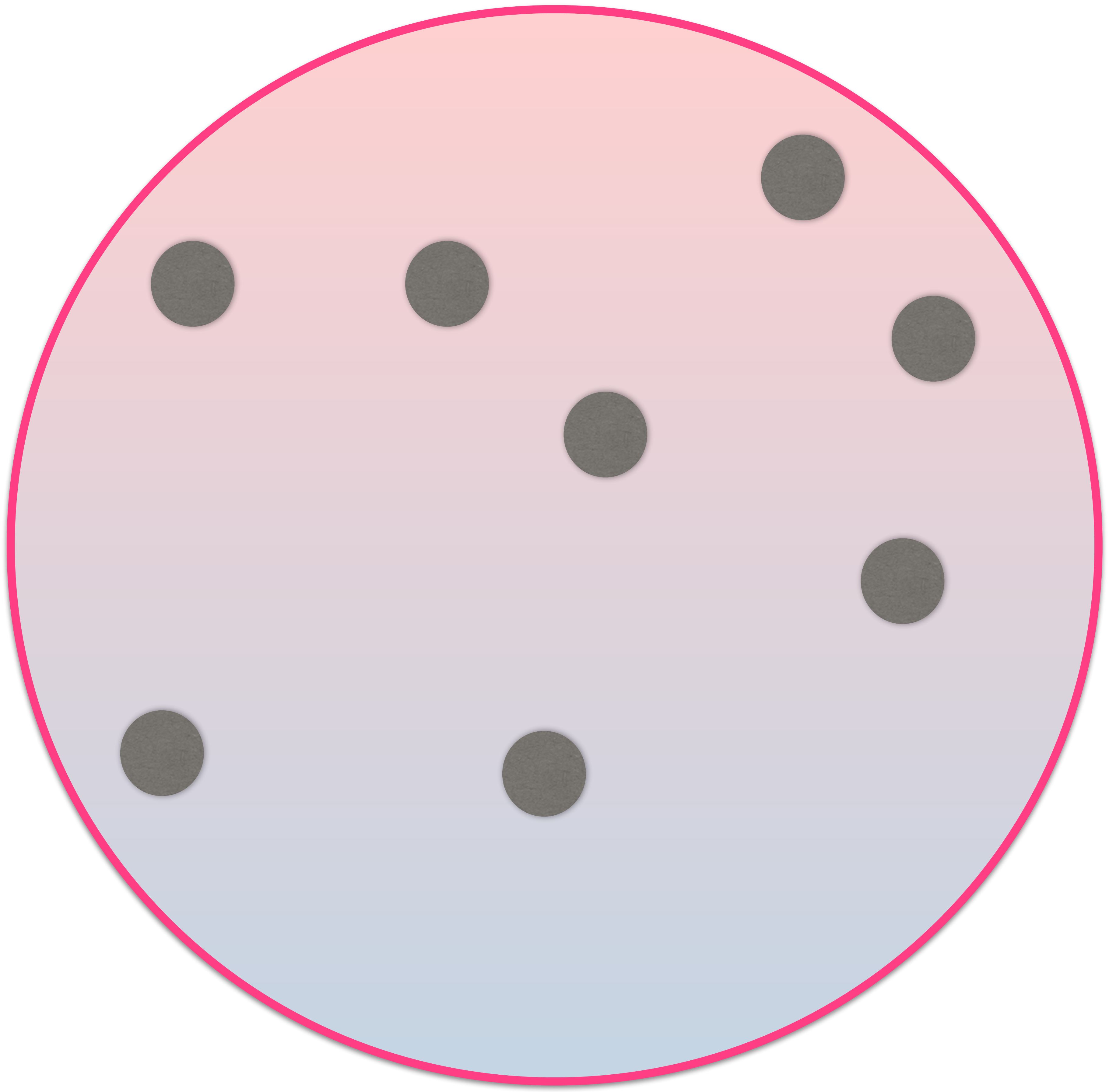


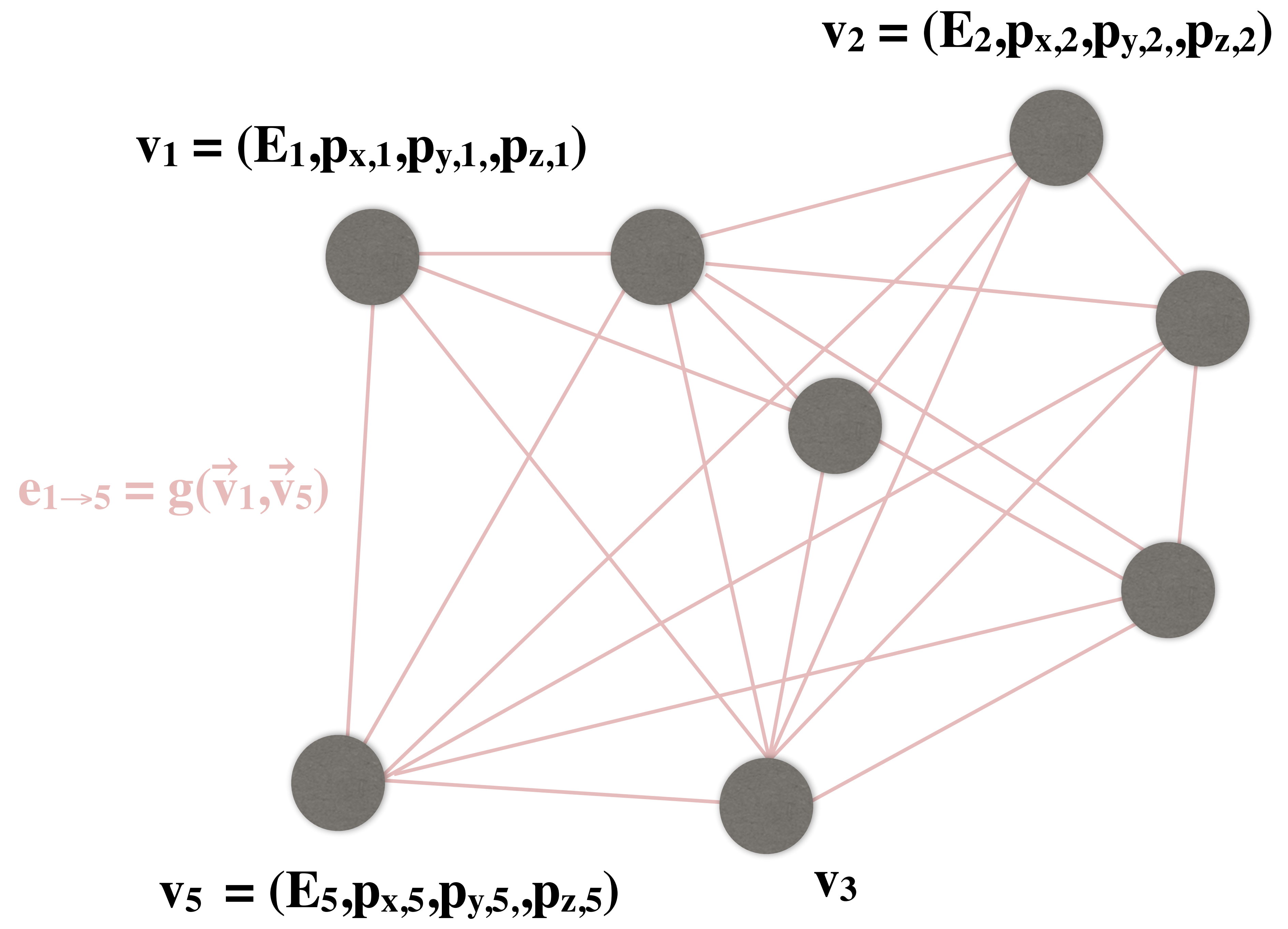
arXiv:1607.08633

Point Cloud



PRD:101.056019



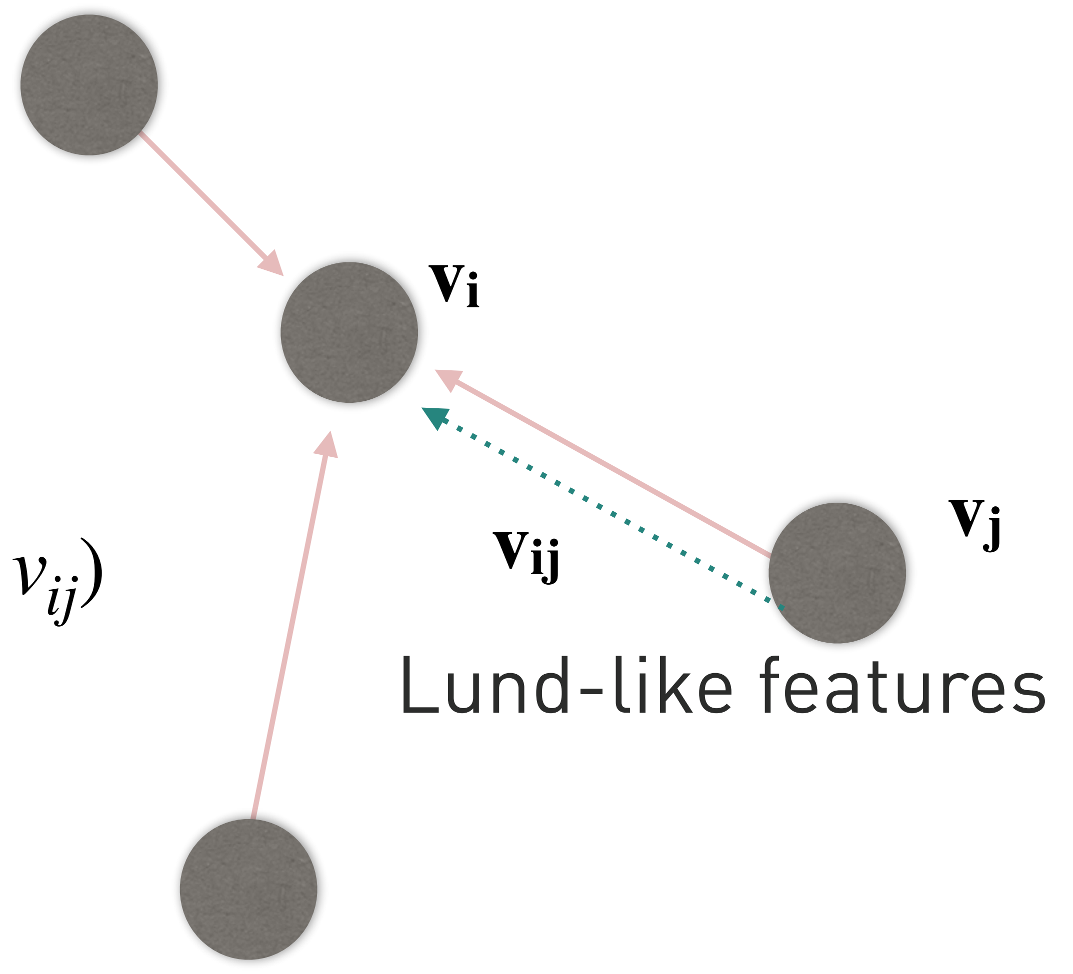


SOTA: Graph Neural Networks acting on point cloud data

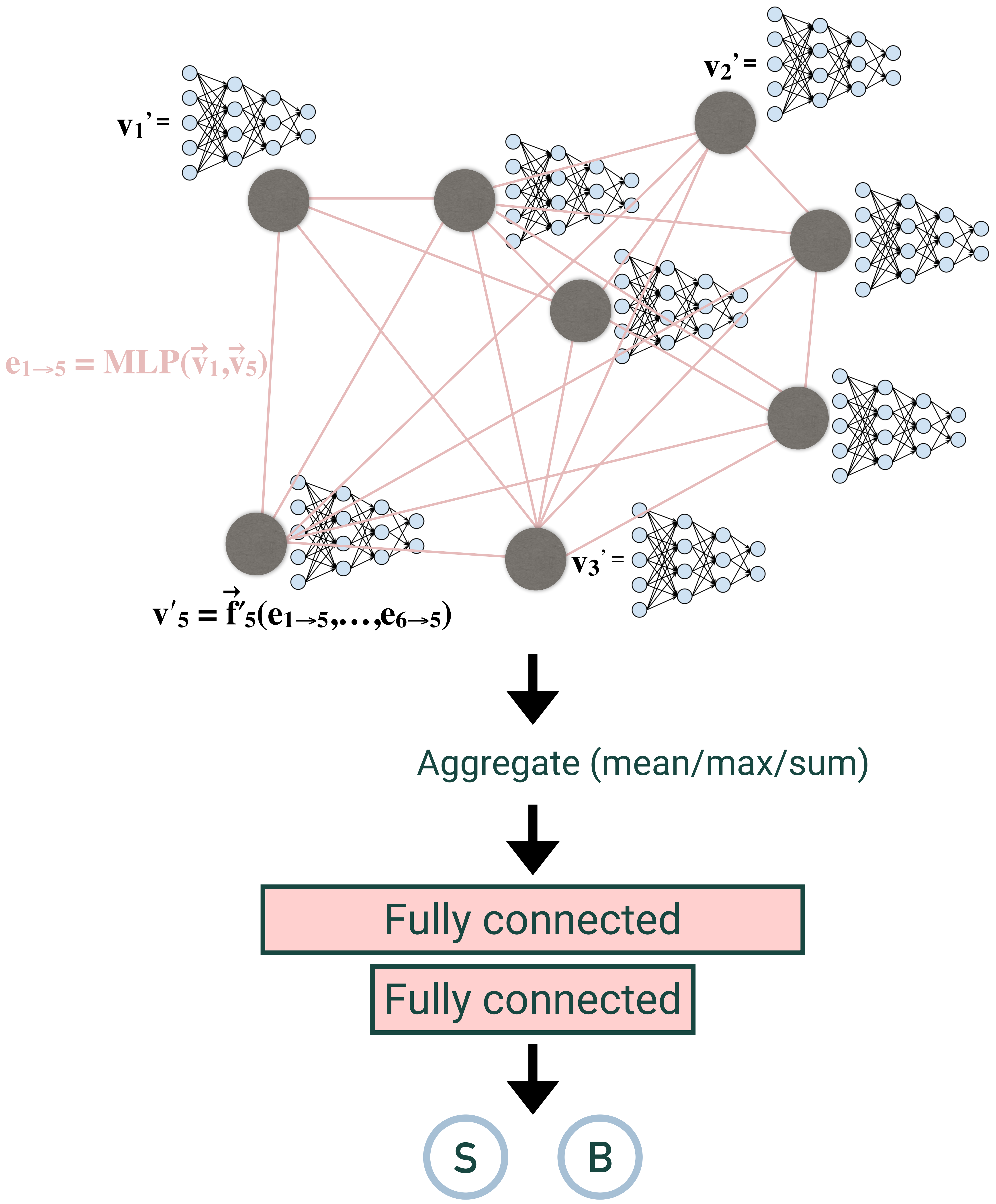
- ParticleNet (GNN on point cloud)
  LundNet (GNN,Lund plane)
  ABCNet (GNN, attention)
  Point Cloud Transformers (transformer, attention)
  ParticleNeXt (GNN, attention, Lund)
  ParT (transformer, attention)

SOTA: Graph Neural Networks acting on point cloud data

- ParticleNet (GNN on point cloud)
  LundNet (GNN,Lund plane)
  ABCNet (GNN, attention)
  Point Cloud Transformers (transformer, attention)
  ParticleNeXt (GNN, attention, Lund)
  ParT (transformer, attention)

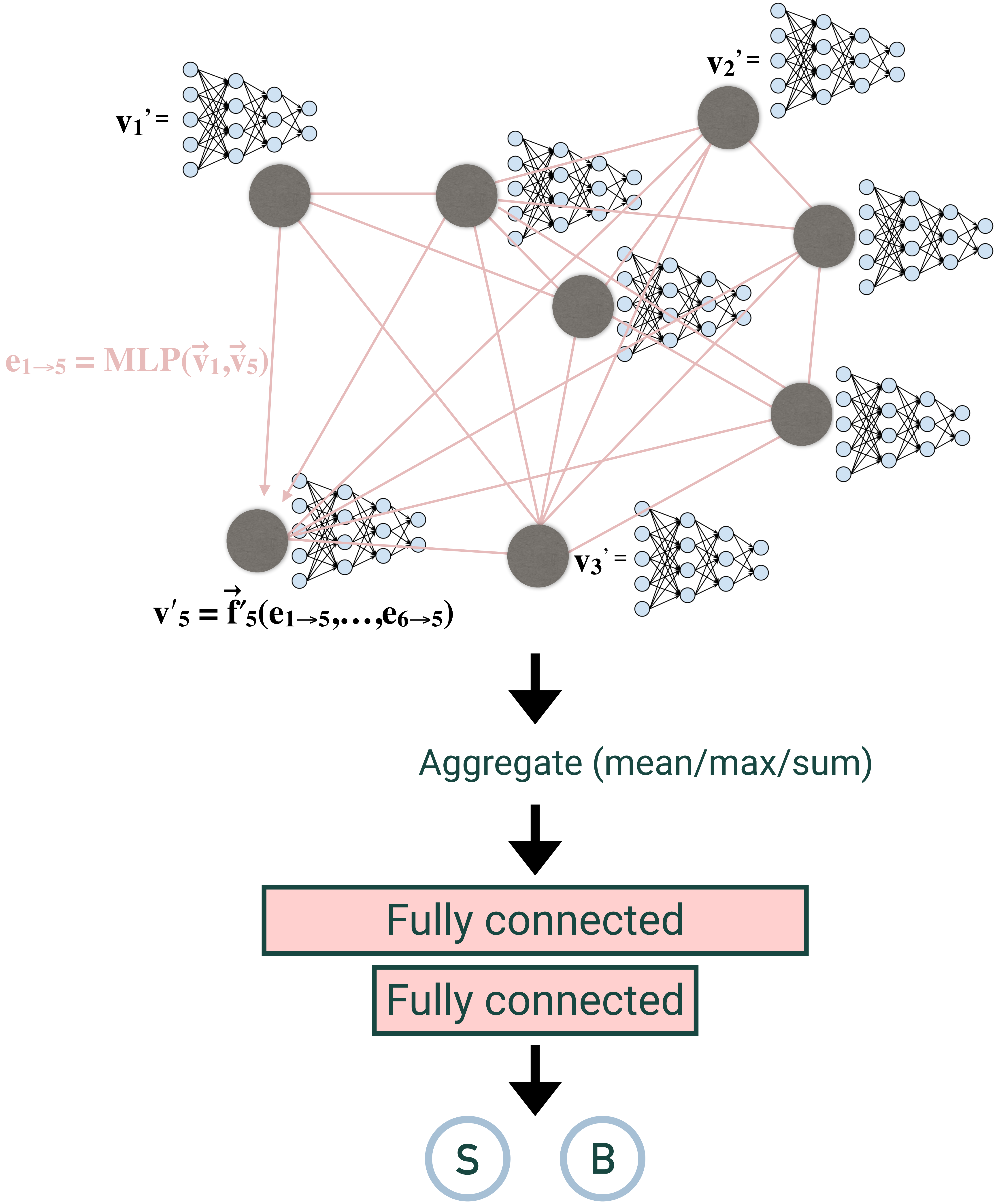


$e_{ij} = MLP(v_i, v_j, v_{ij})$



SOTA: Graph Neural Networks acting on point cloud data

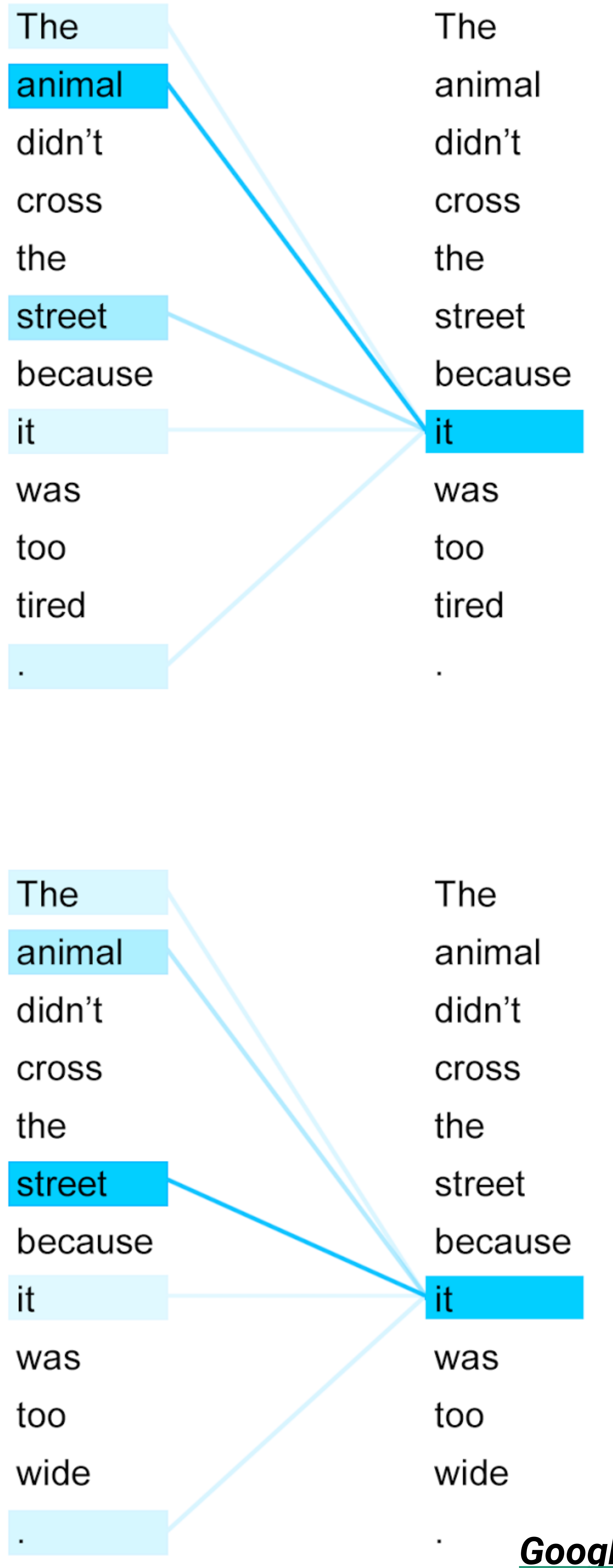
- ParticleNet (GNN on point cloud)
  LundNet (GNN,Lund plane)
  ABCNet (GNN, attention)
  Point Cloud Transformers (transformer, attention)
  ParticleNeXt (GNN, attention, Lund)
  ParT (transformer, attention)



Transformers and (self-)attention

(Self-)Attention

- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores

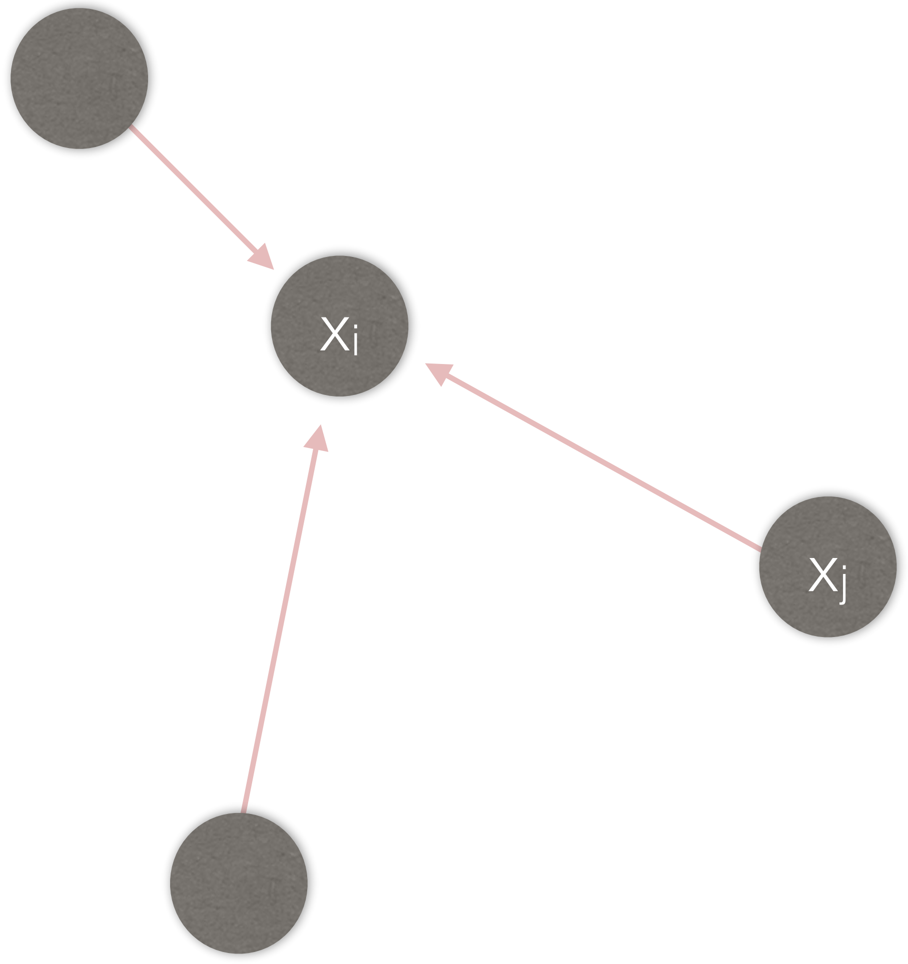


Google AI blog

Transformers and (self-)attention

(Self-)Attention

- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores



Weighted sum over all input vectors:

$$\boldsymbol{y}_i = \sum_j w_{ij} \boldsymbol{x}_j$$

Weight (how related inputs are):

$$w'_{ij} = \boldsymbol{x}_i{}^{\mathsf{T}} \boldsymbol{x}_j$$

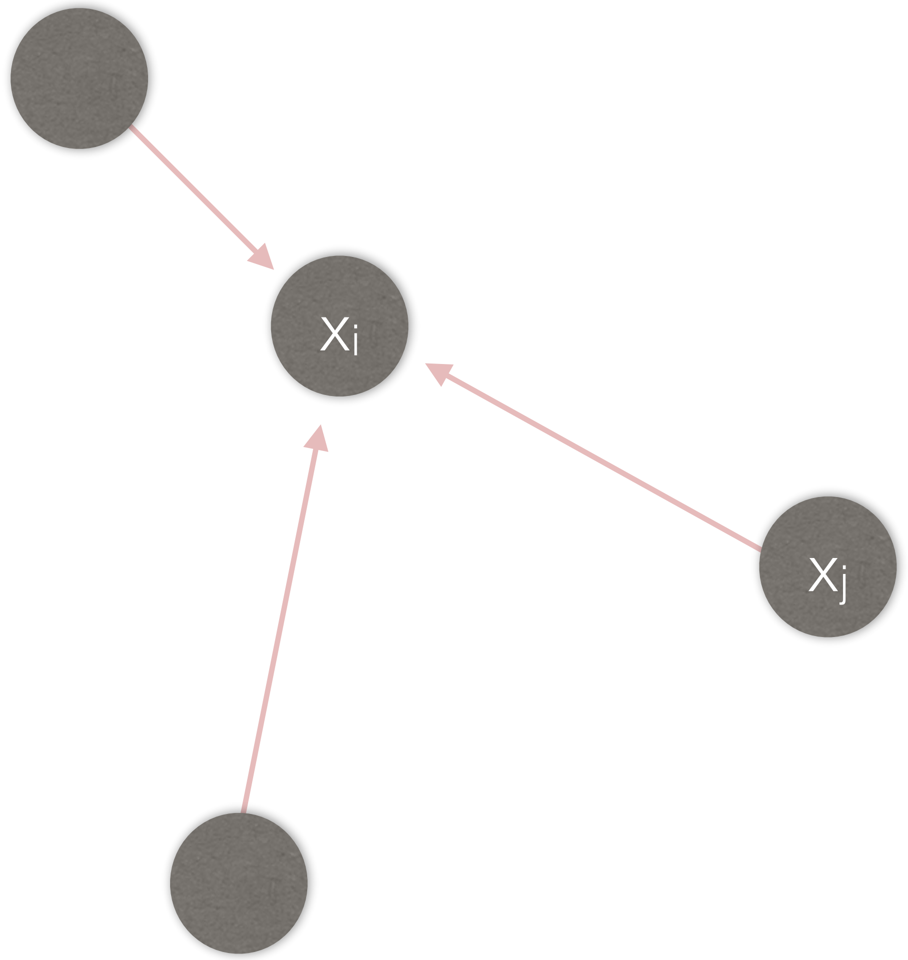
Map to [0,1]:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

Transformers and (self-)attention

(Self-)Attention

- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores



Weighted sum over all input vectors:

$$\boldsymbol{y}_i = \sum_j w_{ij} \boldsymbol{x}_j$$

Weight (how related inputs are):

$x_j \rightarrow \mathrm{MLP}(x_J)$
$x_i \rightarrow \mathrm{MLP}(x_i)$

$$w'_{ij} = \boldsymbol{x}_i{}^{\mathsf{T}} \boldsymbol{x}_j$$

Map to [0,1]:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

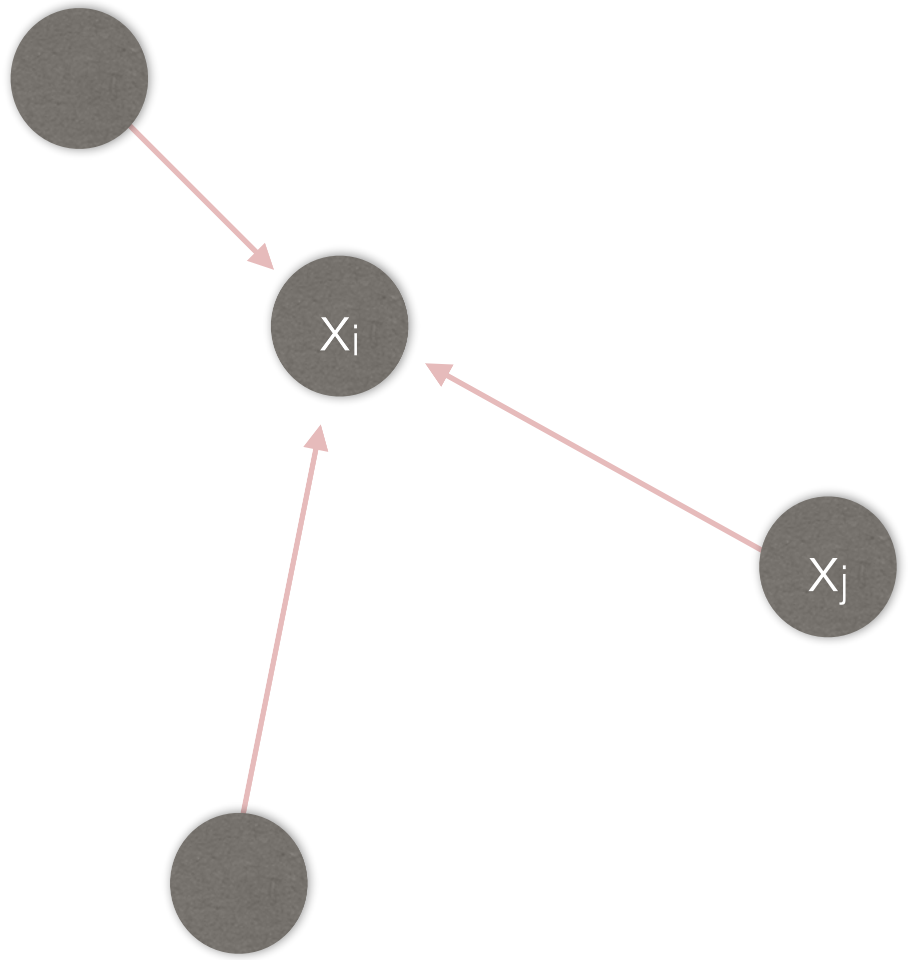
Transformers and (self-)attention

(Self-)Attention

- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores

Attention weights: weighted importance between each pair of particles

- Determine relationship between all particles of point cloud
- Jet features become parameters of the model
- Several attention layers → different important features (multi-head attention)



Weighted sum over all input vectors:

$$y_i = \sum_j w_{ij} x_j$$

Weight (how related inputs are):

$$w'_{ij} = x_i^{\mathsf{T}} x_j$$

$x_j \rightarrow \mathrm{MLP}(x_J)$
$x_i \rightarrow \mathrm{MLP}(x_i)$

Map to [0,1]:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

Transformers and (self-)attention

(Self-)Attention

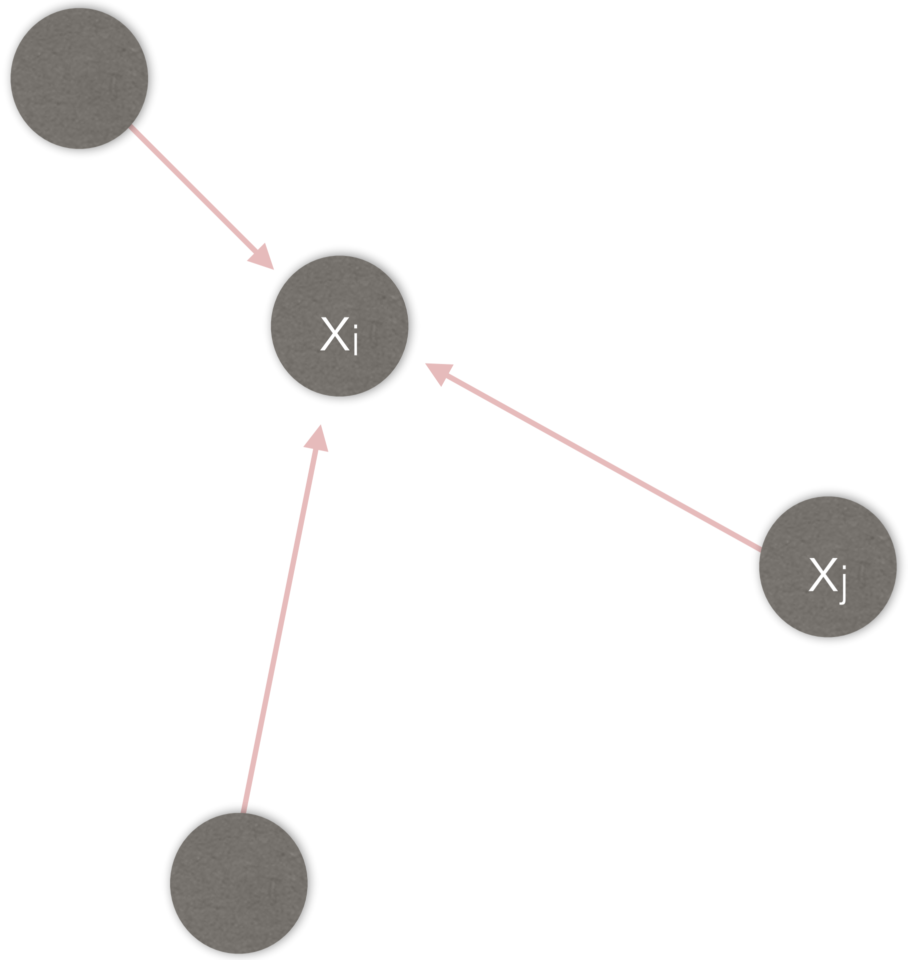
- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores

Attention weights: weighted importance between each pair of particles

- Determine relationship between all particles of point cloud
- Jet features become parameters of the model
- Several attention layers → different important features (multi-head attention)

Transformer:

- Only set of interaction between units is self-attention!



Weighted sum over all input vectors:

$$y_i = \sum_j w_{ij} x_j$$

Weight (how related inputs are):

$$w'_{ij} = x_i^T x_j$$

$x_j \rightarrow \mathrm{MLP}(x_J)$
$x_i \rightarrow \mathrm{MLP}(x_i)$

Map to [0,1]:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

Transformers and (self-)attention

(Self-)Attention

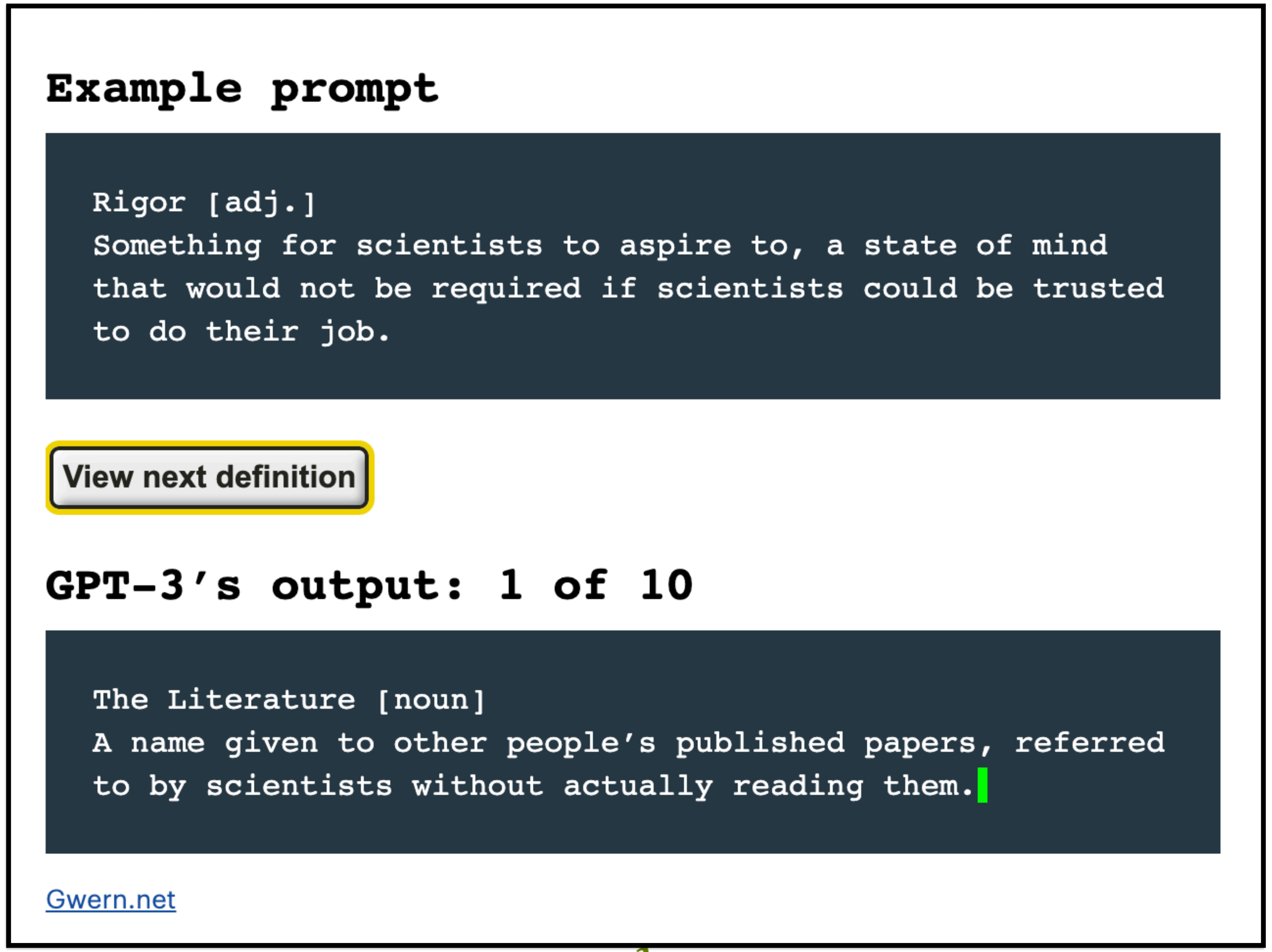
- Allows inputs to interact with each other ("self") and find out who they should pay more attention to ("attention").
- Outputs: aggregates of interactions and attention scores

Attention weights: weighted importance between each pair of particles

- Determine relationship between all particles of point cloud
- Jet features become parameters of the model
- Several attention layers → different important features (multi-head attention)

Transformer:

- Only set of interaction between units is self-attention!



Weight (how related inputs are):

$x_j \rightarrow MLP(x_J)$
$x_i \rightarrow MLP(x_i)$

$$w'_{ij} = x_i^{T} x_j$$

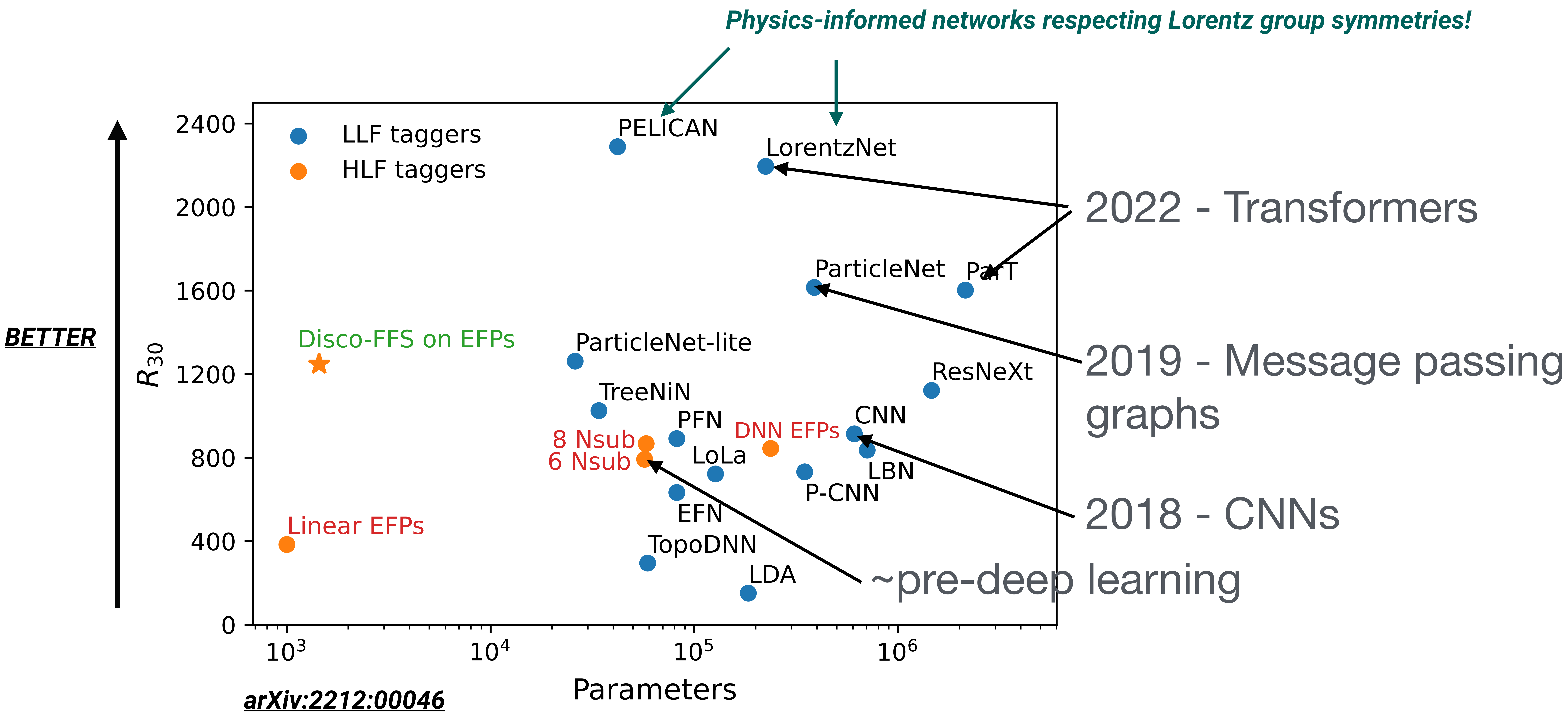
Map to [0,1]:

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

ABCNet:

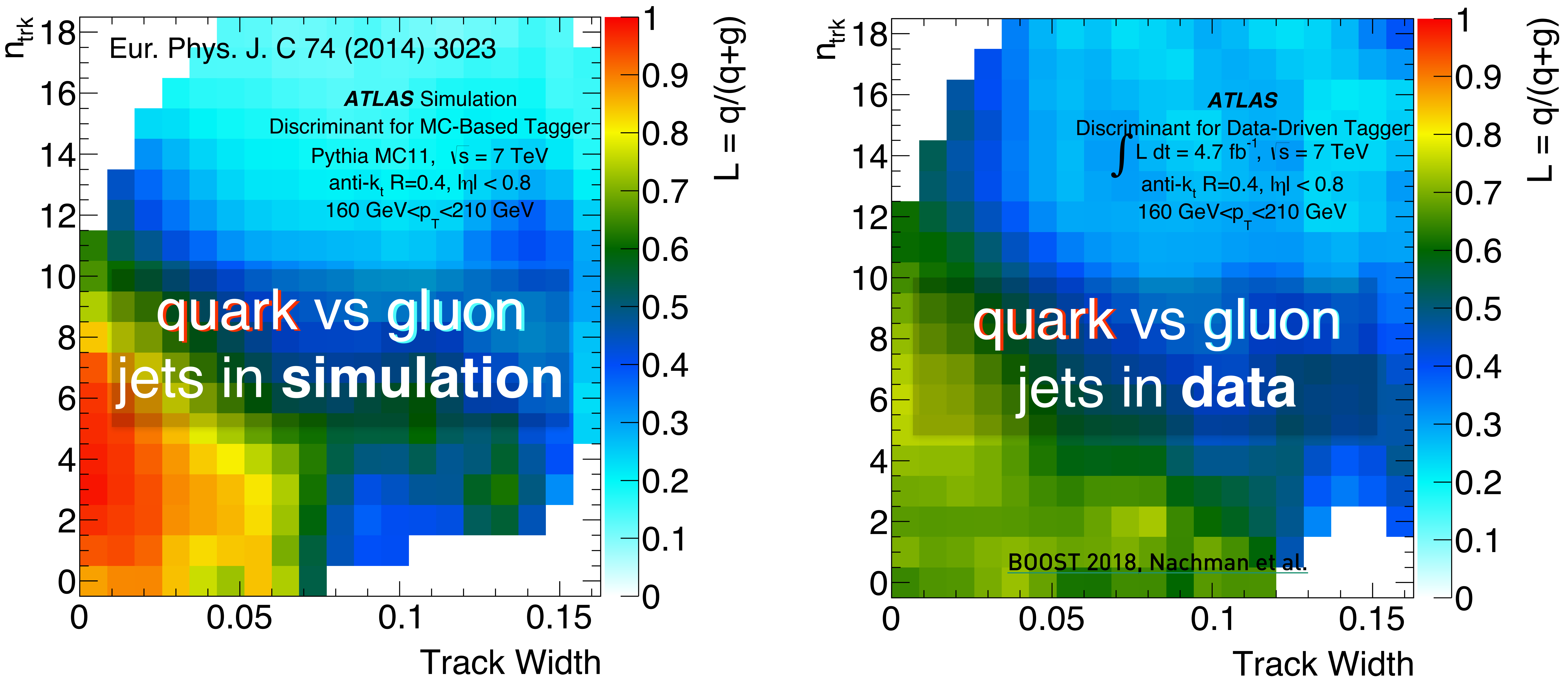
Pixel intensity = particle importance w.r.t most energetic particle in jet, from attention weights

No substructure information given, learned through attention layers!



