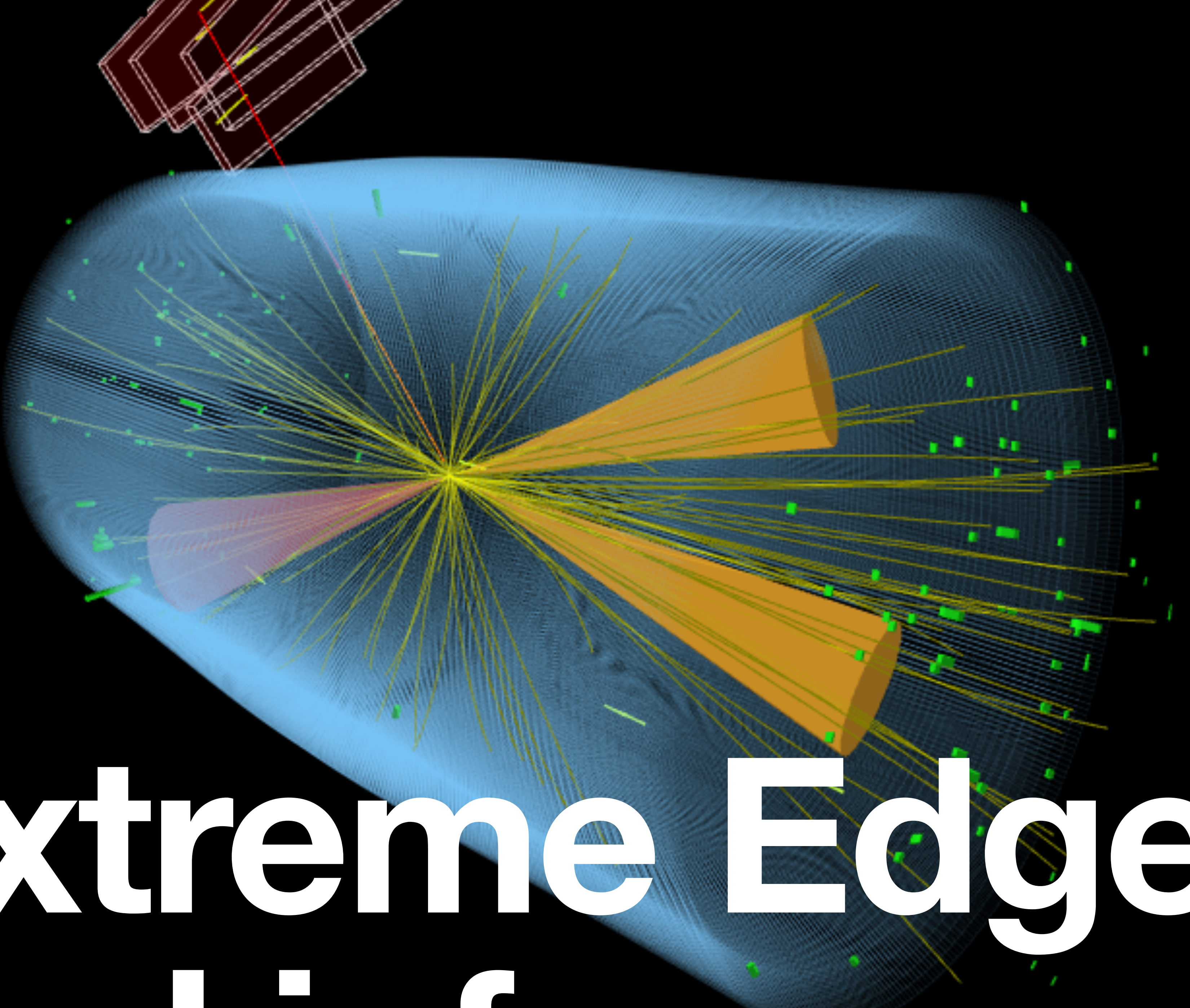
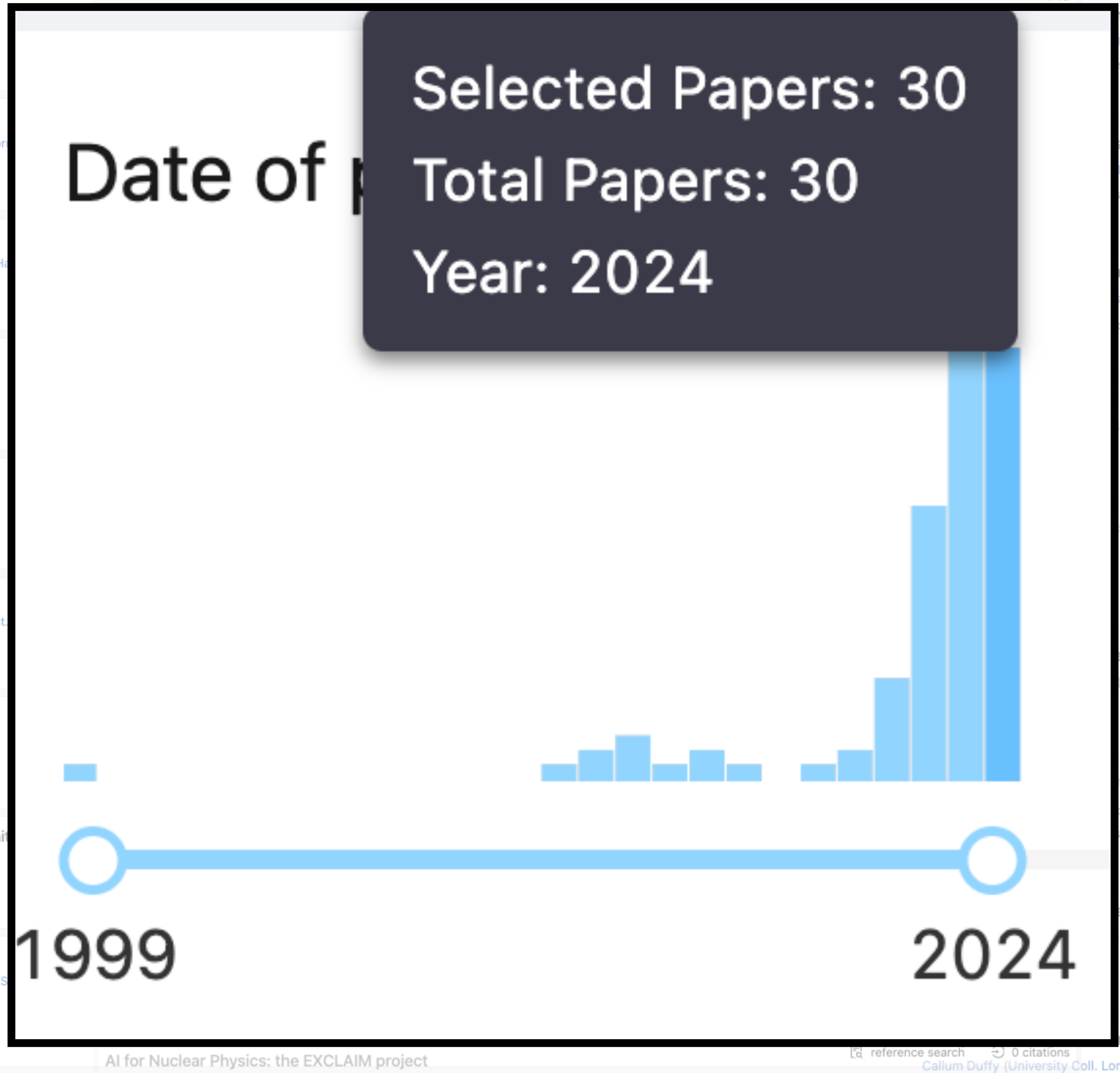
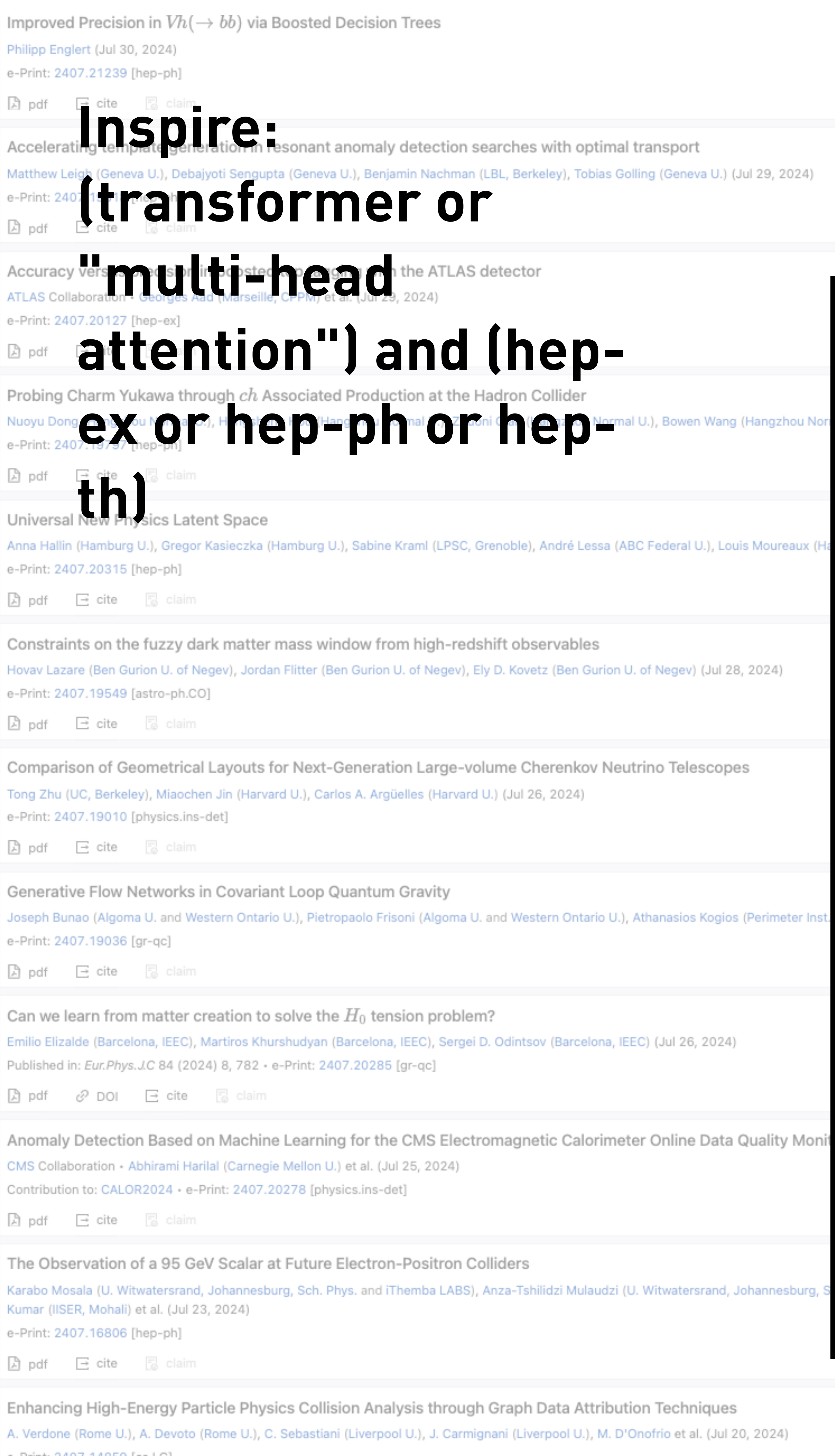


AI at the Extreme Edge: Nanosecond inference for New Physics Discovery at the LHC

Thea Klaeboe Aarrestad (ETH Zürich)

Inspire: (transformer or "multi-head attention") and (hep-ex or hep-ph or hep-th)




Improved Precision in $Vh(\to bb)$ via Boosted Decision Trees
Philipp Englert (Jul 30, 2024)
e-Print: 2407.21239 [hep-ph]

Accelerating template generation in resonant anomaly detection searches with optimal transport
Matthew Leigh (Geneva U.), Debajyoti Sengupta (Geneva U.), Benjamin Nachman (LBL, Berkeley), Tobias Golling (Geneva U.) (Jul 29, 2024)

Accuracy versus precision in boosted top tagging with the ATLAS detector
ATLAS Collaboration • Georges Aad (Marseille, CPPM) et al. (Jul 29, 2024)
e-Print: 2407.20127 [hep-ex]

Probing Charm Yukawa through ch Associated Production at the Hadron Collider

Universal New Physics Latent Space
Anna Hallin (Hamburg U.), Gregor Kasieczka (Hamburg U.), Sabine Kraml (LPSC, Grenoble), André Lessa (ABC Federal U.), Louis Moureaux
e-Print: 2407.20315 [hep-ph]

Constraints on the fuzzy dark matter mass window from high-redshift observables
Hovav Lazare (Ben Gurion U. of Negev), Jordan Flitter (Ben Gurion U. of Negev), Ely D. Kovetz (Ben Gurion U. of Negev) (Jul 28, 2024)
e-Print: 2407.19549 [astro-ph.CO]

Comparison of Geometrical Layouts for Next-Generation Large-volume Cherenkov Neutrino Telescopes
Tong Zhu (UC, Berkeley), Miaochen Jin (Harvard U.), Carlos A. Argüelles (Harvard U.) (Jul 26, 2024)
e-Print: 2407.19010 [physics.ins-det]

Generative Flow Networks in Covariant Loop Quantum Gravity
Joseph Bunao (Algoma U. and Western Ontario U.), Pietropaolo Frisoni (Algoma U. and Western Ontario U.), Athanasios Kogios (Perimeter Inst.
e-Print: 2407.19036 [gr-qc]

Can we learn from matter creation to solve the H_0 tension problem?
Emilio Elizalde (Barcelona, IEEC), Martiros Khurshudyan (Barcelona, IEEC), Sergei D. Odintsov (Barcelona, IEEC) (Jul 26, 2024)
Published in: Eur.Phys.J.C 84 (2024) 8, 782 • e-Print: 2407.20285 [gr-qc]

Anomaly Detection Based on Machine Learning for the CMS Electromagnetic Calorimeter Online Data Quality Moni
CMS Collaboration • Abhirami Harilal (Carnegie Mellon U.) et al. (Jul 25, 2024)
Contribution to: CALOR2024 • e-Print: 2407.20278 [physics.ins-det]

The Observation of a 95 GeV Scalar at Future Electron-Positron Colliders
Karabo Mosala (U. Witwatersrand, Johannesburg, Sch. Phys. and iThemba LABS), Anza-Tshilidzi Mulaudzi (U. Witwatersrand, Johannesburg,
Kumar (IISER, Mohali) et al. (Jul 23, 2024)
e-Print: 2407.16806 [hep-ph]

Enhancing High-Energy Particle Physics Collision Analysis through Graph Data Attribution Techniques
A. Verdone (Rome U.), A. Devoto (Rome U.), C. Sebastiani (Liverpool U.), J. Carmignani (Liverpool U.), M. D'Onofrio et al. (Jul 20, 2024)

Pay Attention To Mean Fields For Point Cloud Generation
Benno Käch (DESY), Isabell Melzer-Pellmann (DESY), Dirk Krücker (DESY) (Aug 9, 2024)
e-Print: 2408.04997 [hep-ex]

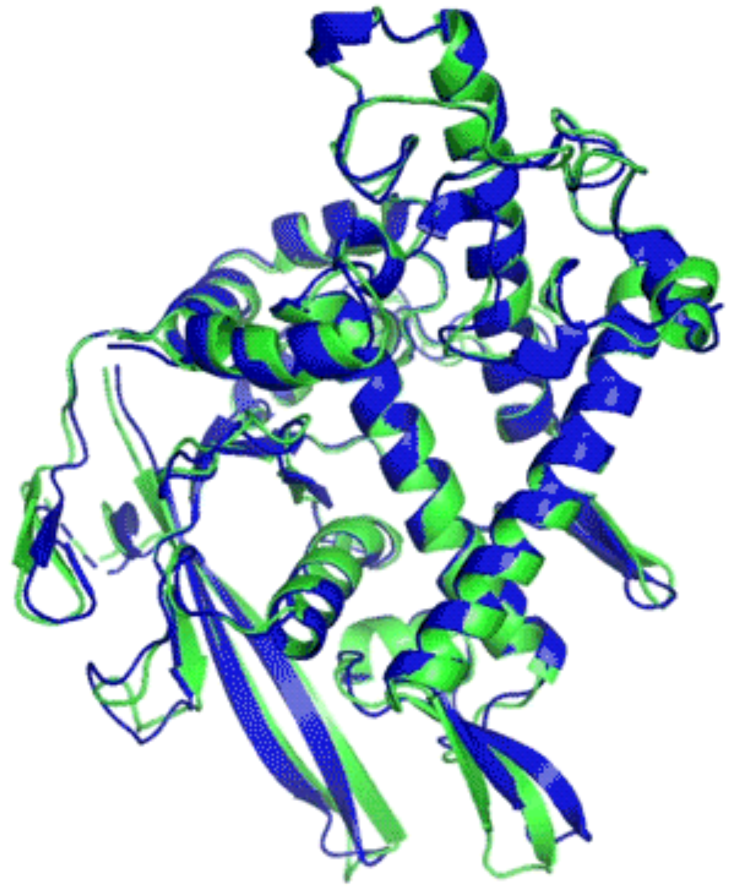
Extracting Signal Electron Trajectories in the COMET Phase-I Cylindrical Drift Chamber Using Deep Learning
Fumihiro Kaneko, Yoshitaka Kuno (Osaka U.), Joe Sato (Yokohama Natl. U.), Ikuya Sato (Saitama U.), Dorian Pieters (Osaka U.) et al. (Aug 8, 2024)
e-Print: 2408.04795 [hep-ex]

AI for Nuclear Physics: the EXCLAIM project
Simonetta Liuti, Douglas Adams, Marie Boër, Gia-Wei Chern, Marija Cuic et al. (Jul 31, 2024)
e-Print: 2408.00163 [hep-ph]

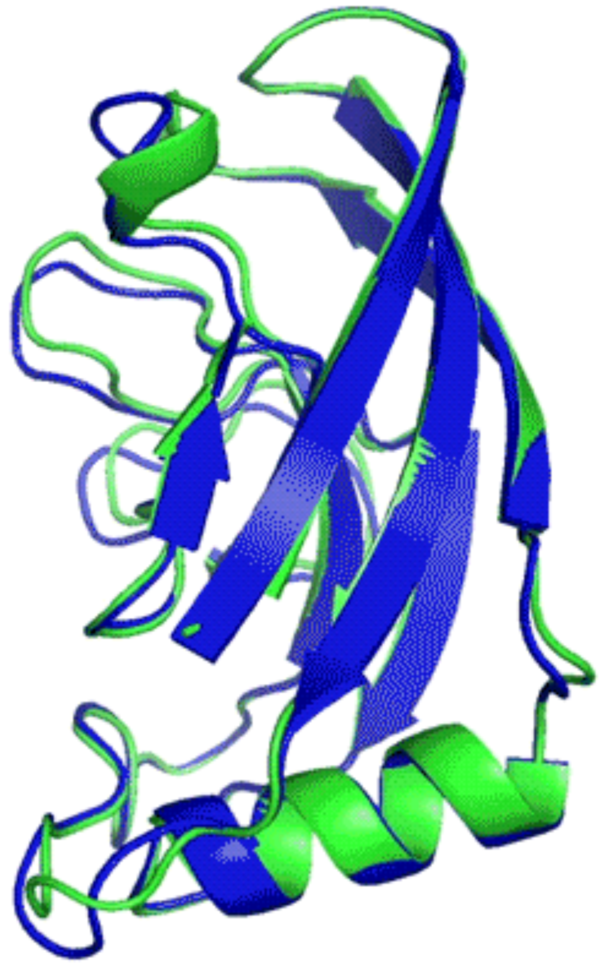
Deep learning-driven likelihood-free parameter inference for 21-cm forest observations
Tian-Yang Sun, Yue Shao, Yichao Li, Yidong Xu, Xin Zhang (Jul 19, 2024)
e-Print: 2407.14298 [astro-ph.CO]

Unruh-DeWitt Quantum Computing: Realizing Quantum Shannon Theory With Quantum Fields
Eric W. Aspling (SUNY, Binghamton) (Jul 18, 2024)
e-Print: 2407.13628 [quant-ph]

Standard-Model Event Discovery at the LHC with a Novel Quantum Autoencoder
Callum Duffy (University Coll. London), Mohammad Hassanshah (University Coll. London), Marcin Jastrzebski (University Coll. Lond
e-Print: 2407.07961 [quant-ph]



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

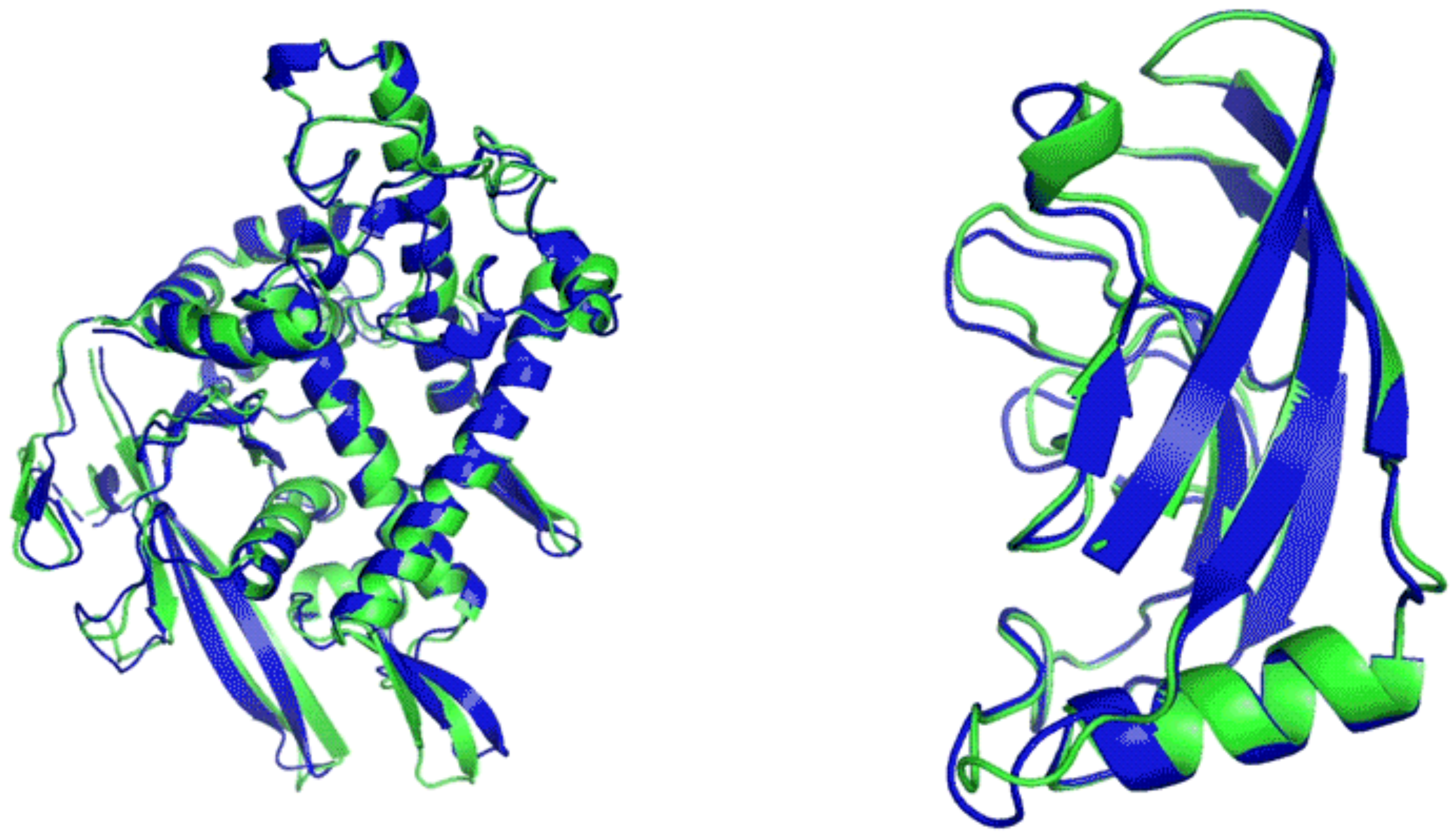


T1049 / 6y4f
93.3 GDT
(adhesin tip)

🟢 Experimental result
🔵 Computational prediction

AlphaFold nature cover

sequence—the structure prediction component of the ‘protein folding problem’[8]—has been an important open research problem for more than 50 years[9]. Despite recent



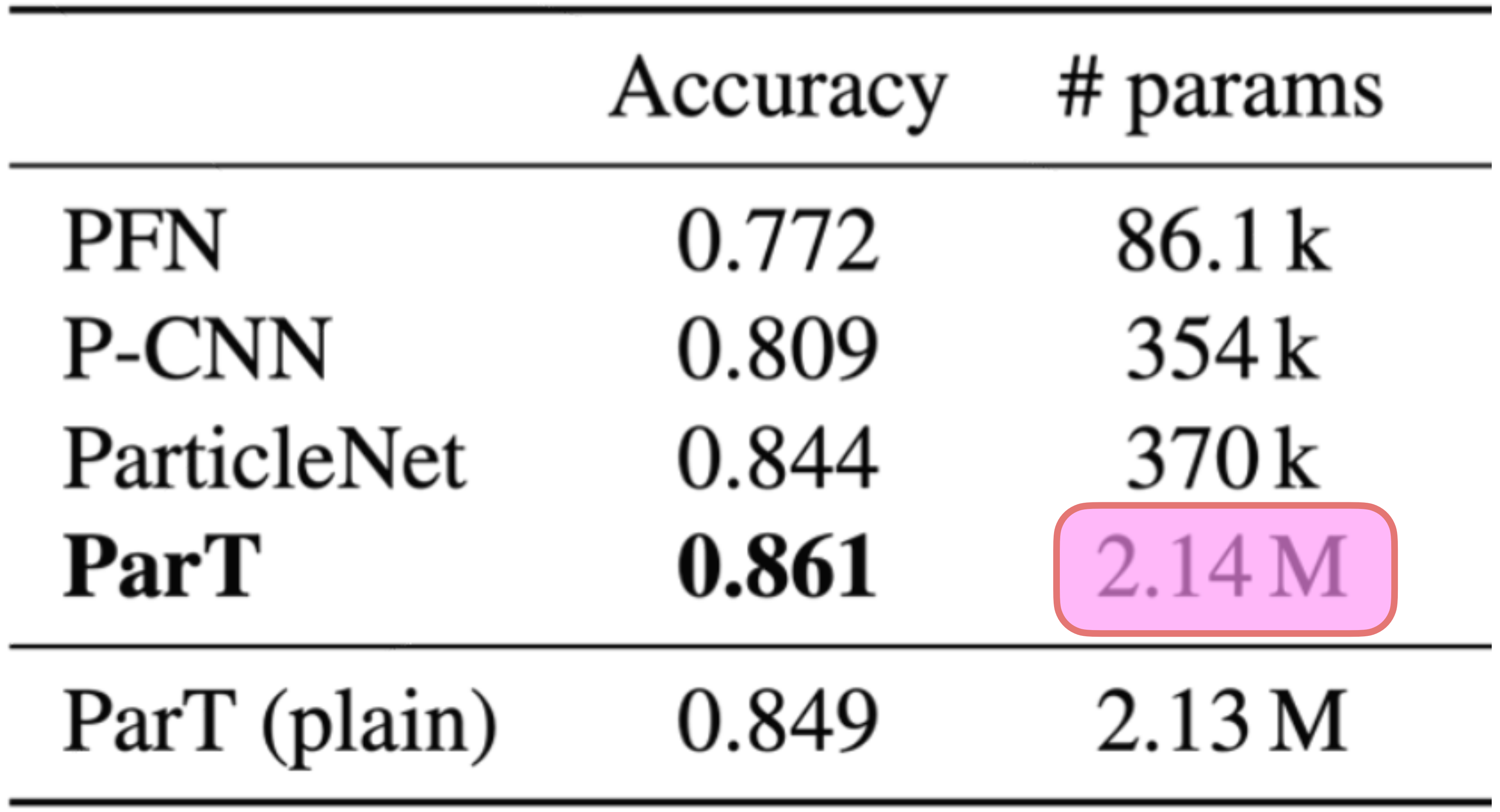
T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

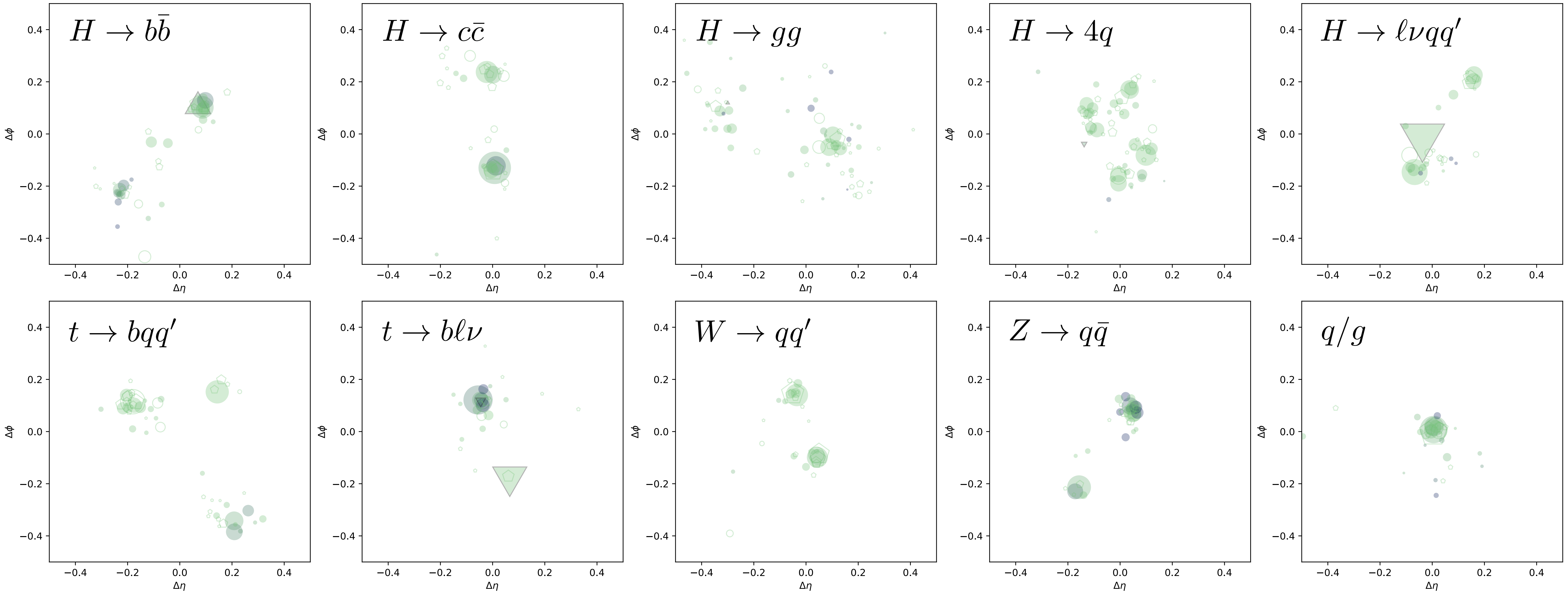
● Experimental result
● Computational prediction

sequence—the structure prediction component of the 'protein folding problem'[8]—has been an important open research problem for more than 50 years[9]. Despite recent

100 million jets for training

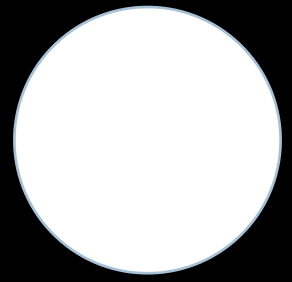


	Accuracy	# params
PFN	0.772	86.1 k
P-CNN	0.809	354 k
ParticleNet	0.844	370 k
ParT	**0.861**	2.14 M
ParT (plain)	0.849	2.13 M



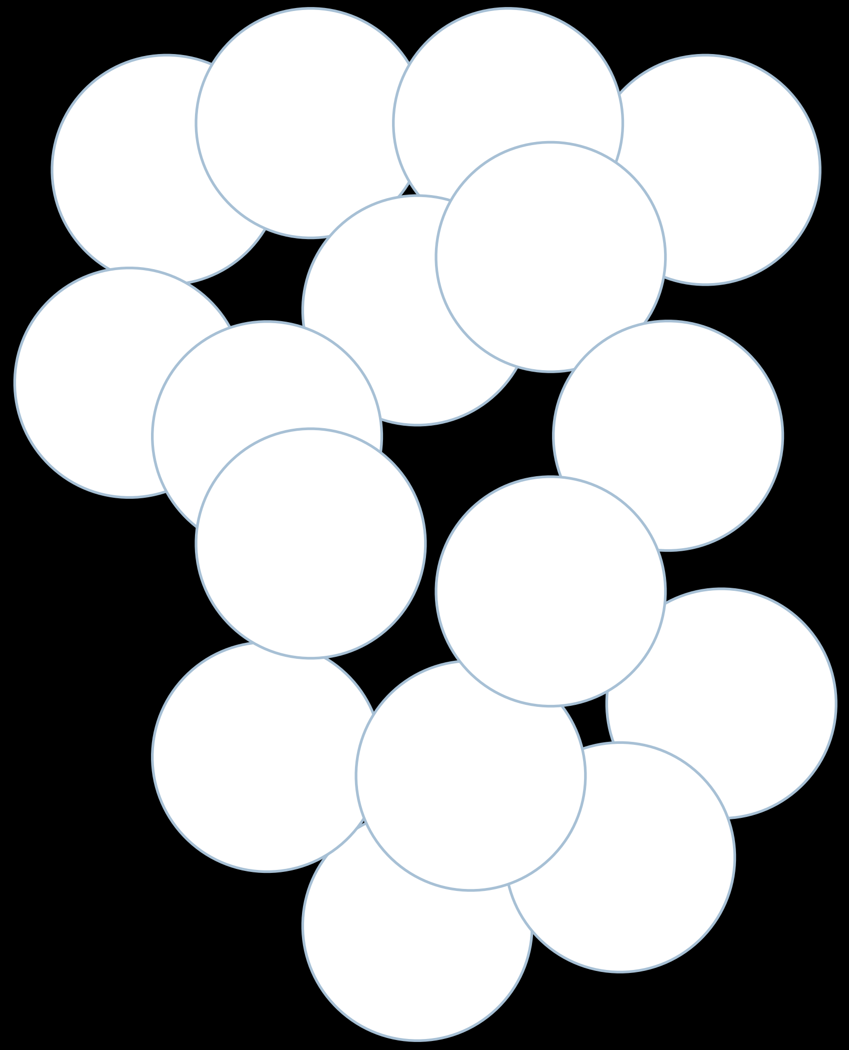
"Particle Transformer For Jet Tagging" H. Qu, C. Li, S. Qian

GPT-3



175,000,000,000
(0.16% of neurons in your brain)

GPT-4 (MoE)



1,800,000,000,000
(1.6% of neurons in your brain)

semianalysis 2023

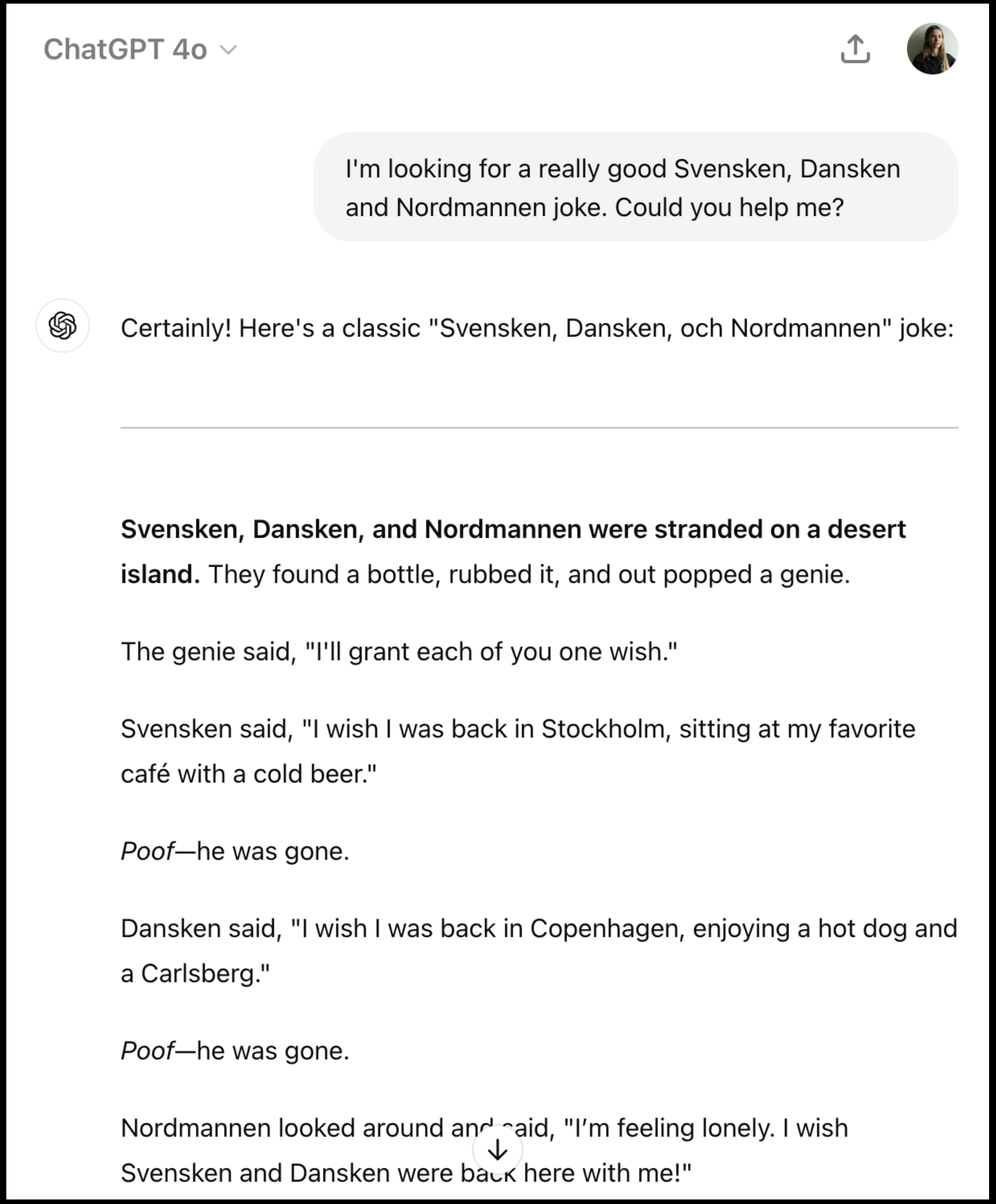
Train (GPT-4):

- 2.15^{25} floating point operations
- ~25,000 A100 GPUs
- 90-100 days
- $63 million
- Trained on 13 trillion tokens

semianalysis 2023

Train (GPT-4):

- 2.15^{25} floating point operations
- ~25,000 A100 GPUs
- 90-100 days
- $63 million
- Trained on 13 trillion tokens



I'm looking for a really good Svensken, Dansken and Nordmannen joke. Could you help me?

Certainly! Here's a classic "Svensken, Dansken, och Nordmannen" joke:

Svensken, Dansken, and Nordmannen were stranded on a desert island. They found a bottle, rubbed it, and out popped a genie.

The genie said, "I'll grant each of you one wish."

Svensken said, "I wish I was back in Stockholm, sitting at my favorite café with a cold beer."

Poof—he was gone.

Dansken said, "I wish I was back in Copenhagen, enjoying a hot dog and a Carlsberg."

Poof—he was gone.

Nordmannen looked around and said, "I'm feeling lonely. I wish Svensken and Dansken were back here with me!"

Inference (GPT-4):

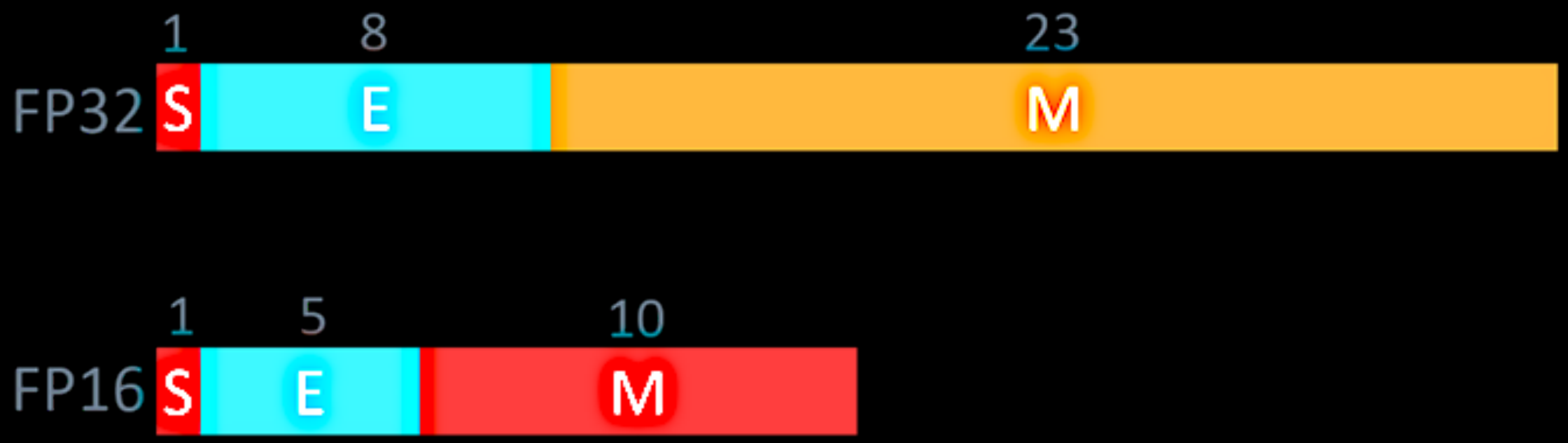
- Multiple clusters of 128 GPUs
- Model carefully mapped onto hardware

GPT-3

↓

175,000,000,000

FP16 vs FP32



	Range	Accuracy
FP32	10^{-38} - 10^{38}	0.000006%
FP16	$6x10^{-5}$ - $6x10^{4}$	0.05%

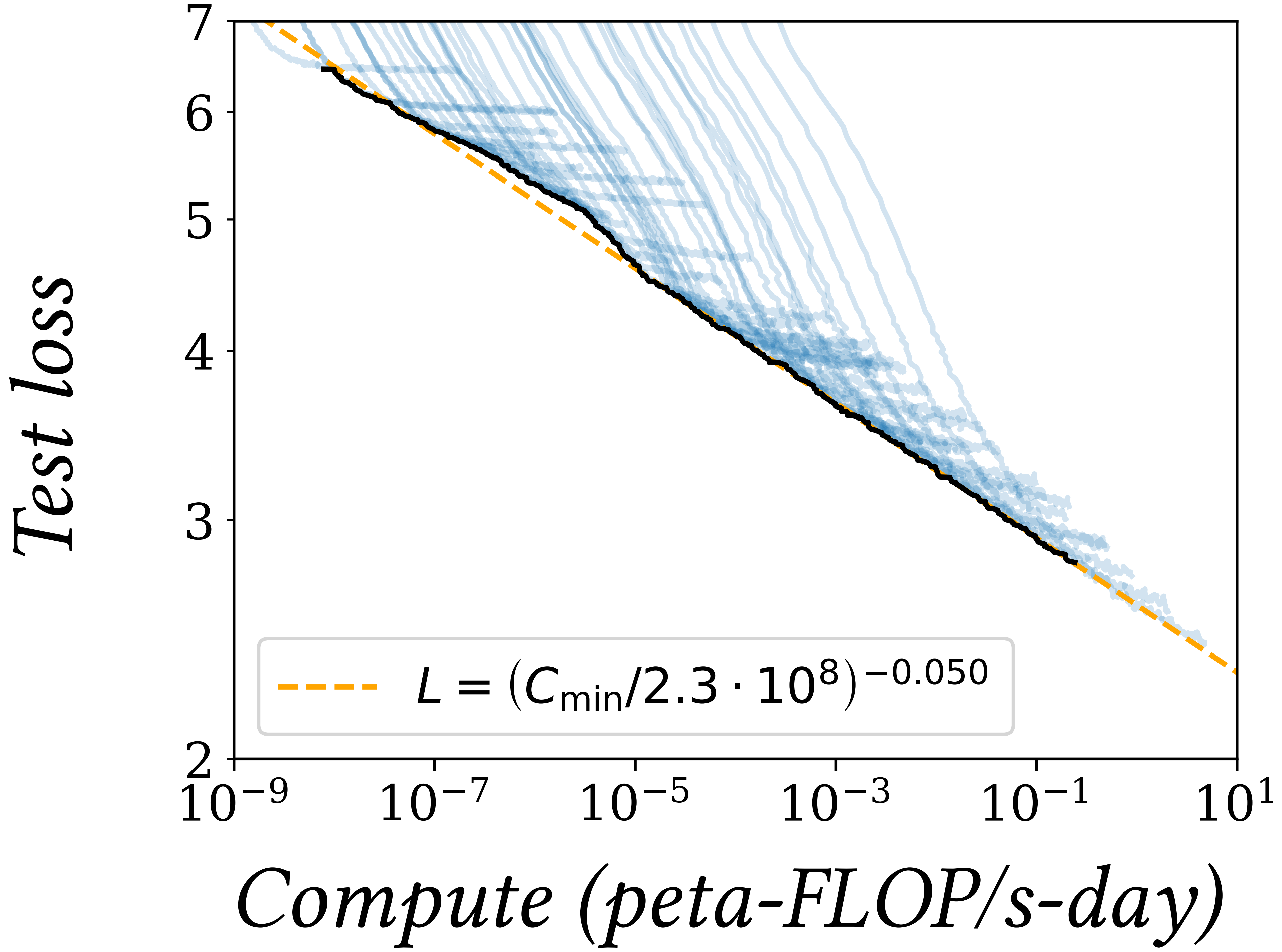
→ ~700 GB
(175B × 4B/par)
→ $O(10^1)$> single GPU memory

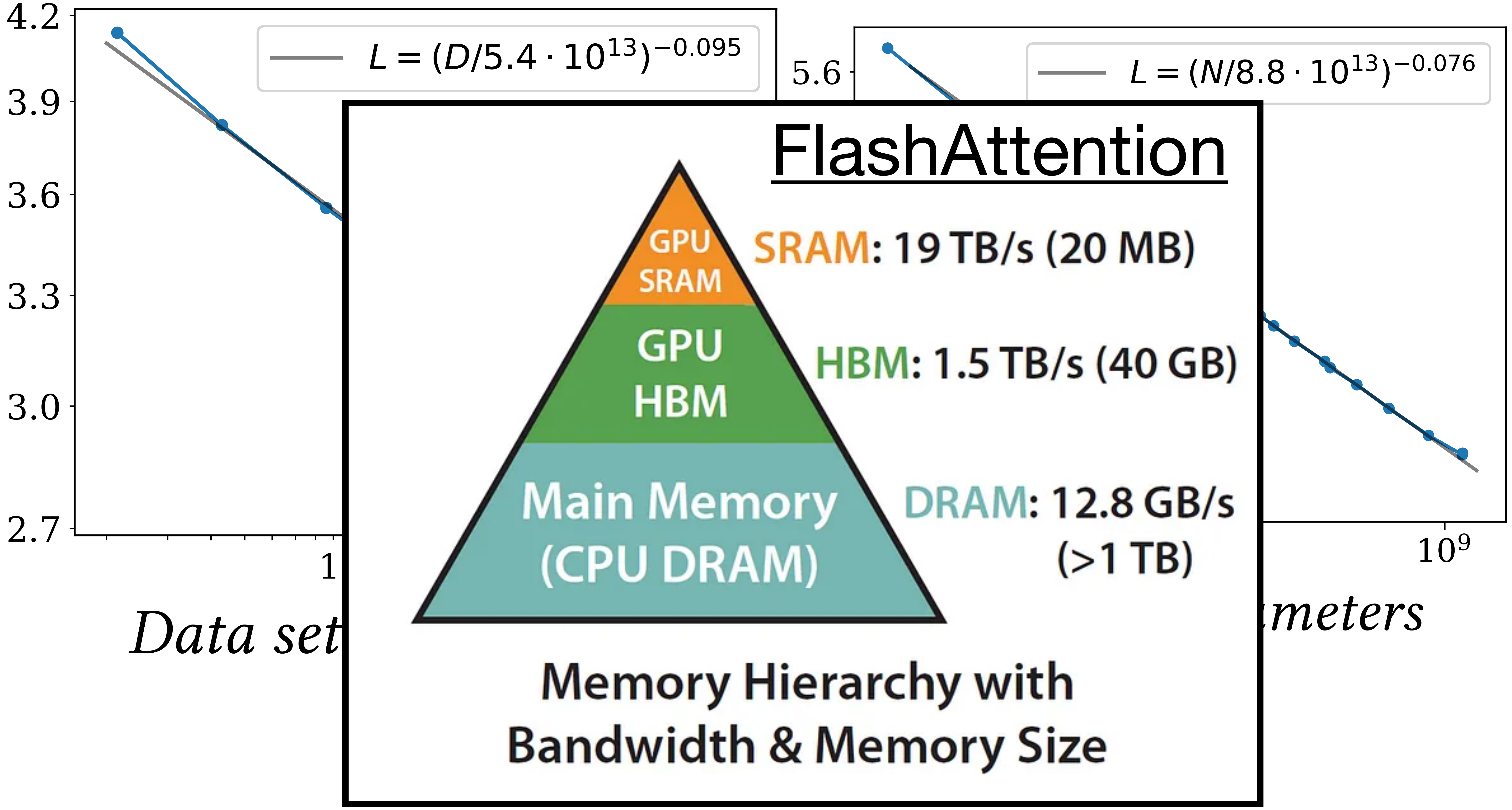
→ ~350 GB
(175B × 2B/par)
→ 11 NVIDIA V100 ($10 000/ea)

5 of these to fit one GPT-3 at inference time! Price: $17,000

AI & Memory Wall

Kaplan et al. (2020)





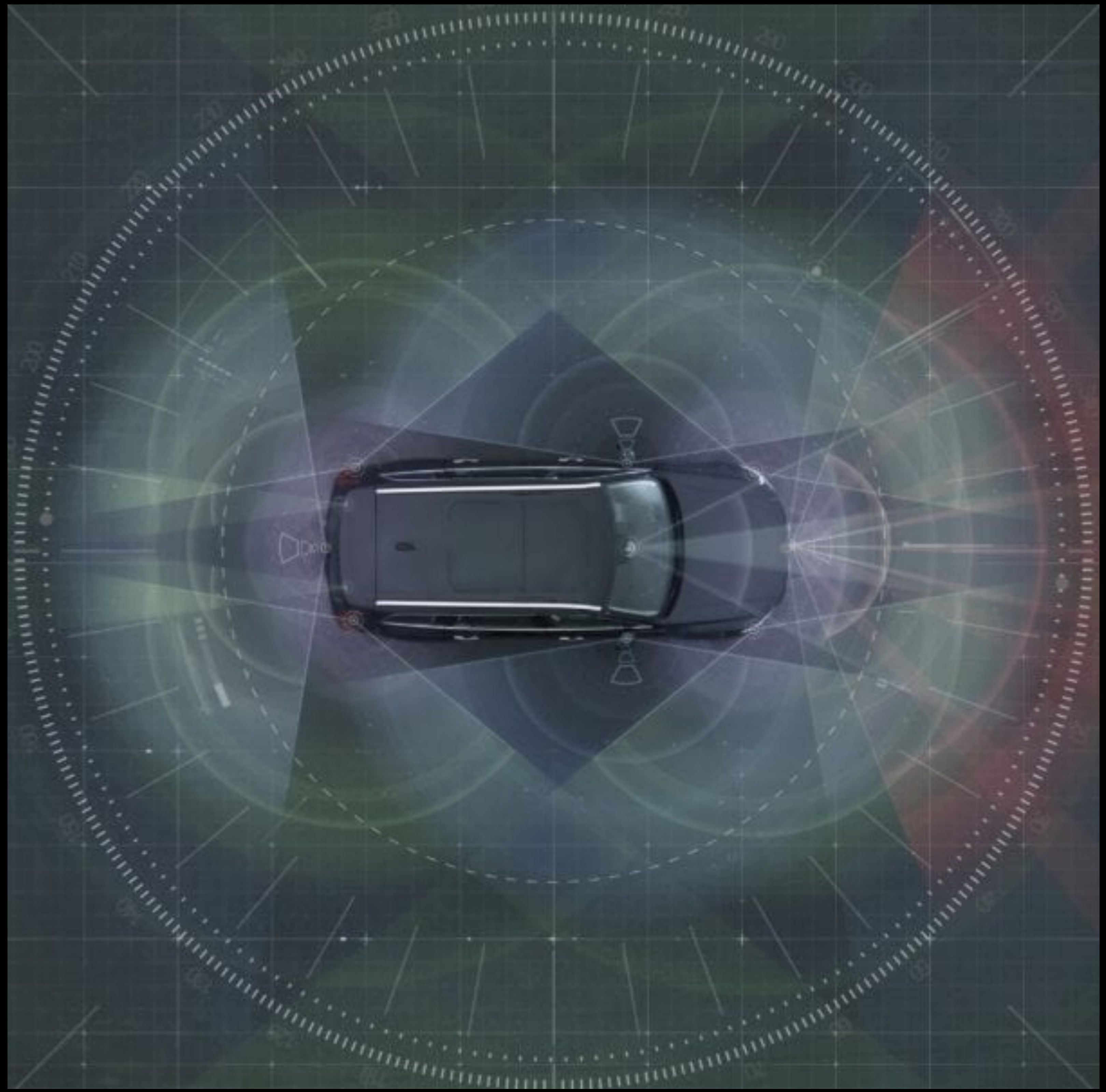
FlashAttention

Memory Hierarchy with
Bandwidth & Memory Size

Kaplan et al. (2020)

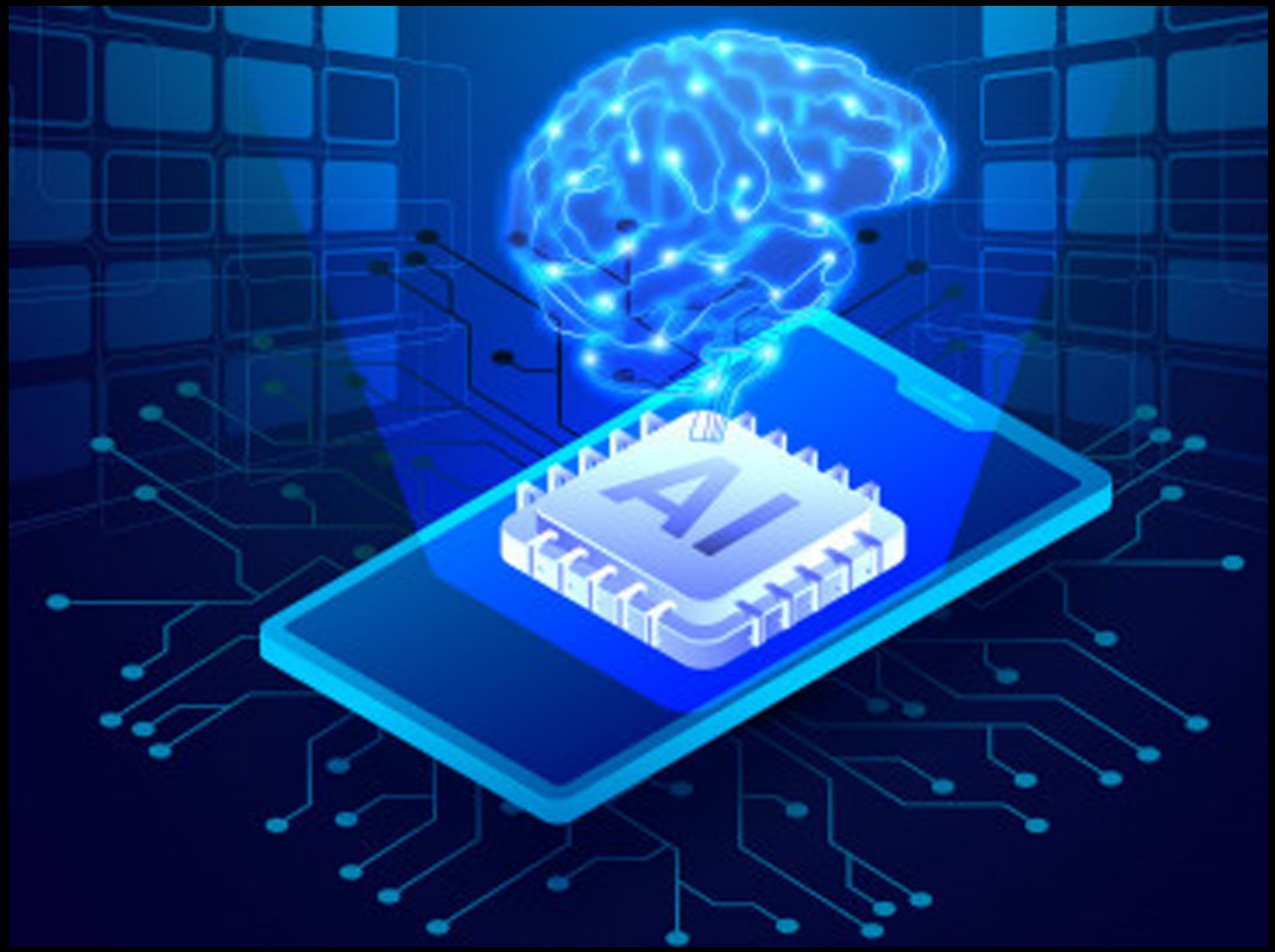
EDGE

O(1) ms



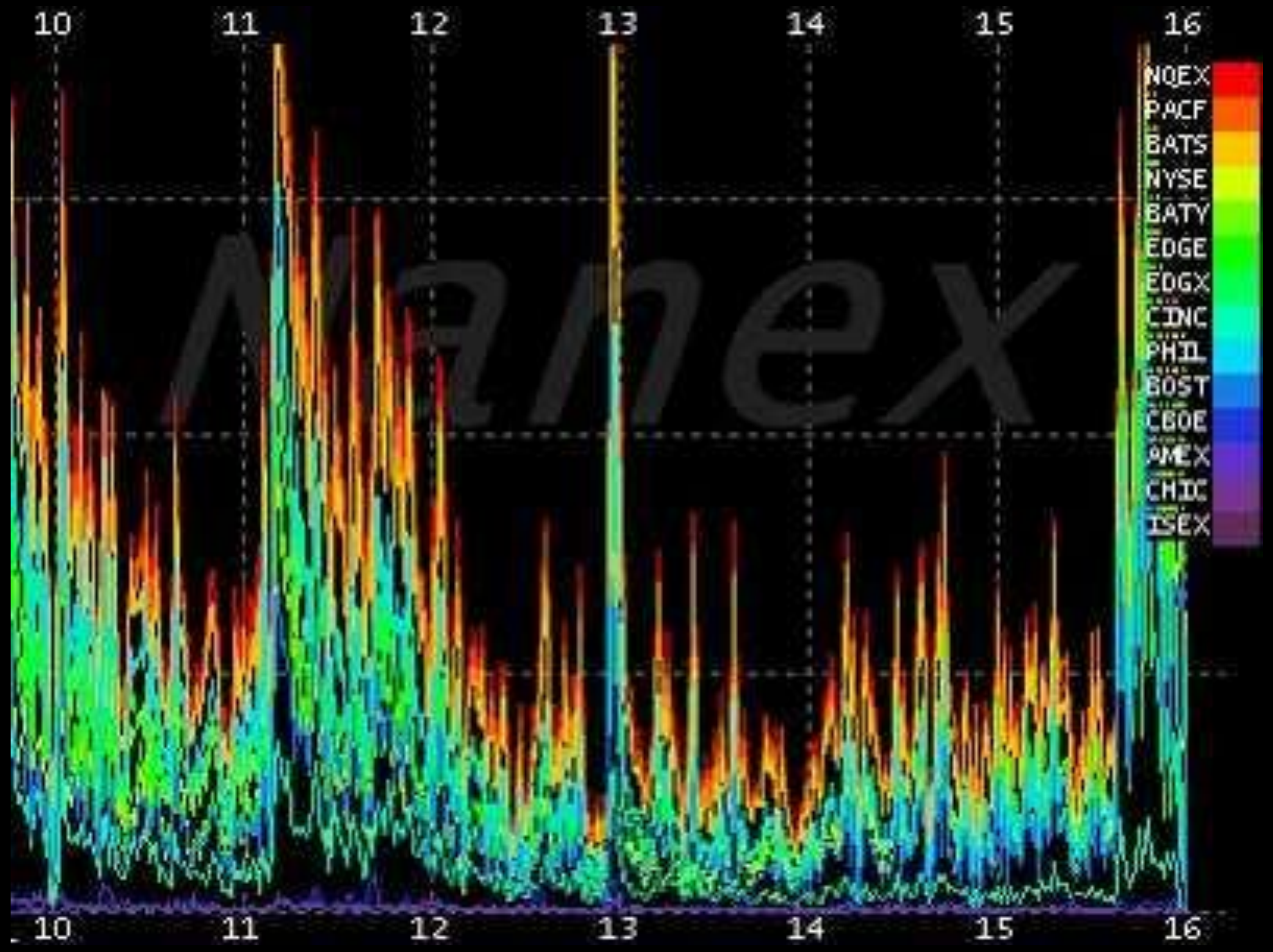
ASIC/GPU

O(1) ms



ASIC

O(1) ns



FPGA

Low power

On-device

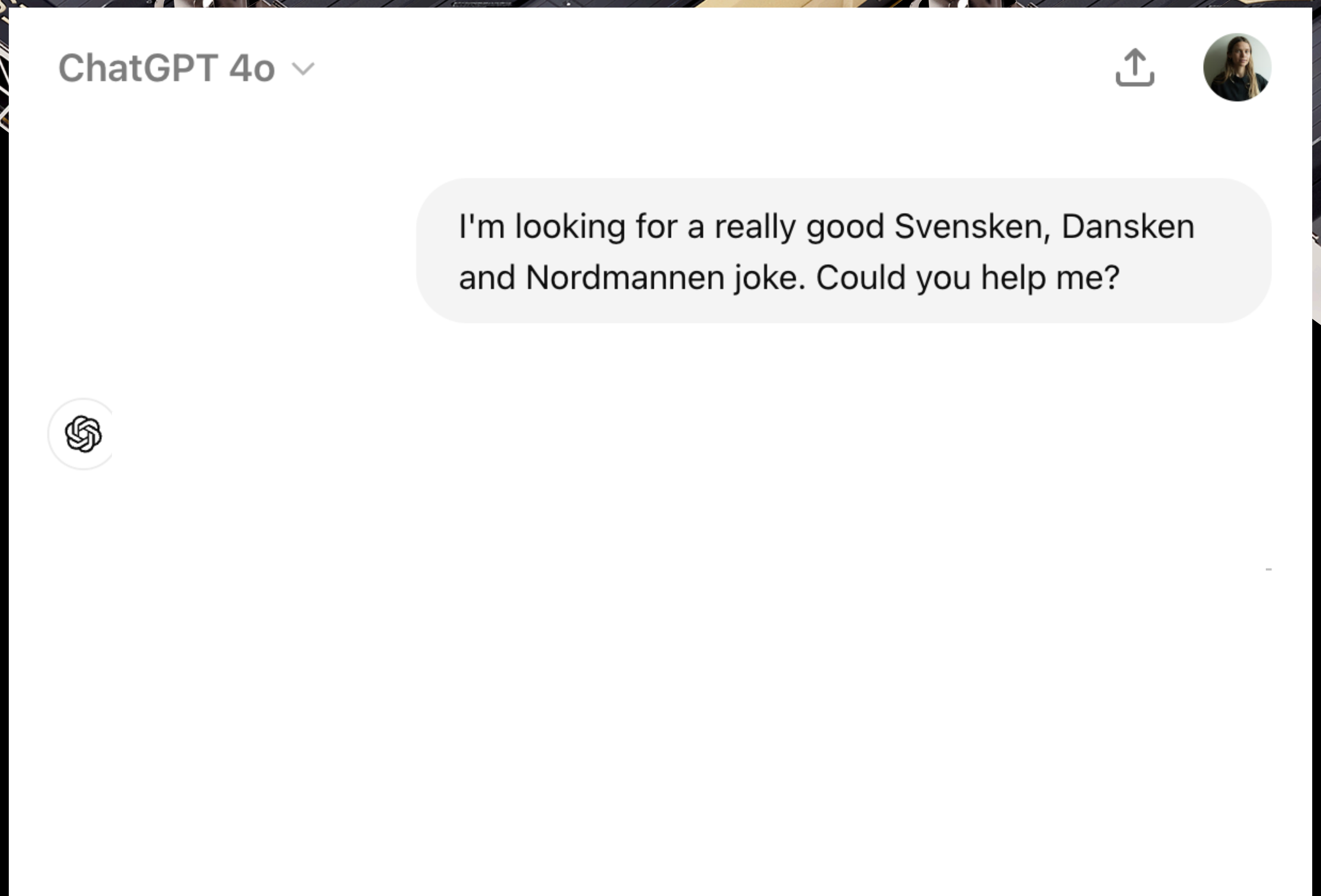
High-throughput

EFFICIENT ML ALGORITHMS

(Data-efficient)

Low-latency

Resources: 128 interconnected GPUs
Latency : 10 seconds

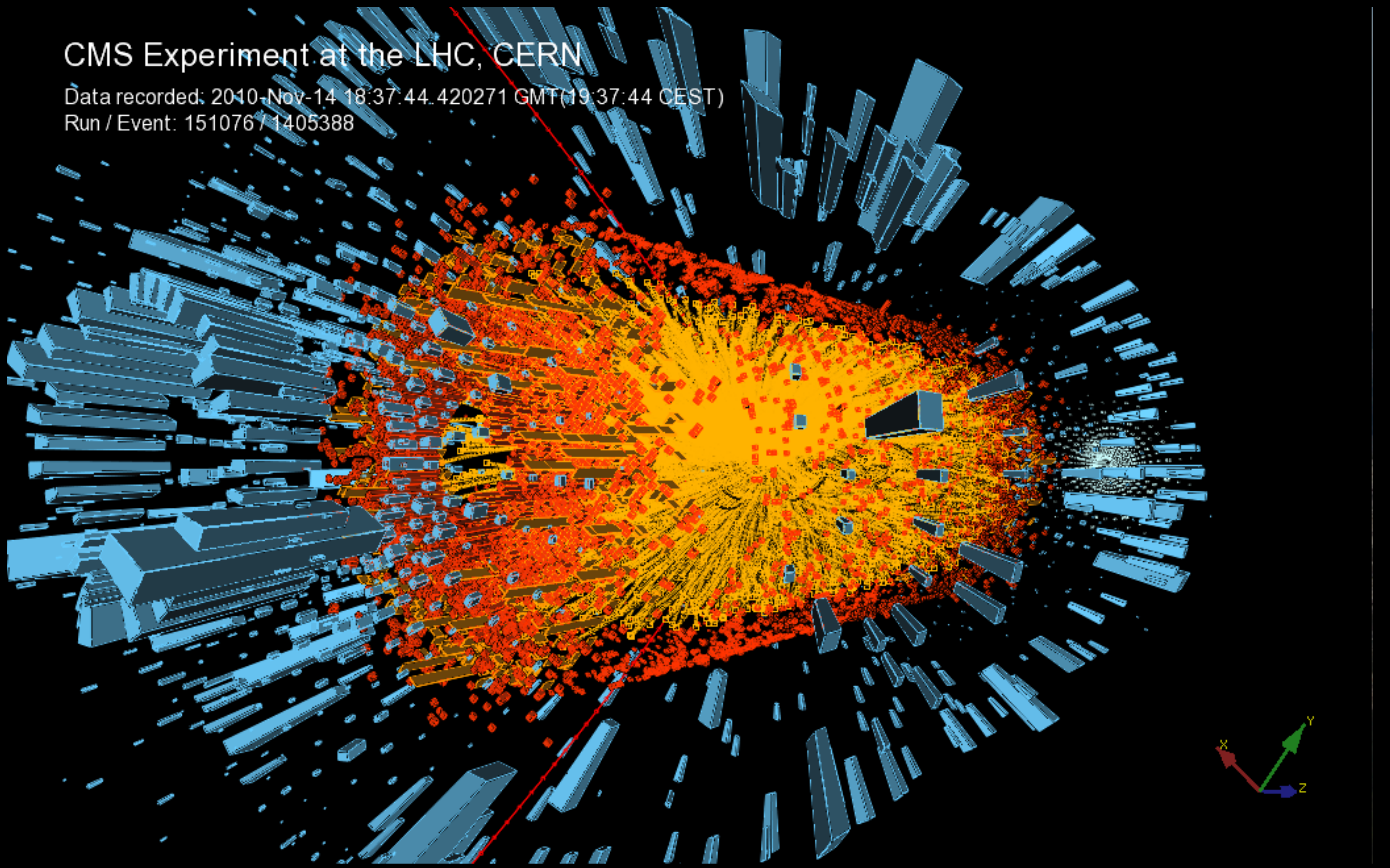


I'm looking for a really good Svensken, Dansken and Nordmannen joke. Could you help me?