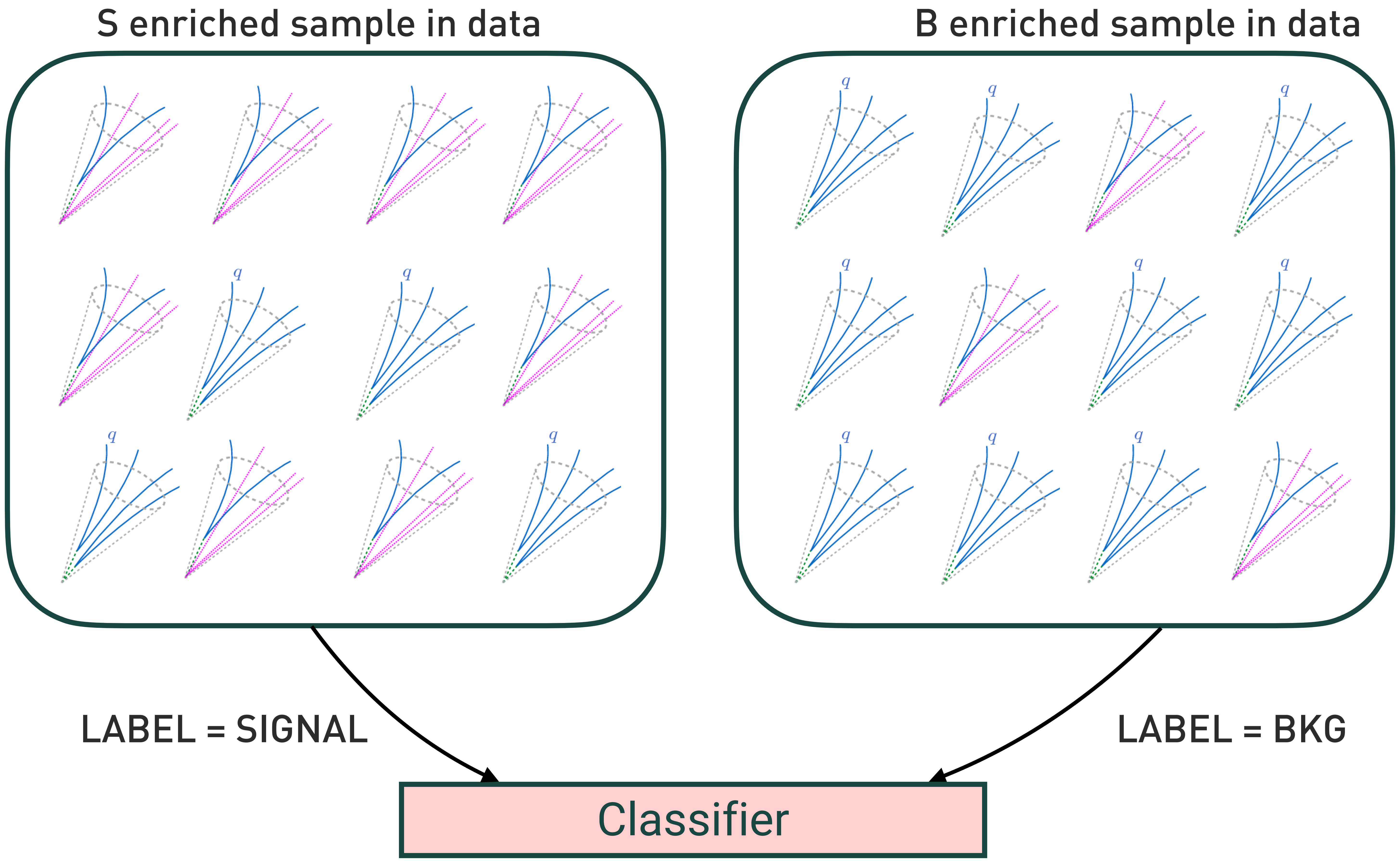
Physics-informed networks respecting Lorentz group symmetries!
BETTER
R_{30}
LLF taggers
HLF taggers
PELICAN
LorentzNet
ParticleNet
ParT
Disco-FFS on EFPs
ParticleNet-lite
ResNeXt
TreeNiN
CNN
PFN
DNN EFPs
8 Nsub
6 Nsub
LoLa
LBN
P-CNN
EFN
Linear EFPs
TopoDNN
LDA
Parameters
2022 - Transformers
2019 - Message passing graphs
2018 - CNNs
~pre-deep learning
arXiv:2212:00046

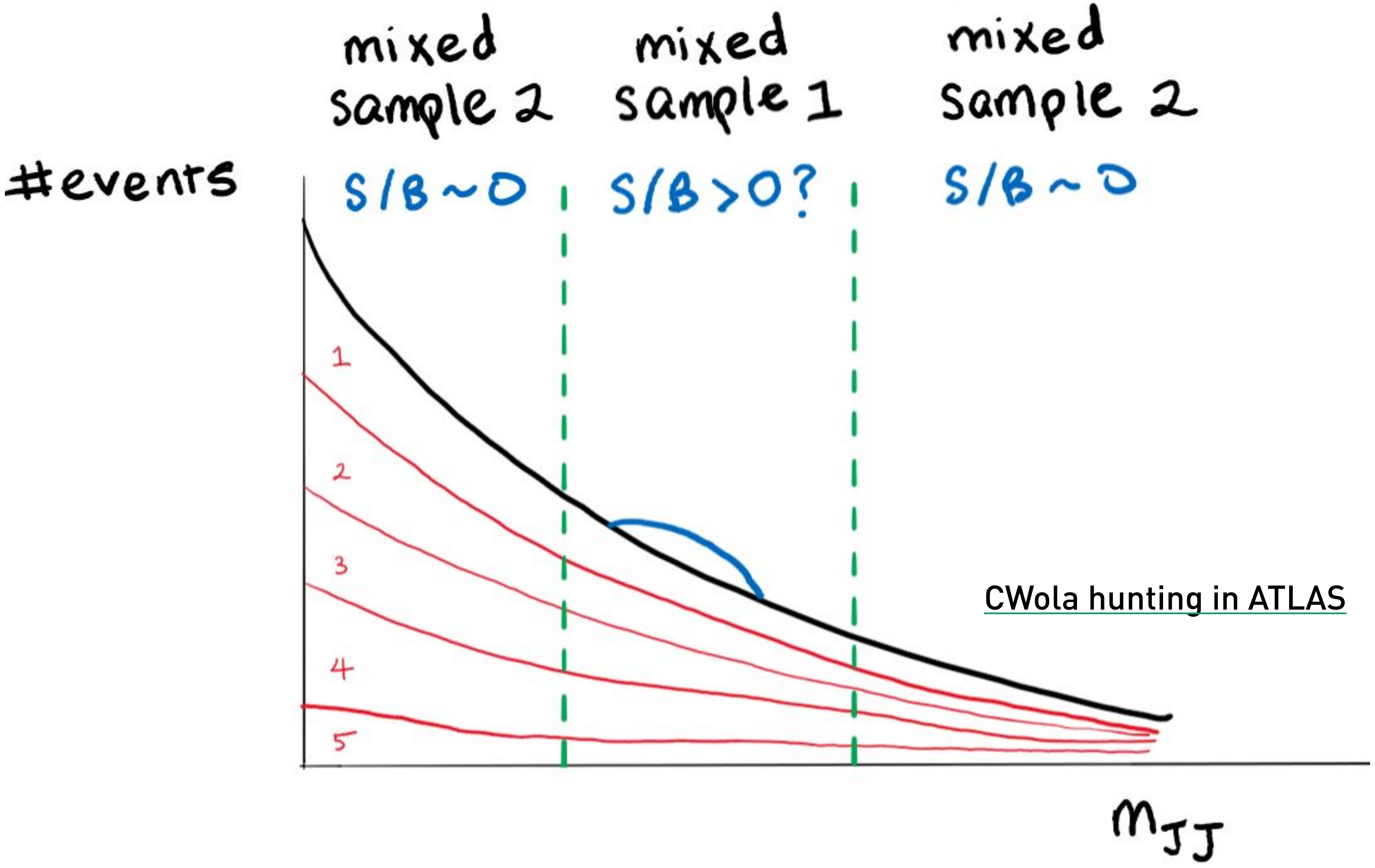
Train on simulation, test on data



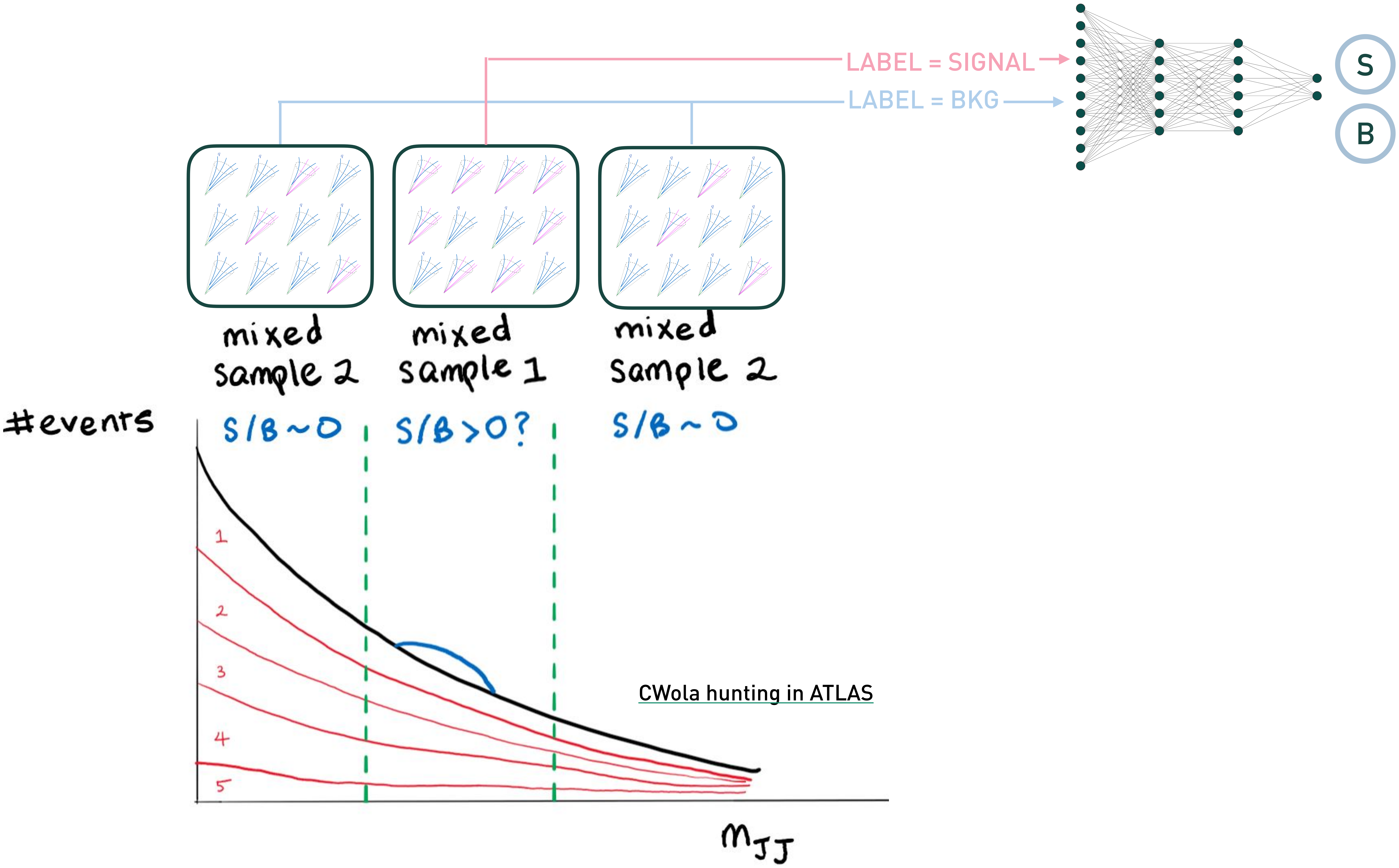
If data and simulation differ, this is sub-optimal!

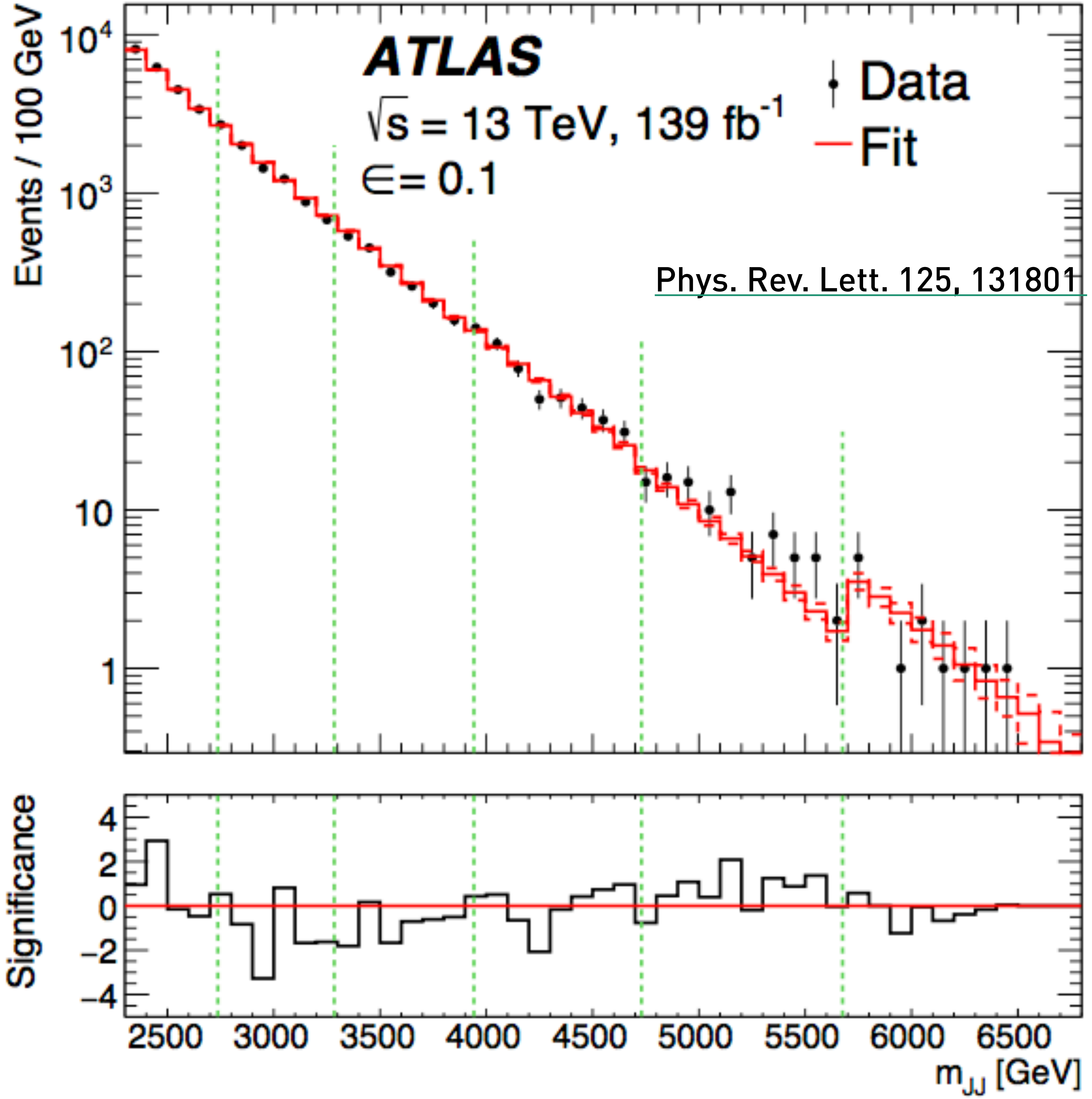
Semisupervised: Classification without Labels





[CWola hunting in ATLAS]

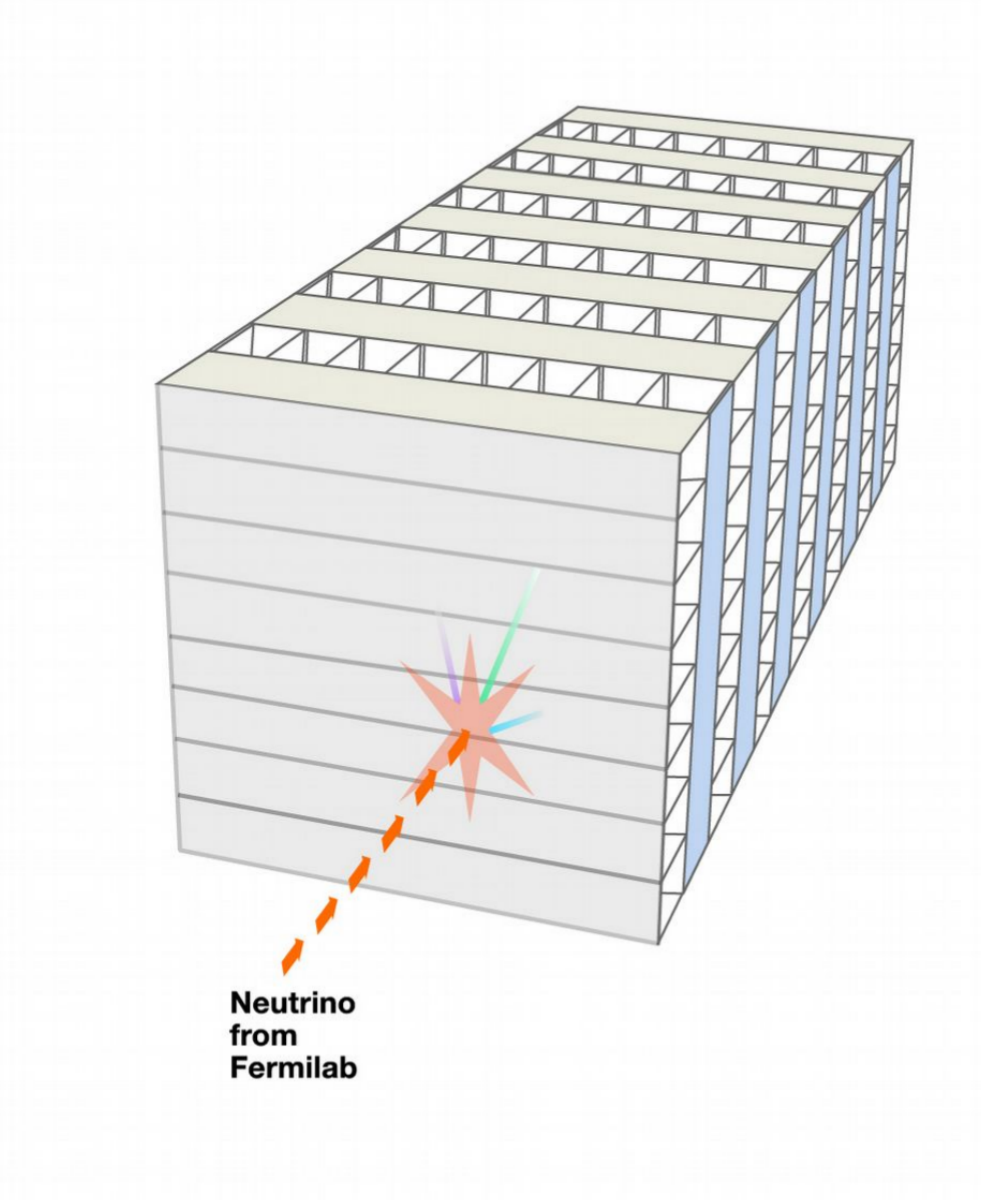

LABEL = SIGNAL
LABEL = BKG
S
B
mixed sample 2
mixed sample 1
mixed sample 2
#events
S/B ~ 0
S/B > 0?
S/B ~ 0
1
2
3
4
5
m_{JJ}
CWola hunting in ATLAS

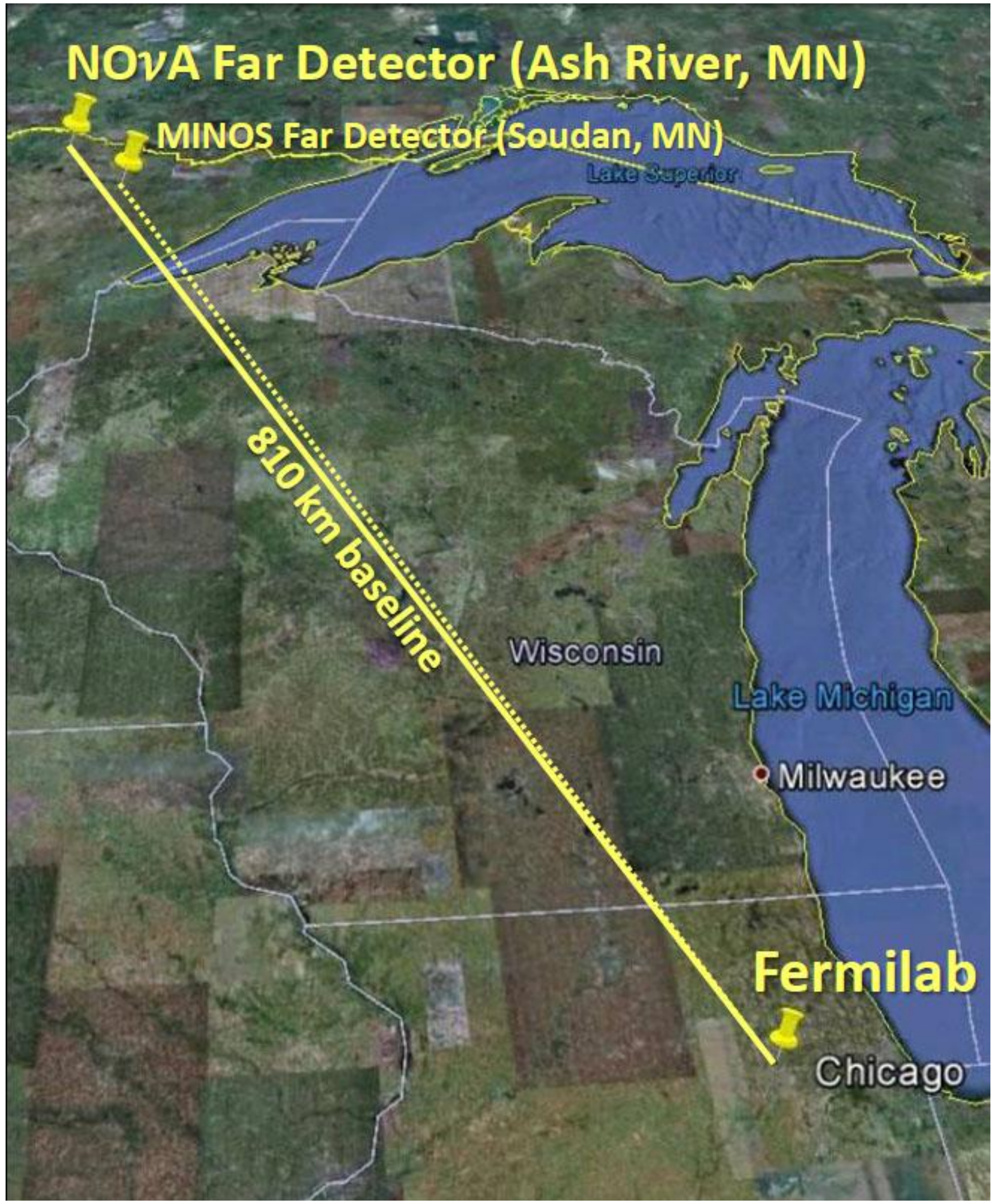


Phys. Rev. Lett. 125, 131801

Hybrid approaches - NoVa

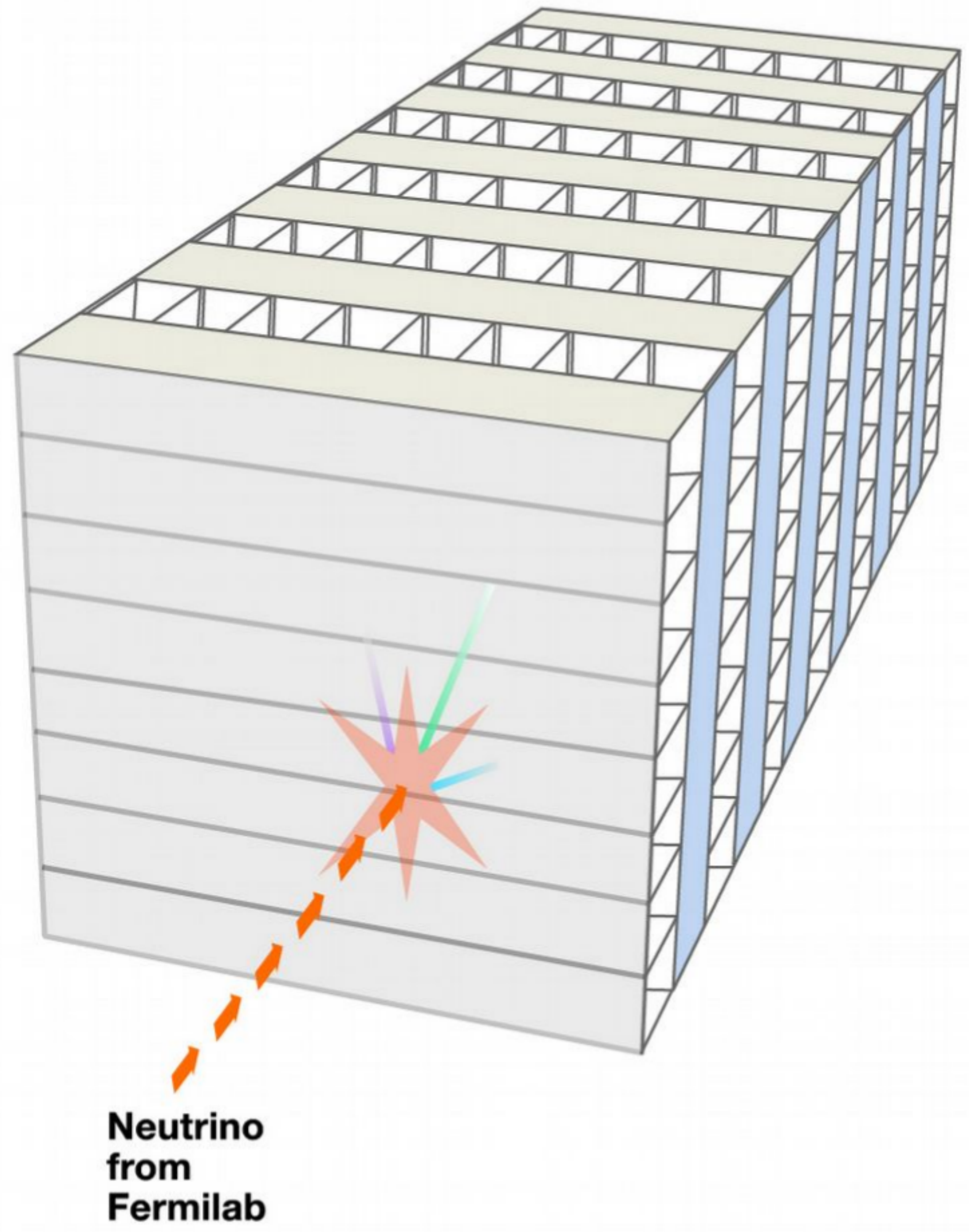
Aurisano et al
K. Sachdev

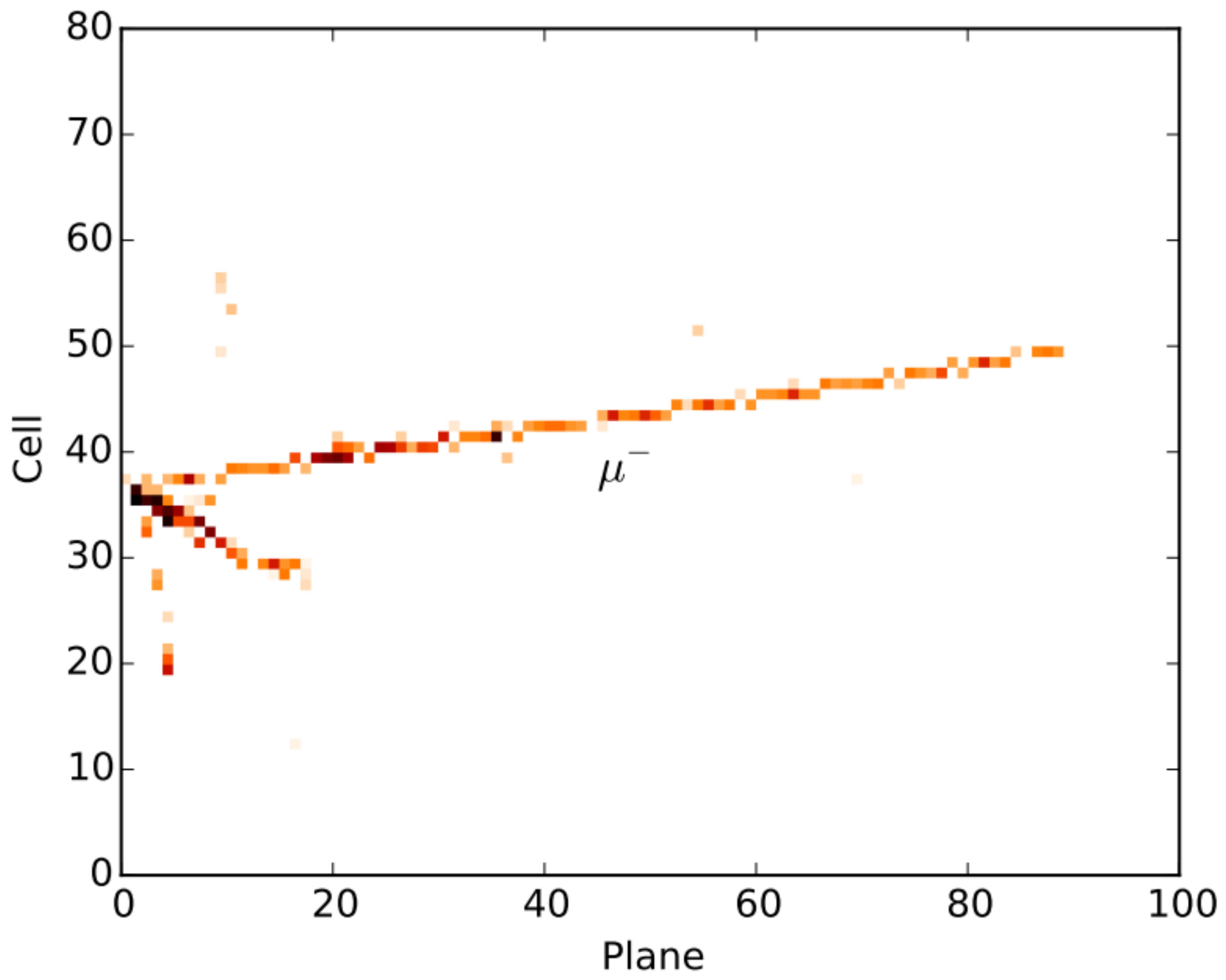


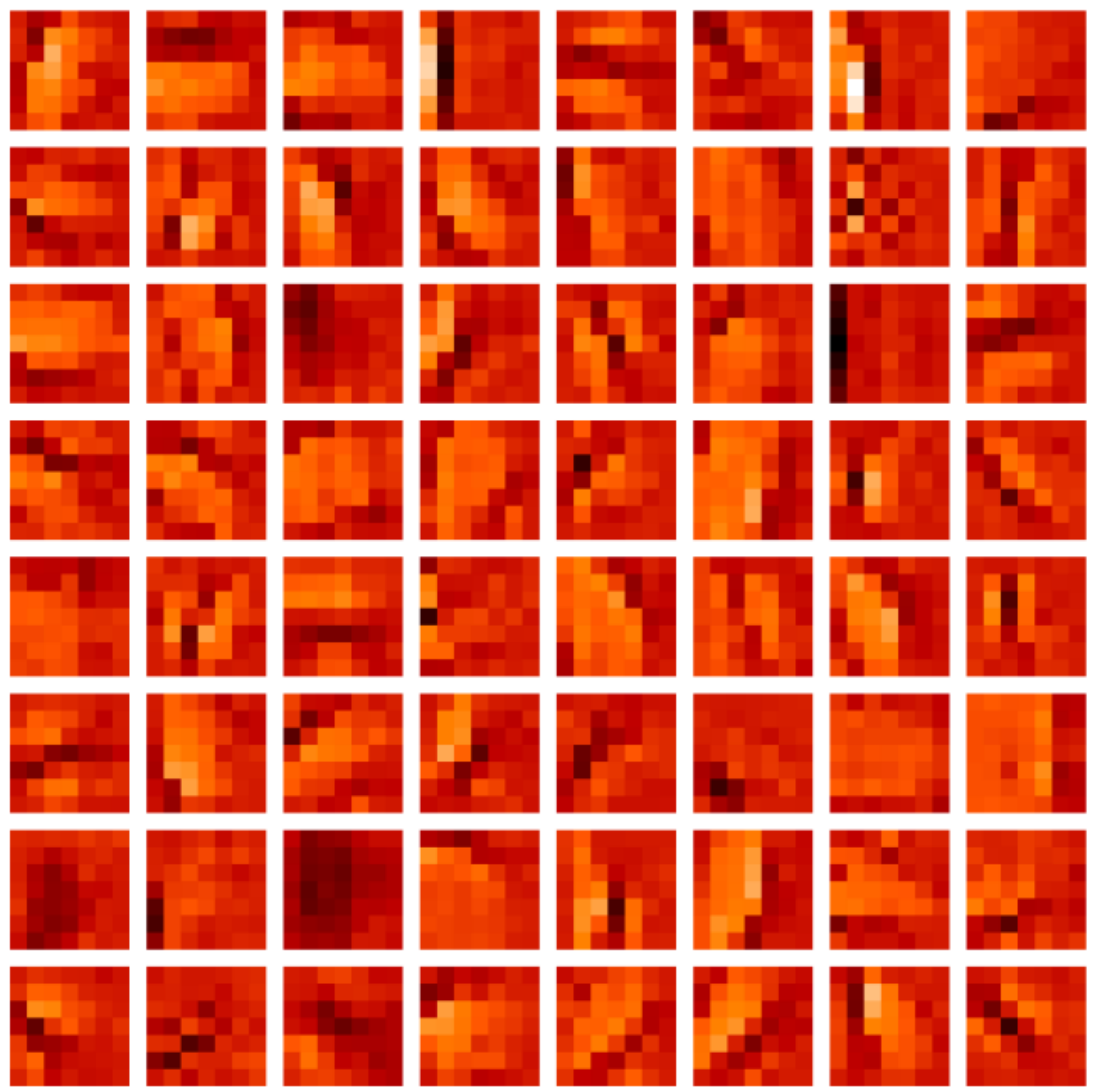


Hybrid approaches - NoVa

Aurisano et al
K. Sachdev



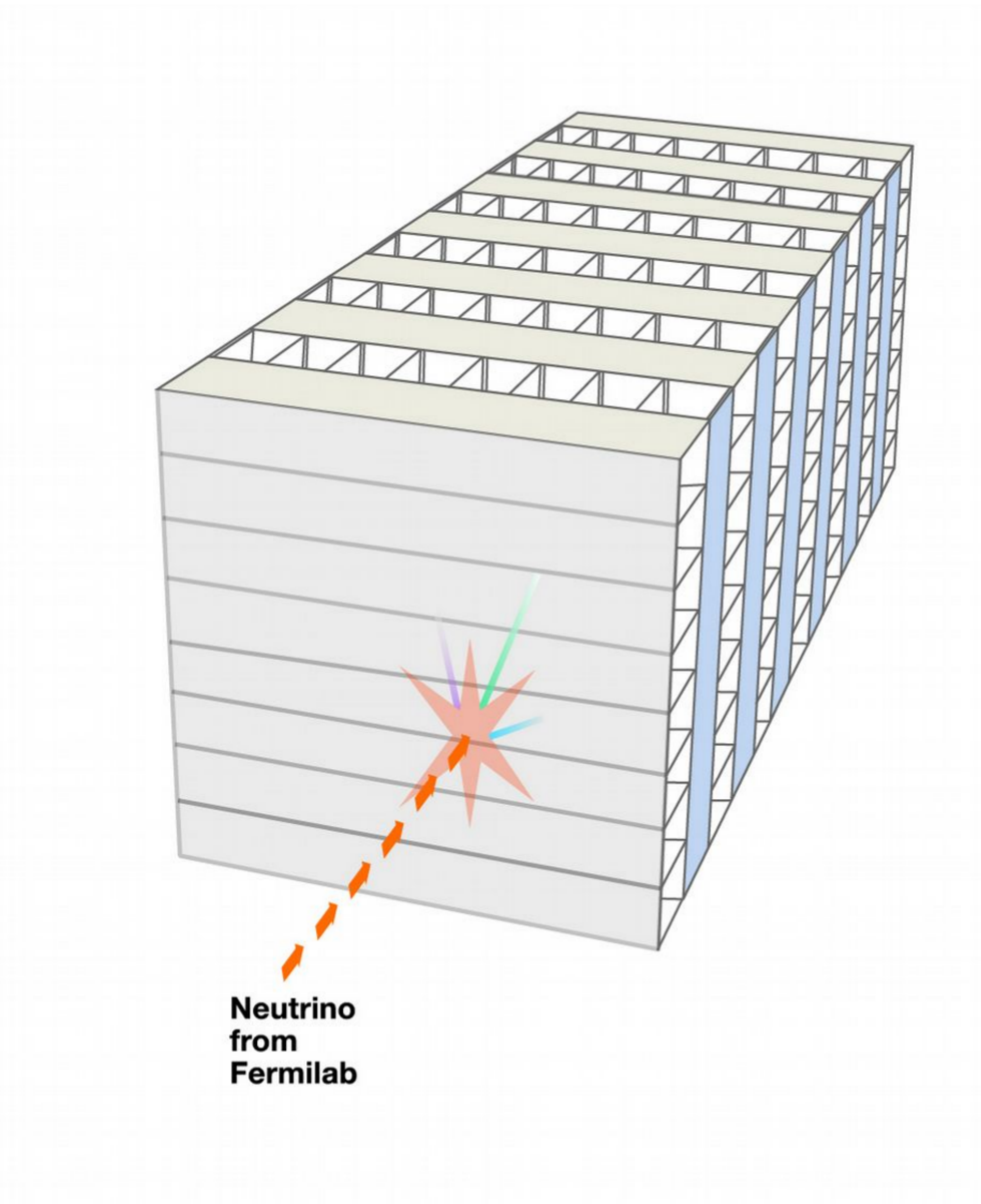




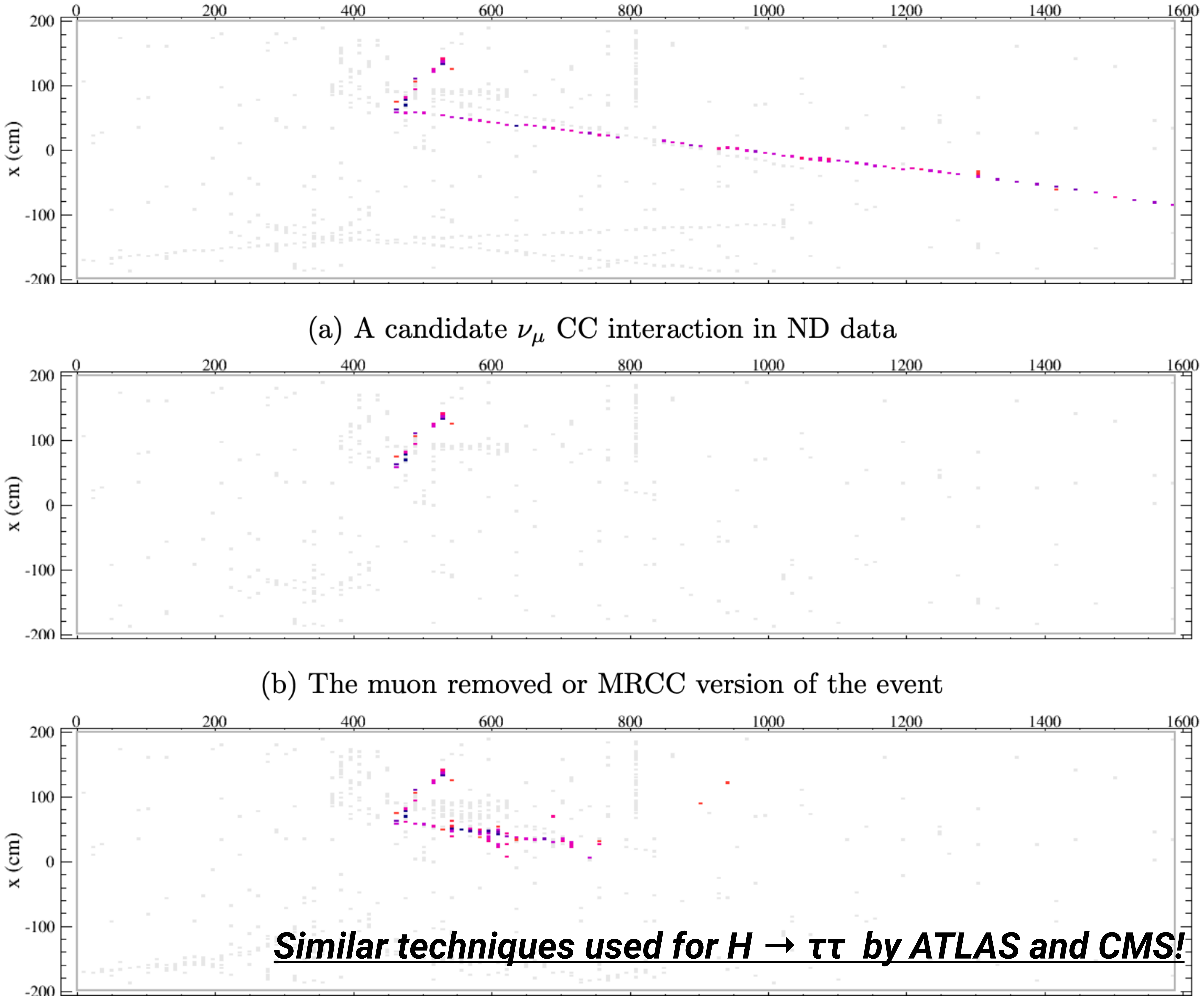
Efficiency of selecting electron neutrinos improved by 40%

**Aurisano et al**
**K. Sachdev**

Hybrid approaches - NoVa



Efficiency of selecting electron neutrinos improved by 40%



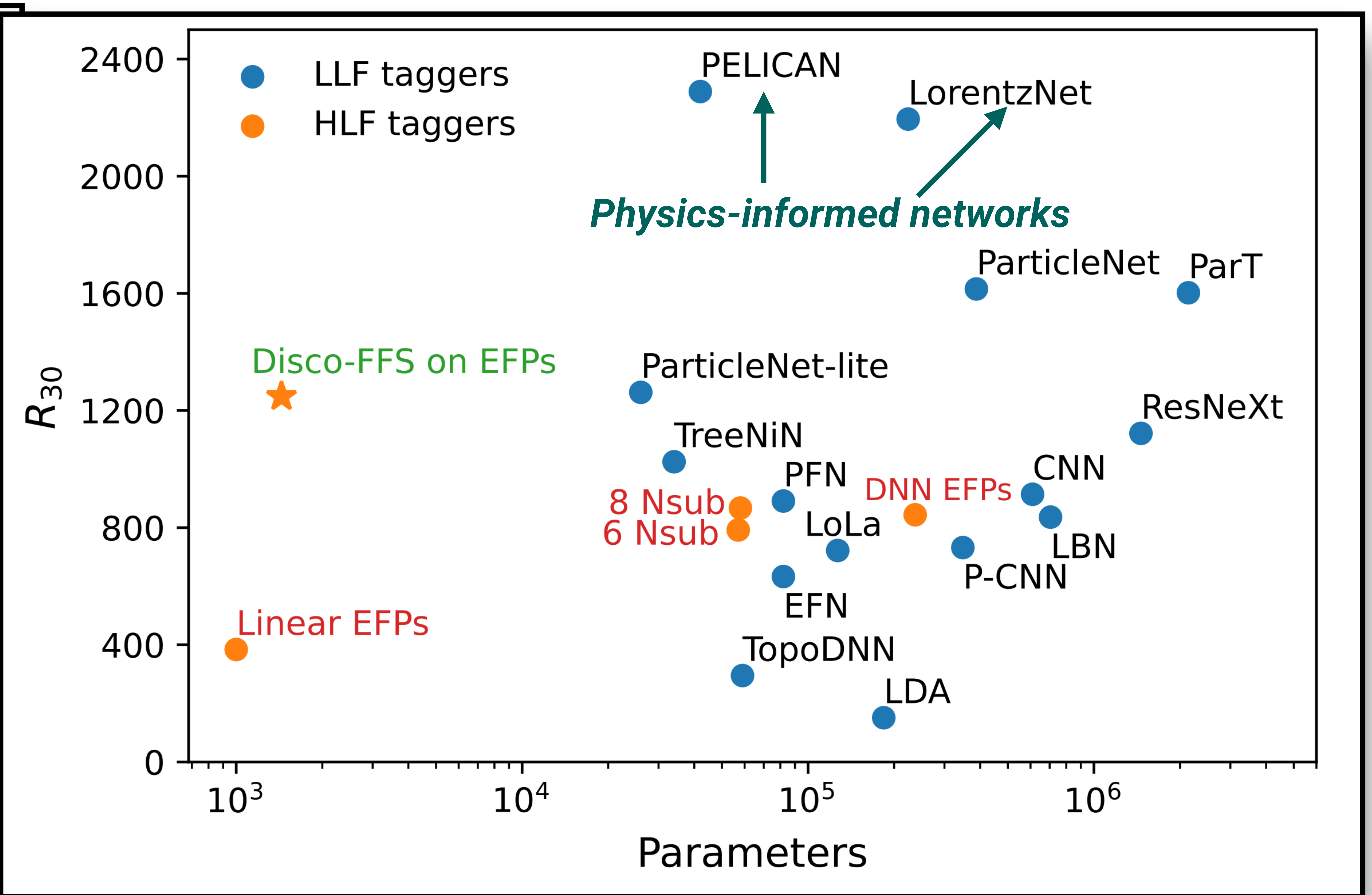
(a) A candidate ν_μ CC interaction in ND data

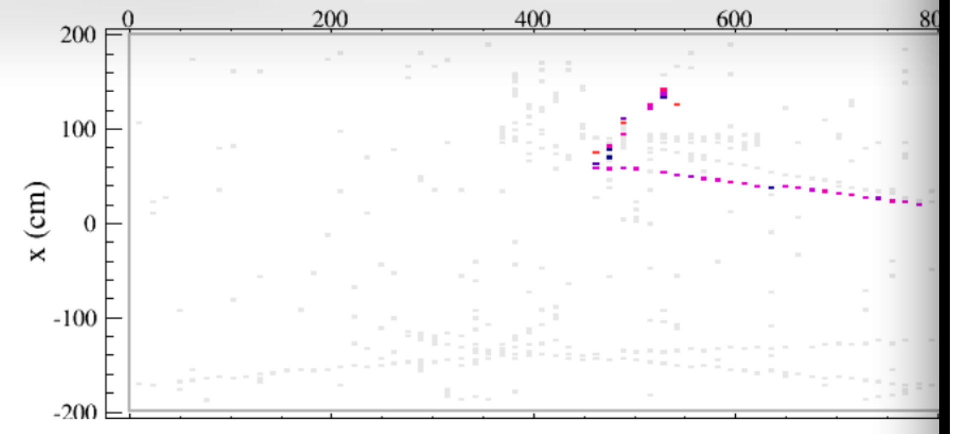
(b) The muon removed or MRCC version of the event

**Similar techniques used for $H \rightarrow \tau\tau$ by ATLAS and CMS!**

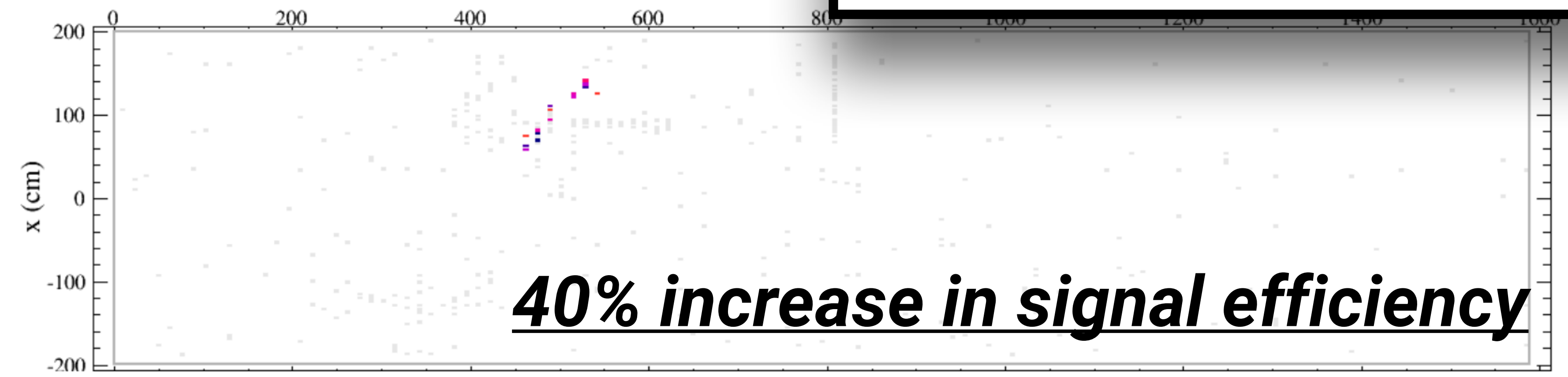
(c) A simulated electron is inserted in place of the muon to make an MRE event.