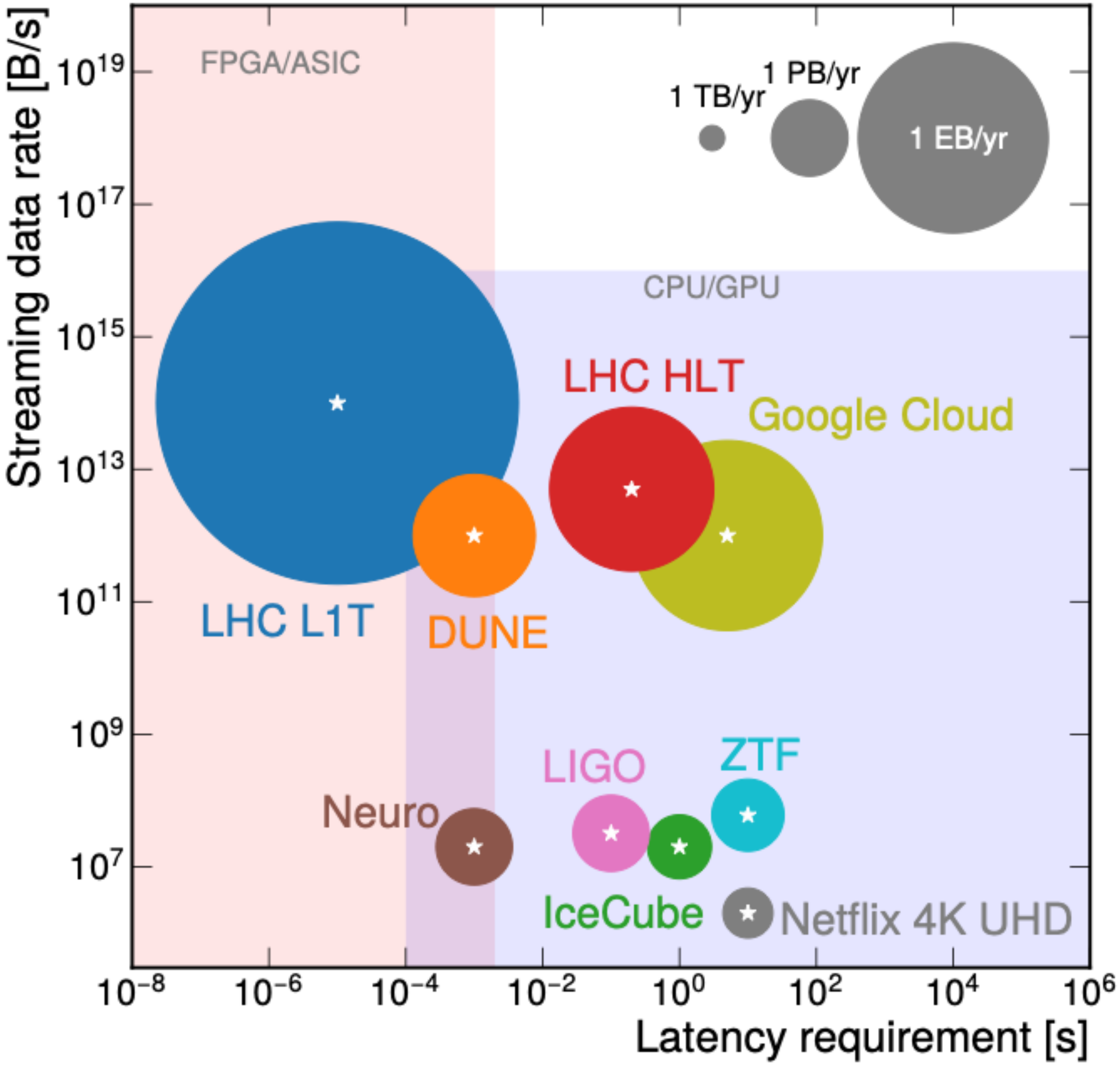
Resources: ***128 interconnected GPUs***
Latency : ***10 seconds***

I'm looking for a really good Svensken, Dansken and Nordmannen joke. Could you help me?

Resources: ***1 single chips***
Latency: ***1 millionth of a second***
Throughput: ***5% of internet traffic***



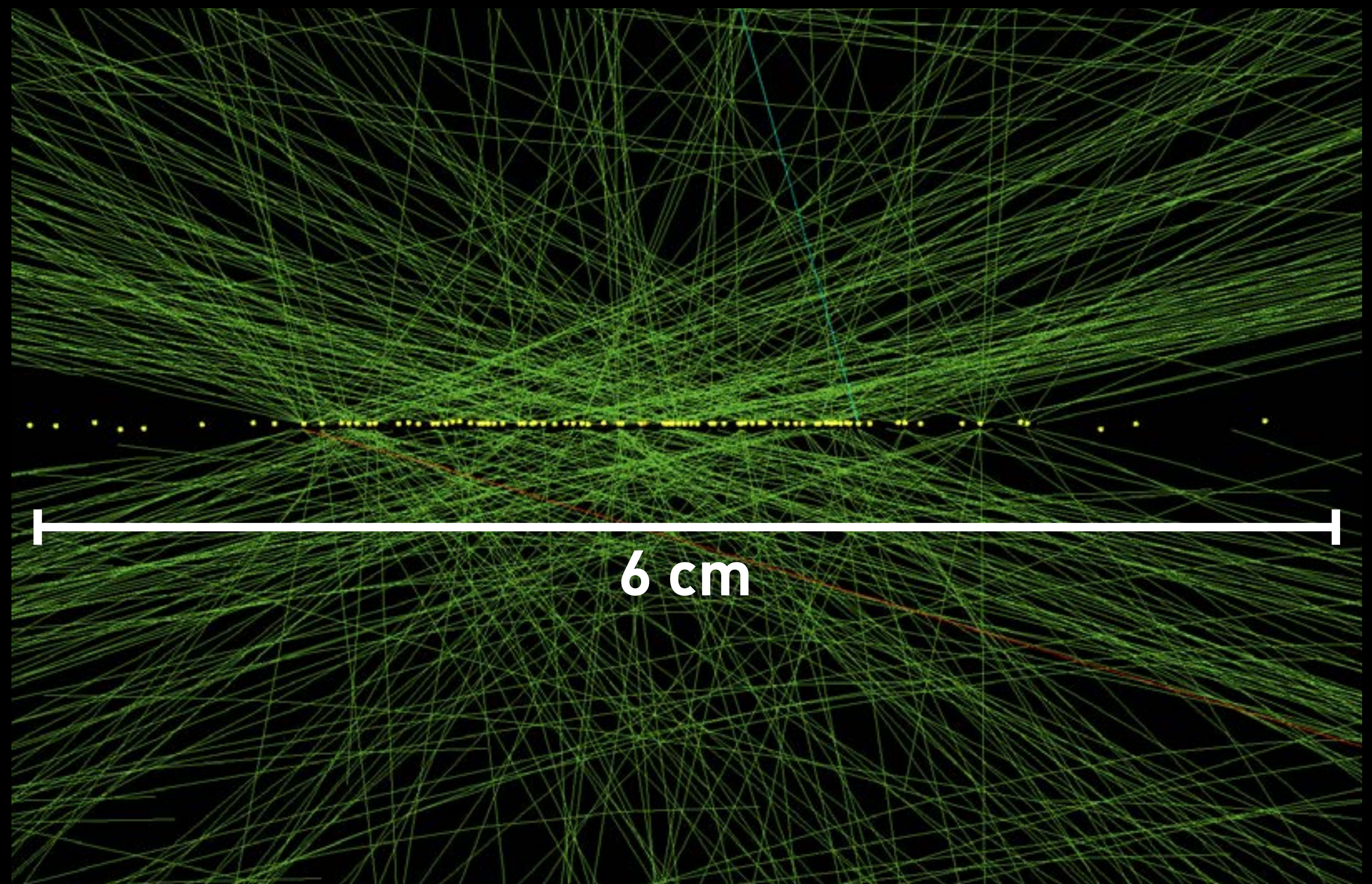
A3D3 Institute




Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
LHC

2,500 bunches of
100 billion protons,
11,000 rotations per second


10^{11} protons
6.8 TeV
25 ns
7.5 m
~60 pp collisions
per crossing


6 cm

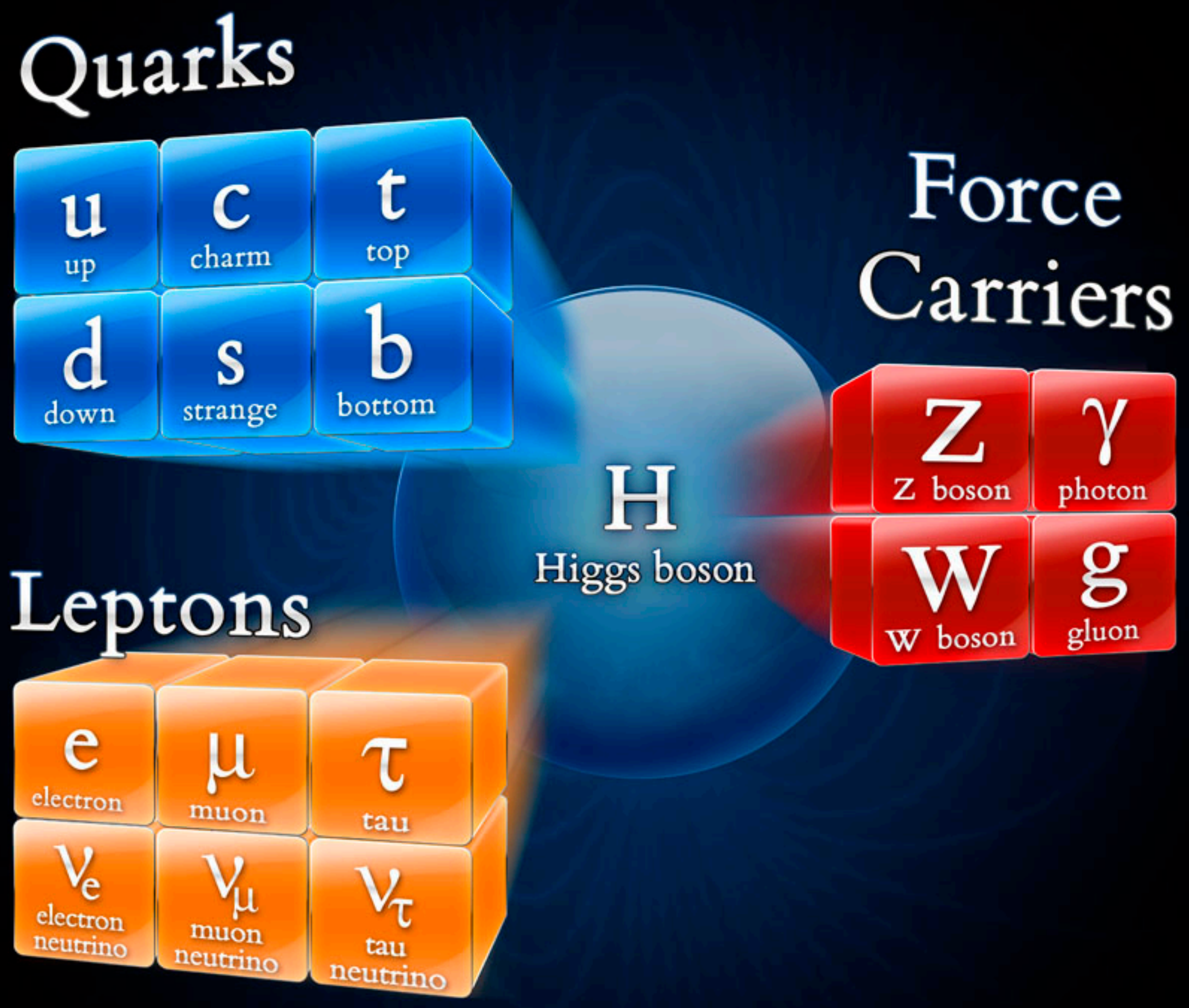

CMS
CMS Experiment at the LHC, CERN
Data recorded: 2010-Nov-14 18:37:44.420271 GMT(19:37:44 CEST)
Run / Event: 151076 / 1405388
X
Y
Z

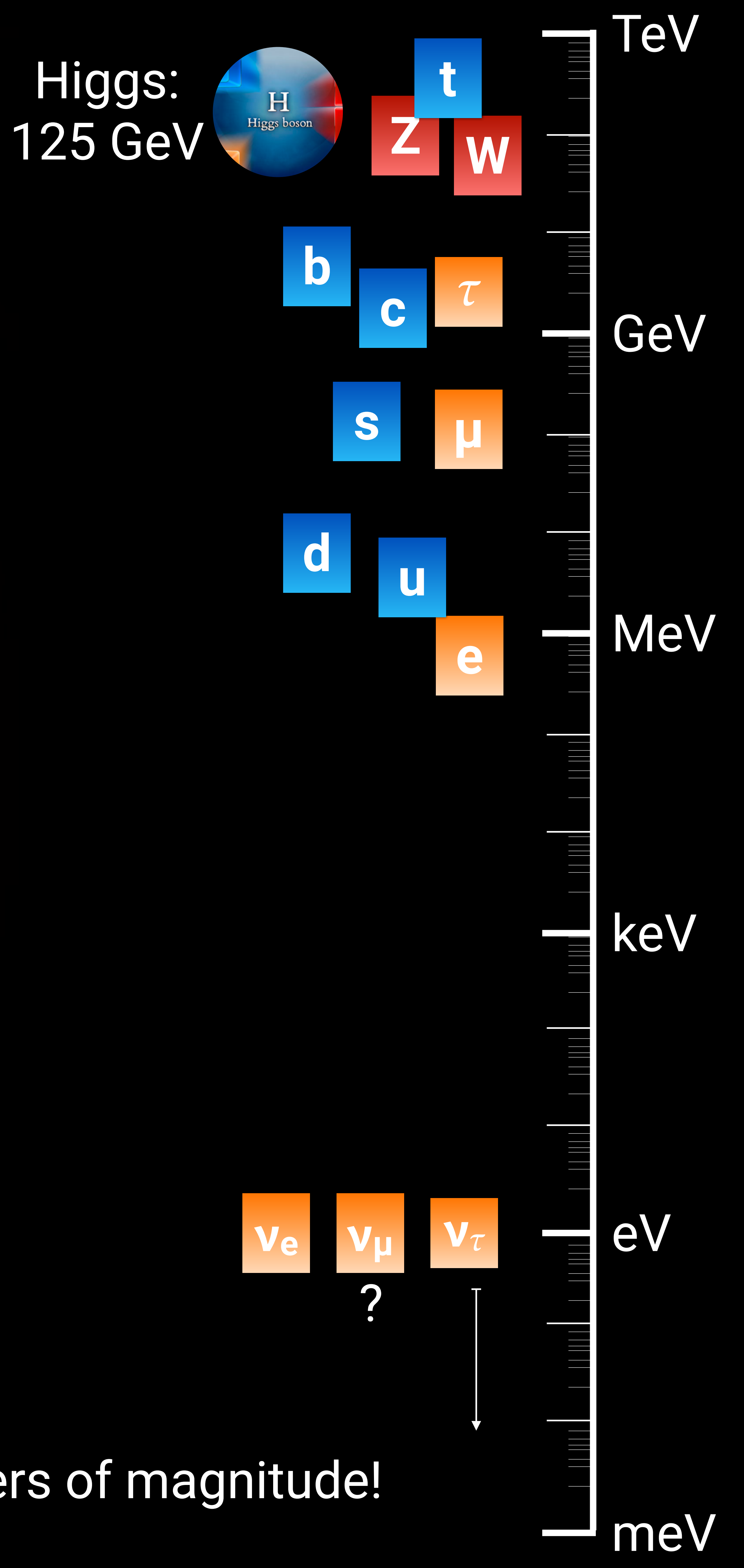
$E=mc^2$

1 billion collisions /s

~1 MB of data / collision

~1 PB of data / s

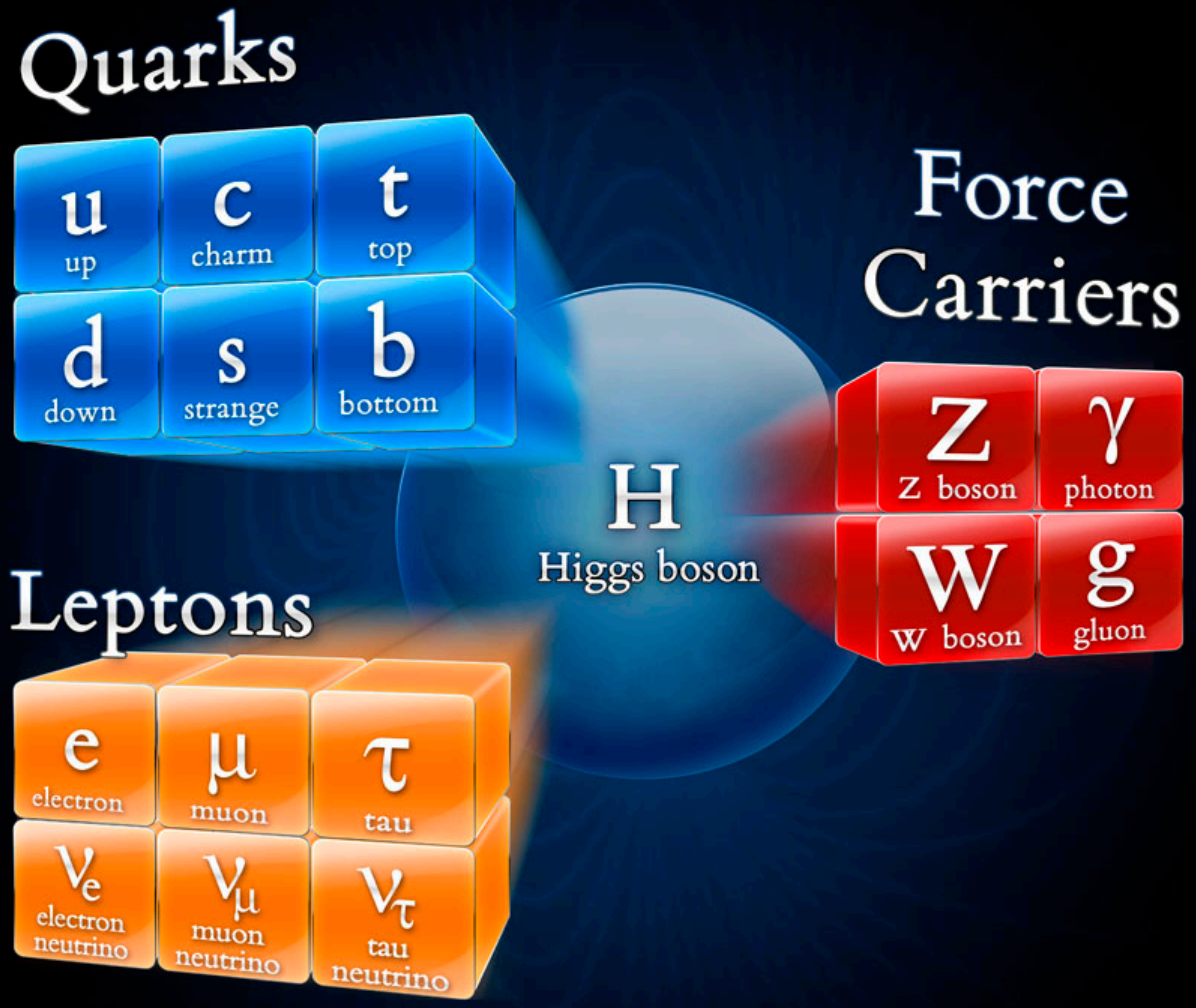




Masses span 9 orders of magnitude!


ZEC
SERVICE


ZEC
SERVICE



Quarks

u up
c charm
t top
d down
s strange
b bottom

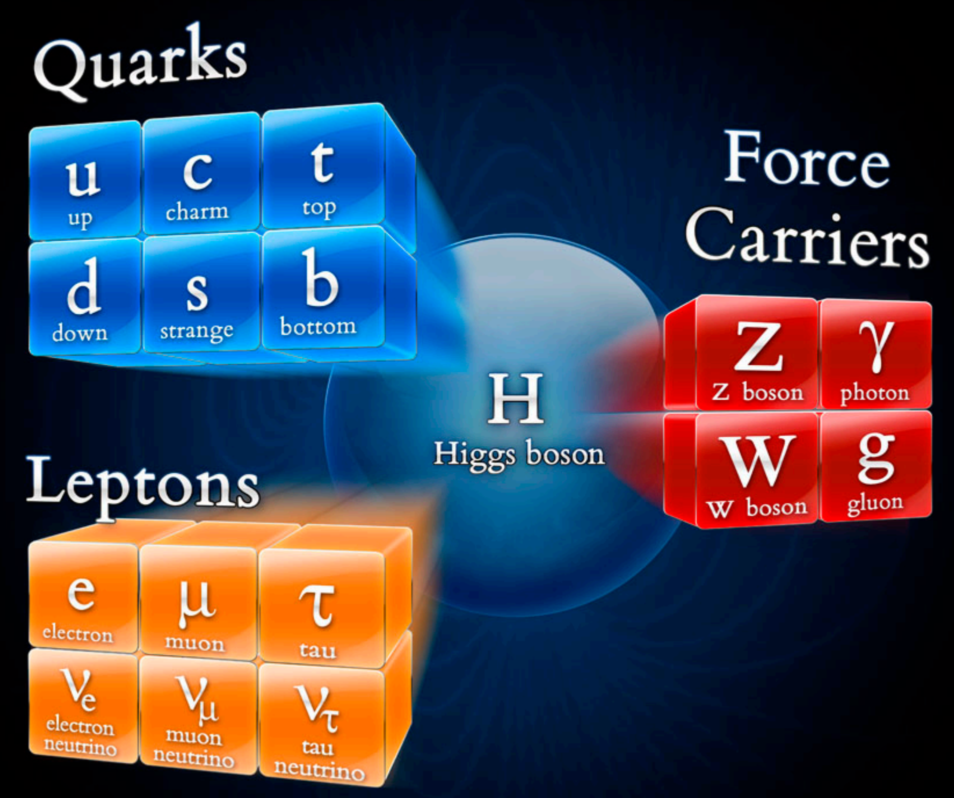
Leptons

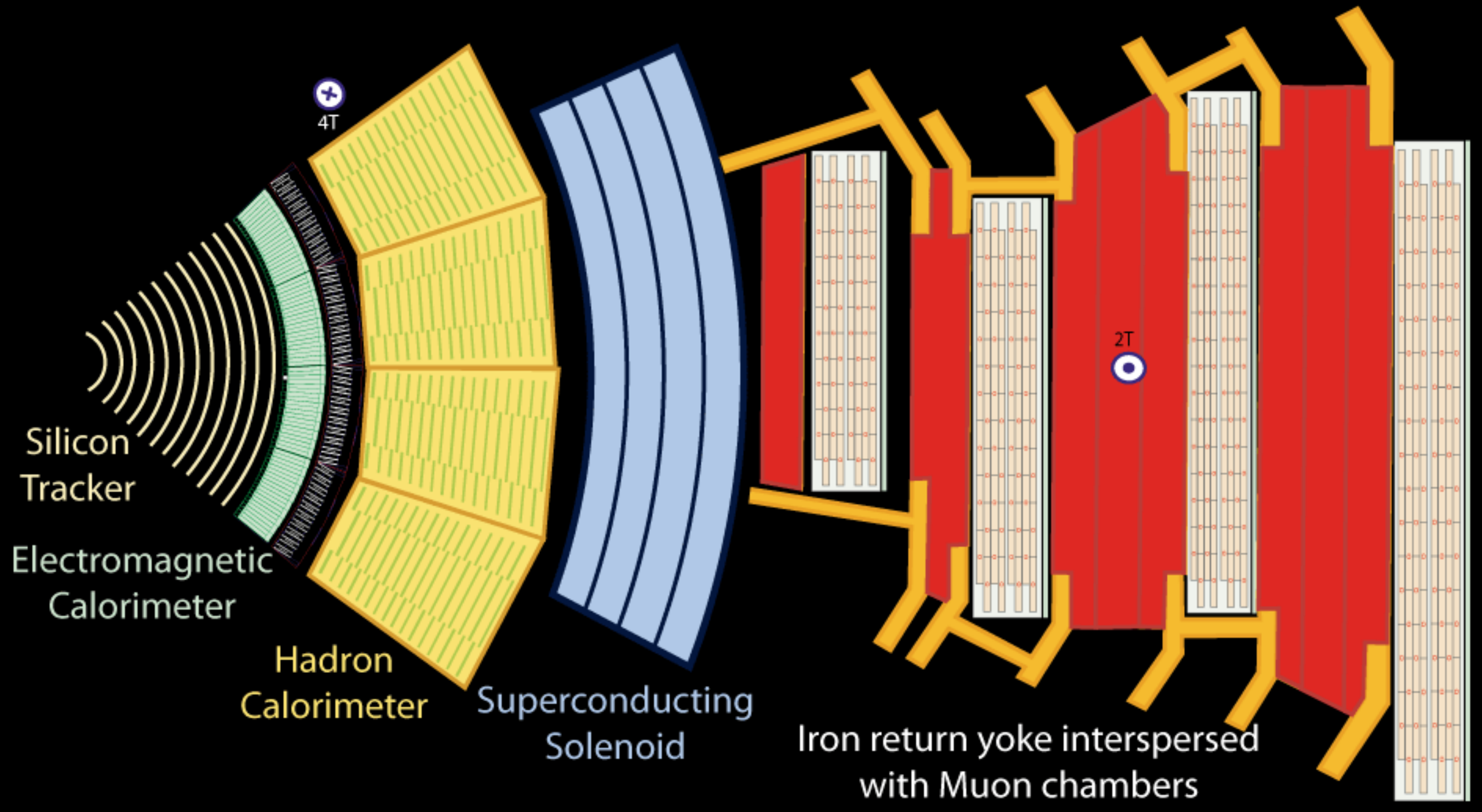
e electron
μ muon
τ tau
ν_e electron neutrino
ν_μ muon neutrino
ν_τ tau neutrino

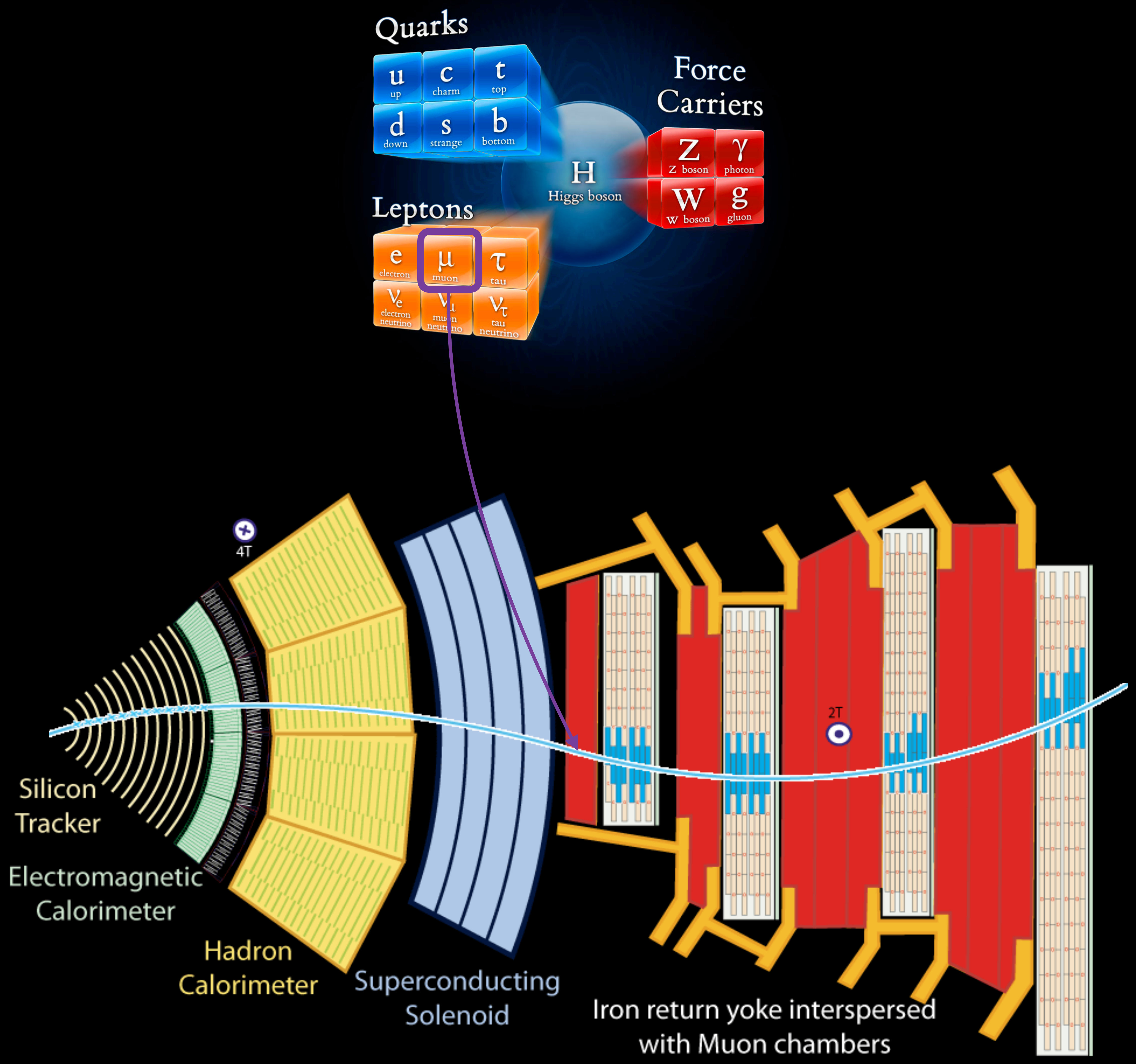
H Higgs boson

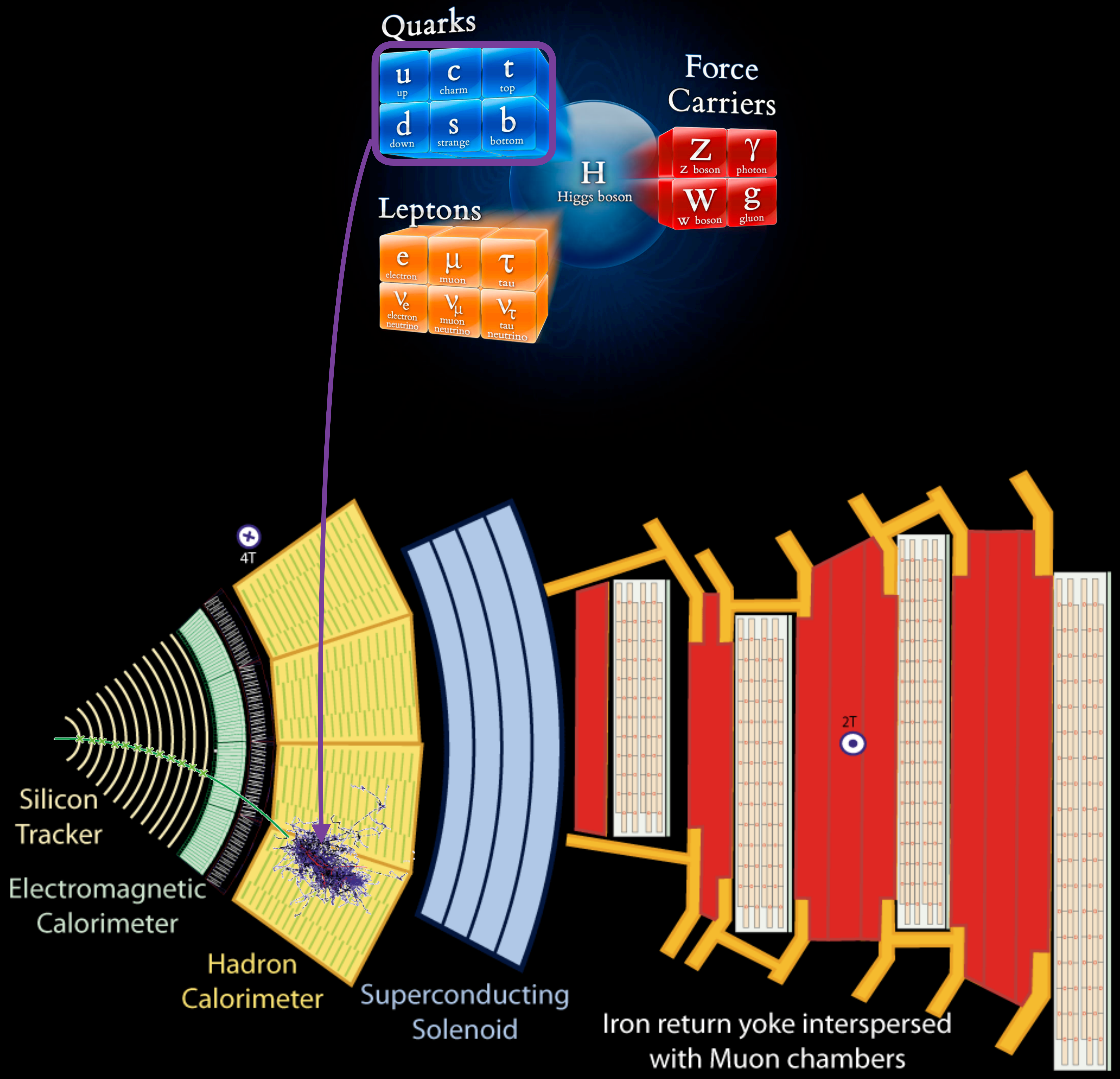
Force Carriers

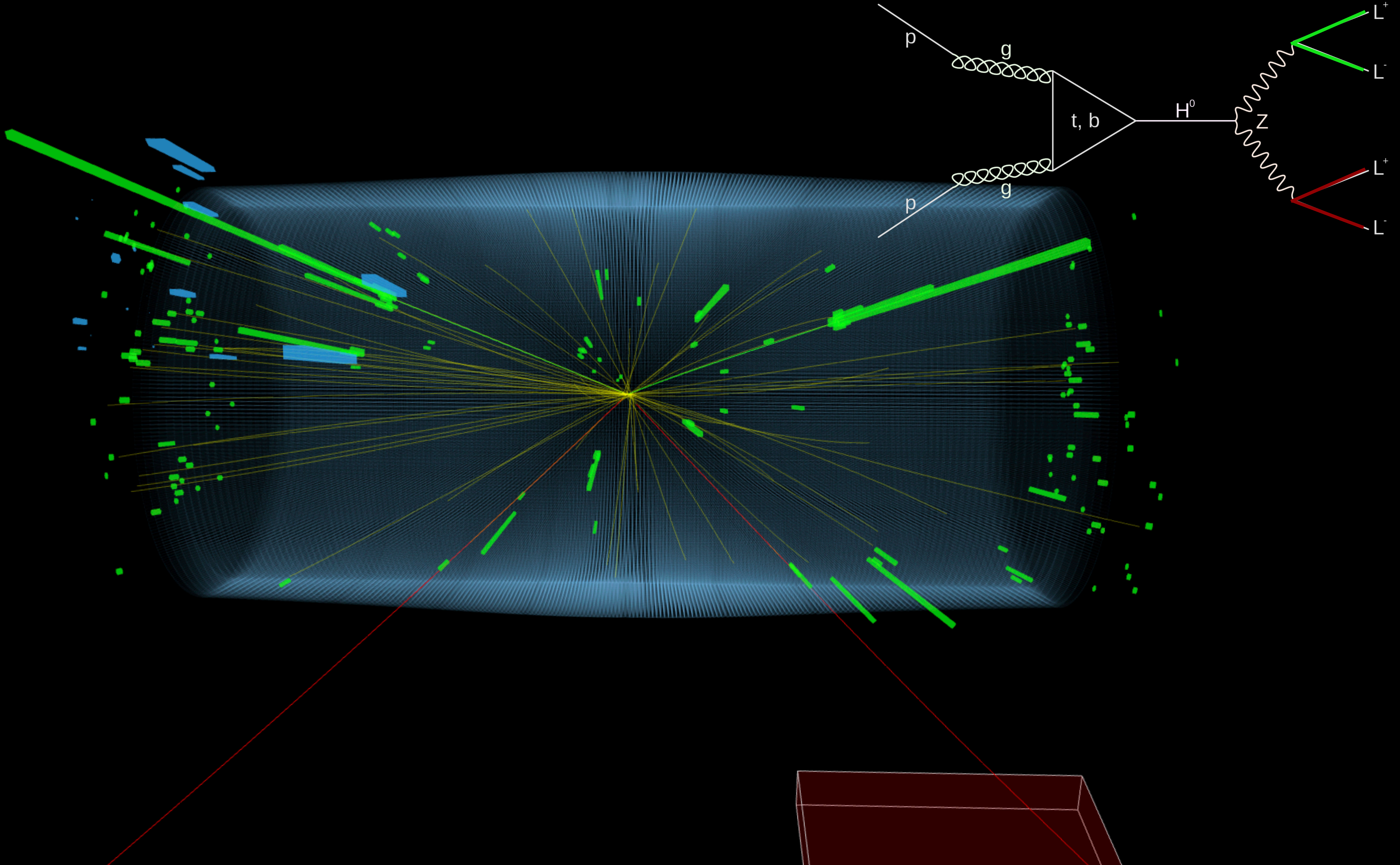
Z Z boson
γ photon
W W boson
g gluon


Quarks
u
up
c
charm
t
top
d
down
s
strange
b
bottom
Leptons
e
electron
μ
muon
τ
tau
νe
electron neutrino
νμ
muon neutrino
ντ
tau neutrino
H
Higgs boson
Force Carriers
Z
Z boson
γ
photon
W
W boson
g
gluon

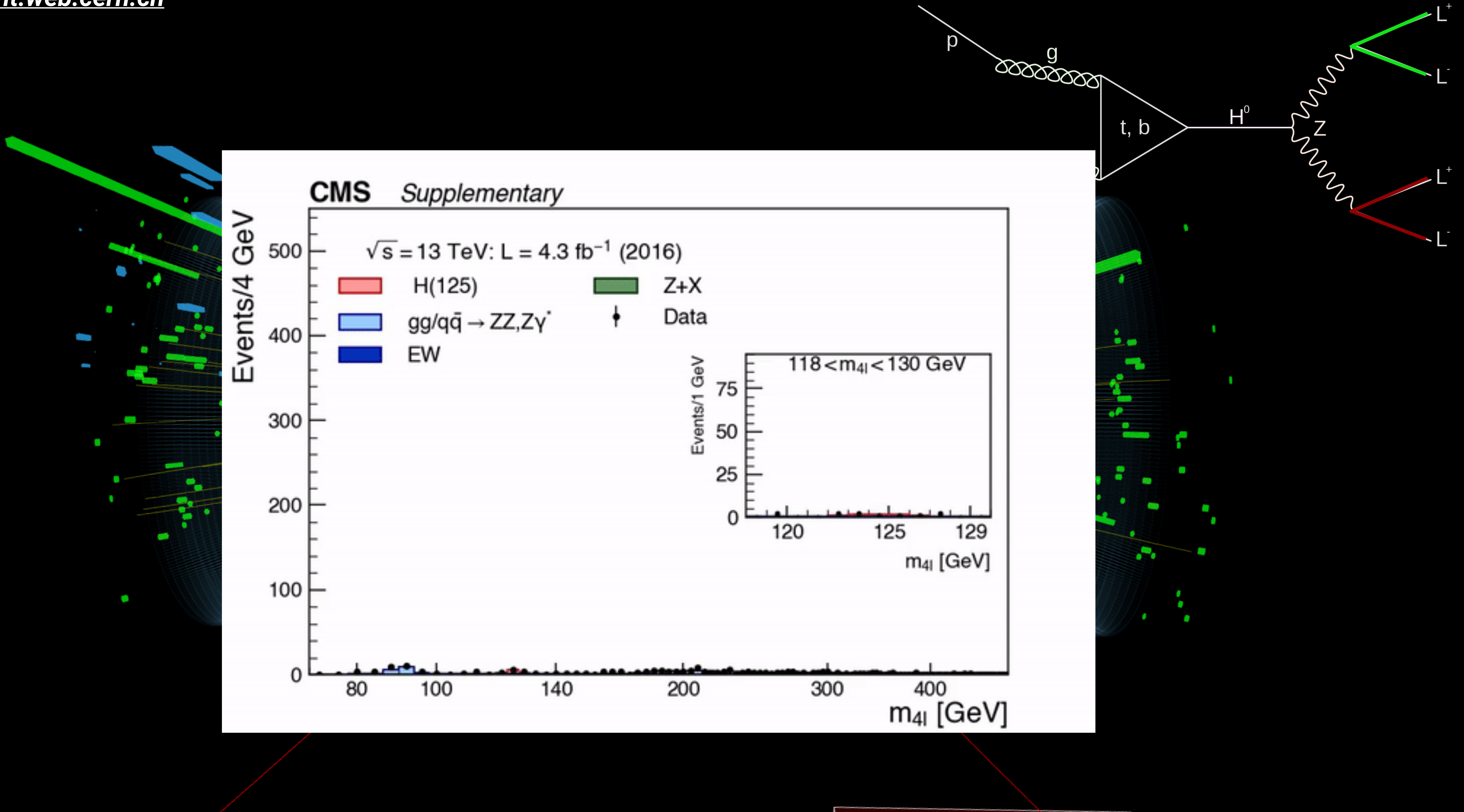

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers


Quarks
u
up
c
charm
t
top
d
down
s
strange
b
bottom
Force
Carriers
Z
Z boson
γ
photon
W
W boson
g
gluon
H
Higgs boson
Leptons
e
electron
μ
muon
τ
tau
νe
electron neutrino
νμ
muon neutrino
ντ
tau neutrino
4T
2T
Silicon
Tracker
Electromagnetic
Calorimeter
Hadron
Calorimeter
Superconducting
Solenoid
Iron return yoke interspersed
with Muon chambers


Quarks
u
up
c
charm
t
top
d
down
s
strange
b
bottom
Leptons
e
electron
μ
muon
τ
tau
νe
electron neutrino
νμ
muon neutrino
ντ
tau neutrino
H
Higgs boson
Force Carriers
Z
Z boson
γ
photon
W
W boson
g
gluon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers

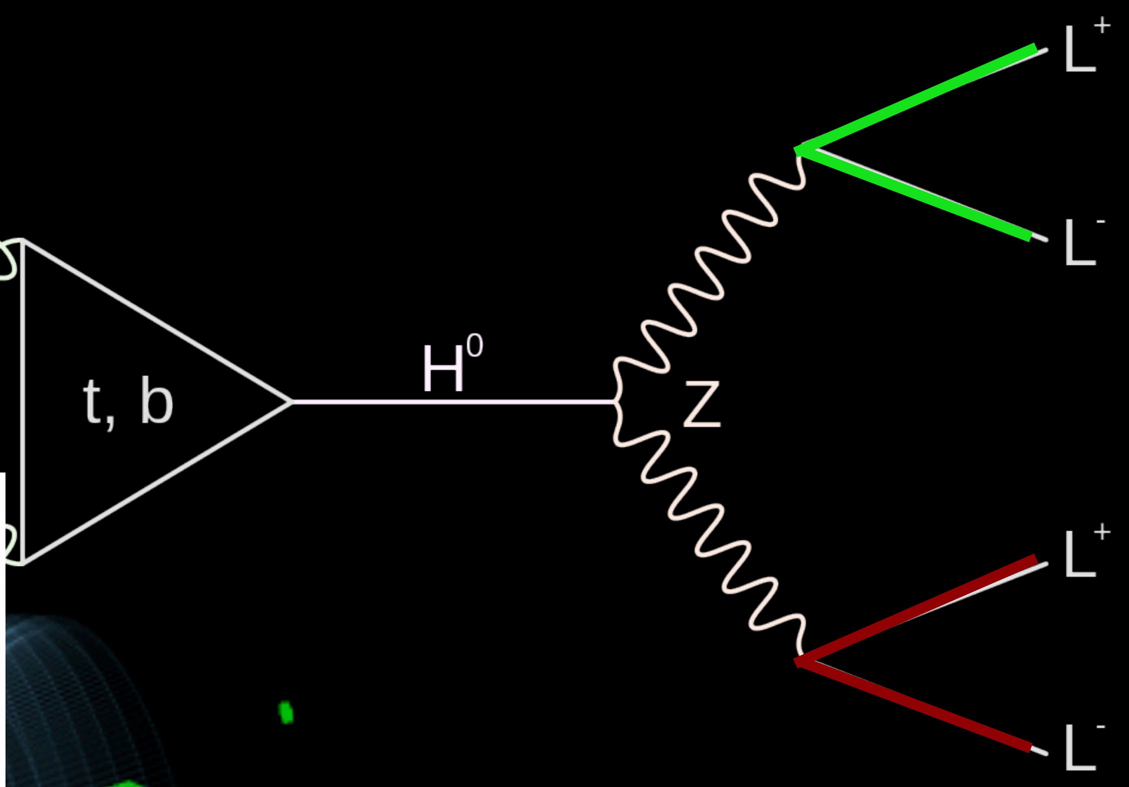

p
g
t, b
H0
Z
L+
L-
L+
L-
g
p

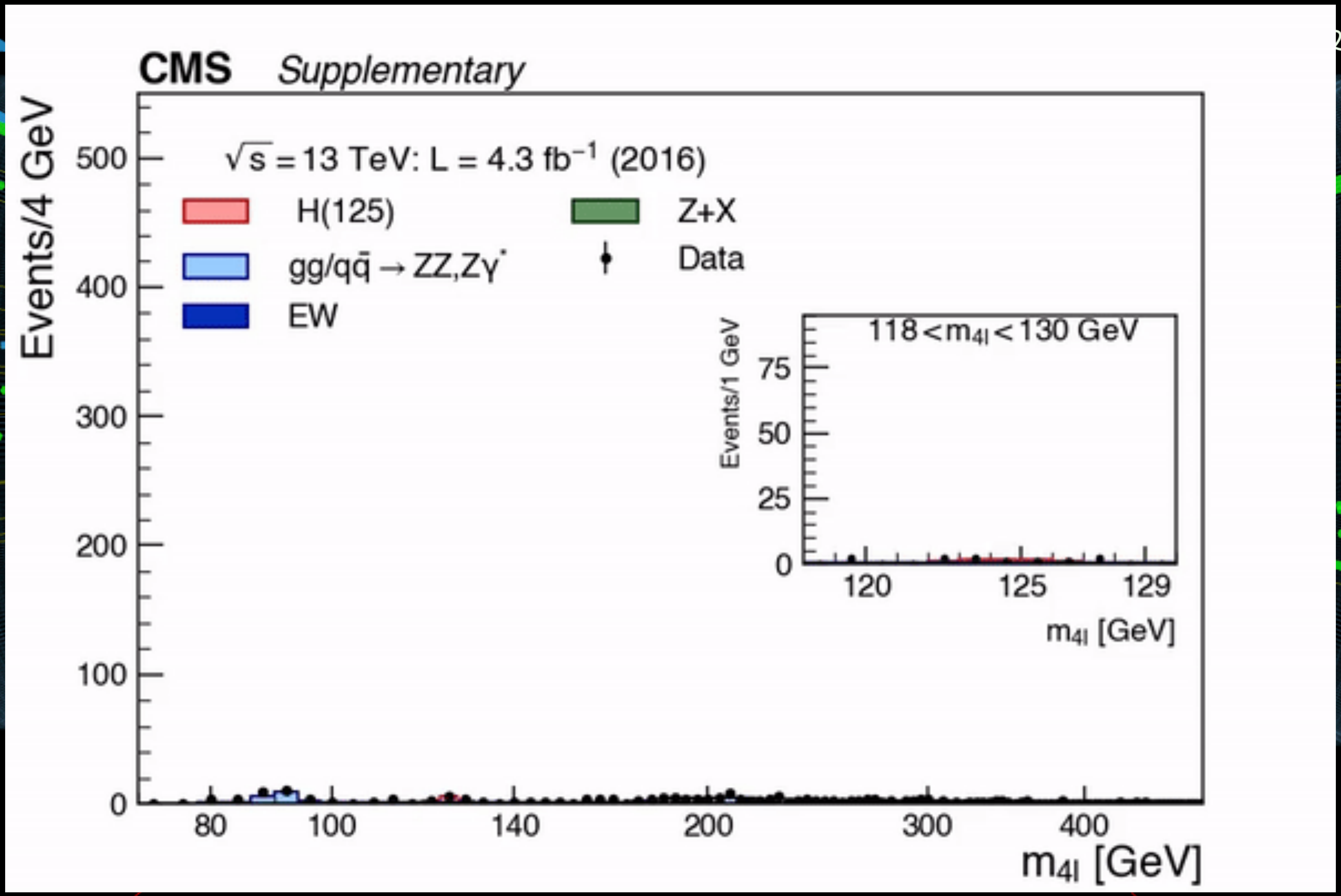
cmsexperiment.web.cern.ch



We had to collide billions of protons,
only around 10 signal events were needed to claim discovery!

cmsexperiment.web.cern.ch





We had to collide billions of protons,
only around 10 signal events were needed to claim discovery!

The Standard Model

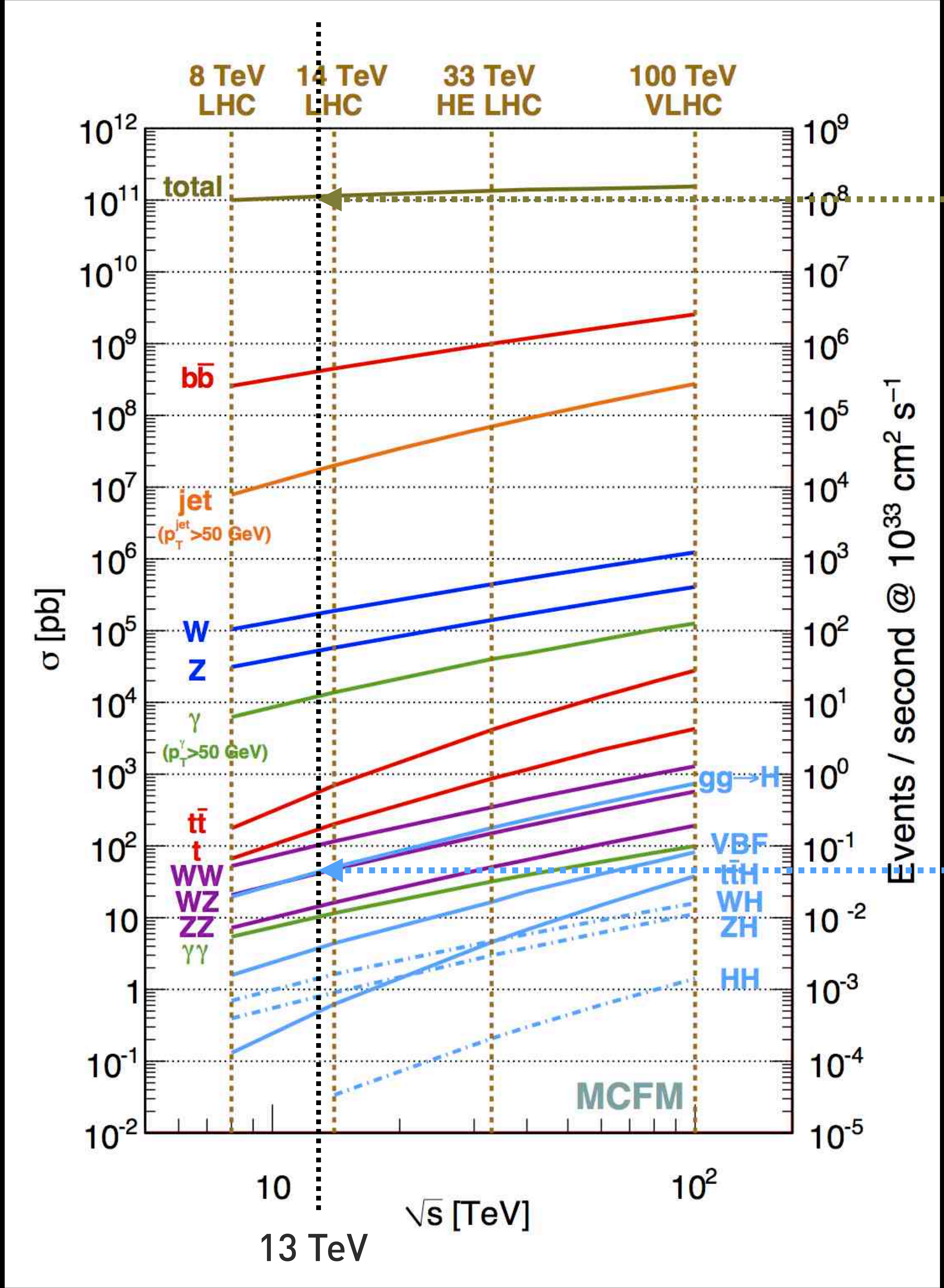
$$
\begin{gathered}
-\tfrac{1}{2}\partial_\nu g^a_\mu \partial_\nu g^a_\mu - g_s f^{abc}\partial_\mu g^a_\nu g^b_\mu g^c_\nu - \tfrac{1}{4}g_s^2 f^{abc}f^{ade}g^b_\mu g^c_\nu g^d_\mu g^e_\nu + \\
\tfrac{1}{2}ig_s^2(\bar{q}^\sigma_i\gamma^\mu q^\sigma_j)g^a_\mu + \bar{G}^a\partial^2 G^a + g_s f^{abc}\partial_\mu \bar{G}^a G^b g^c_\mu - \partial_\nu W^+_\mu \partial_\nu W^-_\mu - \\
M^2 W^+_\mu W^-_\mu - \tfrac12 \partial_\nu Z^0_\mu\partial_\nu Z^0_\mu - \tfrac{1}{2c_w^2}M^2 Z^0_\mu Z^0_\mu - \tfrac12\partial_\mu A_\nu\partial_\mu A_\nu - \tfrac12\partial_\mu H\partial_\mu H - \\
\tfrac12 m_h^2 H^2 - \partial_\mu\phi^+\partial_\mu\phi^- - M^2\phi^+\phi^- - \tfrac12\partial_\mu\phi^0\partial_\mu\phi^0 - \tfrac{1}{2c_w^2}M\phi^0\phi^0 - \beta_h[\tfrac{2M^2}{g^2} + \\
\tfrac{2M}{g}H + \tfrac12(H^2+\phi^0\phi^0+2\phi^+\phi^-)] + \tfrac{2M^4}{g^2}\alpha_h - igc_w[\partial_\nu Z^0_\mu(W^+_\mu W^-_\nu - \\
W^+_\nu W^-_\mu) - Z^0_\nu(W^+_\mu\partial_\nu W^-_\mu - W^-_\mu\partial_\nu W^+_\mu) + Z^0_\mu(W^+_\nu\partial_\nu W^-_\mu - \\
W^-_\nu\partial_\nu W^+_\mu)] - igs_w[\partial_\nu A_\mu(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - A_\nu(W^+_\mu\partial_\nu W^-_\mu - \\
W^-_\mu\partial_\nu W^+_\mu) + A_\mu(W^+_\nu\partial_\nu W^-_\mu - W^-_\nu\partial_\nu W^+_\mu)] - \tfrac12 g^2 W^+_\mu W^-_\mu W^+_\nu W^-_\nu + \\
\tfrac12 g^2 W^+_\mu W^-_\nu W^+_\mu W^-_\nu + g^2c_w^2(Z^0_\mu W^+_\mu Z^0_\nu W^-_\nu - Z^0_\mu Z^0_\mu W^+_\nu W^-_\nu) + \\
g^2 s_w^2(A_\mu W^+_\mu A_\nu W^-_\nu - A_\mu A_\mu W^+_\nu W^-_\nu) + g^2 s_w c_w[A_\mu Z^0_\nu(W^+_\mu W^-_\nu - \\
W^+_\nu W^-_\mu) - 2A_\mu Z^0_\mu W^+_\nu W^-_\nu] - g\alpha[H^3 + H\phi^0\phi^0 + 2H\phi^+\phi^-] - \\
\tfrac18 g^2\alpha_h[H^4 + (\phi^0)^4 + 4(\phi^+\phi^-)^2 + 4(\phi^0)^2\phi^+\phi^- + 4H^2\phi^+\phi^- + 2(\phi^0)^2H^2] - \\
gMW^+_\mu W^-_\mu H - \tfrac12 g\tfrac{M}{c_w^2}Z^0_\mu Z^0_\mu H - \tfrac12 ig[W^+_\mu(\phi^0\partial_\mu\phi^- - \phi^-\partial_\mu\phi^0) - \\
W^-_\mu(\phi^0\partial_\mu\phi^+ - \phi^+\partial_\mu\phi^0)] + \tfrac12 g[W^+_\mu(H\partial_\mu\phi^- - \phi^-\partial_\mu H) - W^-_\mu(H\partial_\mu\phi^+ - \\
\phi^+\partial_\mu H)] + \tfrac12 g\tfrac{1}{c_w}(Z^0_\mu(H\partial_\mu\phi^0 - \phi^0\partial_\mu H) - ig\tfrac{s_w^2}{c_w}MZ^0_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) + \\
igs_wMA_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) - ig\tfrac{1-2c_w^2}{2c_w}Z^0_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) + \\
igs_wA_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) - \tfrac14 g^2 W^+_\mu W^-_\mu[H^2 + (\phi^0)^2 + 2\phi^+\phi^-] - \\
\tfrac14 g^2\tfrac{1}{c_w^2}Z^0_\mu Z^0_\mu[H^2 + (\phi^0)^2 + 2(2s_w^2-1)^2\phi^+\phi^-] - \tfrac12 g^2\tfrac{s_w^2}{c_w}Z^0_\mu\phi^0(W^+_\mu\phi^- + \\
W^-_\mu\phi^+) - \tfrac12 ig^2\tfrac{s_w^2}{c_w}Z^0_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) + \tfrac12 g^2 s_w A_\mu\phi^0(W^+_\mu\phi^- + \\
W^-_\mu\phi^+) + \tfrac12 ig^2 s_w A_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) - g^2\tfrac{s_w}{c_w}(2c_w^2-1)Z^0_\mu A_\mu\phi^+\phi^- - \\
g^1 s_w^2 A_\mu A_\mu\phi^+\phi^- - \bar e^\lambda(\gamma\partial + m_e^\lambda)e^\lambda - \bar\nu^\lambda\gamma\partial\nu^\lambda - \bar u^\lambda_j(\gamma\partial + m_u^\lambda)u^\lambda_j - \\
\bar d^\lambda_j(\gamma\partial + m_d^\lambda)d^\lambda_j + igs_wA_\mu[-(\bar e^\lambda\gamma^\mu e^\lambda) + \tfrac23(\bar u^\lambda_j\gamma^\mu u^\lambda_j) - \tfrac13(\bar d^\lambda_j\gamma^\mu d^\lambda_j)] + \\
\tfrac{ig}{4c_w}Z^0_\mu[(\bar\nu^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar e^\lambda\gamma^\mu(4s_w^2 - 1 - \gamma^5)e^\lambda) + (\bar u^\lambda_j\gamma^\mu(\tfrac43 s_w^2 - \\
1 - \gamma^5)u^\lambda_j) + (\bar d^\lambda_j\gamma^\mu(1 - \tfrac83 s_w^2 - \gamma^5)d^\lambda_j)] + \tfrac{ig}{2\sqrt2}W^+_\mu[(\bar\nu^\lambda\gamma^\mu(1+\gamma^5)\text{[illegible]}^\lambda) + \\
(\bar u^\lambda_j\gamma^\mu(1+\gamma^5)C_{\lambda\kappa}d^\kappa_j)] + \tfrac{ig}{2\sqrt2}W^-_\mu[(\bar e^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar d^\kappa_j C^\dagger_{\lambda\kappa}\gamma^\mu(1 + \\
\gamma^5)u^\lambda_j)] + \tfrac{ig}{2\sqrt2}\tfrac{m_e^\lambda}{M}[-\phi^+(\bar\nu^\lambda(1-\gamma^5)e^\lambda) + \phi^-(\bar e^\lambda(1+\gamma^5)\nu^\lambda)] - \\
\tfrac g2\tfrac{m_e^\lambda}{M}[H(\bar e^\lambda e^\lambda) + i\phi^0(\bar e^\lambda\gamma^5 e^\lambda)] + \tfrac{ig}{2M\sqrt2}\phi^+[-m_d^\kappa(\bar u^\lambda_j C_{\lambda\kappa}(1-\gamma^5)d^\kappa_j) + \\
m_u^\lambda(\bar u^\lambda_j C_{\lambda\kappa}(1+\gamma^5)d^\kappa_j] + \tfrac{ig}{2M\sqrt2}\phi^-[m_d^\lambda(\bar d^\lambda_j C^\dagger_{\lambda\kappa}(1+\gamma^5)u^\kappa_j) - m_u^\kappa(\bar d^\lambda_j C^\dagger_{\lambda\kappa}(1 - \\
\gamma^5)u^\kappa_j] - \tfrac g2\tfrac{m_u^\lambda}{M}H(\bar u^\lambda_j u^\lambda_j) - \tfrac g2\tfrac{m_d^\lambda}{M}H(\bar d^\lambda_j d^\lambda_j) + \tfrac{ig}{2}\tfrac{m_u^\lambda}{M}\phi^0(\bar u^\lambda_j\gamma^5 u^\lambda_j) - \\
\tfrac{ig}{2}\tfrac{m_d^\lambda}{M}\phi^0(\bar d^\lambda_j\gamma^5 d^\lambda_j) + \bar X^+(\partial^2 - M^2)X^+ + \bar X^-(\partial^2 - M^2)X^- + \bar X^0(\partial^2 - \\
\tfrac{M^2}{c_w^2})X^0 + \bar Y\partial^2 Y + igc_wW^+_\mu(\partial_\mu\bar X^0 X^- - \partial_\mu\bar X^+ X^0) + igs_wW^+_\mu(\partial_\mu\bar Y X^- - \\
\partial_\mu\bar X^+ Y) + igc_wW^-_\mu(\partial_\mu\bar X^- X^0 - \partial_\mu\bar X^0 X^+) + igs_wW^-_\mu(\partial_\mu\bar X^- Y - \\
\partial_\mu\bar Y X^+) + igc_wZ^0_\mu(\partial_\mu\bar X^+ X^+ - \partial_\mu\bar X^- X^-) + igs_wA_\mu(\partial_\mu\bar X^+ X^+ - \\
\partial_\mu\bar X^- X^-) - \tfrac12 gM[\bar X^+X^+H + \bar X^-X^-H + \tfrac{1}{c_w^2}\bar X^0X^0H] + \\
\tfrac{1-2c_w^2}{2c_w}igM[\bar X^+X^0\phi^+ - \bar X^-X^0\phi^-] + \tfrac{1}{2c_w}igM[\bar X^0X^-\phi^+ - \bar X^0X^+\phi^-] + \\
igMs_w[\bar X^0X^-\phi^+ - \bar X^0X^+\phi^-] + \tfrac12 igM[\bar X^+X^+\phi^0 - \bar X^-X^-\phi^0]
\end{gathered}
$$

1 billion collisions /s

~1 MB of data / collision

~1 PB of data / s

1 PB of data / second



“Probability” of producing “anything”

“Probability” of producing a Higgs

Higgs produced
~1 in a billion collisions!

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

Software rate reduction
(GPU+CPU)

2 step rate reduction
(hardware+software)

Continous read-out
(CPU+GPU)

2 step rate reduction
(hardware+software)

2 step rate reduction
(hardware+software)

Data temporarily stored in detector electronics for 4 µs

5% internet traffic to L1
(63 Tb/s)

5% internet traffic to L1
(63 Tb/s)

L1 trigger:
~1000 Xilinx/AMD
Ultrascale FPGAs

Decide which event to
keep within ~4 μs latency

Discard >99% of
collisions!

L1 bit:
Accept = 1
Reject = 0

DATA
99.72% of events rejected!
110 thousand events/

DATA

99.72% of events rejected!

110 thousand events/

High Level Trigger:
25’600 CPUs / 400 GPUs
Latency: 3-400 ms

Reject further 99%!

High Level Trigger:
Latency O(100) ms

DATA
99.9975% of events rejected!
1000 events/second
~5 GB/s

TIER 0: ∞

High Level Trigger:
Latency O(100) ms

TIER 0: ∞

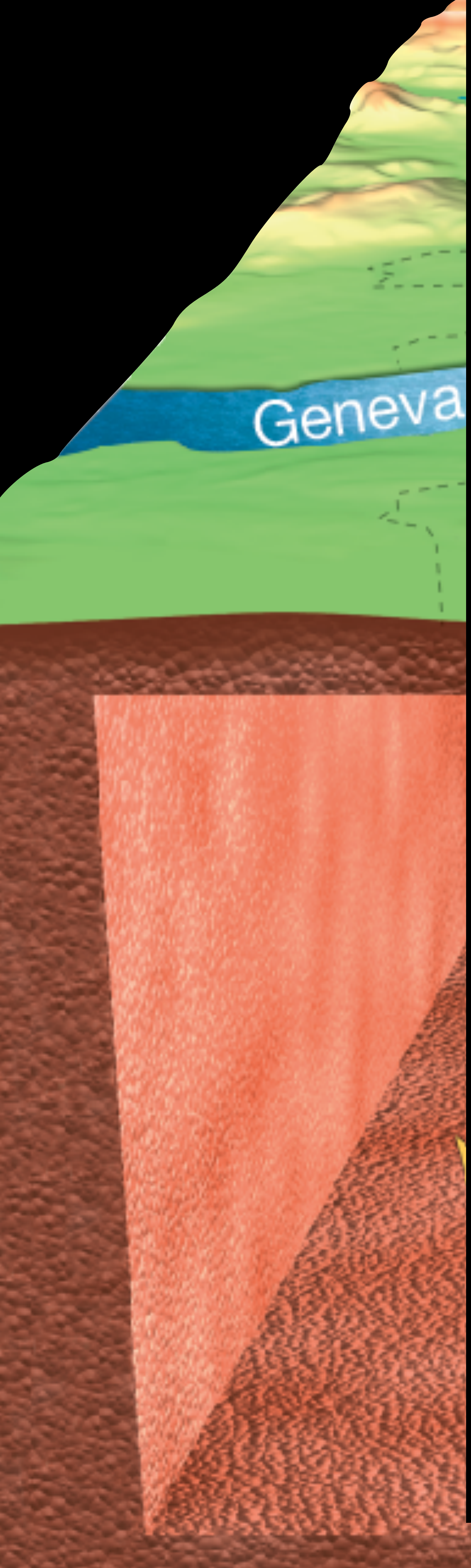
DATA
99.9975% of events rejected!
1000 events/second
~5 GB/s

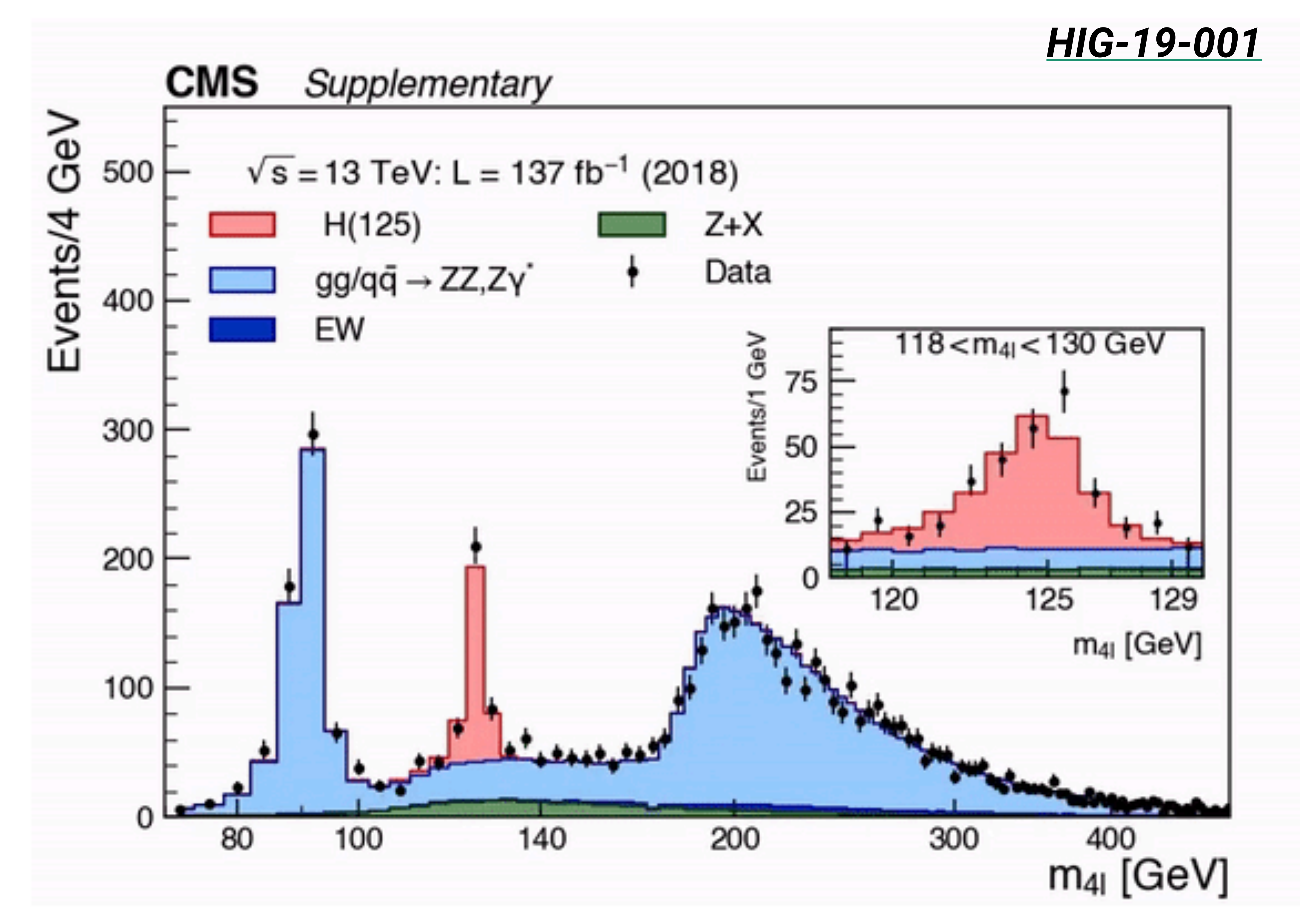
High Level Trigger:
Latency 0(100) ms

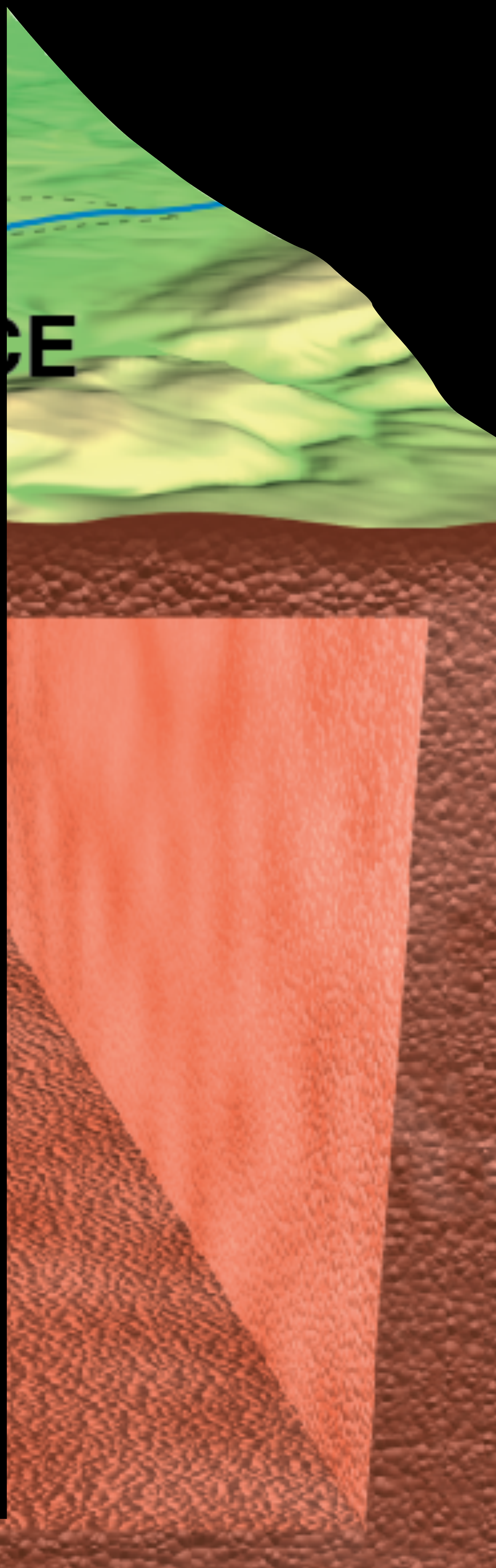
TIER 0: ∞

0.0025% of collision events remaining

DATA
99.9975% of events rejected!
1000 events/second
~5 GB/s

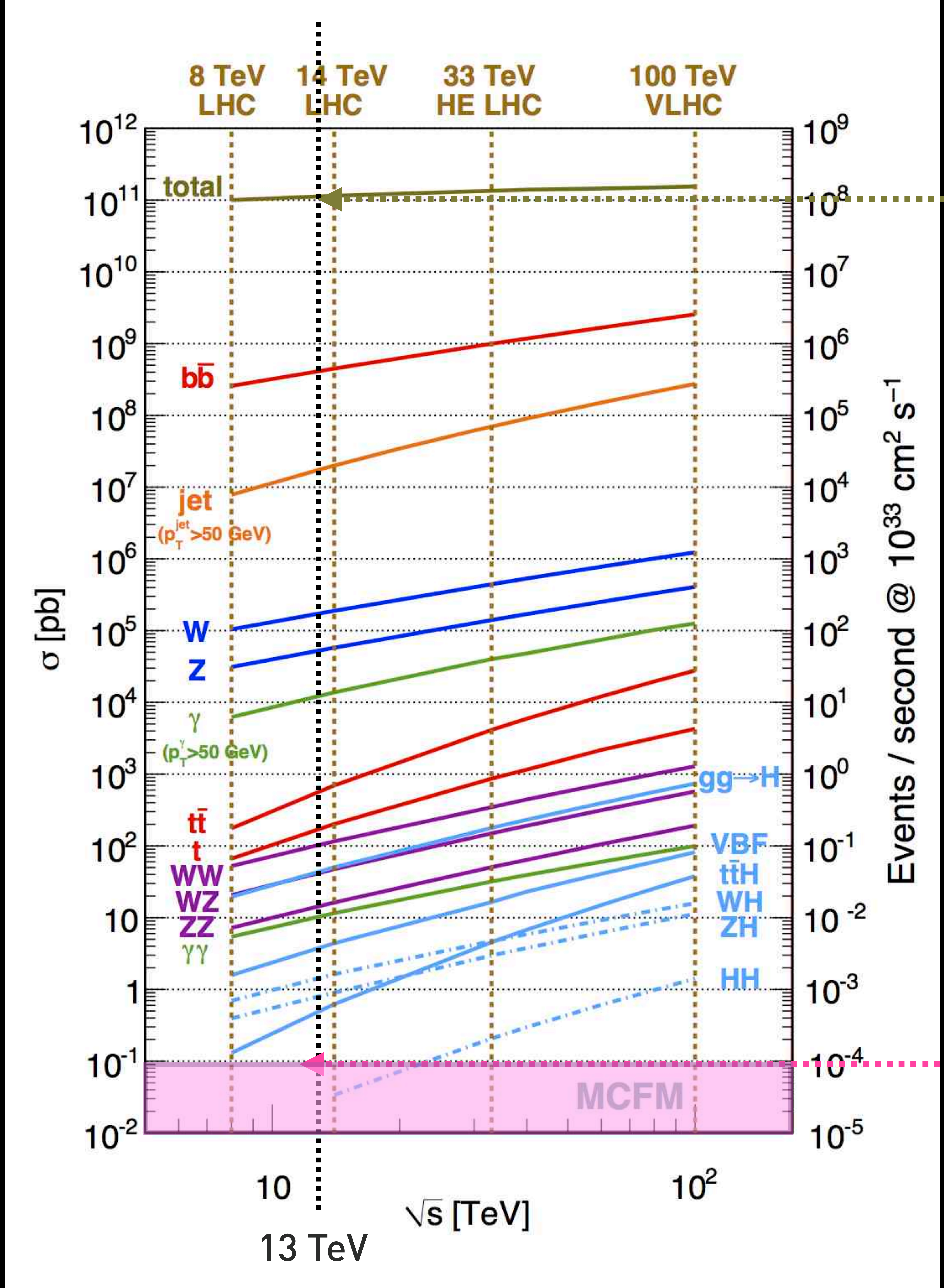






To make sure we select “the right” 0.0025%, algorithms must be

- Fast (get more data through)
- Accurate (select the right data)



“Probability” of producing “anything”

New Physics?

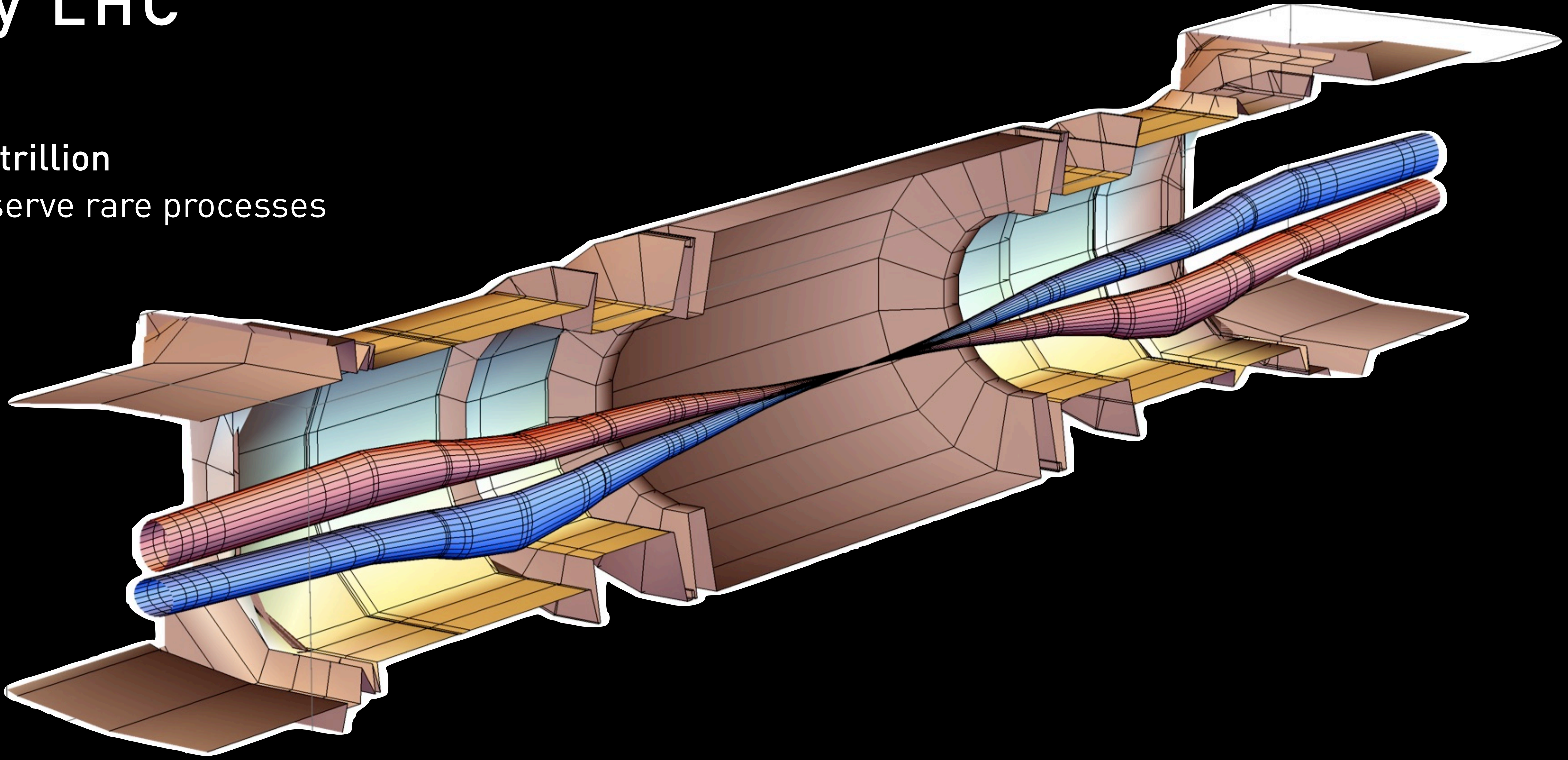
New Physics is produced less than 1 in a trillion (if at all)

Need more data!

High Luminosity LHC

New Physics is produced 1 in a trillion
- Need more collisions to observe rare processes

High Luminosity LHC
- x10 data size
- x3 collisions/s



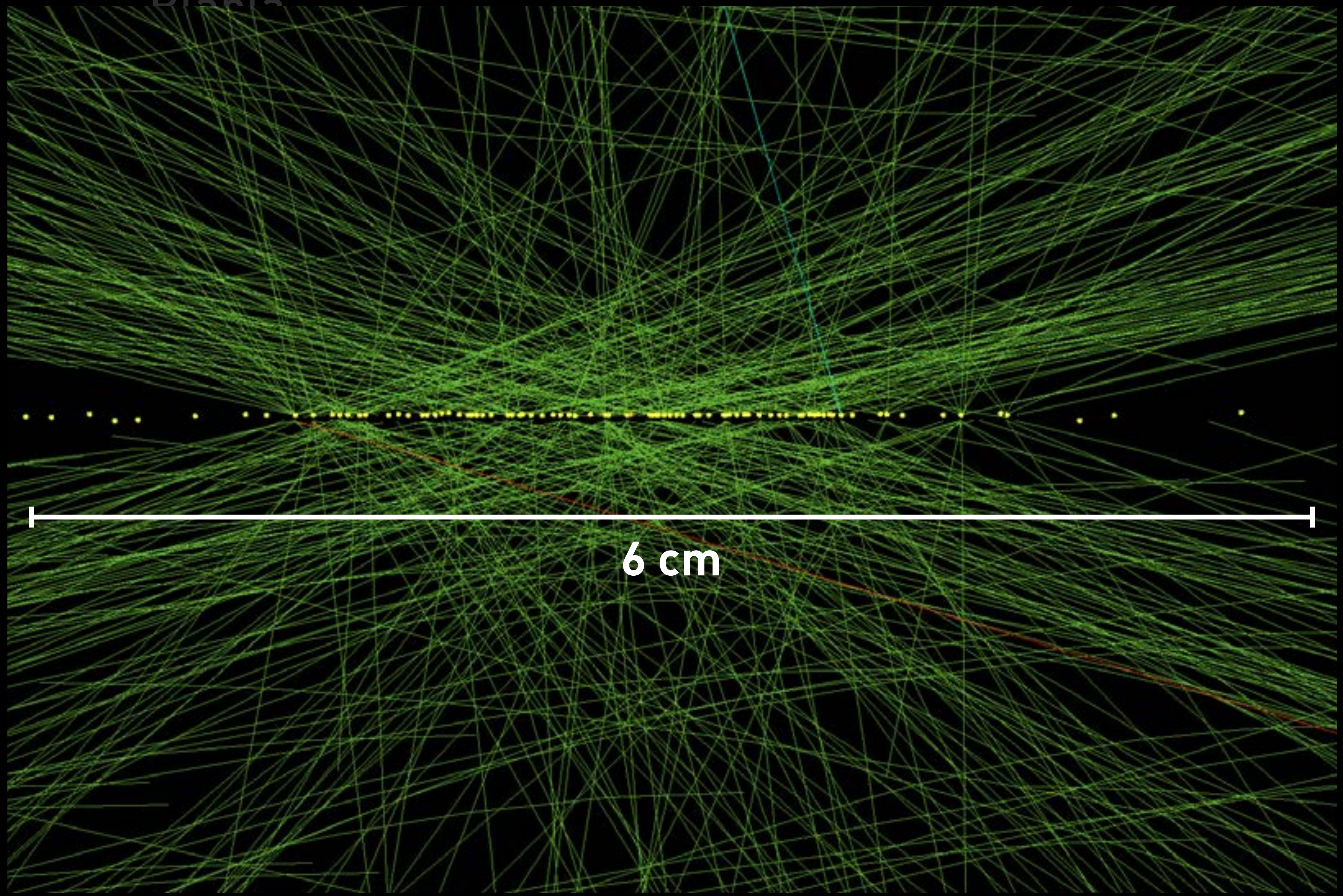
2022 - 2025
LHC (TODAY!)

2026 - 2028
MAJOR UPGRADE

2029 - 2038
HL-LHC

LHC

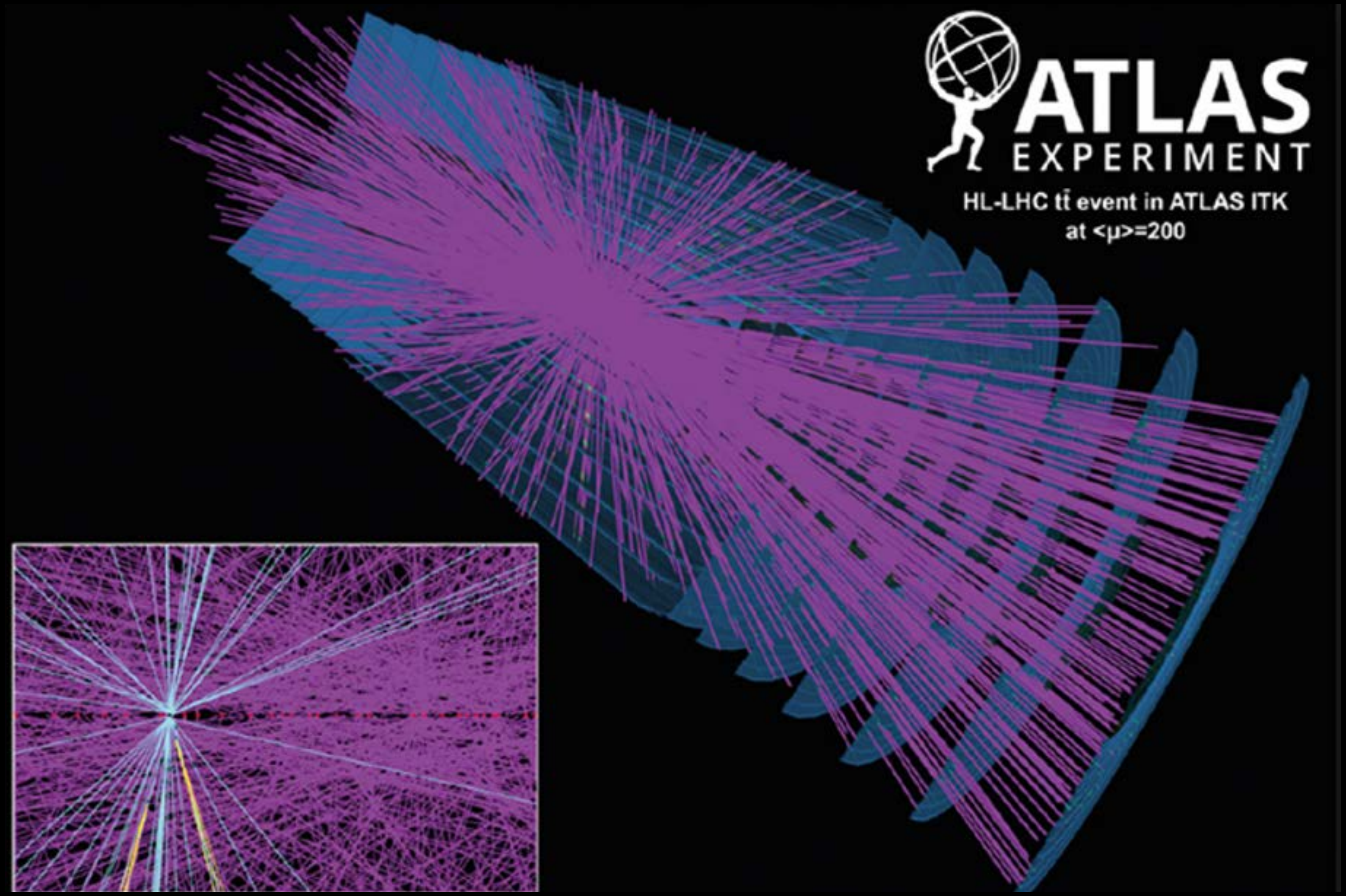
78 vertices
(average 60)



Run 3

High Luminosity LHC

200 vertices
(average 140)



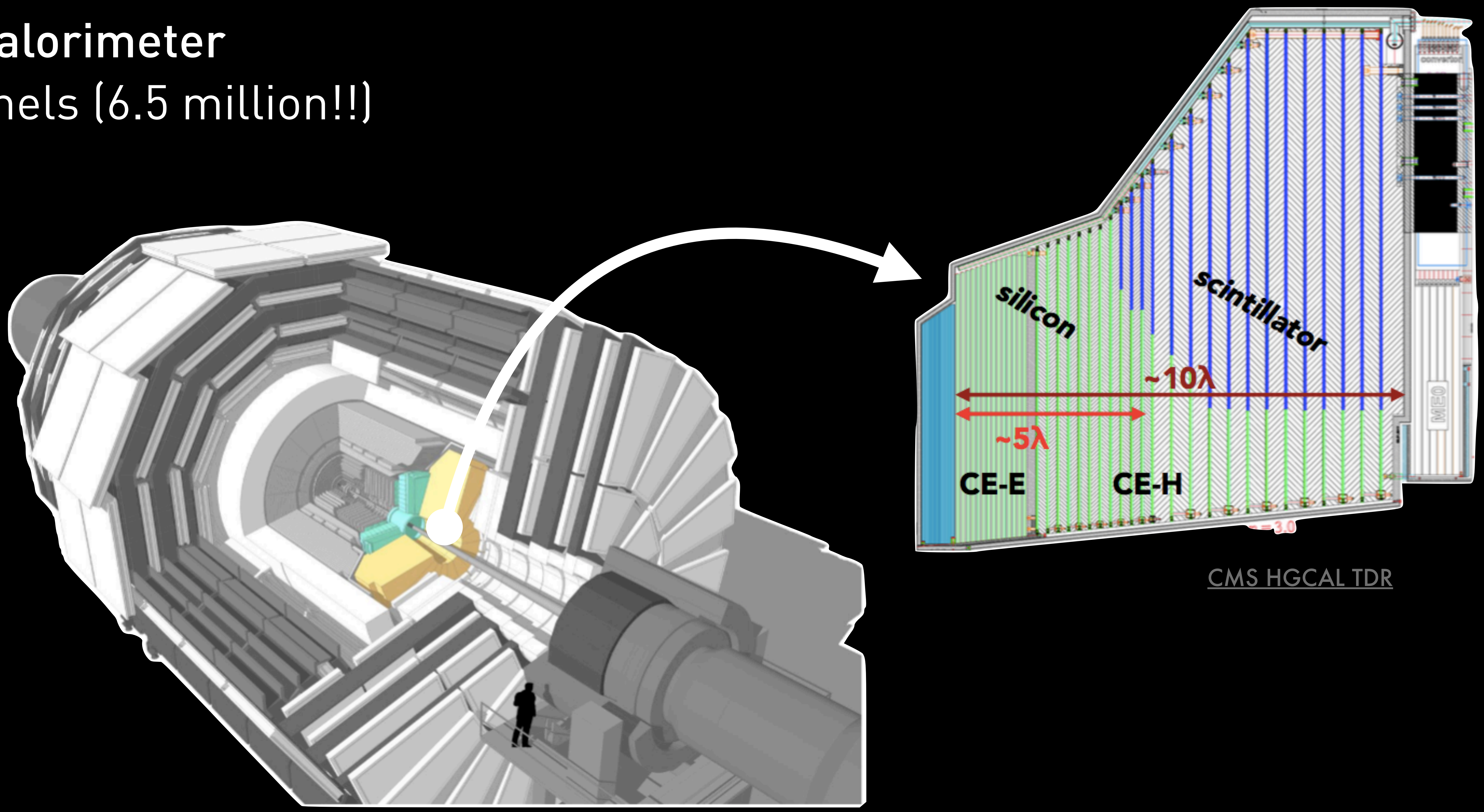
Run 4+5

Maintain physics acceptance → better detectors

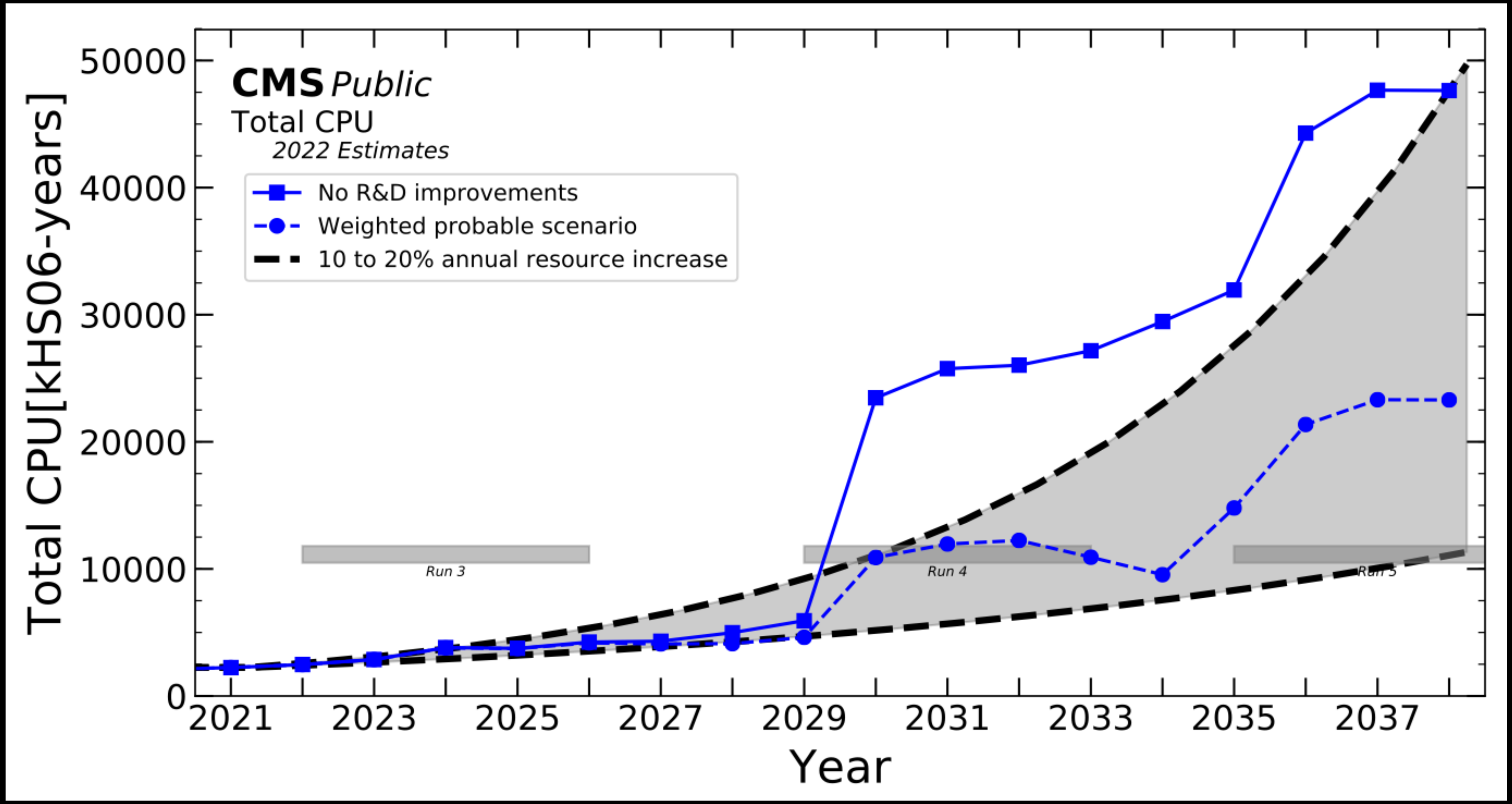
CMS High Granularity (endcap) calorimeter

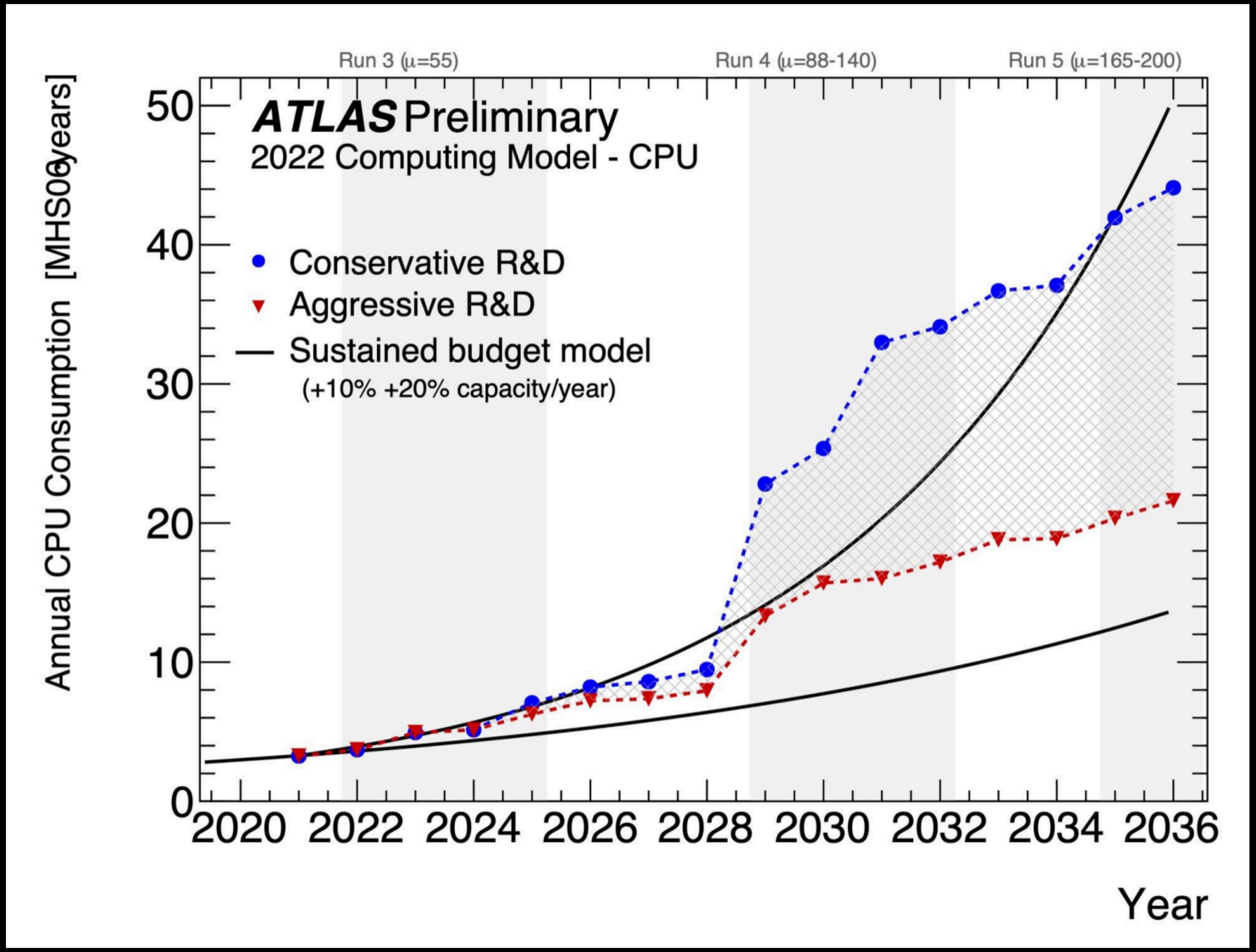
- X20 times more readout channels (6.5 million!!)

More collisions
More readout channels



CMS HGCAL TDR


CMS Public
Total CPU
2022 Estimates
No R&D improvements
Weighted probable scenario
10 to 20% annual resource increase
Run 3
Run 4
Run 5
Total CPU[kHS06-years]
0
10000
20000
30000
40000
50000
2021
2023
2025
2027
2029
2031
2033
2035
2037
Year


Run 3 (μ=55)
Run 4 (μ=88-140)
Run 5 (μ=165-200)
ATLAS Preliminary
2022 Computing Model - CPU
Conservative R&D
Aggressive R&D
Sustained budget model
(+10% +20% capacity/year)
Annual CPU Consumption [MHS06years]
0
10
20
30
40
50
2020
2022
2024
2026
2028
2030
2032
2034
2036
Year

Our current algorithms won’t be able to cope with HL-LHC data rates! Need innovation!

Can we use modern Machine Learning to
be faster
more accurate
and do more?

GPU inference

HLT trigger:
Latency O(100) ms

Fast inference on specialised hardware

ASIC inference

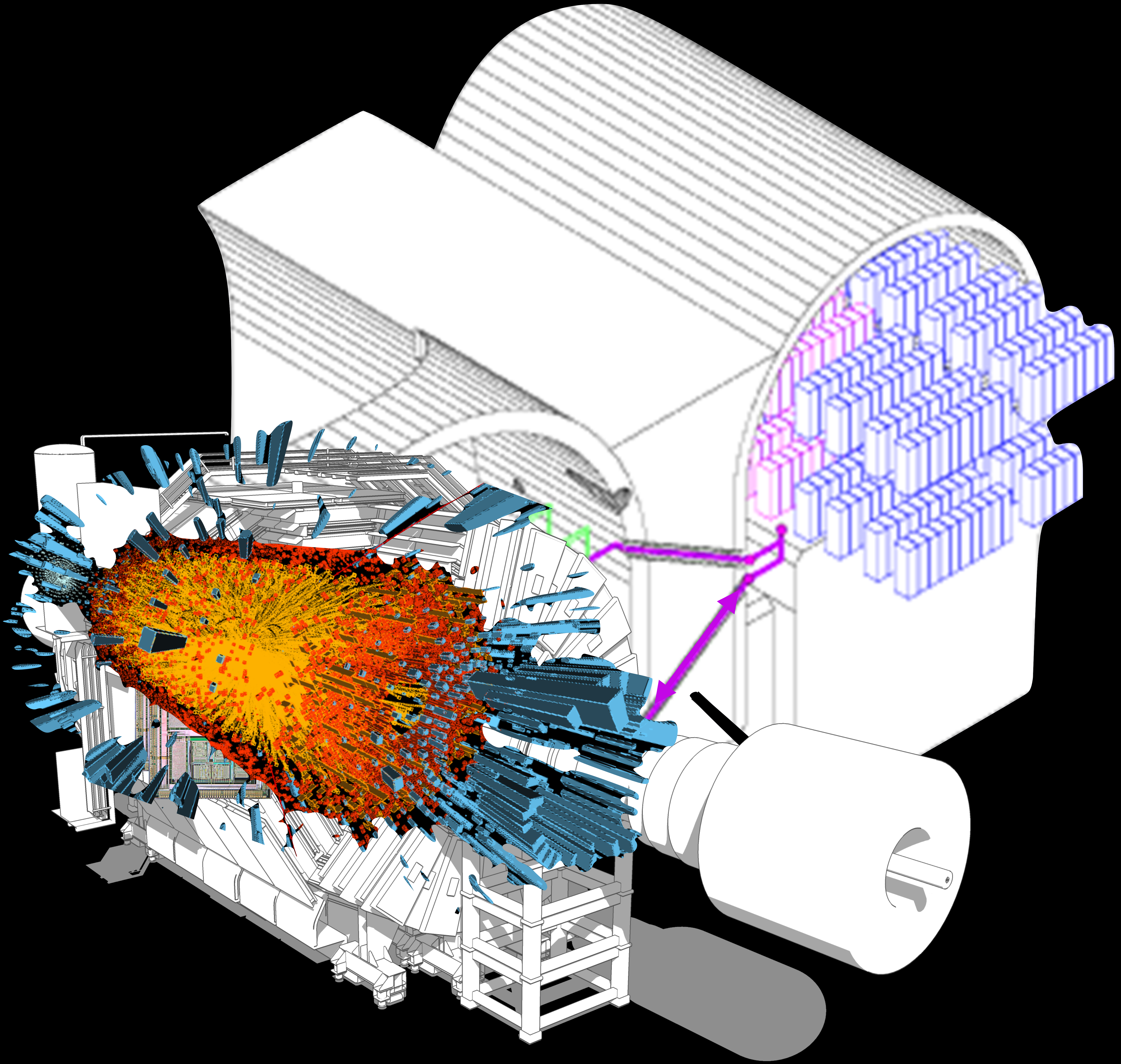
FPGA inference

Level-1 trigger:
Latency O(1) μs

Detector:
40 MHz
~Pb/s

HL-LHC Level-1:

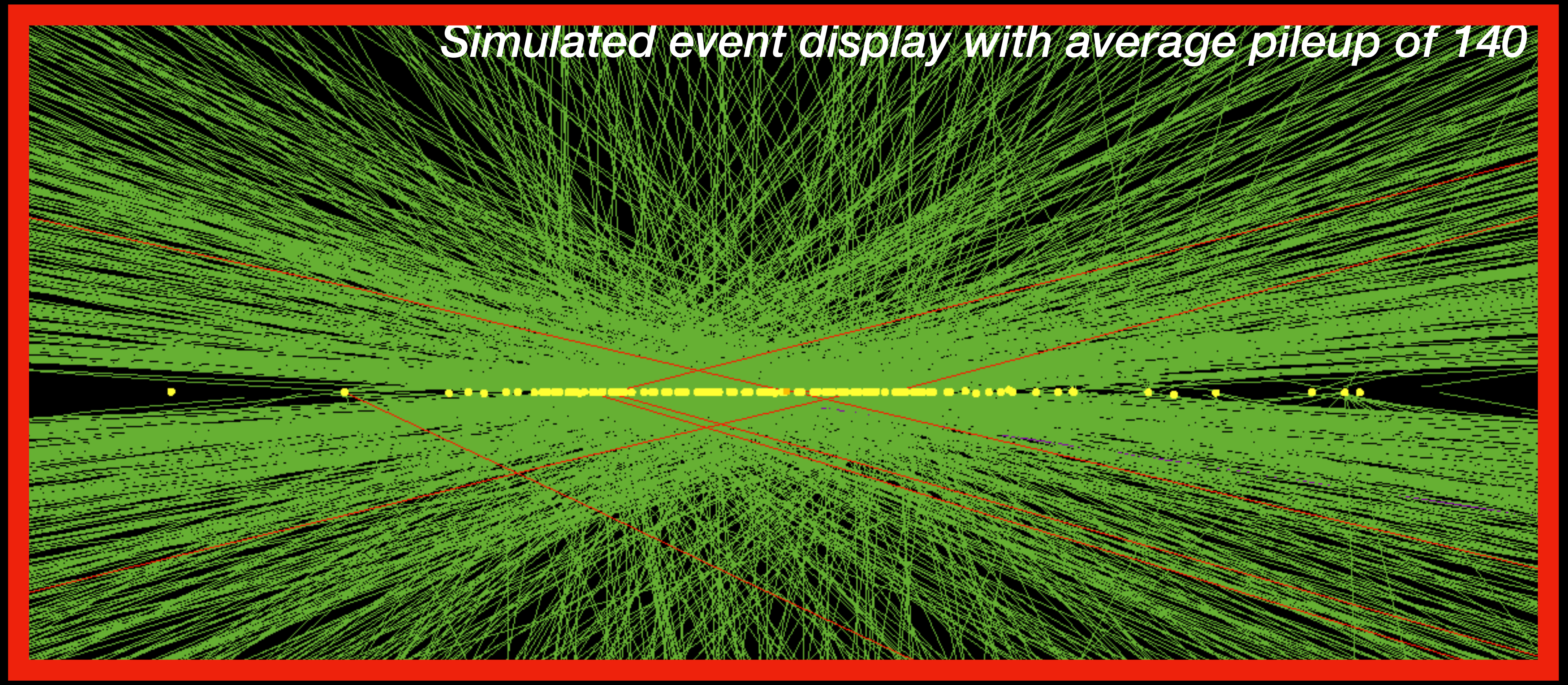
Complete re-design of Level-1

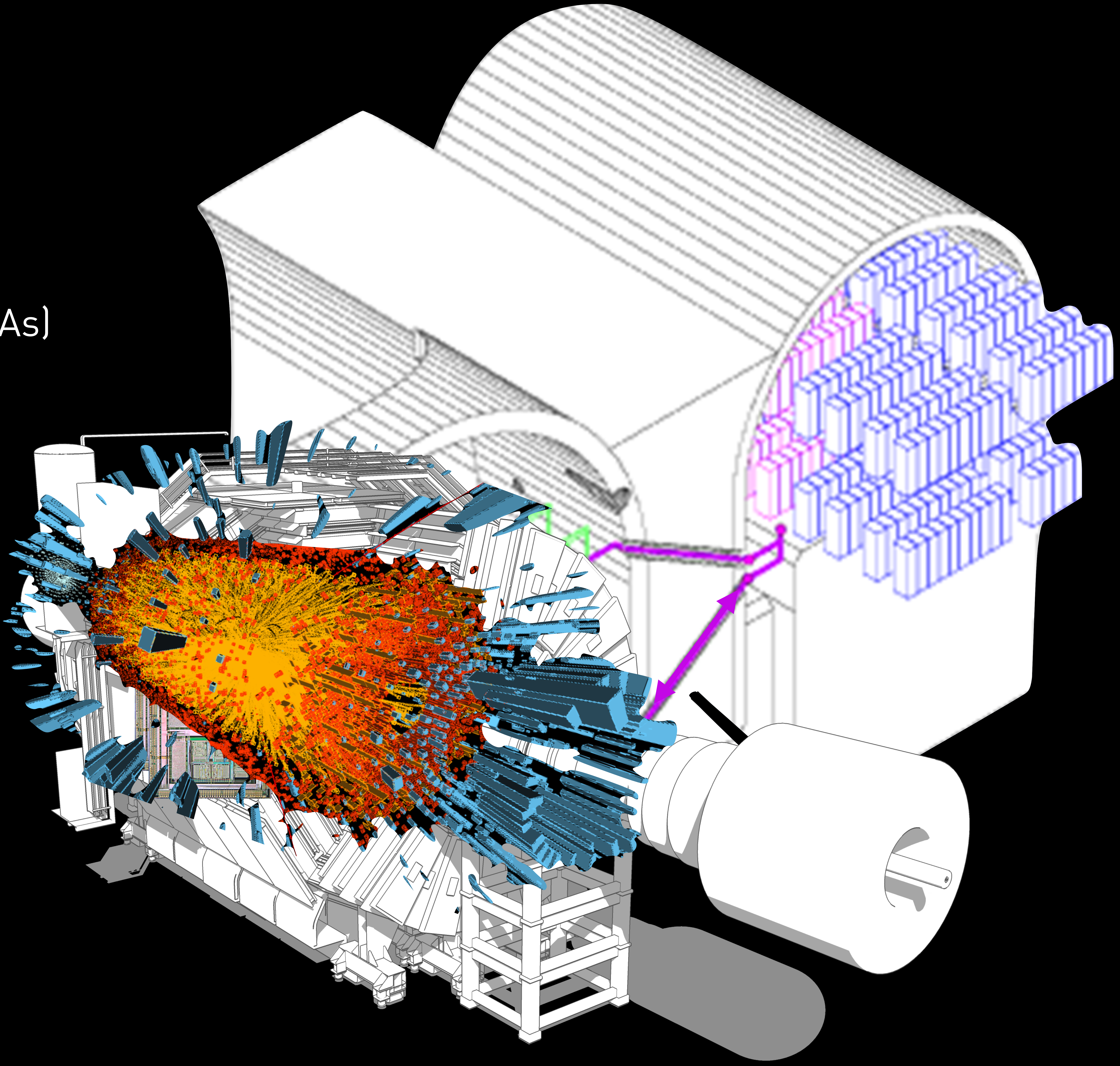


HL-LHC Level-1:

Complete re-design of Level-1

- Charged particle tracks (6.4 Tb/s, 200 FPGAs)

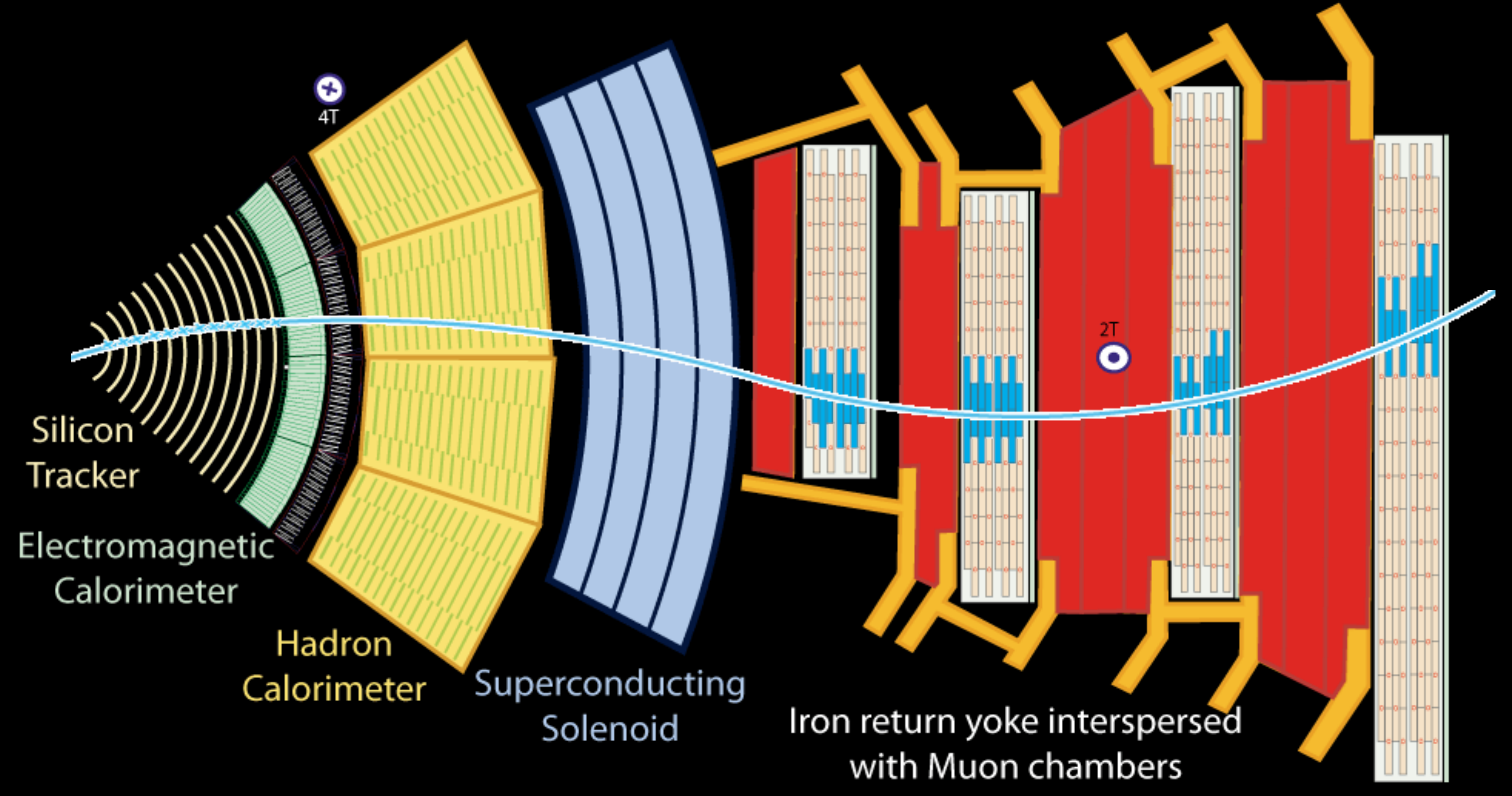


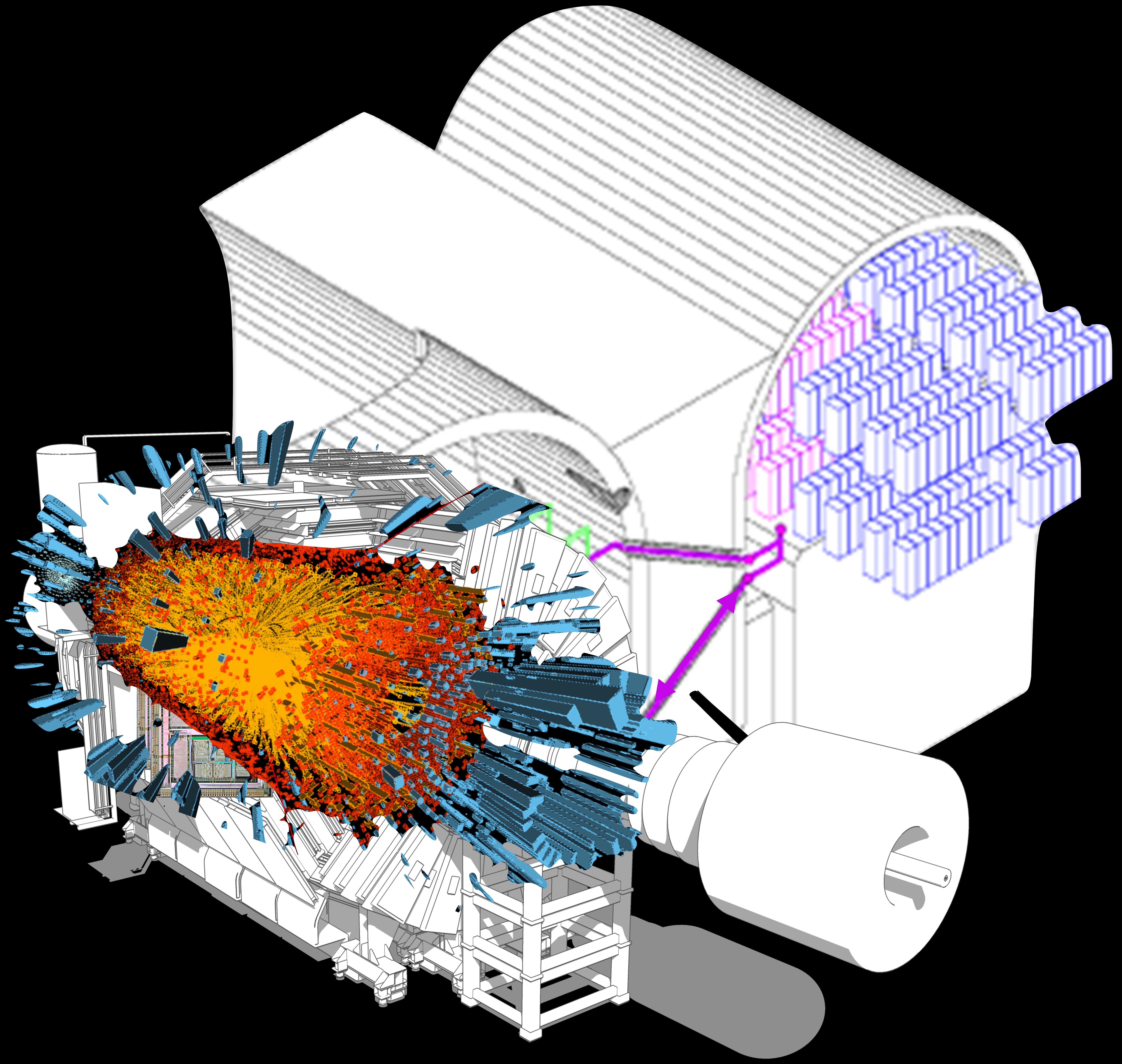


HL-LHC Level-1:

Complete re-design of Level-1

- Charged particle tracks
- Particle Flow (40 FPGAs)