Analysis	Years of data collection	Sensitivity without machine learning	Sensitivity with machine learning	Ratio of P values	Additional data required
CMS[24] $H \to \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \to \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \to bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \to bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \to bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%

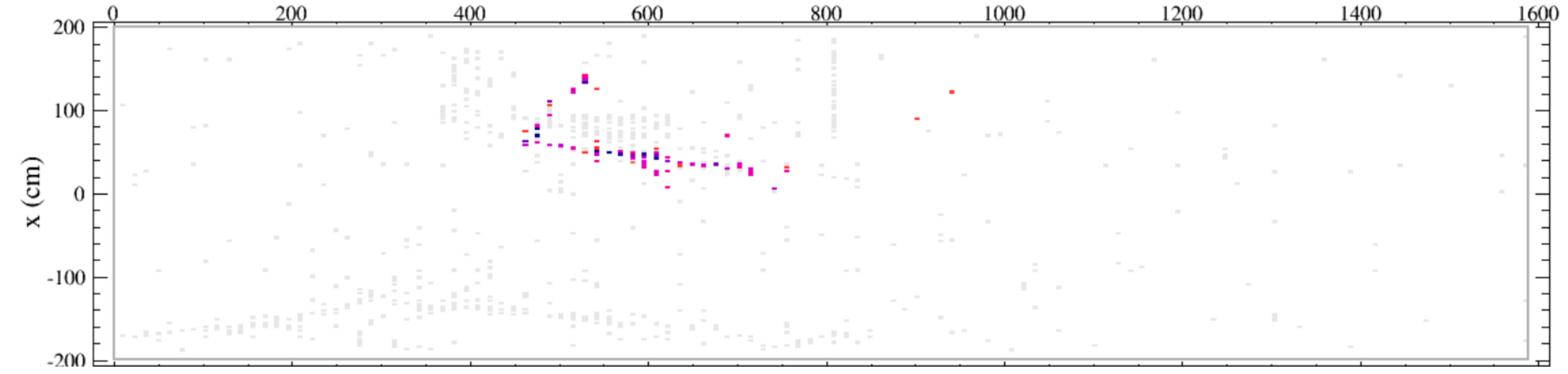




(a) A candidate ν_μ CC in

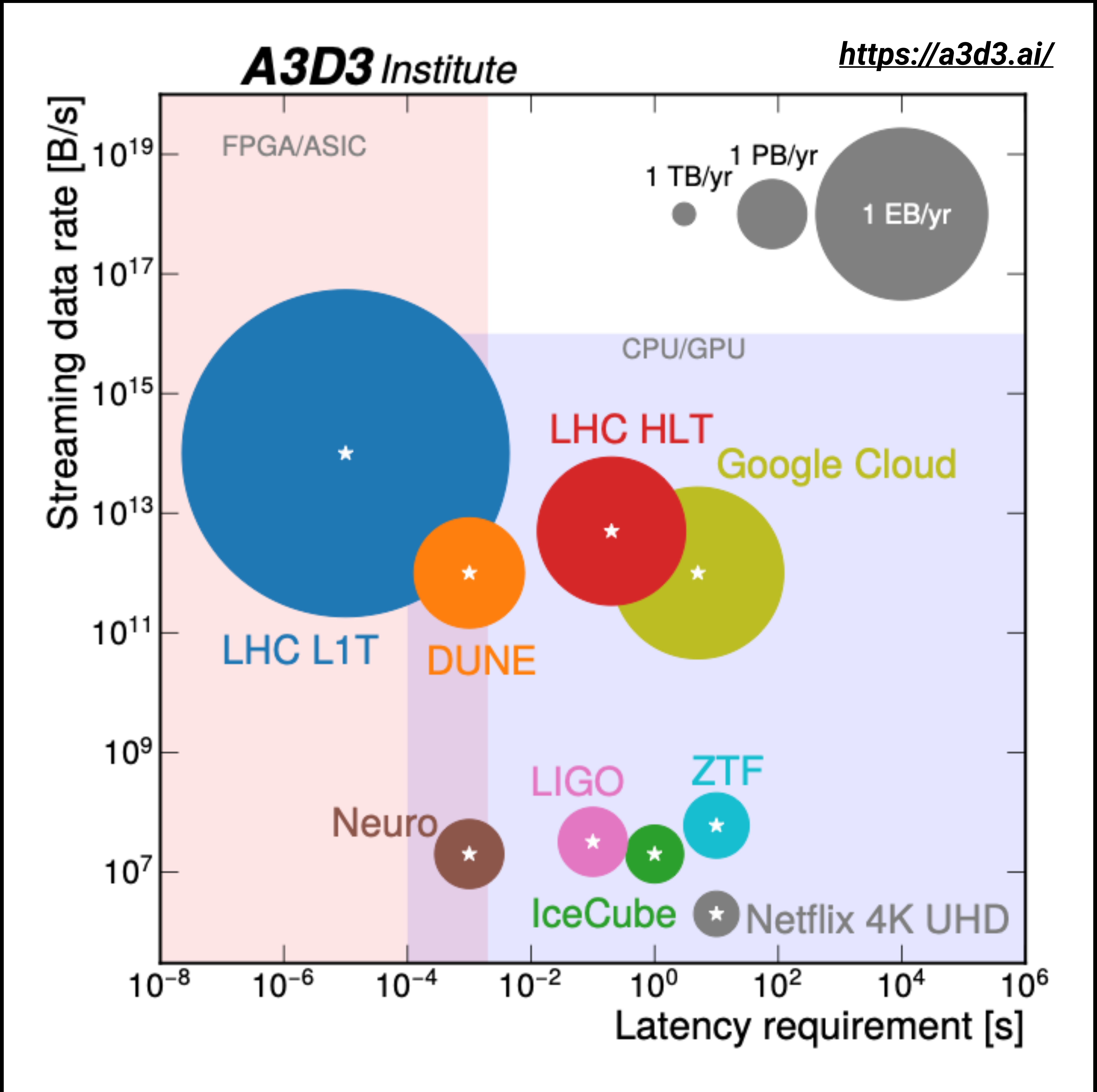


(b) The muon removed or MRCC version of the event



(c) A simulated electron is inserted in place of the muon to make an MRE event.

ML for higher sensitivity $\leftrightarrow$ ML for higher efficiency



A3D3 Institute
https://a3d3.ai/
FPGA/ASIC
CPU/GPU
1 TB/yr
1 PB/yr
1 EB/yr
LHC L1T
DUNE
LHC HLT
Google Cloud
Neuro
LIGO
IceCube
ZTF
Netflix 4K UHD
Streaming data rate [B/s]
Latency requirement [s]
10^{19}
10^{17}
10^{15}
10^{13}
10^{11}
10^{9}
10^{7}
10^{-8} 10^{-6} 10^{-4} 10^{-2} 10^{0} 10^{2} 10^{4} 10^{6}

High Luminosity LHC

New Physics is produced 1 in a trillion

- Need more collisions to observe rare processes

High Luminosity LHC

New Physics is produced 1 in a trillion

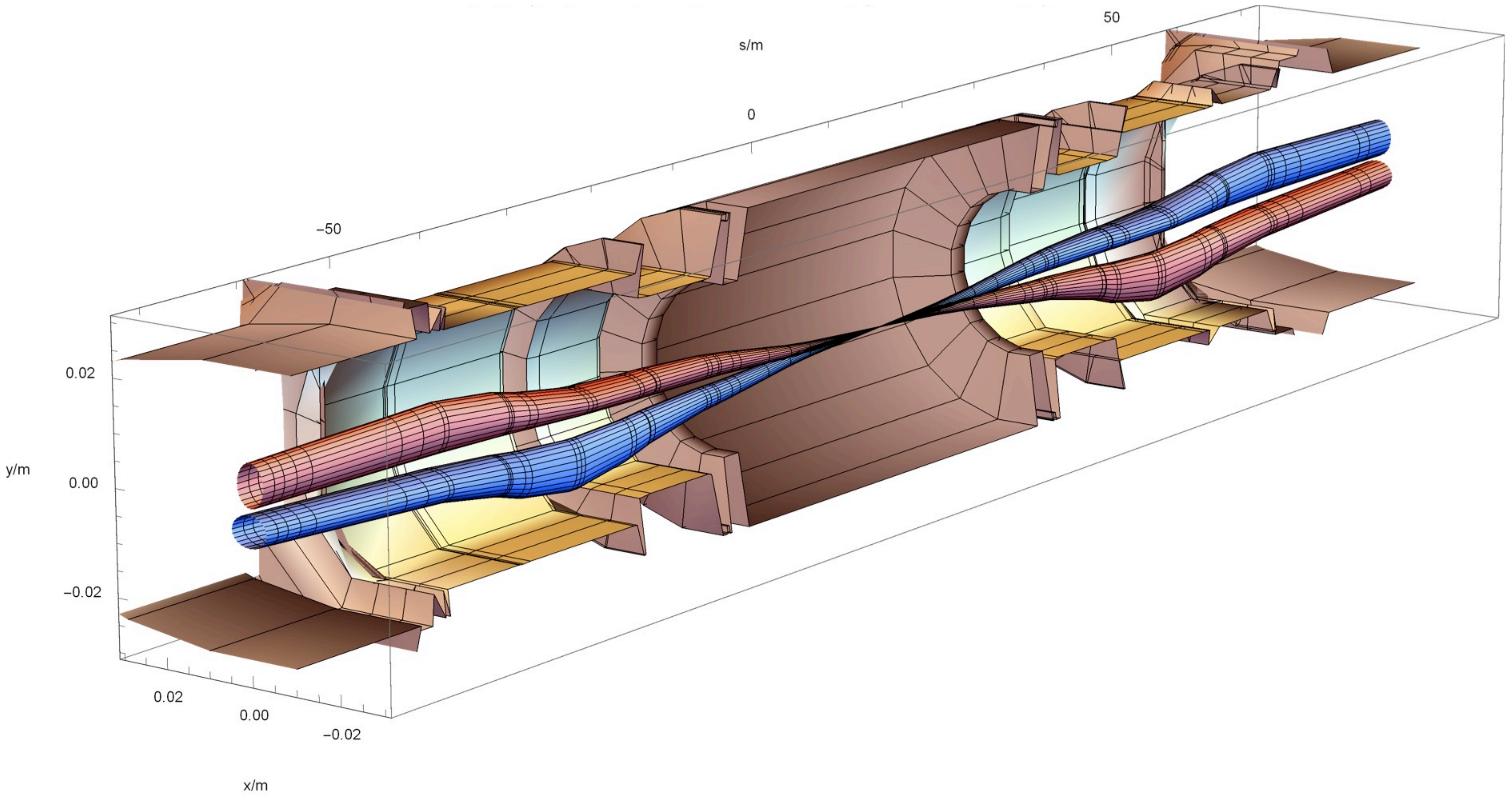
- Need <u>more collisions</u> to observe rare processes

High Luminosity LHC

- $\times$10 increase in data size
- $\times$3 collisions per second

How

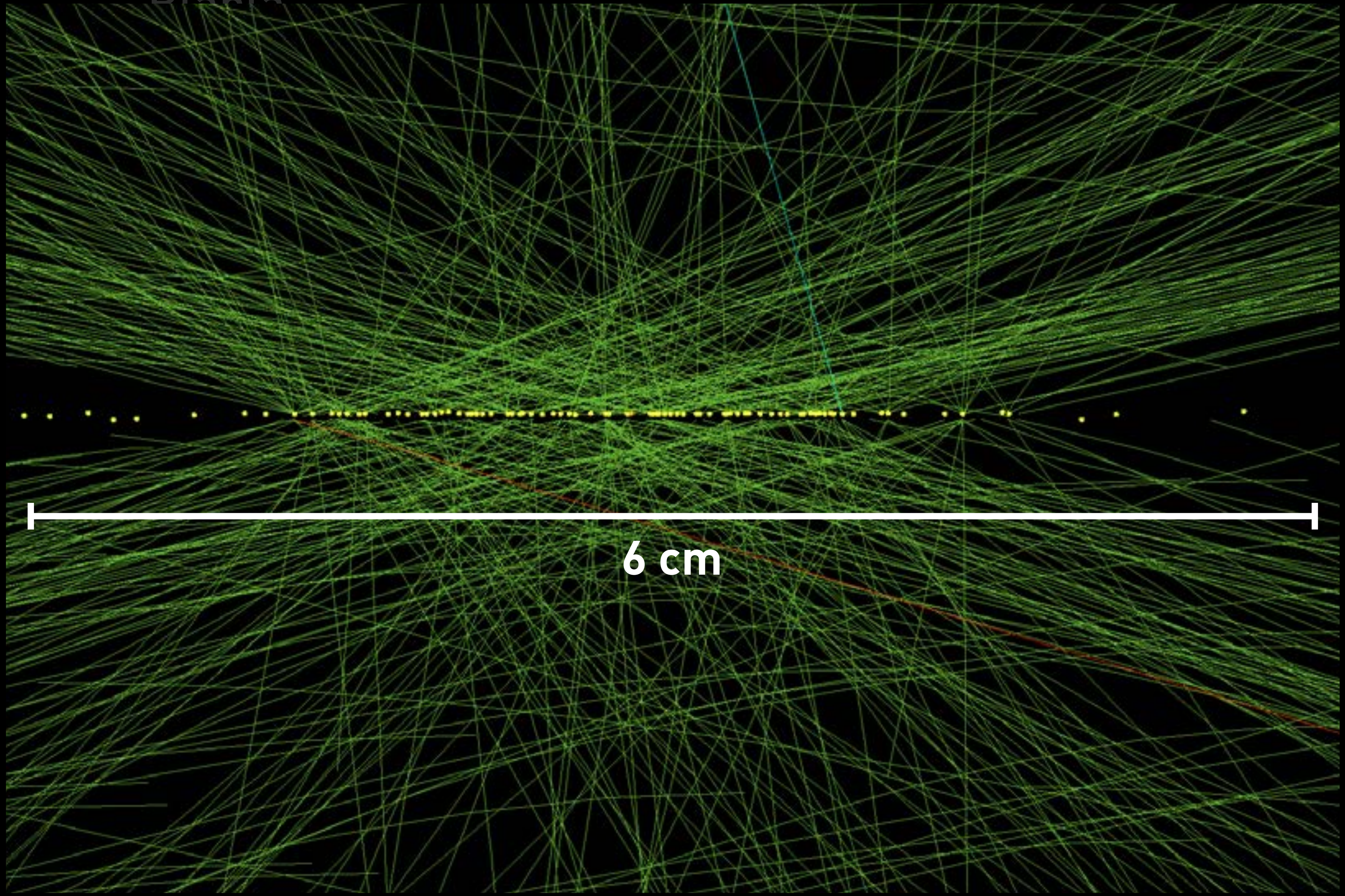
- $\times$2 protons per bunch
- Squeeze beam at interaction point (β*)



2022	2023	2024	2025	2026	2027	2028	2029	...	2037	2038
LHC				MAJOR UPGRADE			HL-LHC			
Run 3							Run 4+5			

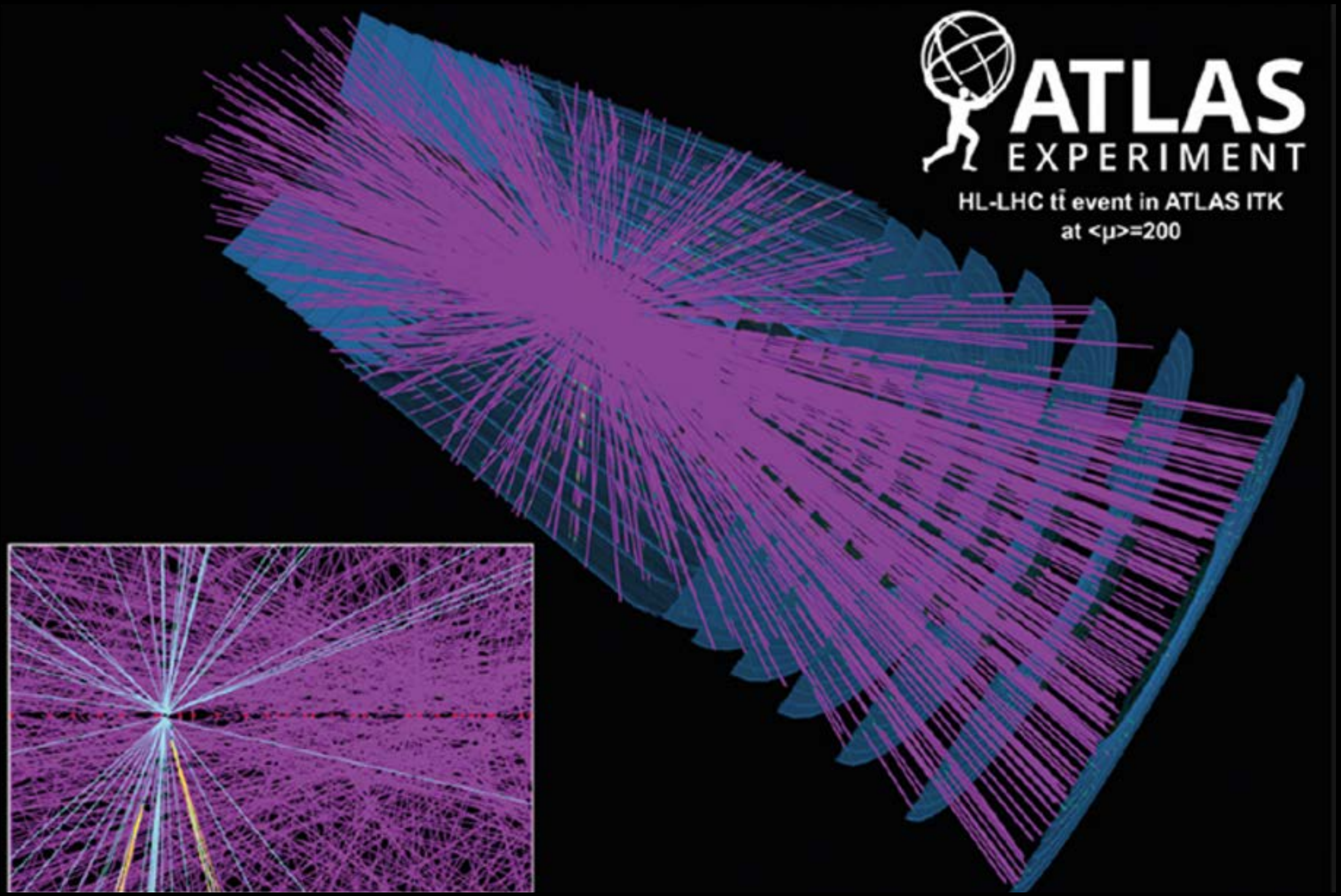
LHC

78 vertices
(average 60)



High Luminosity LHC

200 vertices
(average 140)



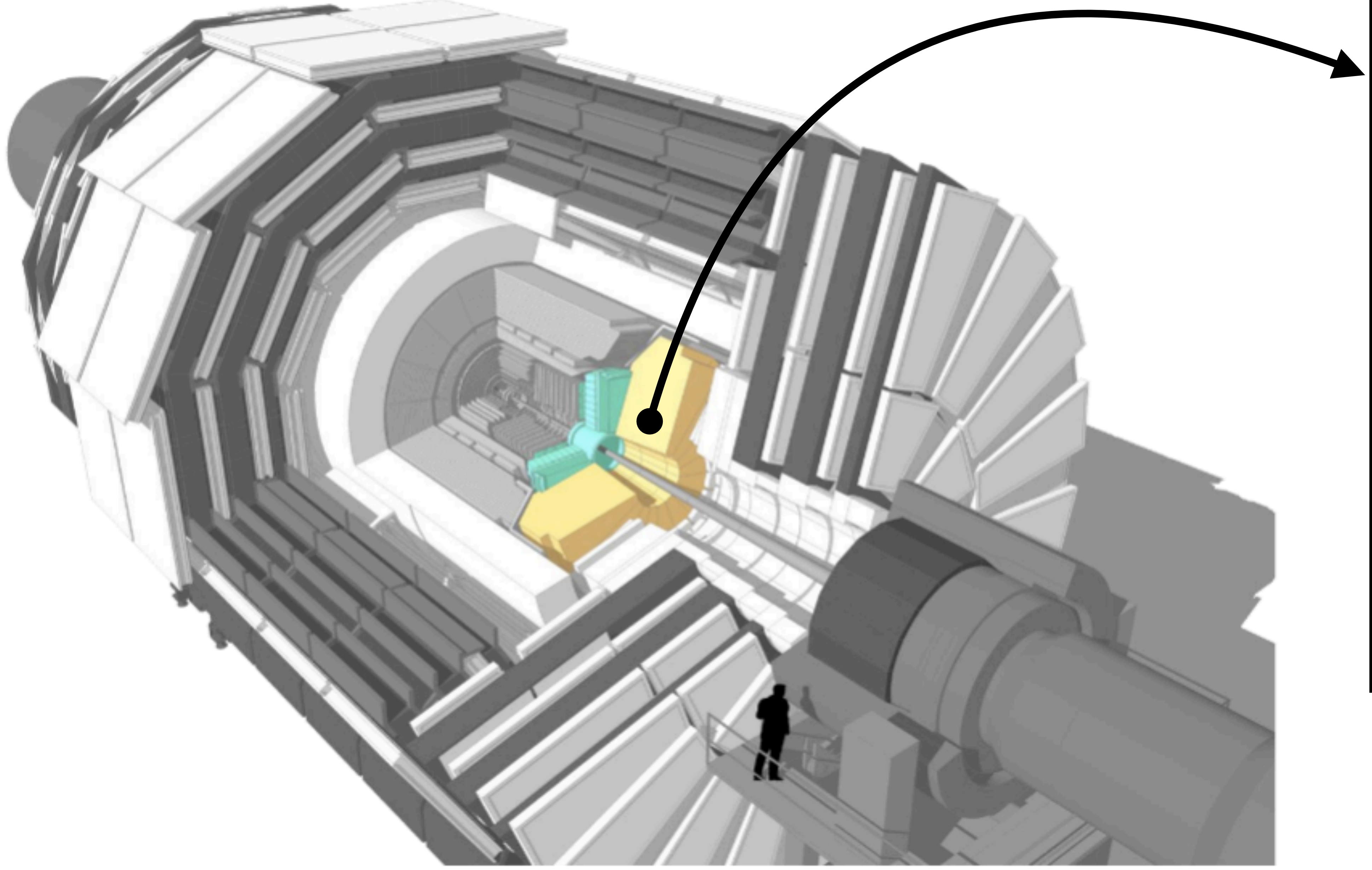
2022	2023	2024	2025	2026	2027	2028	2029	...	2037	2038
LHC				MAJOR UPGRADE			HL-LHC			
Run 3							Run 4+5			

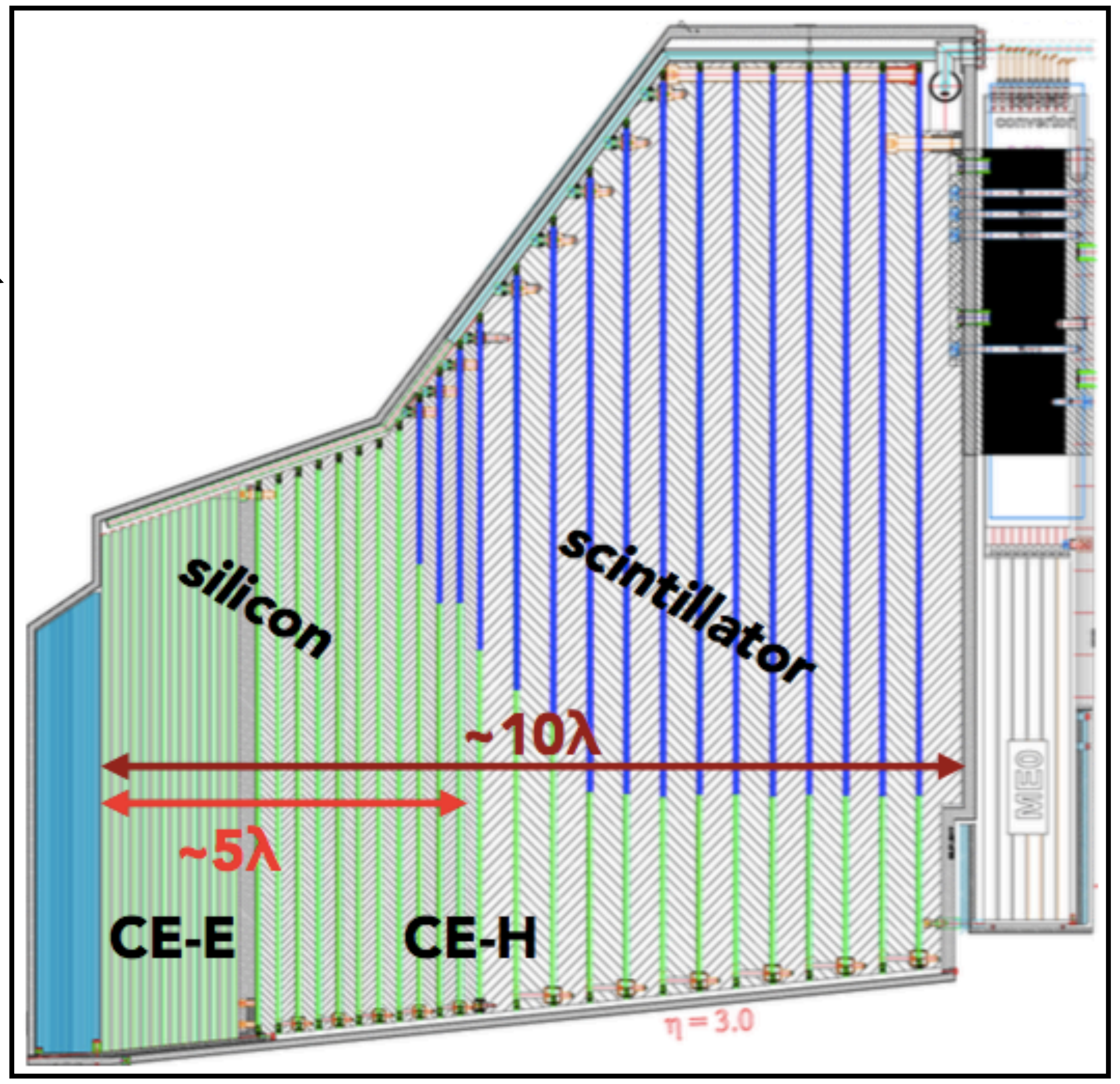
High Luminosity LHC

Must maintain physics acceptance → better detectors

CMS High Granularity (endcap) calorimeter
- 85K (today) → 6M (HL-LHC) readout channels

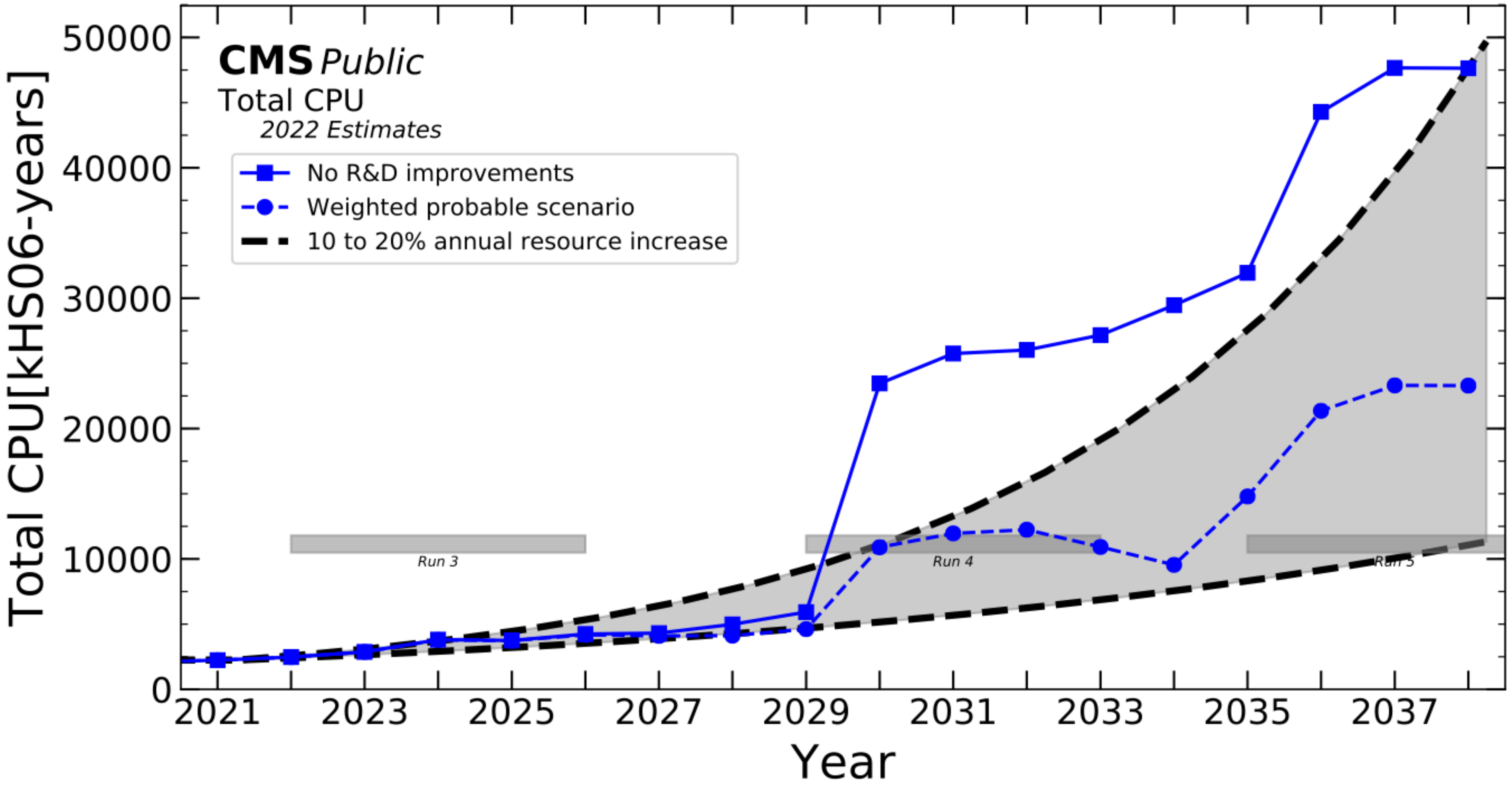
More collisions
More readout channels





CMS HGCAL TDR

Computing resources



... **flat** computing budget

Need innovation and new techniques to maintain physics reach
while staying within throughout requirements!

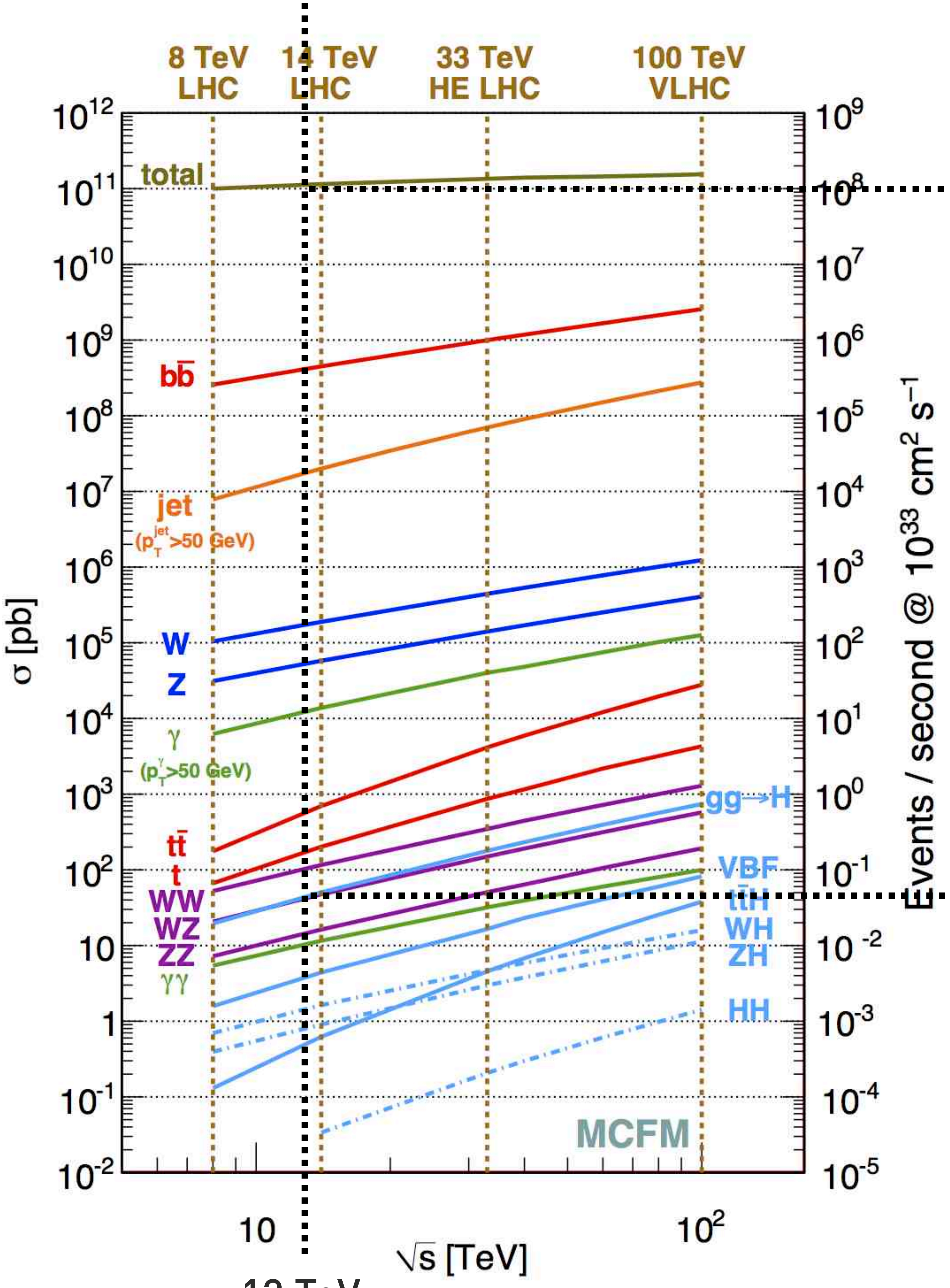
Todays algorithms will not be sustainable in HL-LHC!

→ Utilise modern Machine Learning to become

faster
better
and do more

~10 billion collisions per second

~10 PB of data per second



Total

gg→H

13 TeV

Higgs produced in
~1 out of 10^9 collisions

Saving all collisions not useful
(even if we could)!


Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
~10 billion collisions per second
~10 PB of data per second

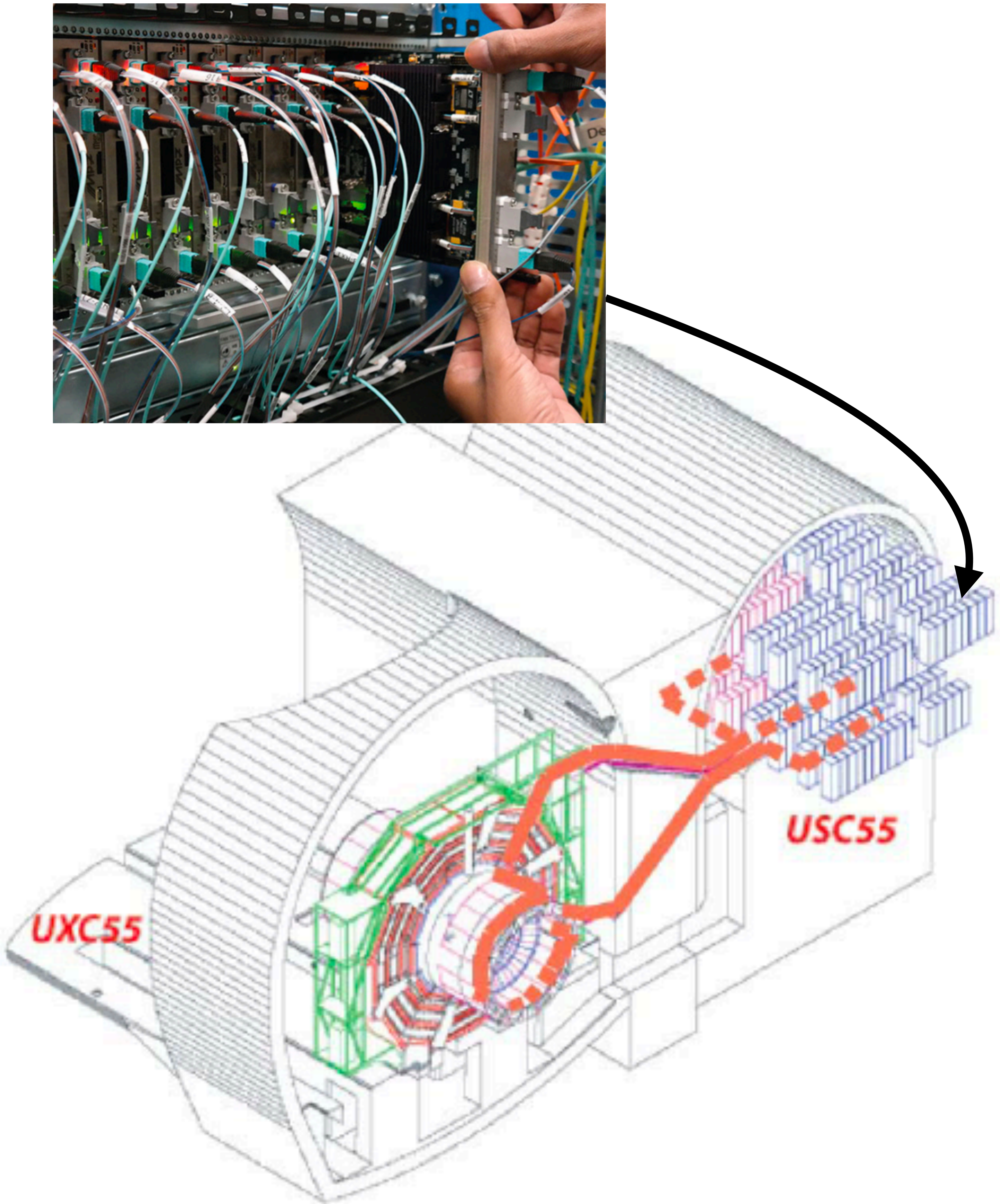

Geneva
Geneva Lake
LHCb
CMS
63 Tb/s to L1
Data temporarily stored
in detector electronics for ~12.4 μs
L1 trigger:
Decide which
event to keep
within O(1) μs
LHC


TIER 0: ∞
~0.02% of collision events remaining
Geneva
Geneva
High Level Trigger:
Latency O(100) ms
LHCb
ATLAS
ALICE
CMS
DATA
99.982% of events rejected
7.5 kHz
~Gb/s
DATA
99.75% of events rejected!
750 kHz
~Tb/s
LHC

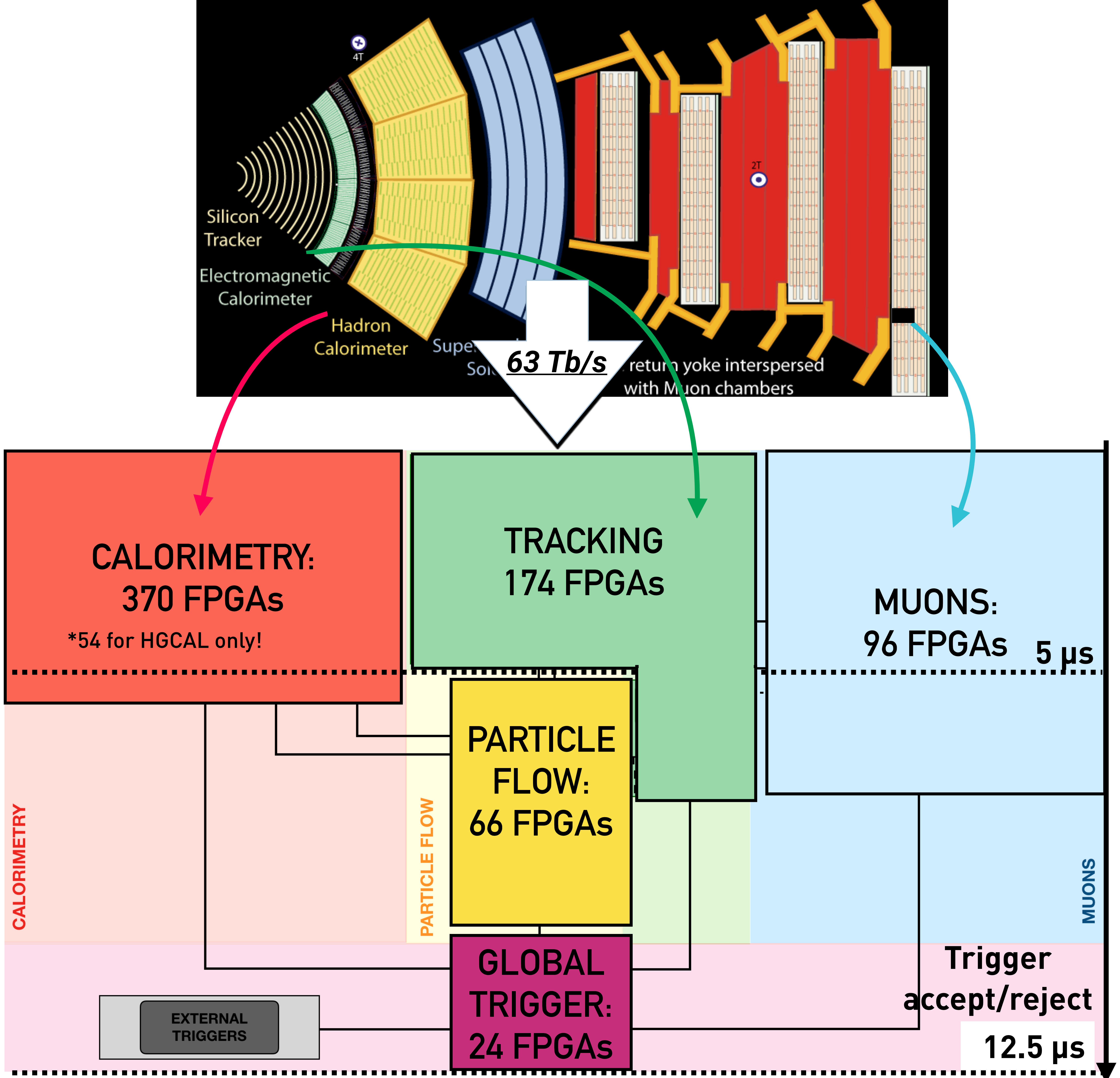
TIER 0: ∞
Geneva
High Level Trigger:
Latency O(100) ms
LHCb
ATLAS
ALICE
CMS
DATA
~1 kHz
~Gb/s
DATA
~100 kHz
~Tb/s
Level-1 trigger:
Latency O(1) μs
Detector:
40 MHz
~Pb/s
intel
Xeon® 7500
TESLA
VIRTEX-5


Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
Fast inference on specialised hardware
ASIC inference
FPGA inference
Level-1 trigger:
Latency O(1) μs
Detector:
40 MHz
~Pb/s