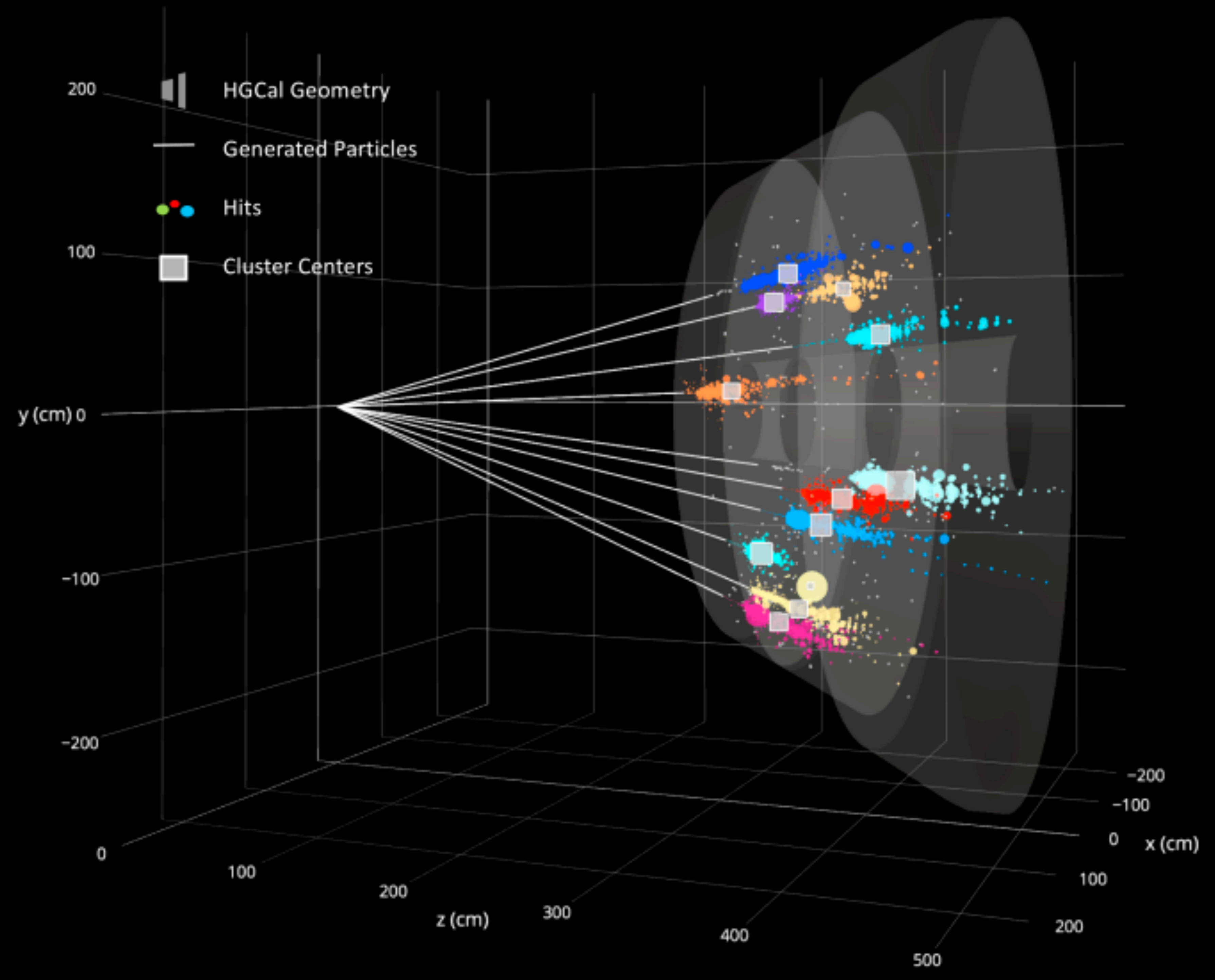


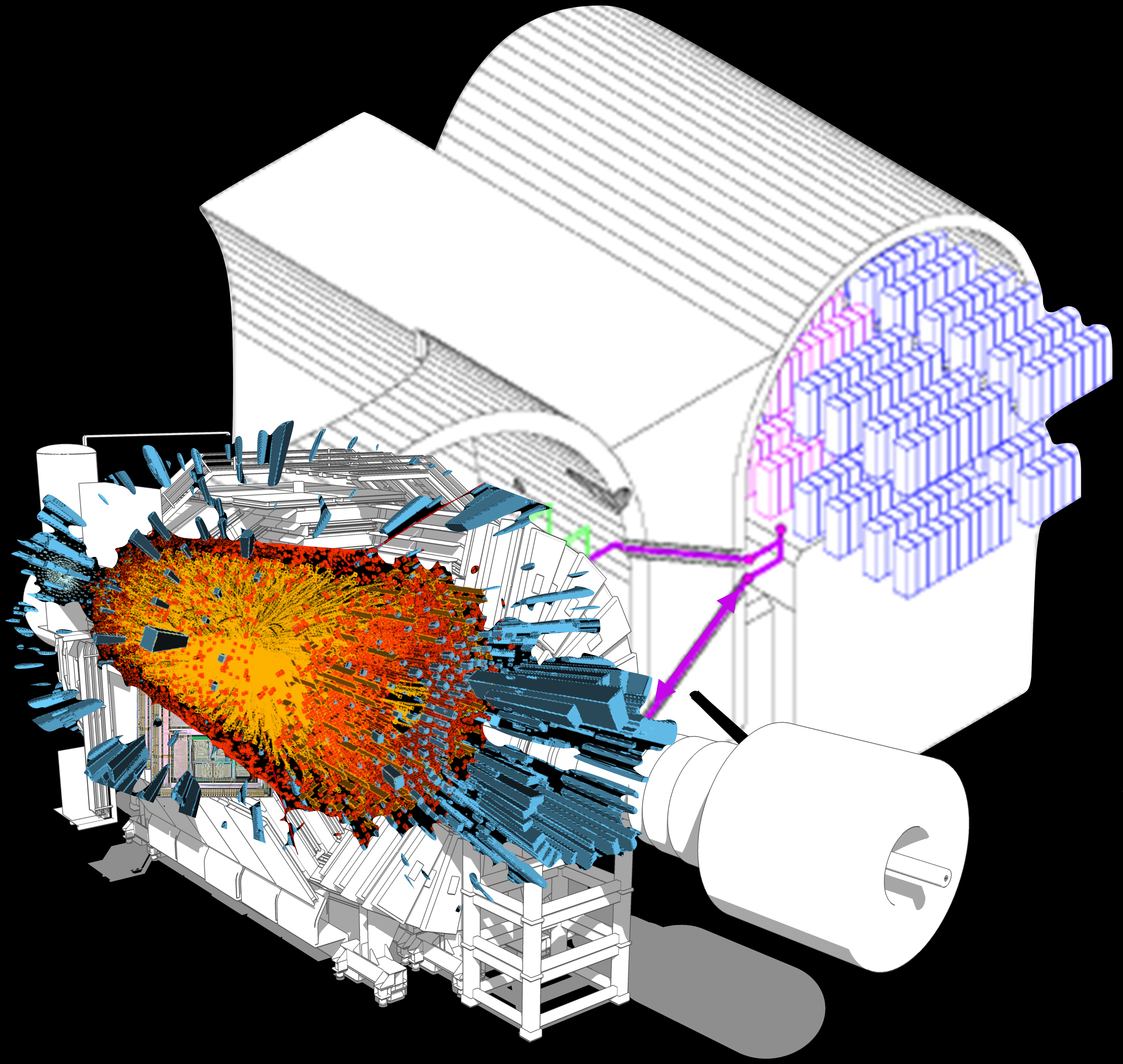


HL-LHC Level-1:

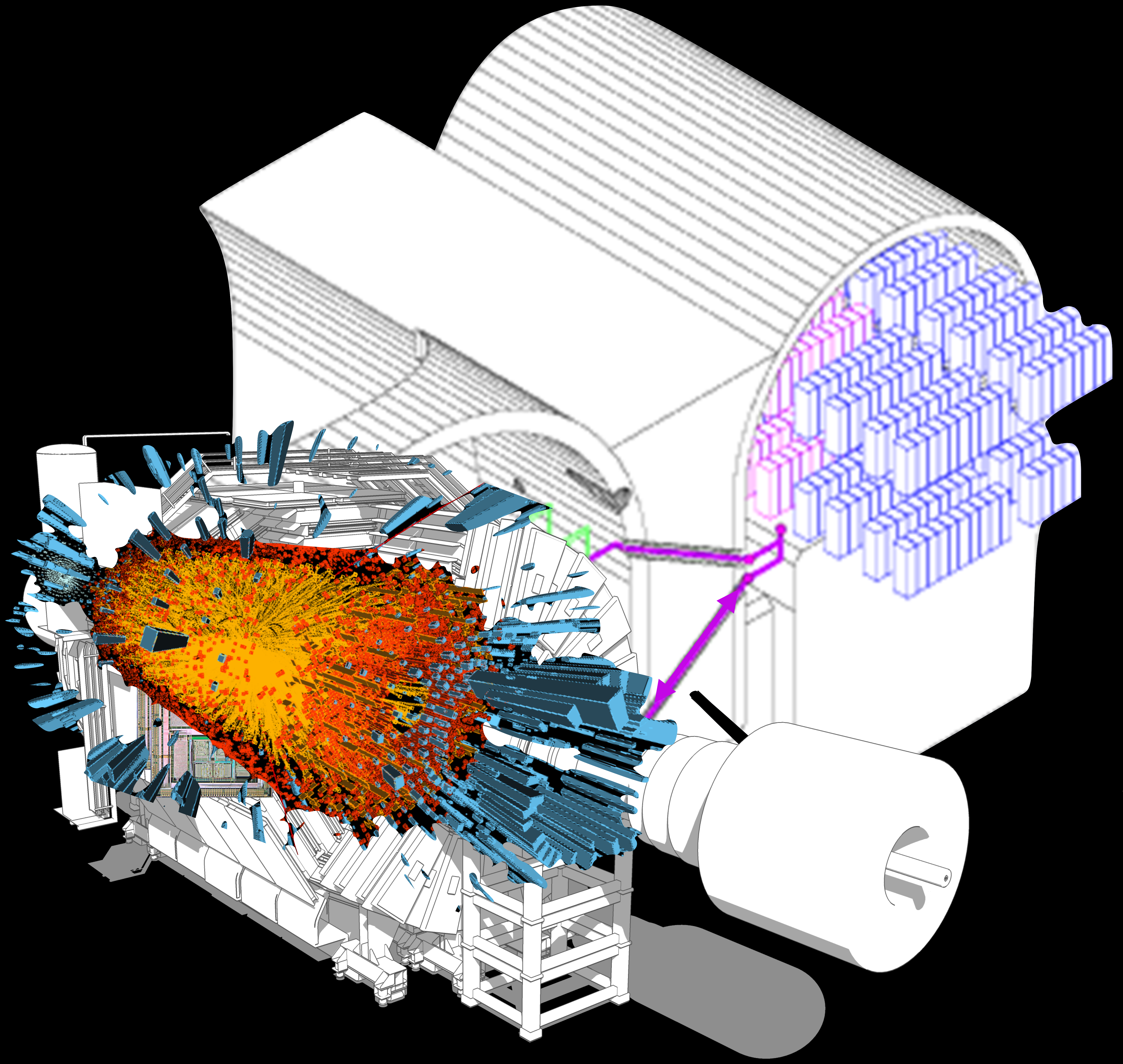
Complete re-design of Level-1

- Charged particle tracks
- Particle Flow
- HGCal (4 Tb/s, 200 FPGAs)





HL-LHC Level-1:



Complete re-design of Level-1

- Charged particle tracks
- Particle Flow
- HGCal

Input data

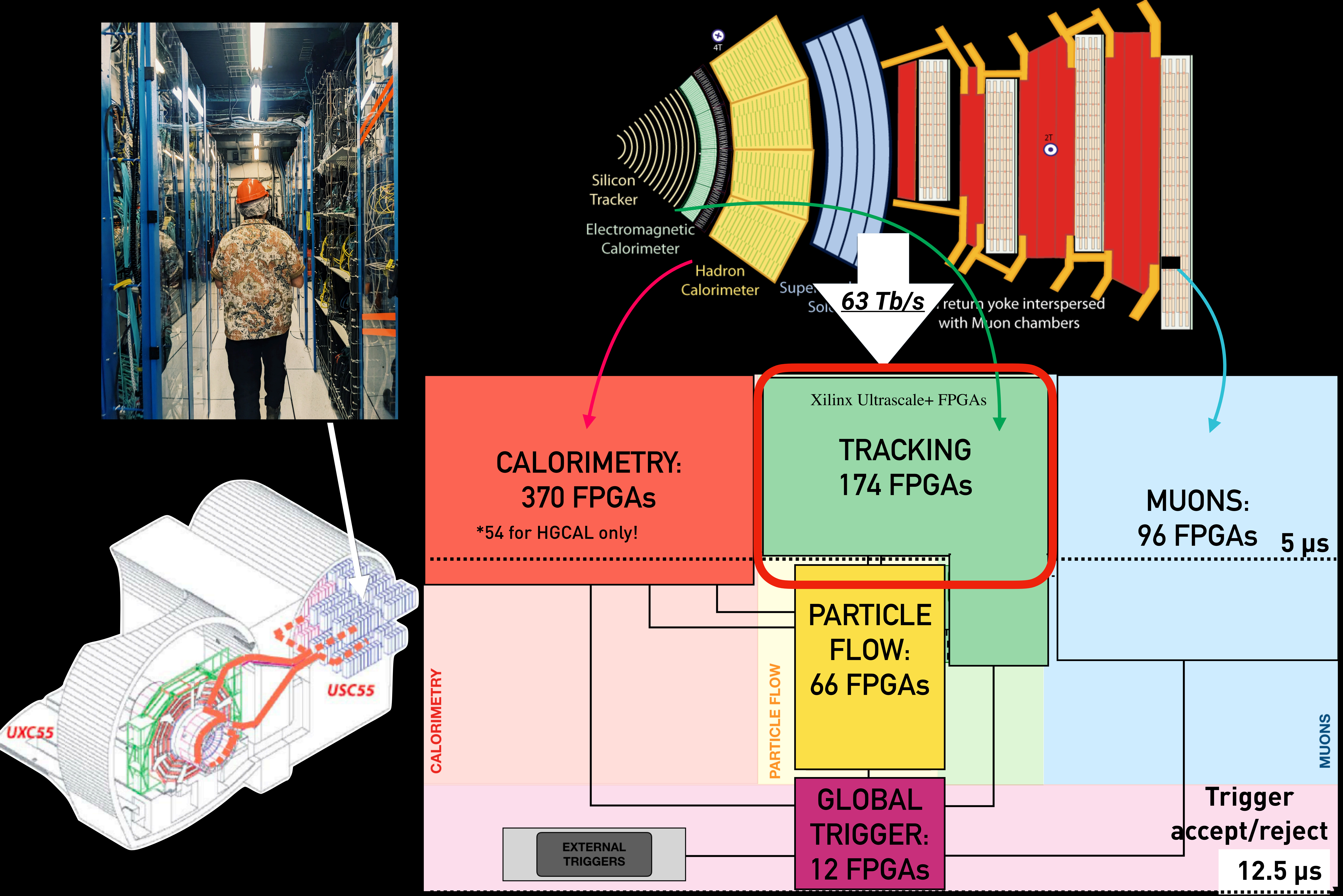
- 2 Tb/s → **63 Tb/s**

Latency

- 4 μs → **12 μs**

Extremely high data complexity,

Extremely little time



Xilinx Ultrascale+ FPGAs

CALORIMETRY: 370 FPGAs

*54 for HGCAL only!

TRACKING 174 FPGAs

MUONS: 96 FPGAs

5 μs

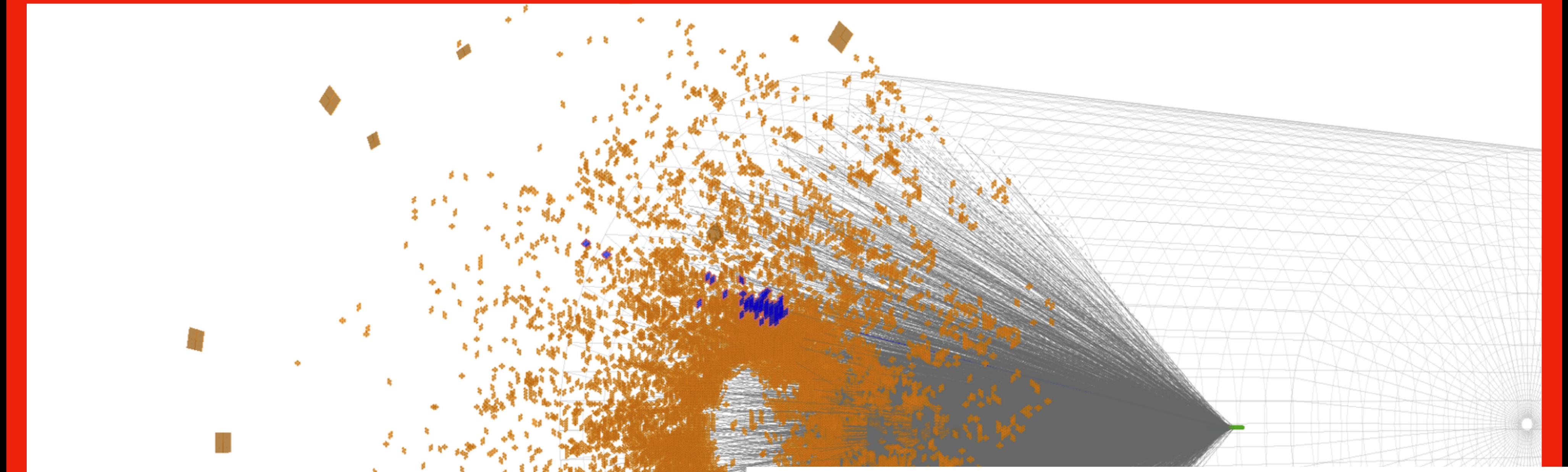
PARTICLE FLOW: 66 FPGAs

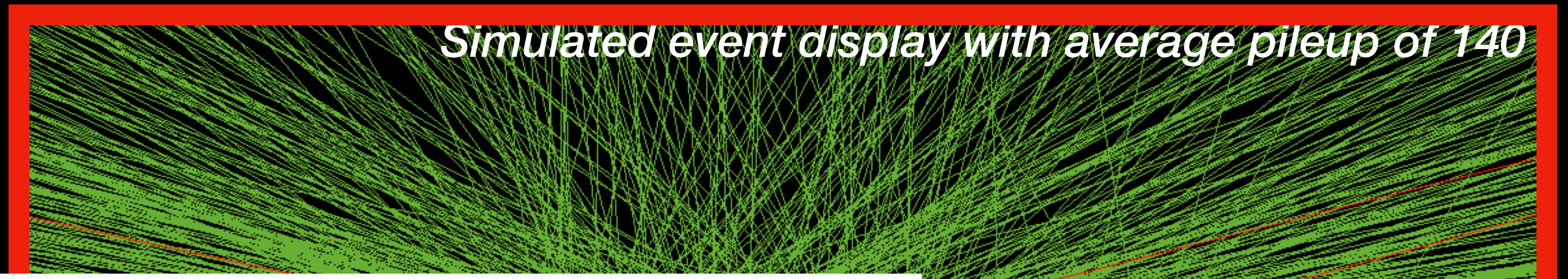
GLOBAL TRIGGER: 12 FPGAs

EXTERNAL TRIGGERS

Trigger accept/reject

12.5 μs

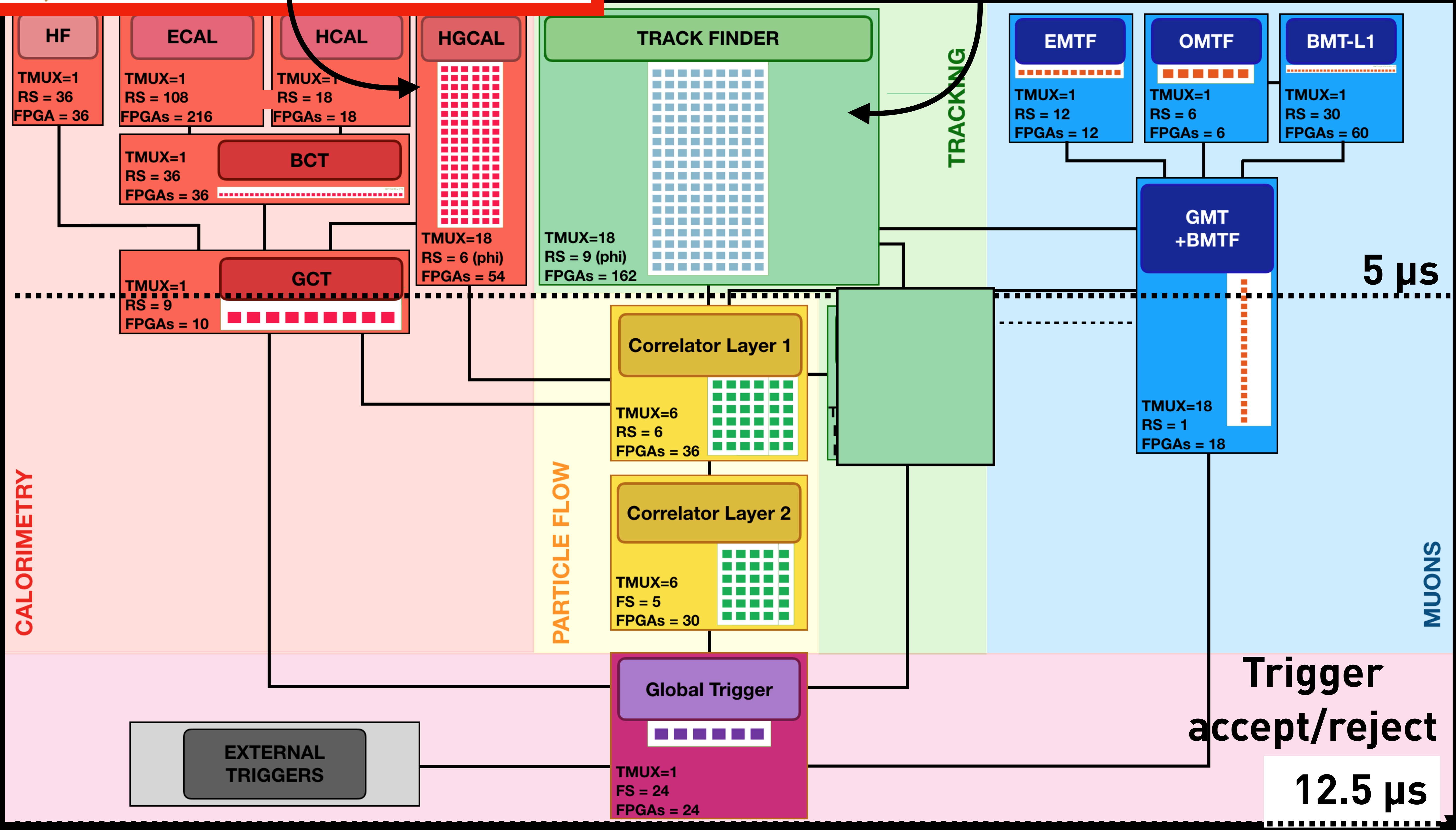




Simulated event display with average pileup of 140

12 microseconds latency

Processing 5% of internet traffic



HF
TMUX=1
RS = 36
FPGA = 36

ECAL
TMUX=1
RS = 108
FPGAs = 216

HCAL
TMUX=1
RS = 18
FPGAs = 18

HGCAL
TMUX=18
RS = 6 (phi)
FPGAs = 54

BCT
TMUX=1
RS = 36
FPGAs = 36

GCT
TMUX=1
RS = 9
FPGAs = 10

TRACK FINDER
TMUX=18
RS = 9 (phi)
FPGAs = 162

TRACKING

EMTF
TMUX=1
RS = 12
FPGAs = 12

OMTF
TMUX=1
RS = 6
FPGAs = 6

BMT-L1
TMUX=1
RS = 30
FPGAs = 60

GMT
+BMTF
TMUX=18
RS = 1
FPGAs = 18

5 μs

Correlator Layer 1
TMUX=6
RS = 6
FPGAs = 36

Correlator Layer 2
TMUX=6
FS = 5
FPGAs = 30

CALORIMETRY

PARTICLE FLOW

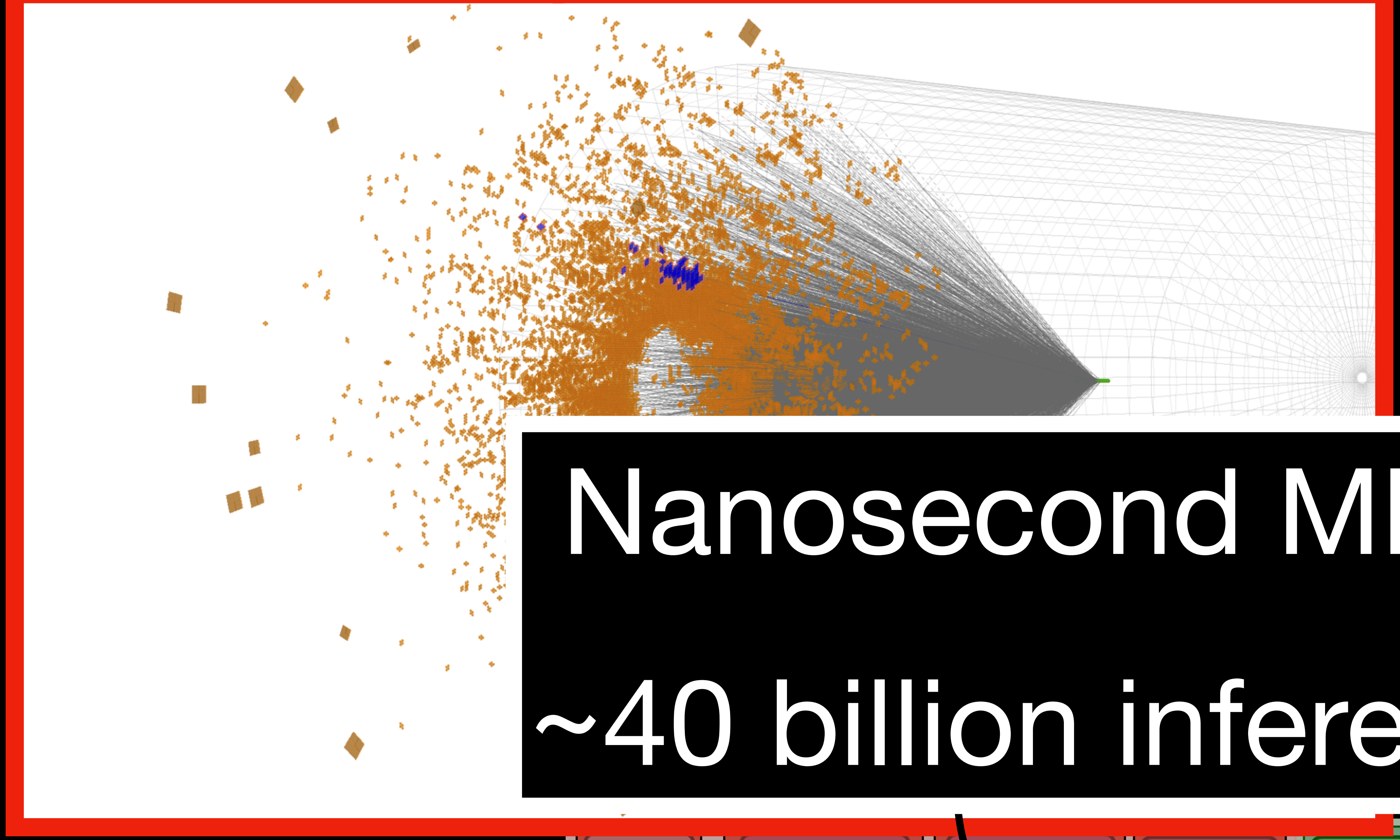
MUONS

Global Trigger
TMUX=1
FS = 24
FPGAs = 24

EXTERNAL TRIGGERS

Trigger accept/reject

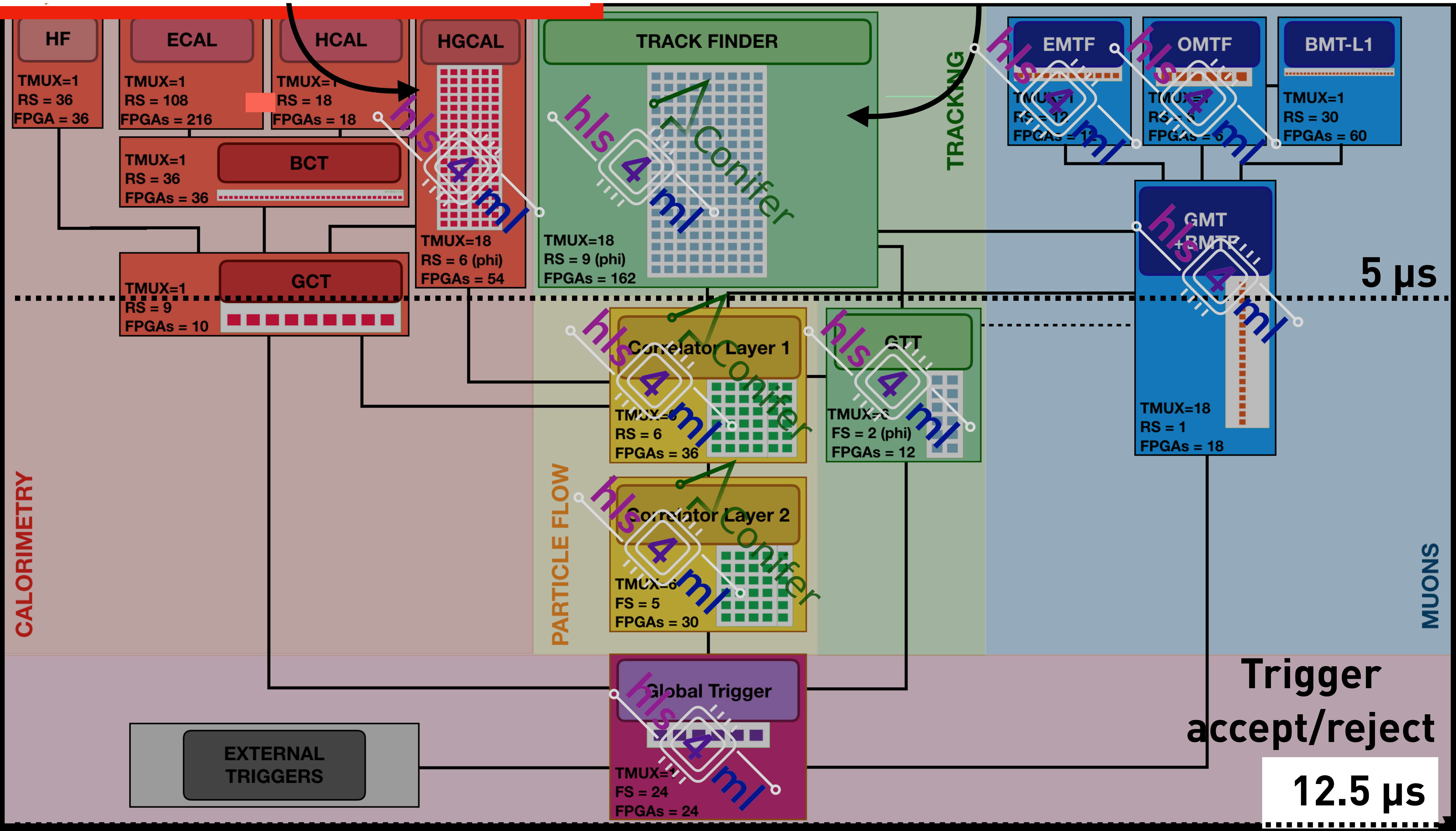
12.5 μs

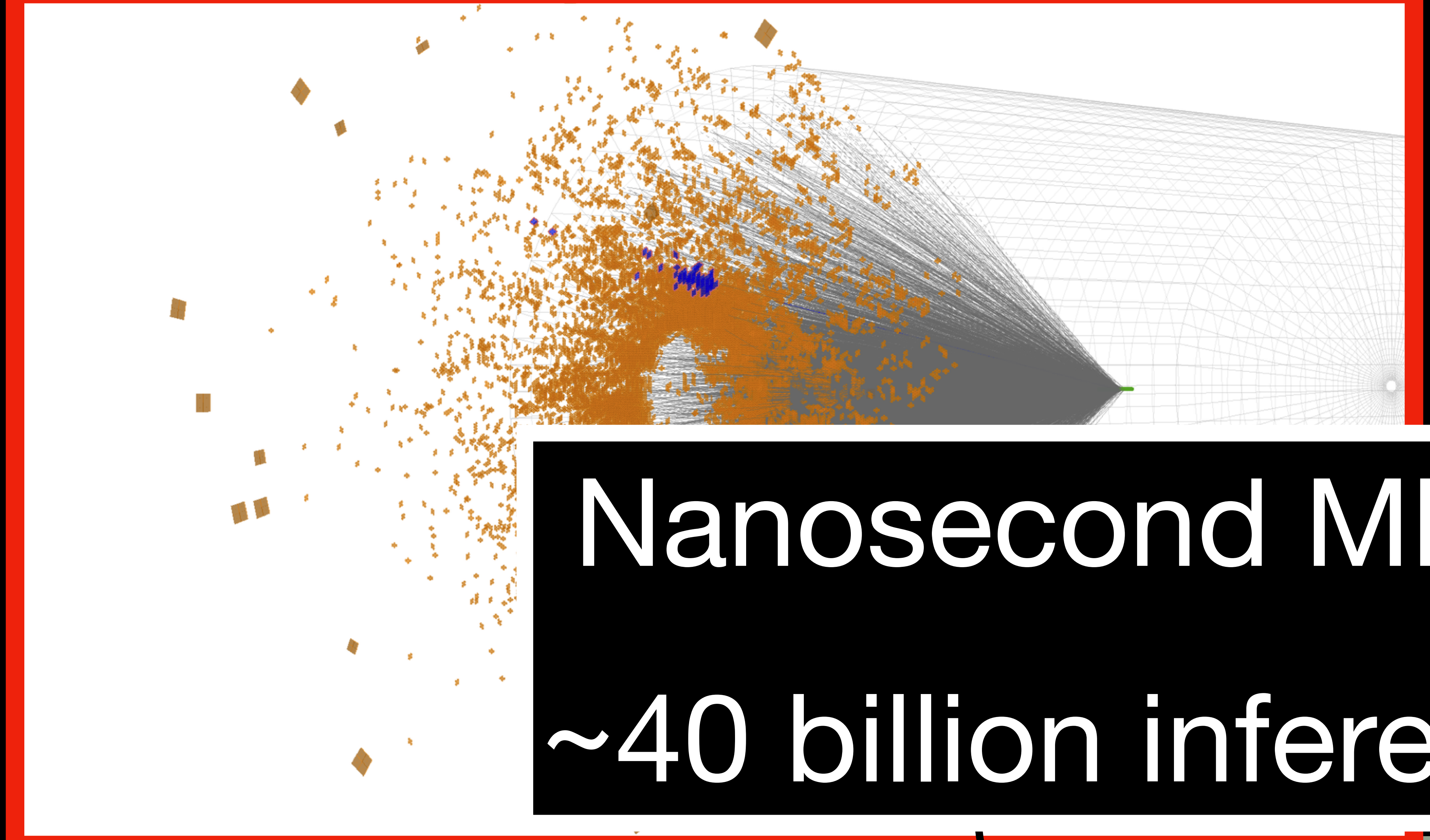



Simulated event display with average pileup of 140

Nanosecond ML inference on FPGAs!

~40 billion inferences/s during HL-LHC

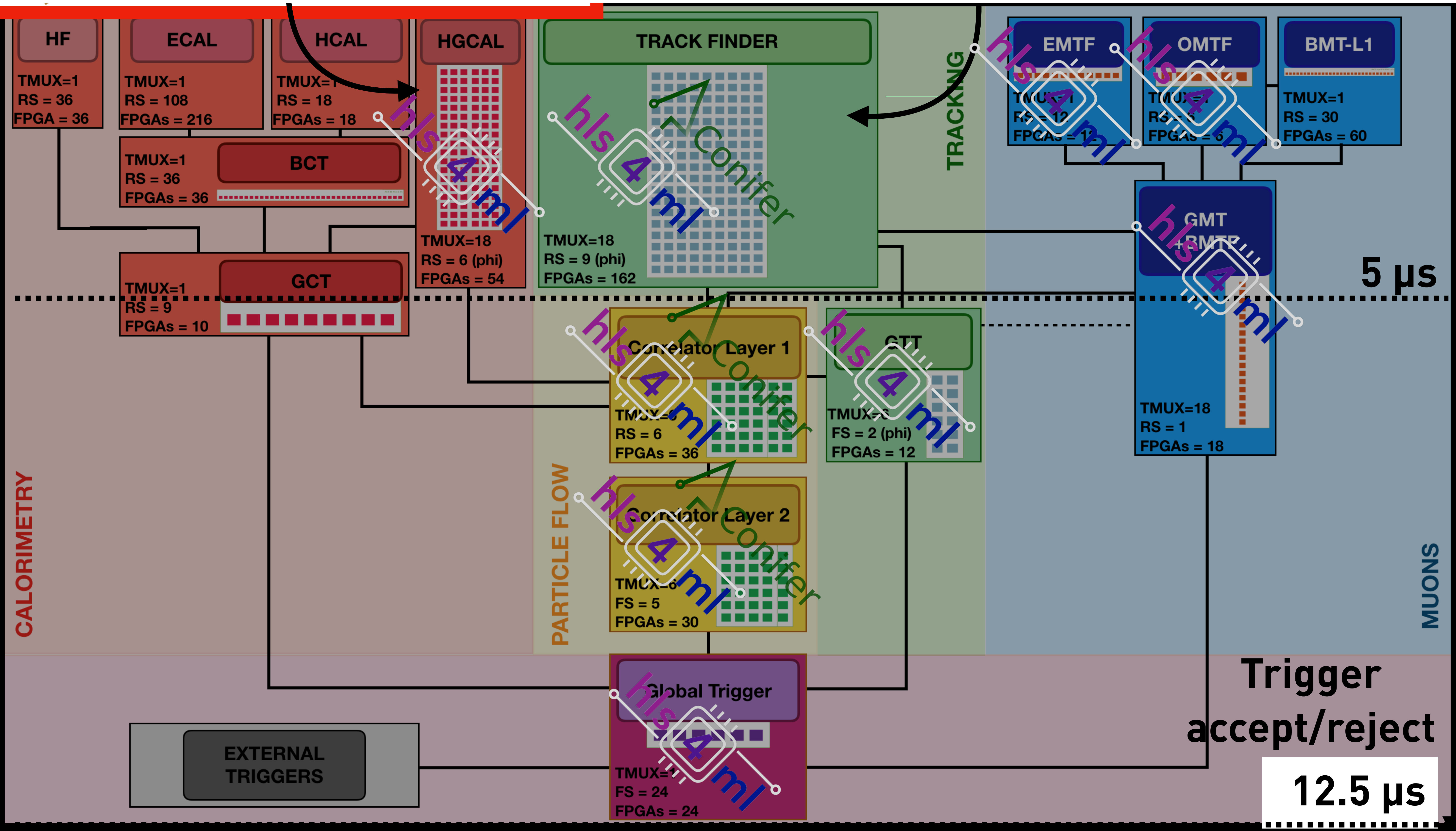




Simulated event display with average pileup of 140

Nanosecond ML inference on FPGAs!

~40 billion inferences/s during HL-LHC



Conifer hls4ml

HEP developed libraries for fast ML on FPGAs

Why FPGAs?

Why FPGAs?

- Latency (resource parallelism)

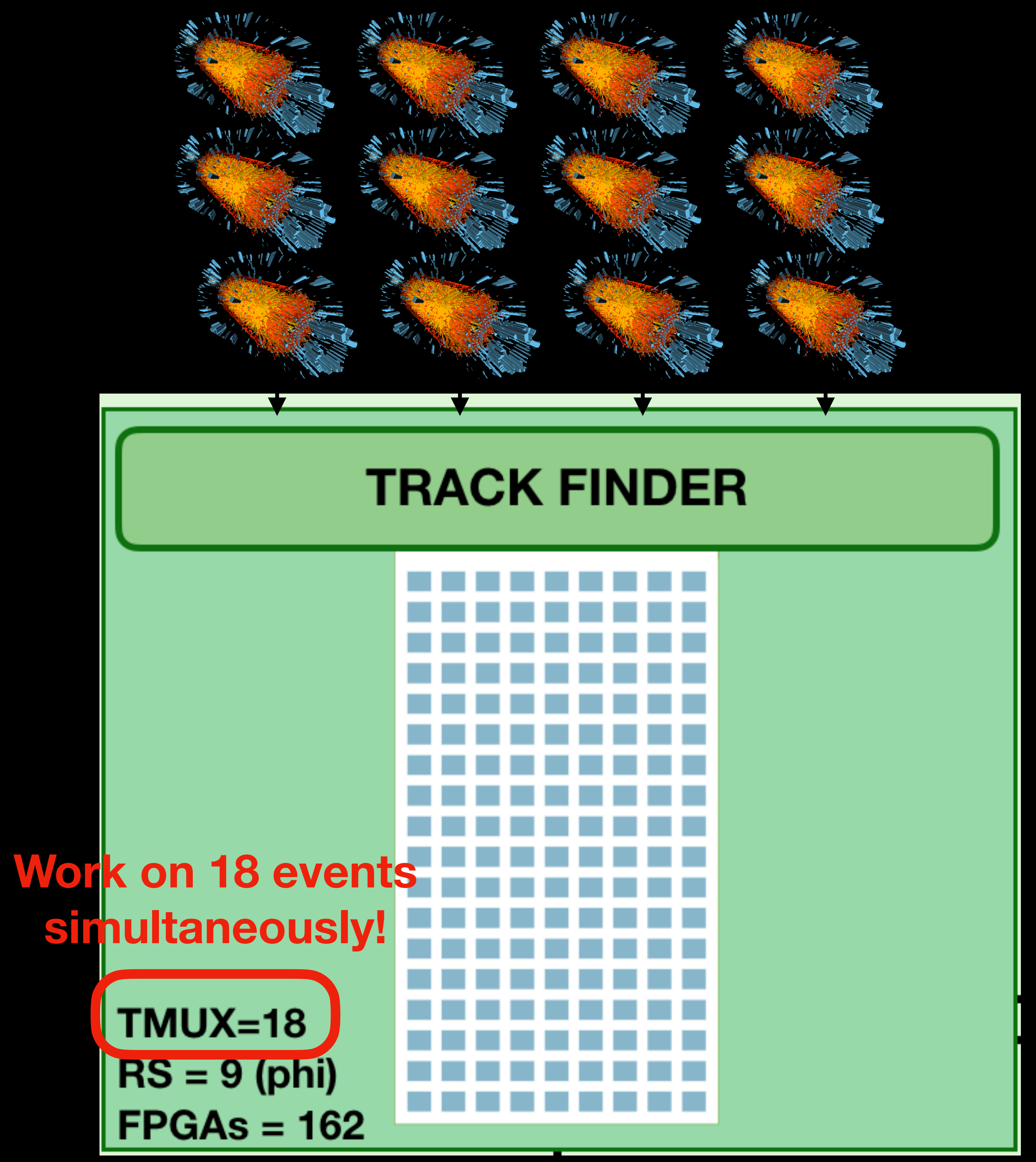
Why FPGAs?

- Throughput (pipeline parallelism)

pipeline

parallelism

- Latency (resource parallelism)
  - Can work on different parts of problem, different data simultaneously
  - Latency strictly limited by detector frontend buffer
- High bandwidth (pipeline parallelism)
  - Phase 2 L1T processes 5% of total internet traffic
- Latency deterministic
  - CPU/GPU processing randomness,
  - FPGAs repeatable and predictable latency



$$\mathbf{x}_n = g_n(\mathbf{W}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{b}_n)$$

activation function

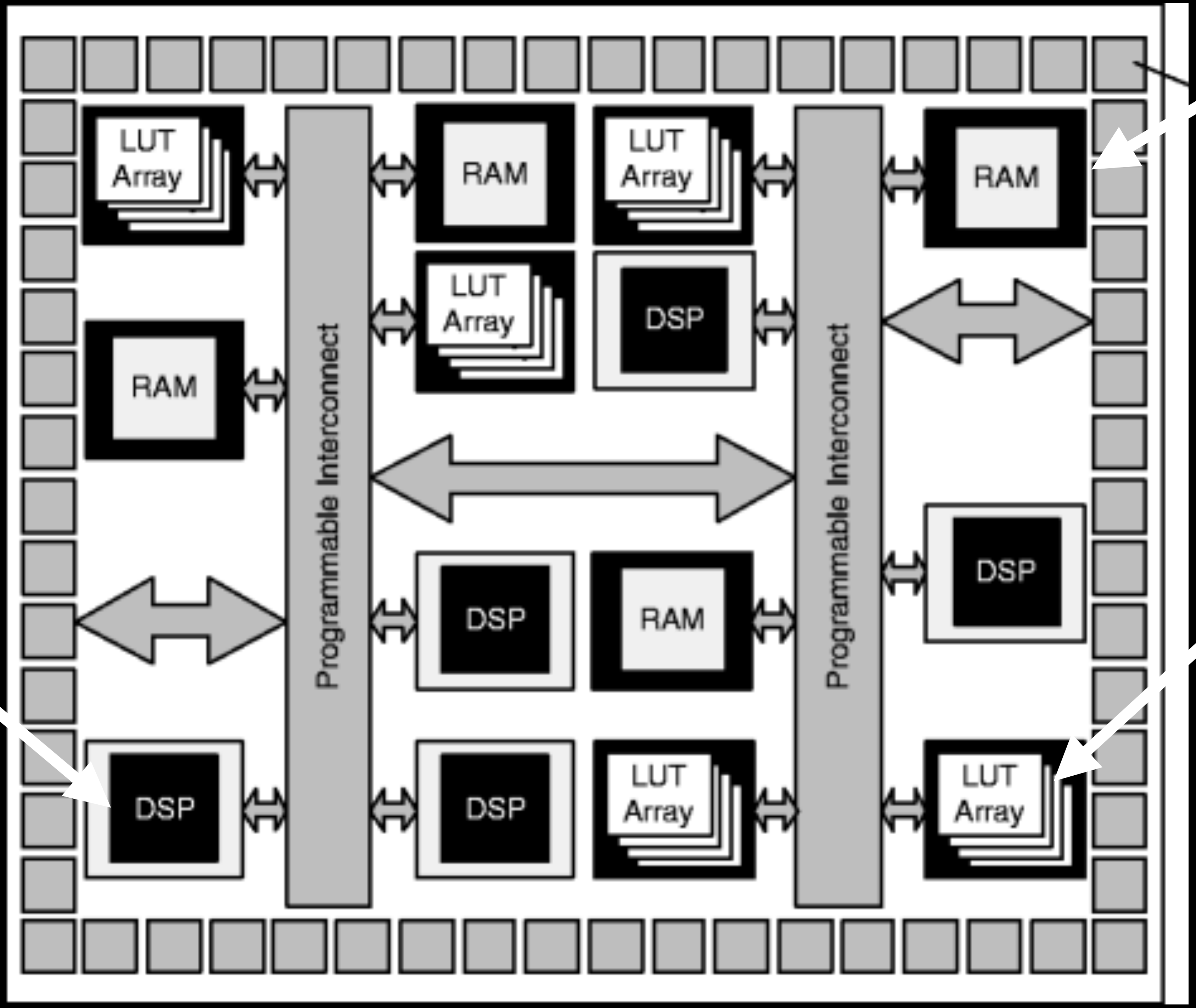
precomputed and
stored in BRAMs

multiplication

DSPs

addition

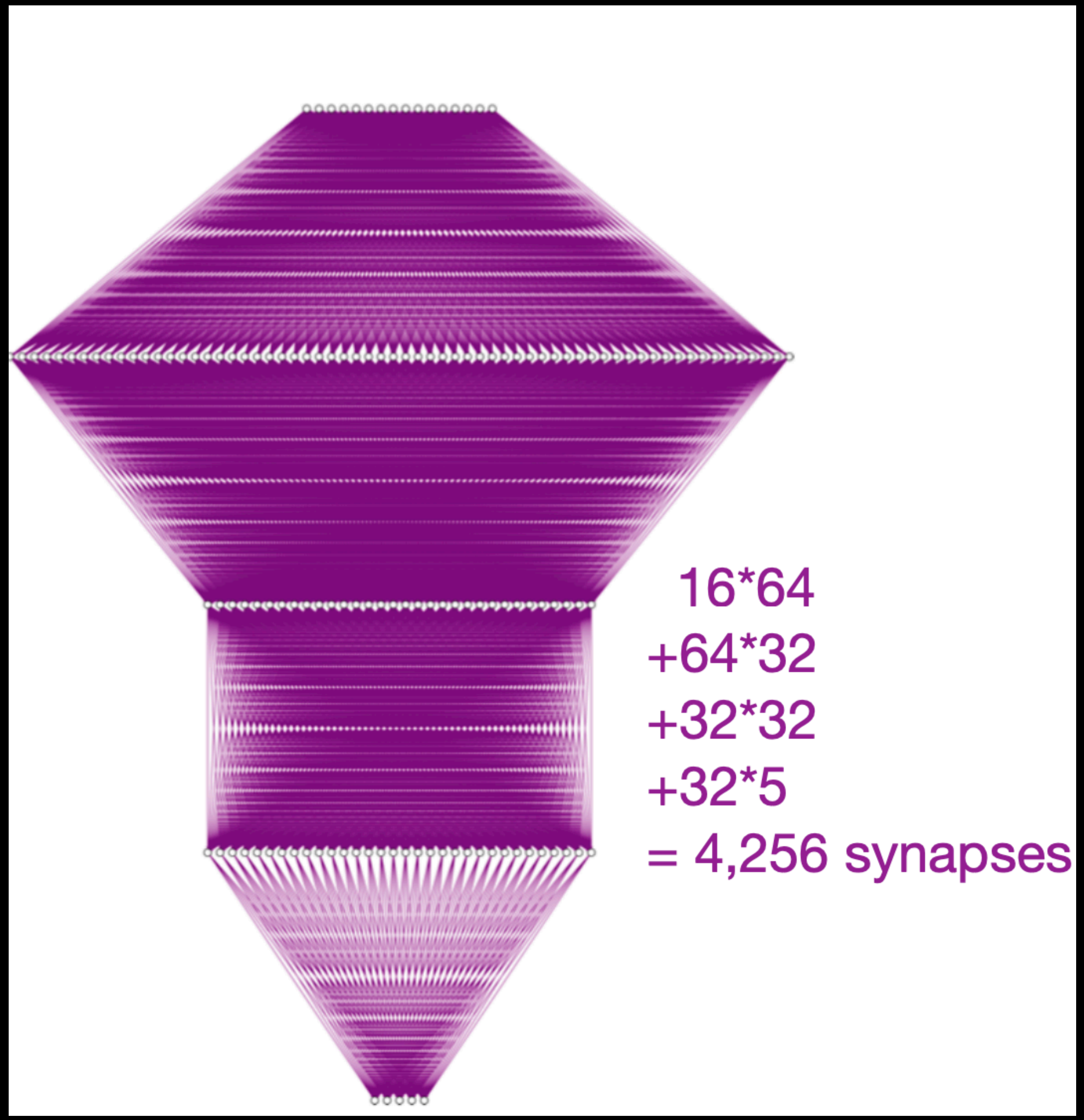
logic cells



Memory (BRAM)
O(2000) units

Digital signal processors (DSPs)
O(5,000) units

Logic cells/lookup tables (LUTs)
O(1) million units



16*64
+64*32
+32*32
+32*5
= 4,256 synapses

$$\mathbf{x}_n = g_n(\mathbf{W}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{b}_n)$$

activation function

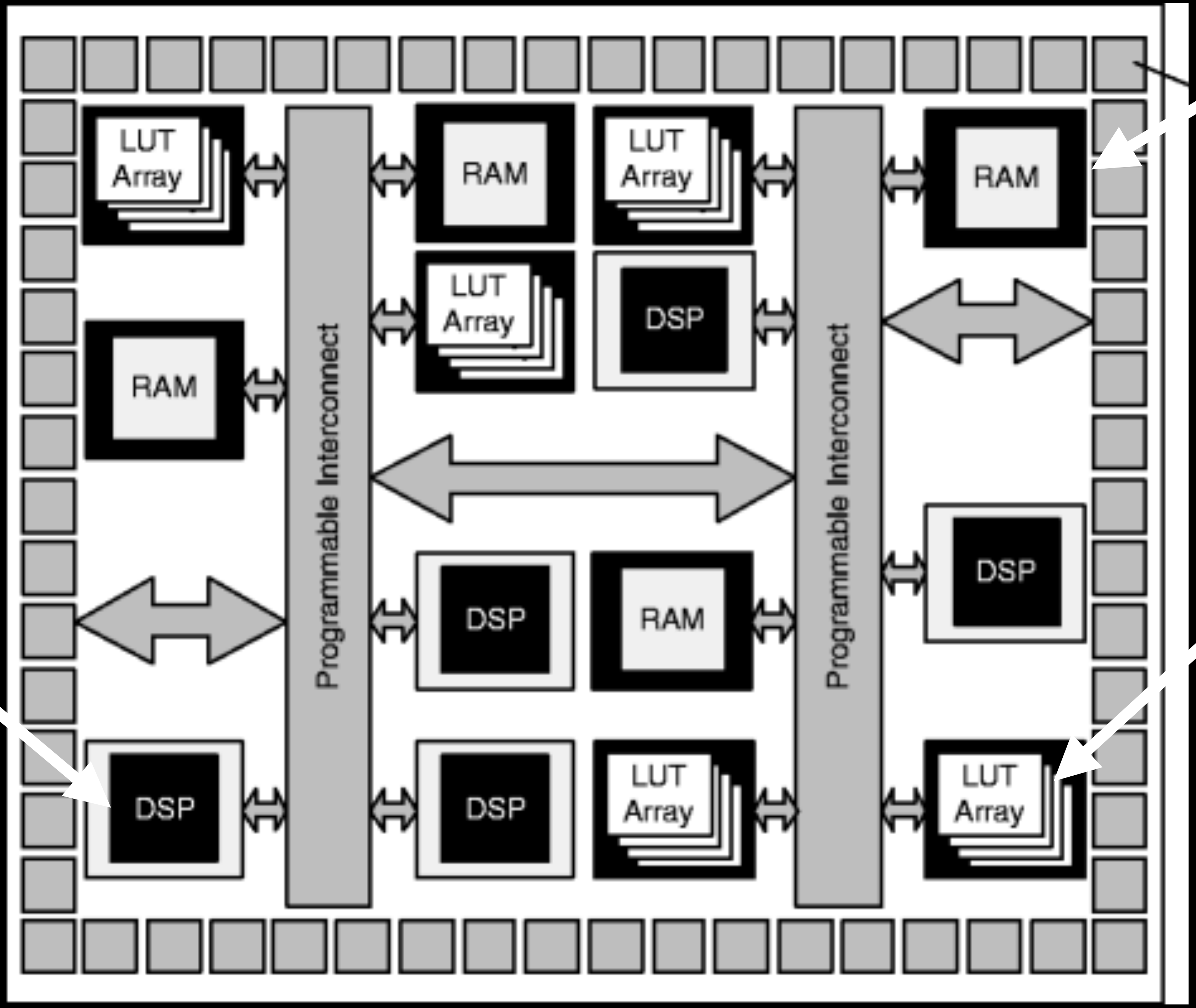
precomputed and
stored in BRAMs

multiplication

DSPs

addition

logic cells



Memory (BRAM)
O(2000) units

Digital signal processors (DSPs)
O(5,000) units

Logic cells/lookup tables (LUTs)
O(1) million units

C/C++ algorithm

Constraints/ Directives

Xilinx

Intel

Siemens

Vitis HLS

Vivado HLS

Vivado Accelerator

Intel HLS

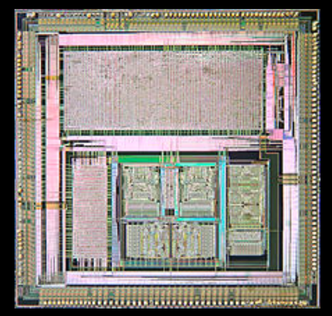
Catapult HLS

(CMS, ATLAS)

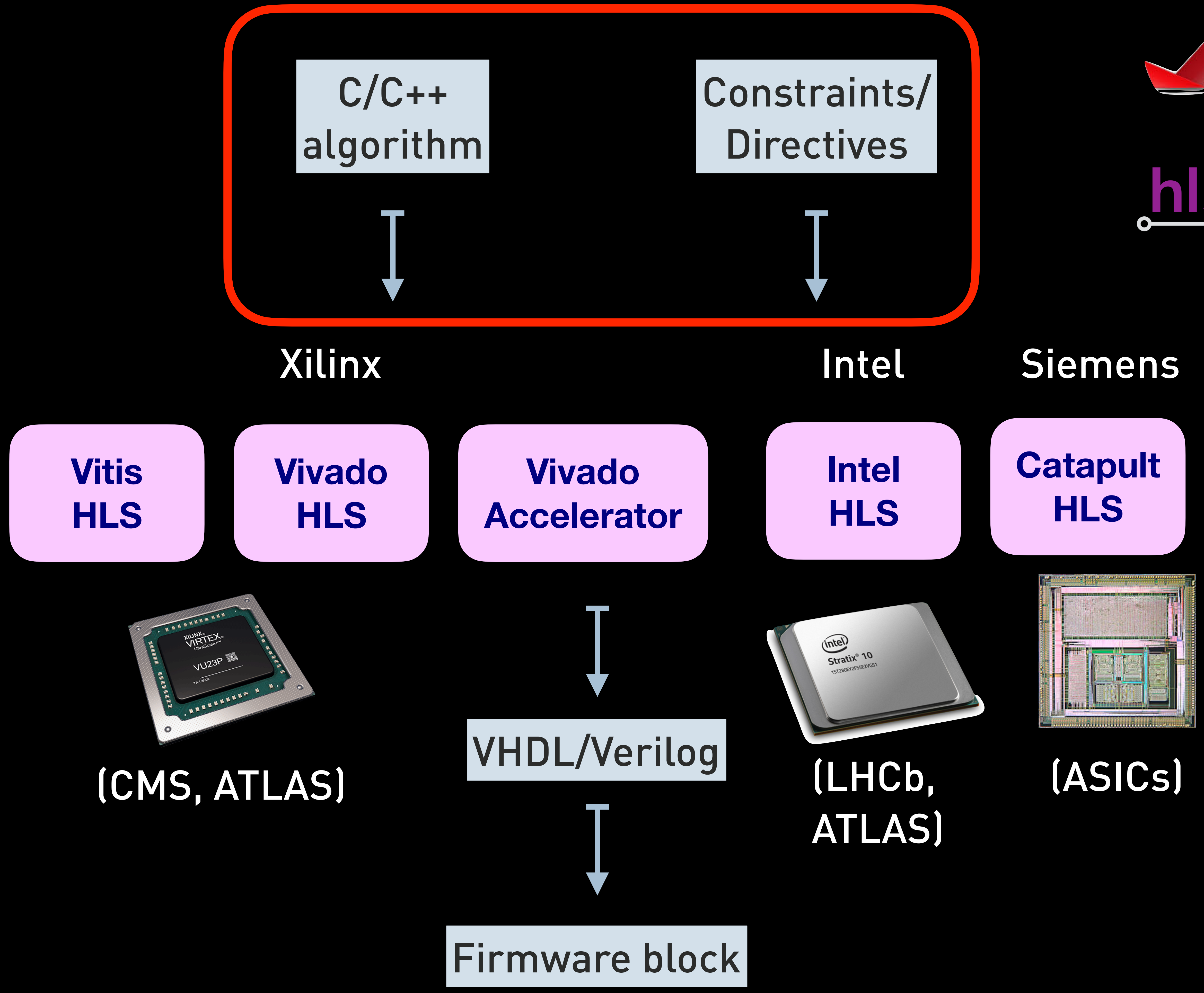
VHDL/Verilog

Firmware block

(LHCb, ATLAS)



(ASICs)



C/C++ algorithm

Constraints/ Directives

Xilinx

Intel

Siemens

Vitis HLS

Vivado HLS

Vivado Accelerator

Intel HLS

Catapult HLS

(CMS, ATLAS)

VHDL/Verilog

Firmware block

(LHCb, ATLAS)

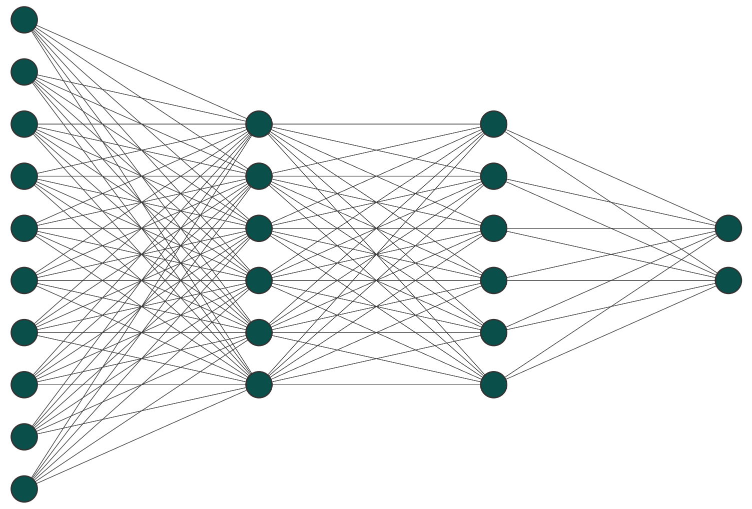
(ASICs)

FINN

hls4ml

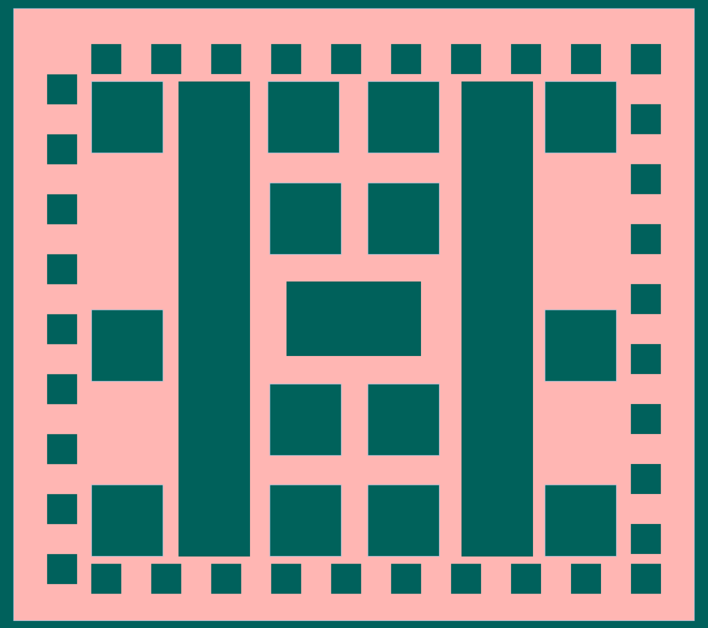
https://github.com/fastmachinelearning/hls4ml
https://fastmachinelearning.org/hls4ml/