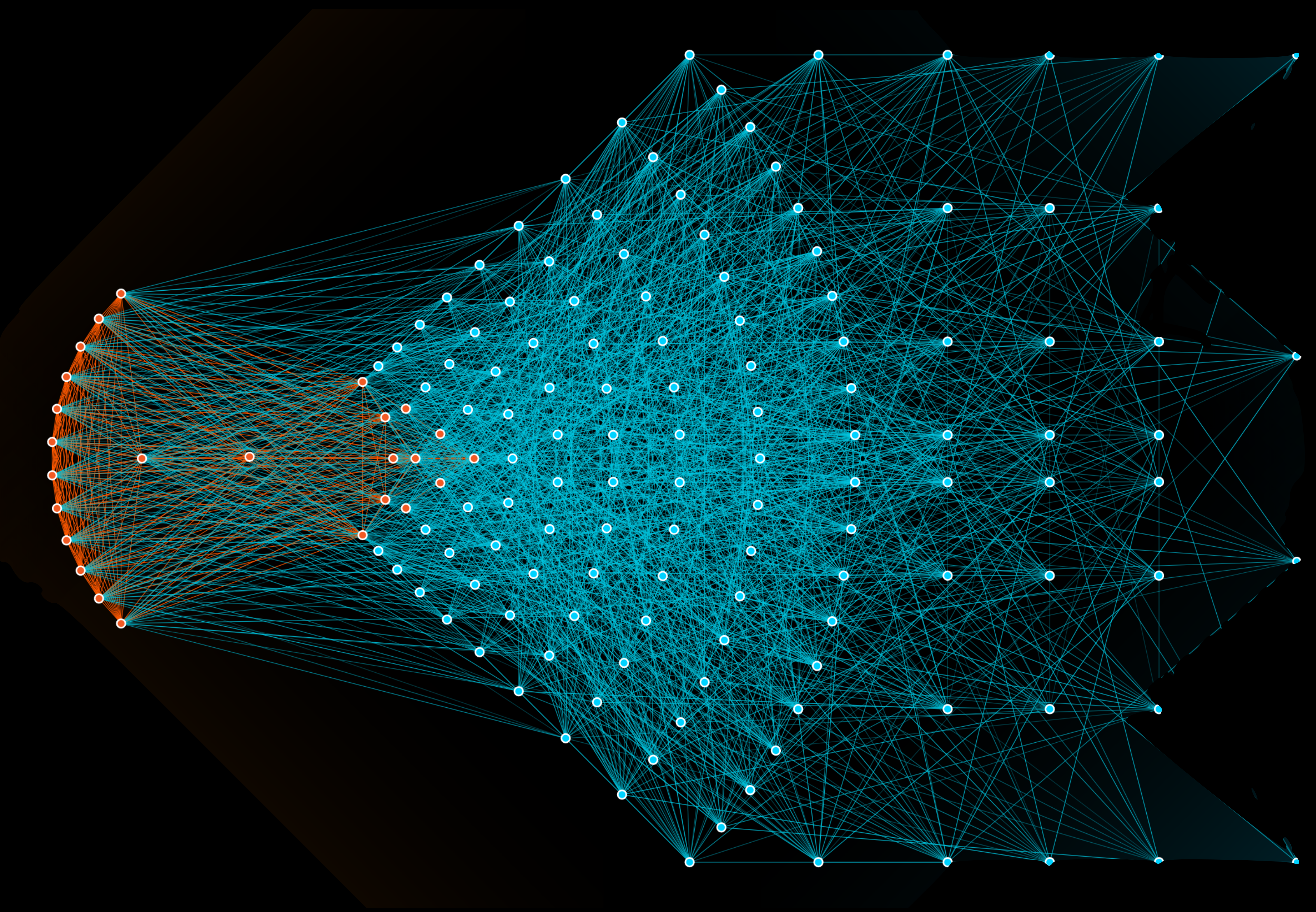
HL-LHC: CMS L1



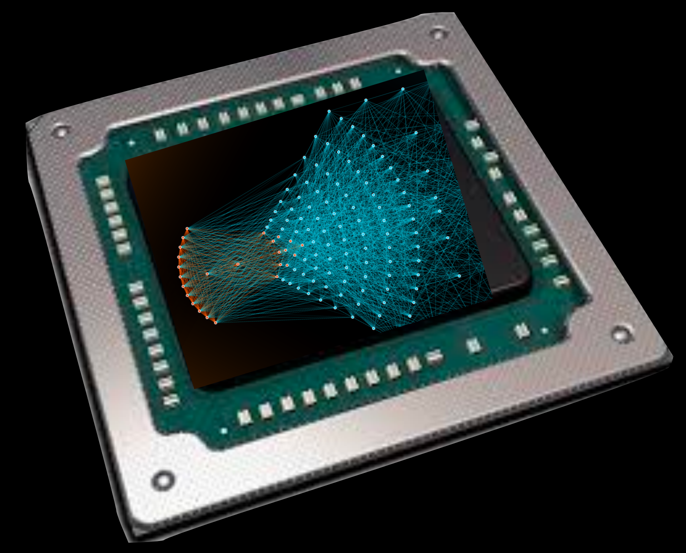
UXC55
USC55



Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
4T
2T
return yoke interspersed with Muon chambers
63 Tb/s
CALORIMETRY: 370 FPGAs
*54 for HGCAL only!
TRACKING 174 FPGAs
MUONS: 96 FPGAs
5 μs
PARTICLE FLOW: 66 FPGAs
GLOBAL TRIGGER: 24 FPGAs
EXTERNAL TRIGGERS
CALORIMETRY
PARTICLE FLOW
MUONS
Trigger accept/reject
12.5 μs



Ideally



Reality

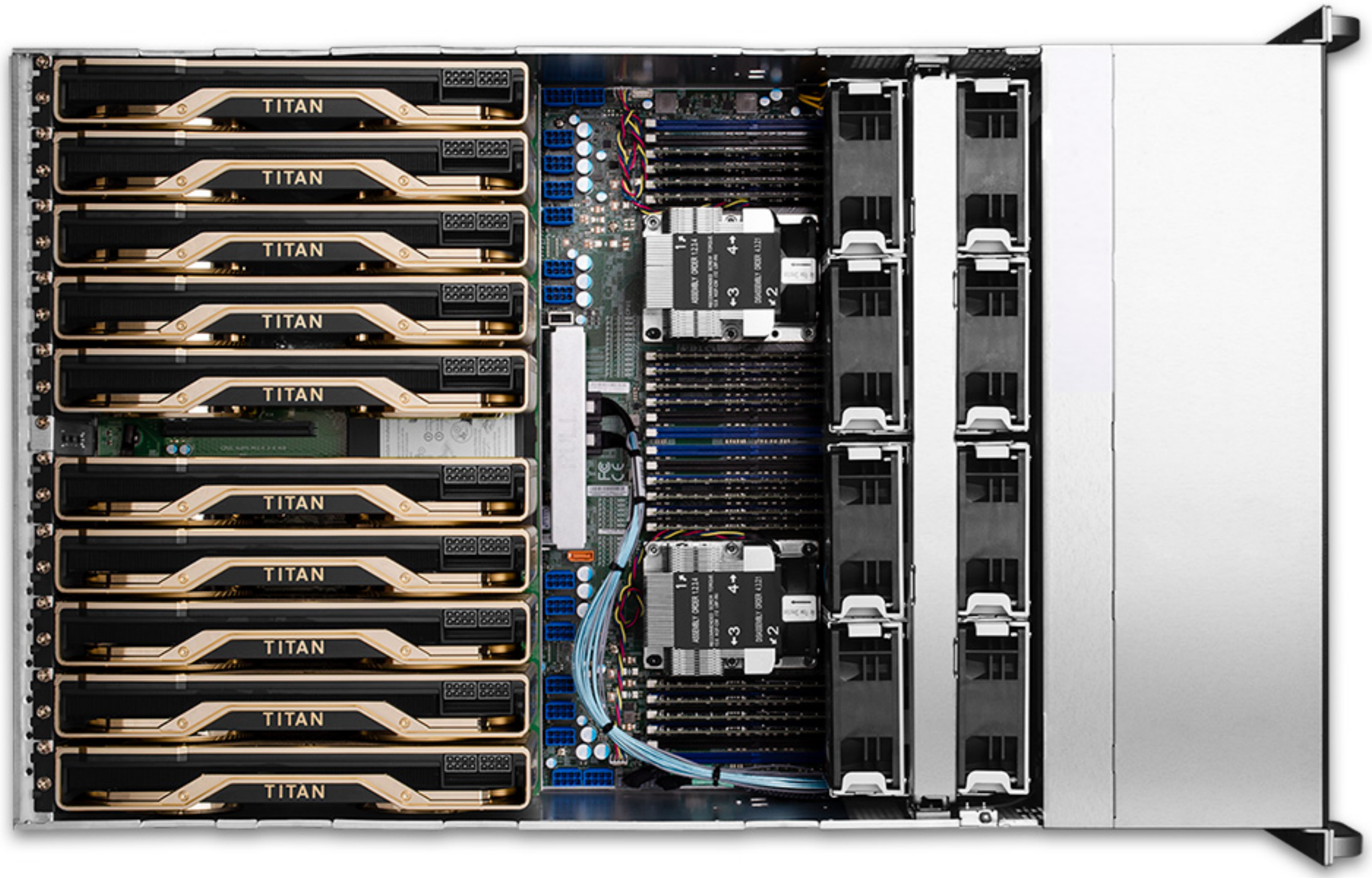
Efficient NN design for edge compute

Before deploying any DNN on chip (CMS trigger, iPhone), must make it efficient!
- Big engineering field in its own right

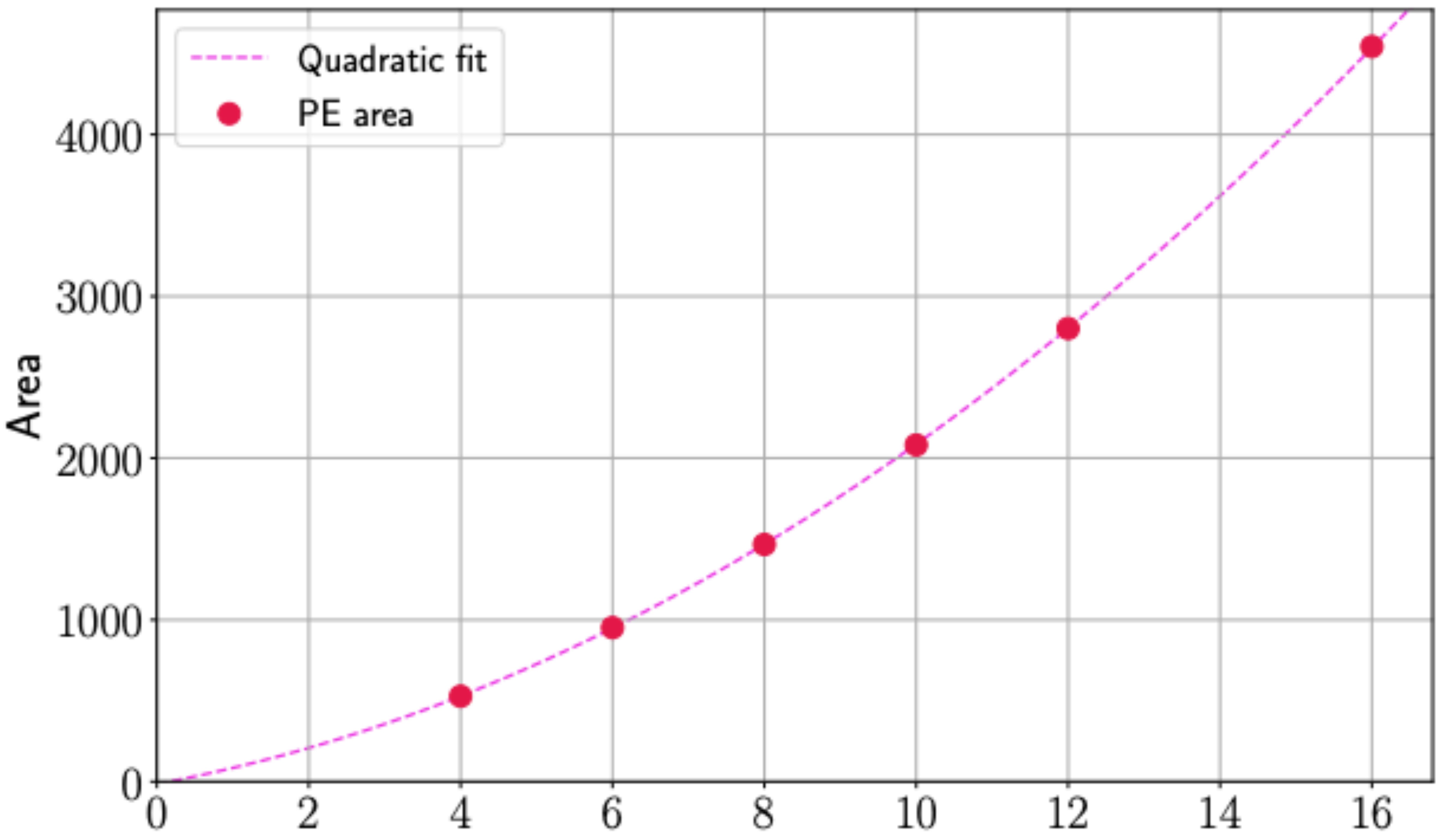
During training
- Quantization: do you really need 32-bit FP precision?
- Pruning: removal insignificant synapses
- Knowledge distillation

Post-training
- Parallelisation (lower latency ↔ more resources)

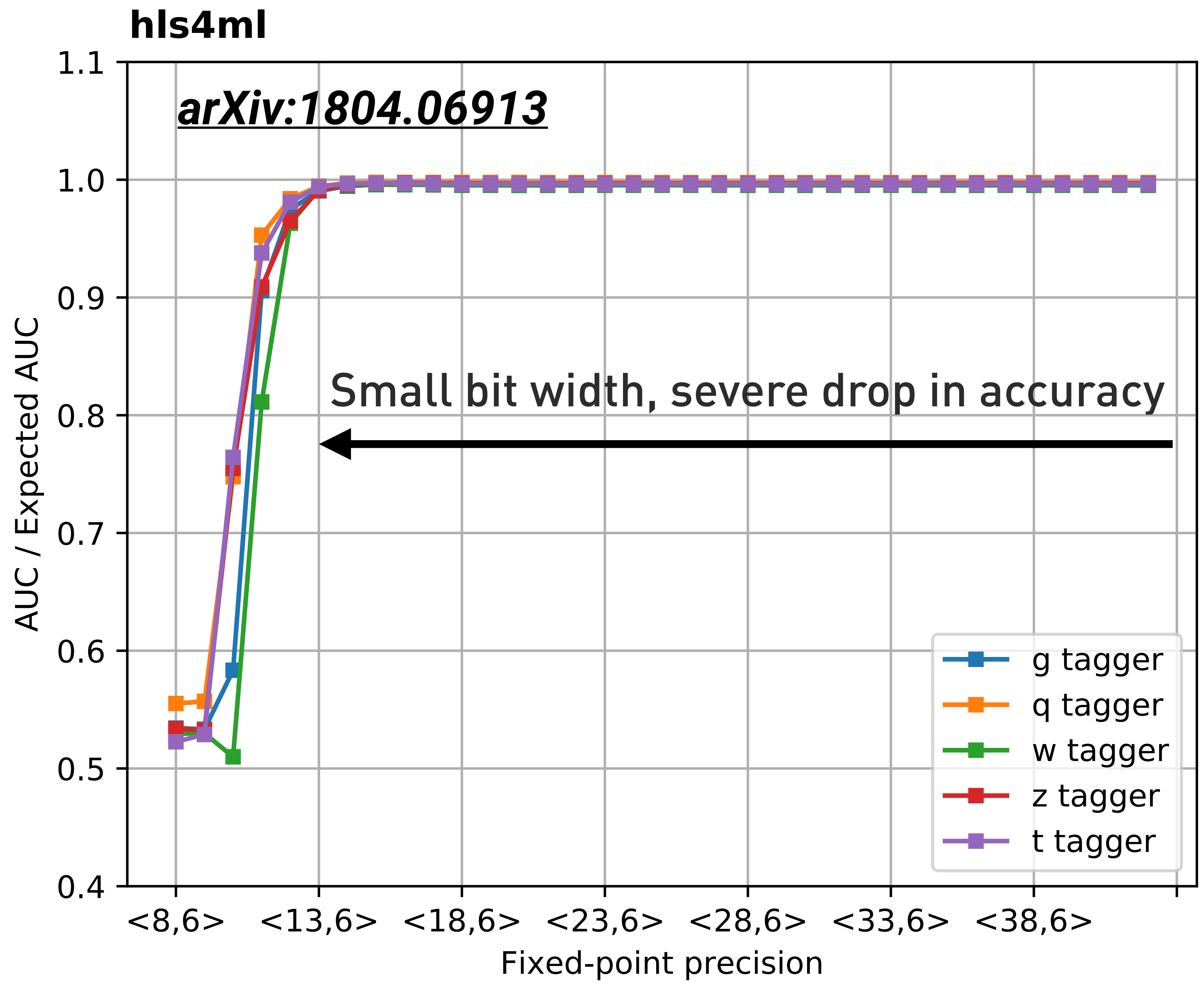
From 8 GPU server to tiny FPGA!



Quantization

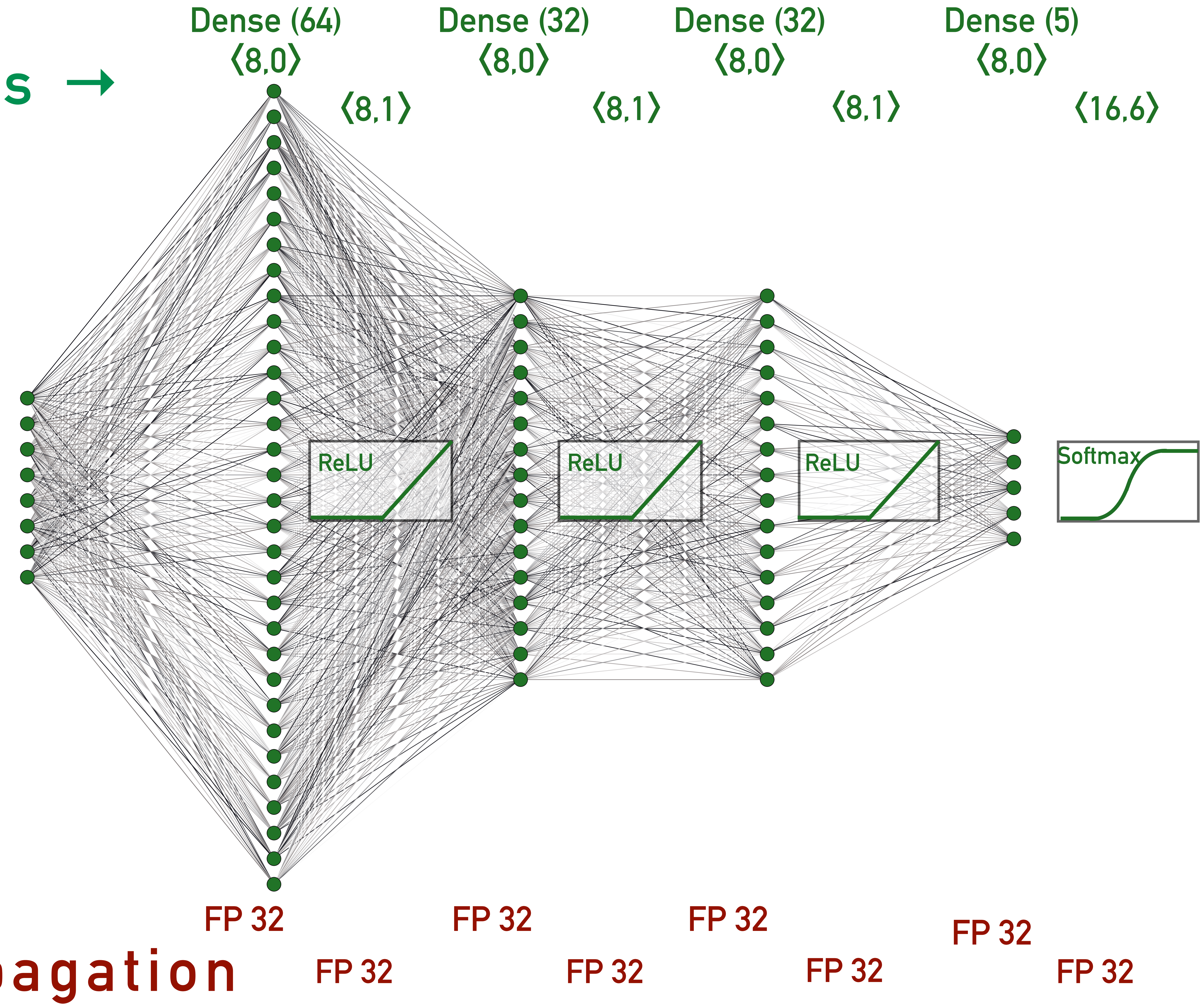


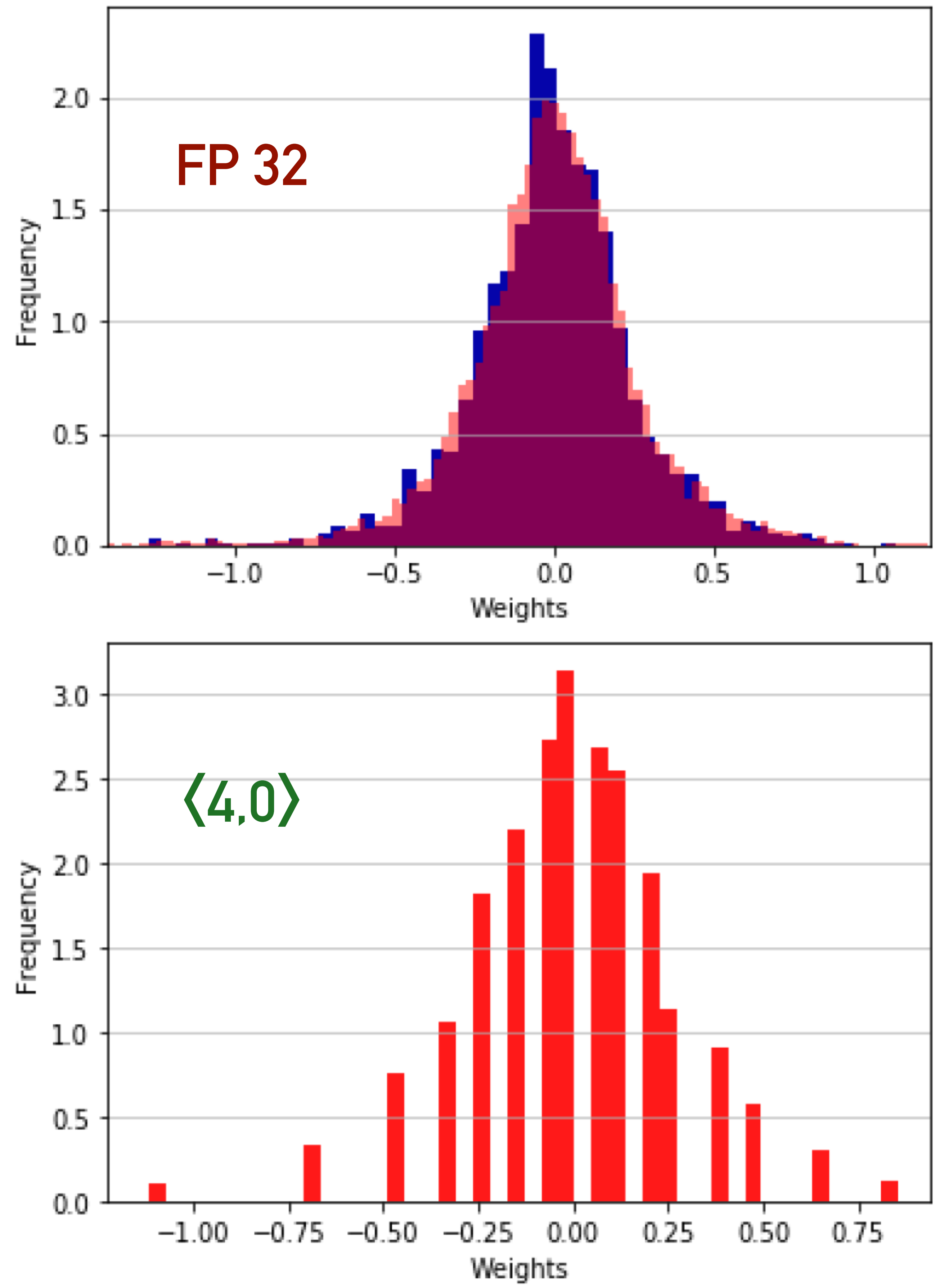
arxiv:2004.08906



Quantization-aware training

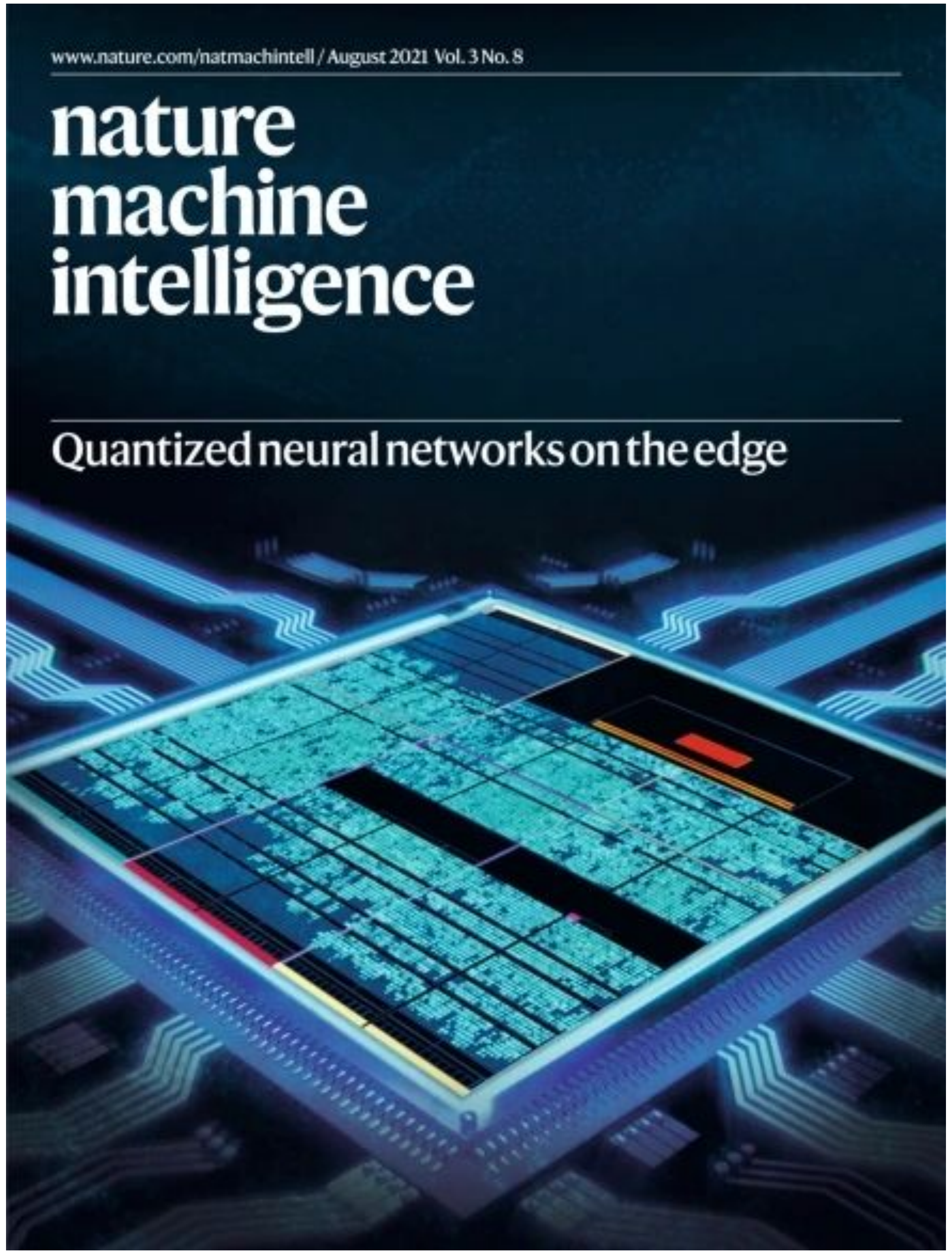
Forward pass →

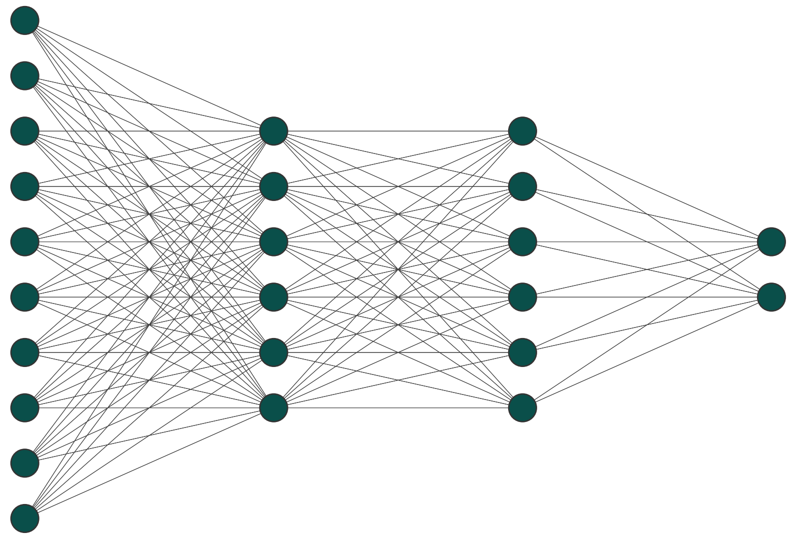




← Back propagation

Nature Machine Intelligence 3 (2021)





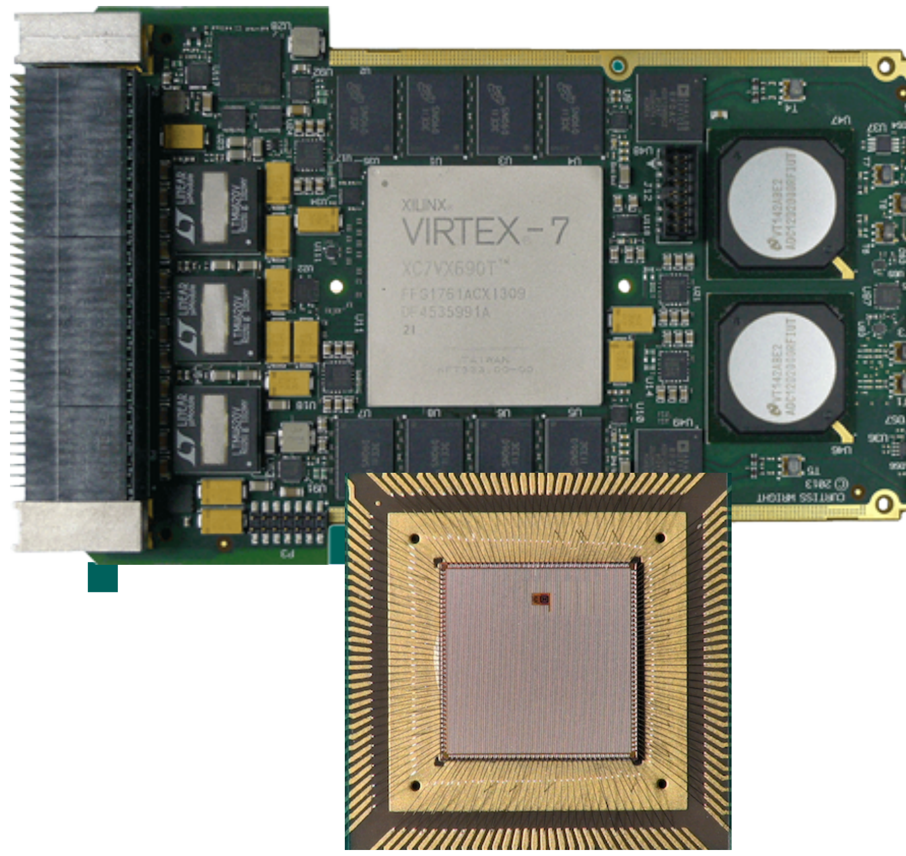
QKeras Quantization-aware training

Google AI

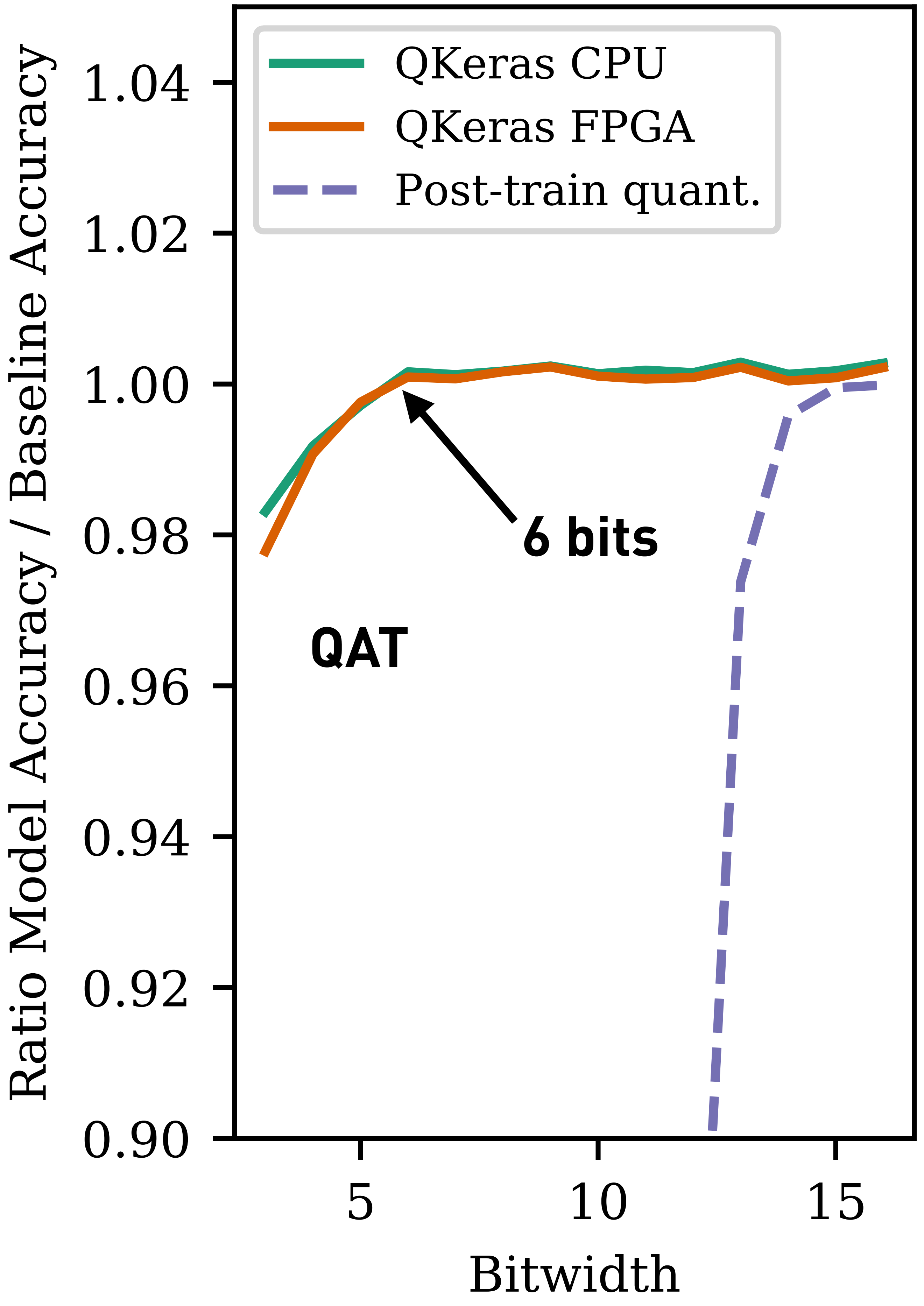
QKeras model

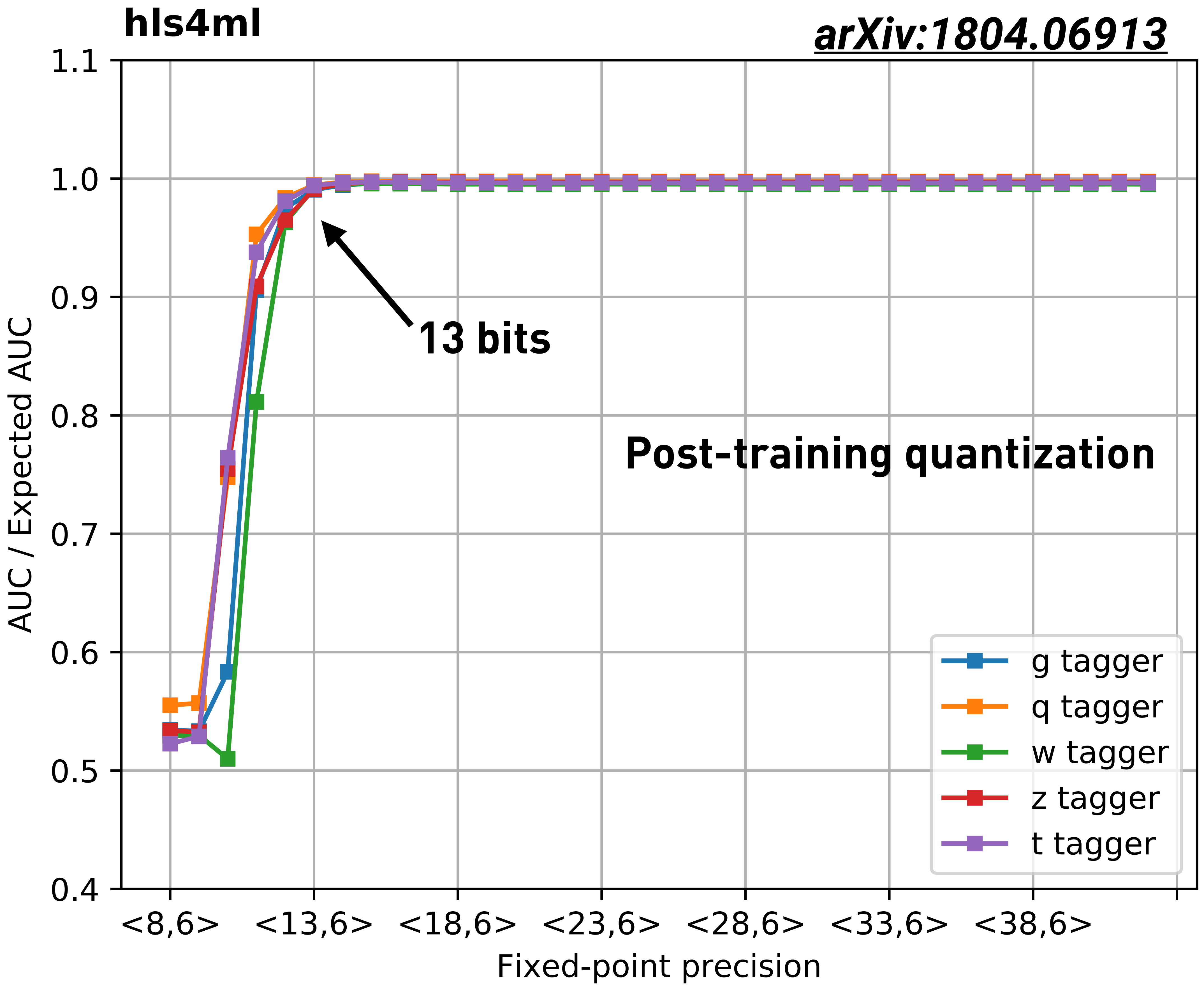
hls4ml

CERN



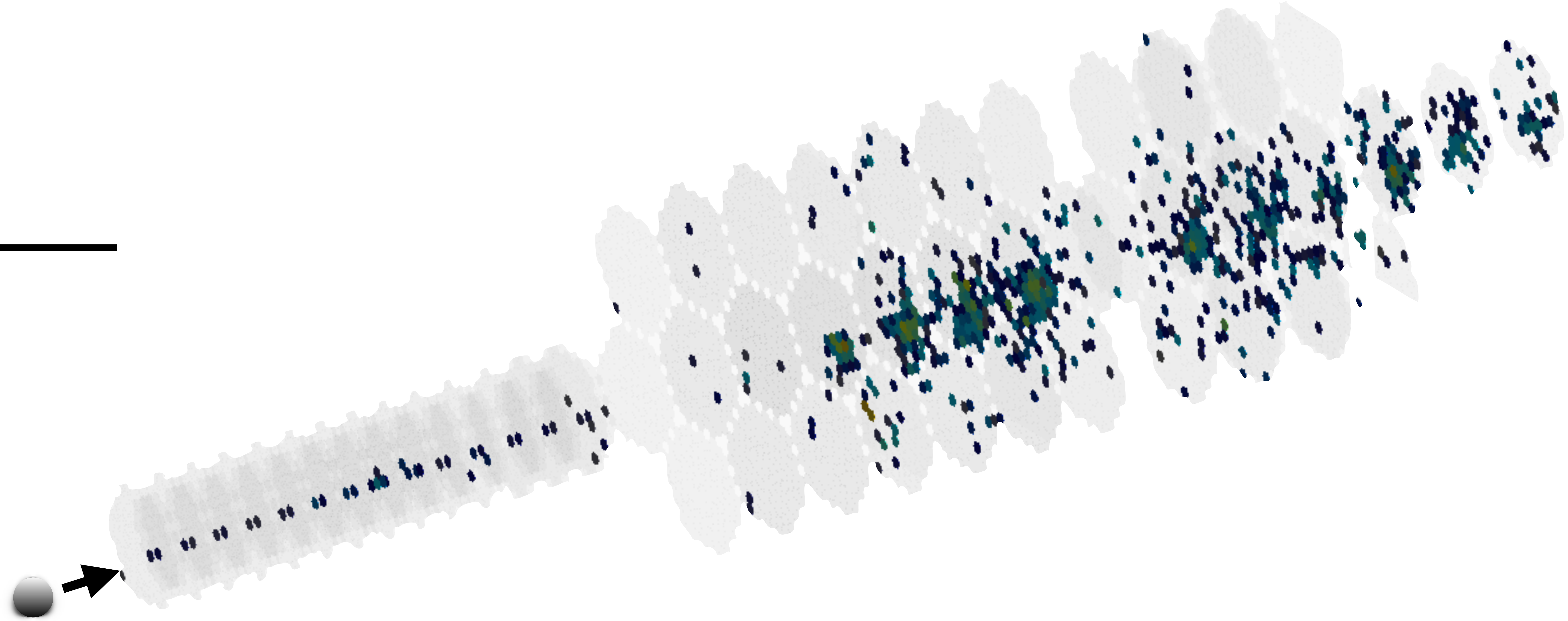
FPGA performance





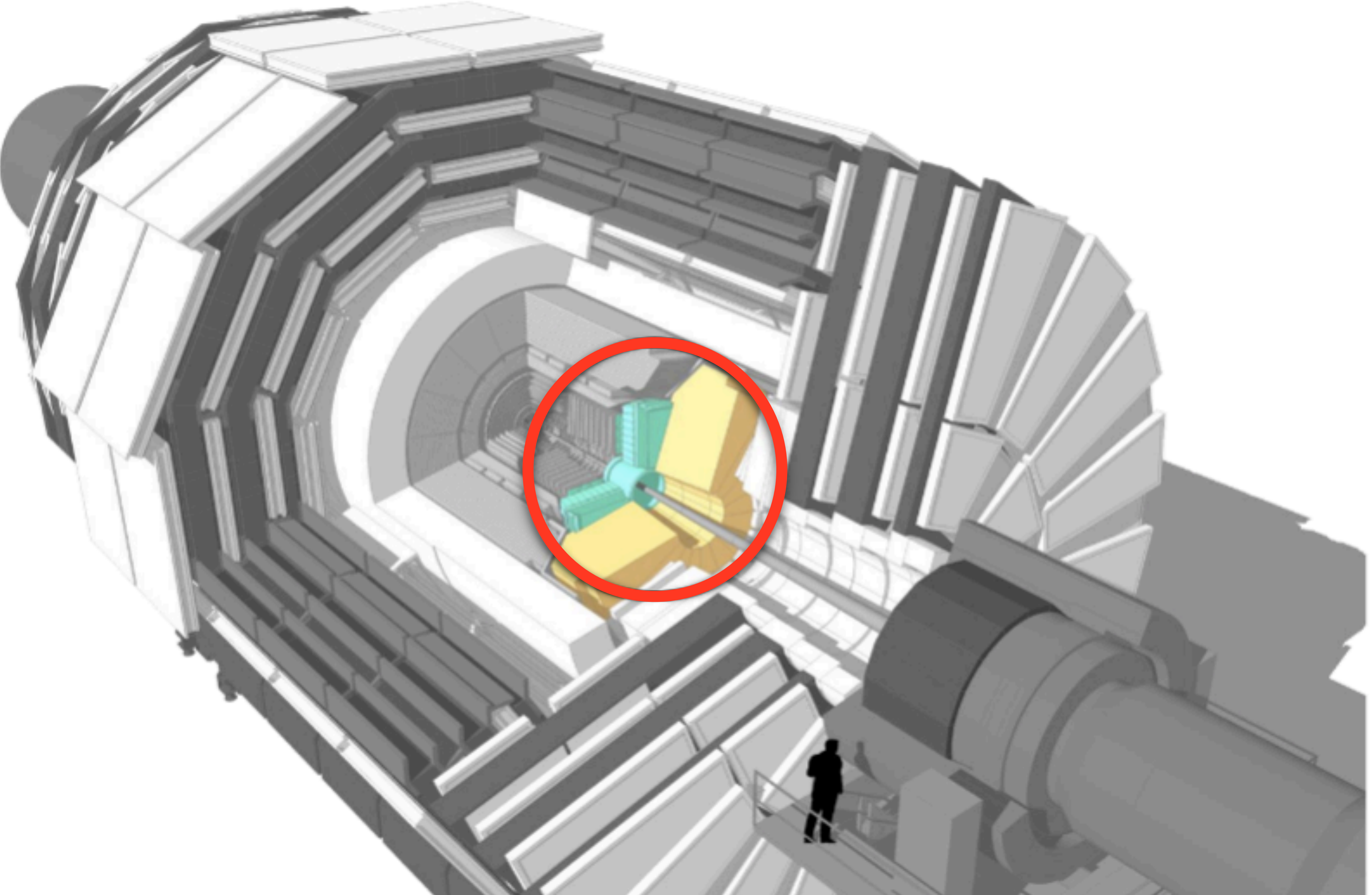
Nature Machine Intelligence 3 (2021)

ML for compression



CMS High Granularity calorimeter

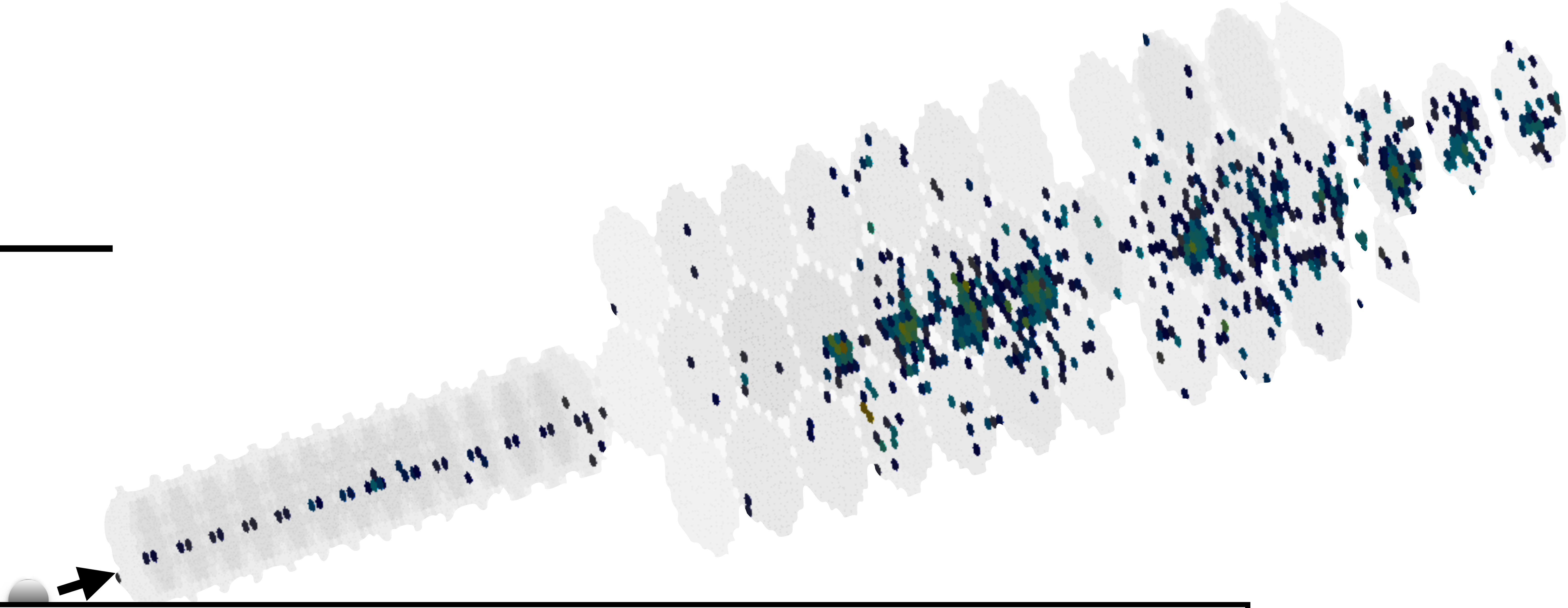
- 6.5 million readout channels, 50 layers