KERAS / PyTorch / ONNX

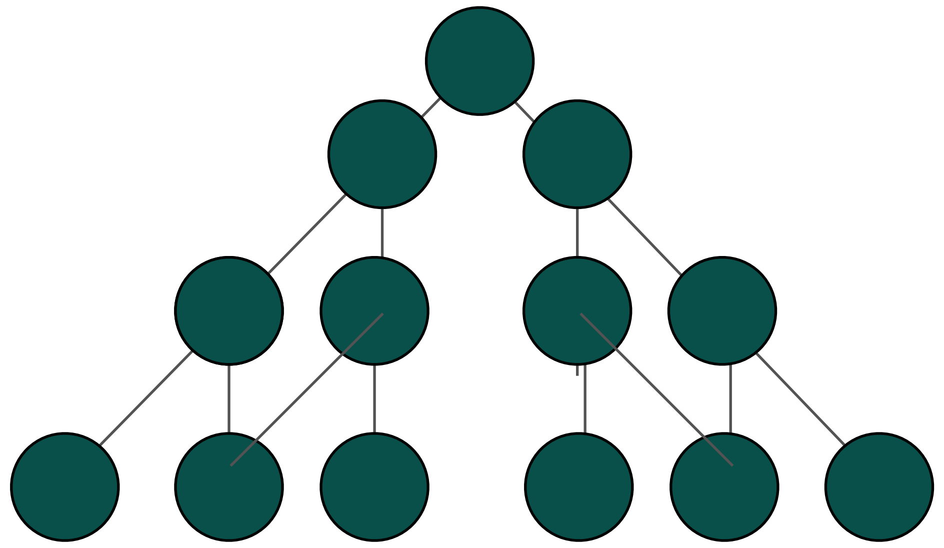


→ **hls4ml** →

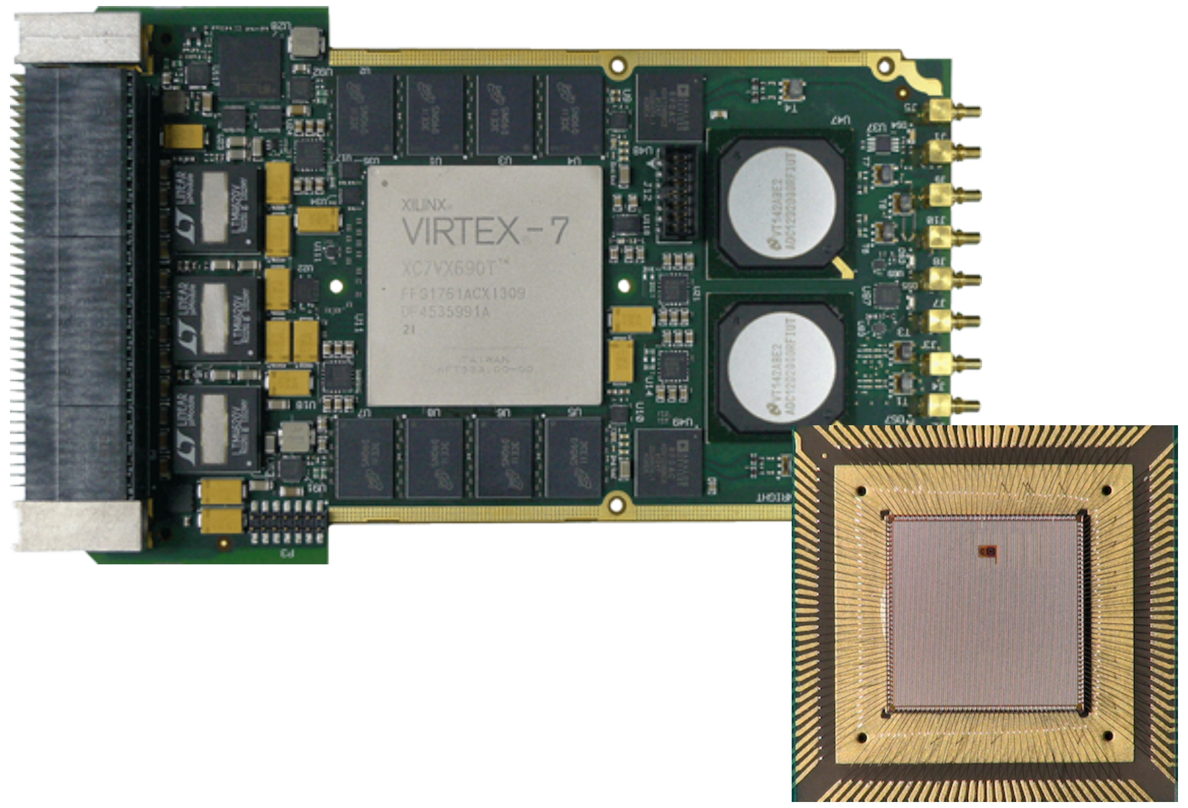
HLS project:
Vivado / Vitis / Intel Quartus / IntelOne API / Catapult



TensorFlow DF / scikit-learn / XGBoost



→ **Conifer** →



```
pip install hls4ml
pip install conifer
```

K + TensorFlow

ONNX

PYTORCH

Model (quantized/pruned)

Quantized:

QONNX

QKERAS

K + TensorFlow

ONNX

PYTORCH

Model (quantized/pruned)

→

hls4ml

Convert model to internal representation

Write HLS project targeting specified backend (configurable parallelization/ quantization)

Run emulation

Run synthesis

Quantized:

QONNX

QKERAS

Catapult HLS SIEMENS

Vivado™ HLS

XILINX VITIS™

intel HLS COMPILER

K + TensorFlow

ONNX

PYTORCH

Model (quantized/pruned)

→

Quantized:

QONNX

QKERAS

hls4ml

Convert model to internal representation

Write HLS project targeting specified backend (configurable parallelization/ quantization)

Run emulation

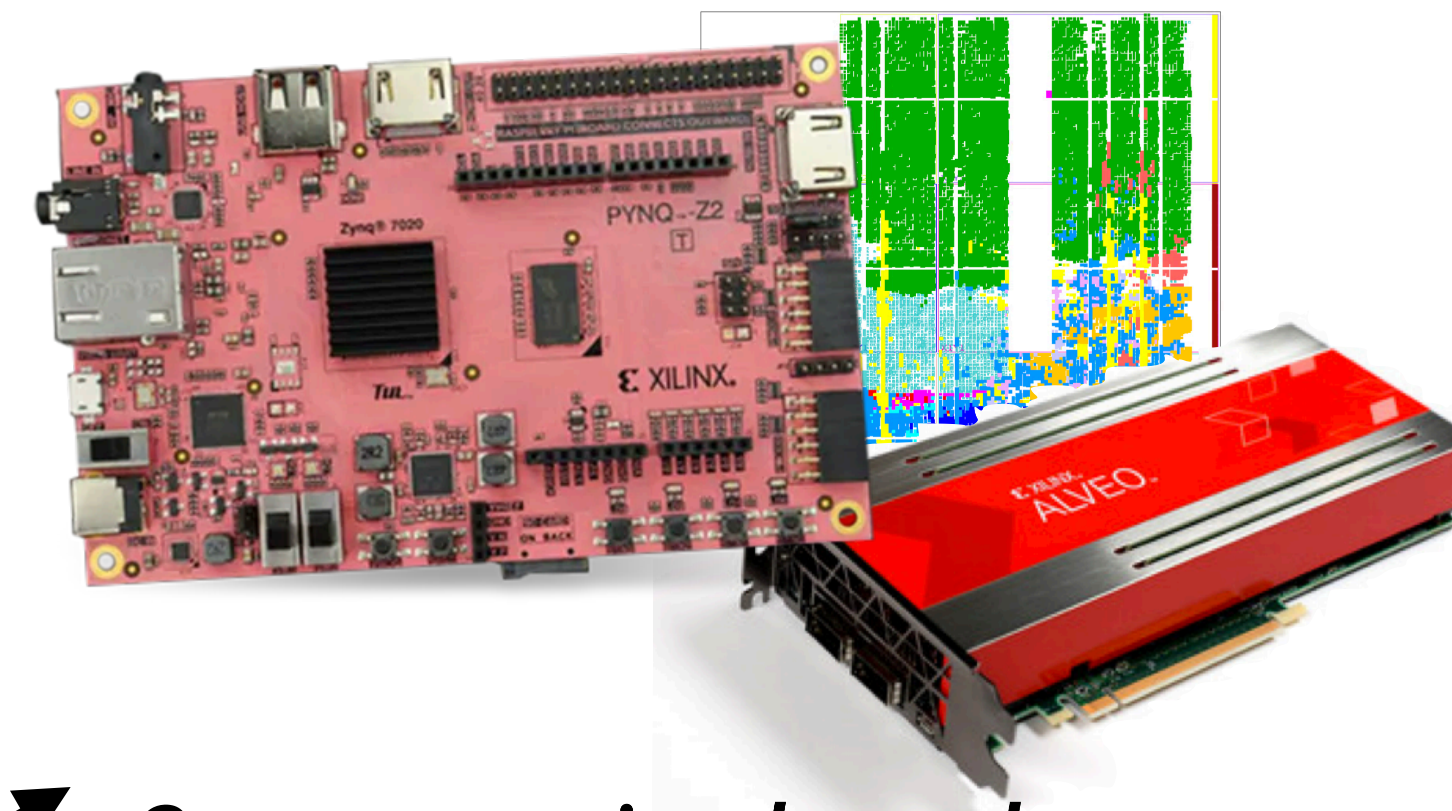
Run synthesis

Catapult HLS SIEMENS

Vivado™ HLS

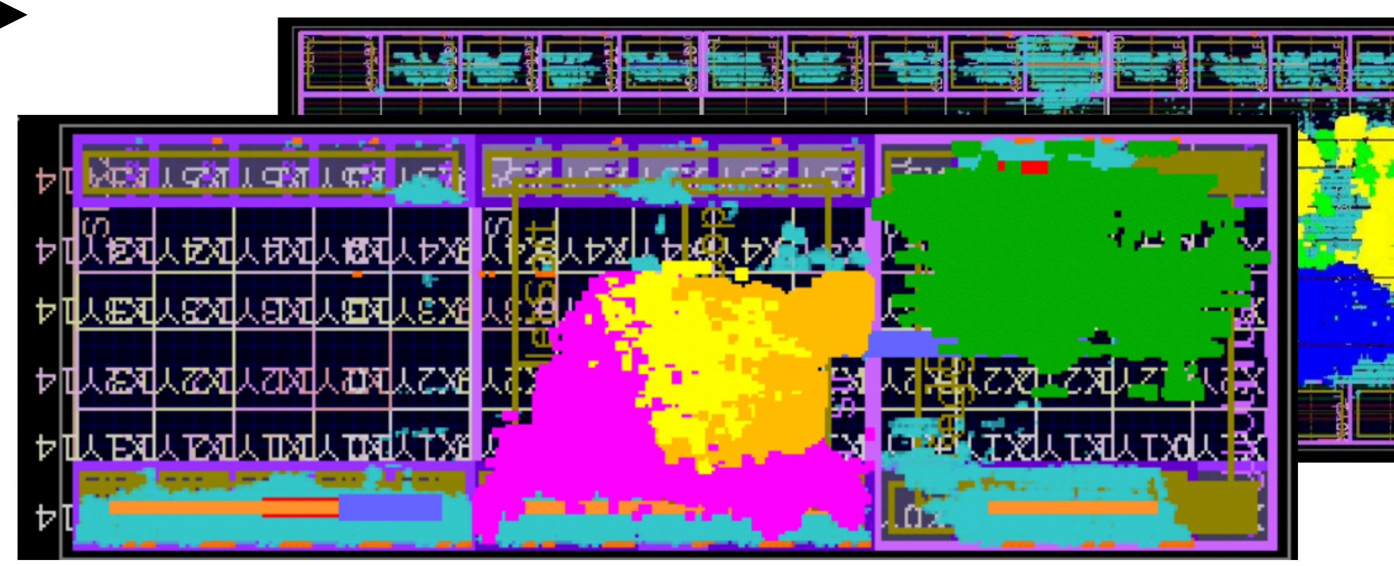
XILINX VITIS

HLS COMPILER

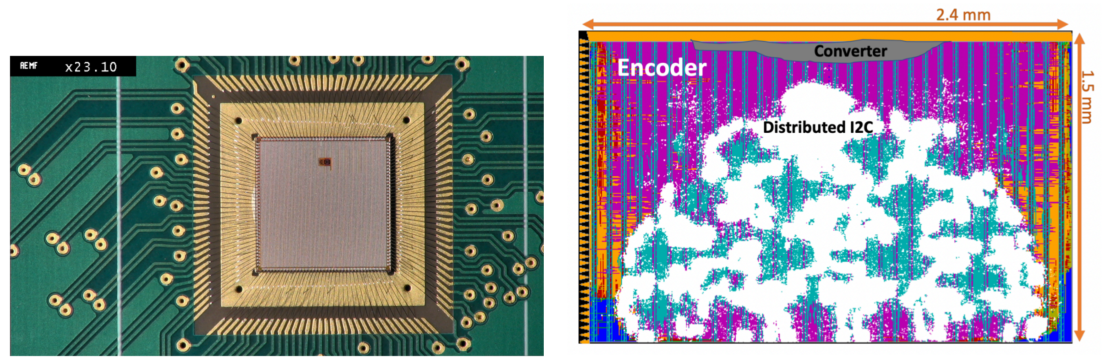


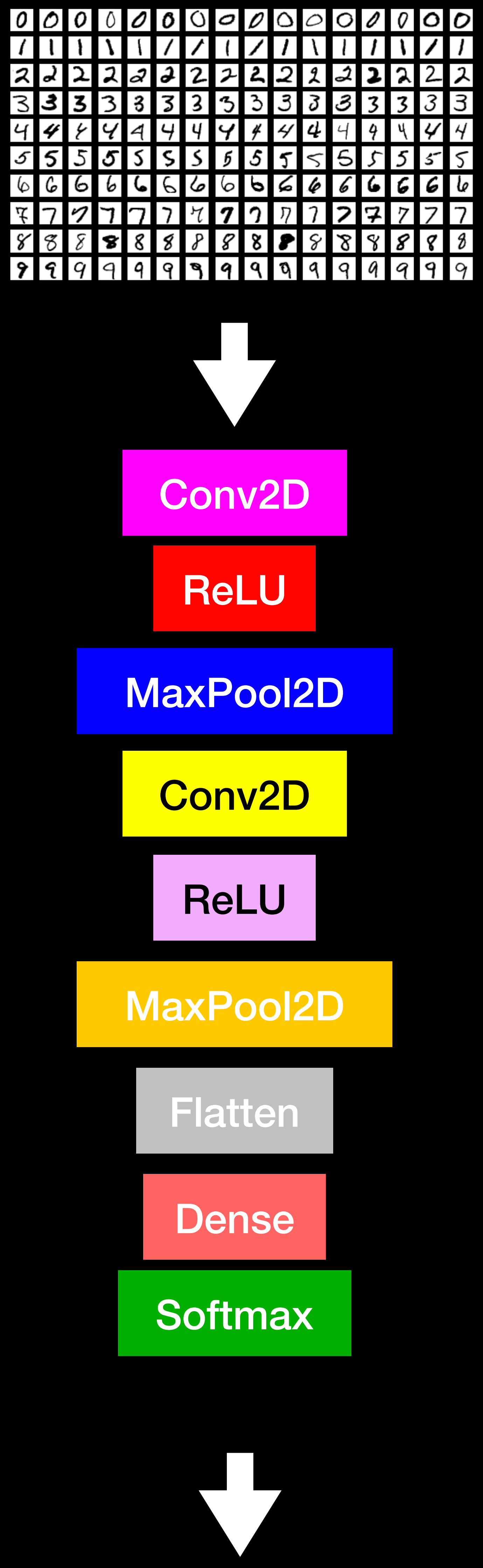
Co-processing kernel (Xilinx accelerators/SoCs)

FPGA custom designs (eg trigger algorithms)



ASICs





Conv2D

ReLU

MaxPool2D

Conv2D

ReLU

MaxPool2D

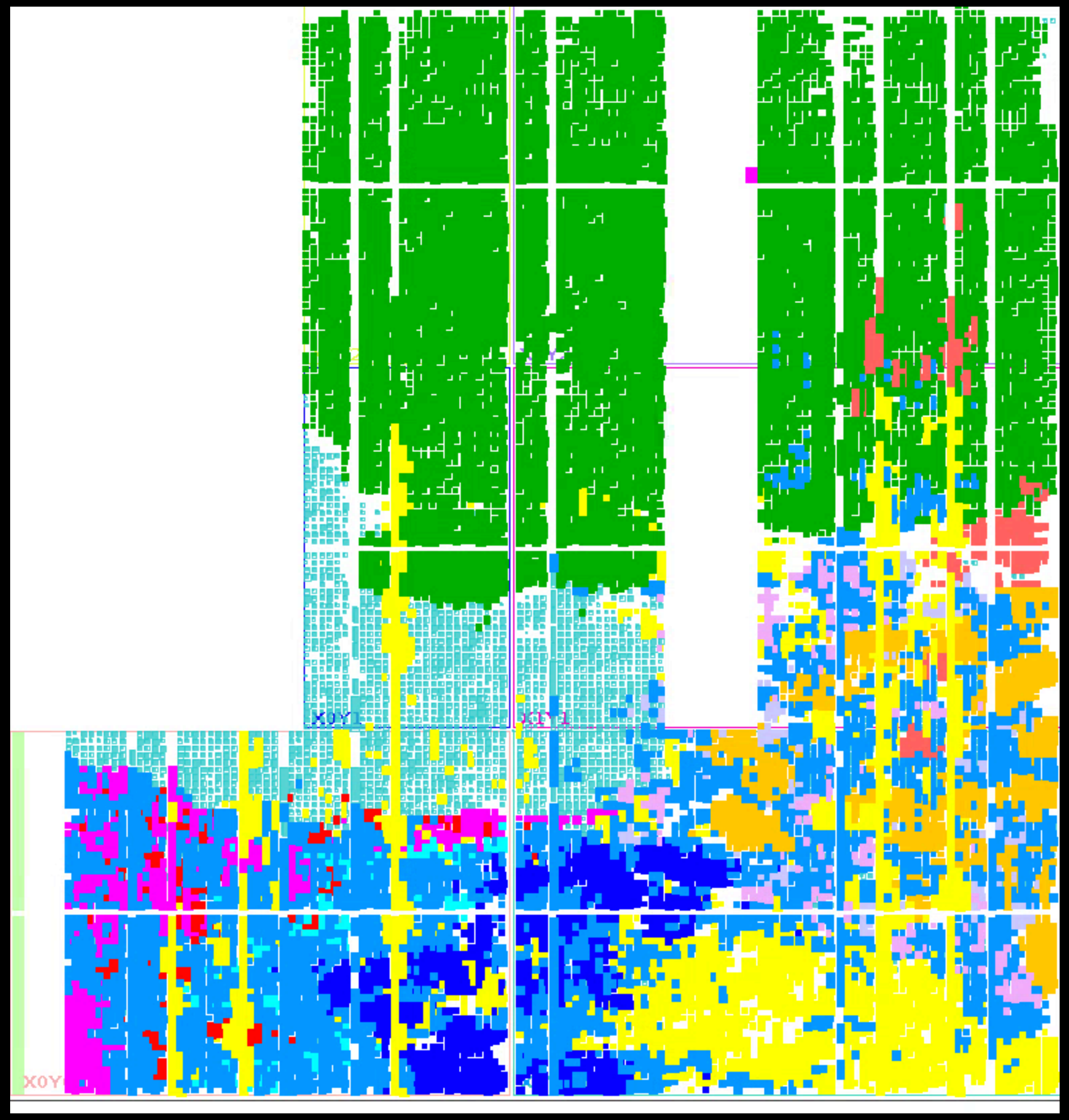
Flatten

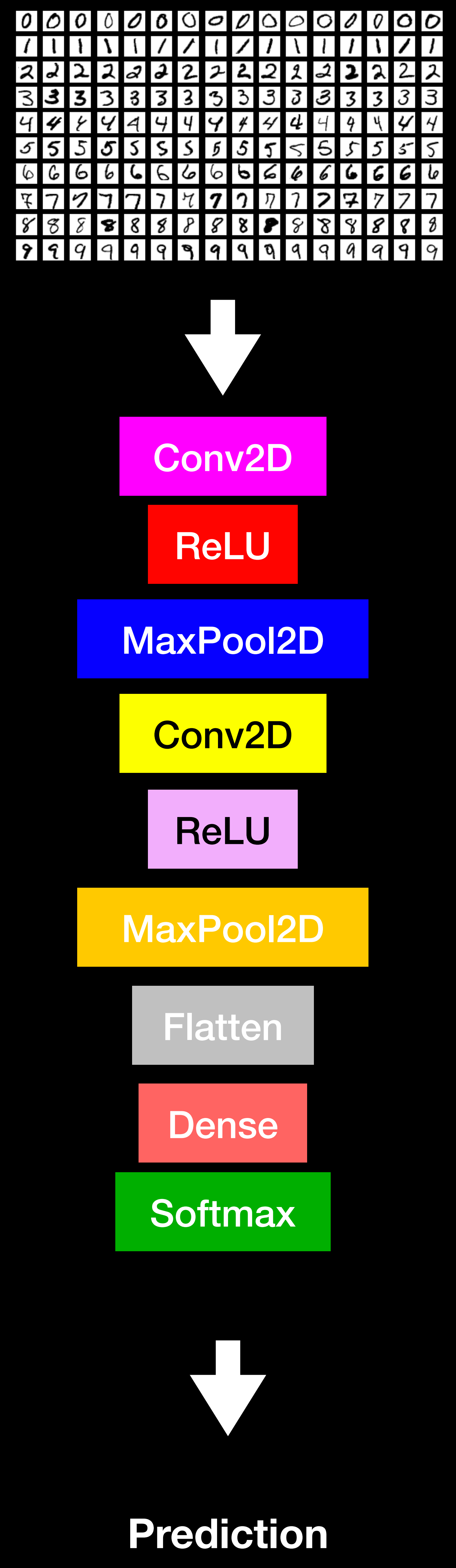
Dense

Softmax

Prediction

hls4ml





Conv2D

ReLU

MaxPool2D

Conv2D

ReLU

MaxPool2D

Flatten

Dense

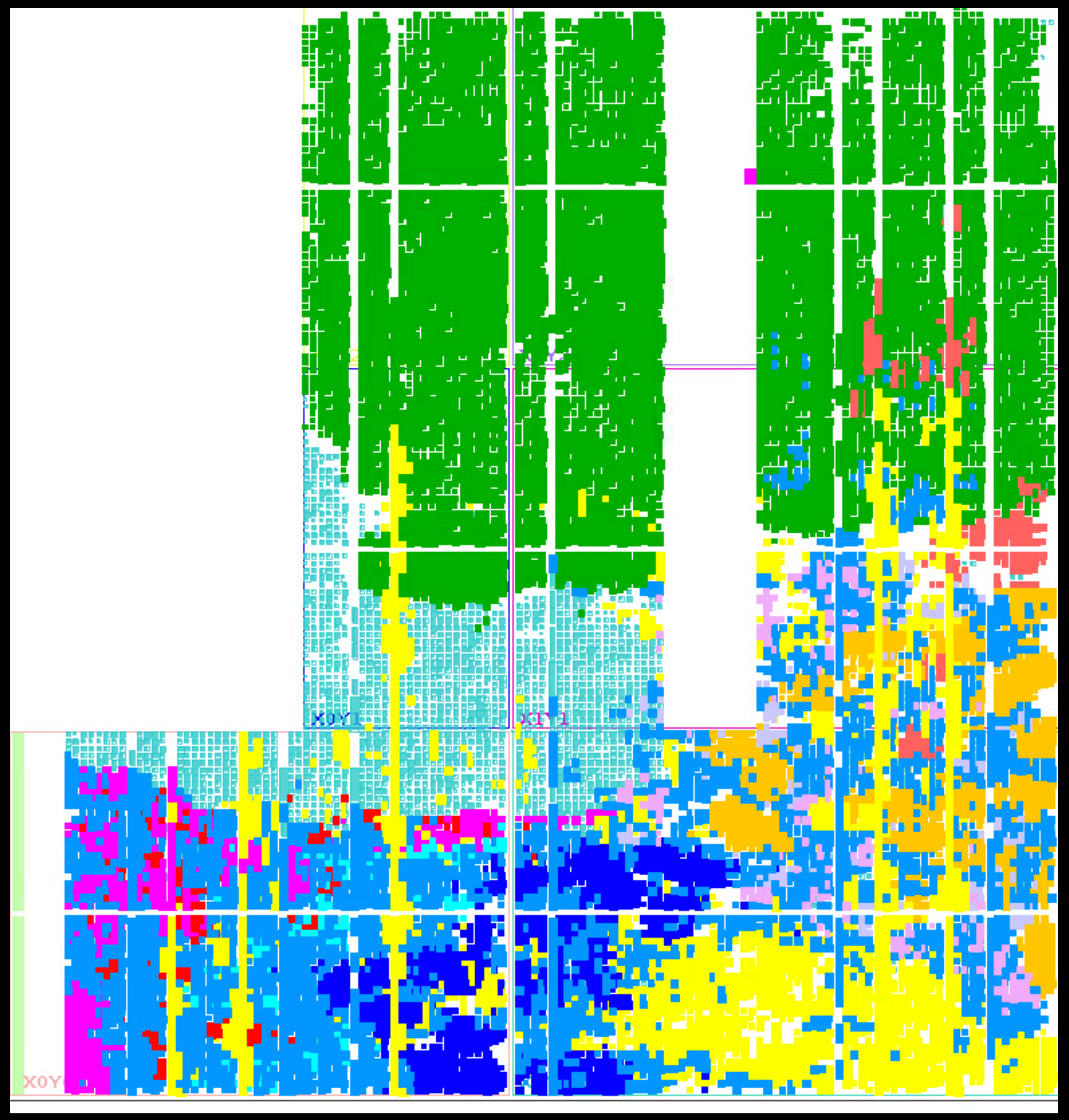
Softmax

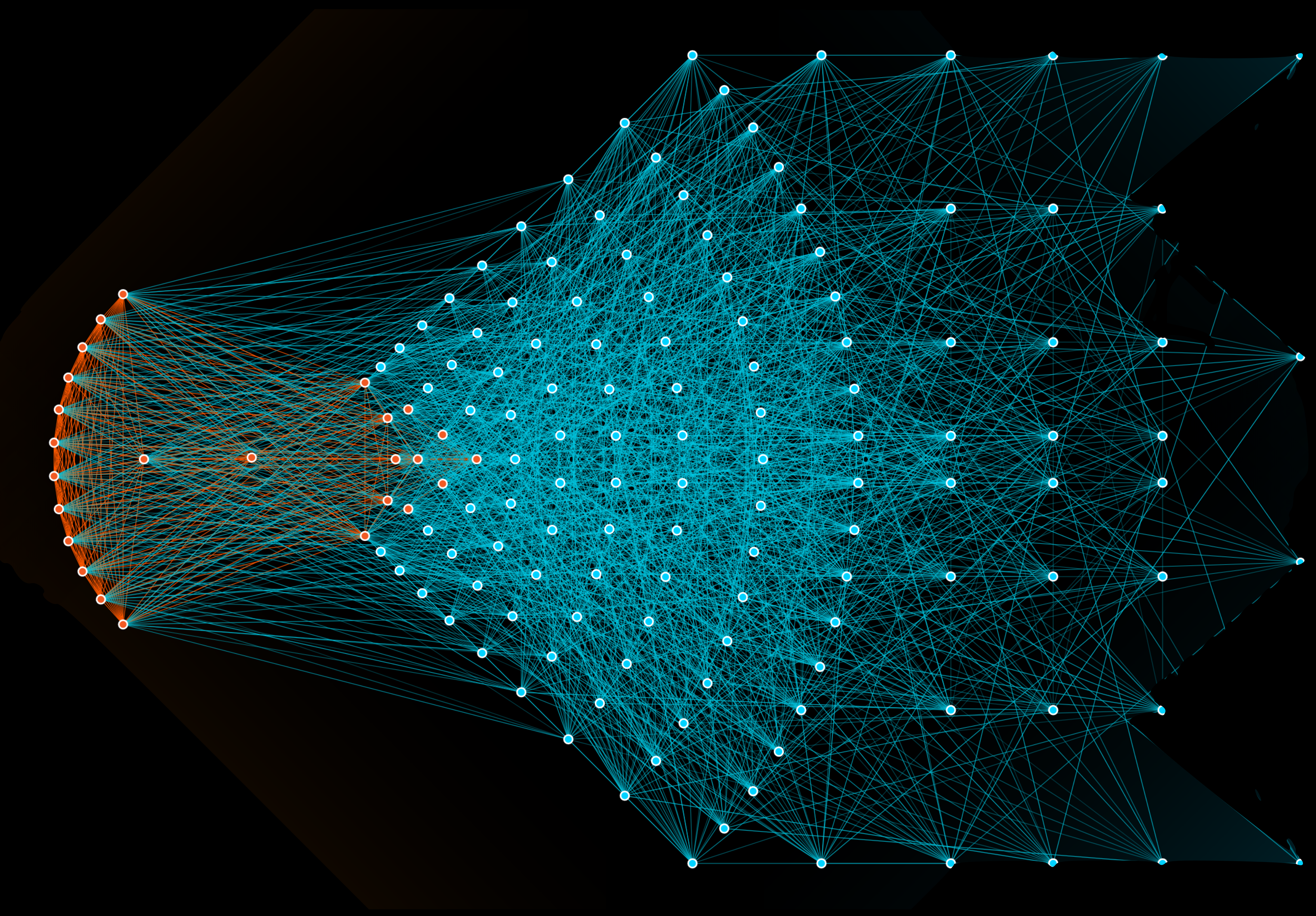
Prediction

hls4ml

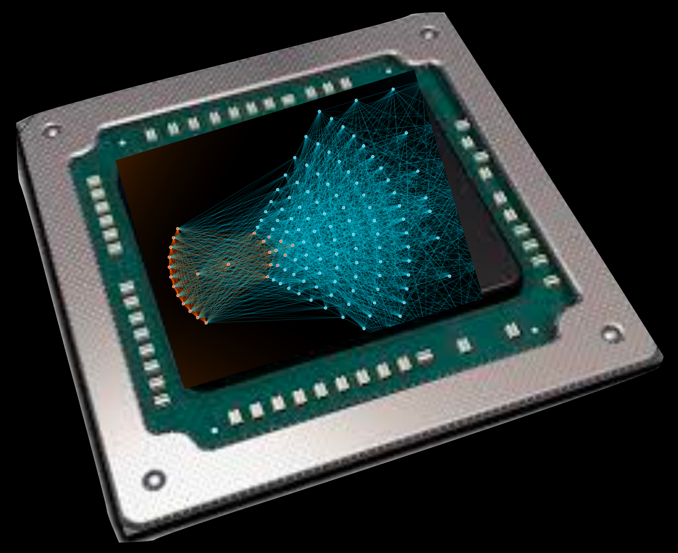
Data flow architecture

- Tailored hardware for a model
- Each layer is separate compute unit
- Stay on-chip
- “Decisions are design time”

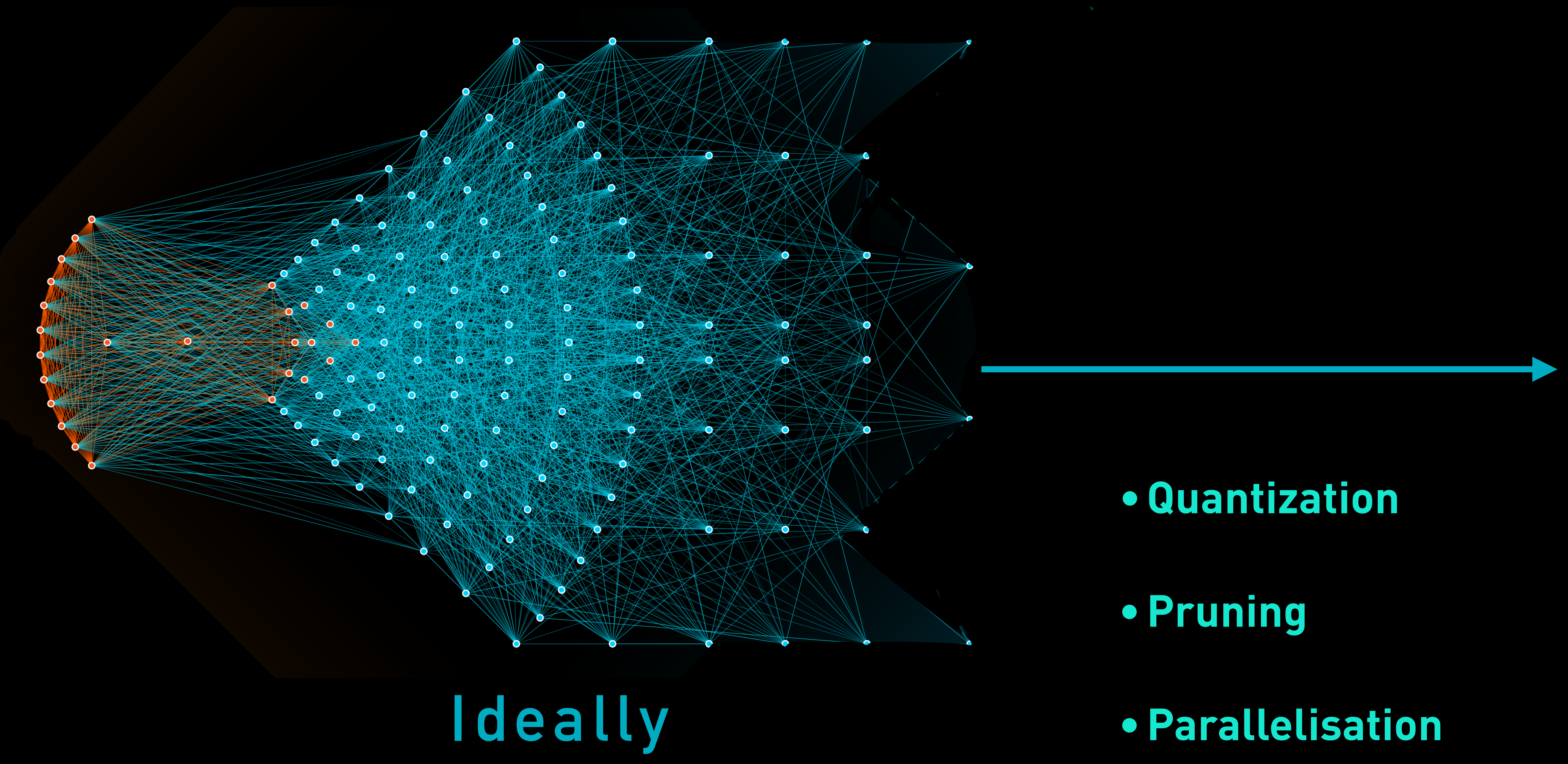




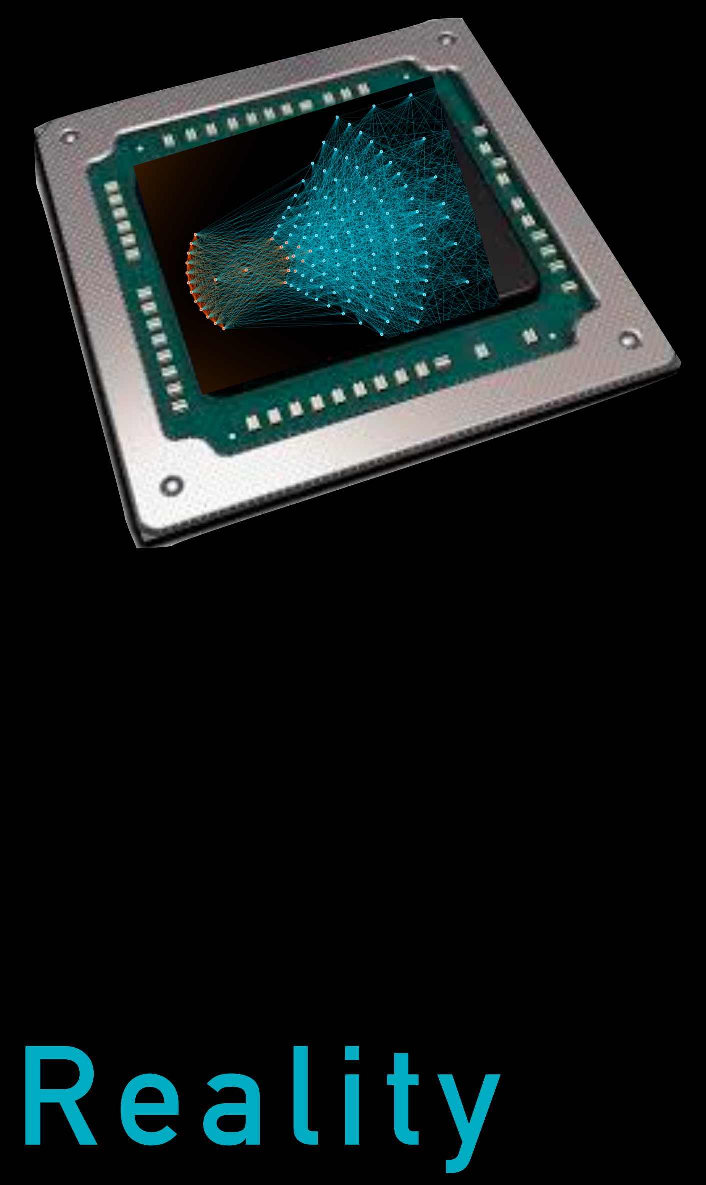
Ideally



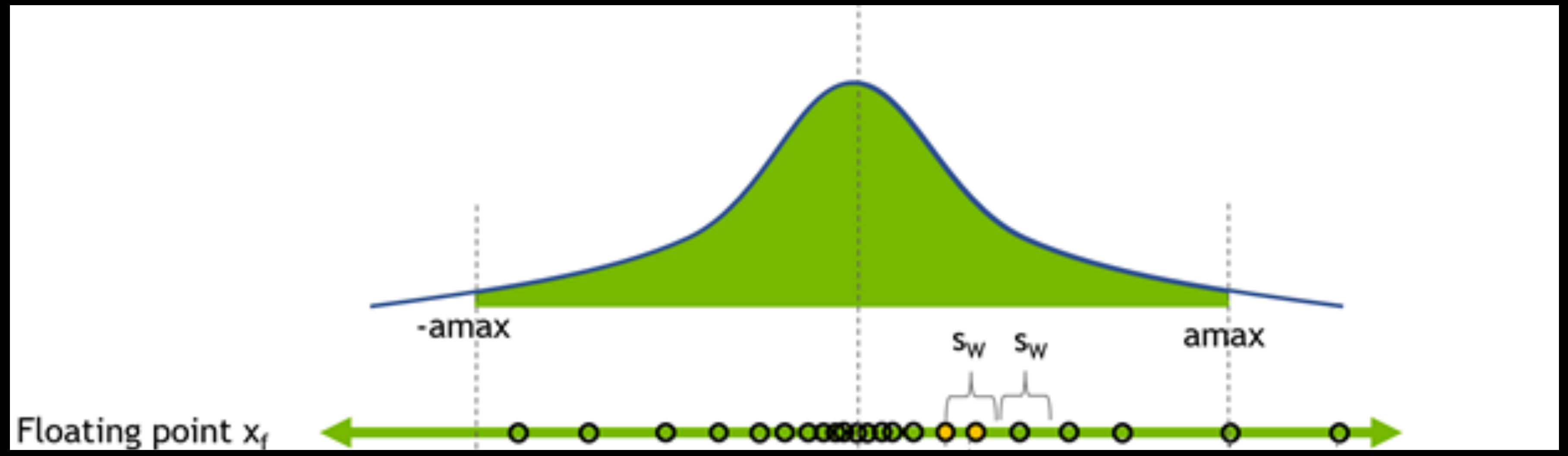
Reality



- Quantization
- Pruning
- Parallelisation
- Knowledge distillation

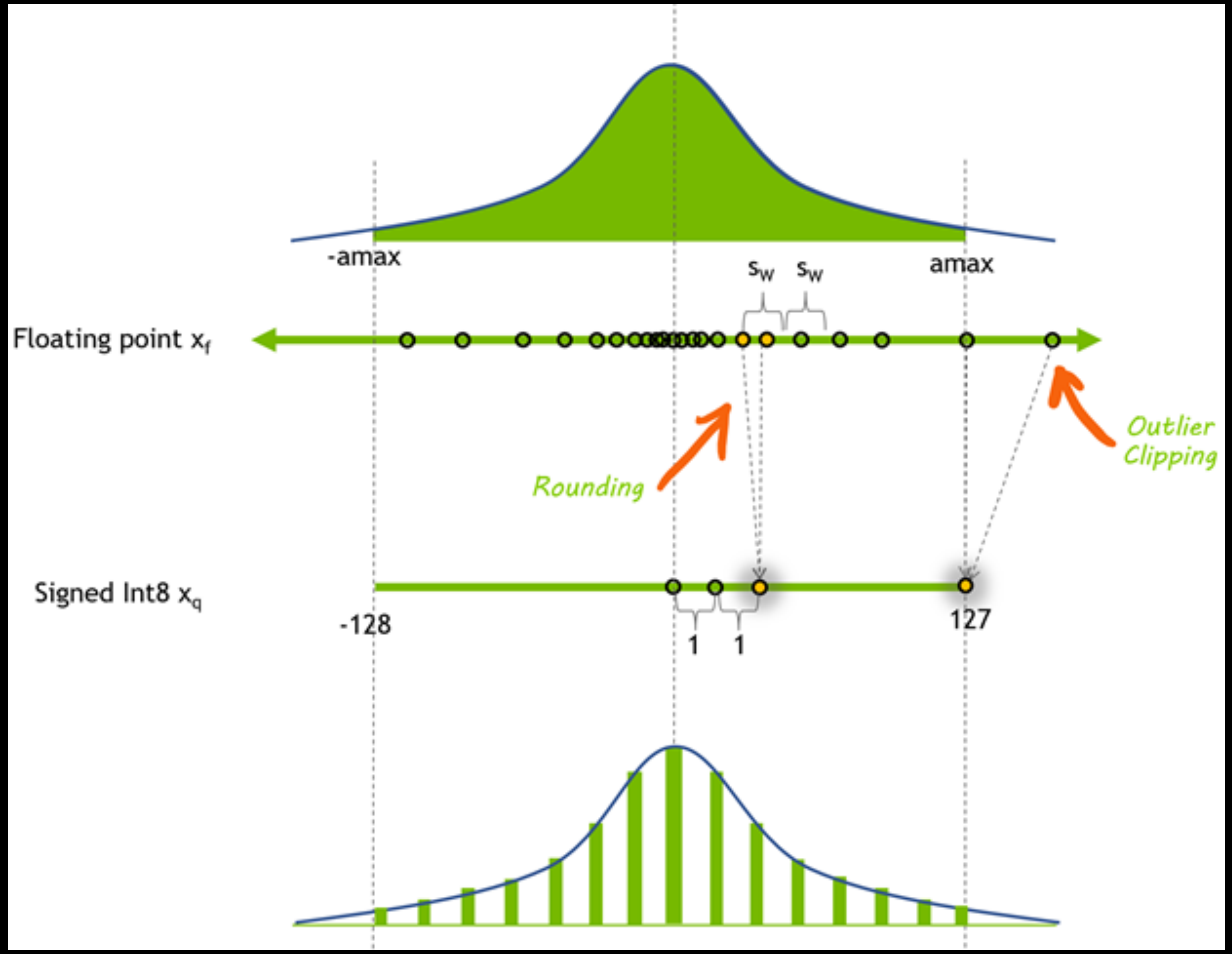


Quantization



Floating point 32:
4B numbers in [-3.4e38, +3.4e38]

Quantization



Quantising:

int8 2^8=256 numbers in [-128,127]

$$x_q = Clip(Round(\frac{x_f}{scale}))$$

Precision	Approx. **Peak GOPS**	**On-chip weights**
1b	**64 000**	**~64 M**
4b	**16 000**	**~16 M**
8b	**4 000**	**~8 M**
32b	**300**	**~2 M**

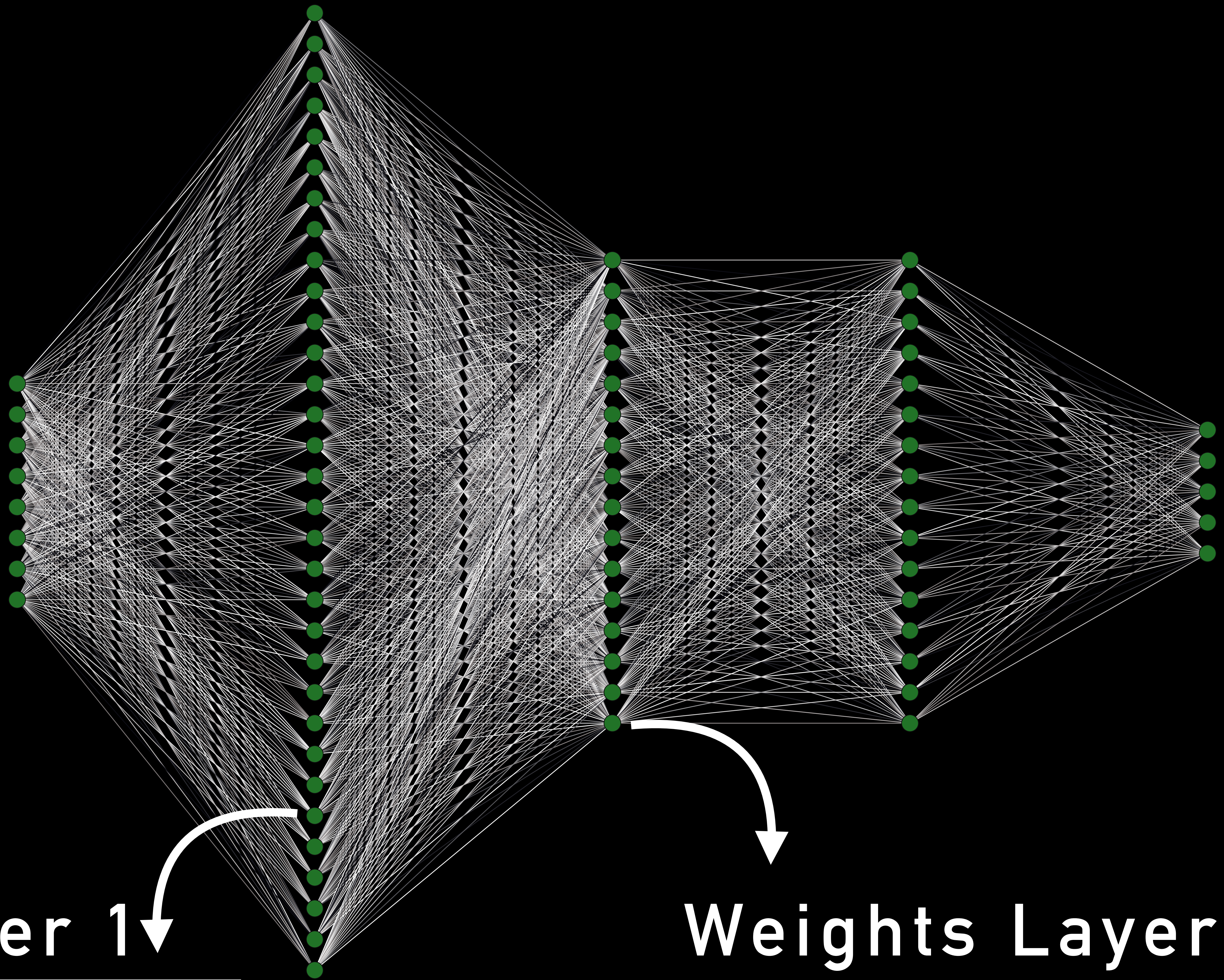
200x

30x

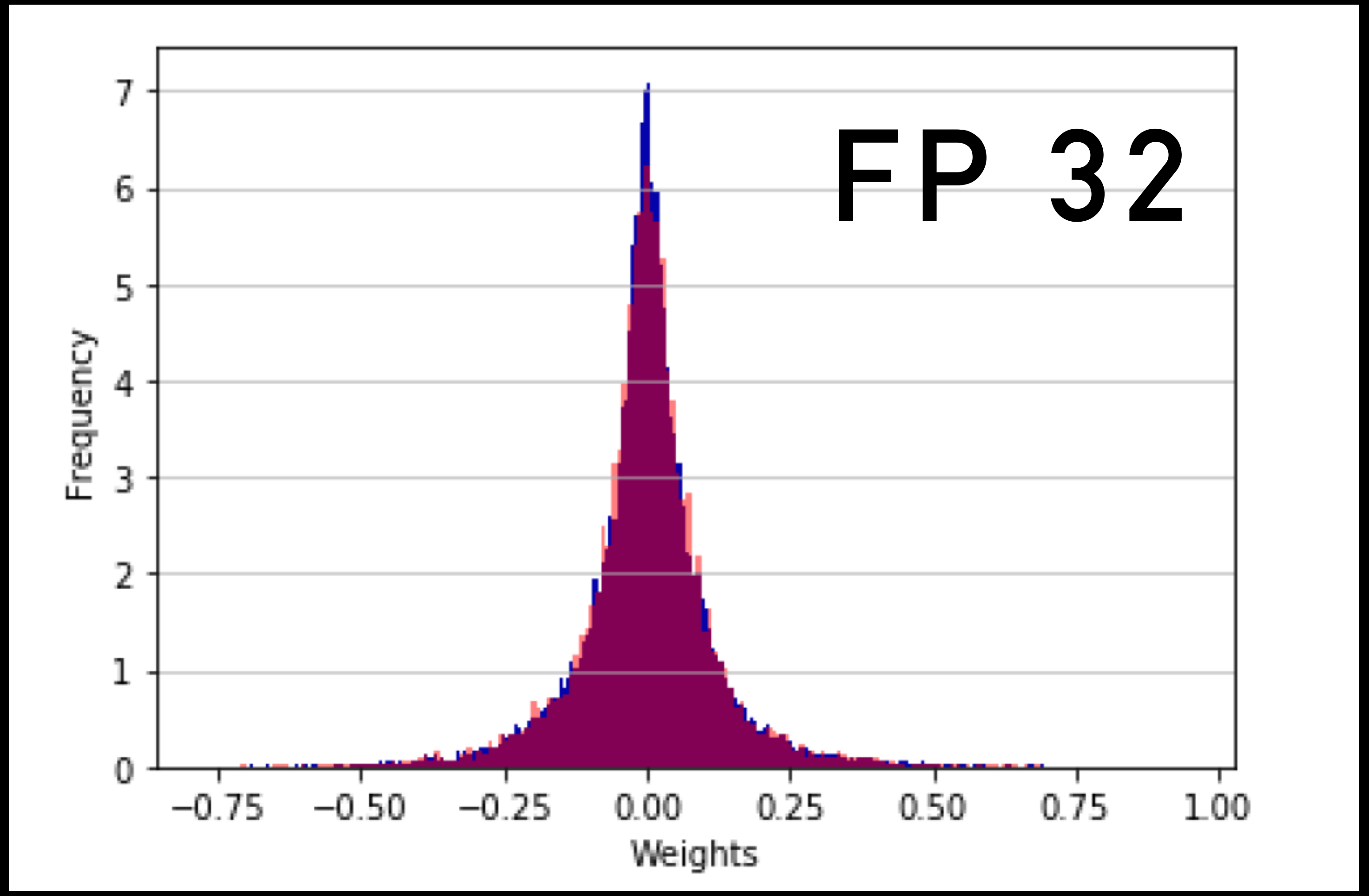
Trillions of quantized operations per second

Weights can stay **entirely on-chip**

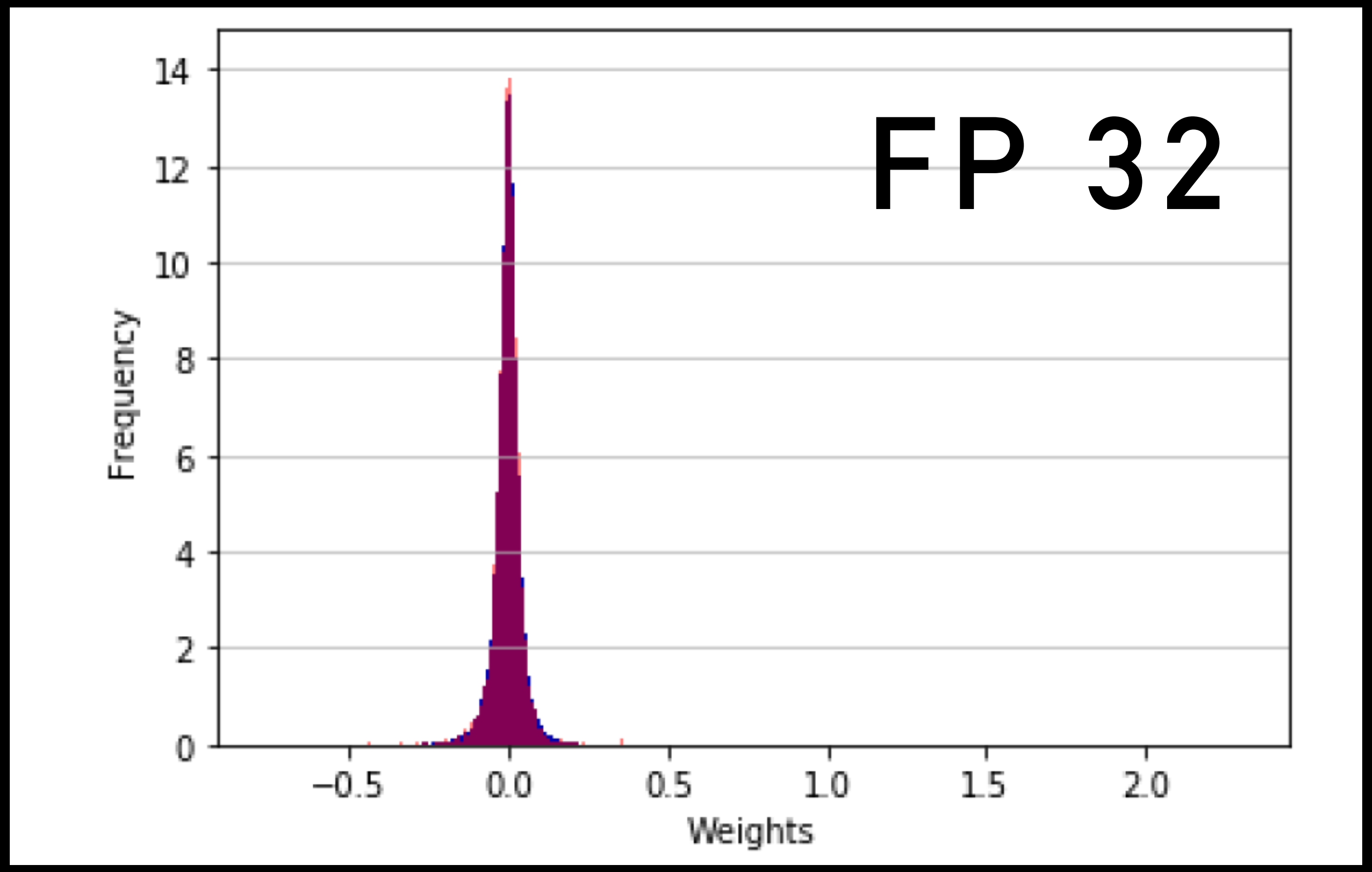
AMD UltraScale+ MPSoC ZU19EG (conservative estimates)



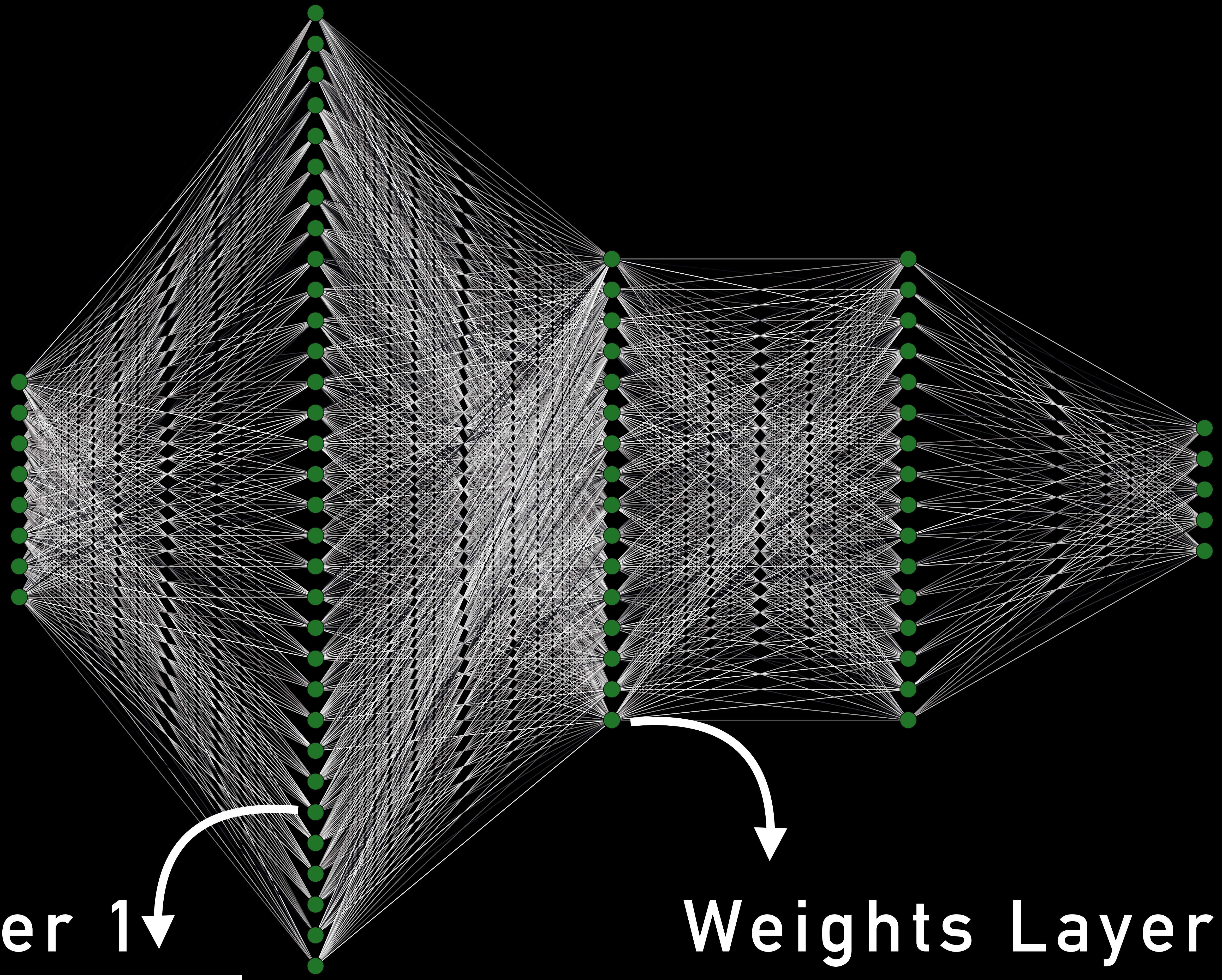
Weights Layer 1



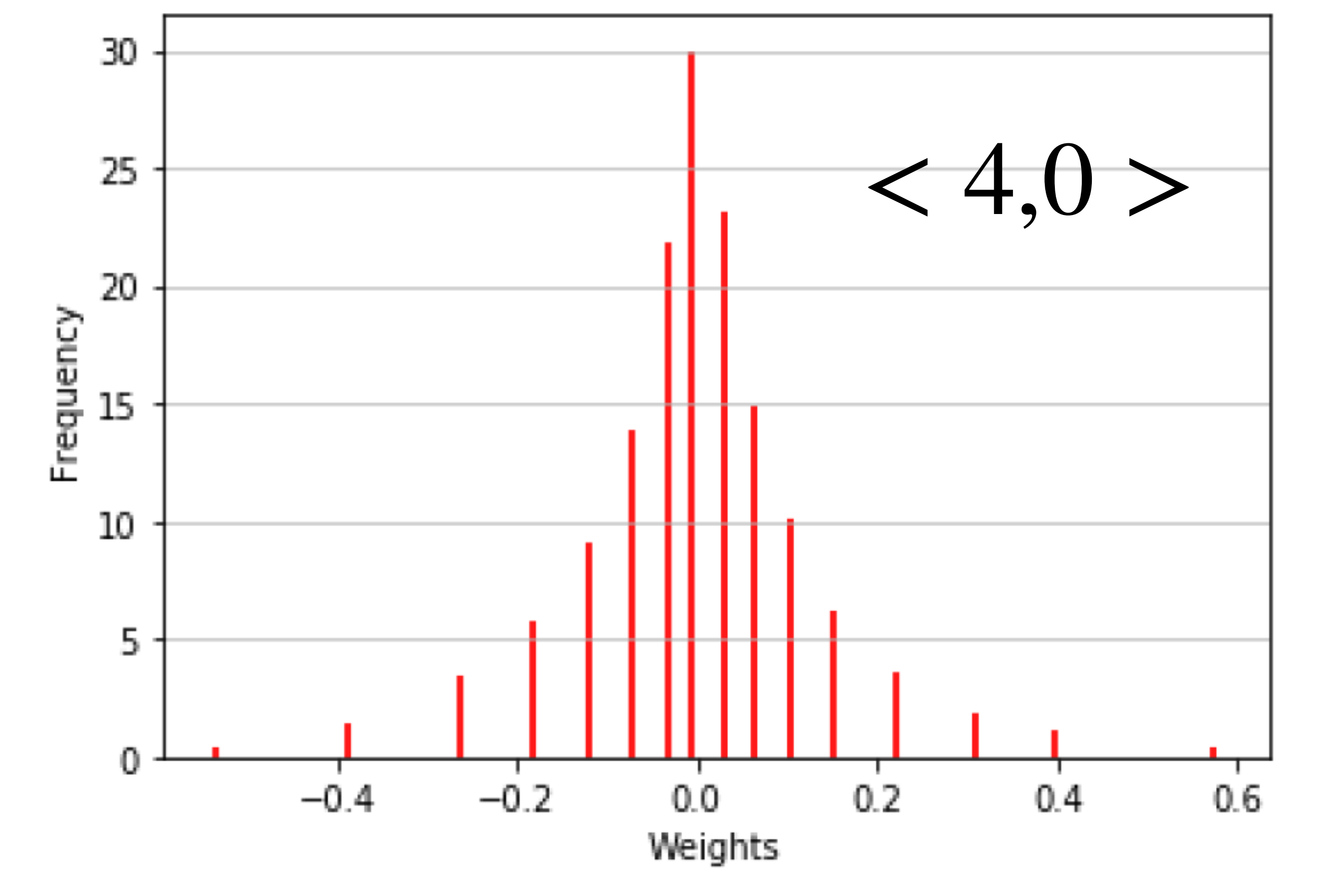
Weights Layer 2



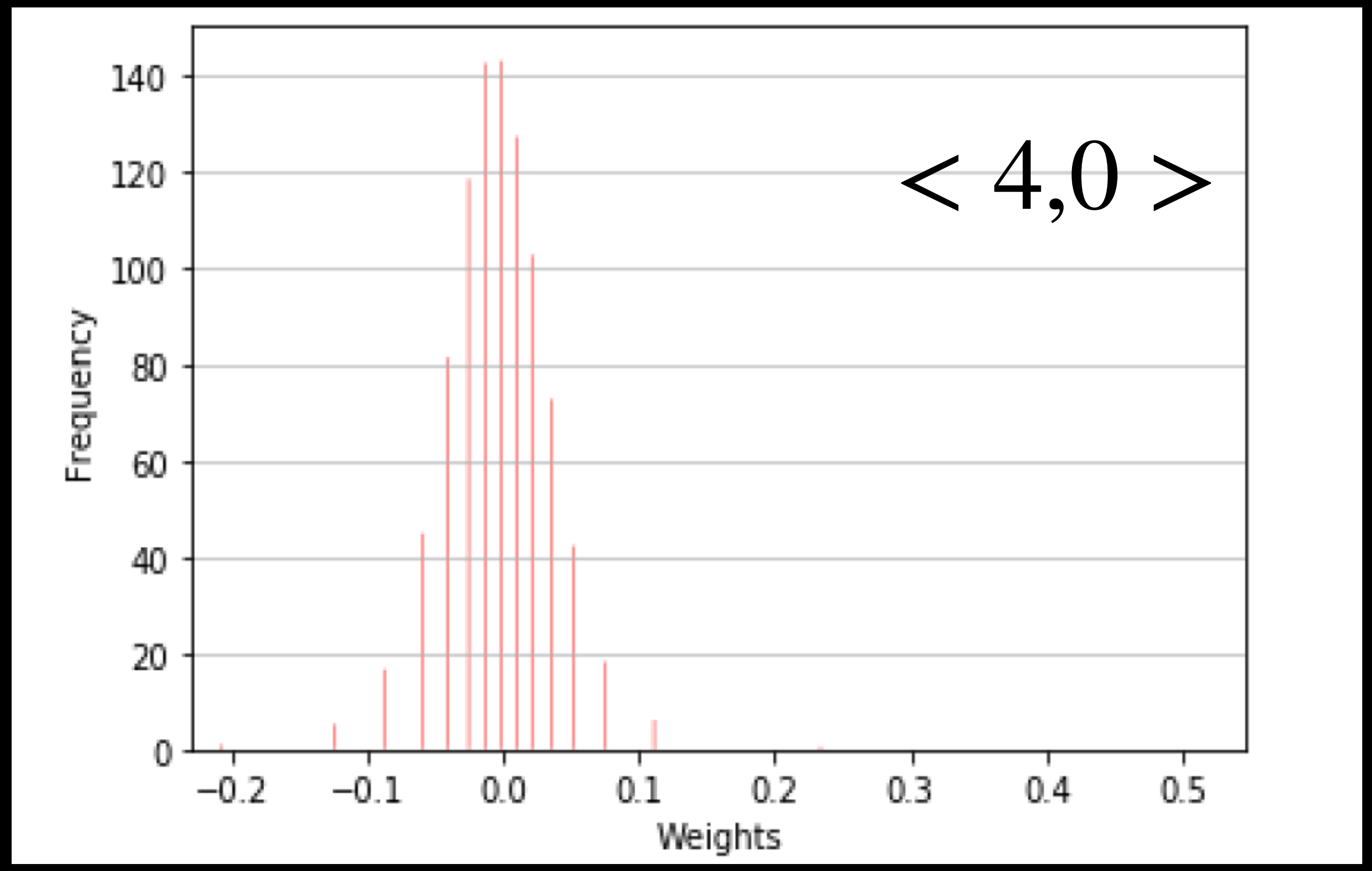
Fixed point



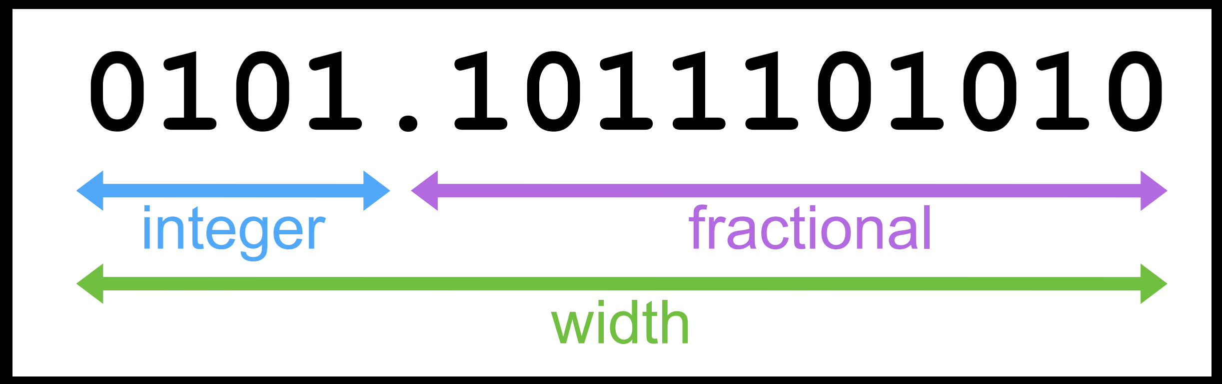
Weights Layer 1

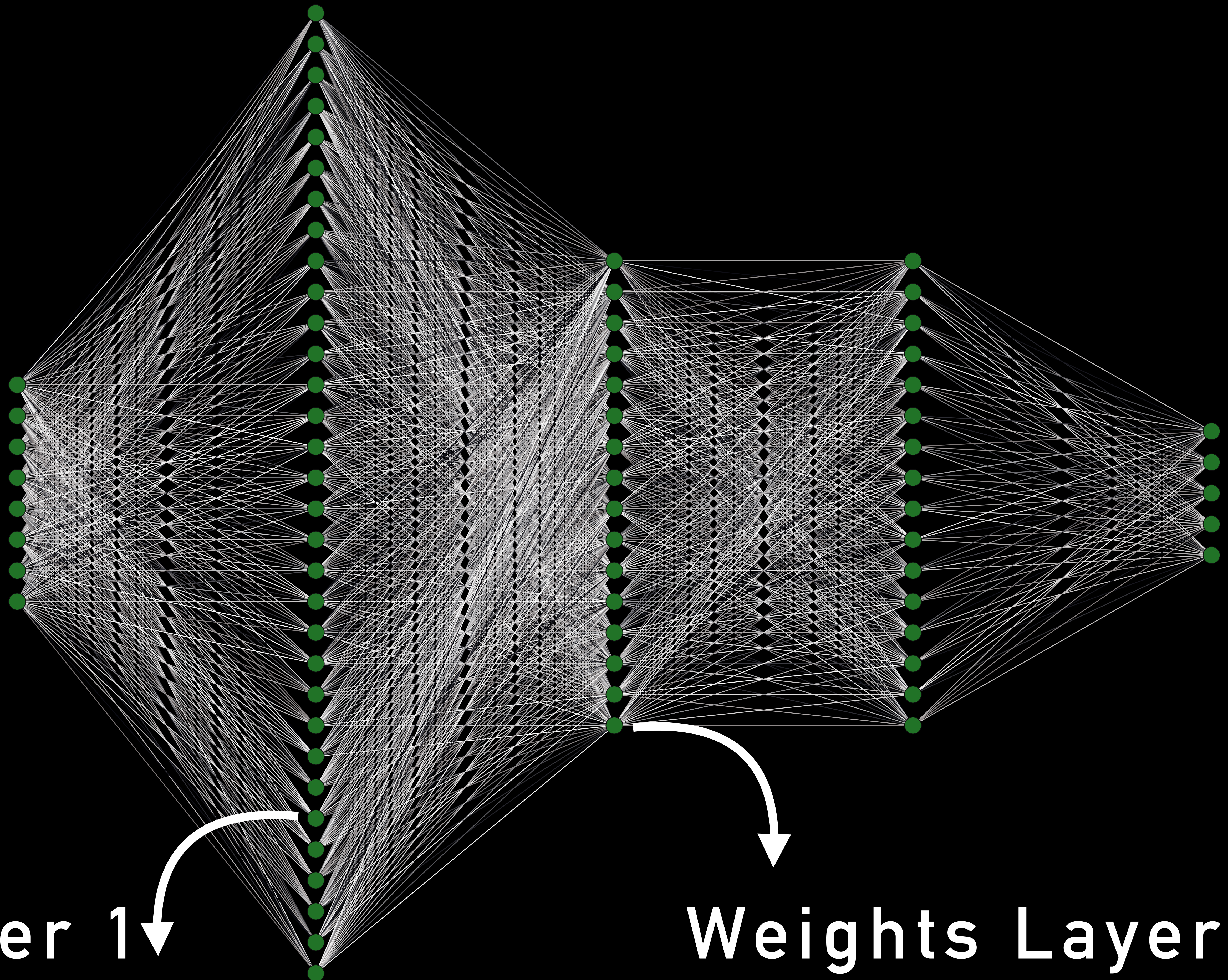


Weights Layer 2

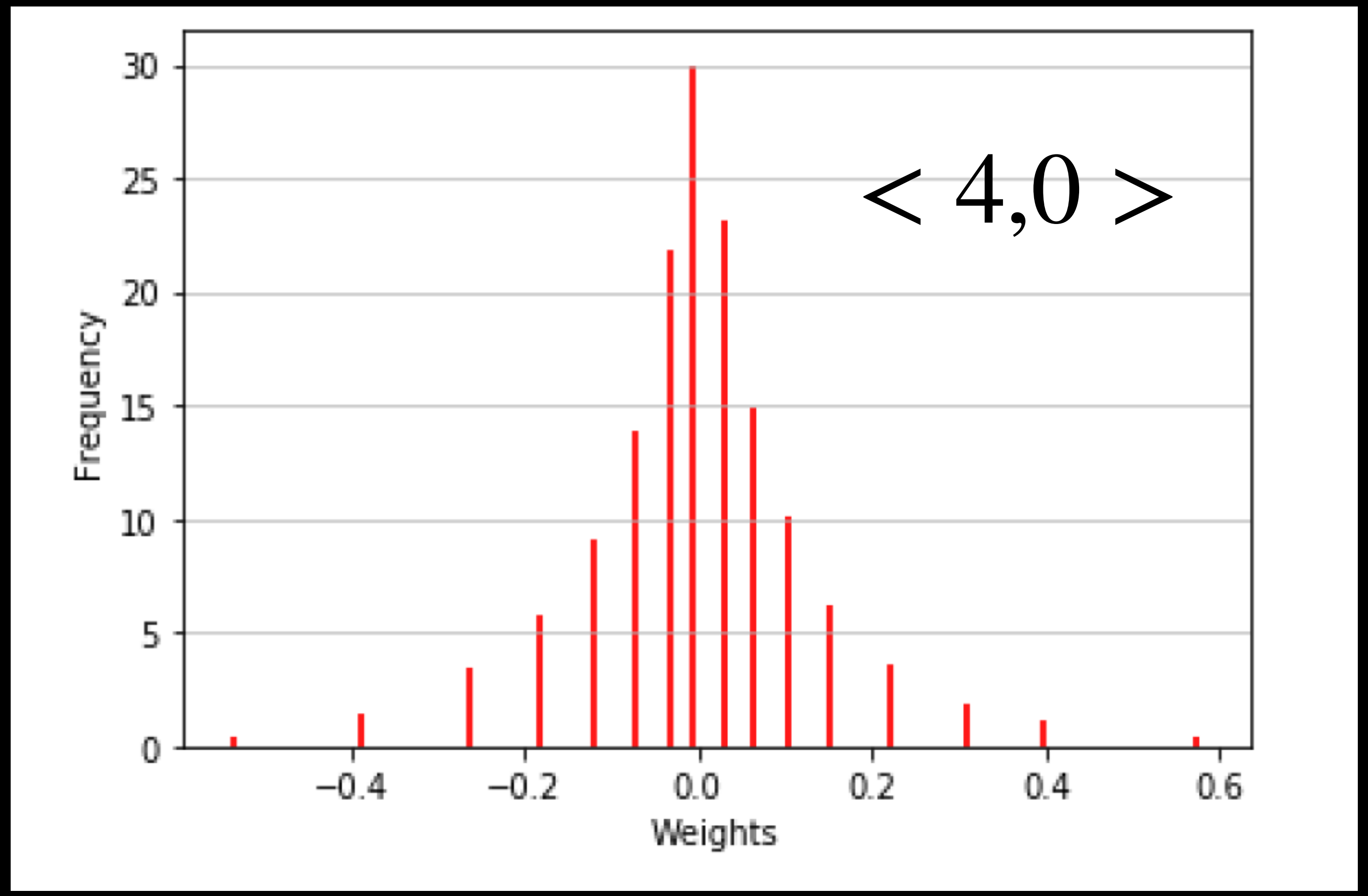


Fixed point

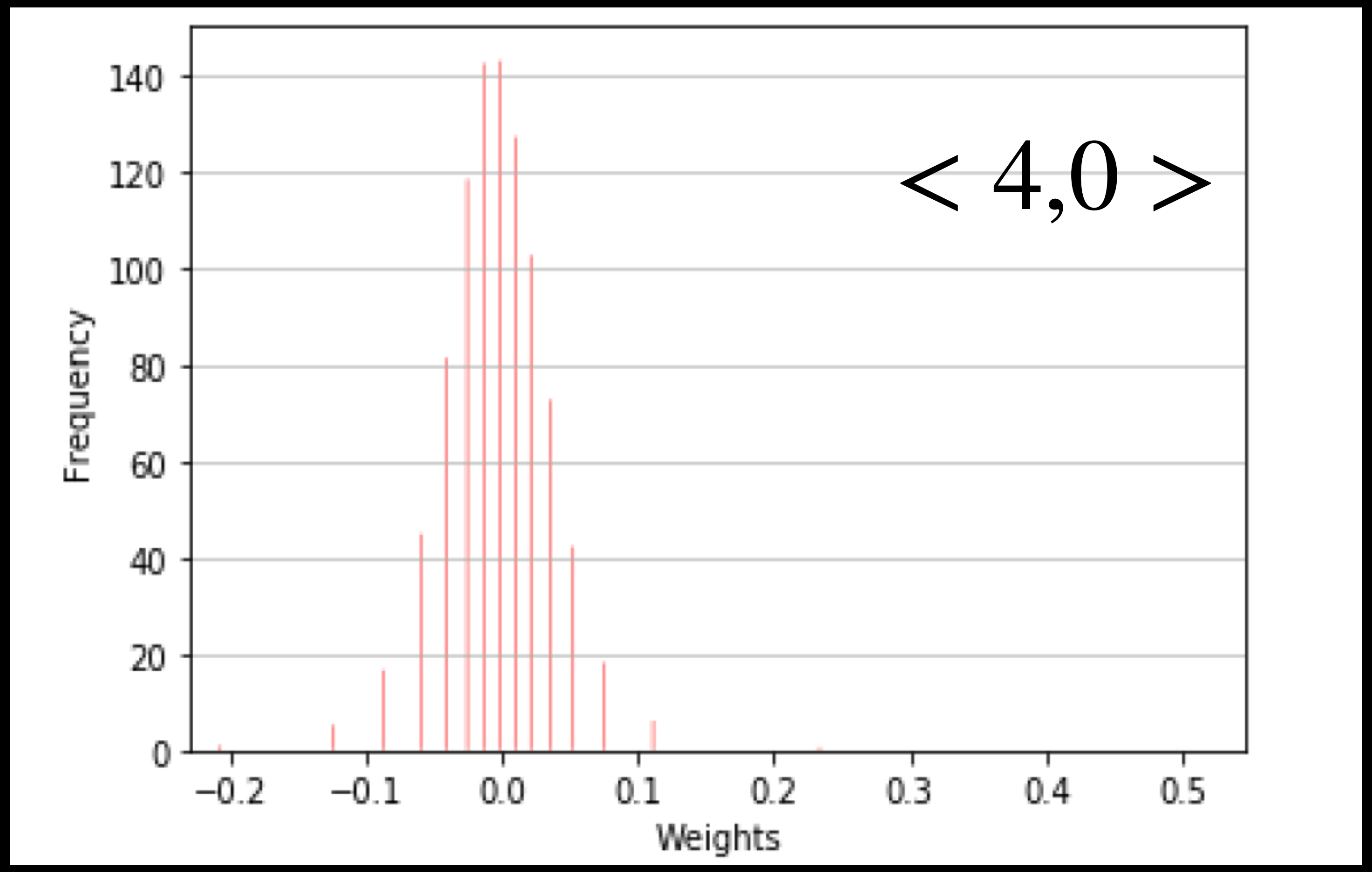


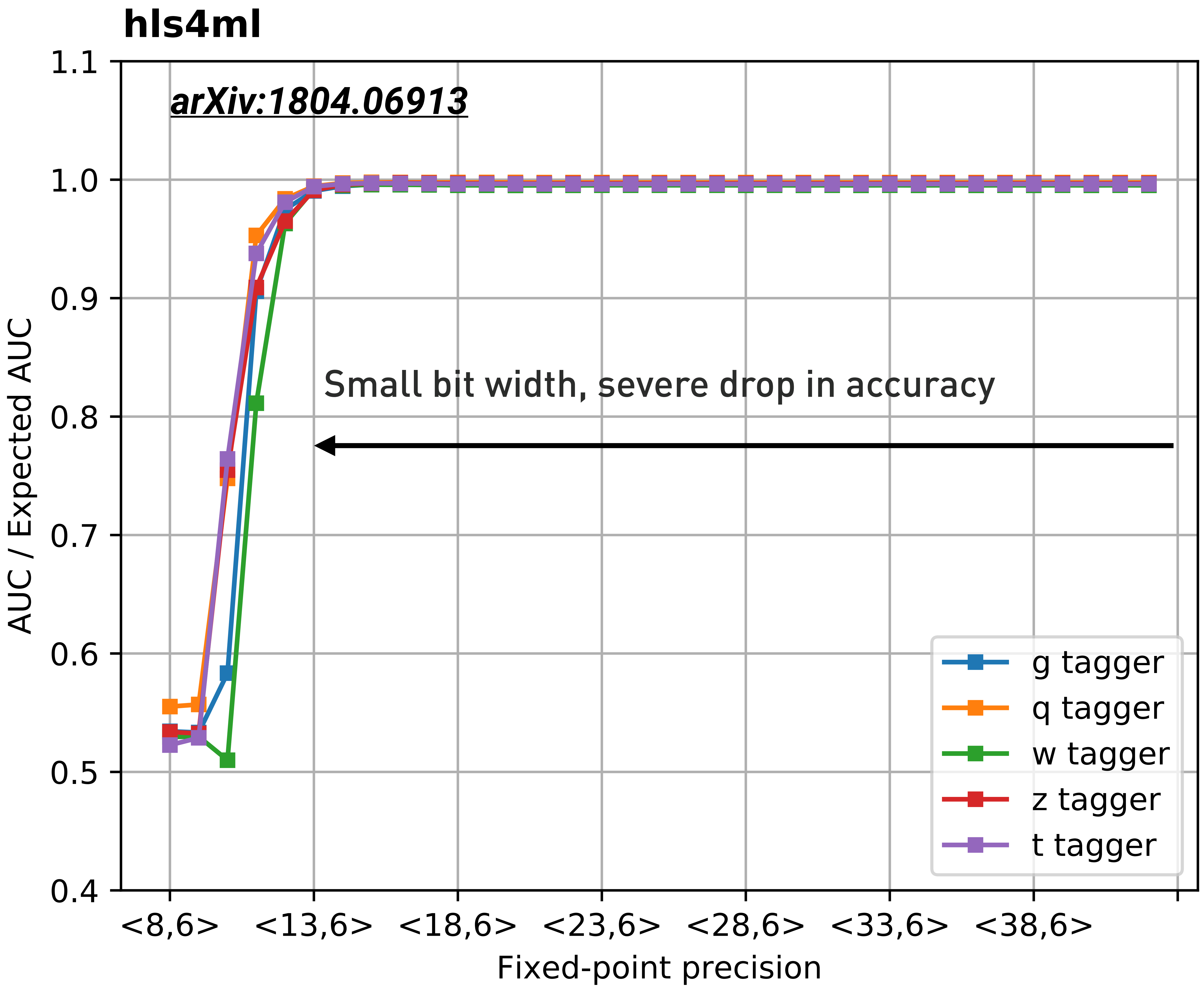


Weights Layer 1



Weights Layer 2

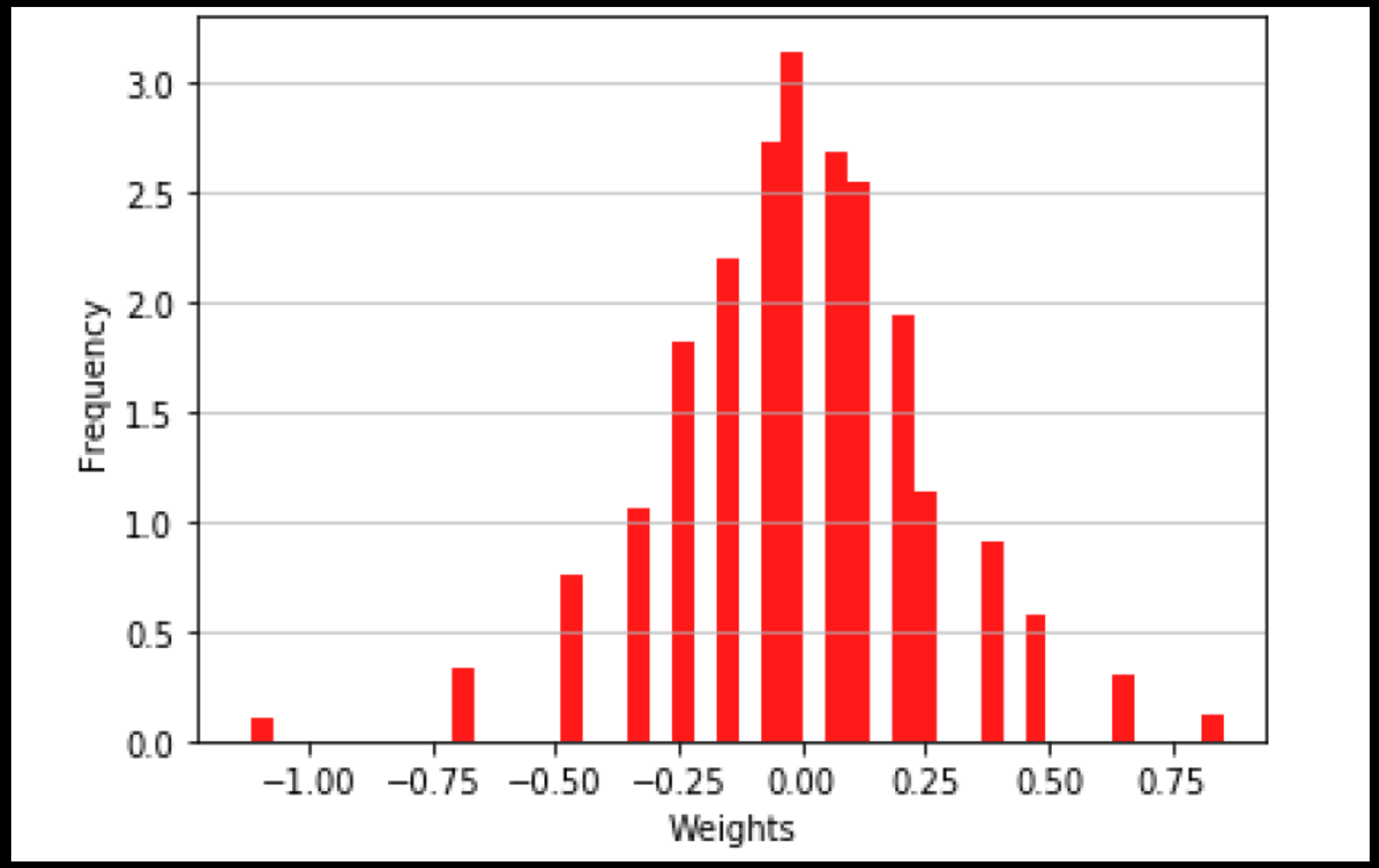


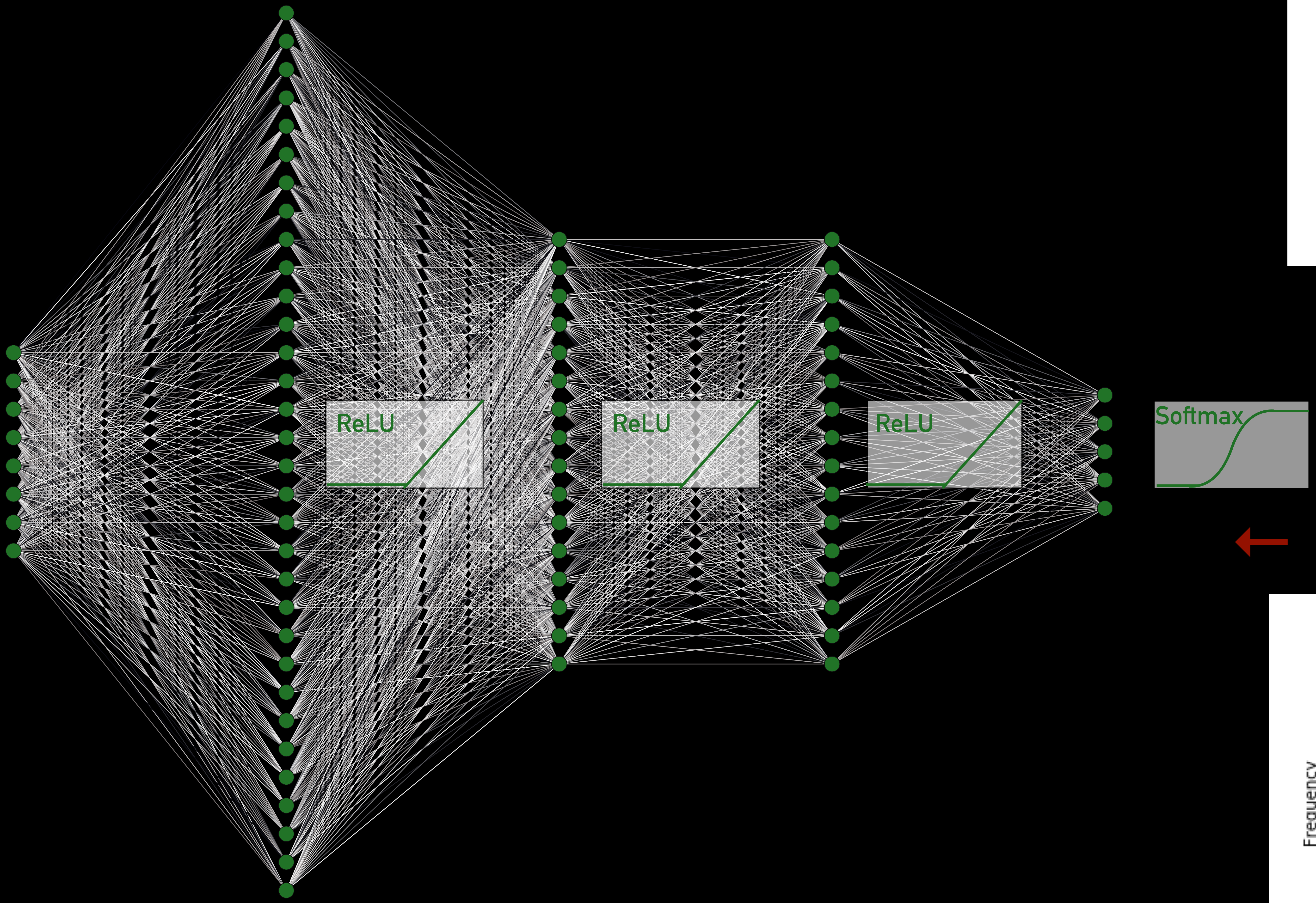

hls4ml
arXiv:1804.06913
Small bit width, severe drop in accuracy
AUC / Expected AUC
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
<8,6>
<13,6>
<18,6>
<23,6>
<28,6>
<33,6>
<38,6>
Fixed-point precision
g tagger
q tagger
w tagger
z tagger
t tagger

hls4ml + Google

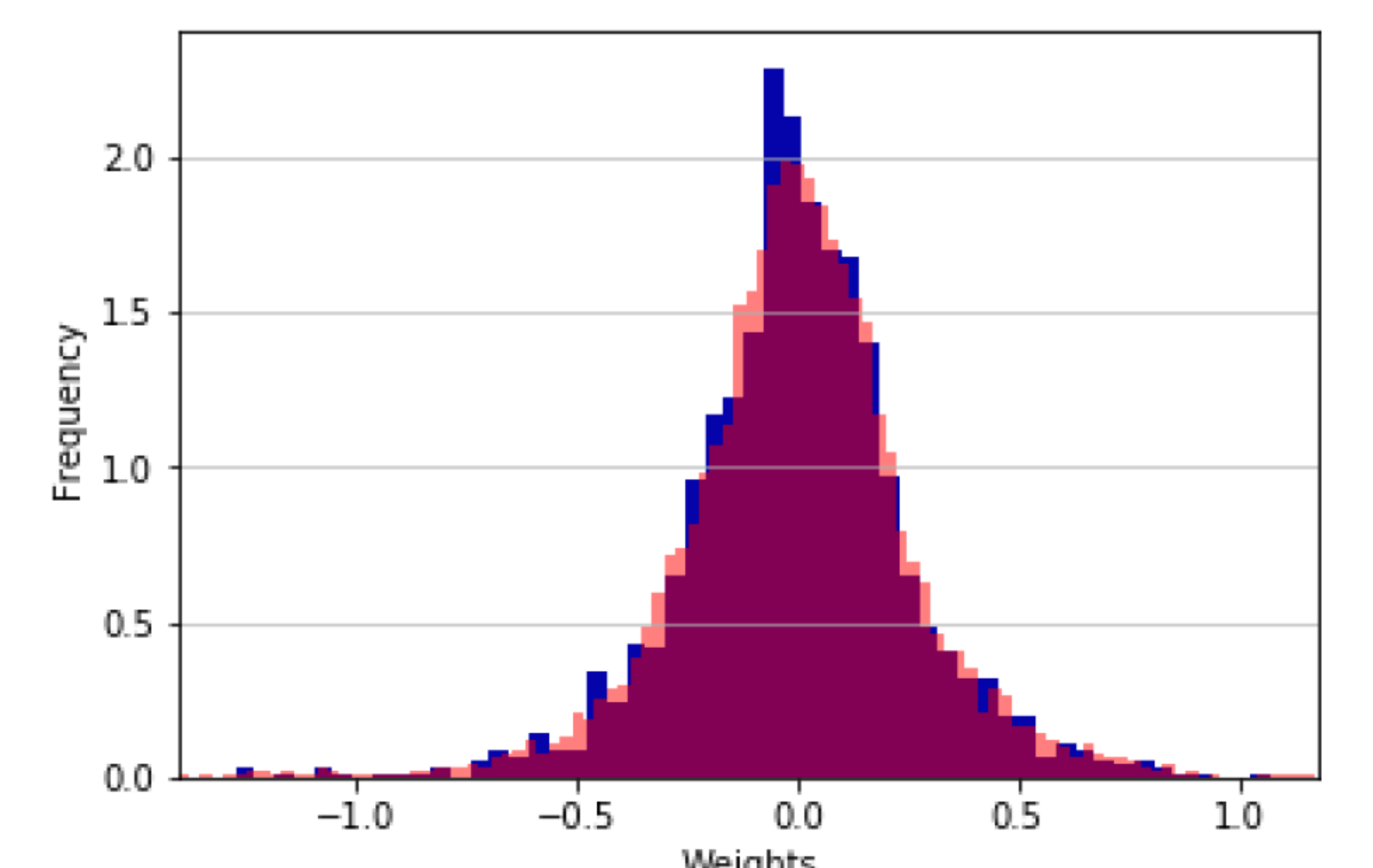
Quantization-aware training

Forward pass →





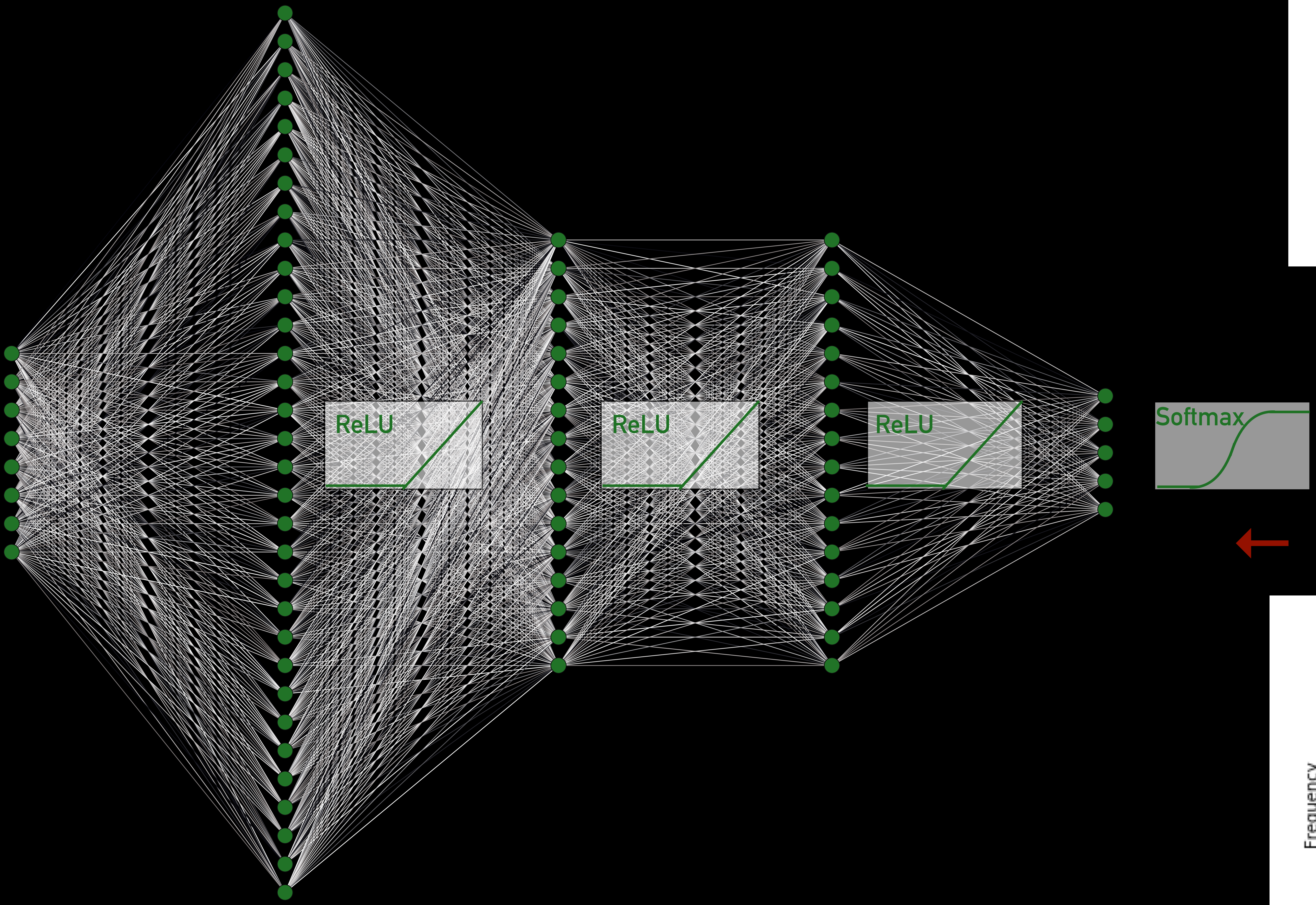
← Back propagation



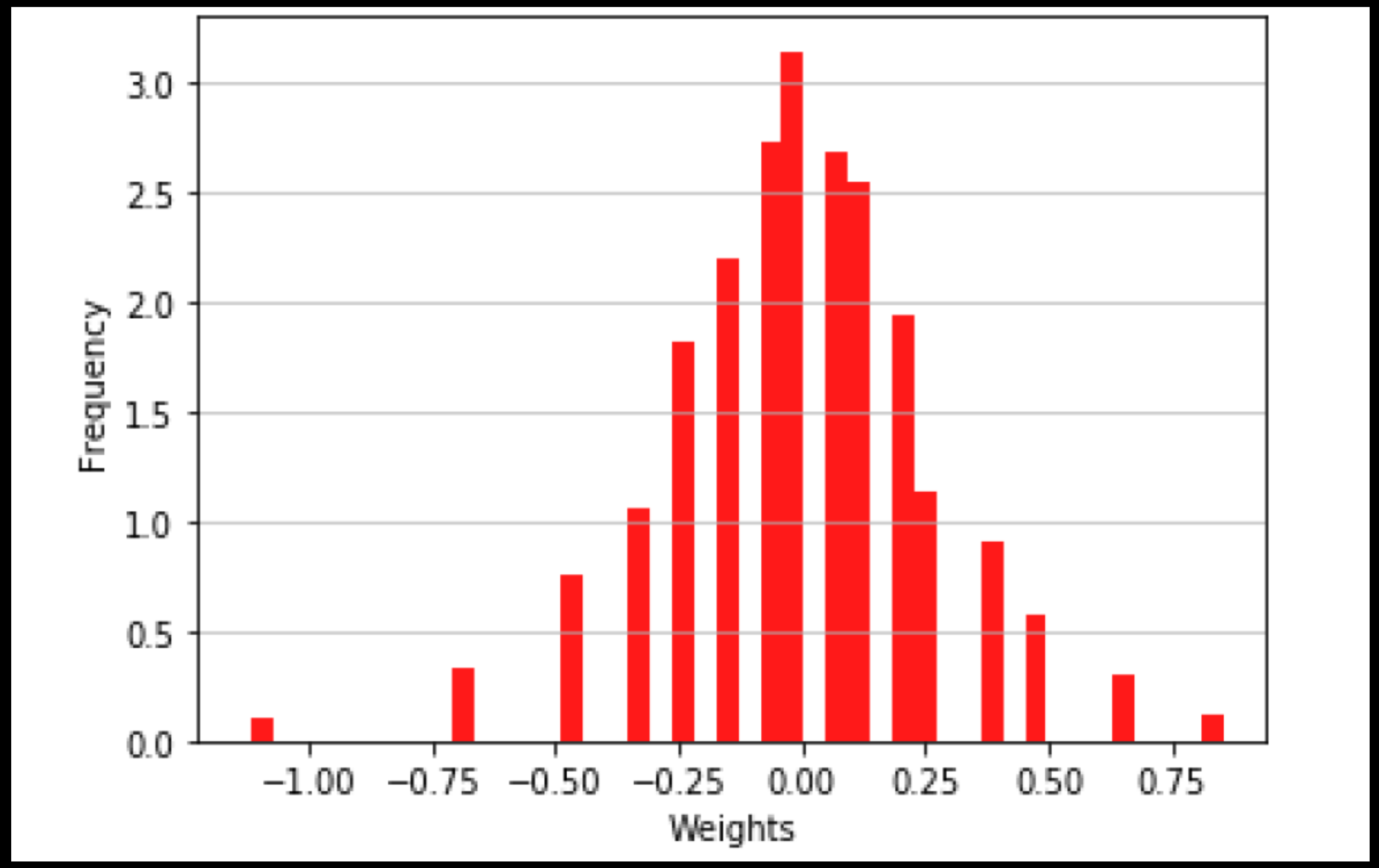
Nature Machine Intelligence 3 (2021)

hls4ml + Google

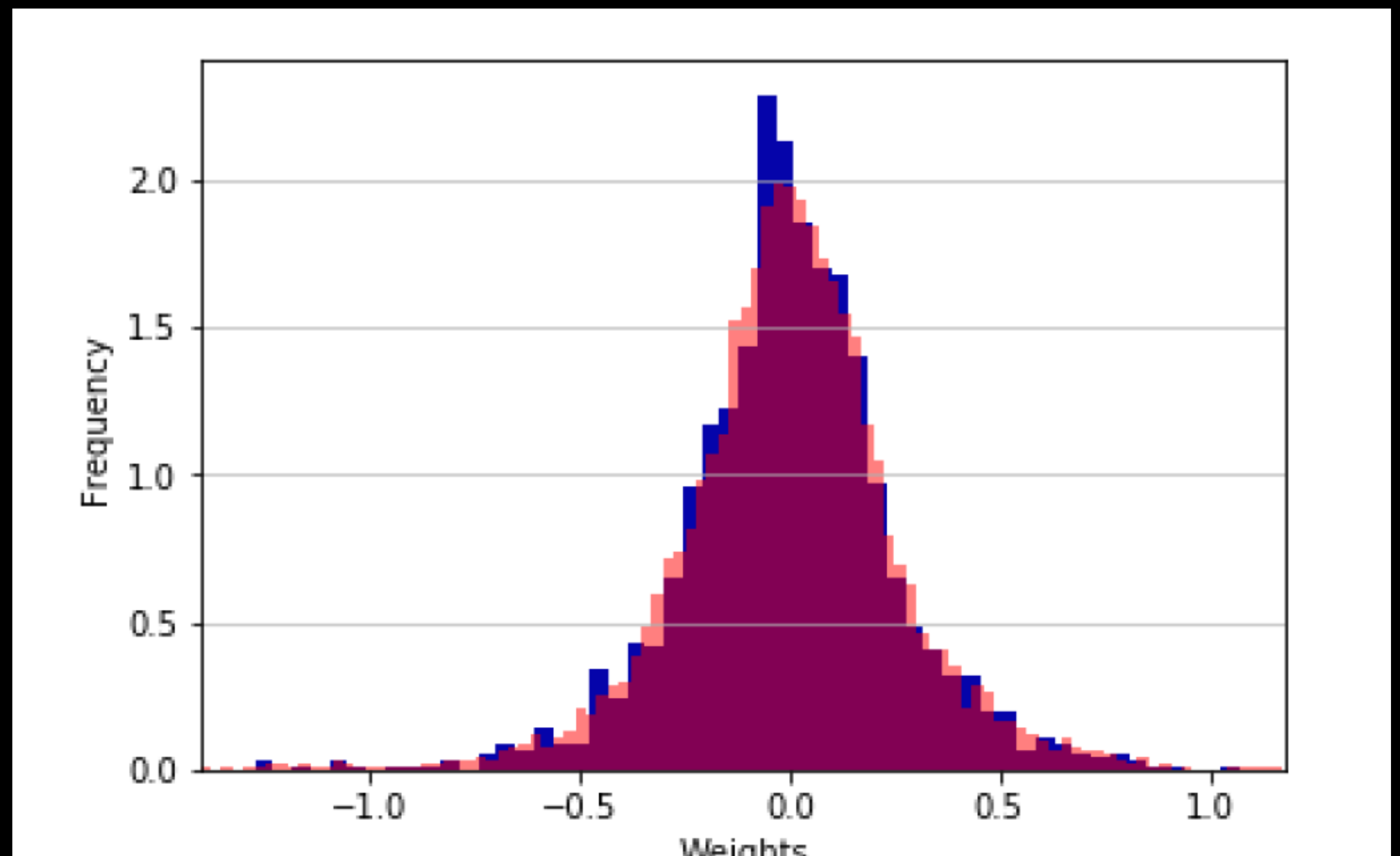
Quantization-aware training



Forward pass →



← Back propagation

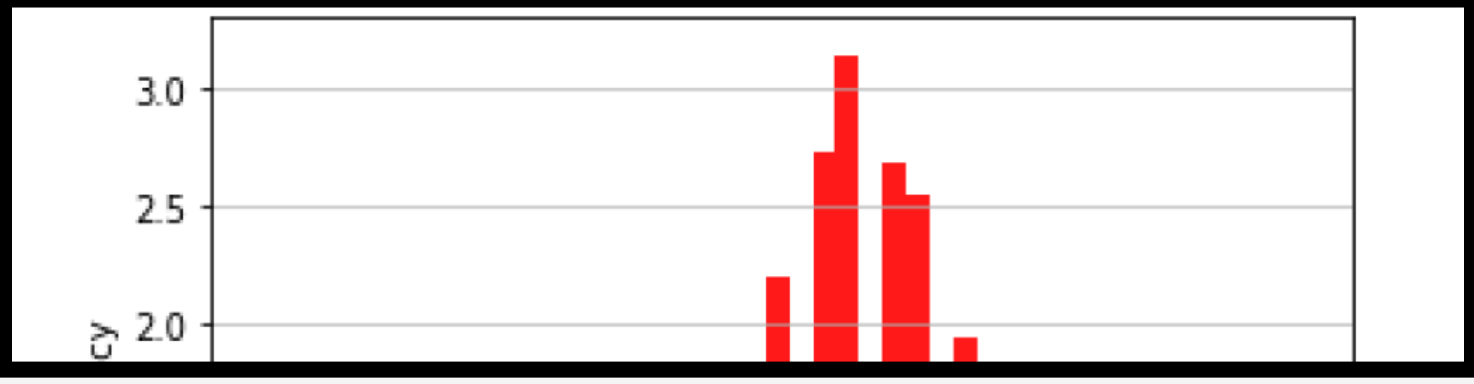


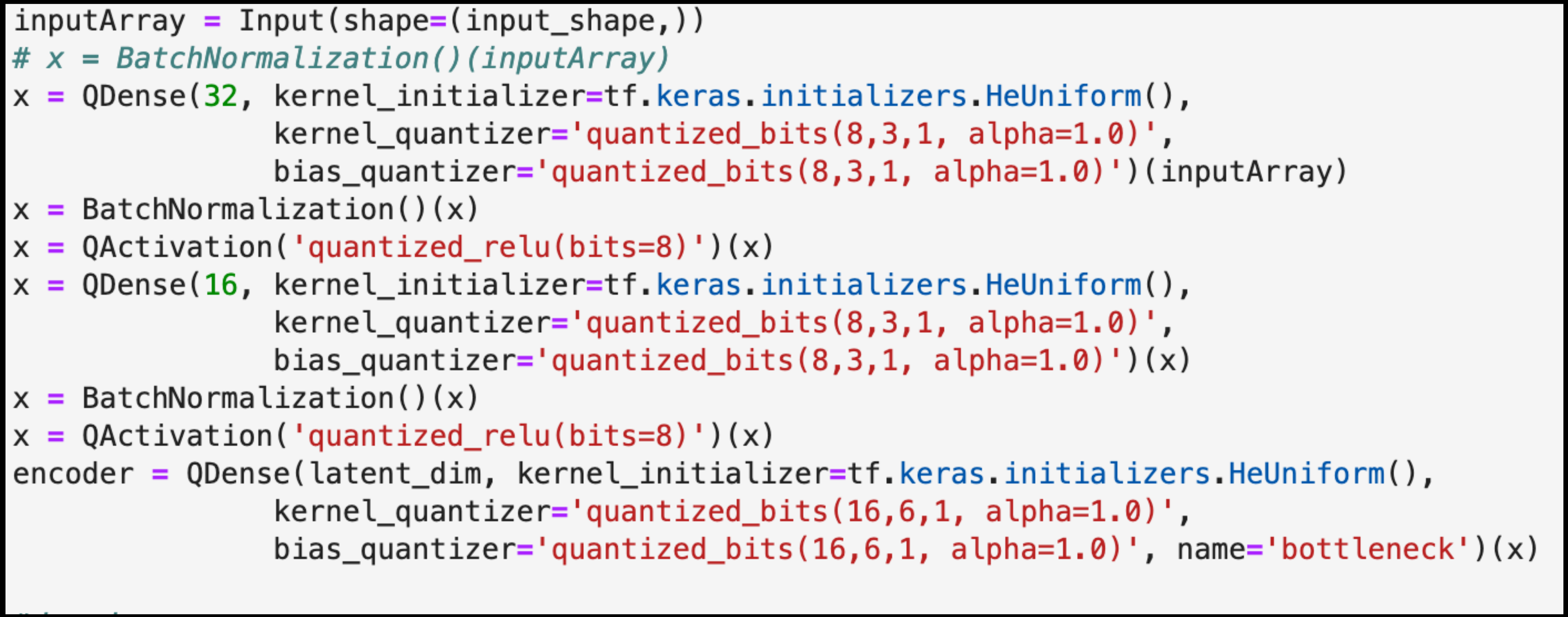
Nature Machine Intelligence 3 (2021)

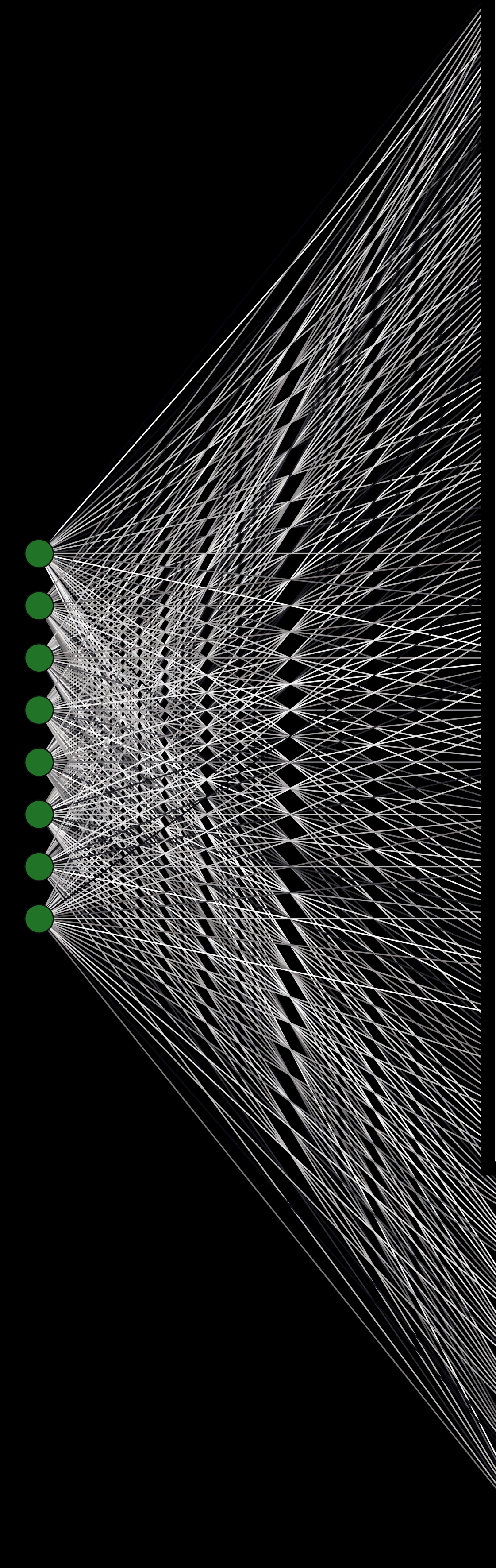
hls4ml + Google

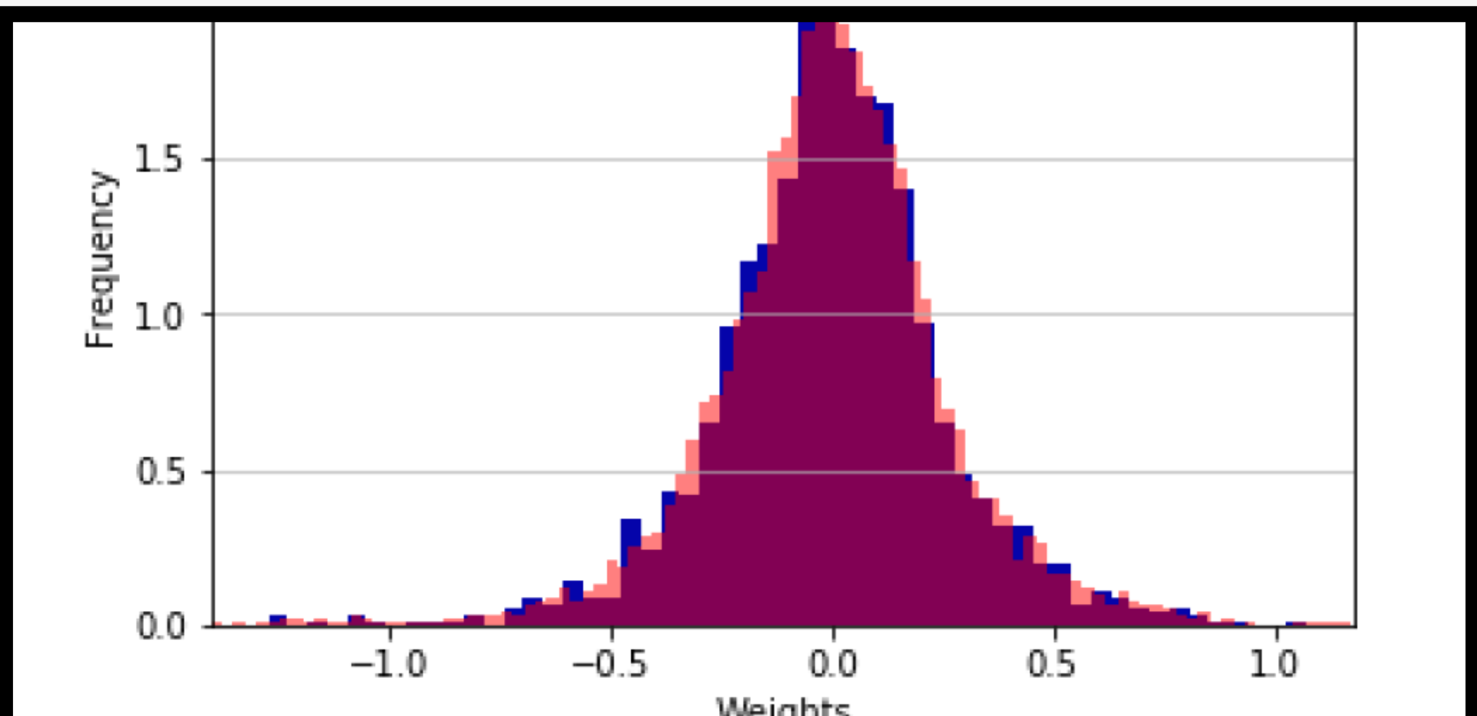
Quantization-aware training

Forward pass →




```
inputArray = Input(shape=(input_shape,))
# x = BatchNormalization()(inputArray)
x = QDense(32, kernel_initializer=tf.keras.initializers.HeUniform(),
               kernel_quantizer='quantized_bits(8,3,1, alpha=1.0)',
               bias_quantizer='quantized_bits(8,3,1, alpha=1.0)')(inputArray)
x = BatchNormalization()(x)
x = QActivation('quantized_relu(bits=8)')(x)
x = QDense(16, kernel_initializer=tf.keras.initializers.HeUniform(),
               kernel_quantizer='quantized_bits(8,3,1, alpha=1.0)',
               bias_quantizer='quantized_bits(8,3,1, alpha=1.0)')(x)
x = BatchNormalization()(x)
x = QActivation('quantized_relu(bits=8)')(x)
encoder = QDense(latent_dim, kernel_initializer=tf.keras.initializers.HeUniform(),
               kernel_quantizer='quantized_bits(16,6,1, alpha=1.0)',
               bias_quantizer='quantized_bits(16,6,1, alpha=1.0)', name='bottleneck')(x)
```





Nature Machine Intelligence 3 (2021)

Estimating energy and size

Some layers more accommodating for aggressive quantization, others require expensive arithmetic

- heterogeneous quantization

Estimating energy and size

Some layers more accommodating for aggressive quantization, others require expensive arithmetic

- heterogeneous quantization

For edge inference, need best possible quantization configuration for

- Highest accuracy ↑...
- ... and lowest resource consumption ↓

→ hyper-parameter scan over quantizers which considers energy and accuracy simultaneously

Estimating energy and size

Some layers more accommodating for aggressive quantization, others require expensive arithmetic

- heterogeneous quantization

For edge inference, need best possible quantization configuration for

- Highest accuracy ↑...
- ... and lowest resource consumption ↓

→ hyper-parameter scan over quantizers which considers energy and accuracy simultaneously

QTools: Estimate QKeras model bit and energy consumption, assuming 45 nm Horowitz process

Model	Accuracy [%]	Per-layer energy consumption [pJ]								Total energy [μJ]	Total bits
		Dense	ReLU	Dense	ReLU	Dense	ReLU	Dense	Softmax		
BF	74.4	1735	53	3240	27	1630	27	281	11	0.00700	61446
Q6	74.8	794	23	1120	11	562	11	99	11	0.00263	26334

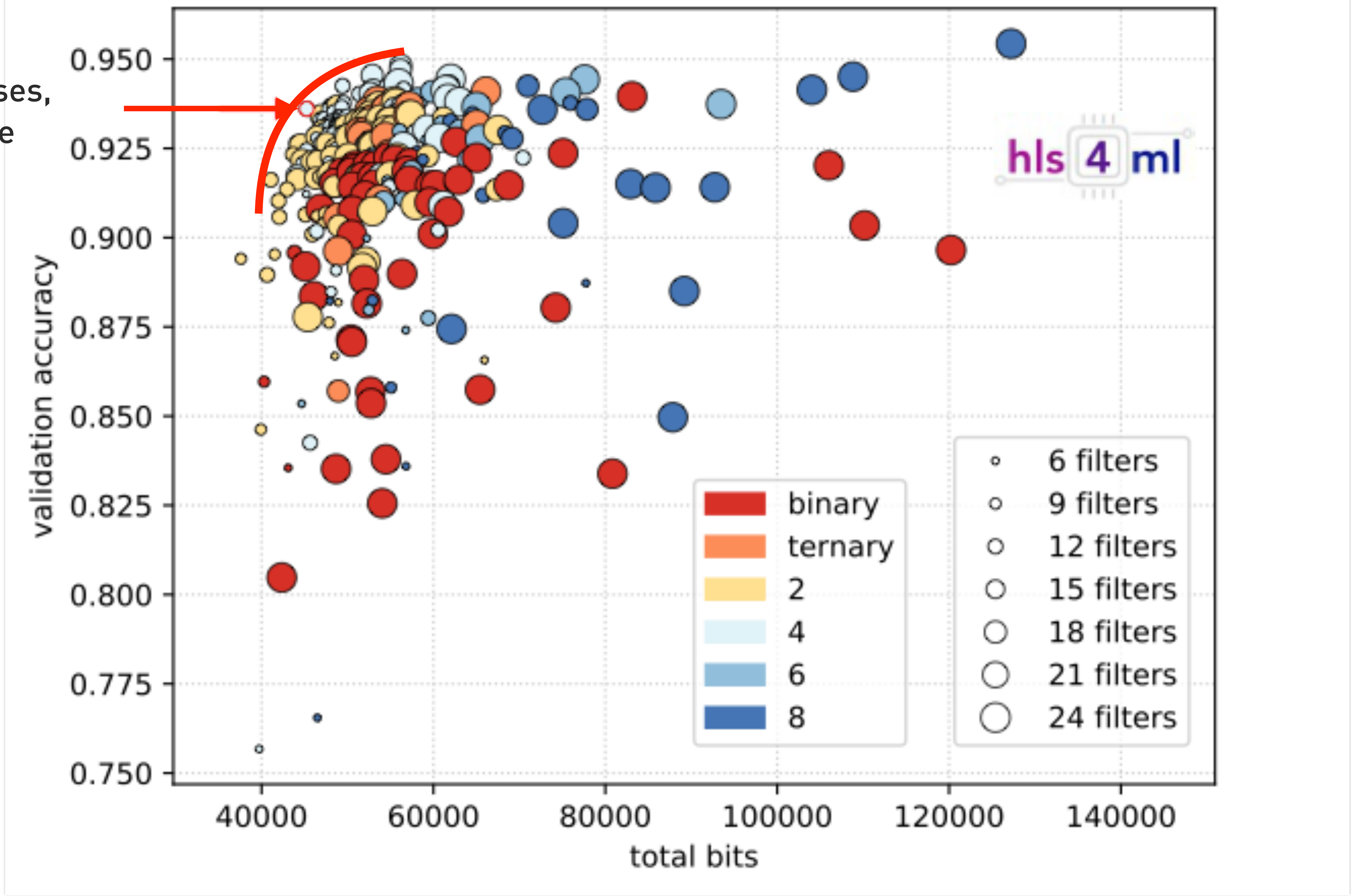
Forgiving Factor $= 1 + \Delta_{accuracy} \times \log_{rate}(S \times \frac{Cost_{ref}}{Cost_{trial}})$

Maximize accuracy + minimizing cost in hyper parameter scan over quantizers: AutoQKeras

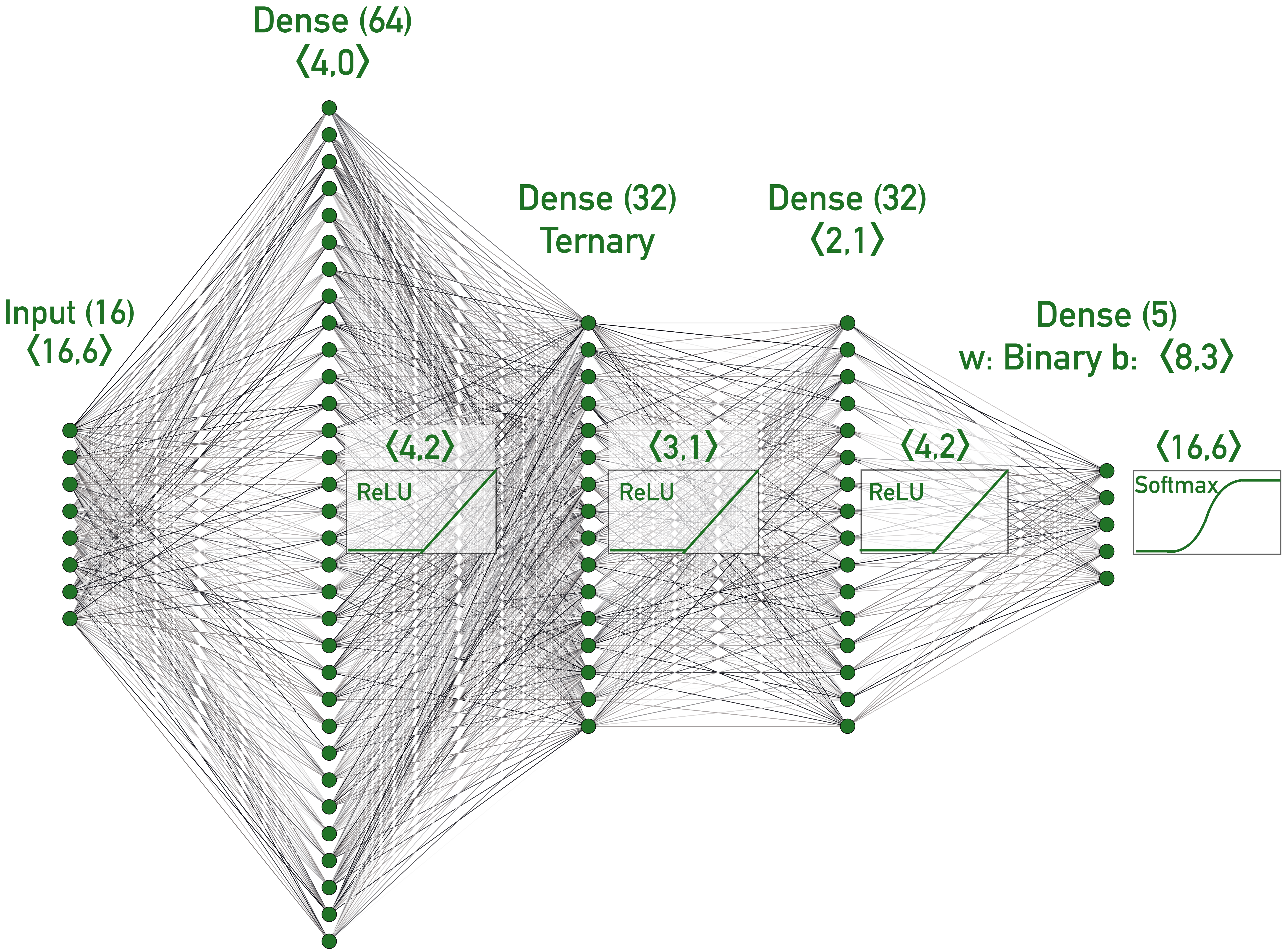
DOI 10.1088/2632-2153/ac0ea1

Example: One convolutional layer

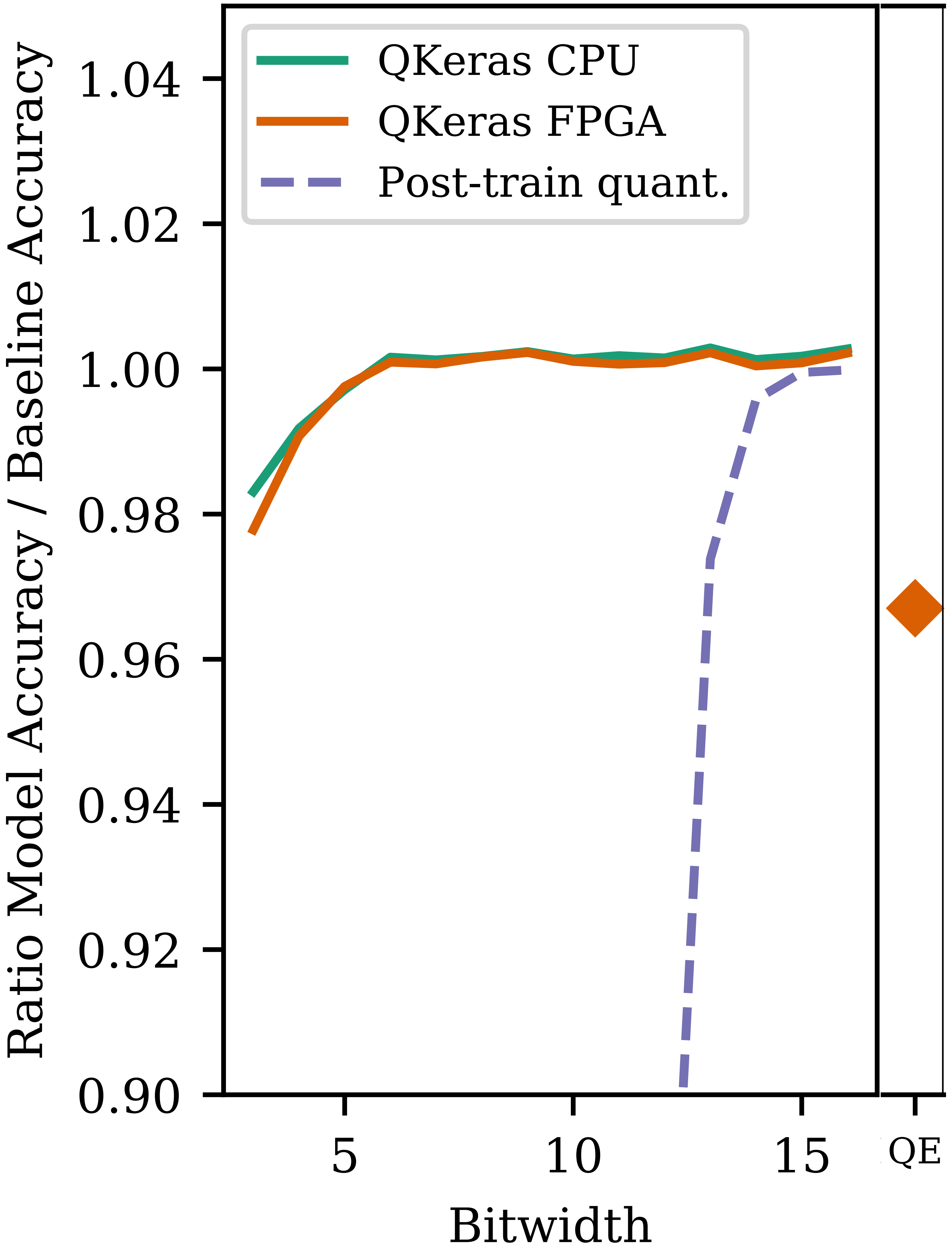
As optimization progresses, best model accuracy/size trade-off is found!