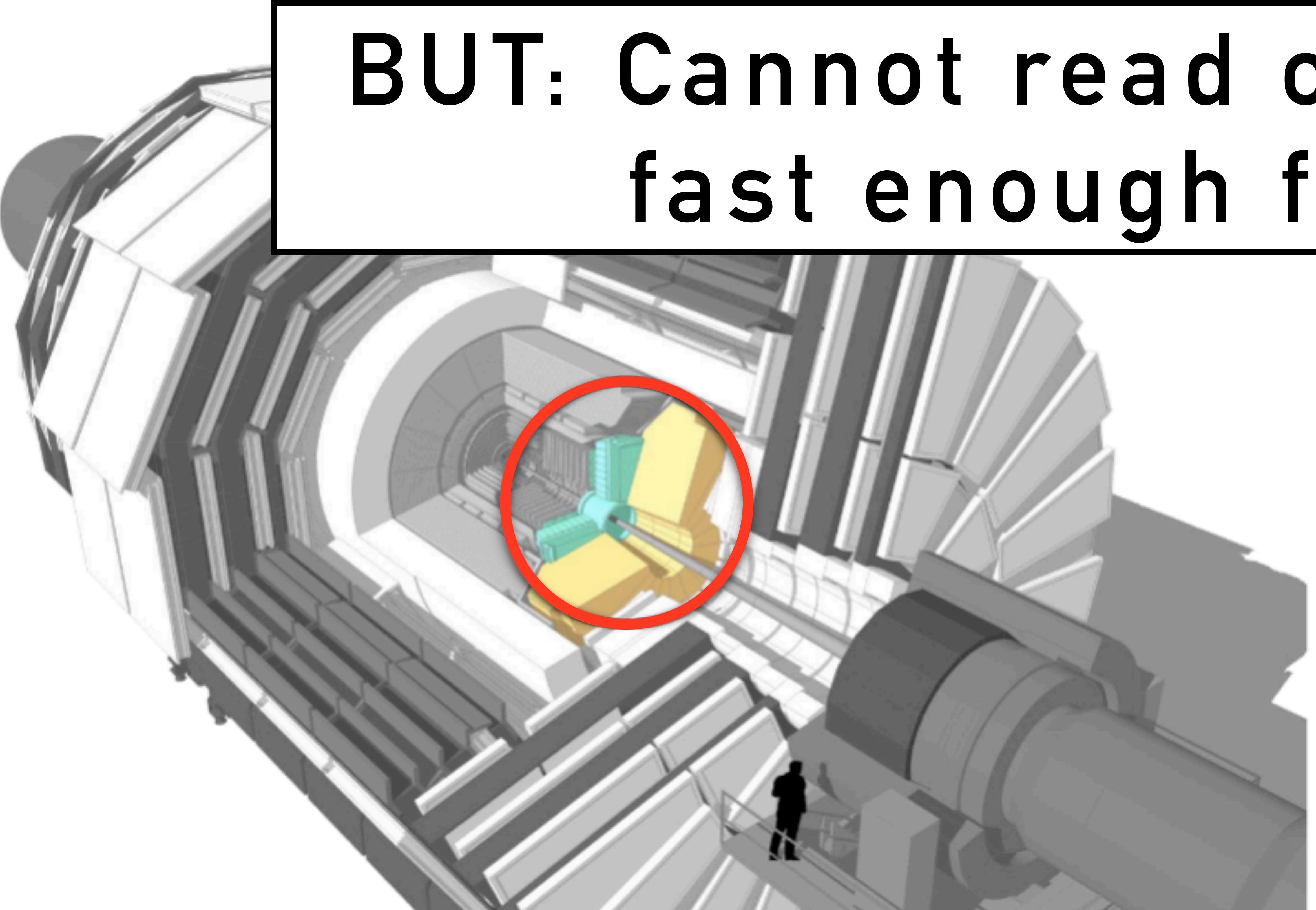
ML for compression

CMS High Granularity calorimeter

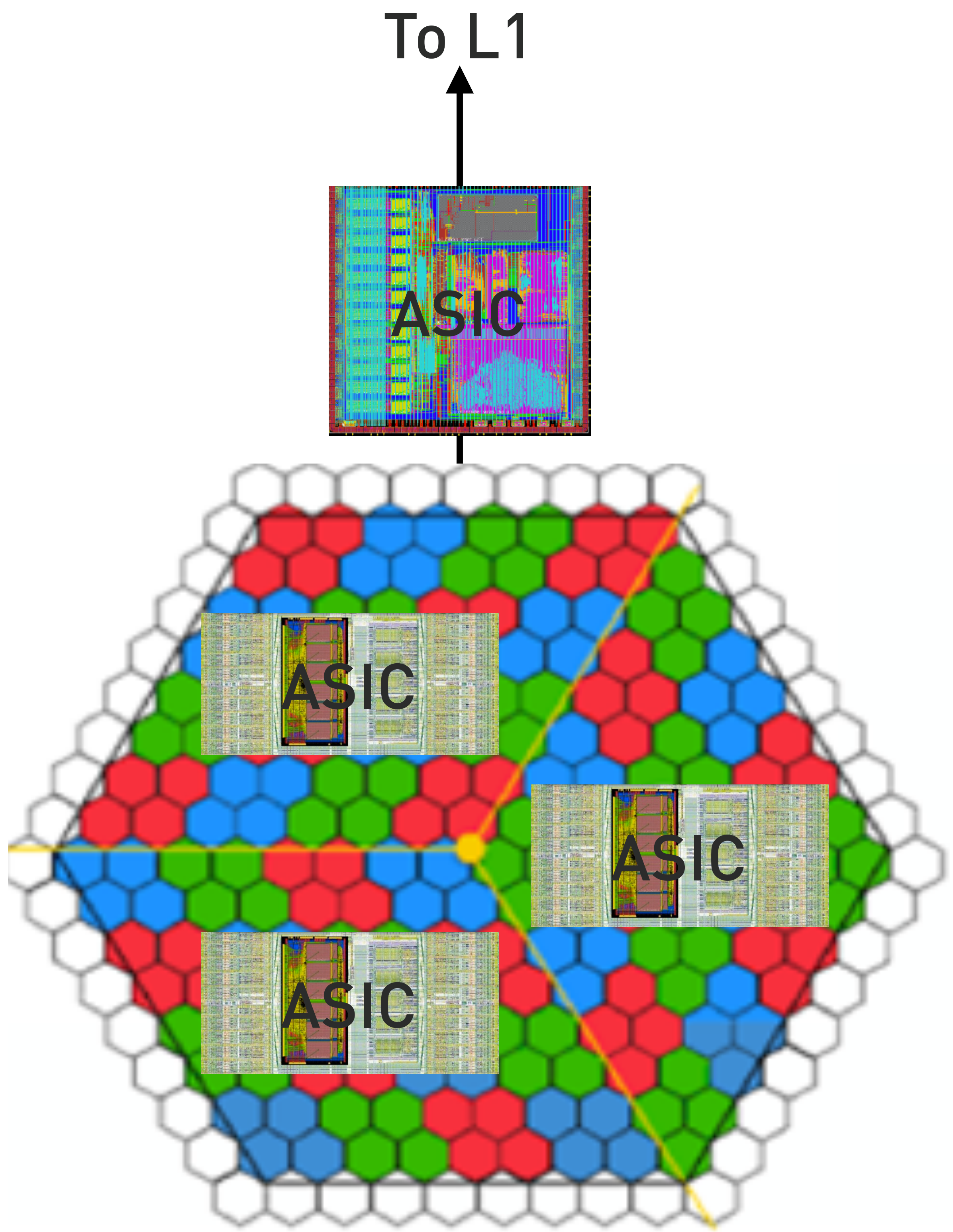
- 6.5 million readout channels, 50 layers



BUT: Cannot read out all these channels fast enough for L1 to trigger!



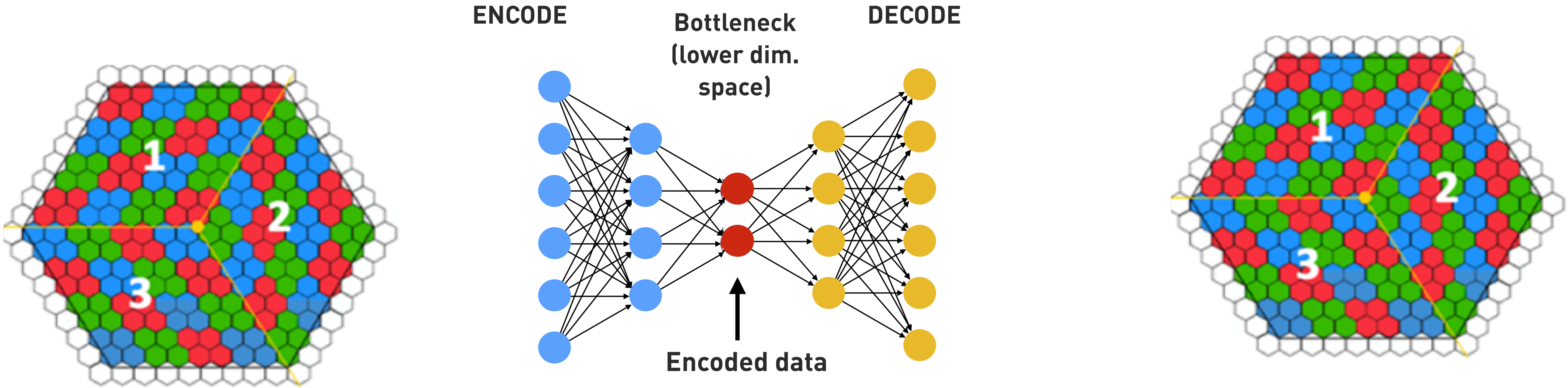
ML for compression



Must compress ON DETECTOR

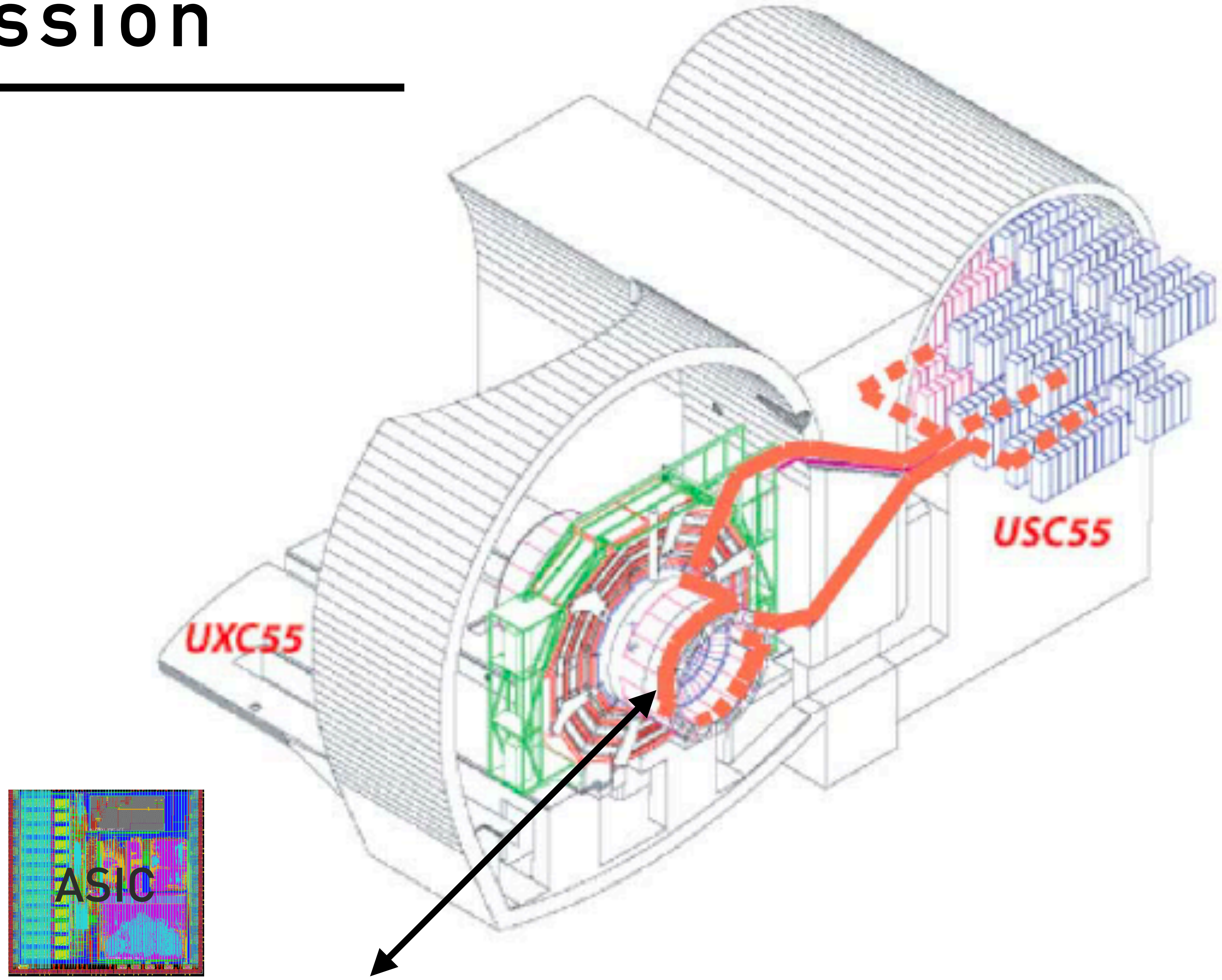
- High radiation
- Cooled to -30 → low power
- 1.5 μs latency

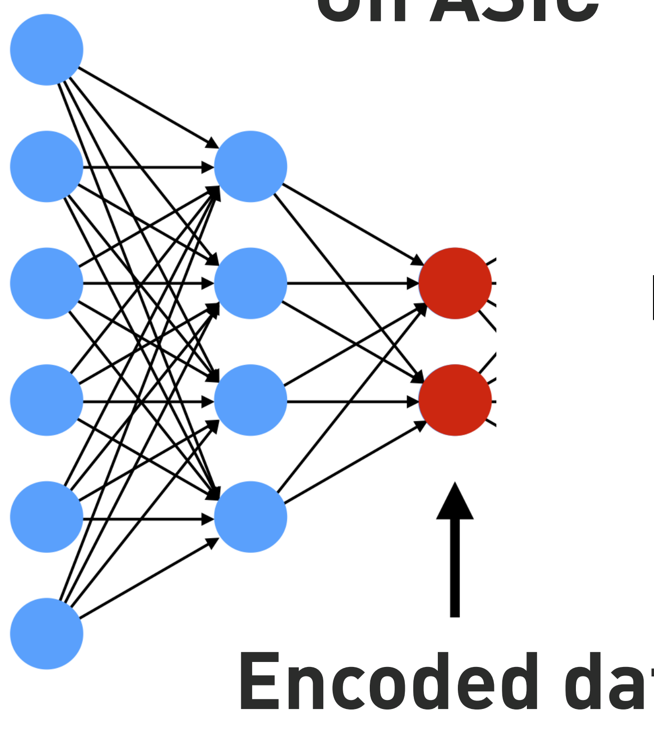
ML for compression

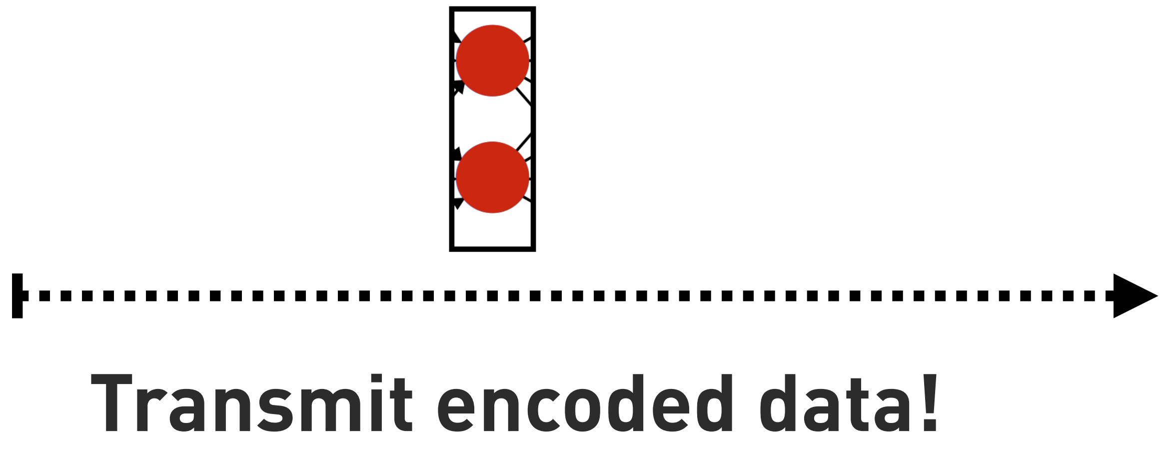


Variational Autoencoder

ML for compression



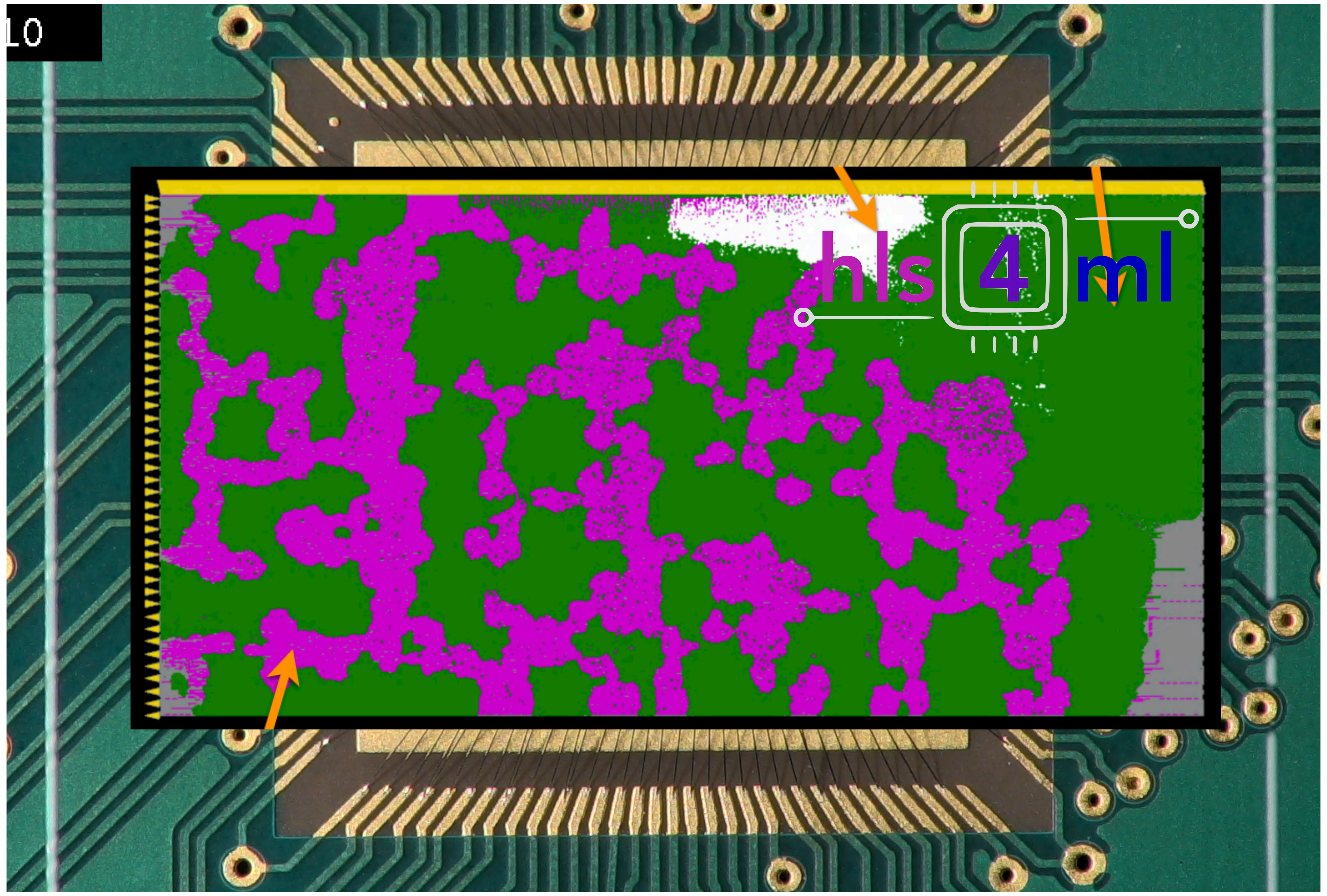


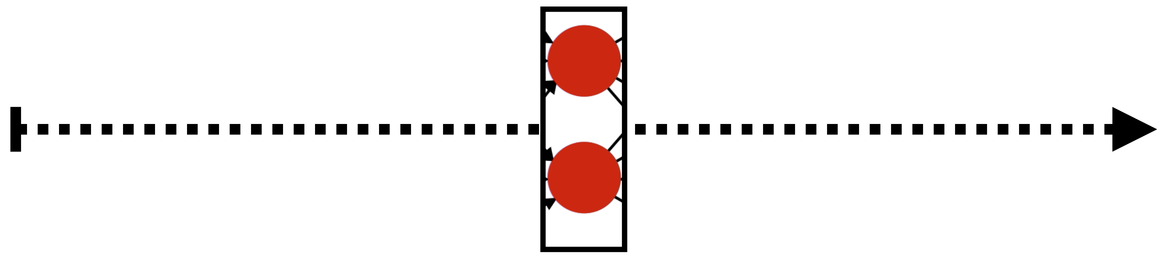


ML for compression

ML for compression

48 trigger cells



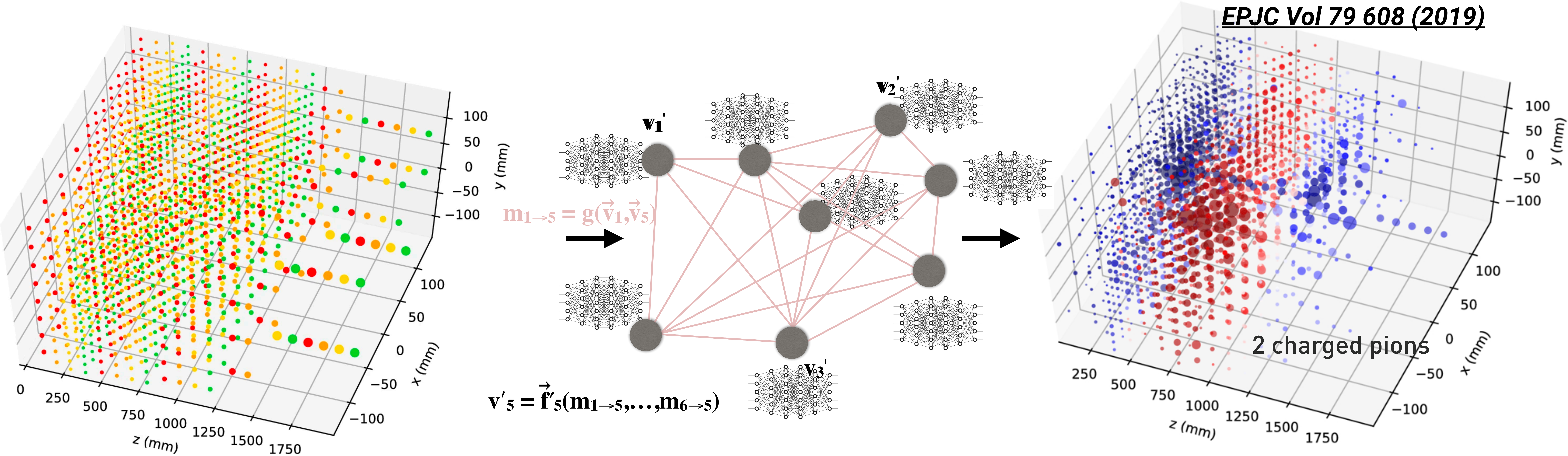


16 ReLU activated nodes

ML for reconstruction

On FPGA: 3.5 μs to cluster energy deposits

ML for reconstruction



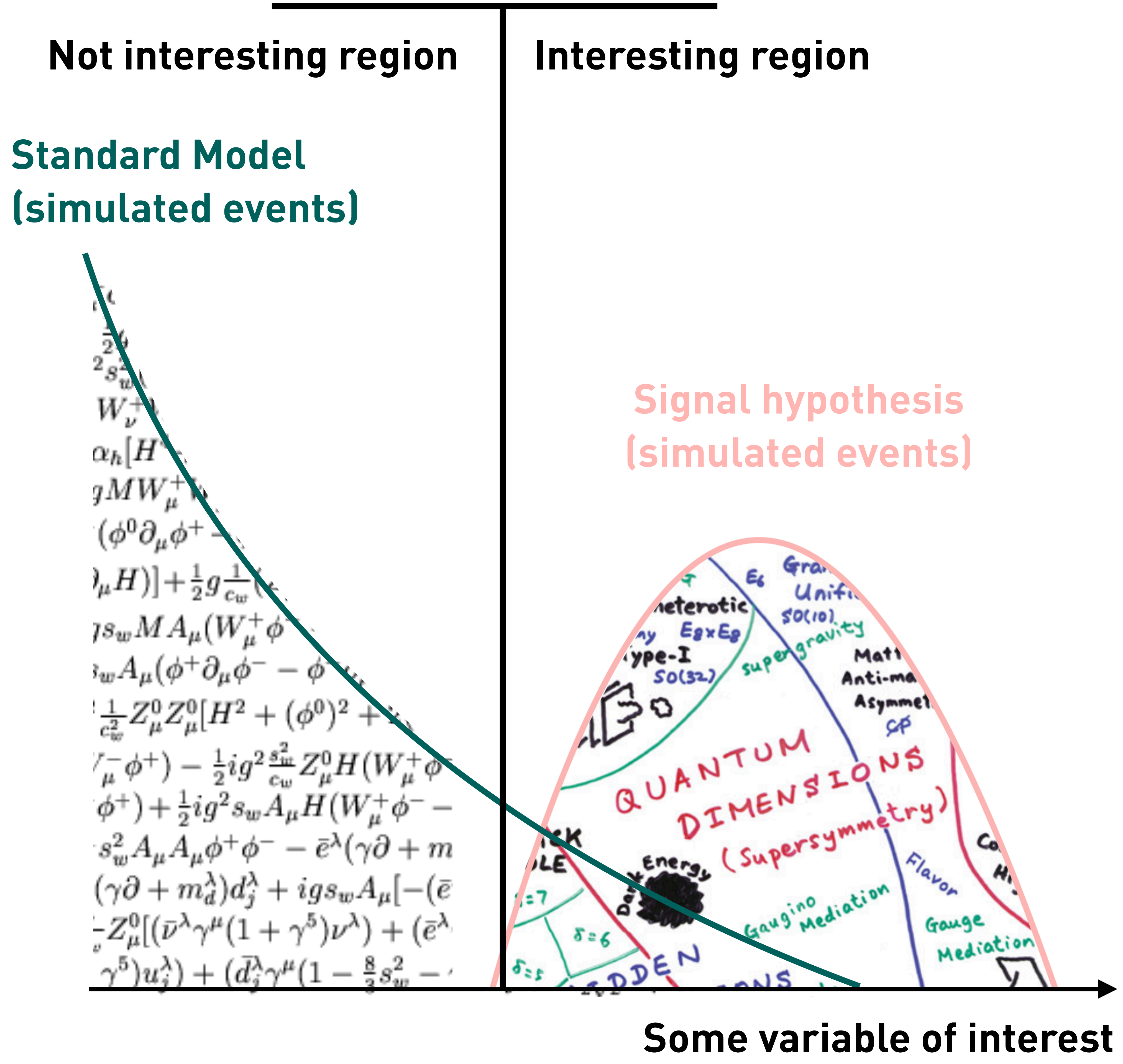
On FPGA: 3.5 μs to cluster energy deposits

- Graph Neural Networks (GarNet/GravNet) for fast clustering of irregular geometry detectors

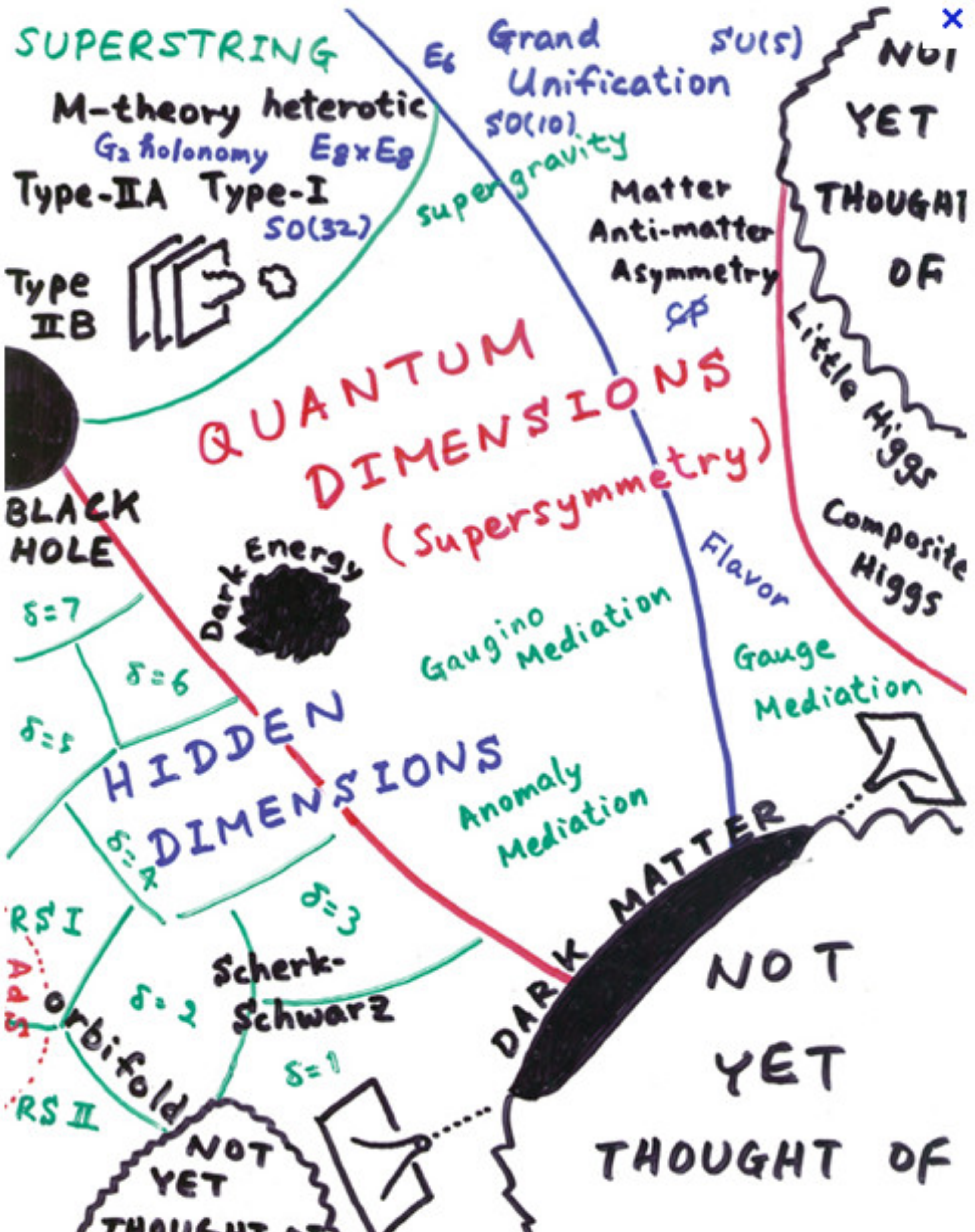
Searches for new particles at LHC

Searches for new particles at LHC

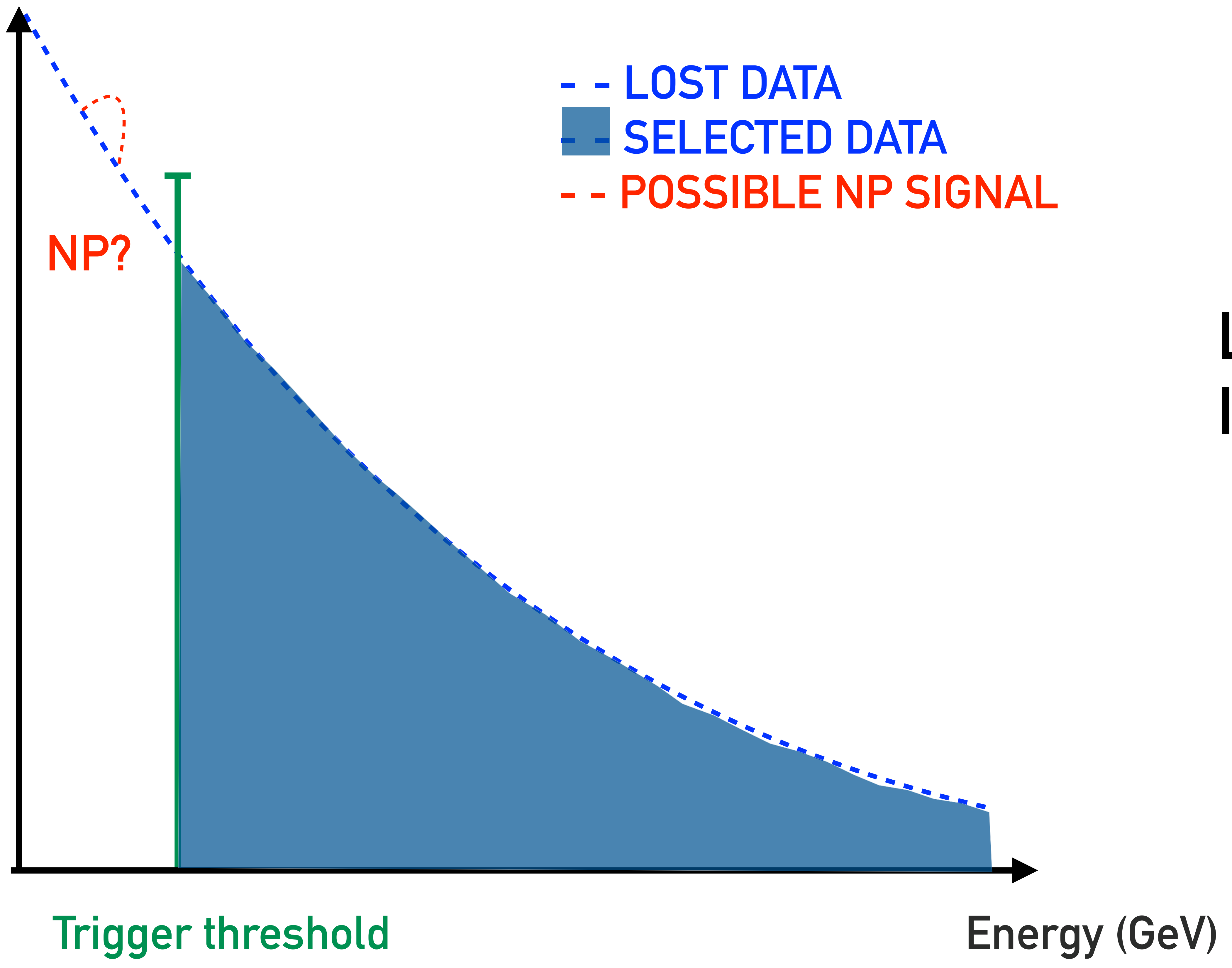
Bias in particle physics



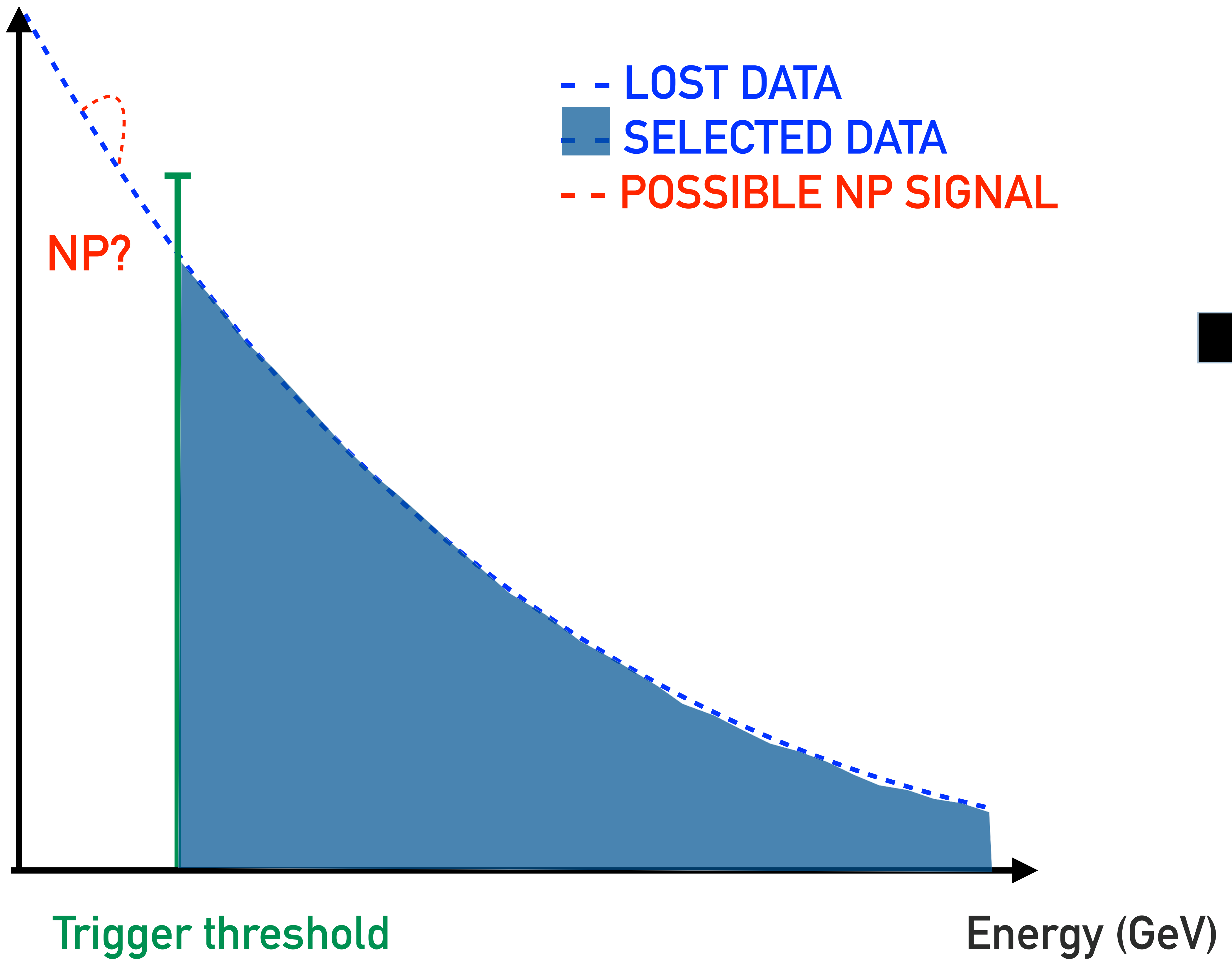
Need to exploit the full capabilities of the LHC and be more generic!



Limitations of current trigger

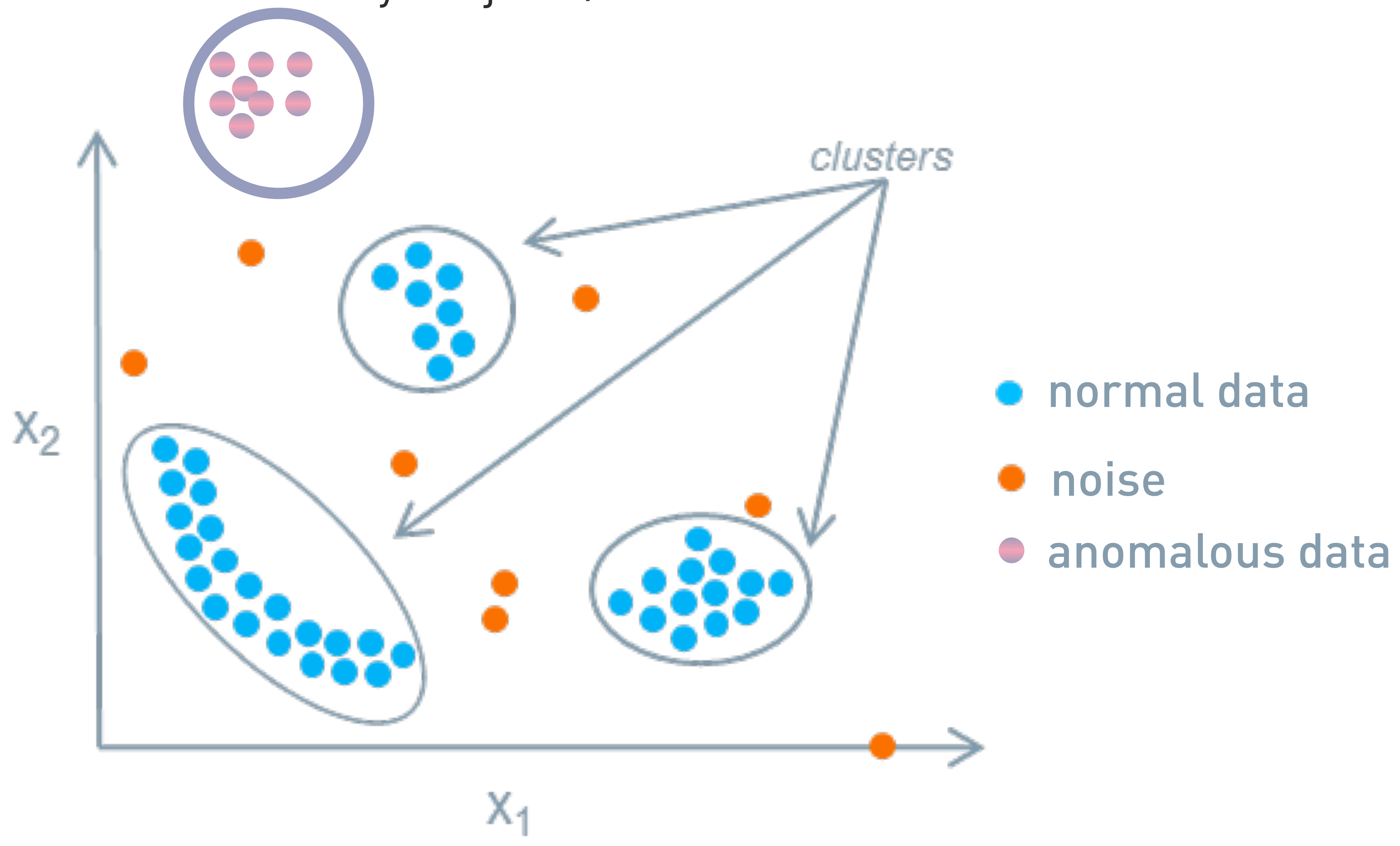


Level-1 rejects >99% of events!
Is there a smarter way to select?



Look at **data** rather than defining signal hypothesis a priori

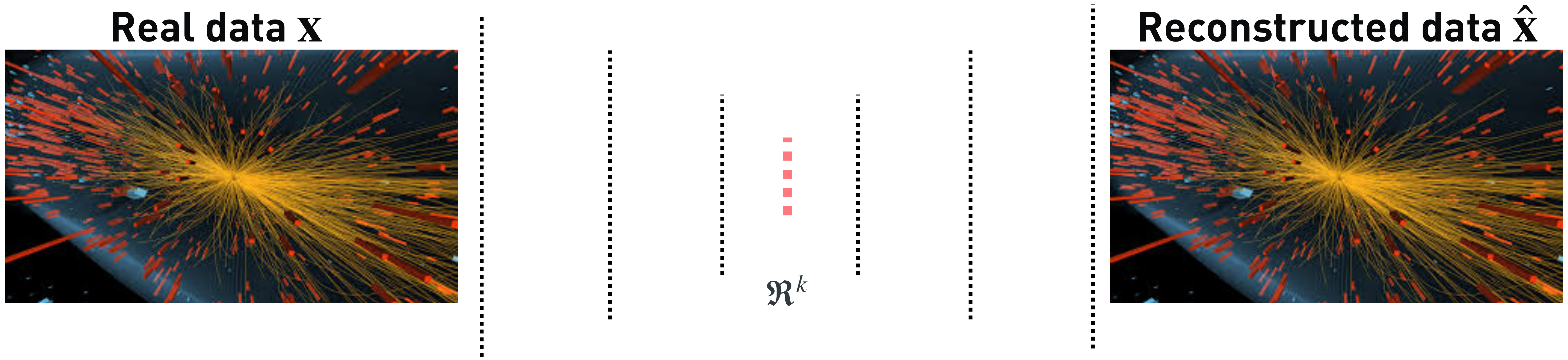
- Can we "classify" objects/events?



ML for anomaly detection

Autoencoders: Learns from data

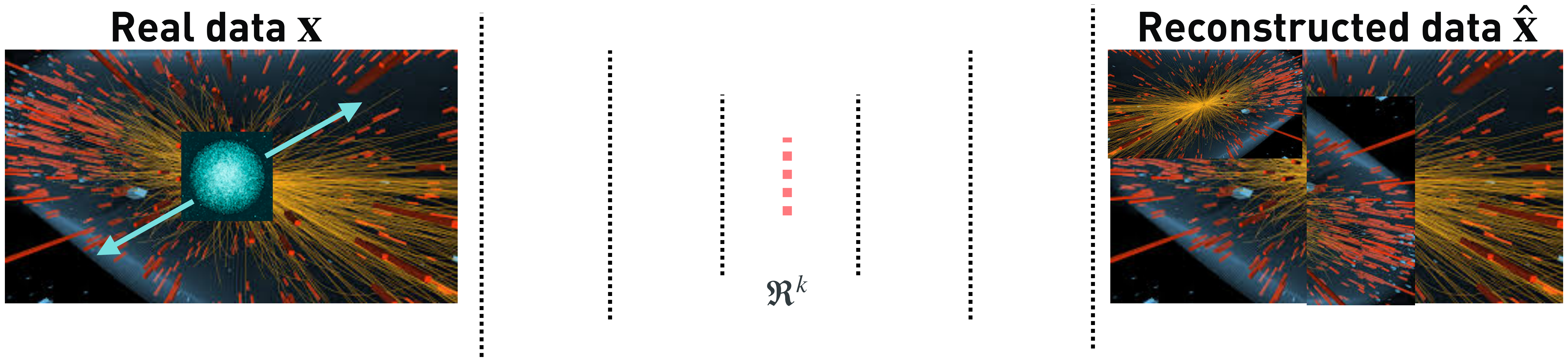
- Trains unsupervised
- Learns to compress, then reconstruct data
- Often used for financial fraud detection
  - Low rate of anomalous events versus high rate "background"



ML for anomaly detection

Autoencoders: Learns from data

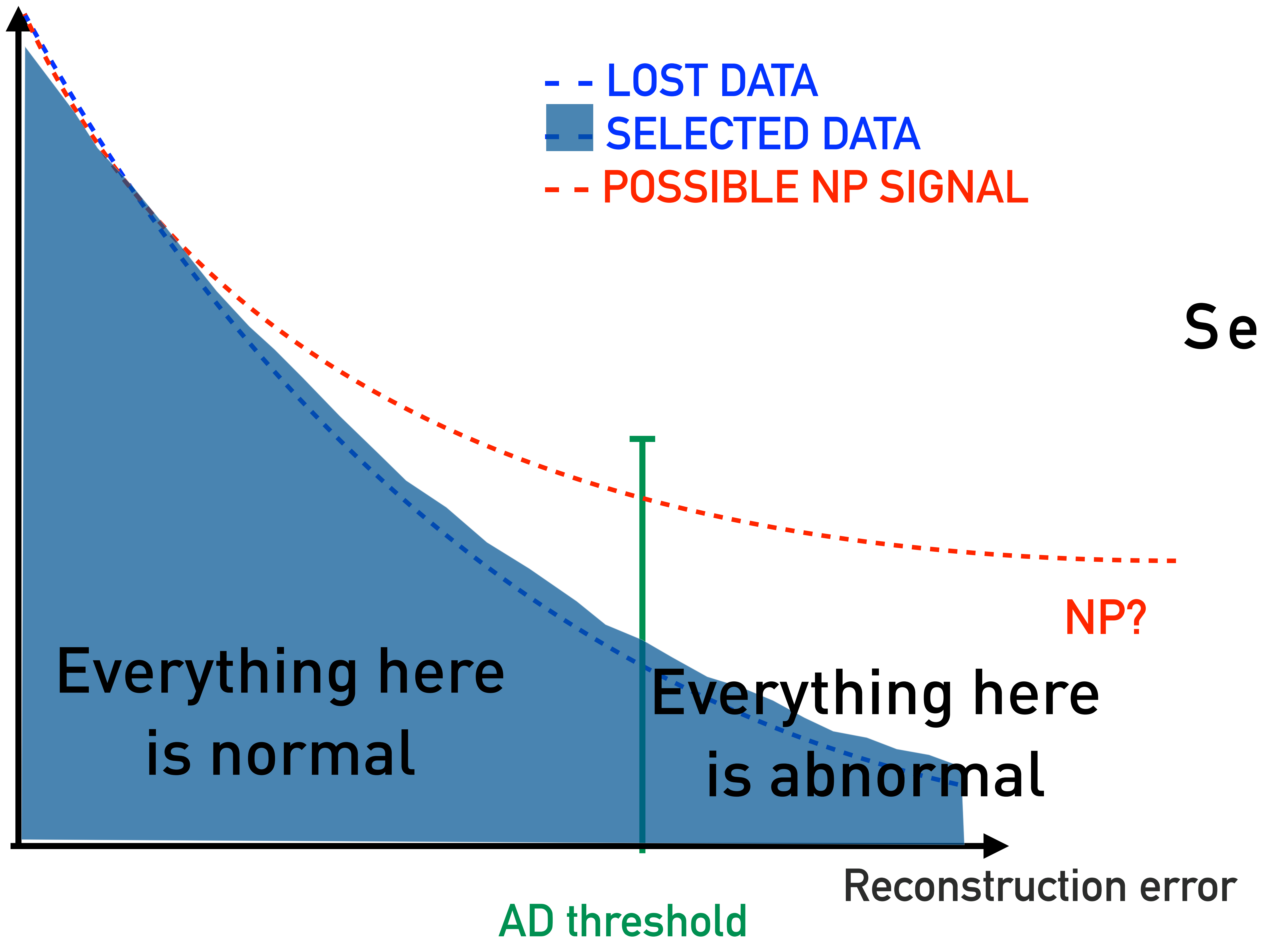
- Trains unsupervised
- Learns to compress, then reconstruct data
- Often used for financial fraud detection
  - Low rate of anomalous events versus high rate "background"



- Difference $\mathbf{x}$-$\hat{\mathbf{x}}$ defines "degree of abnormality"

Nature Machine Intelligence 4, 154 (2022)

ML for anomaly detection



Select based on degree of abnormality!

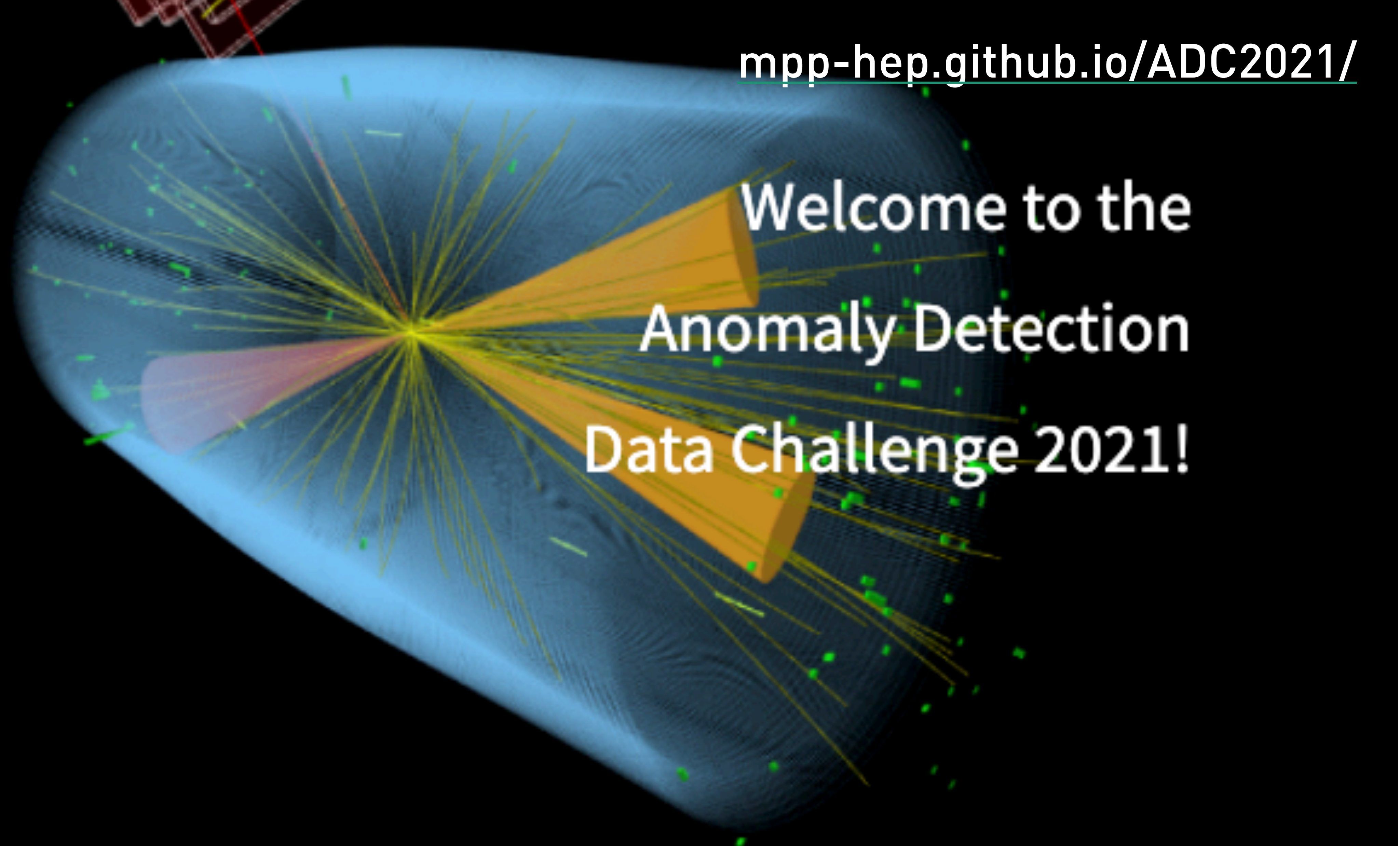
ADC 2021

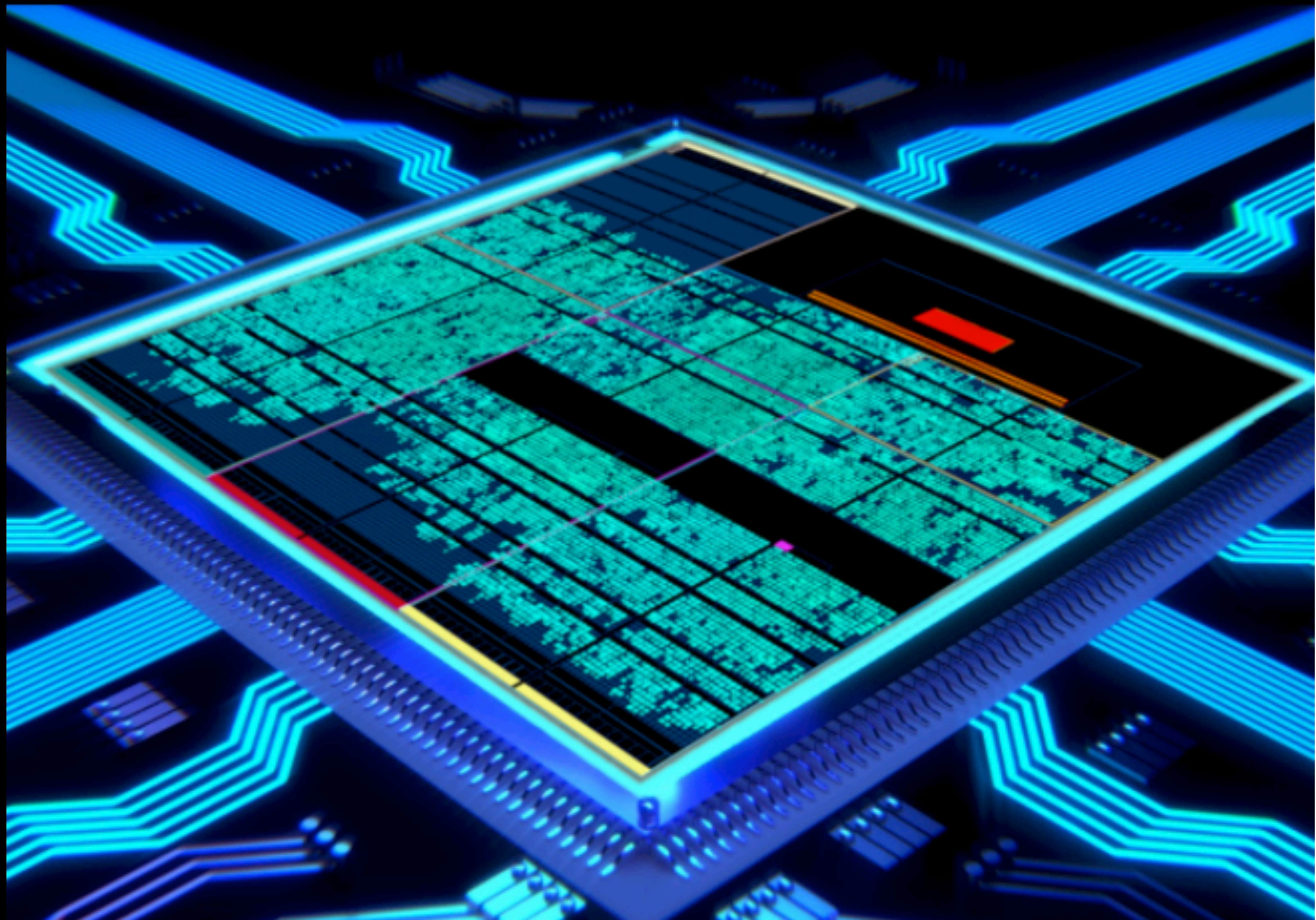
Data challenge on real-time anomaly detection

- Dataset: Nature Scientific Data (2022) 9:118

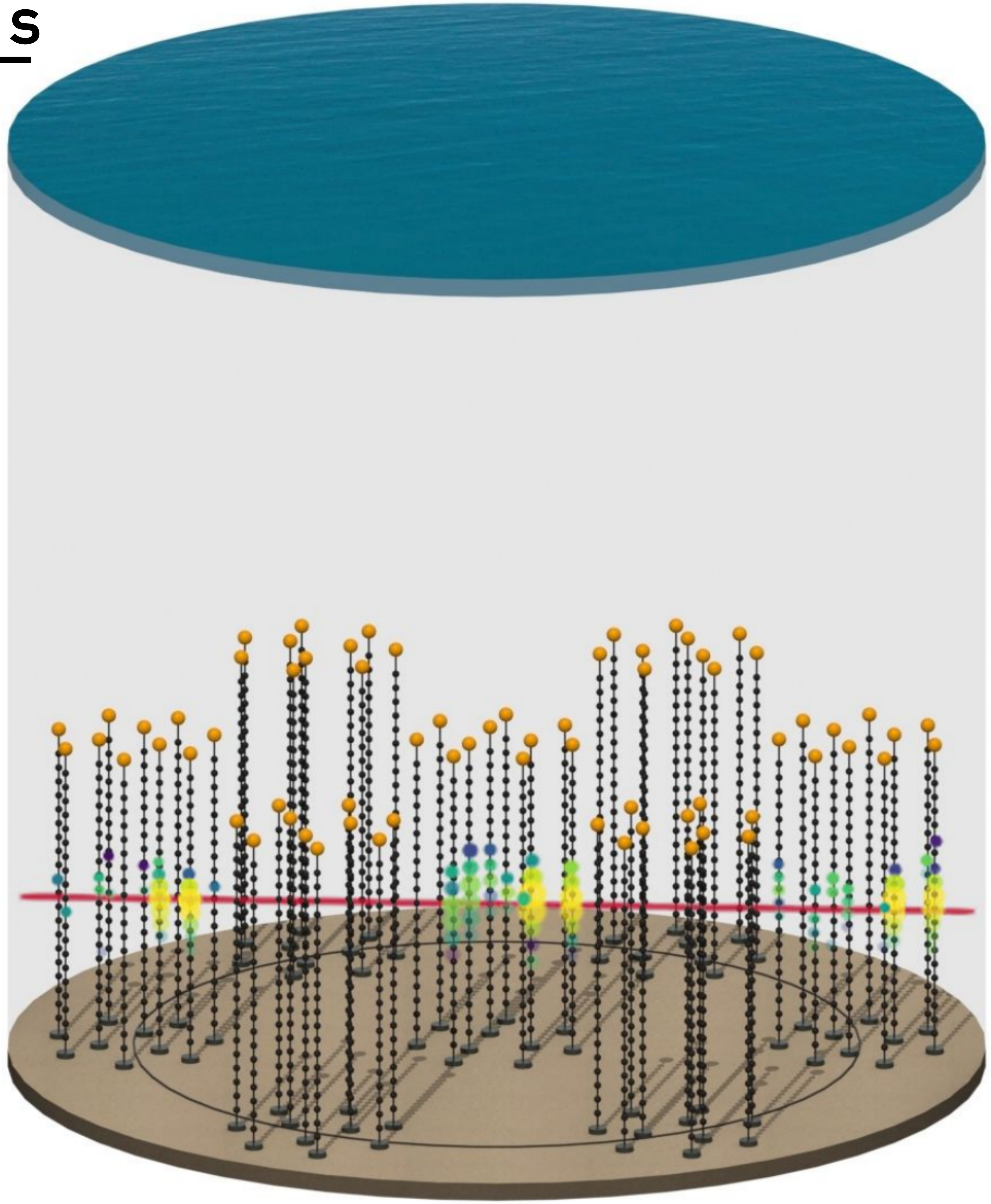
Tutorial: Anomaly detection on FPGA with hls4ml

Help us find new physics!

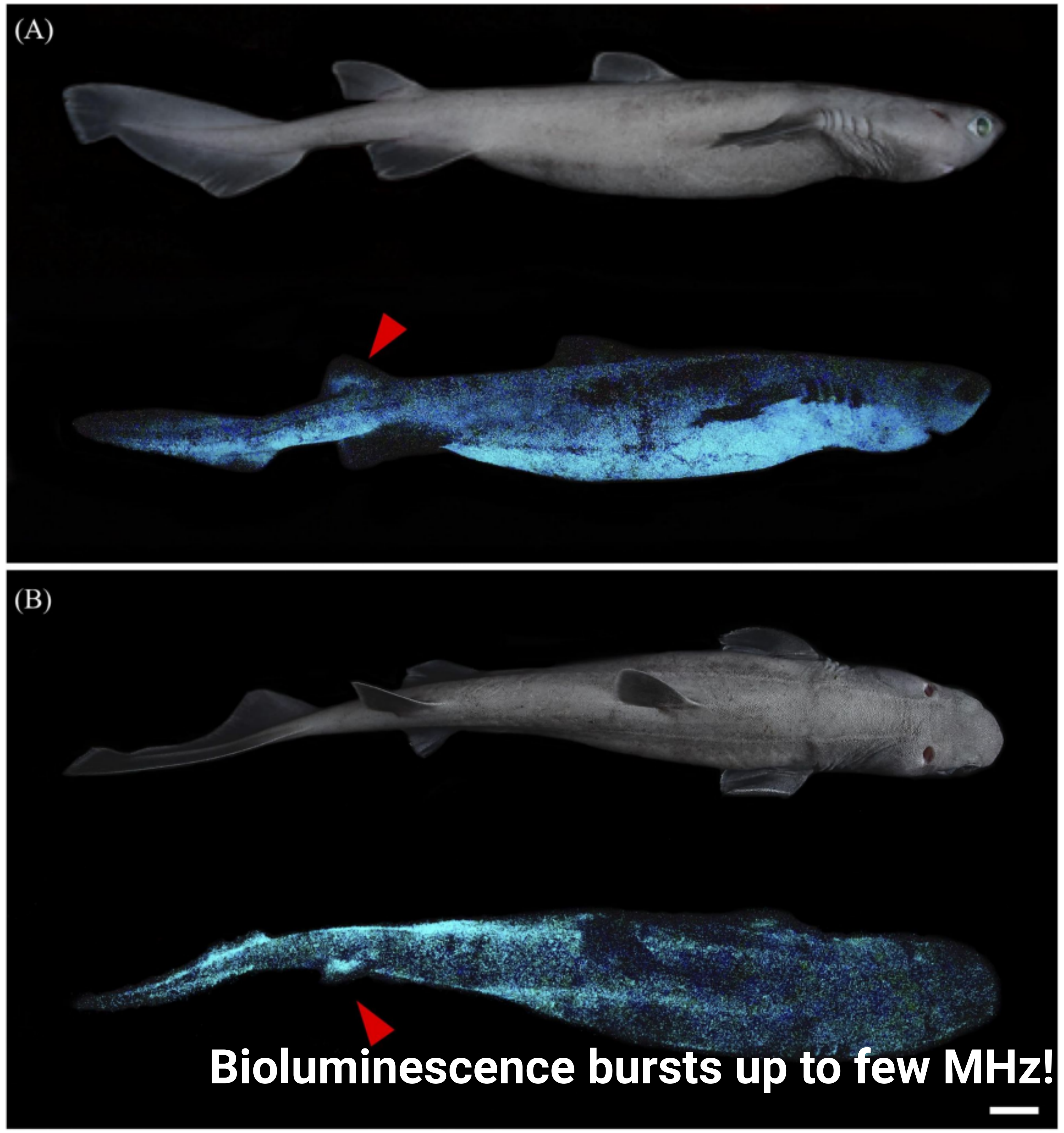


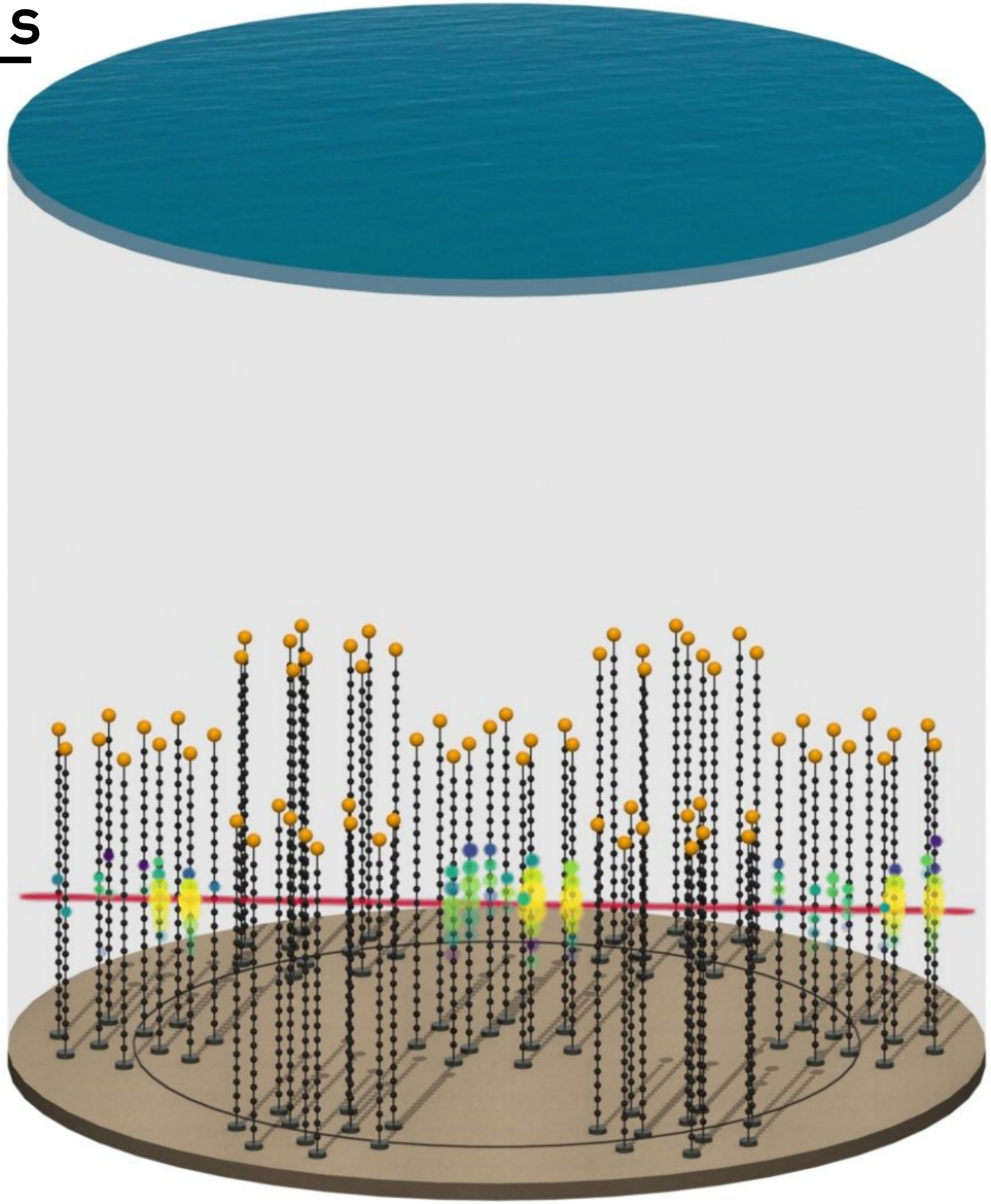


Real-time ML in other experiments

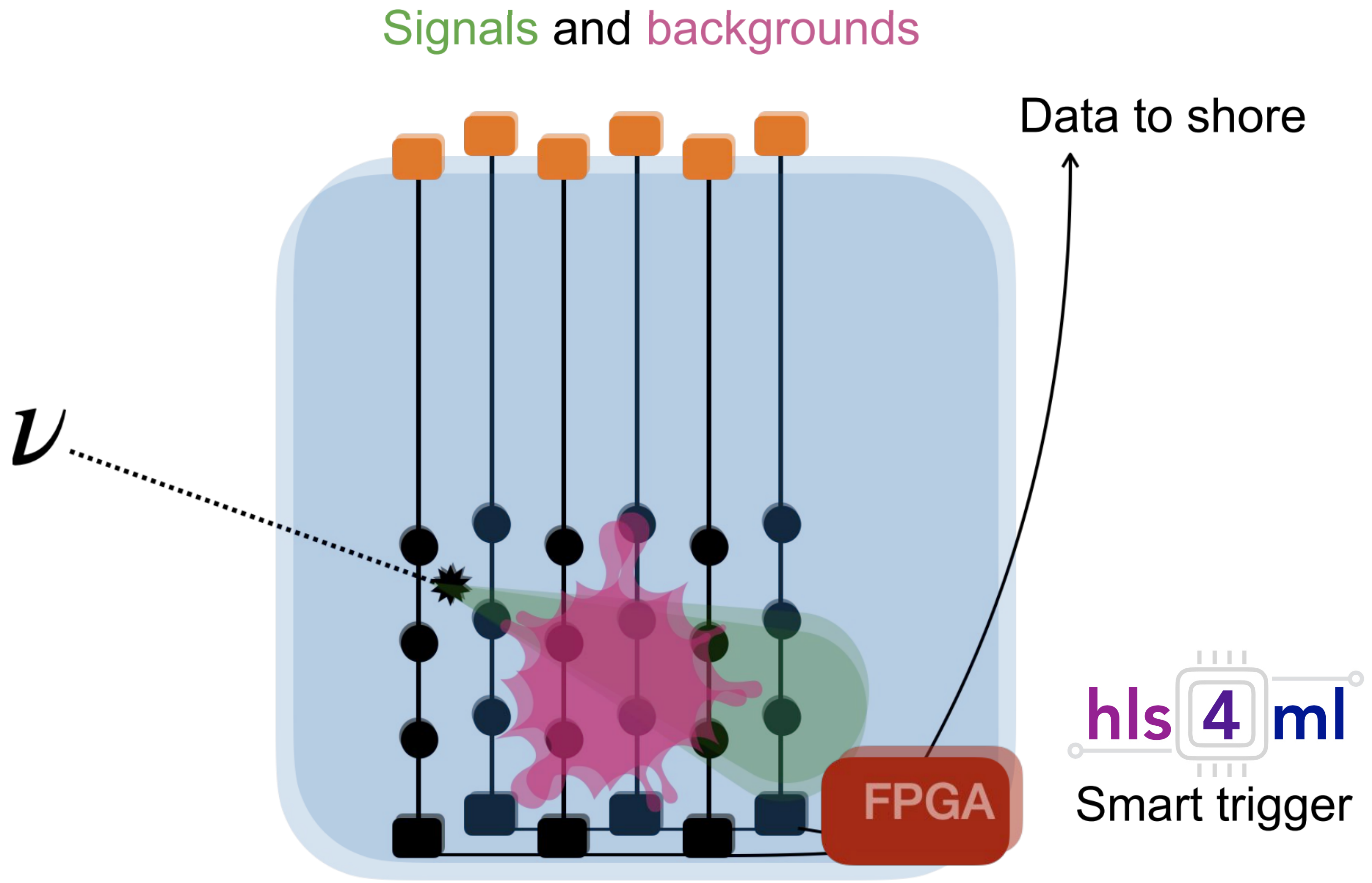


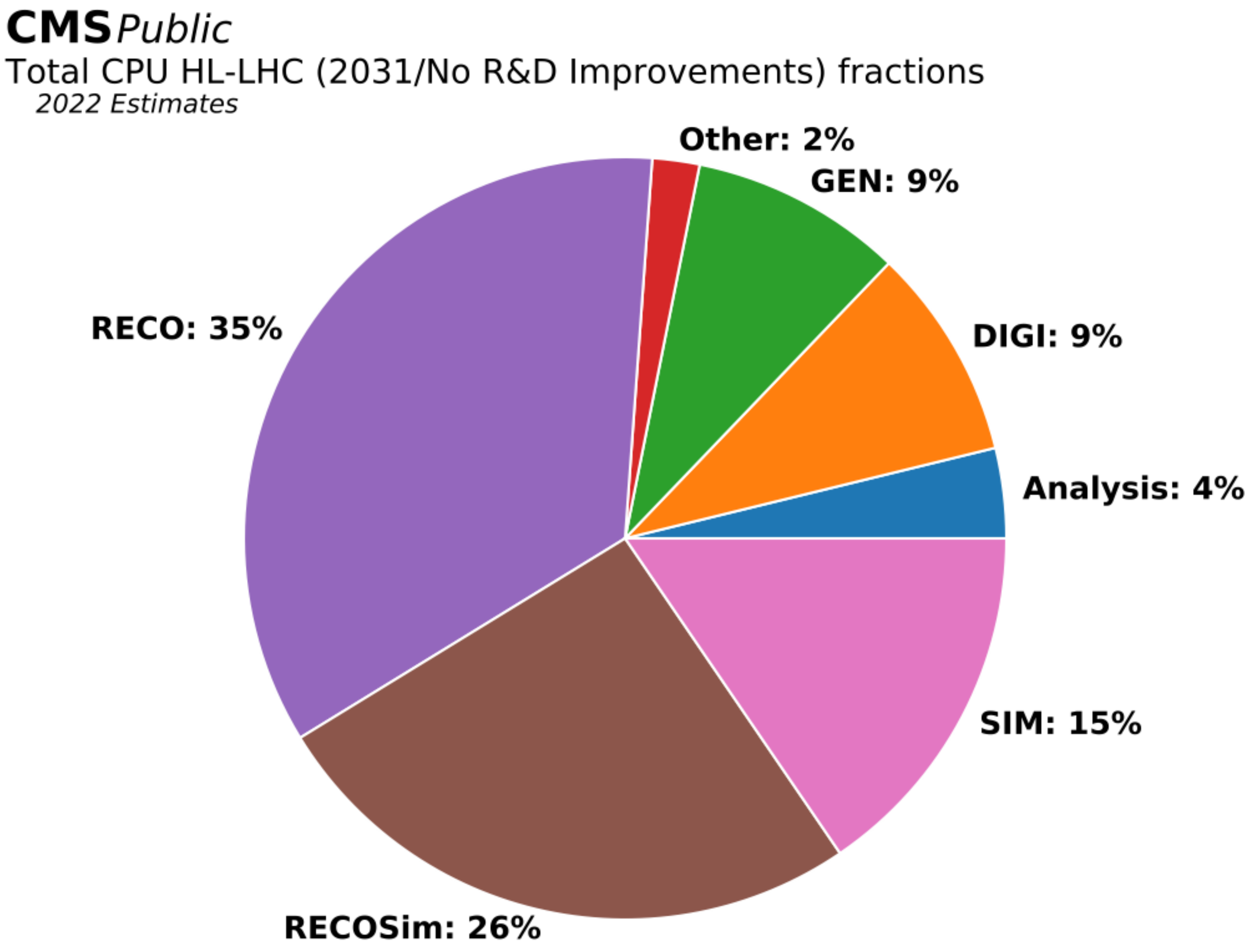
Real-time ML in other experiments



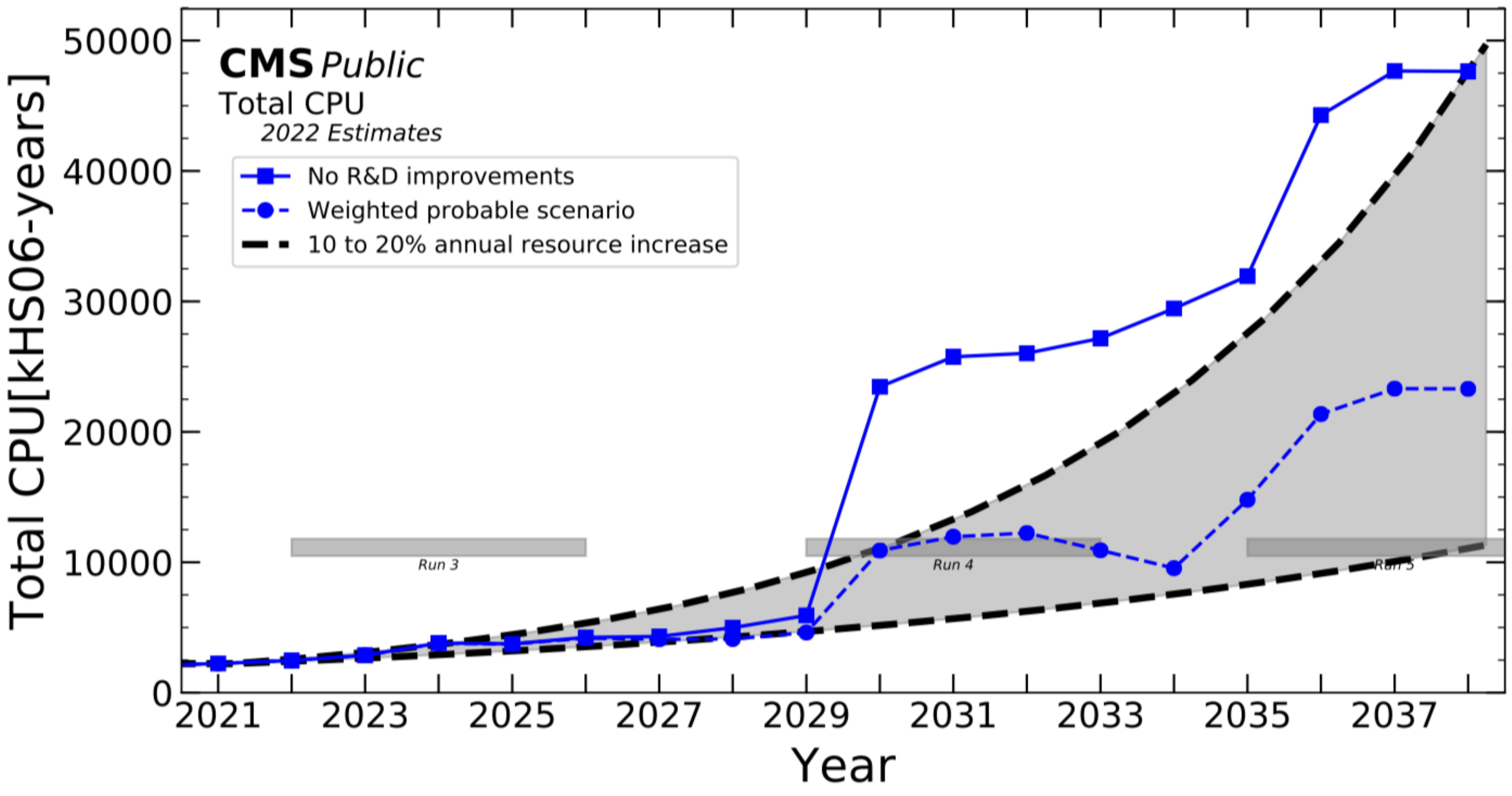


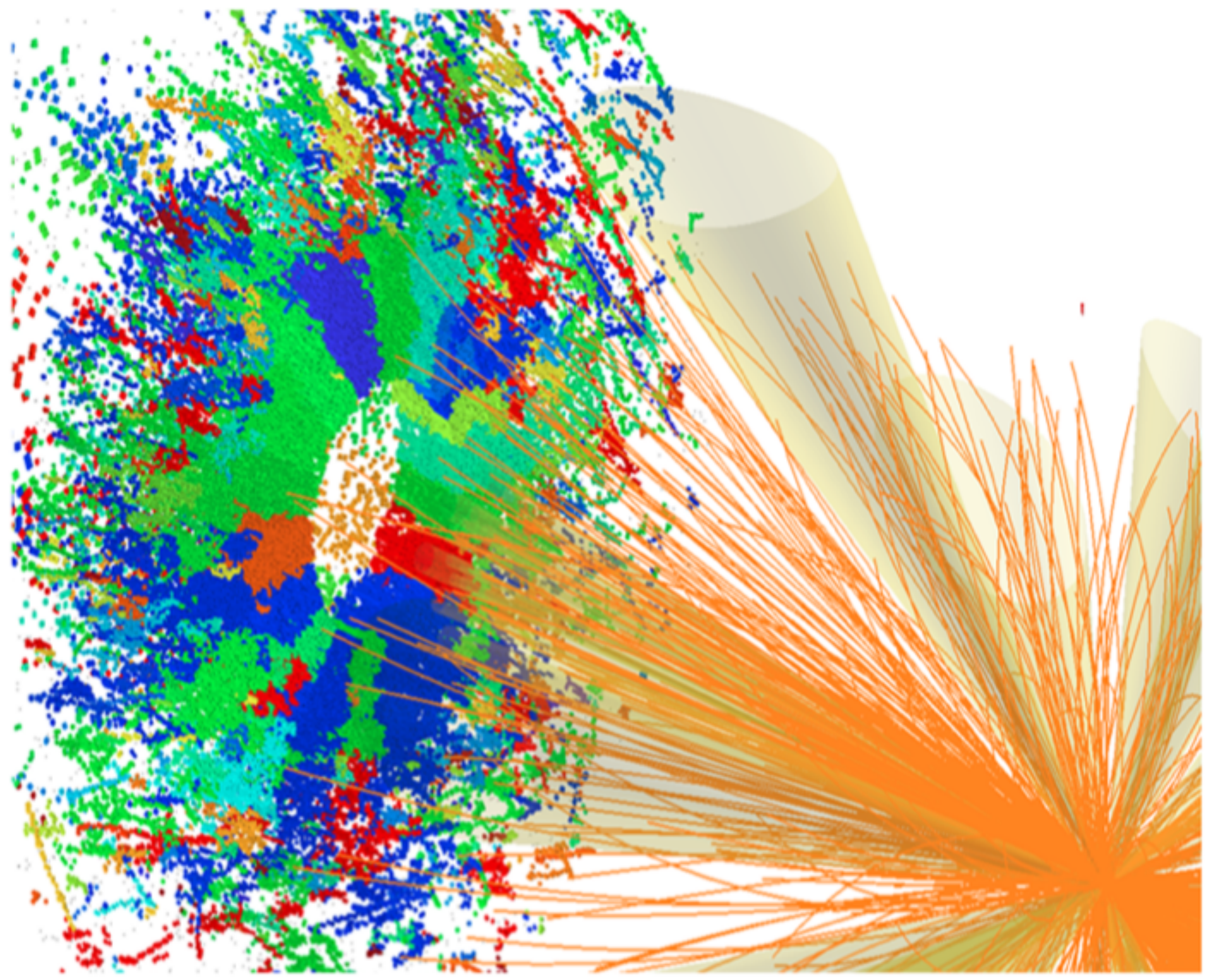
Real-time ML in other experiments




CMS *Public*
Total CPU HL-LHC (2031/No R&D Improvements) fractions
2022 Estimates
Other: 2%
GEN: 9%
DIGI: 9%
Analysis: 4%
SIM: 15%
RECOSim: 26%
RECO: 35%

CMS Offline Computing Results





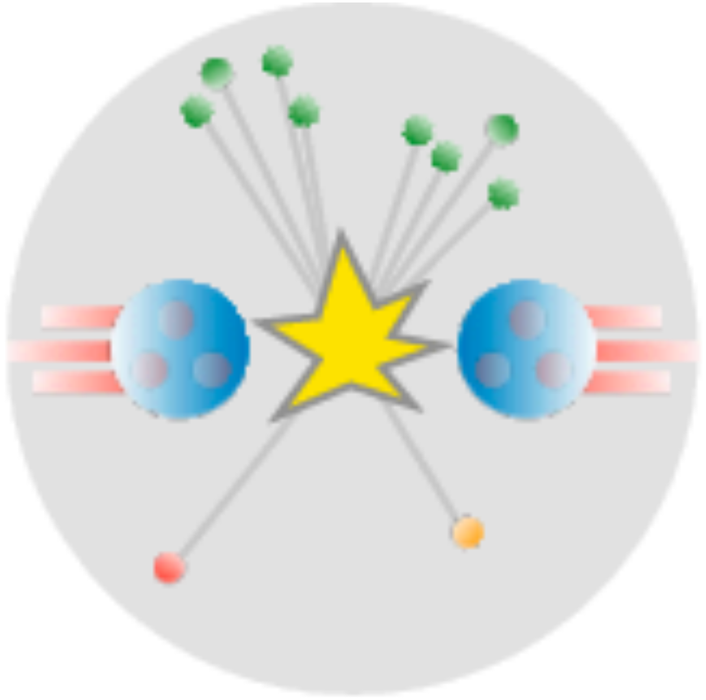
HL-LHC, Simulation of CMS HGCAL with 140 PU


$O(10)$
$O(10^3)$
$O(10^{10})$
10^{-18}m
10^{-15}m
10^{-6}m
100m
2T

$O(10)$ $O(10^3)$ $O(10^{10})$

10^{-18}m 10^{-15}m 10^{-6}m 100m

GEN



pp collisions up to production of stable particles [Easy & Fast]

SIM

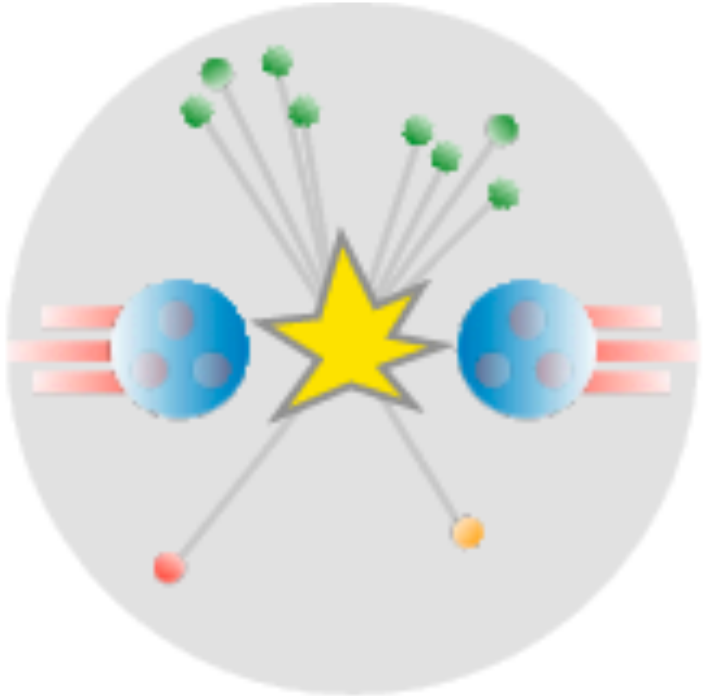
detector response simulation [Hard & Slow]

GEANT4
A SIMULATION TOOLKIT

DIGI+RECO

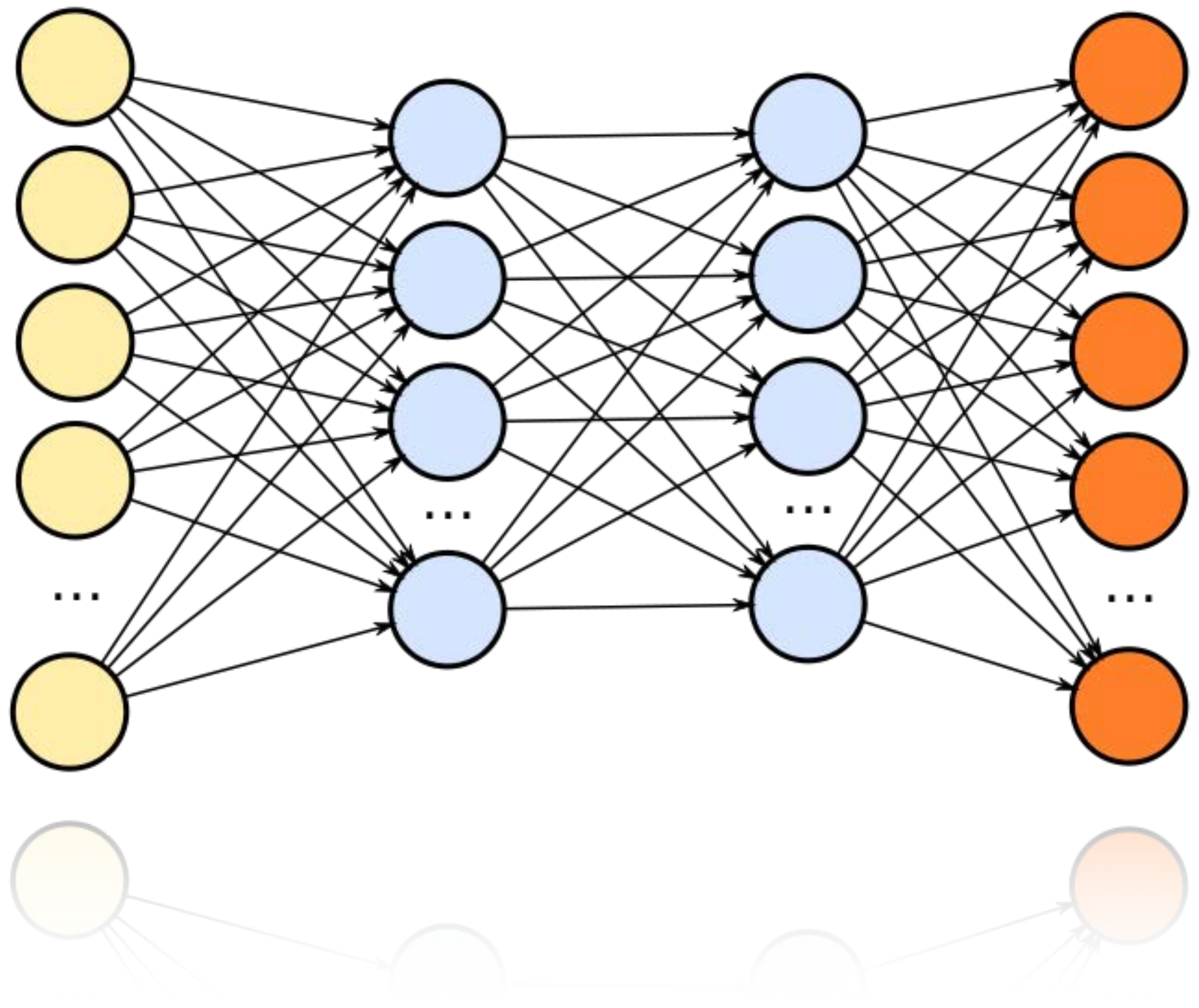
Energy deposits→digital signals→reconstructed by the reconstruction software [Hard & Slow]

GEN



pp collisions up to production of stable particles [Easy & Fast]

SIM



DIGI+RECO

Energy deposits→digital signals→reconstructed by the reconstruction software [Hard & Slow]

GEN | SIM | DIGI+RECO

GEANT4 → 10^{10} events **SLOW but ACCURATE**

GEANT4 → 10^{5} events → Surrogate model → 10^{10} events

(GAN, VAE, Normalizing Flow, …)
Learn underlying distribution of GEANT4 events

FAST and ACCURATE?

ML methods can provide fast and accurate "surrogate models" for GEANT4 etc

p… pro… par…

…posits→digital …constructed by …uction software [Hard & Slow]

T

I'm a bit late with my presentation, could you summarise the latest progress in Machine Learning for Particle Physics?

Sure! In recent years, there has been significant progress in the application of machine learning techniques to particle physics. Some key areas where machine learning has made significant contributions include:

1. Classification of particles: Machine learning algorithms, particularly deep learning algorithms, have been successfully used to classify different types of particles based on their properties and characteristics. For example, they can be used to identify and distinguish between different types of hadrons (such as protons and neutrons) or leptons (such as electrons and muons).
2. Event reconstruction: Machine learning algorithms have also been used to improve the accuracy of event reconstruction in particle physics experiments. Event reconstruction refers to the process of reconstructing the collision events that occurred in a particle accelerator from the raw data collected by detectors. Machine learning algorithms can be used to identify patterns and correlations in the data that can help scientists more accurately reconstruct the events that occurred.
3. Particle identification: Machine learning algorithms have been used to identify and classify particles based on their properties and characteristics. This can be particularly useful in the context of high-energy particle colliders, where large numbers of particles are produced in each collision event.
4. Simulation and modeling: Machine learning algorithms have also been used to improve the accuracy of simulations and models in particle physics. For example, they have been used to model the interactions between particles and the underlying physical processes that take place during collisions.
5. Data analysis and interpretation: Machine learning algorithms have been used to analyze and interpret large datasets generated by particle physics experiments. For example, they have been used to identify patterns and correlations in the data that can help scientists better understand the underlying physical processes at play.

Overall, machine learning has made significant contributions to particle physics and is expected to continue to play a major role in this field in the future.

processes that take place during collisions.

5. Data analysis and interpretation: Machine learning algorithms have been used to analyze and interpret large datasets generated by particle physics experiments. For example, they have been used to identify patterns and correlations in the data that can help scientists better understand the underlying physical processes at play.

Overall, machine learning has made significant contributions to particle physics and is expected to continue to play a major role in this field in the future.