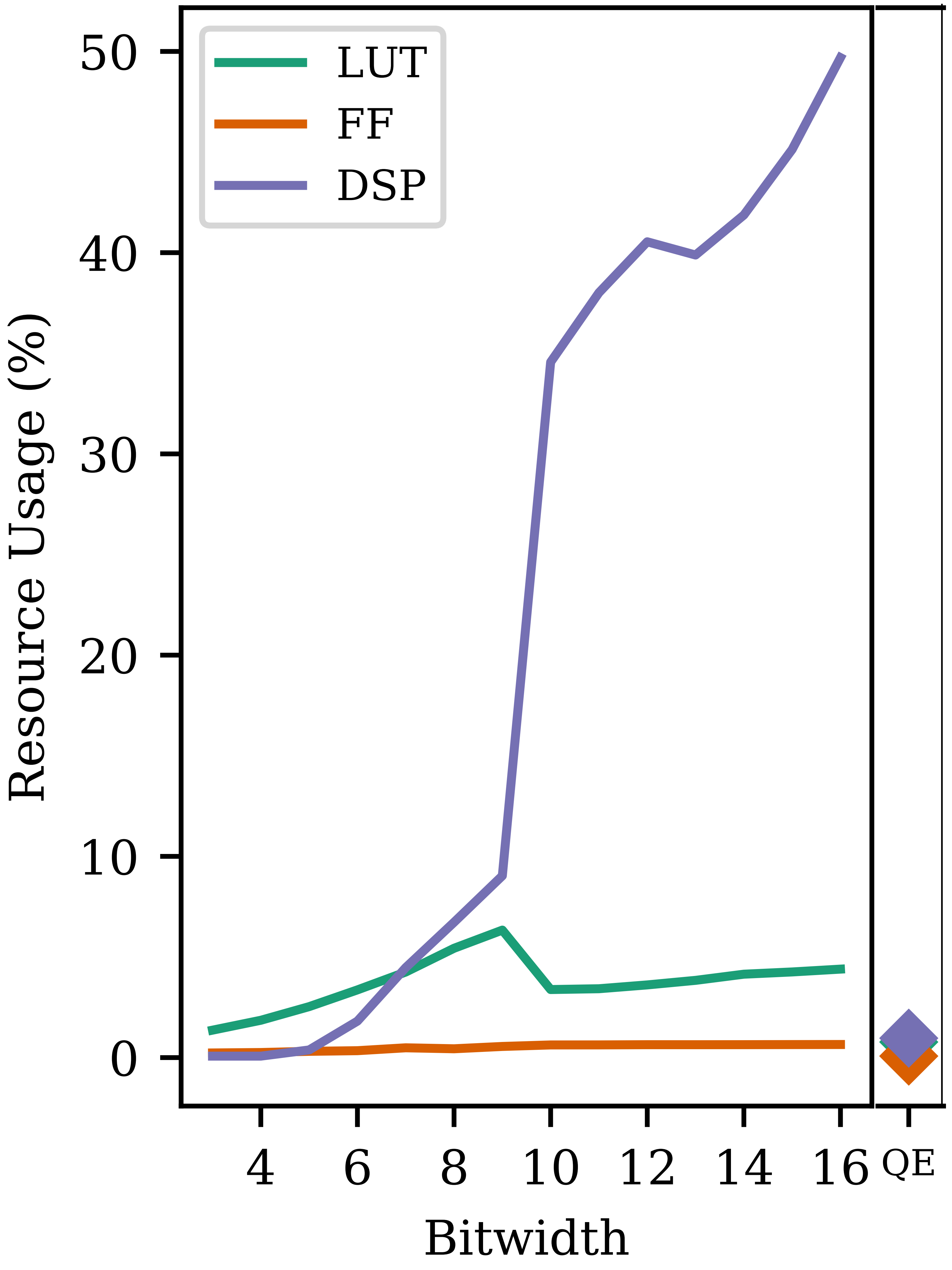


Nature Machine Intelligence 3 (2021)



Multiplications move to LUTs at bit width <10.



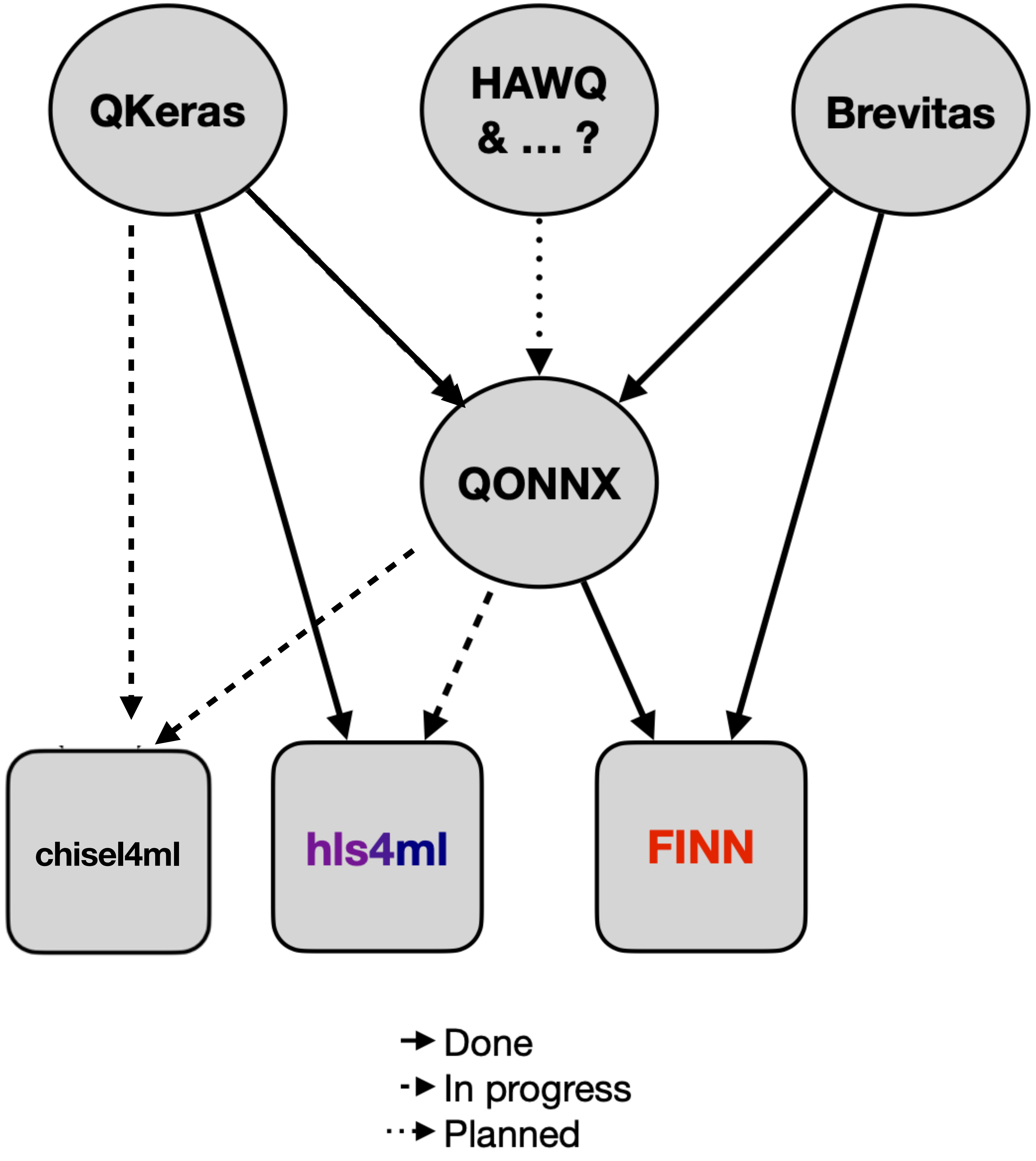


Brevitas and QONNX

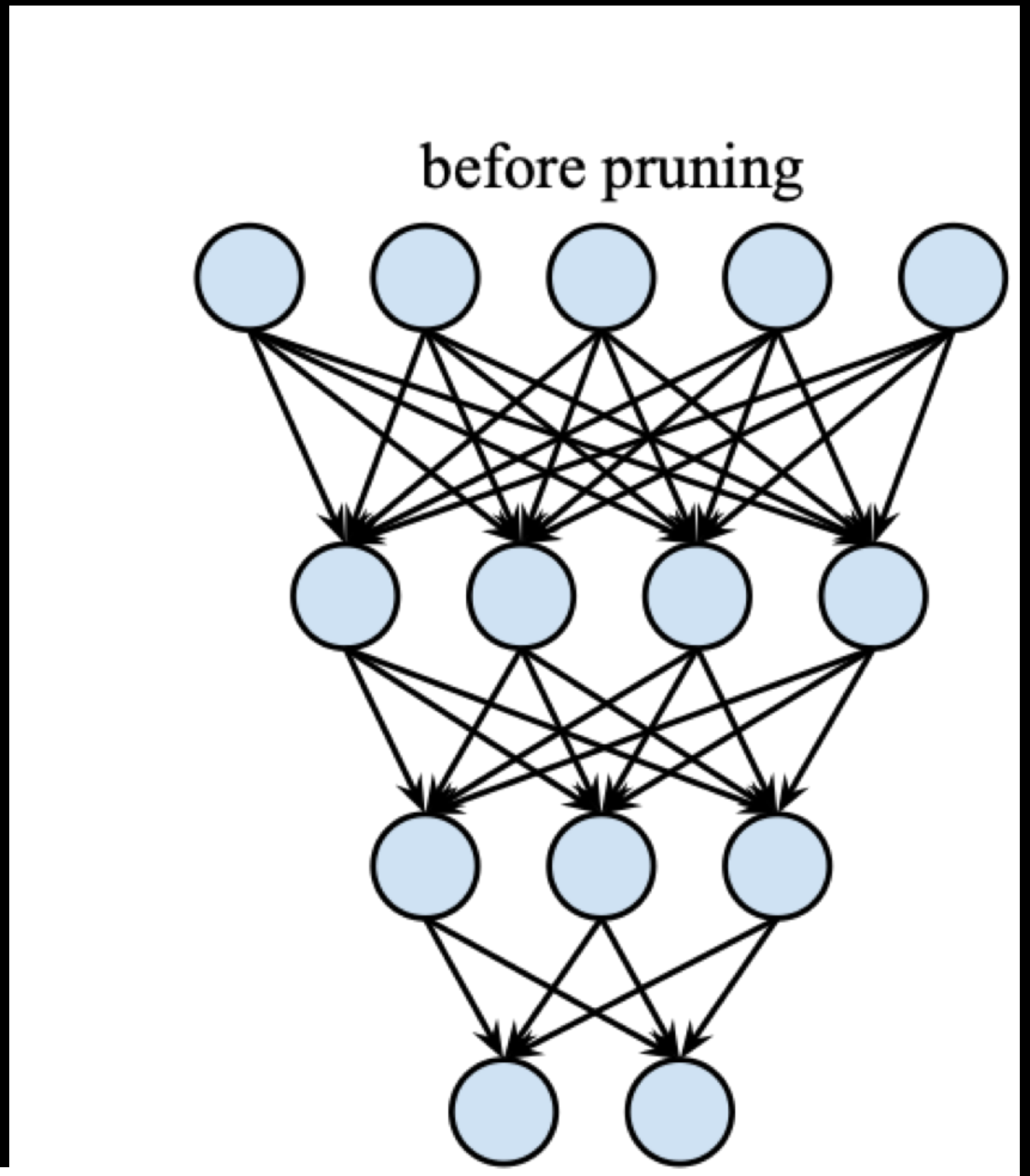
Brevitas like QKeras, but for PyTorch
- QAT library
- Support most common layers and activation functions

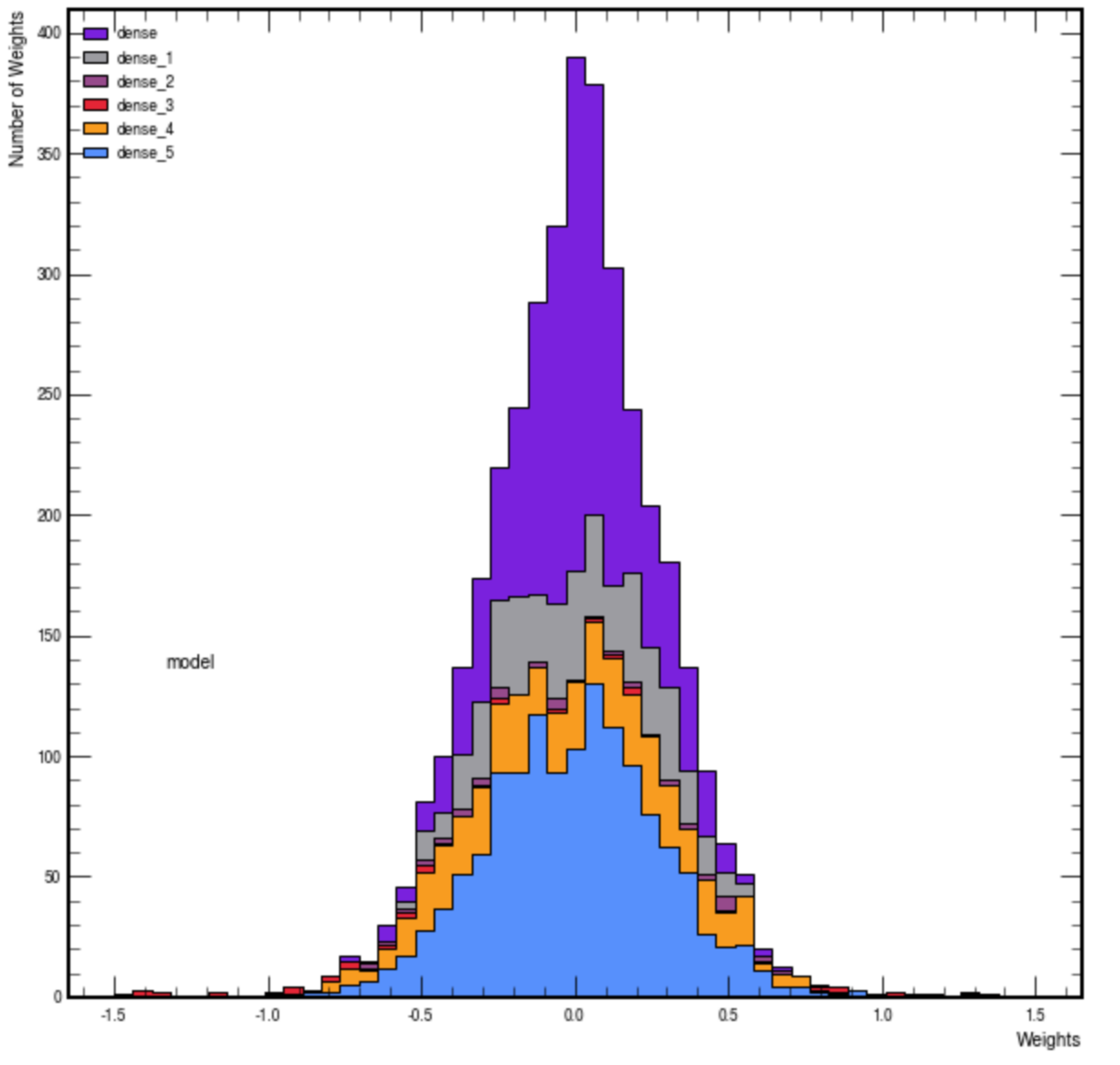
hls4ml collaborate with AMD/Xilinx FINN team to develop QONX
- Introducing 'Quant' node to ONNX graph
- Brevitas (PyTorch) and QKeras (Keras) can export QONNX
- hls4ml and FINN can import QONNX

Quantized ONNX (QONNX), J. Mitrevski et. al

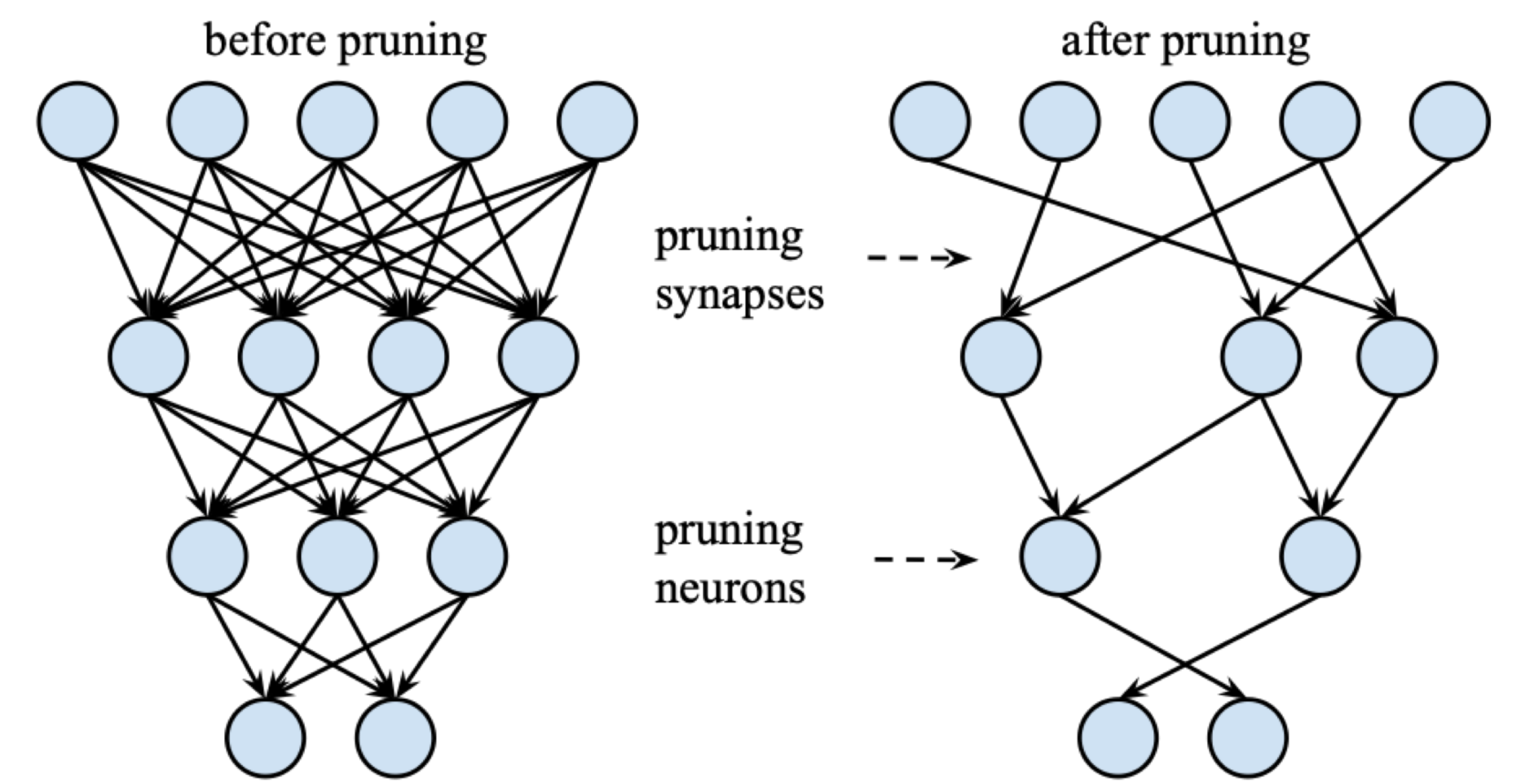


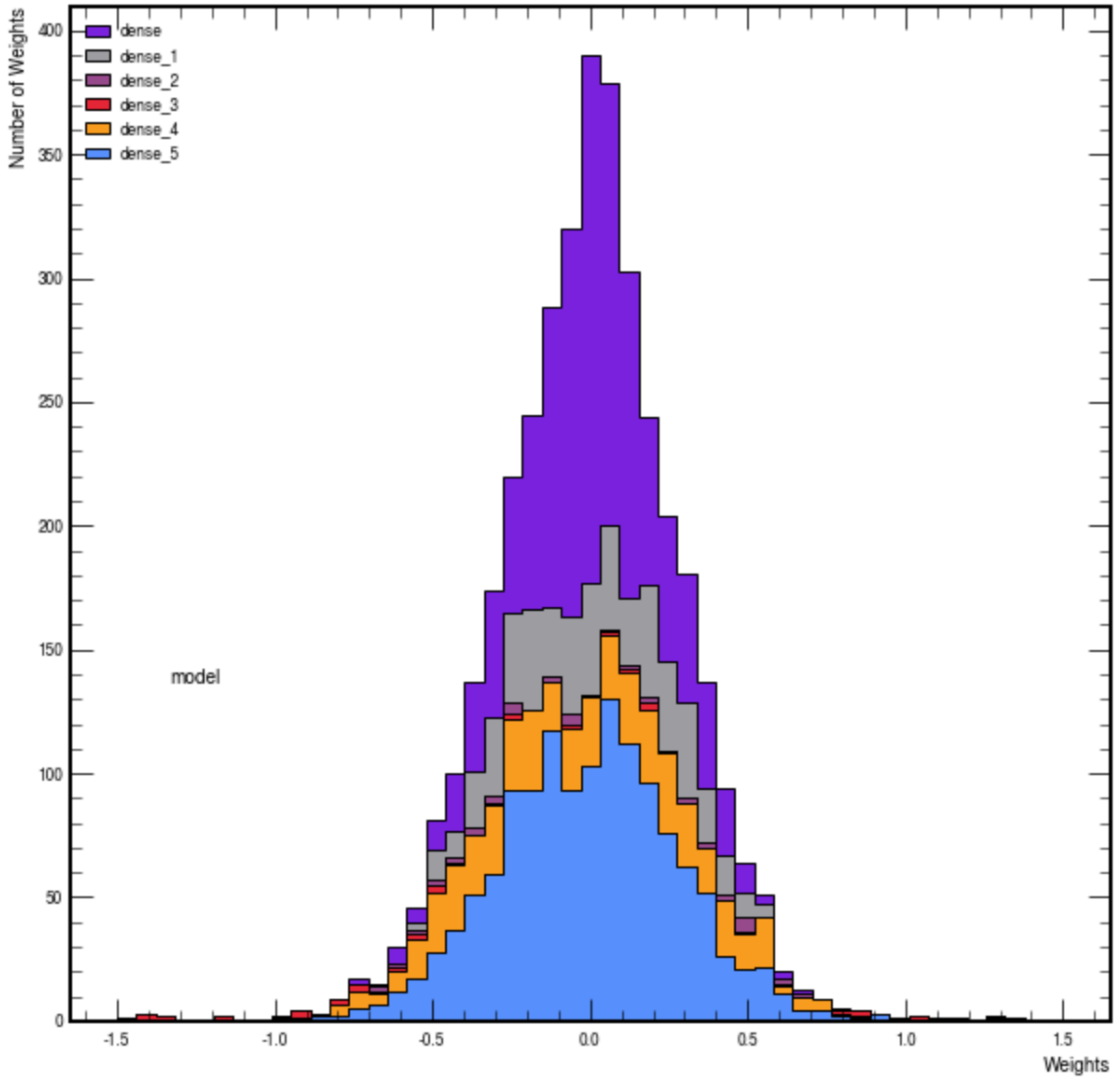
Pruning

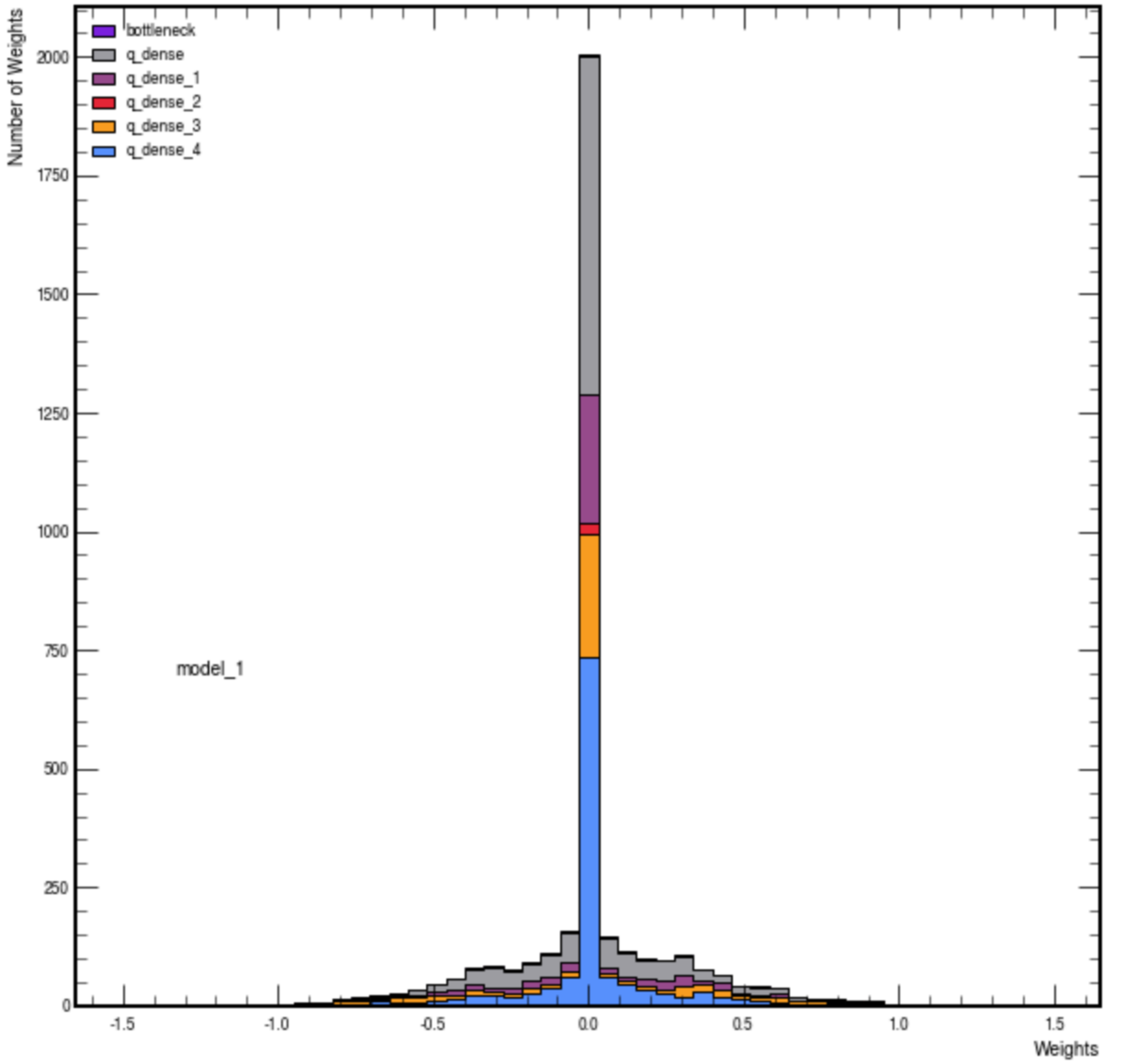

before pruning


Number of Weights
400
350
300
250
200
150
100
50
0
dense
dense_1
dense_2
dense_3
dense_4
dense_5
model
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Weights

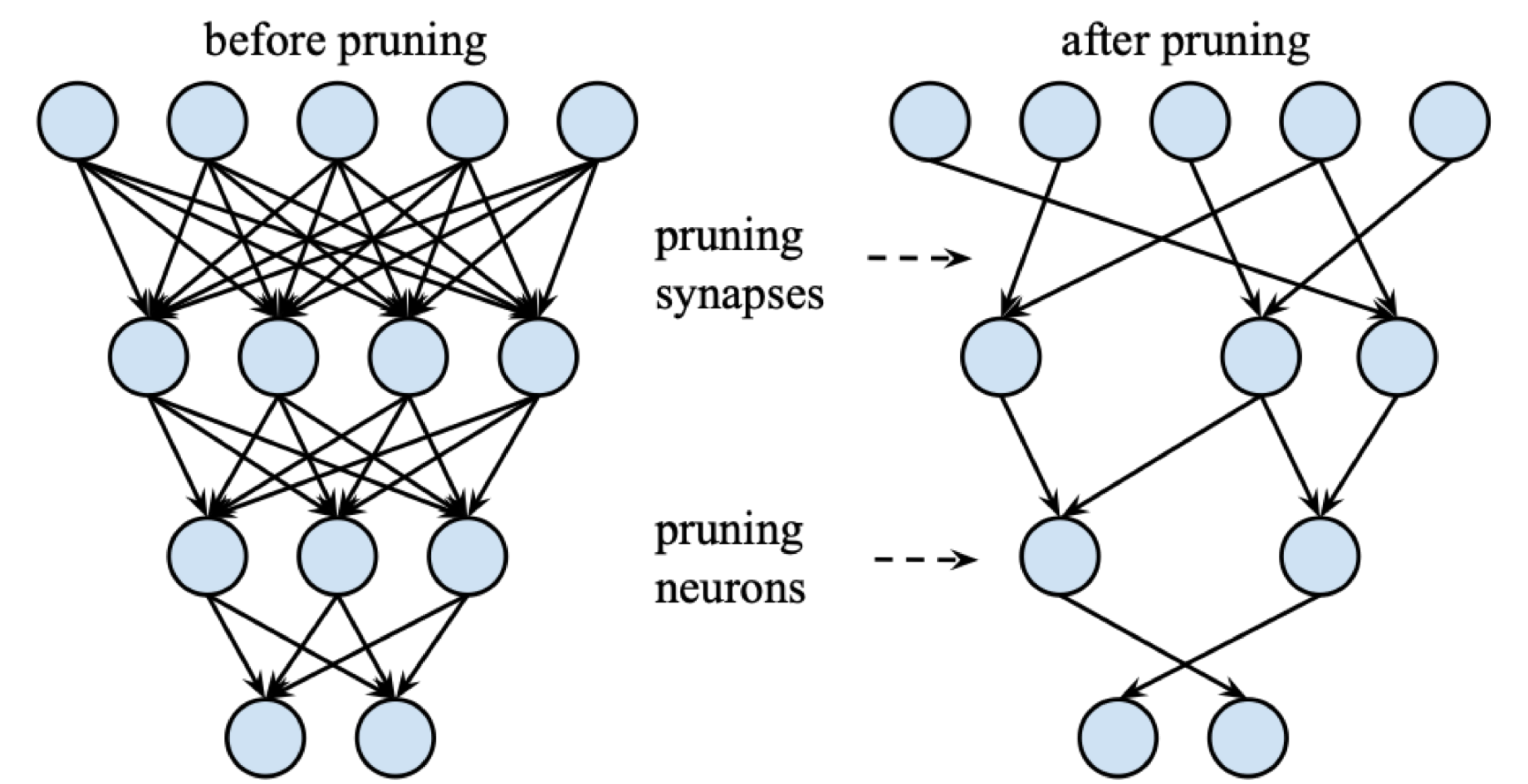
Pruning

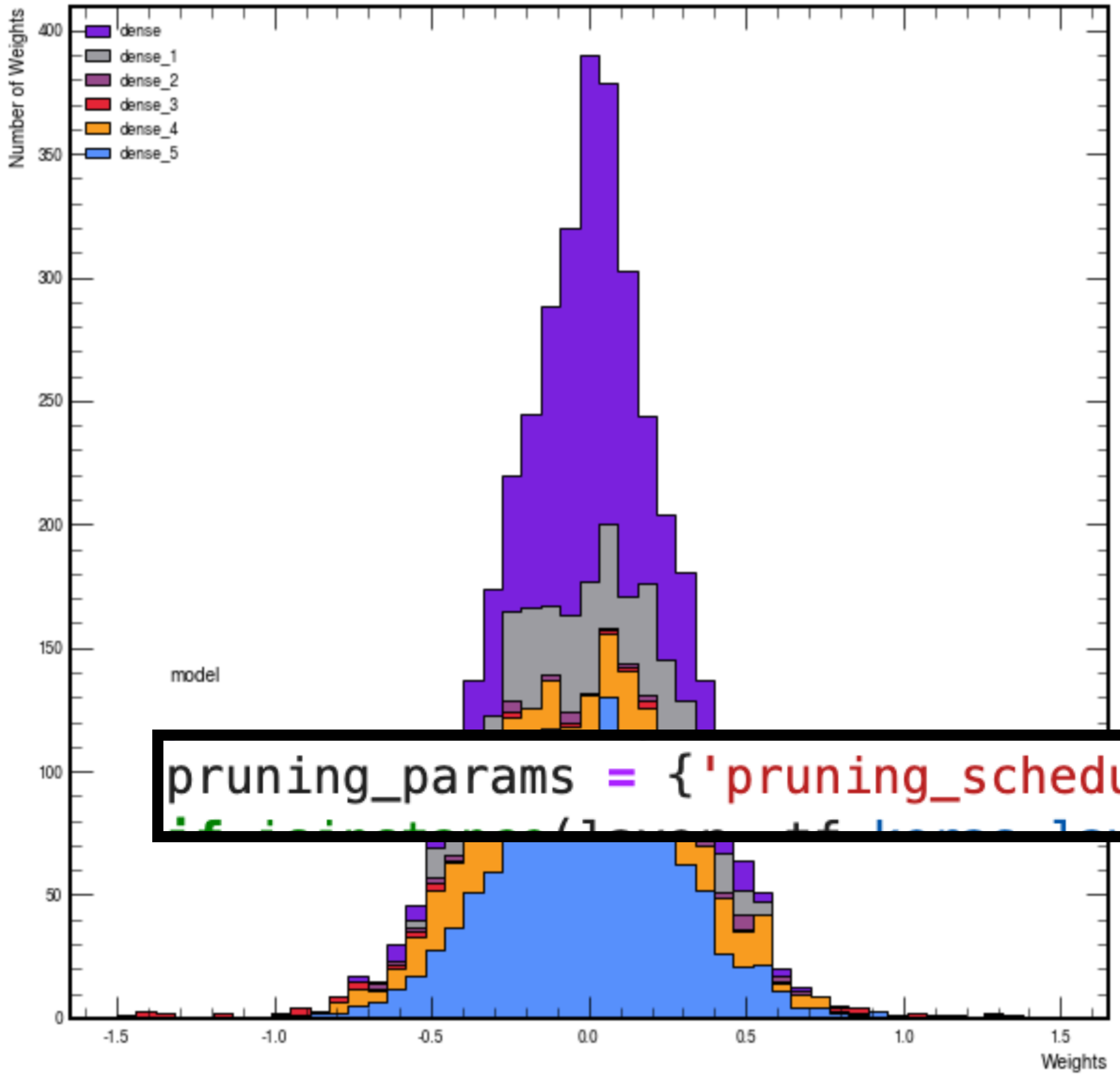

before pruning
after pruning
pruning synapses
pruning neurons


Number of Weights
400
350
300
250
200
150
100
50
0
dense
dense_1
dense_2
dense_3
dense_4
dense_5
model
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Weights

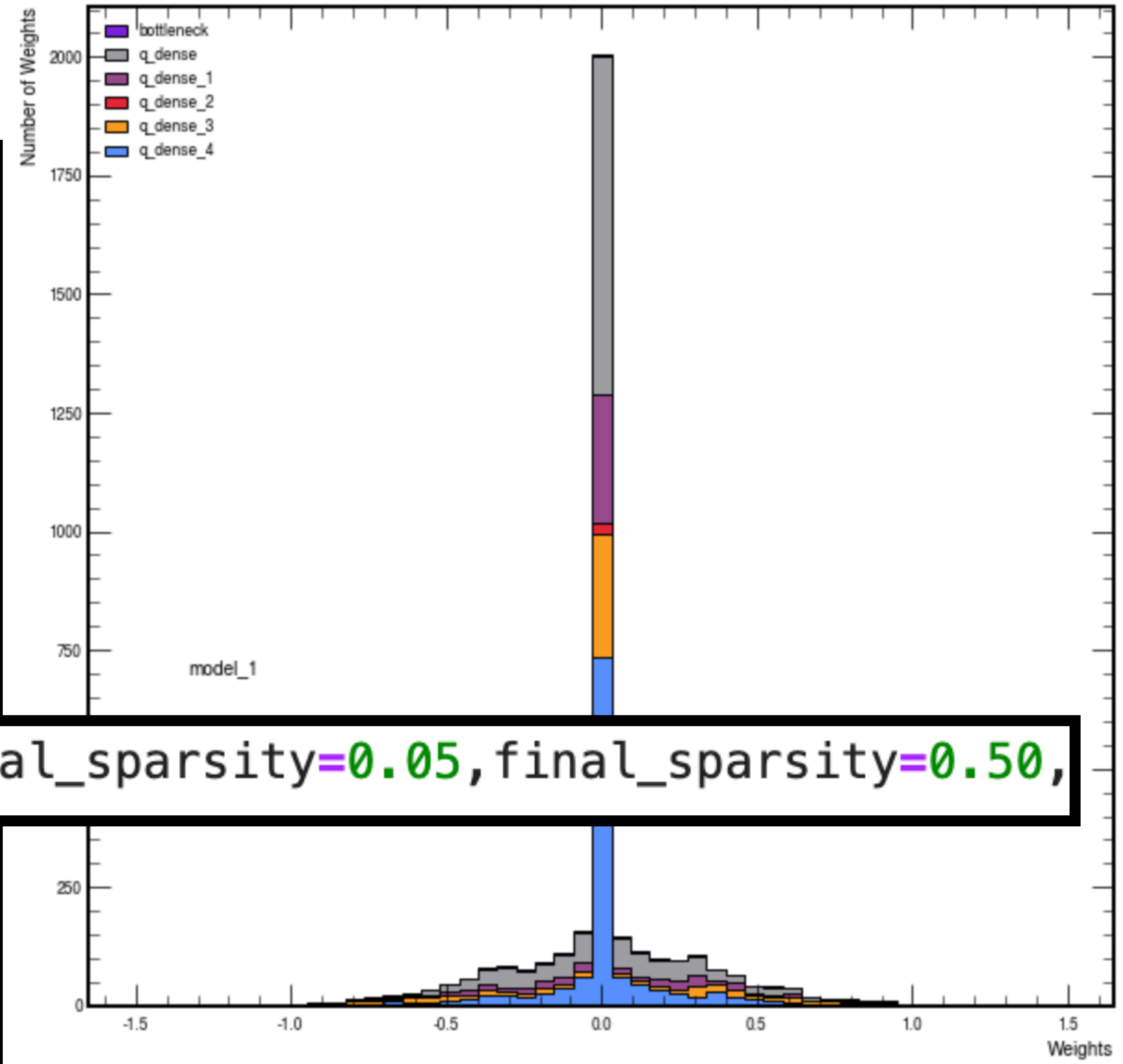

Number of Weights
2000
1750
1500
1250
1000
750
500
250
0
bottleneck
q_dense
q_dense_1
q_dense_2
q_dense_3
q_dense_4
model_1
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Weights

Pruning





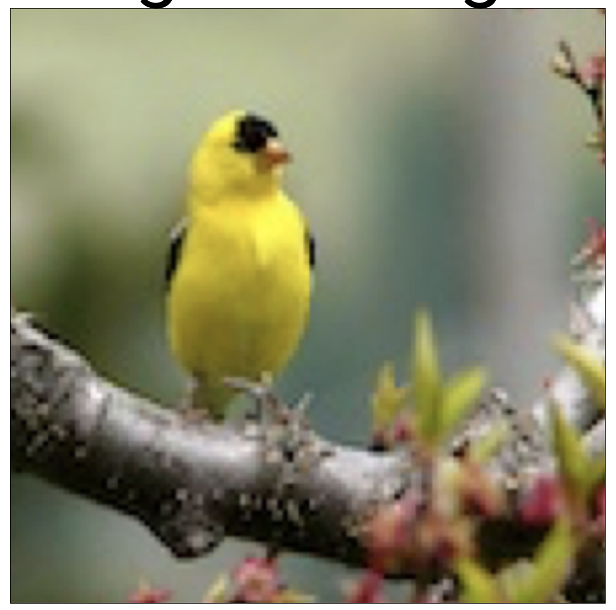
```
pruning_params = {'pruning_schedule': pruning_schedule.PolynomialDecay(initial_sparsity=0.05,final_sparsity=0.50,
```

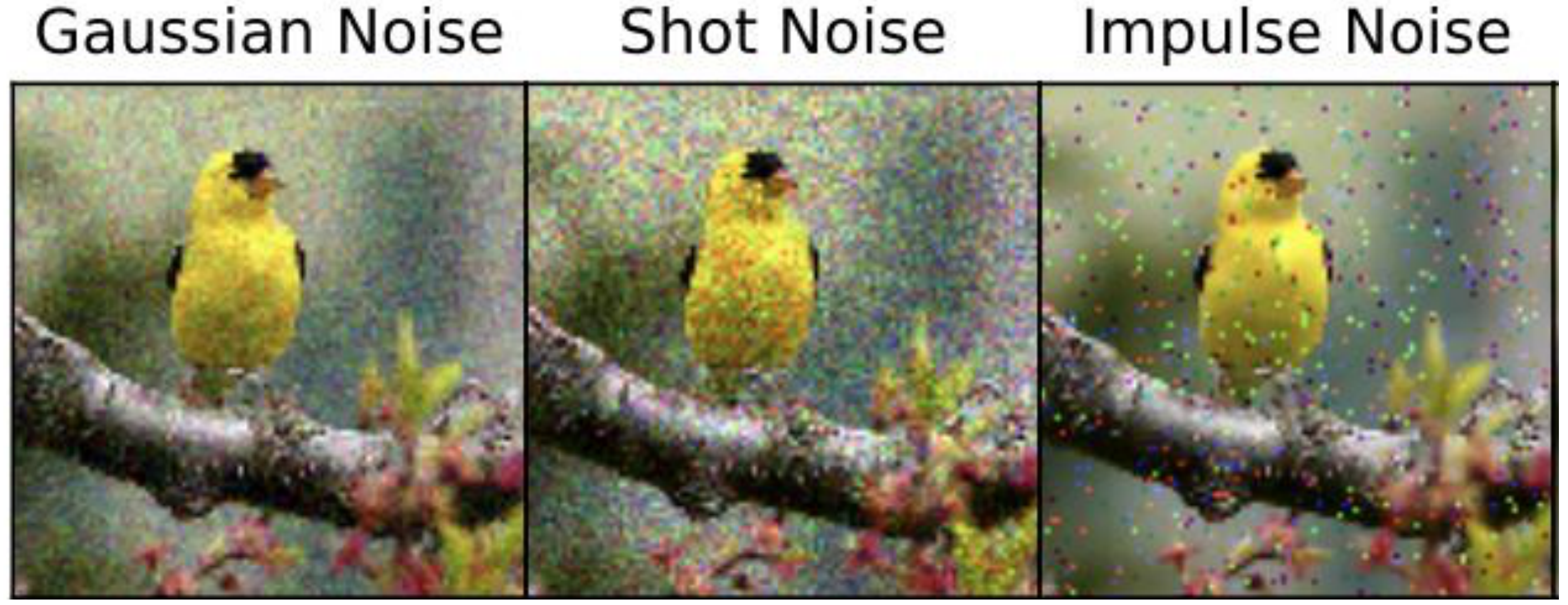


Hooker et al. (2021)

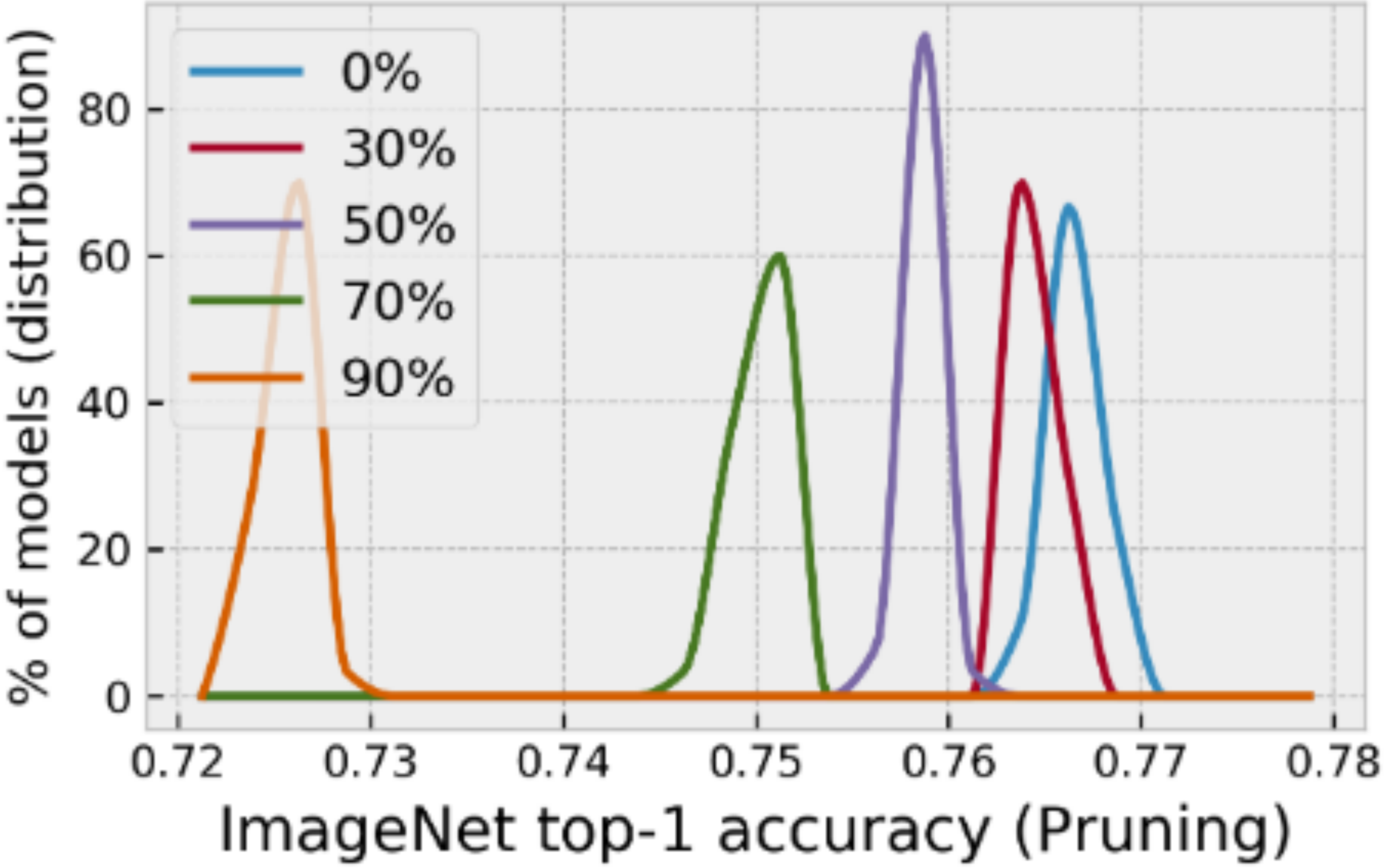
From Brian Bartoldson

Original image

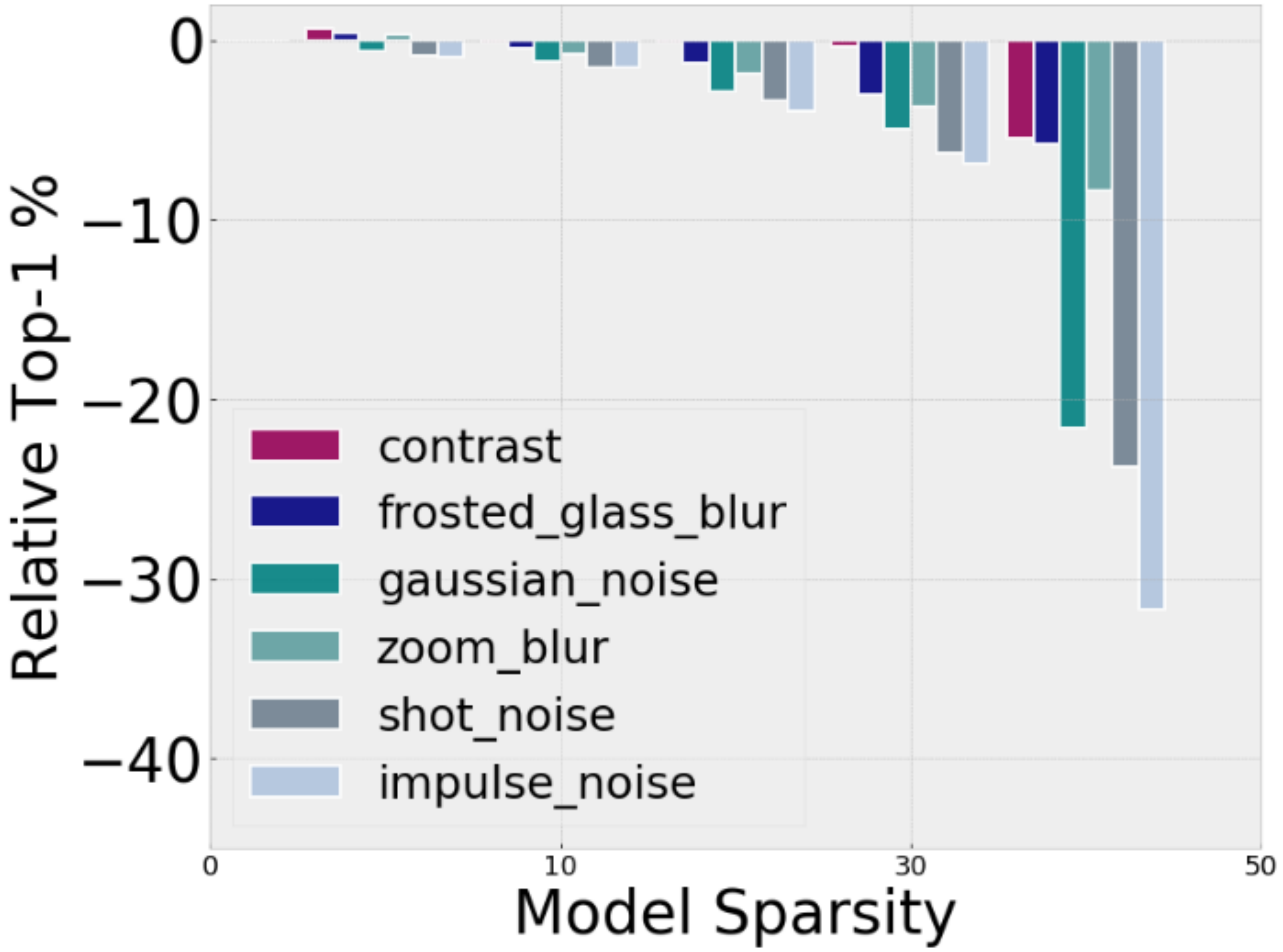




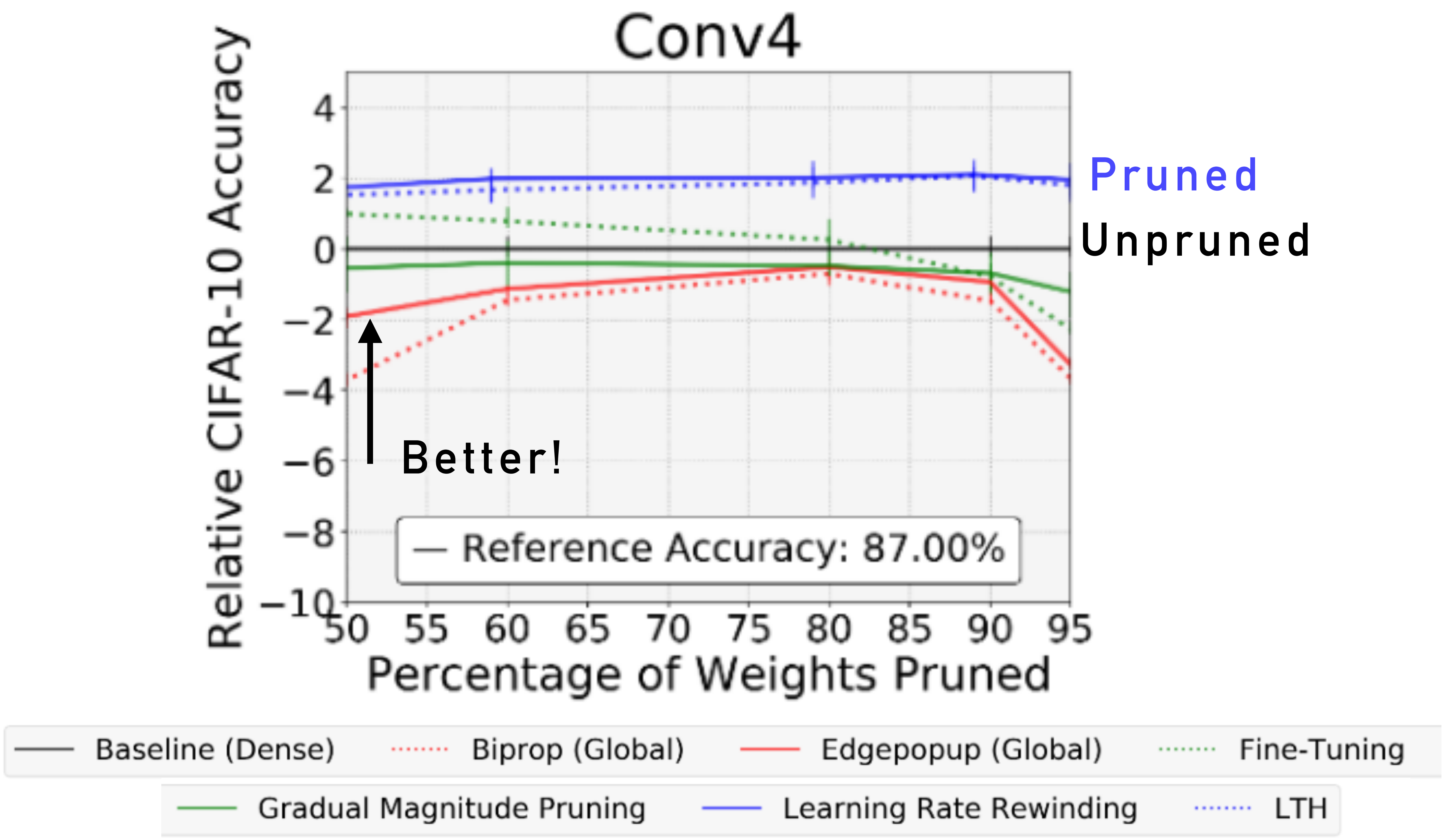
Less accurate



Less robust to noise

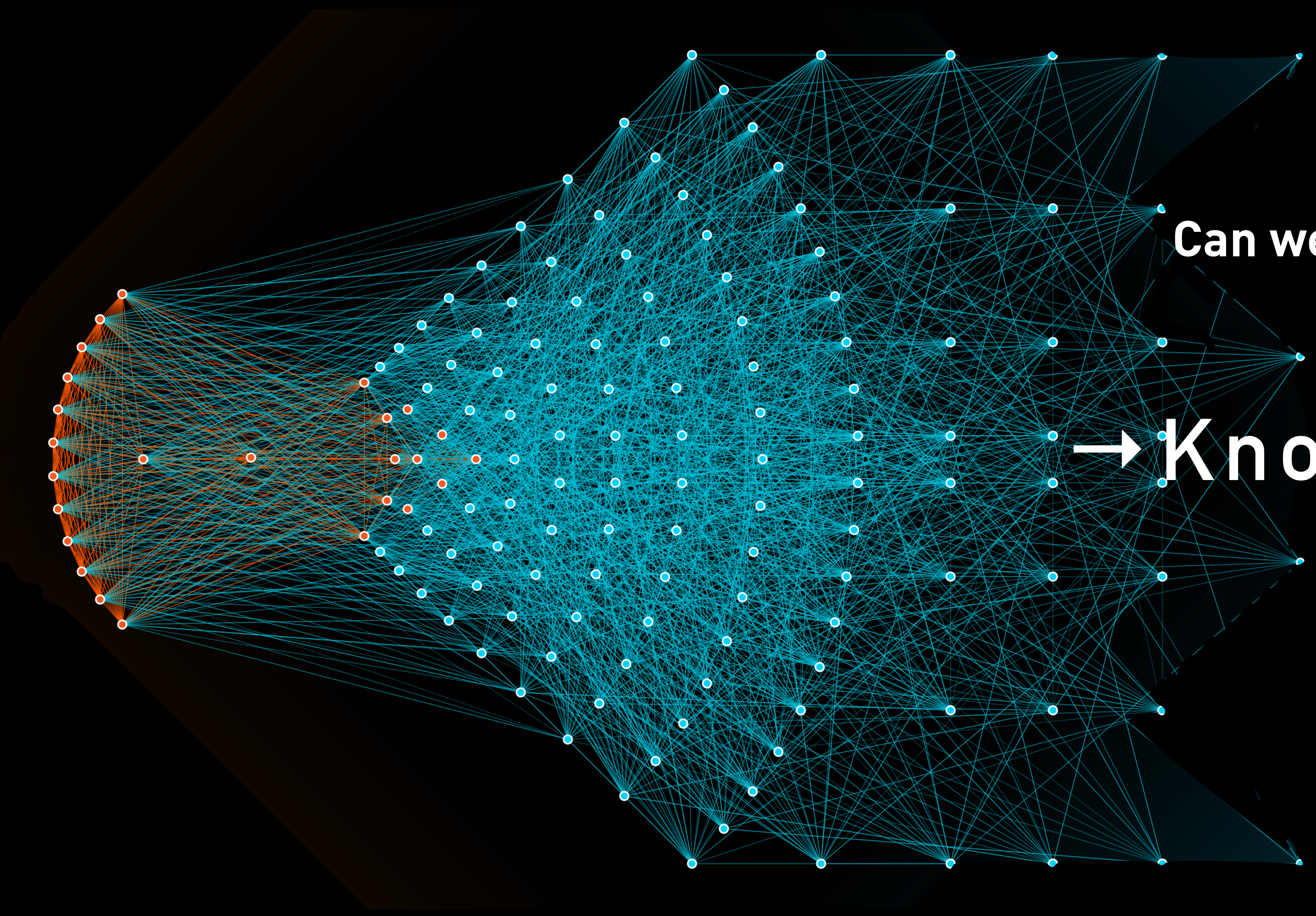


__From Brian Bartoldson__



There exists a optimal network WITHIN each network (lottery ticket)
Uncover it through pruning!