Regenerate response

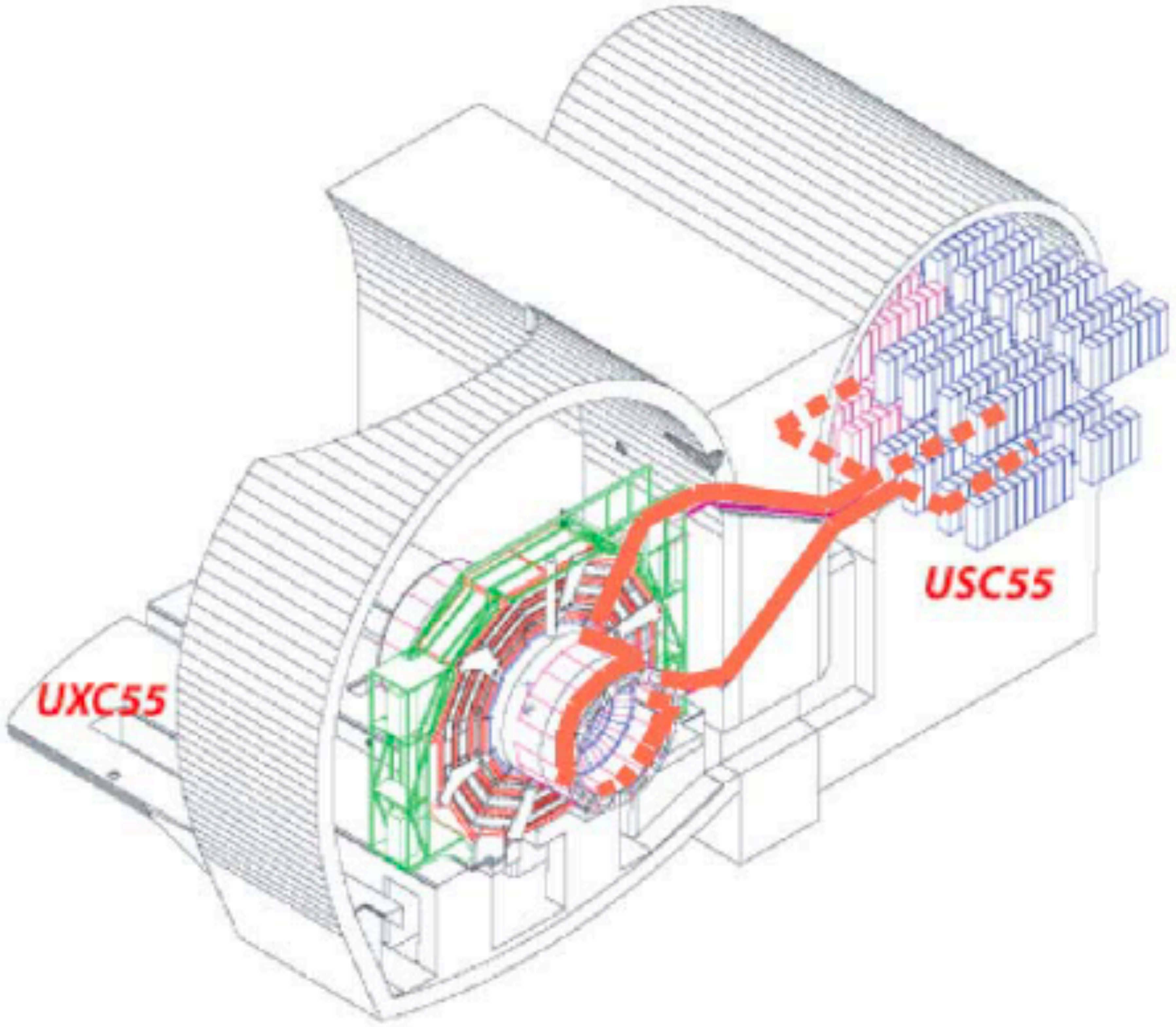
PT Dec 15 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us imp

Backup

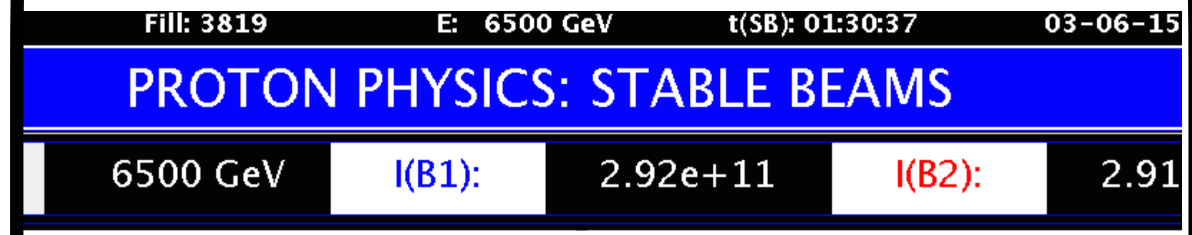
FPGAs as AI accelerators

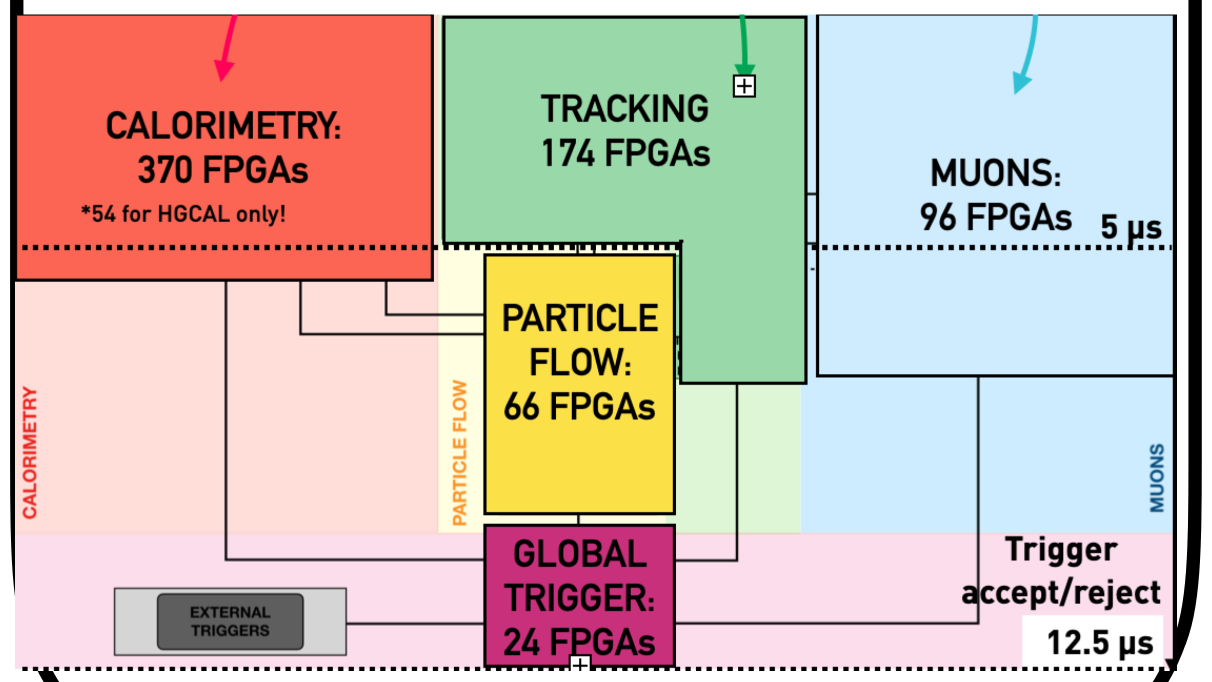
At LHC, DAQ FPGAs are idle ~50% of the time (no collisions)

- Could these be utilised for co-processing?
- Running AI inference for reconstruction tasks!

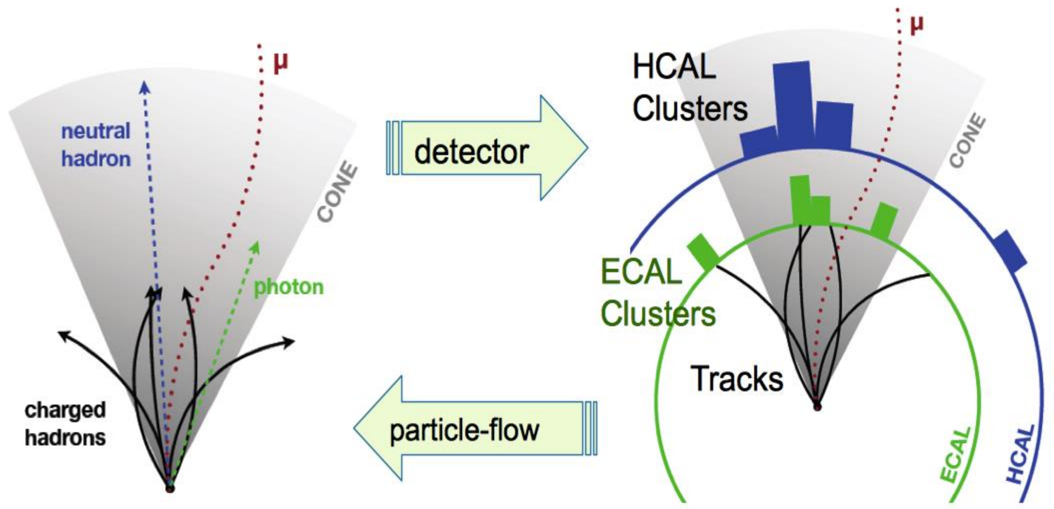


DAQ





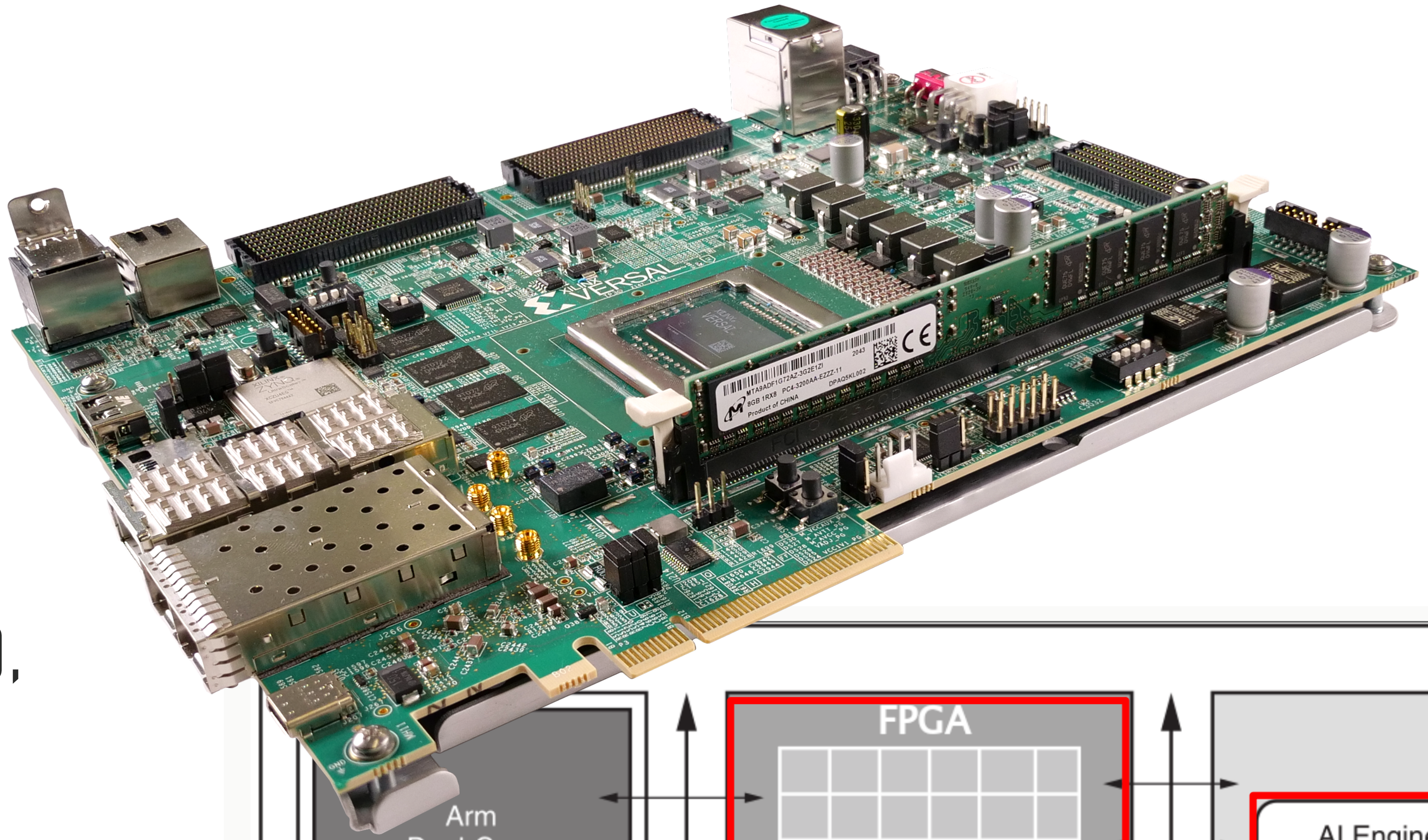
CO-PROCESSING



C. Beteta, I. Bezshyiko, N. Serra

Hardware: AI engines

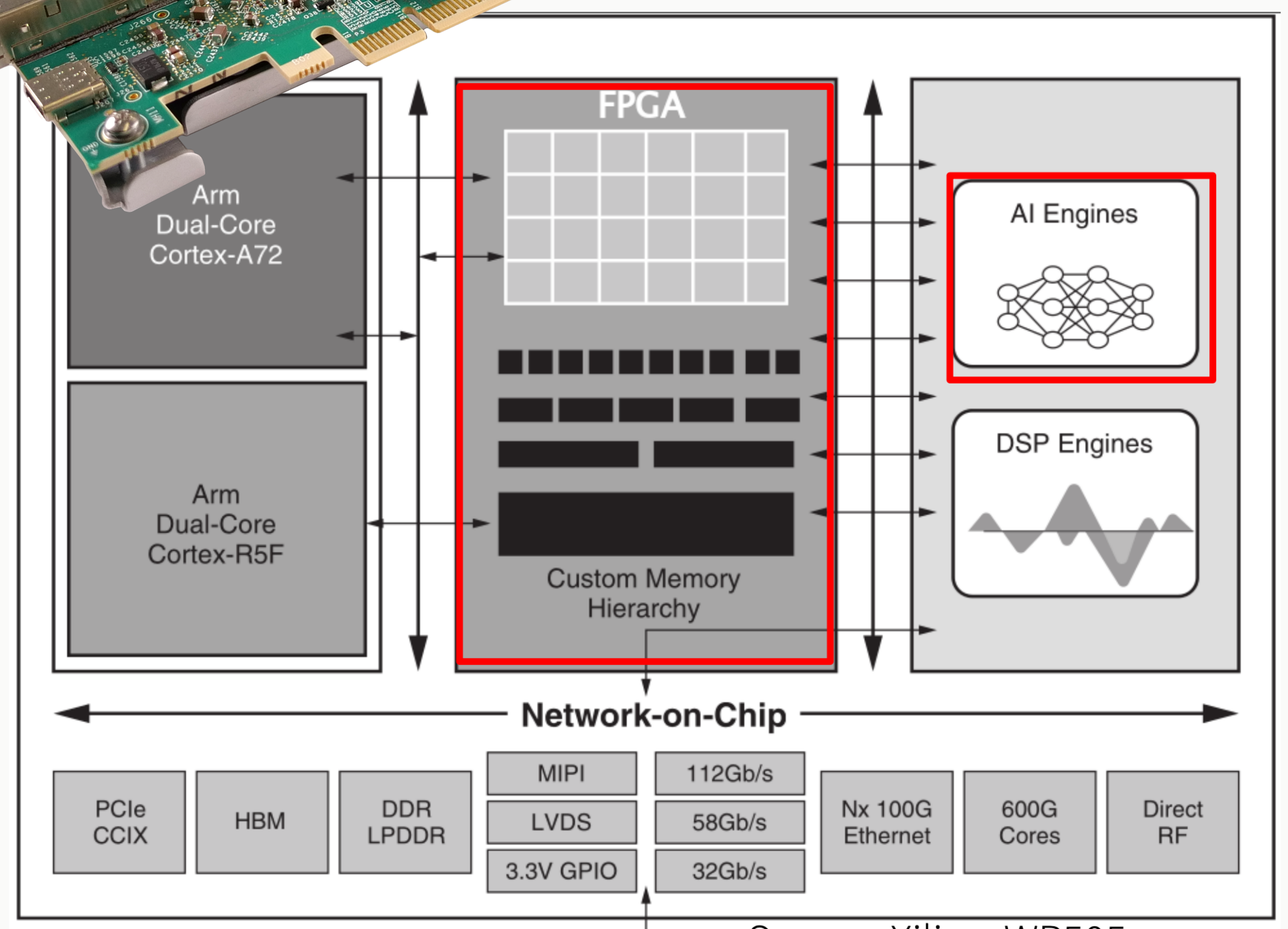
GNNs with Versal AI, P. Schwaebig



More and more dedicated AI processors on the market
- We should explore these to speed up our inferences!

Xilinx Versal AI processors
- Example Xilinx ACAP board: 400 AI processors, ~2M logic cells (FPGA), 2k DSPs, Arm CPU, Arm RPU
- Data can move back and forth between AI Engines and FPGA

Currently explored for real-time tracking in trigger application
- Interaction Network for pattern recognition (similar to **DeZoort et al**)
- Deployed on Xilinx Versal VC1902 ACAP



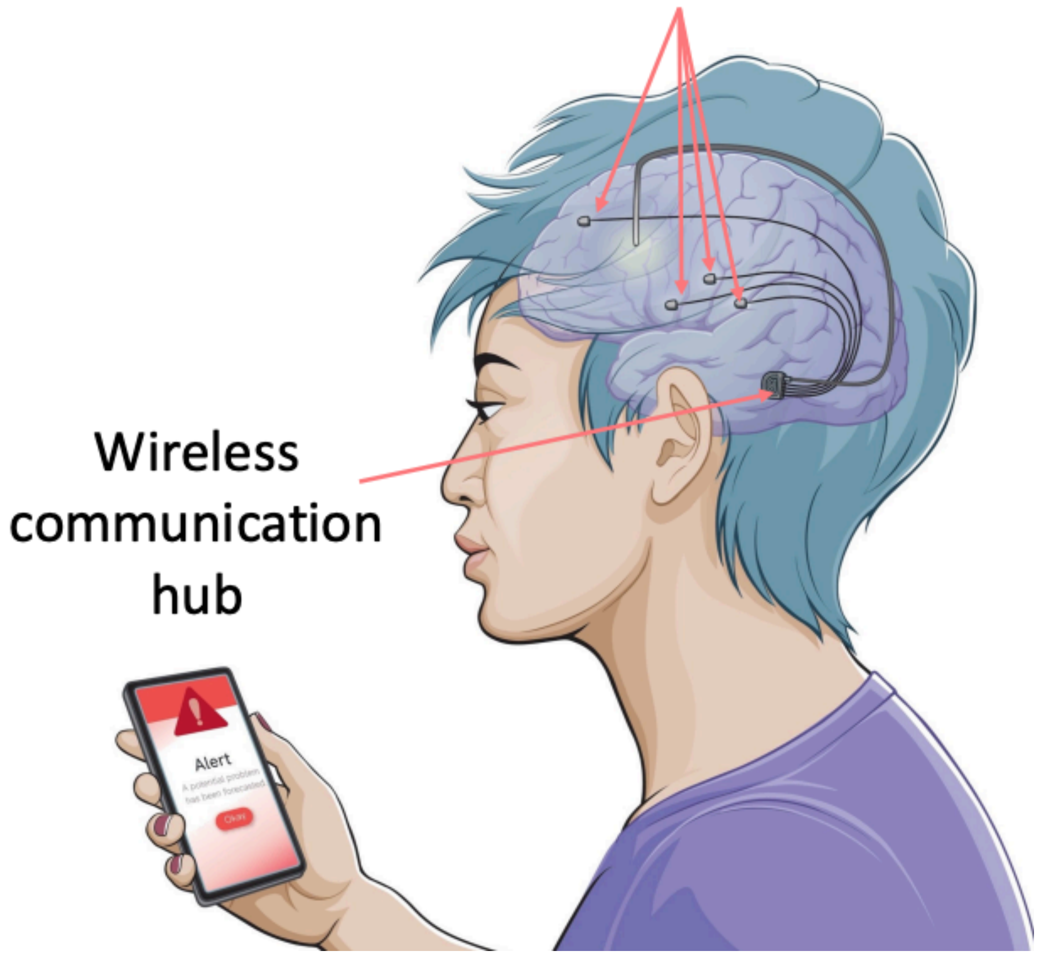
Source: Xilinx, WP505,

...and more!

Semantic segmentation for autonomous vehicles

N. Ghielmetti et al.

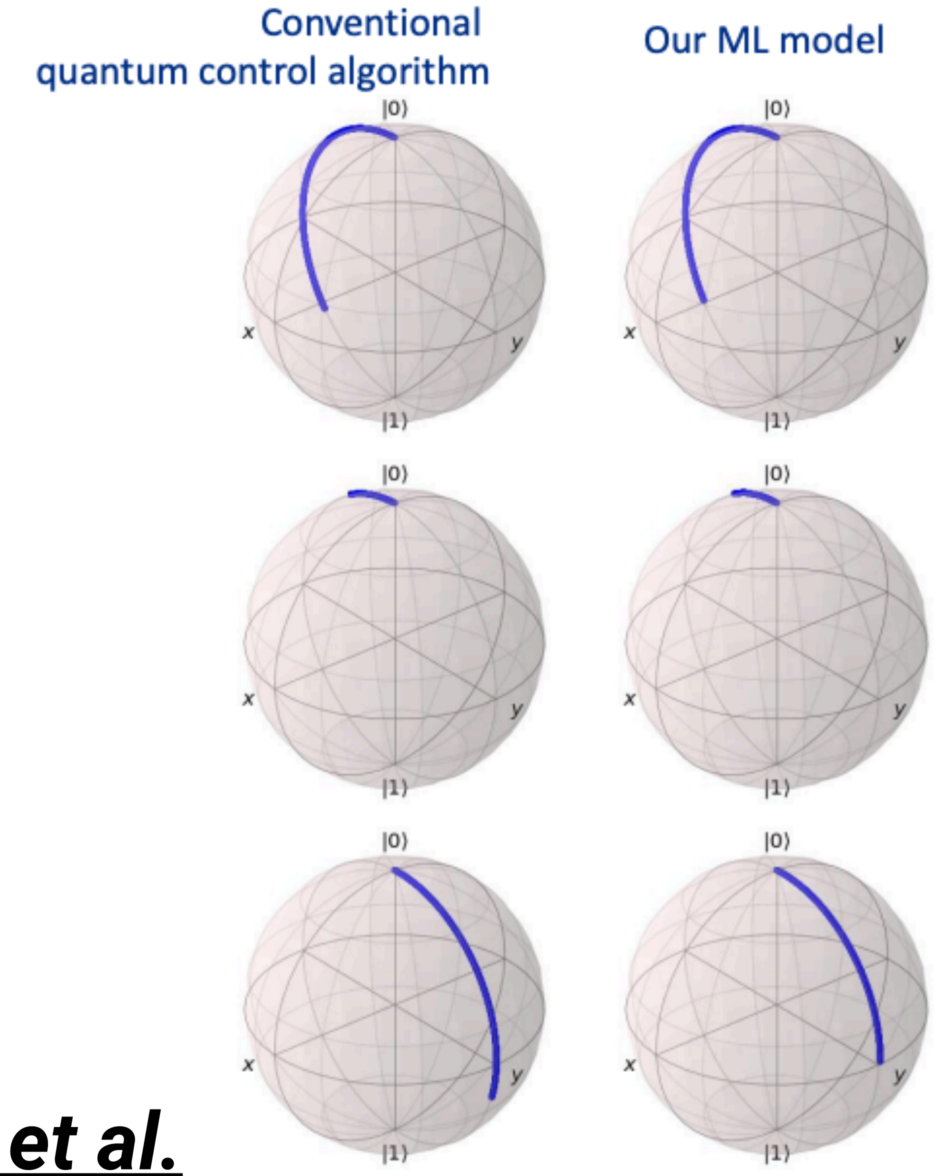
Seizure Predicting Brain Implant



W. Lemaire et al.

NN accelerator for quantum control

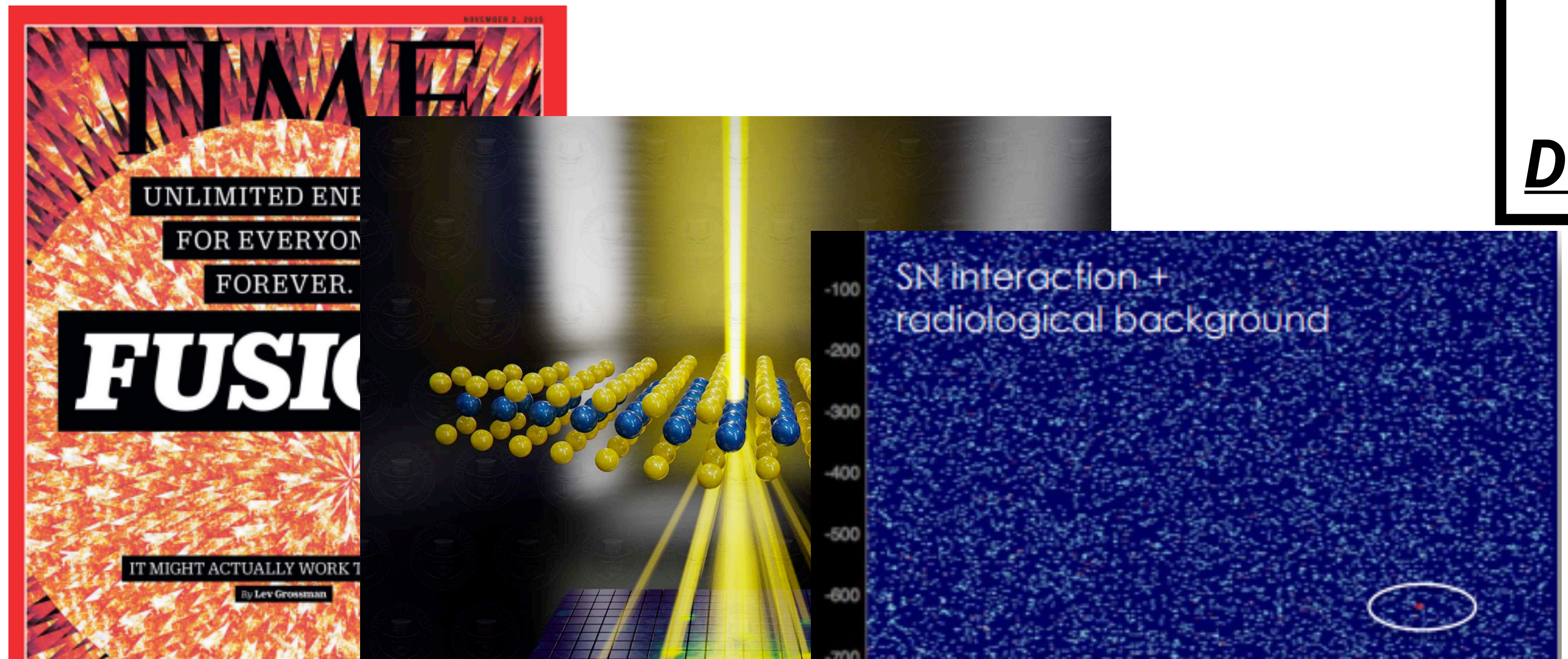
- Putting control in cryostat (e.g optimal pulse parameters)



D Xu et al.

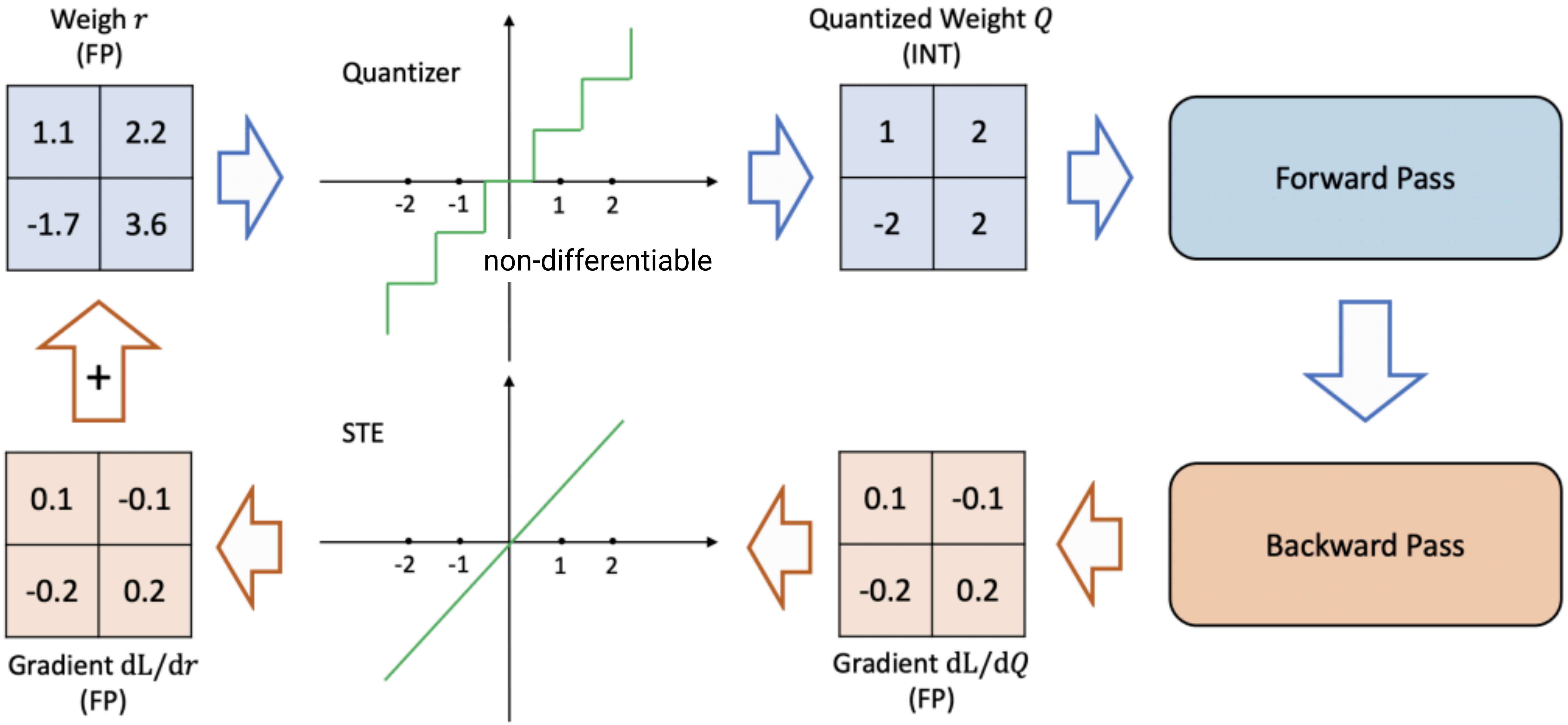
Other examples

- *For fusion science phase/mode monitoring*
- *Crystal structure detection*
- *Triggering in DUNE*
- *Accelerator control*
- *Magnet Quench Detection*
- *MLPerf tinyML benchmarking*
- *Food contamination detection*
- etc….



Quantization-aware training

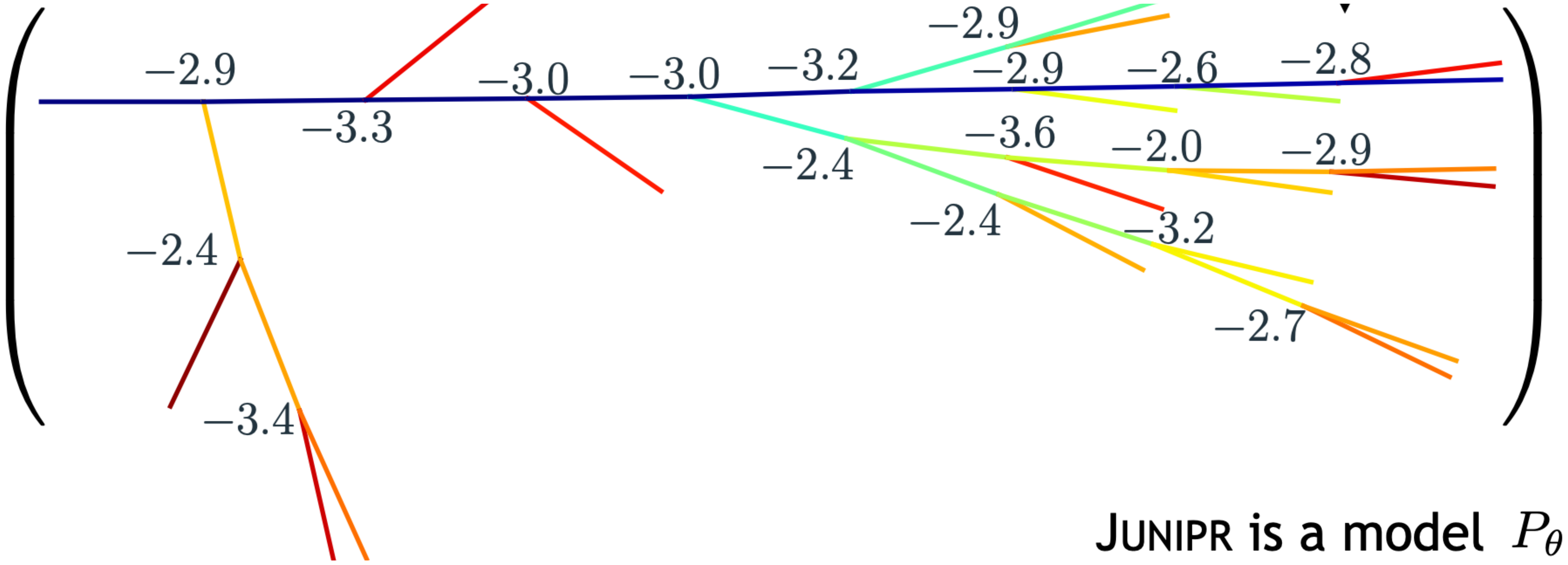
Lossless quantization for deep neural networks!



arXiv:1804.09720

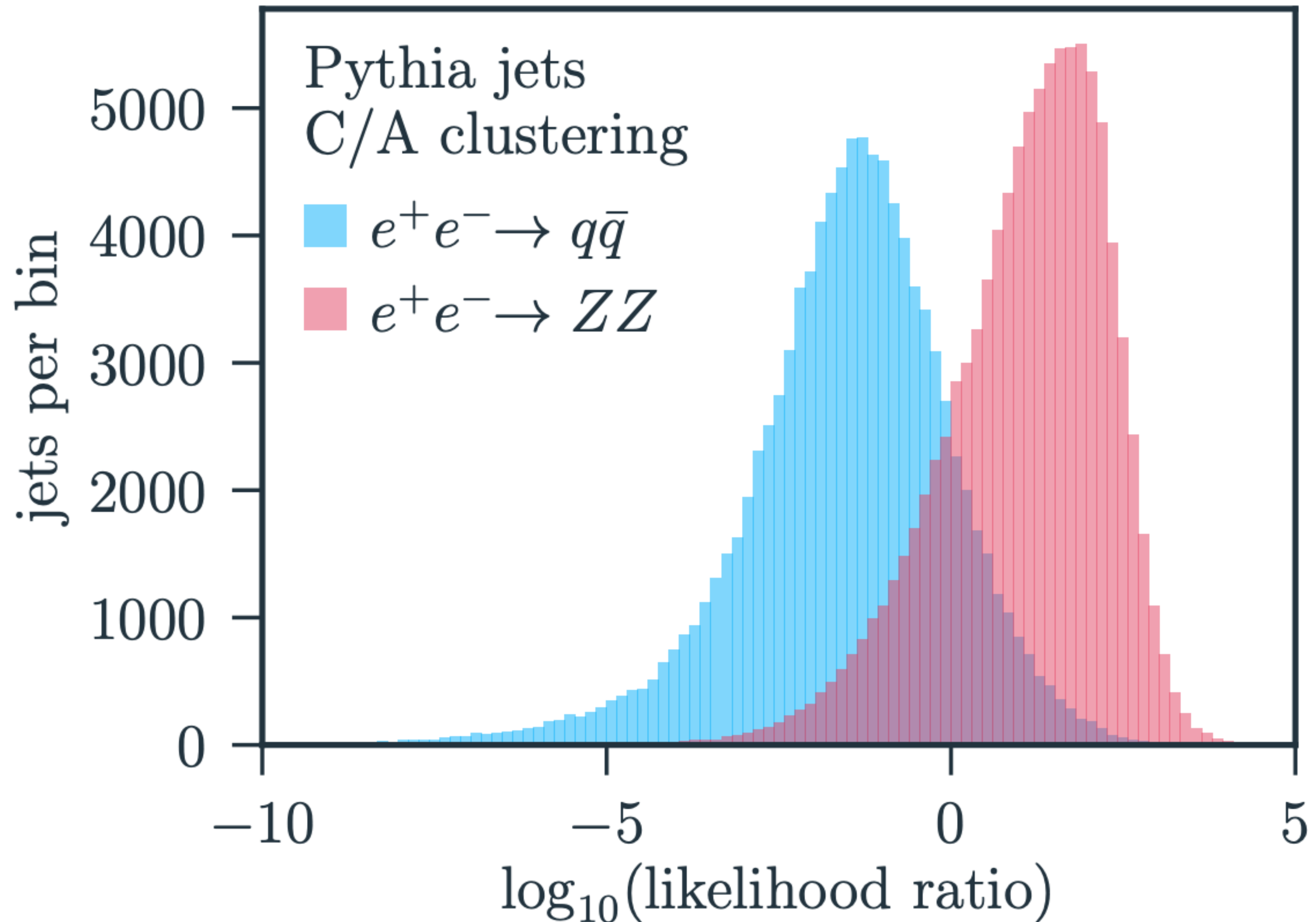
JUNIPR - learning the likelihood

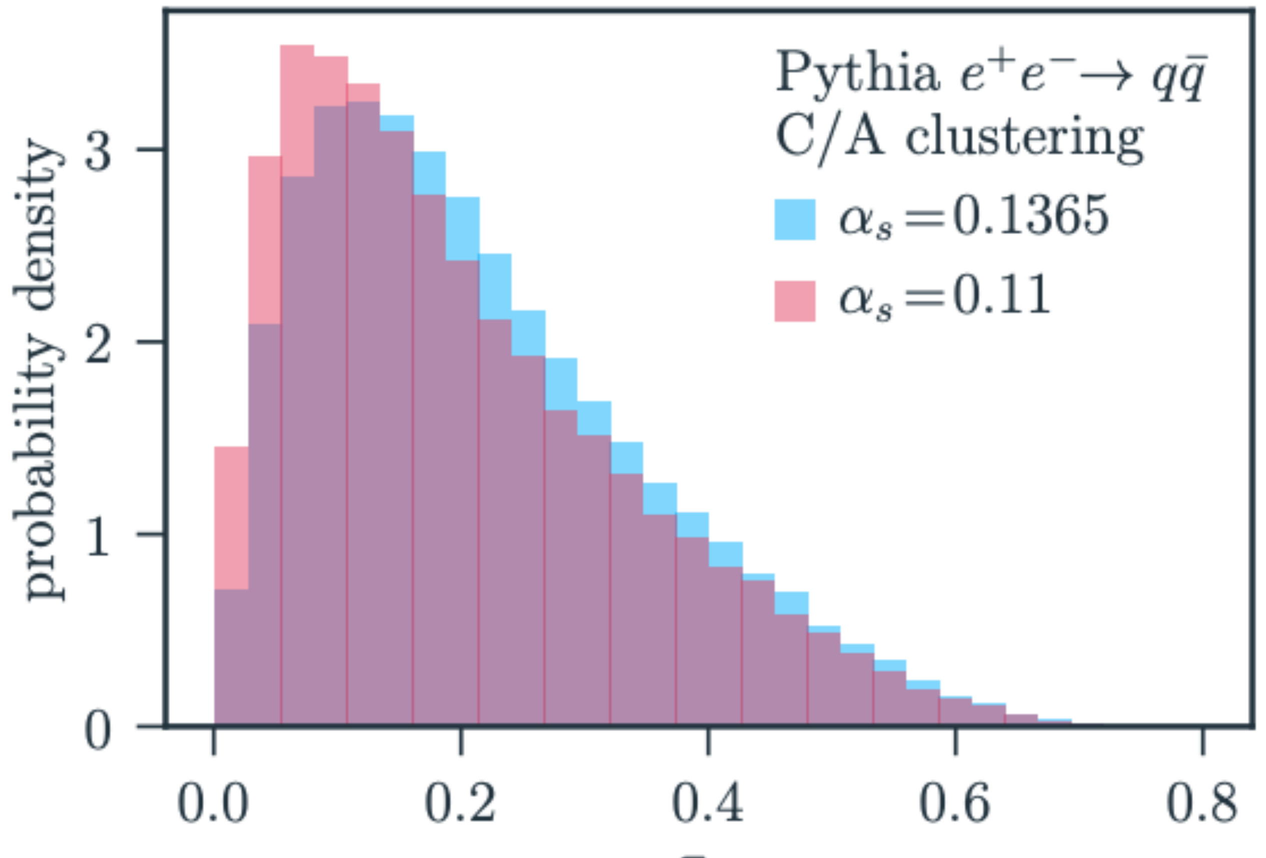
P_{jet}



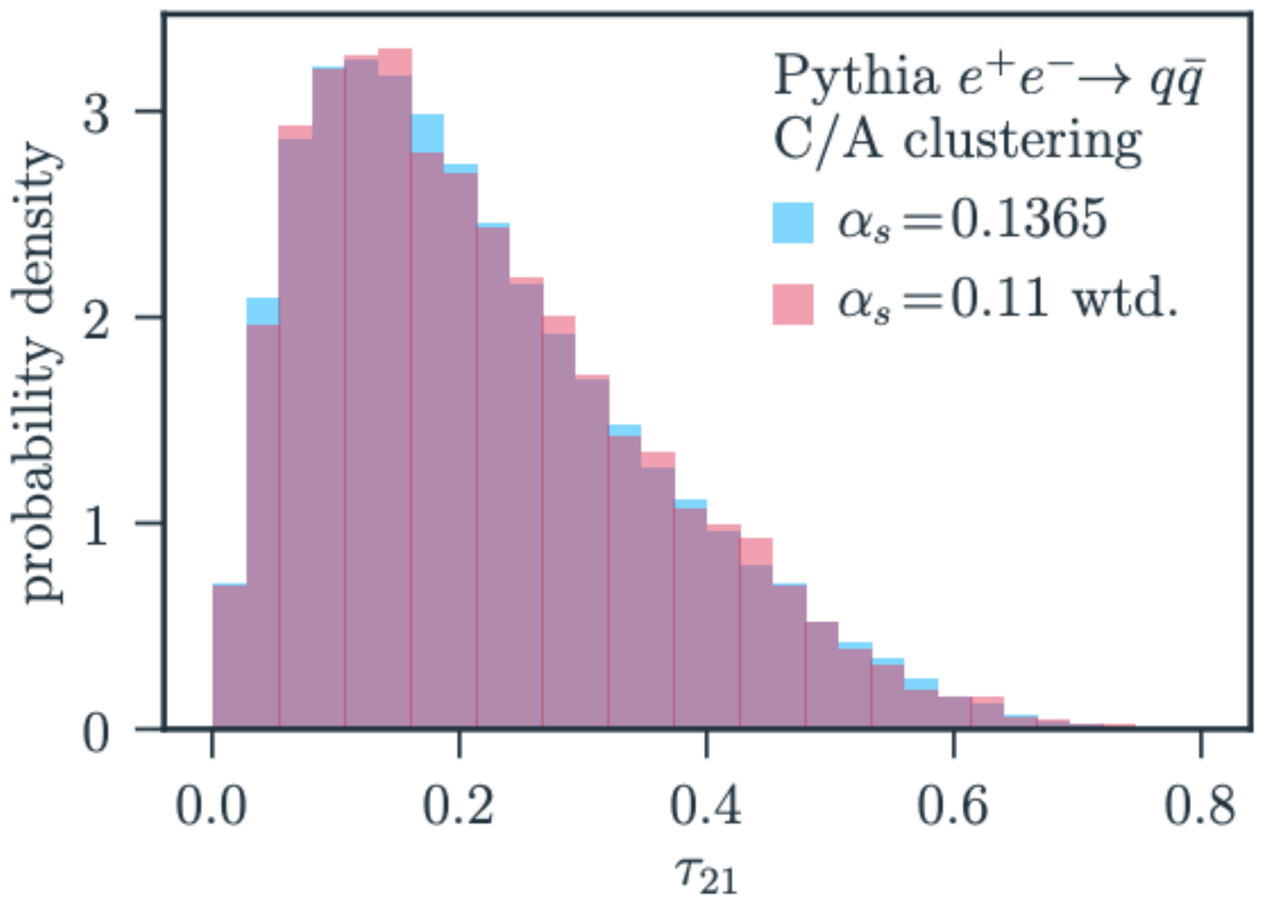
$$\sim \frac{d\sigma}{d^3p_1 \cdots d^3p_n}$$

JUNIPR is a model $P_\theta(\text{jet})$ with 10^6 parameters θ

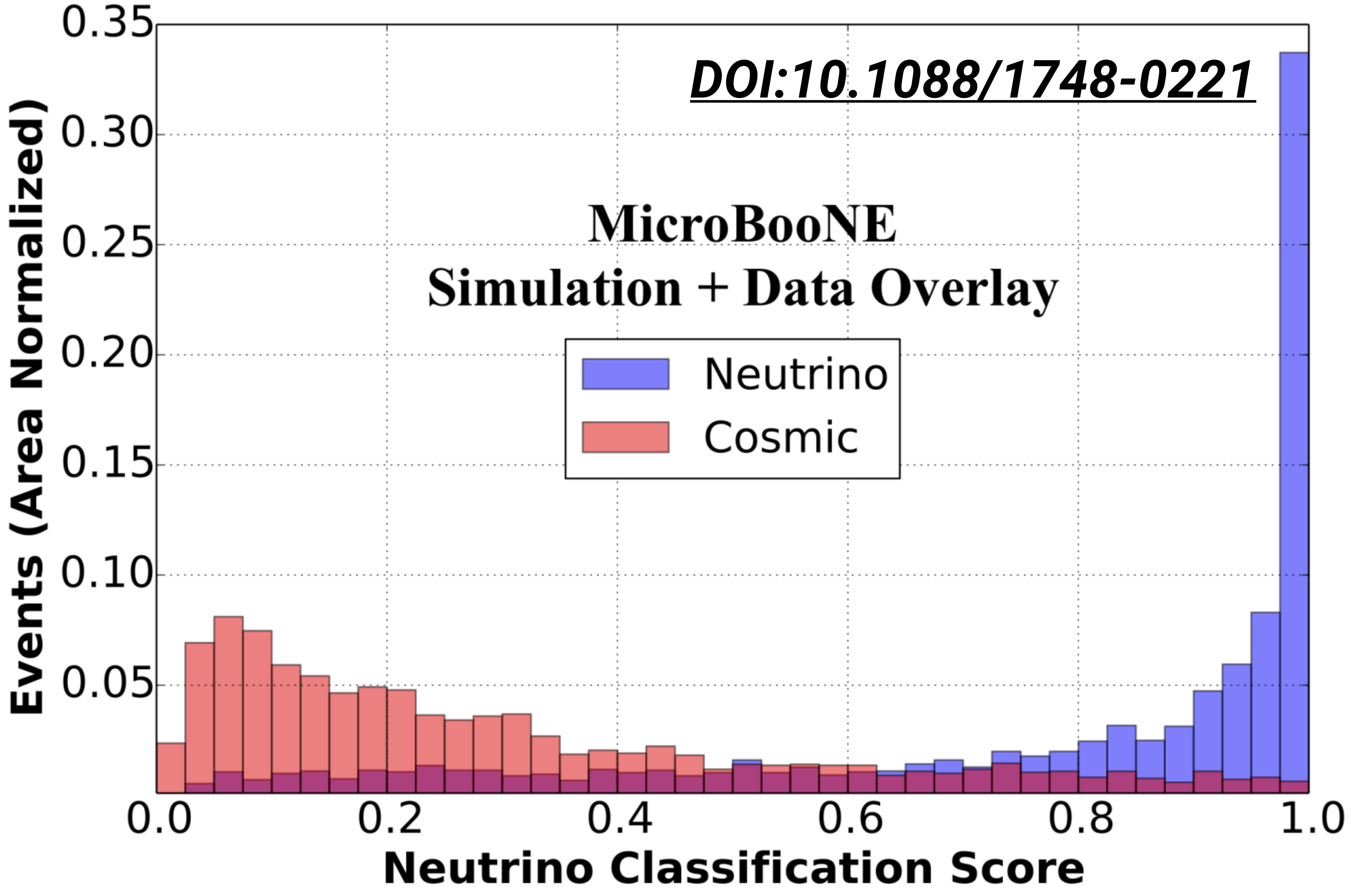


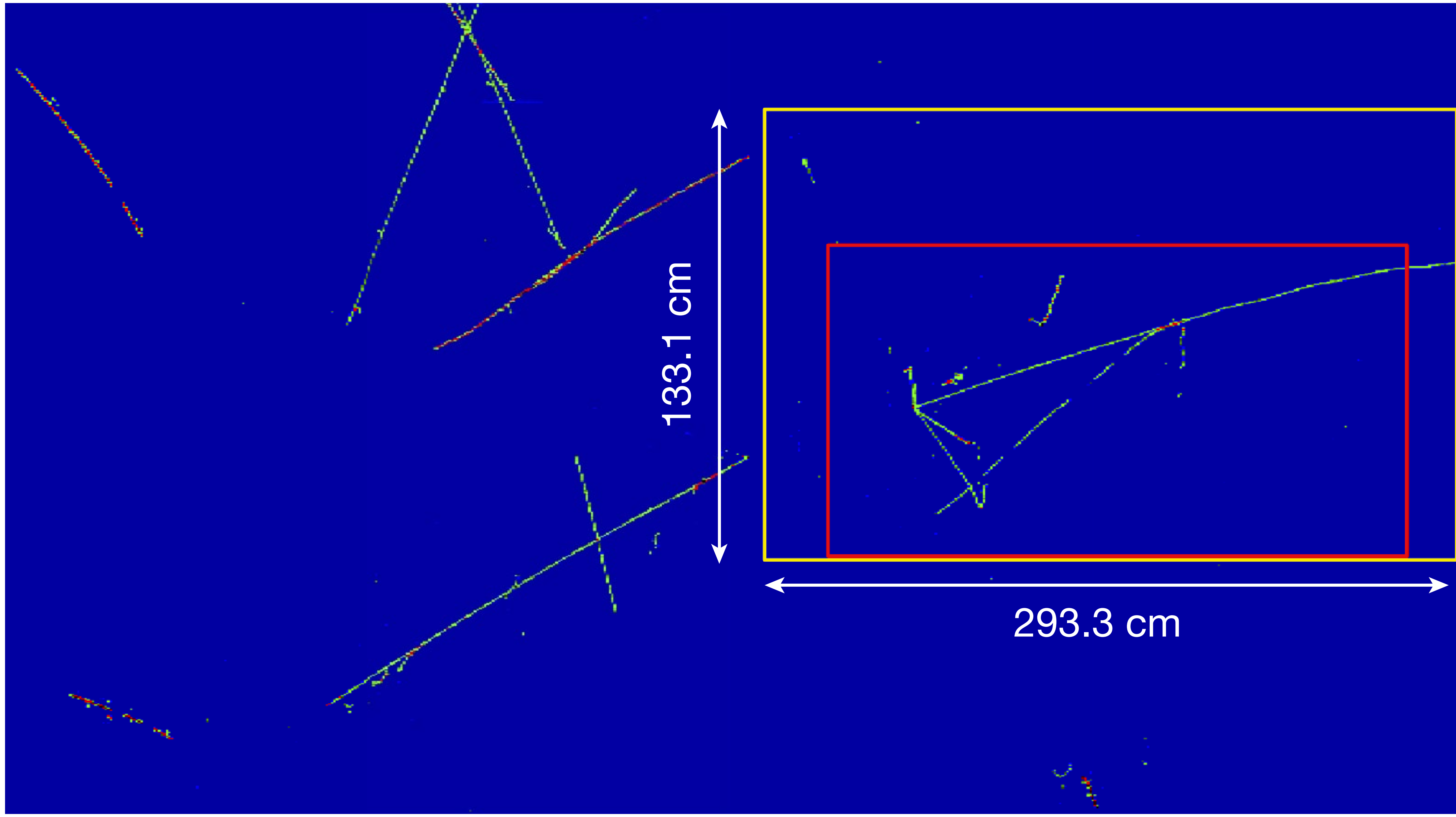


$$* \frac{P_{\text{LHC}}(\text{jet})}{P_{\text{sim}}(\text{jet})} =$$



Hybrid approaches - MicroBooNE





DNN likelihood

Alternative approach: End-to-end DNN search

- How do we get around defining a signal hypothesis?
- What is alternate hypothesis to test reference?

Idea: Assume alternate model n(x|w) can be parametrised in terms of reference model n(x|R)

$$n(x \,|\, \overrightarrow{w}) = n(x \,|\, R)e^{f(x;\overrightarrow{w})}$$ ← Set of real functions

- Let DNN parametrise alternative model

$$f(x;\overrightarrow{w}) = NN$$

DNN likelihood

Alternative approach: End-to-end DNN search

- How do we get around defining a signal hypothesis?
- What is alternate hypothesis to test reference?

Idea: Assume alternate model n(x|w) can be parametrised in terms of reference model n(x|R)

$$n(x\,|\,\vec{w}) = n(x\,|\,R)e^{f(x;\vec{w})} \longleftarrow \text{Set of real functions}$$

- Let DNN parametrise alternative model

$$f(x;\vec{w}) = NN$$

- Formulate loss as log likelihood.
  → Trained DNN **is** the maximum likelihood fit to data and reference log-ratio
  → best approximate of true data distribution

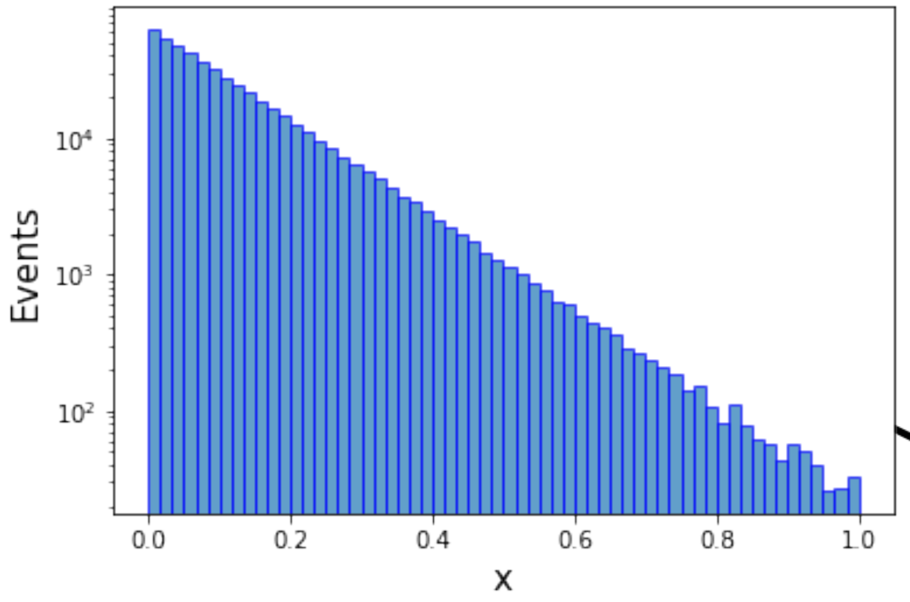
$$f(x, \widehat{\mathbf{w}}) \simeq \log\left[\frac{n(x|\mathrm{T})}{n(x|\mathrm{R})}\right]$$

← True underlying data distribution
← MC distribution

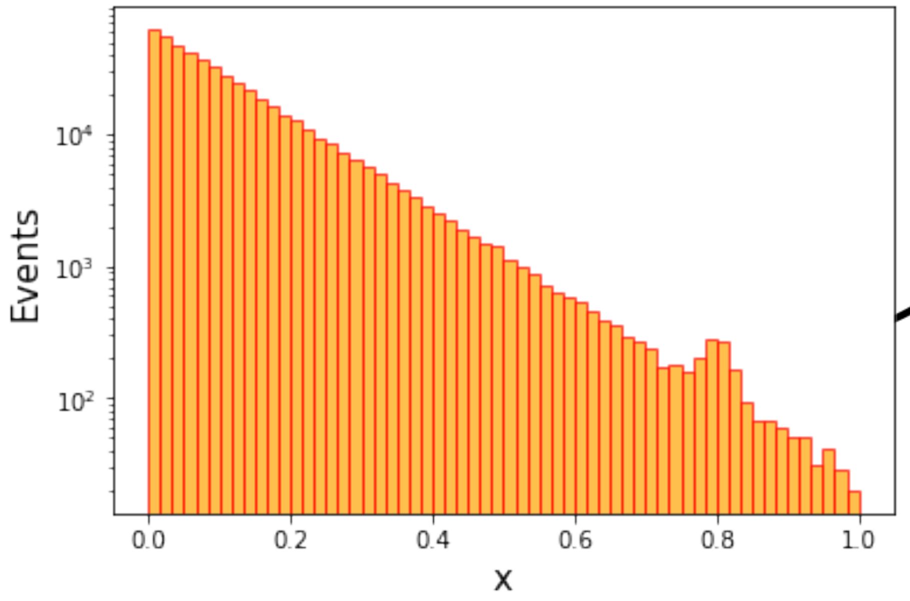
INPUTS

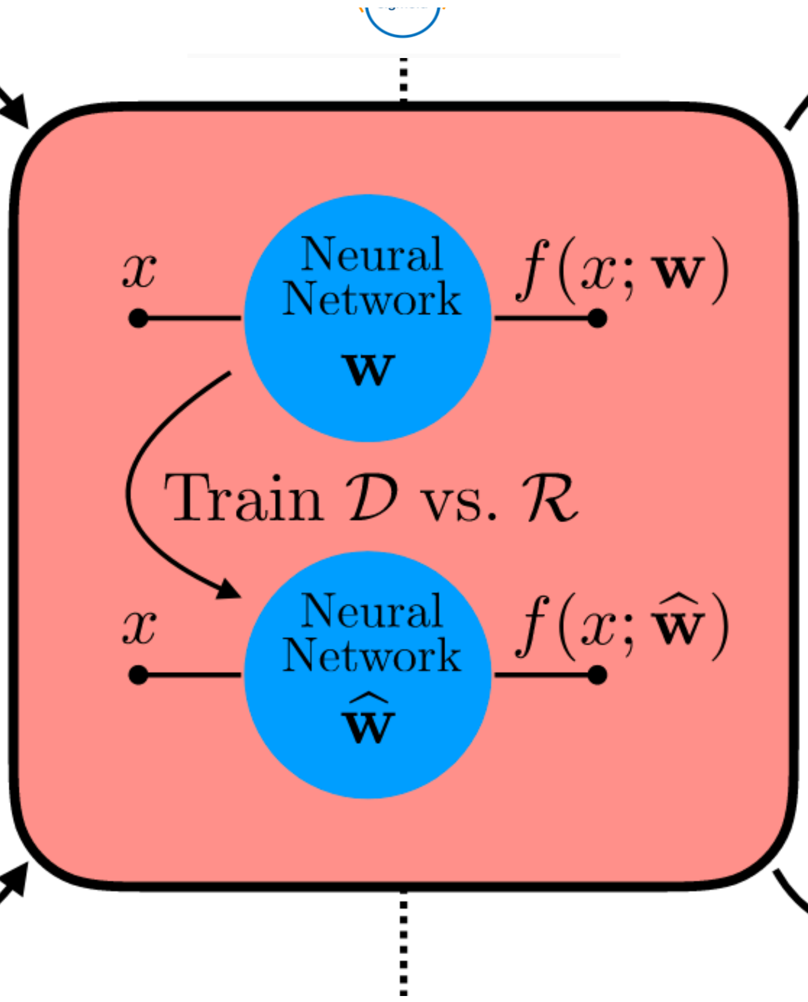
- any high level features

QCD MC R



CMS DATA D

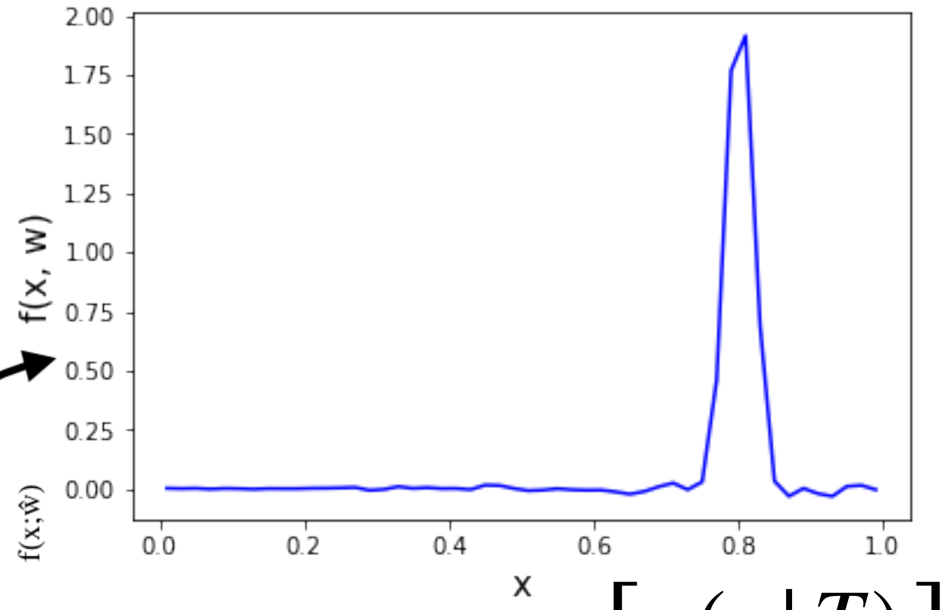




OUTPUTS

-t_{obs} and f(x; ŵ)

1) Best fit log ratio of data and MC PDFs



$$f(x,\hat{w}) \simeq \log\left[\frac{n(x\,|\,T)}{n(x\,|\,R)}\right]$$

2) test-statistic on data sample t_{obs}

$$t(\mathcal{D}) = -2 \operatorname*{Min}_{\{\mathbf{w}\}} L[f]$$ ← **DNN loss function!**

Can be used to build hypothesis test + p-value

Data → toys under R, repeat

$$f(x,\widehat{\mathbf{w}}) \simeq \log\left[\frac{n(x|\mathrm{T})}{n(x|\mathrm{R})}\right]$$

← True underlying data distribution

← MC distribution

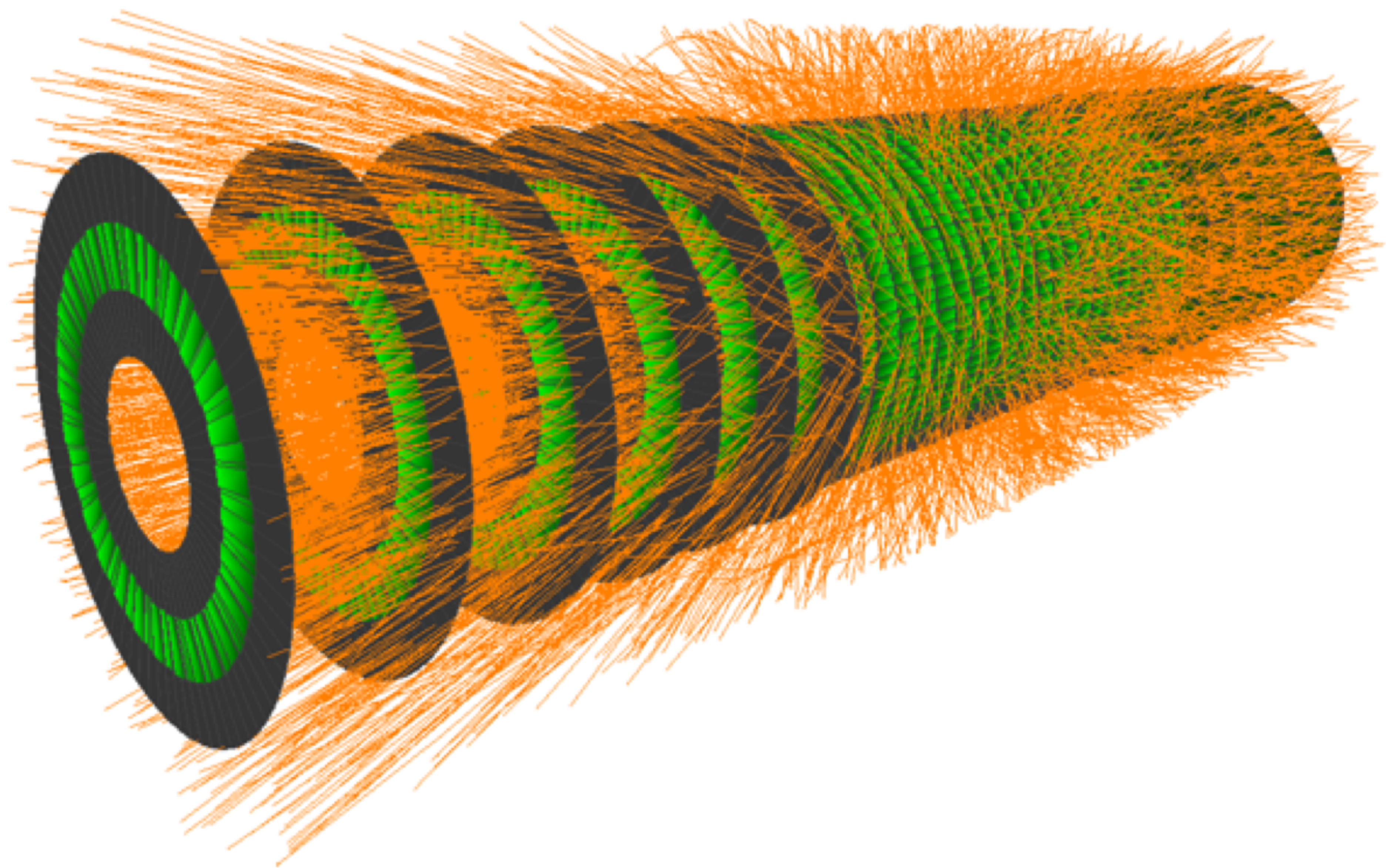
DOI:10.3389/fdata.2022.828666

ML on FPGA for tracking

In HL-LHC, will need to do track finding at L1

- O(1000) hits, O(100) tracks, 40 MHz rate, ~5 μs latency

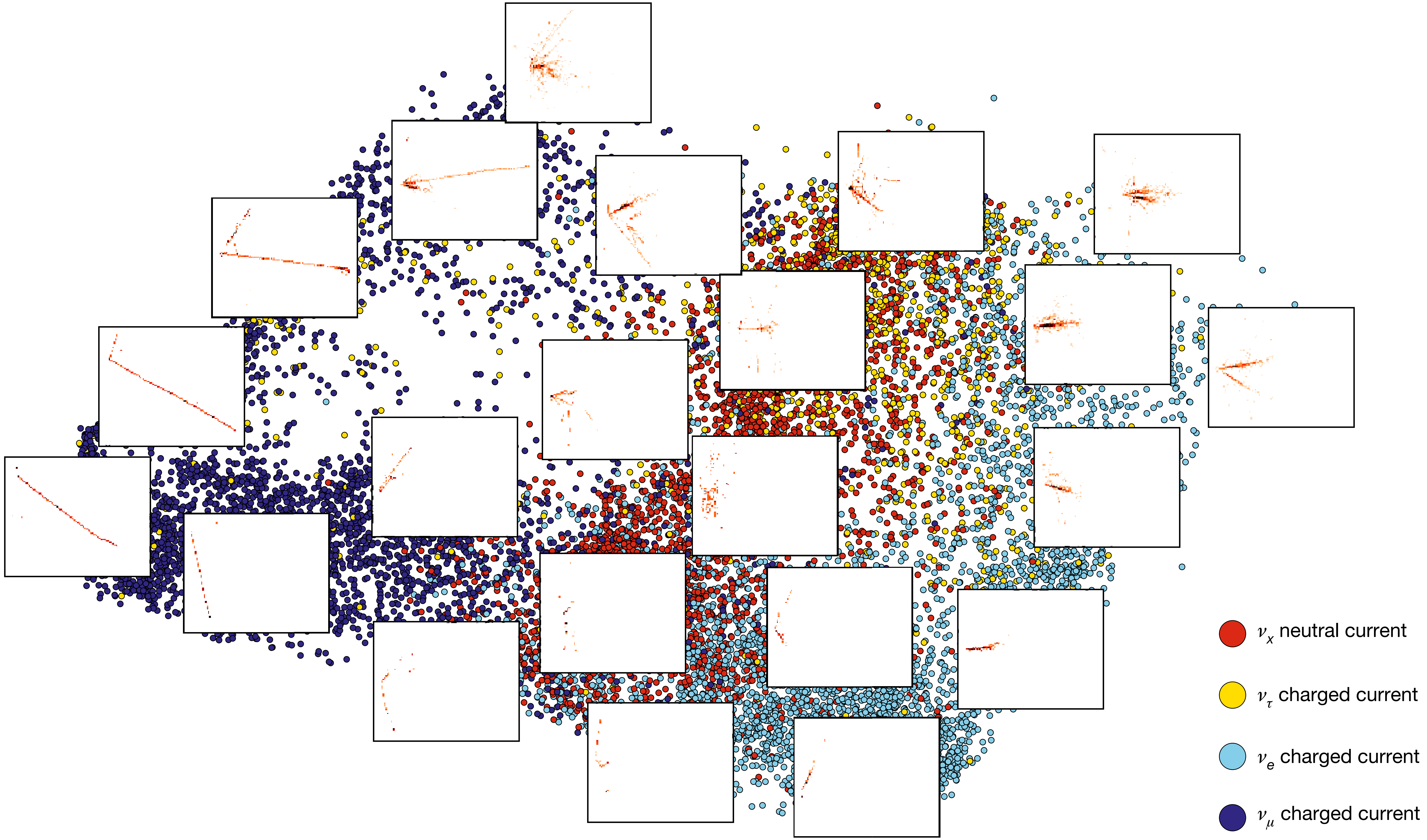
Graph Neural Networks for fast charged particle tracking



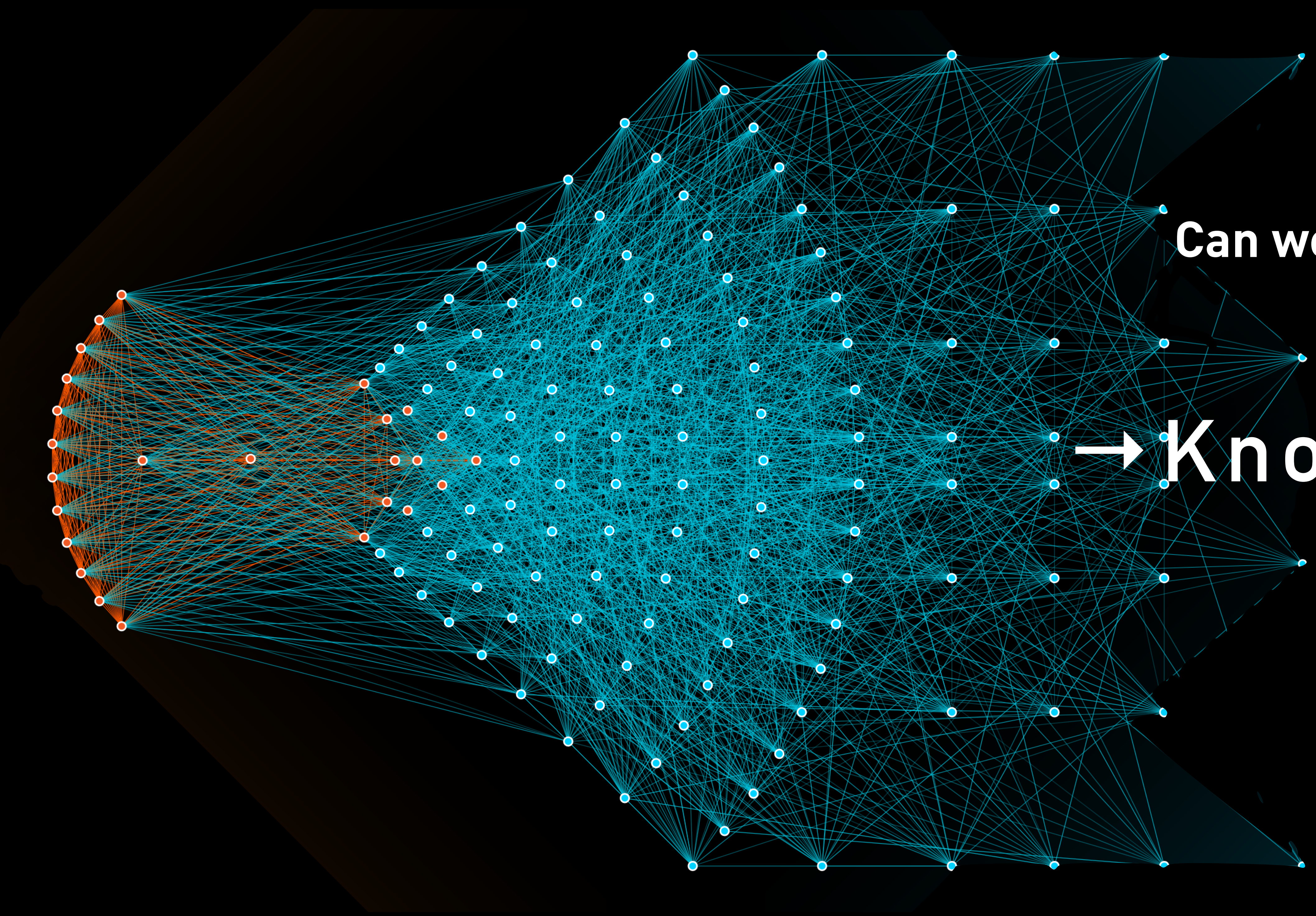
Design	$(n_{\text{nodes}}, n_{\text{edges}})$	RF	Precision	Latency [cycles]	II [cycles]	DSP [%]	LUT [%]	FF [%]	BRAM [%]
Throughput-opt.	(28, 56)	1	`ap_fixed<14,7>`	59	1	99.9	66.0	11.7	0.7

The target FPGA is a Xilinx Virtex UltraScale+ VU9P FPGA (part number xcvu9p-flga2104-2L-e), which has 6,840 DSPs, 1,182,240 LUTs, 2,364,480 FFs, and 75.9 Mb of BRAM (Xilinx, Inc., 2021). A 5 ns clock period is used.

Probing algorithm response: t-SNE for NoVA

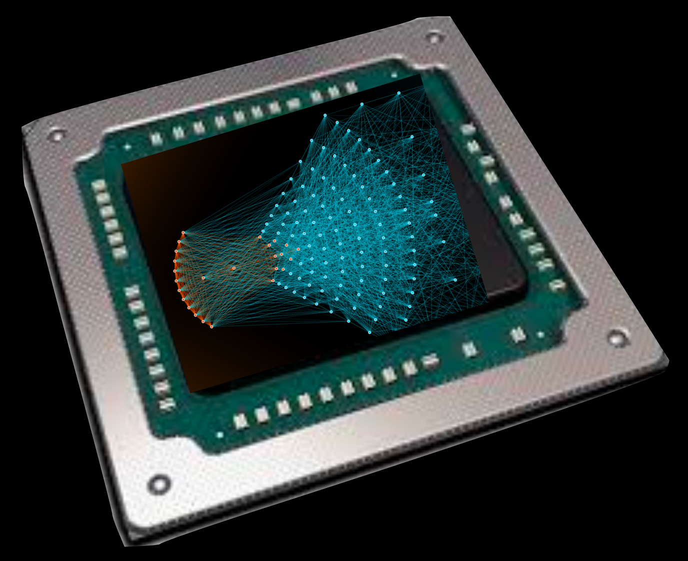


Nature Review



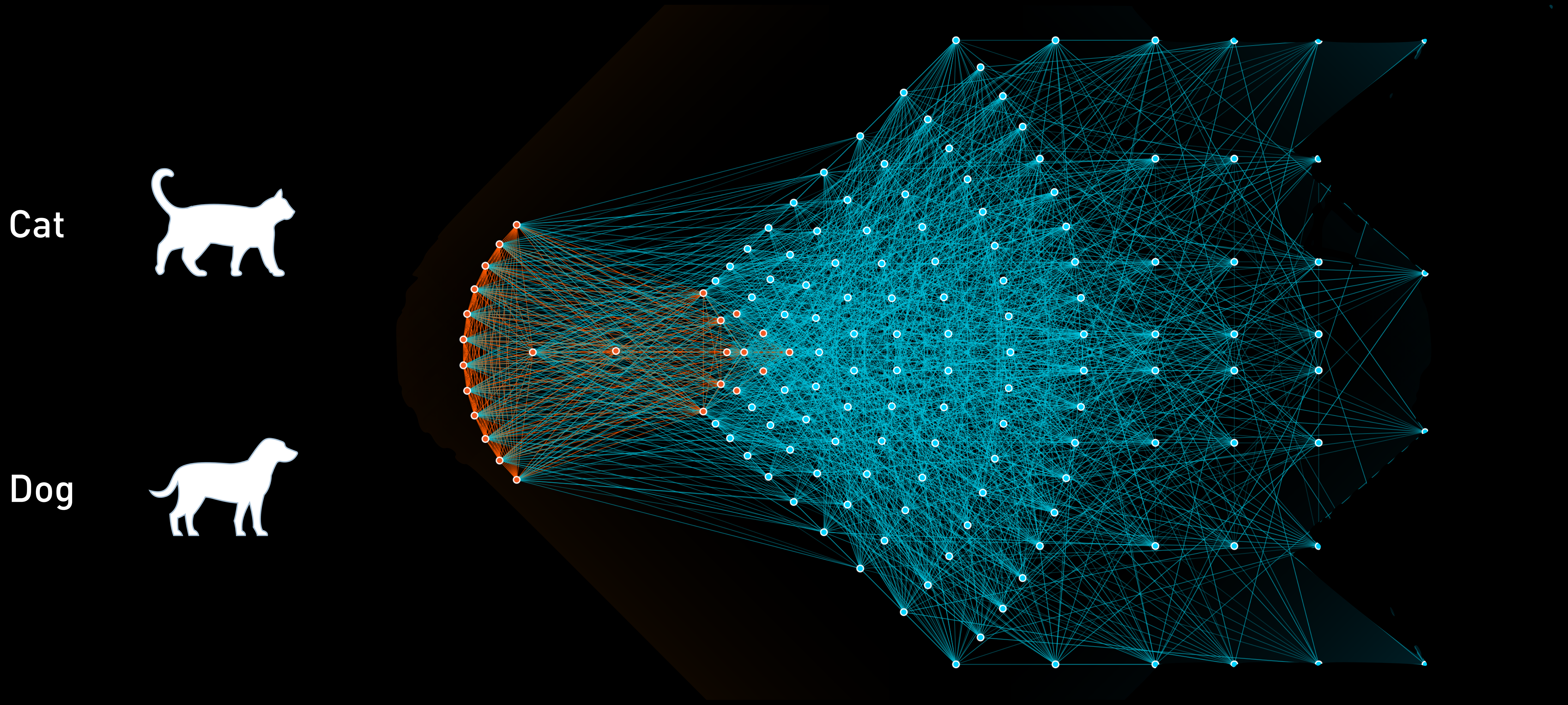
Can we have the best of both worlds?

→Knowledge Distillation

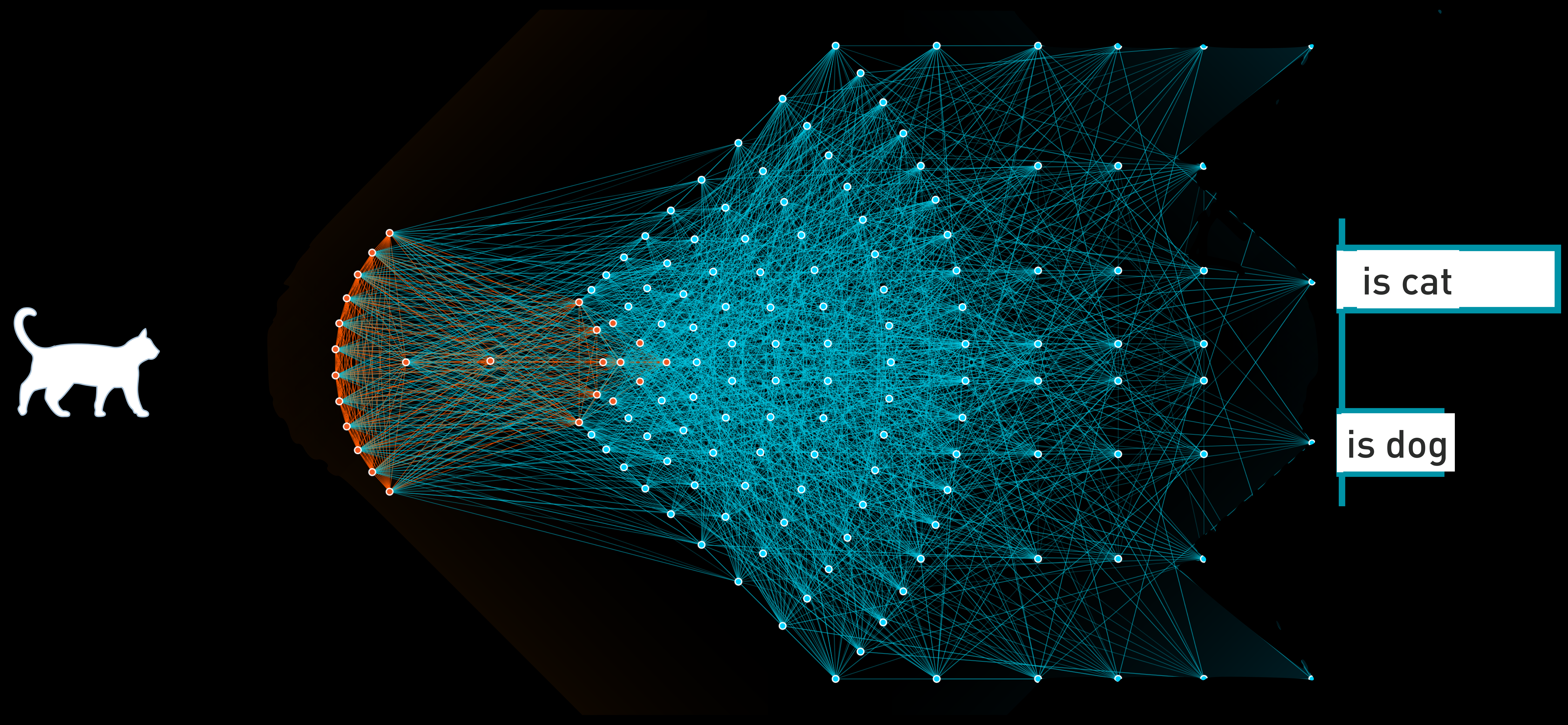


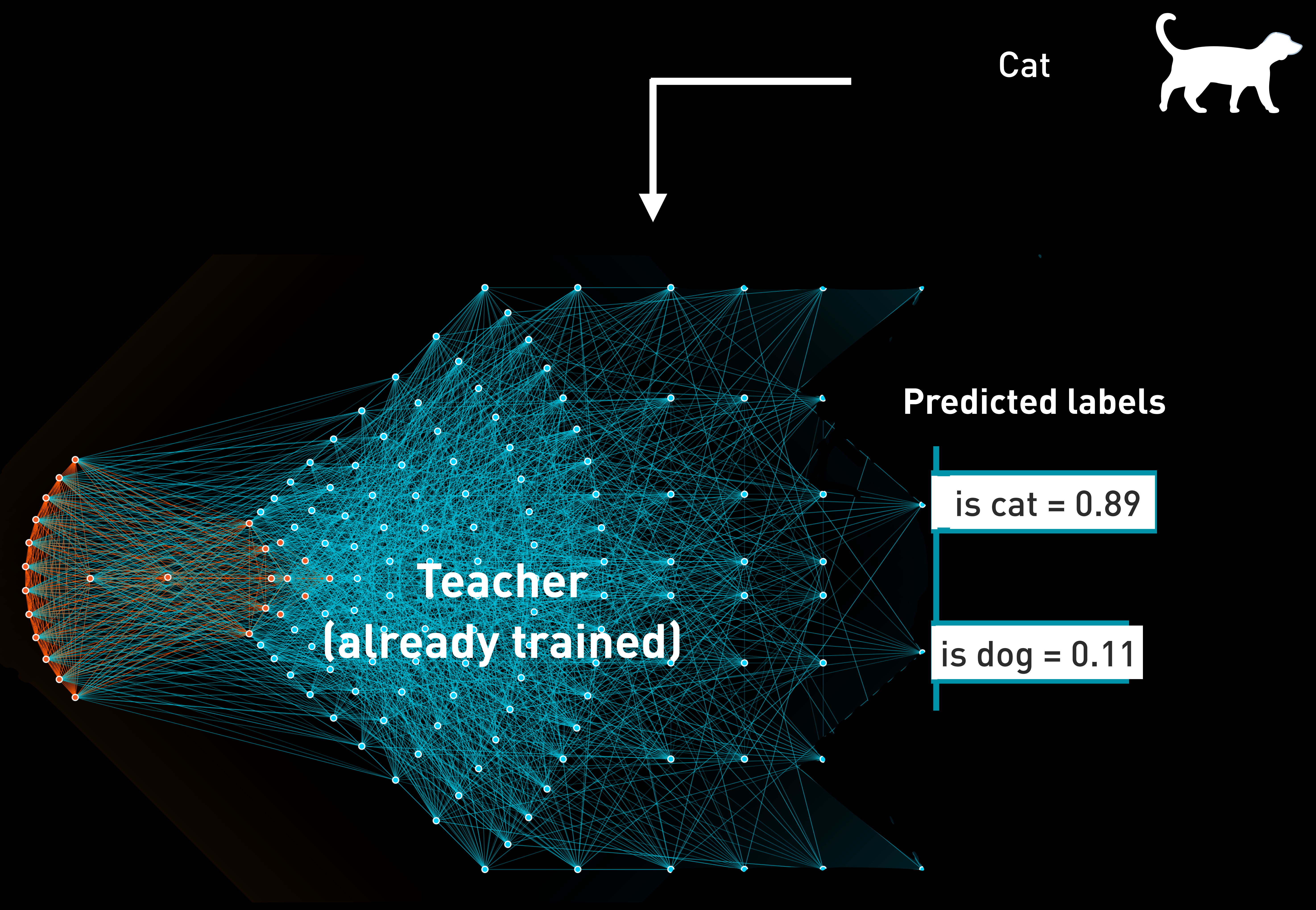
Train

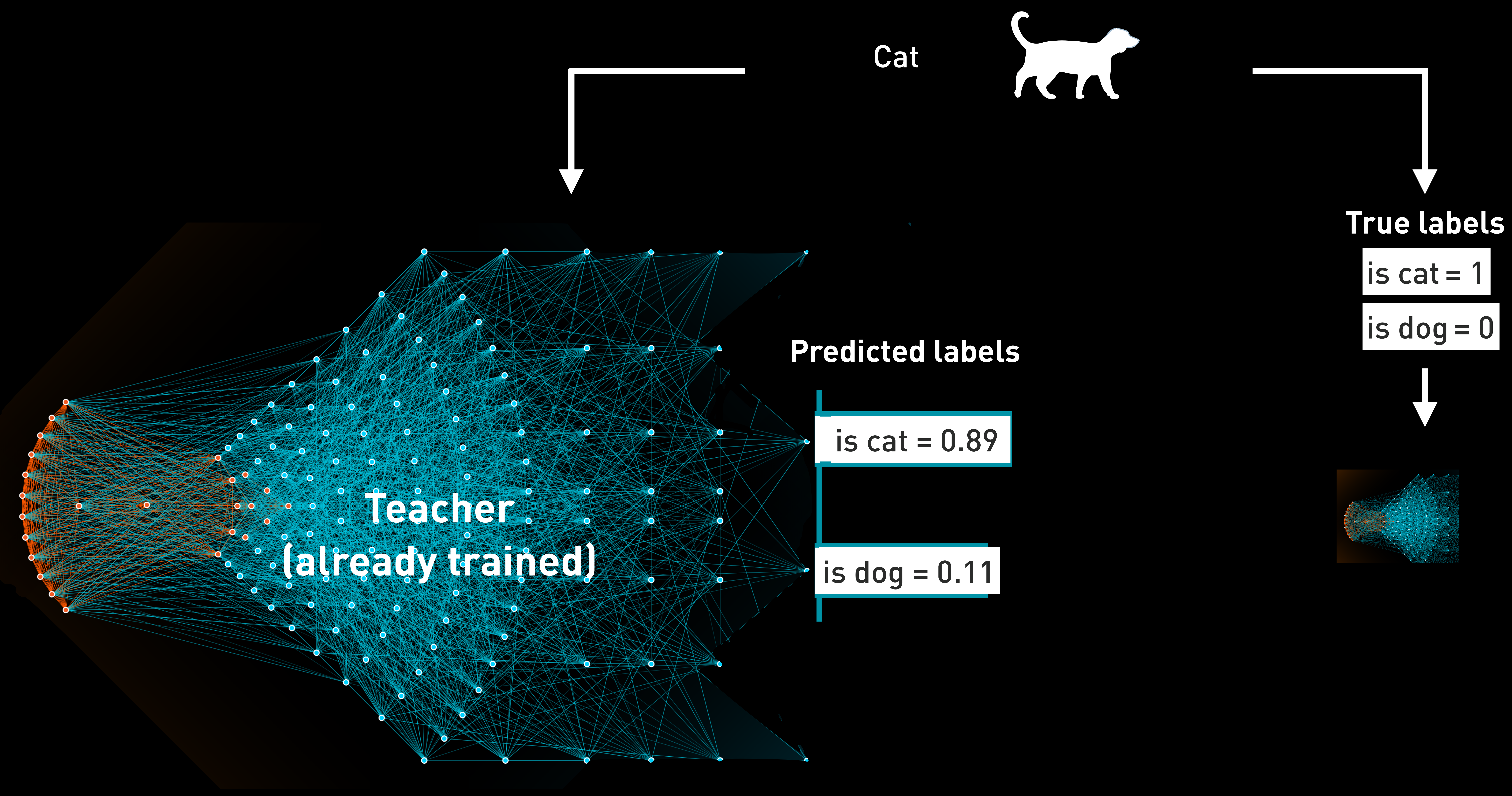
Inference


Cat
Dog

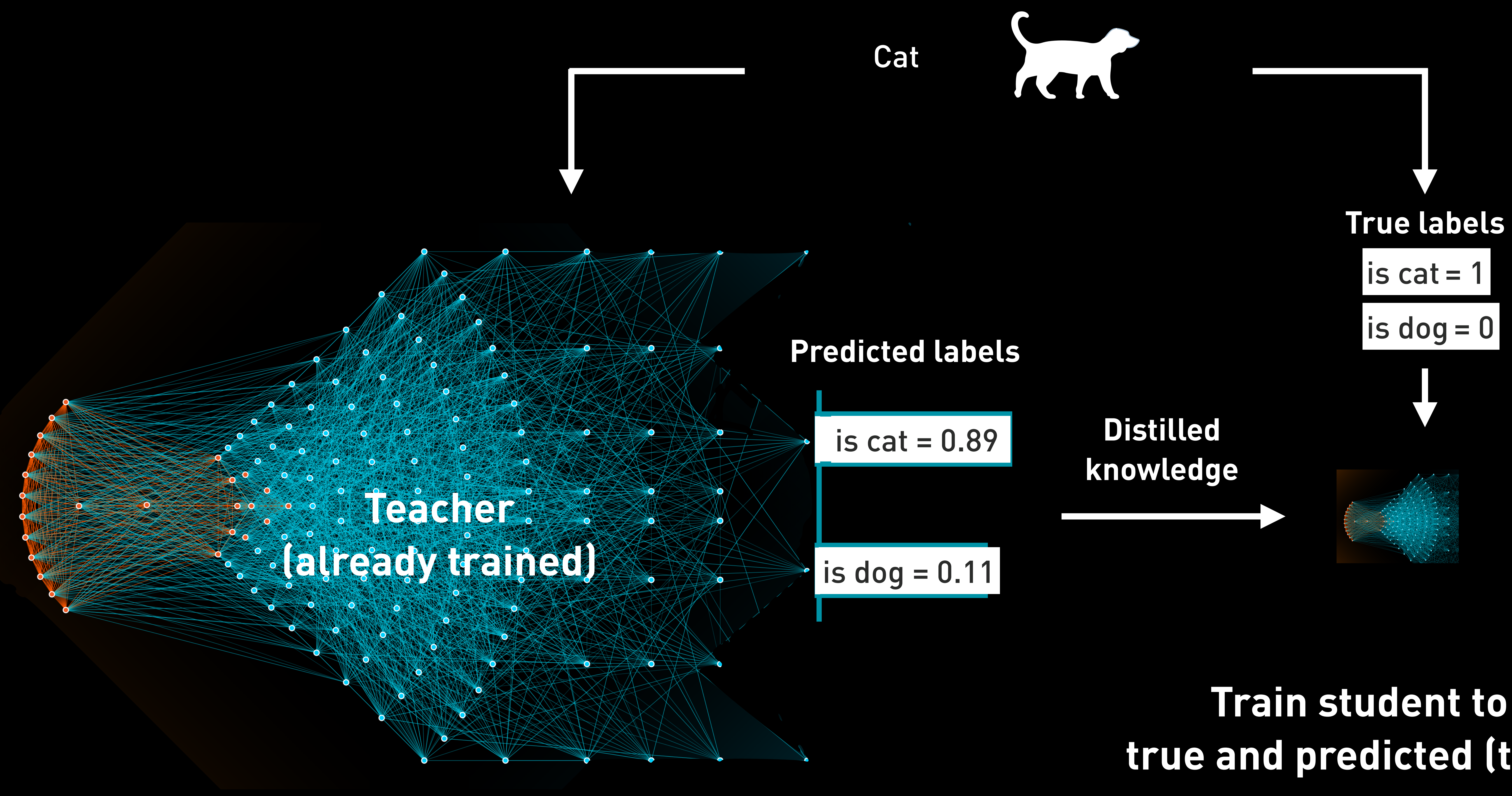
Cat




Cat
Teacher
(already trained)
Predicted labels
is cat = 0.89
is dog = 0.11



Cat
True labels
is cat = 1
is dog = 0
Predicted labels
is cat = 0.89
is dog = 0.11
Teacher
(already trained)



Train student to learn both true and predicted (teacher) labels!

$$L_{total} = \beta \times L_{Distillation} + \alpha \times L_{student}$$

Student learns subtle learned features from teacher!