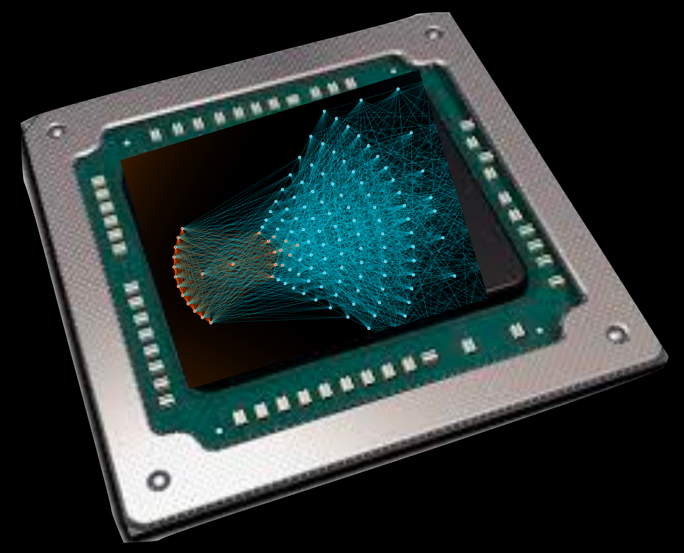
Diffenderfer, Bartoldson, et al. (2021)



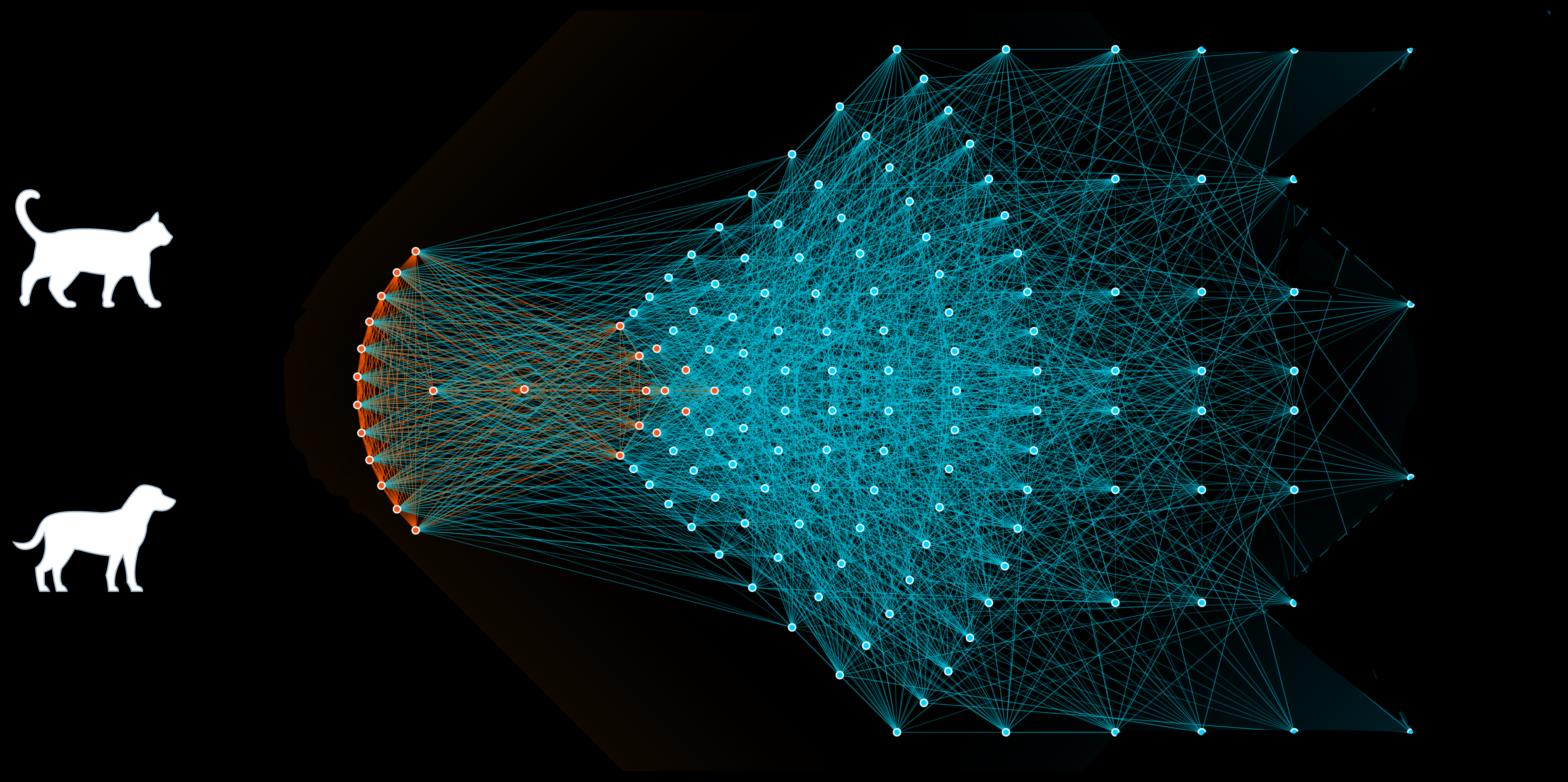
Train

Can we have the best of both worlds?

→Knowledge Distillation



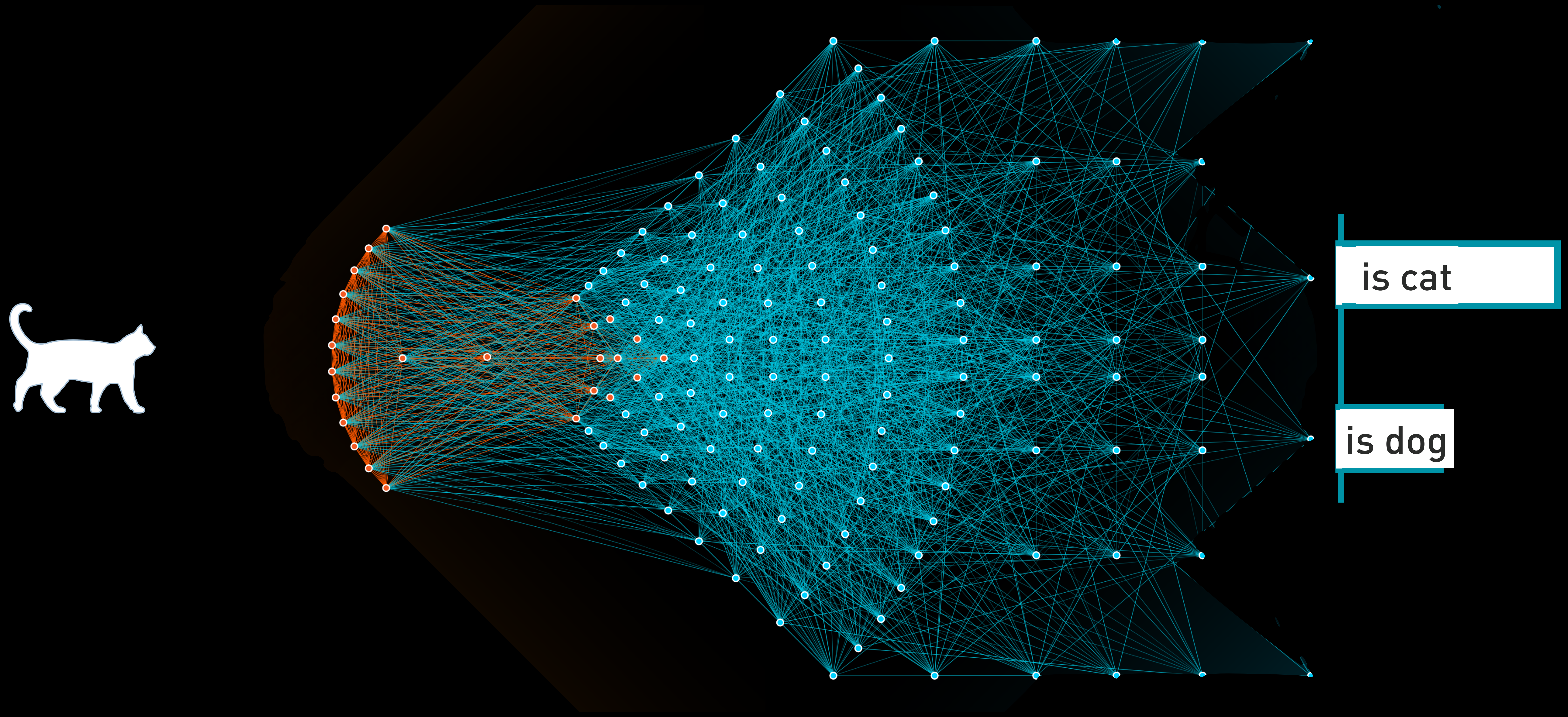
Inference

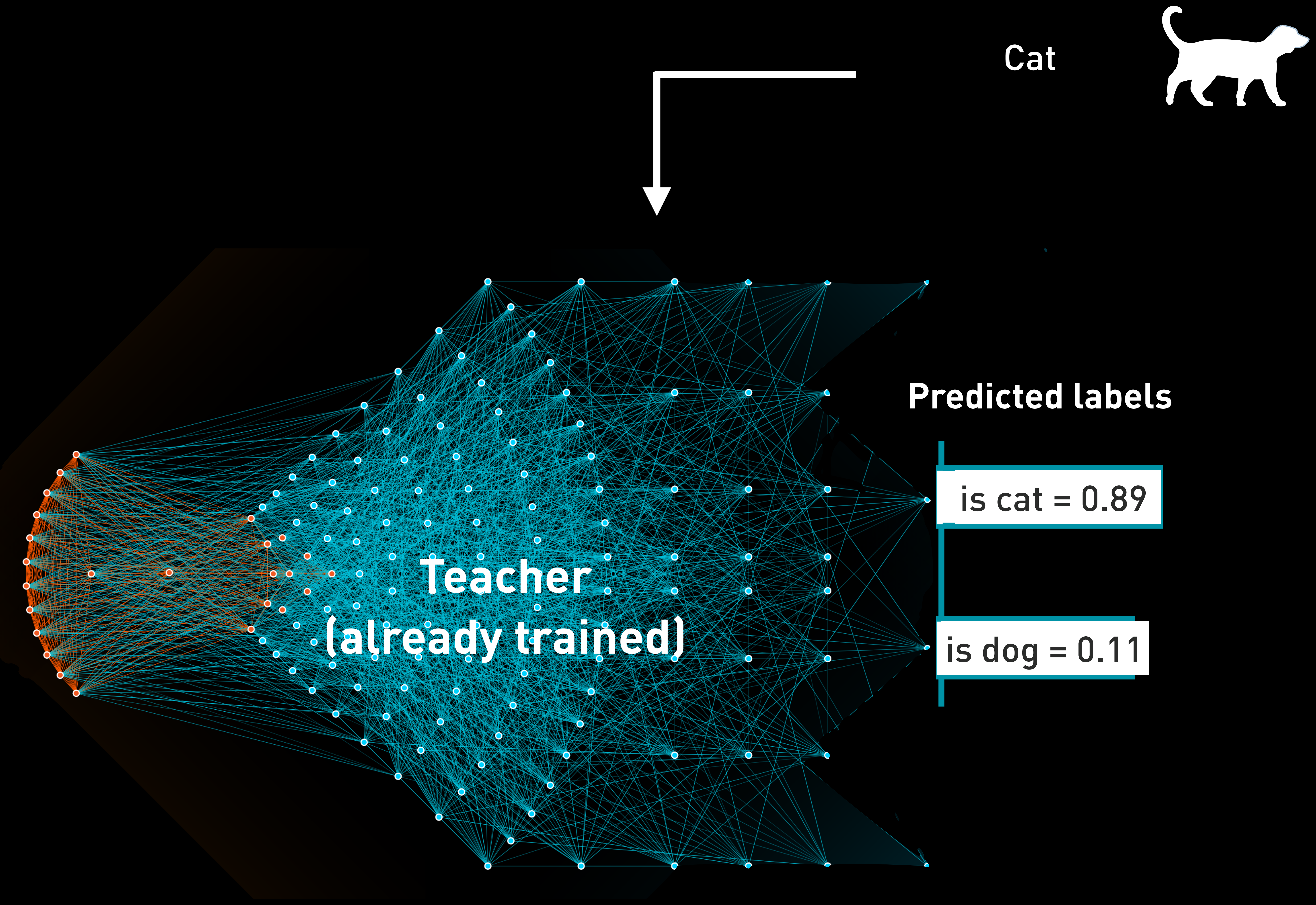


Cat

Dog

Cat





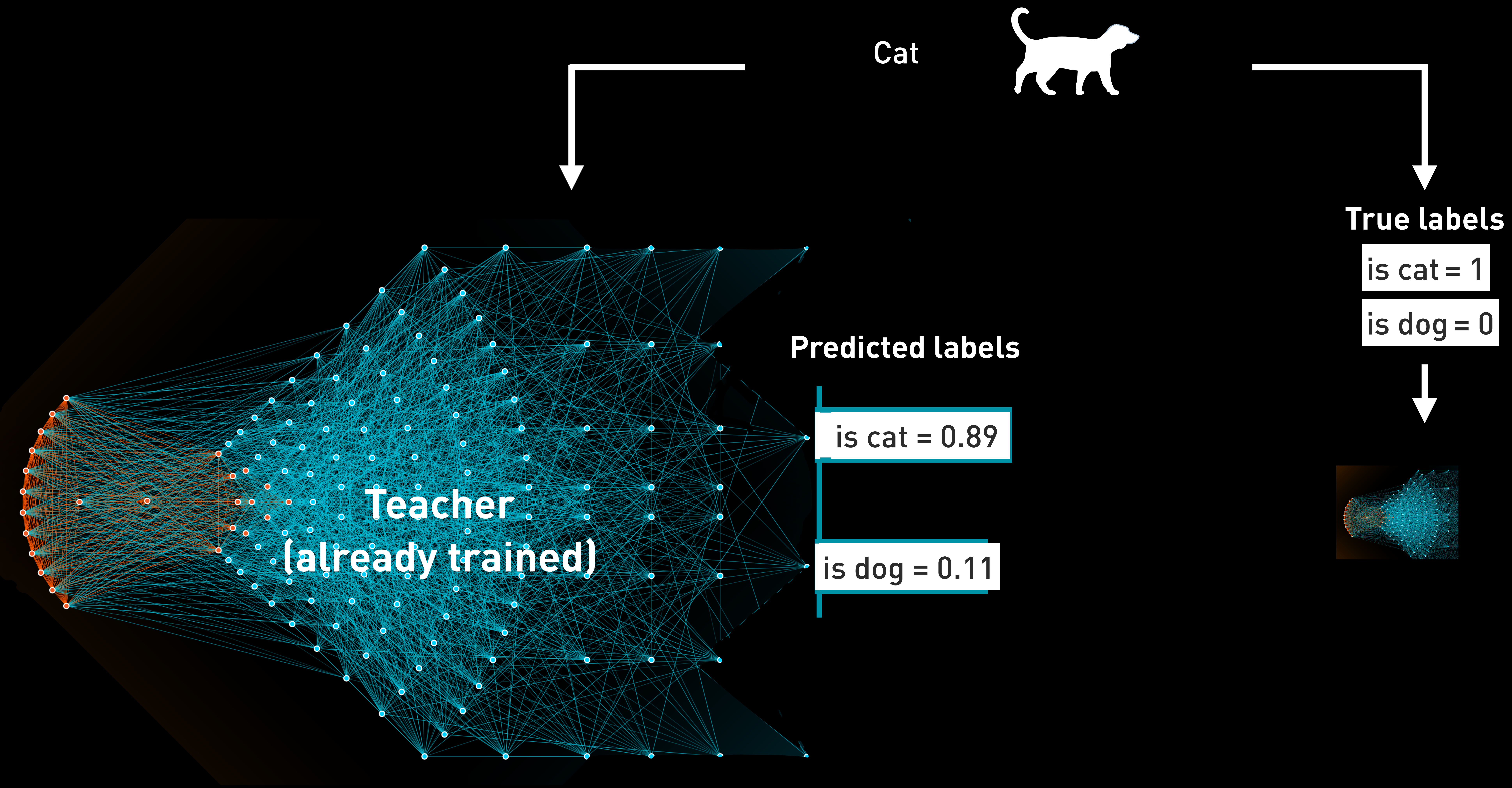
Cat

Teacher
(already trained)

Predicted labels

is cat = 0.89

is dog = 0.11



Cat

Teacher (already trained)

Predicted labels

is cat = 0.89

is dog = 0.11

True labels

is cat = 1

is dog = 0

Predicted labels

is cat = 0.46

is dog = 0.54

True labels

is cat = 0

is dog = 1

Soft labels contain information!!

Predicted labels

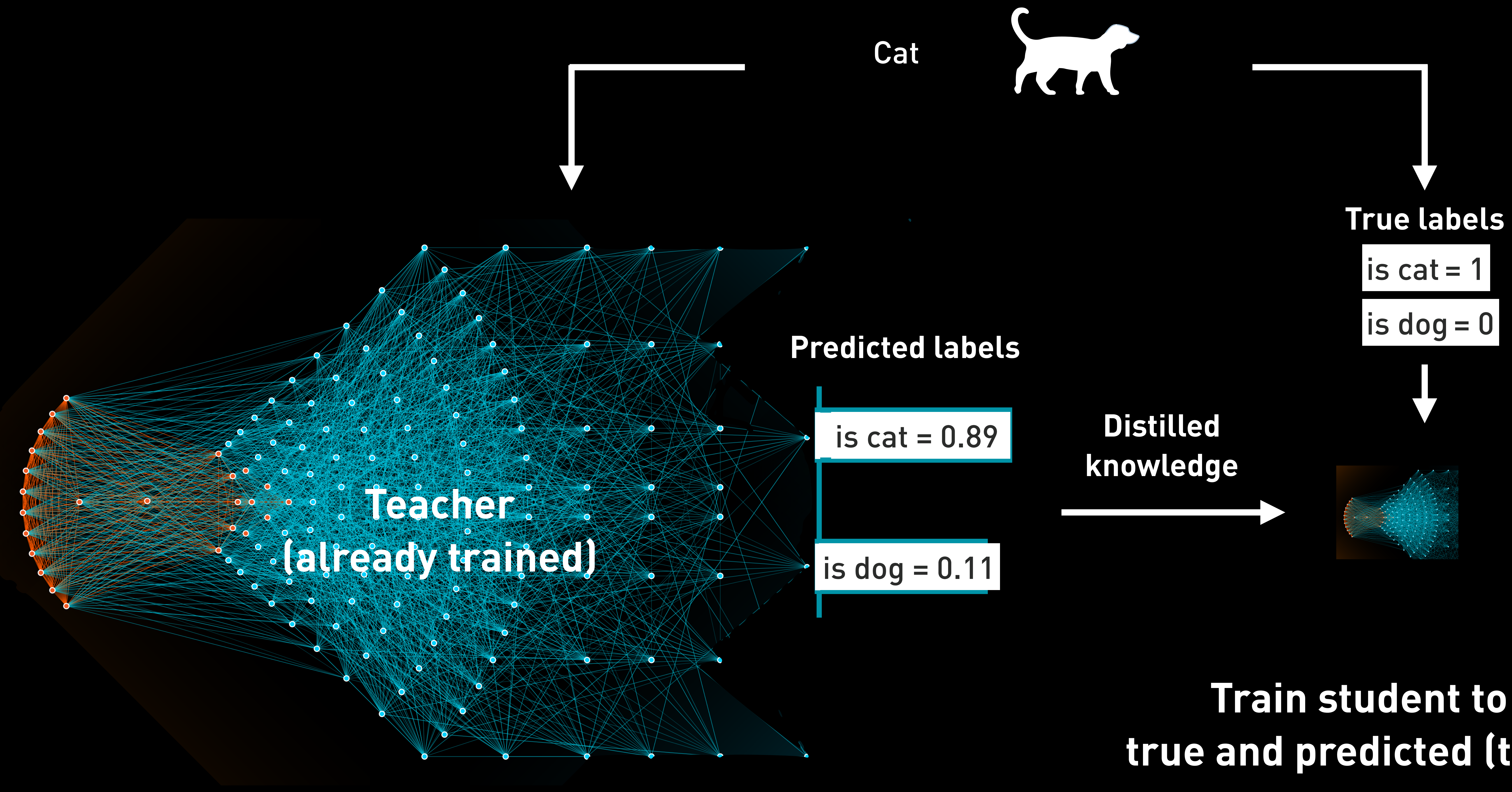
is cat = 0.03

is dog = 0.97

True labels

is cat = 0

is dog = 1


Cat

Teacher (already trained)

Predicted labels

is cat = 0.89

is dog = 0.11

Distilled knowledge

True labels

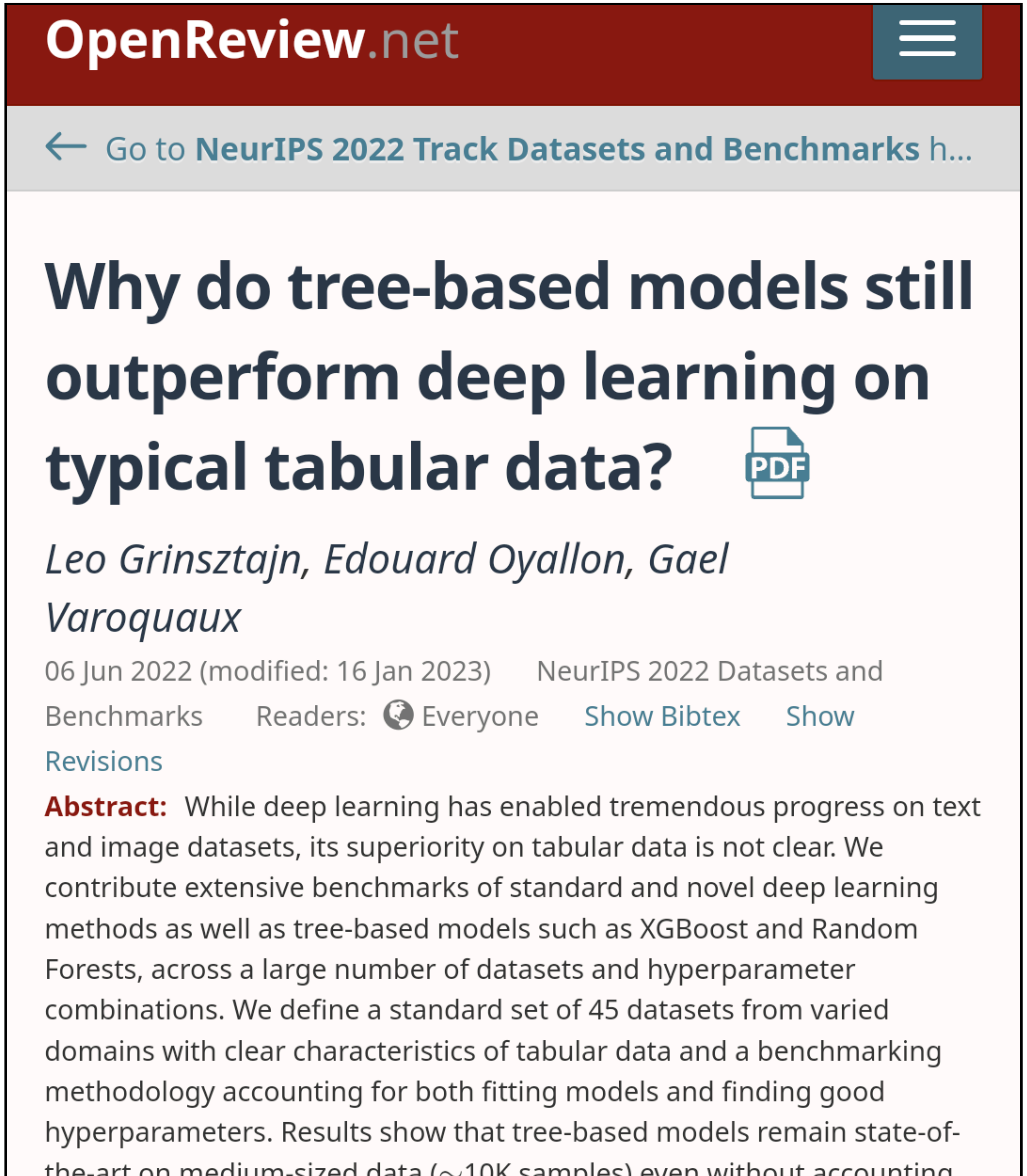
is cat = 1

is dog = 0

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!



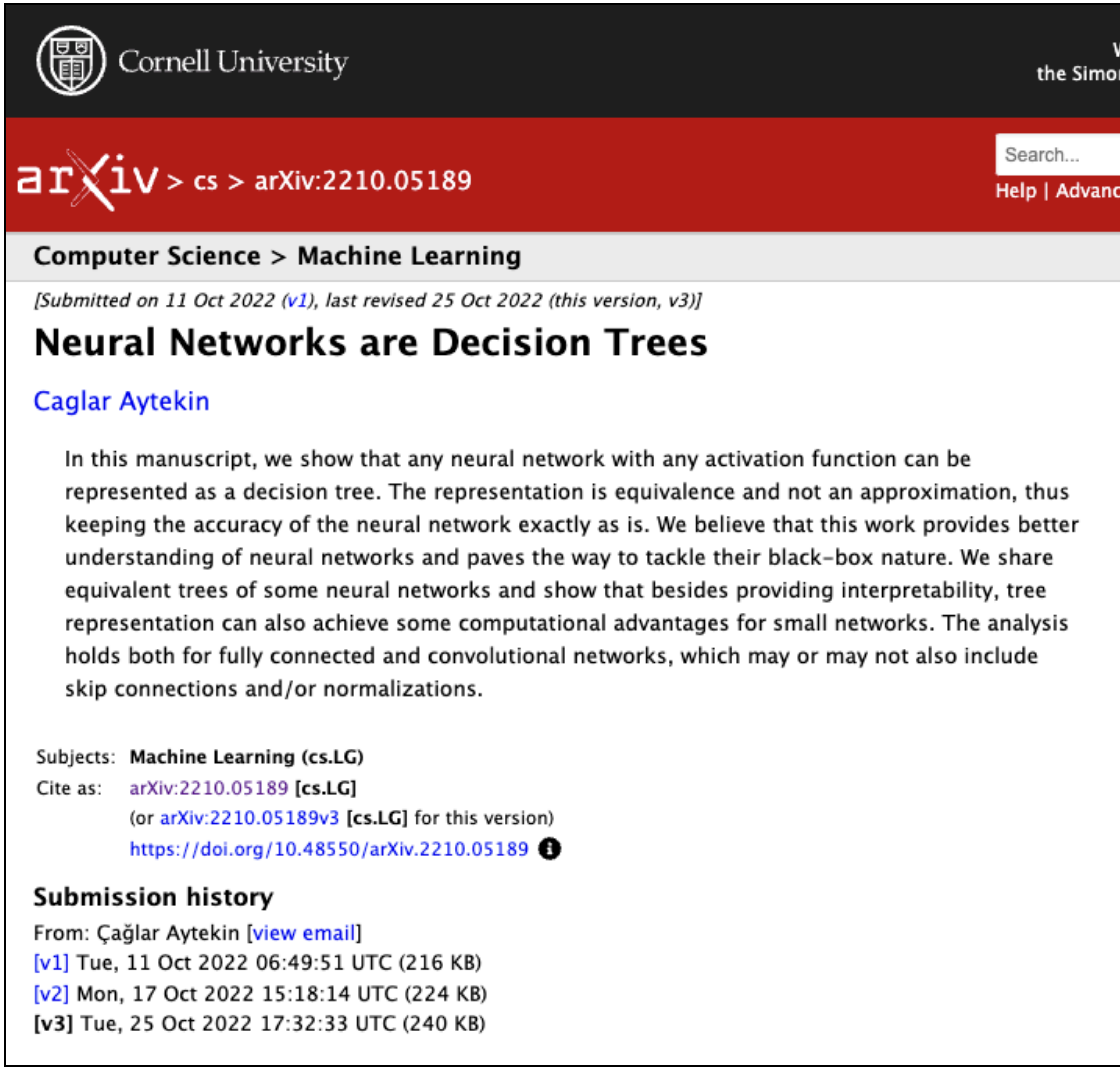
← Go to **NeurIPS 2022 Track Datasets and Benchmarks** h...

Why do tree-based models still outperform deep learning on typical tabular data?

Leo Grinsztajn, Edouard Oyallon, Gael Varoquaux

06 Jun 2022 (modified: 16 Jan 2023) NeurIPS 2022 Datasets and Benchmarks Readers: Everyone Show Bibtex Show Revisions

Abstract: While deep learning has enabled tremendous progress on text and image datasets, its superiority on tabular data is not clear. We contribute extensive benchmarks of standard and novel deep learning methods as well as tree-based models such as XGBoost and Random Forests, across a large number of datasets and hyperparameter combinations. We define a standard set of 45 datasets from varied domains with clear characteristics of tabular data and a benchmarking methodology accounting for both fitting models and finding good hyperparameters. Results show that tree-based models remain state-of-the-art on medium-sized data (~10K samples) even without accounting



Computer Science > Machine Learning

[Submitted on 11 Oct 2022 (v1), last revised 25 Oct 2022 (this version, v3)]

Neural Networks are Decision Trees

Caglar Aytekin

In this manuscript, we show that any neural network with any activation function can be represented as a decision tree. The representation is equivalence and not an approximation, thus keeping the accuracy of the neural network exactly as is. We believe that this work provides better understanding of neural networks and paves the way to tackle their black-box nature. We share equivalent trees of some neural networks and show that besides providing interpretability, tree representation can also achieve some computational advantages for small networks. The analysis holds both for fully connected and convolutional networks, which may or may not also include skip connections and/or normalizations.

Subjects:	**Machine Learning (cs.LG)**
Cite as:	arXiv:2210.05189 **[cs.LG]**
	(or arXiv:2210.05189v3 **[cs.LG]** for this version)
	https://doi.org/10.48550/arXiv.2210.05189

Submission history

From: Çağlar Aytekin [view email]
[v1] Tue, 11 Oct 2022 06:49:51 UTC (216 KB)
[v2] Mon, 17 Oct 2022 15:18:14 UTC (224 KB)
[v3] Tue, 25 Oct 2022 17:32:33 UTC (240 KB)

https://arxiv.org/abs/2210.05189


OpenReview.net

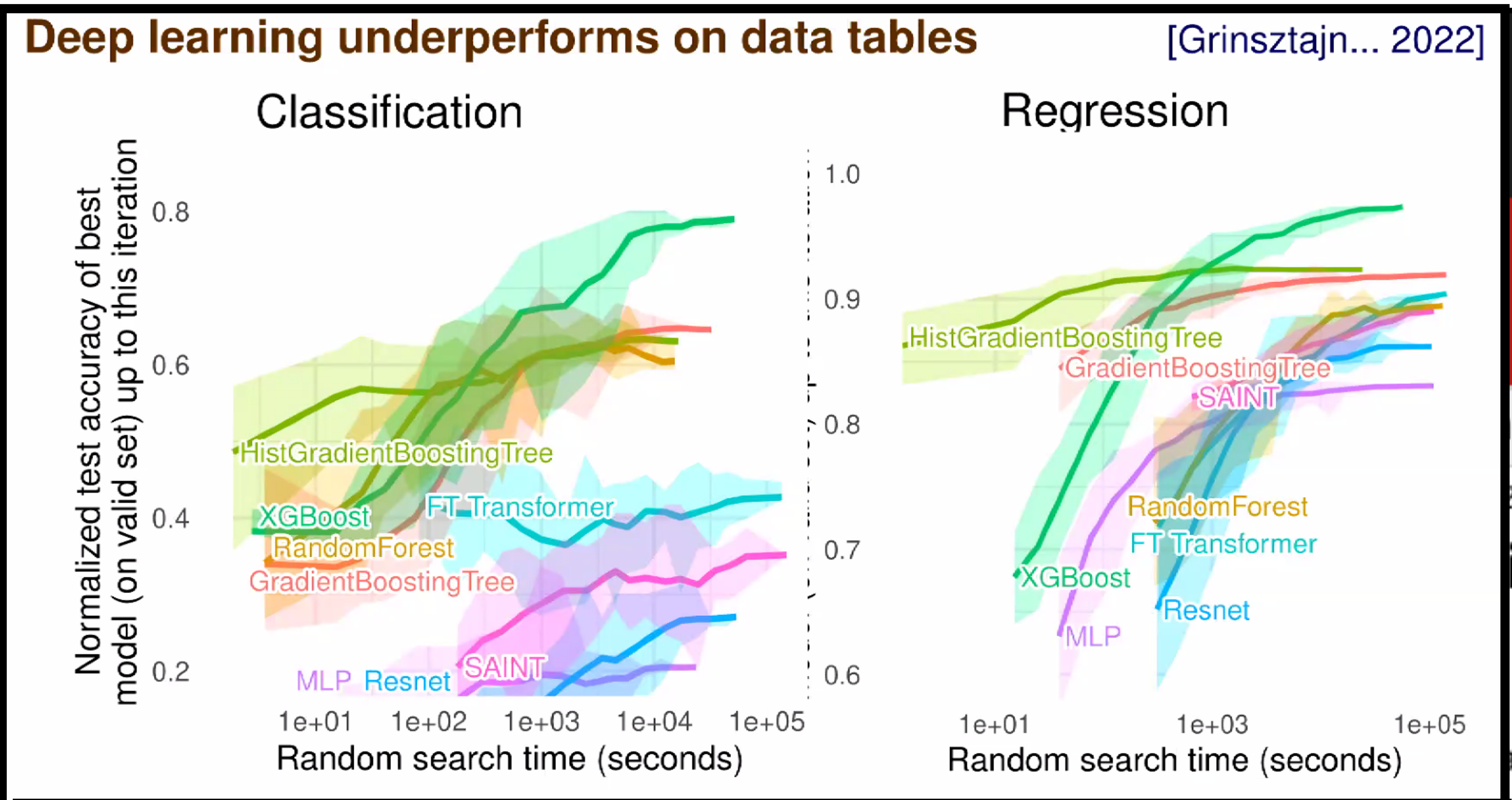
← Go to **NeurIPS 2022 Track Datase**

Why do tree-based outperform deep le typical tabular data

Leo
Varo

06 Jun
Bench
Revisi

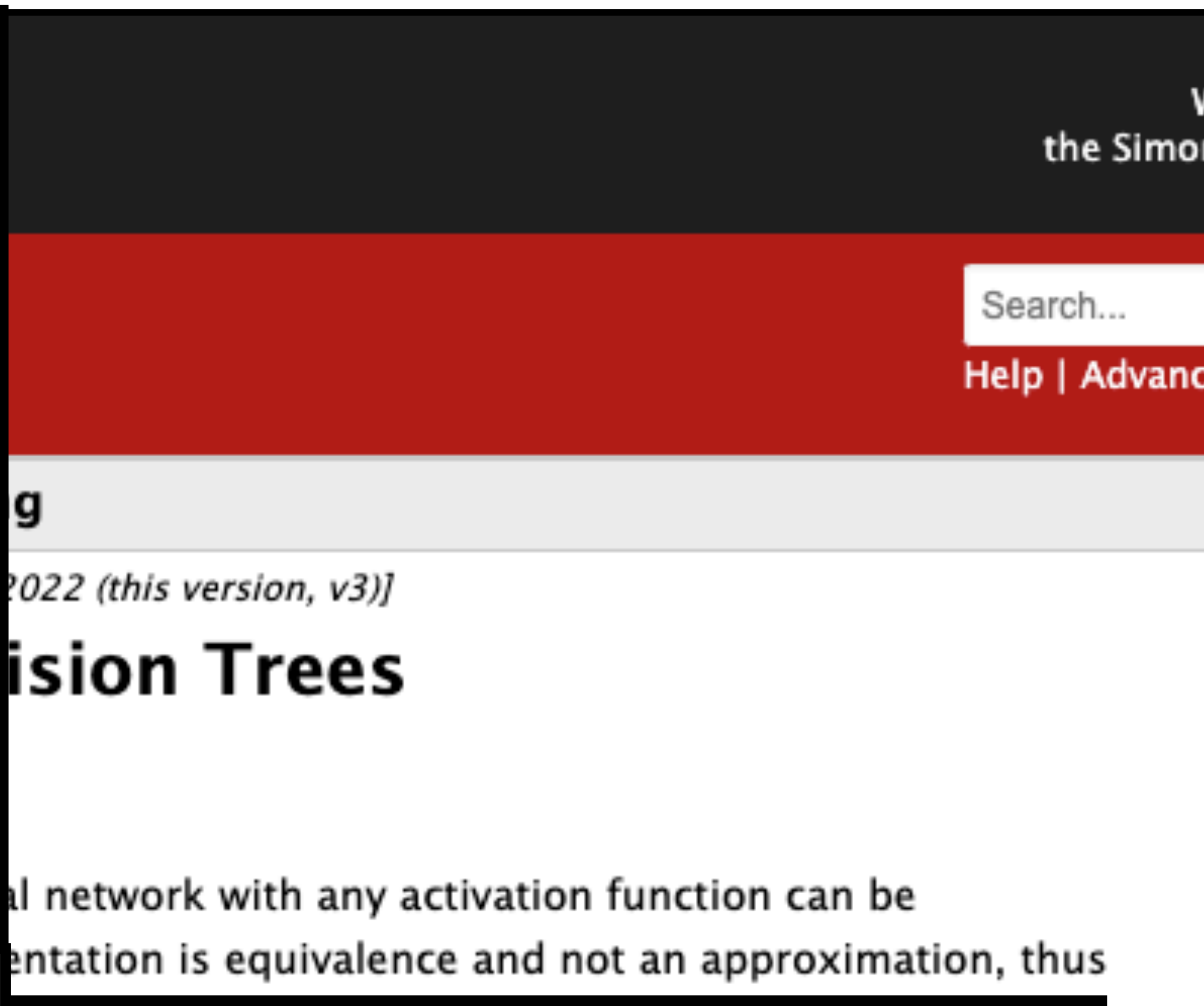
Abstra
and im
contri
metho
Forest
combi
domai
methodology accounting for both fitting models and finding good hyperparameters. Results show that tree-based models remain state-of-the-art on medium-sized data ($\sim$10K samples) even without accounting


Deep learning underperforms on data tables [Grinsztajn... 2022]

From the IML workshop

Inductive bias

- Trees are still very relevant
  (maybe in a few months, we'll make them obsolete ☻)
- It's really not all about scale
  - Architecture matters
  - Some physics does constrain compute
  - The economics matter, and are dodgy


the Simo
Search...
Help | Advanc
g
2022 (this version, v3)]

ision Trees

l network with any activation function can be
entation is equivalence and not an approximation, thus

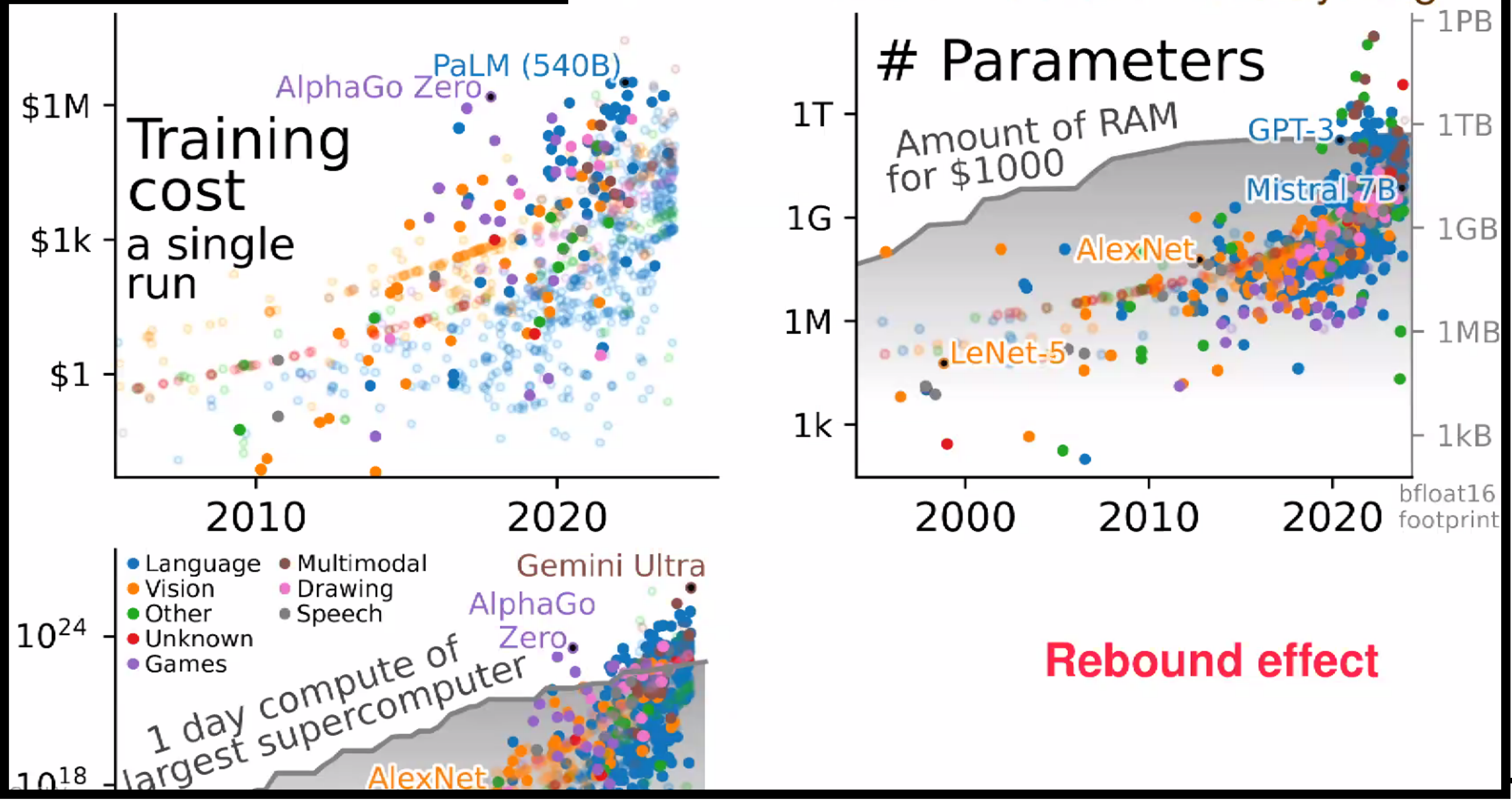
represent
holds bot
skip conn

Subjects: Mac
Cite as: arXi
(or a
http

Submissio
From: Çağlar
[v1] Tue, 11 O
[v2] Mon, 17
[v3] Tue, 25

networks are cool

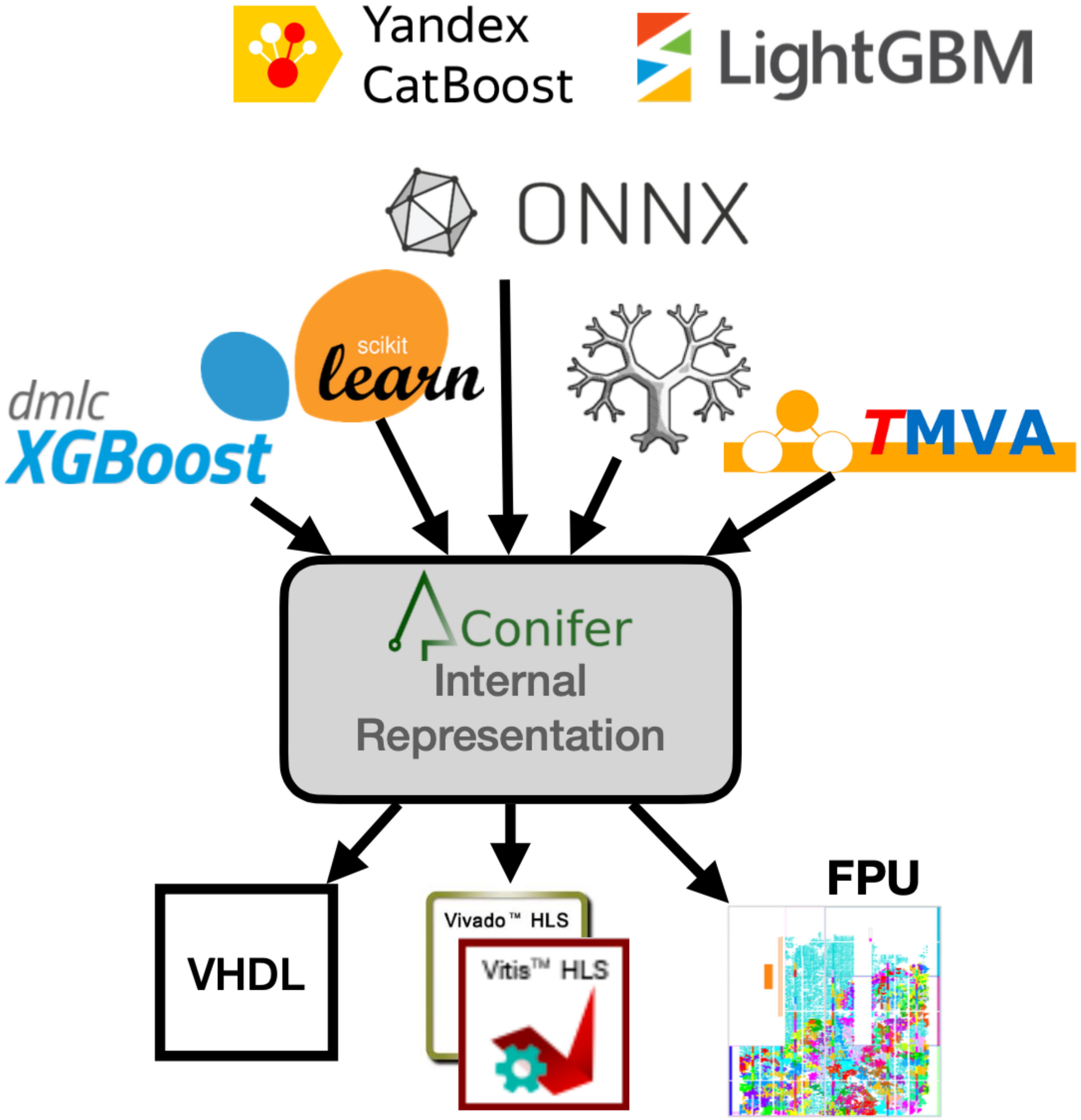
e shown us data + GPU solves everything



Rebound effect

https://arxiv.org/abs/2210.05189

%VU9P	Accuracy	Latency	DSP	LUT
qDNN	75.6%	40 ns	22 (~0%)	1%
BDT	74.9%	5 ns	-	0.5%



hls4ml


Quantised input data

Floating point model

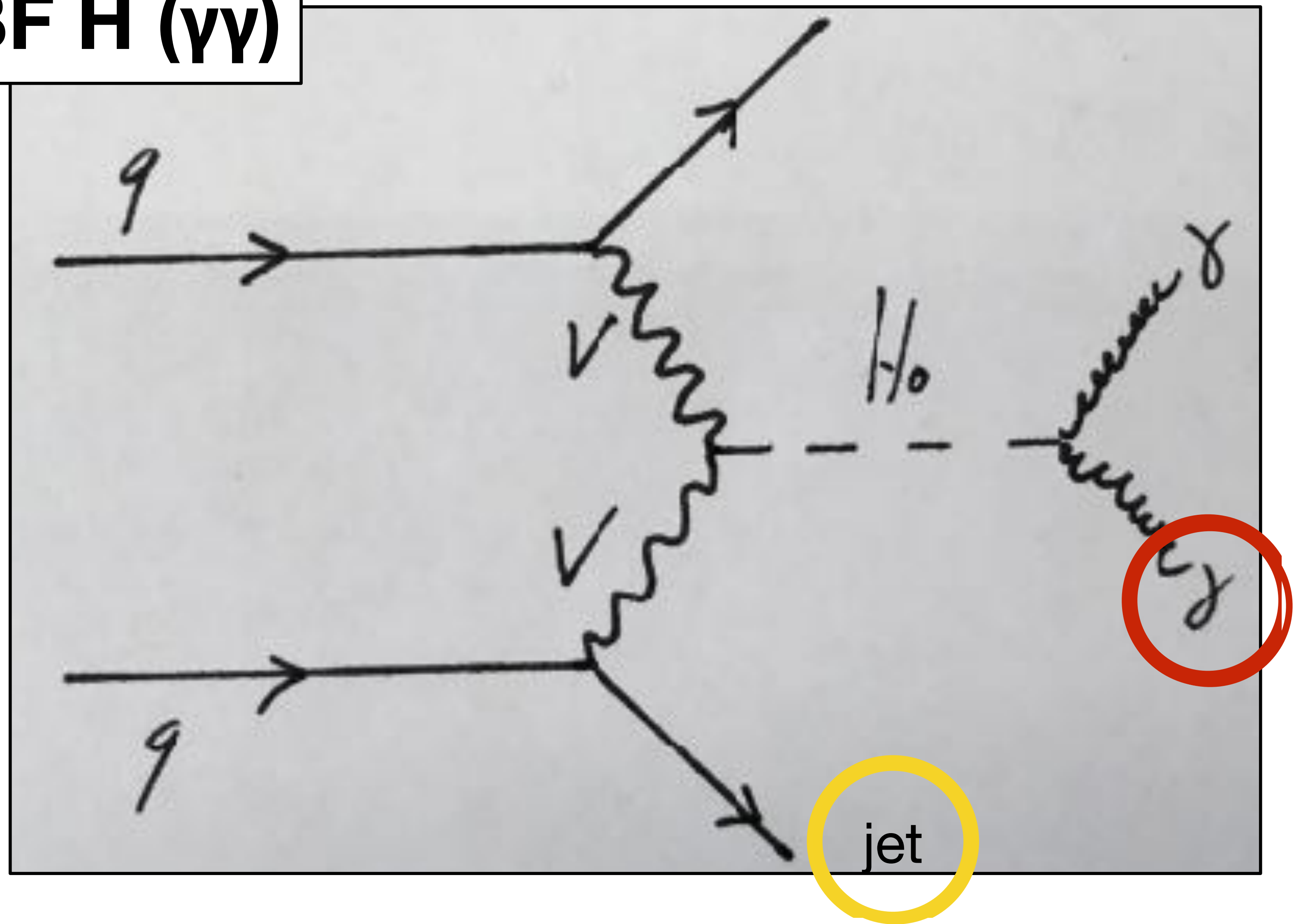
Compressed model
(Quantised + Pruned)

Conifer
hls4ml

Firmware design

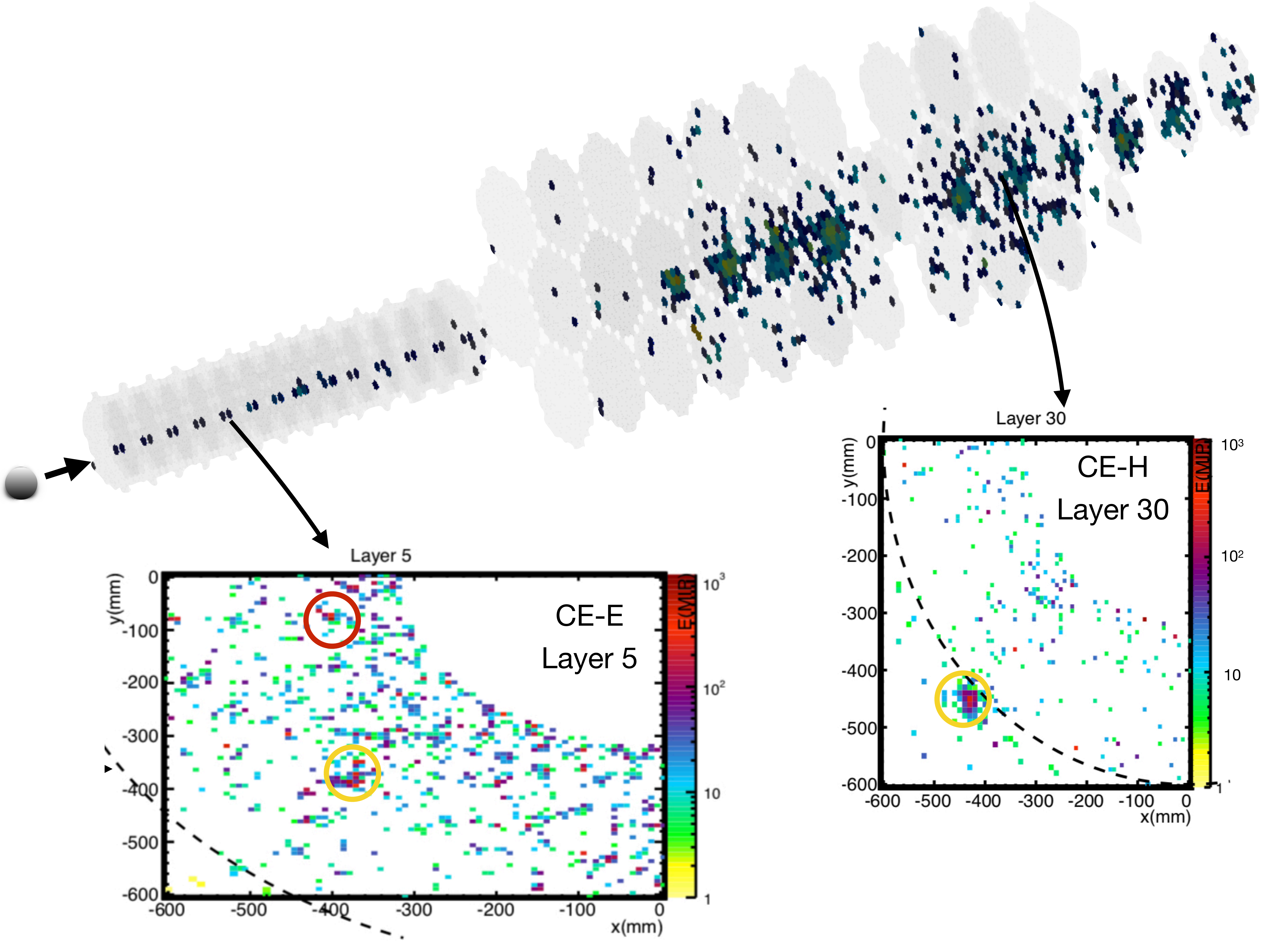
AI inside our detectors

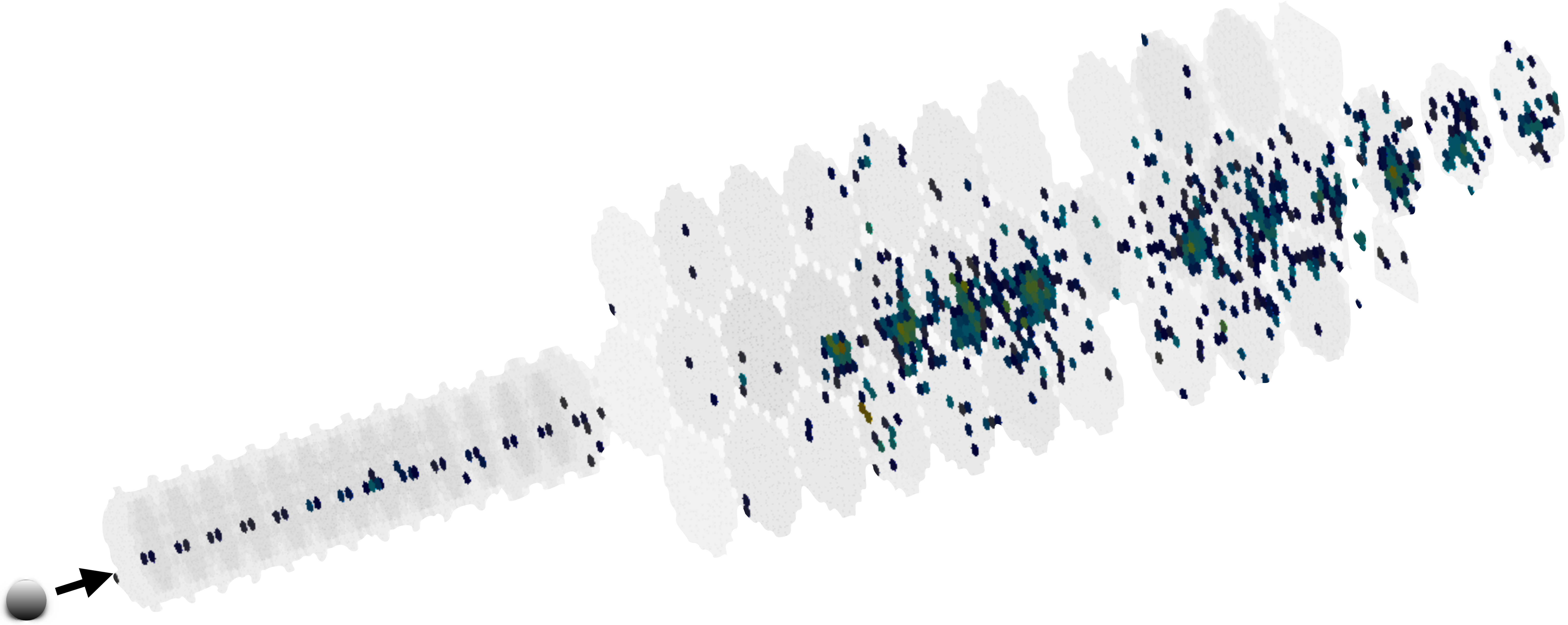
VBF H (γγ)


q
V
V
H₀
γ
γ
q
jet

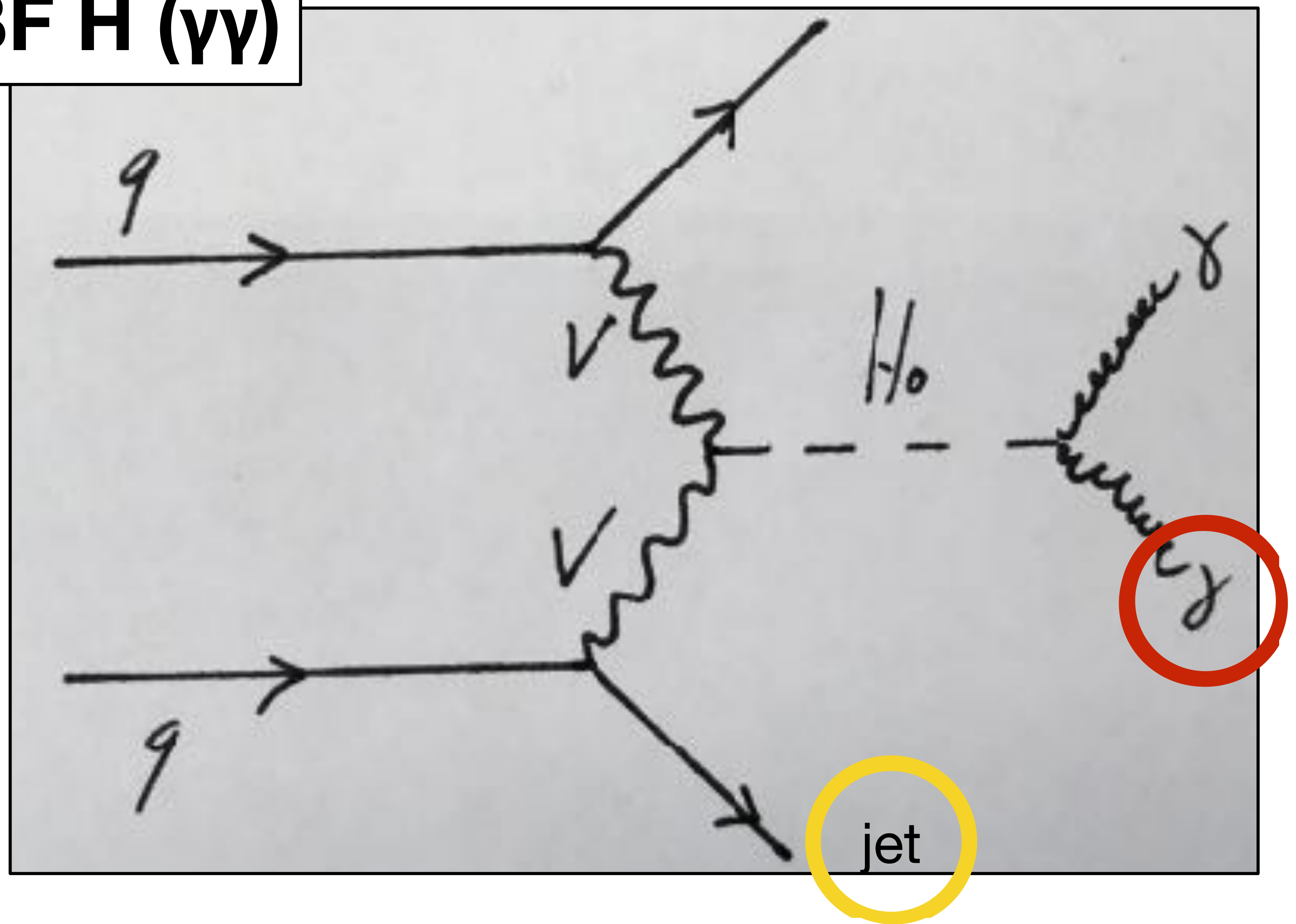
+


200 vertices

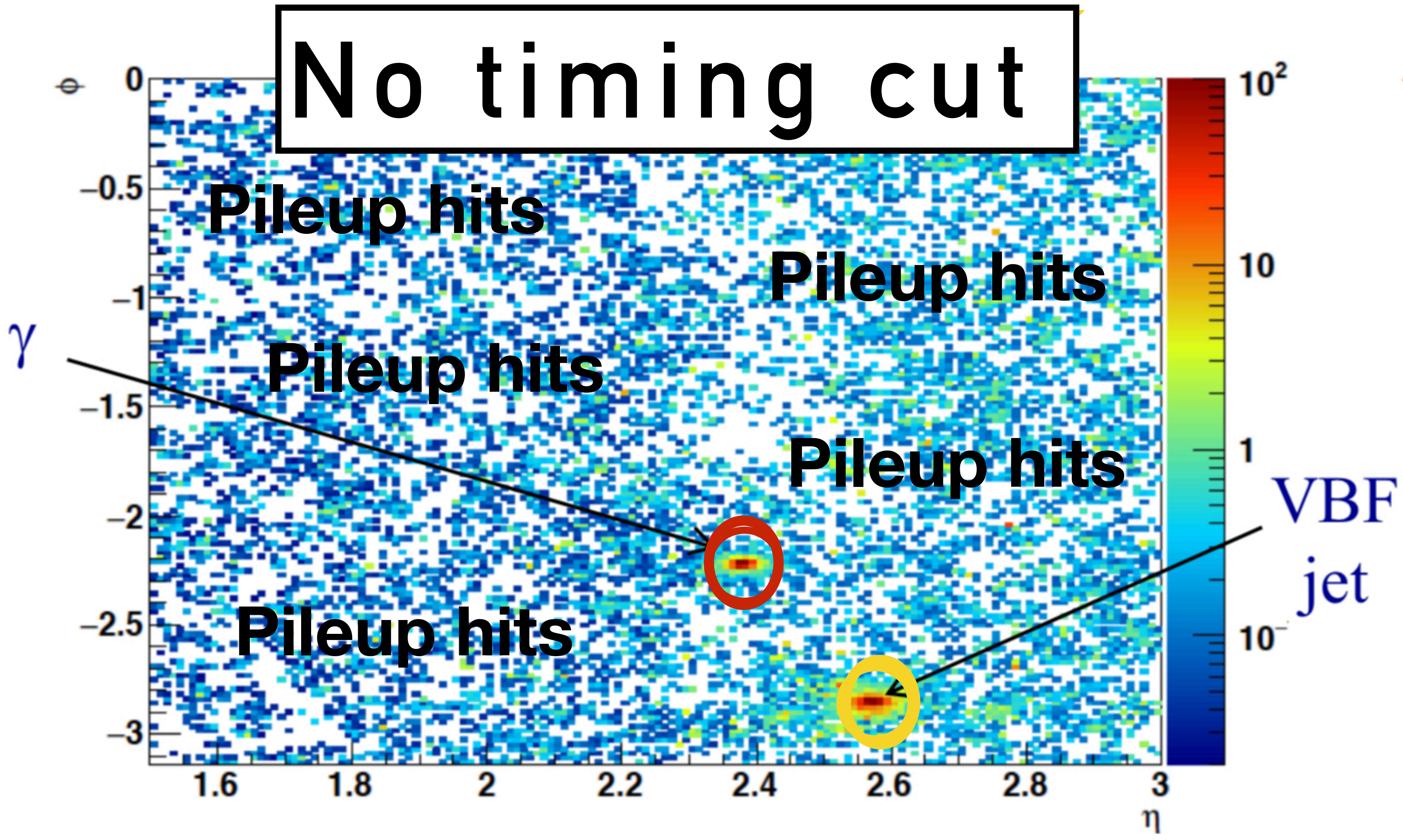

Layer 5
CE-E
Layer 5
y(mm)
0
-100
-200
-300
-400
-500
-600
x(mm)
-600
-500
-400
-300
-200
-100
0
E(MIP)
1
10
10²
10³
Layer 30
CE-H
Layer 30
y(mm)
0
-100
-200
-300
-400
-500
-600
x(mm)
-600
-500
-400
-300
-200
-100
0
E(MIP)
1
10
10²
10³

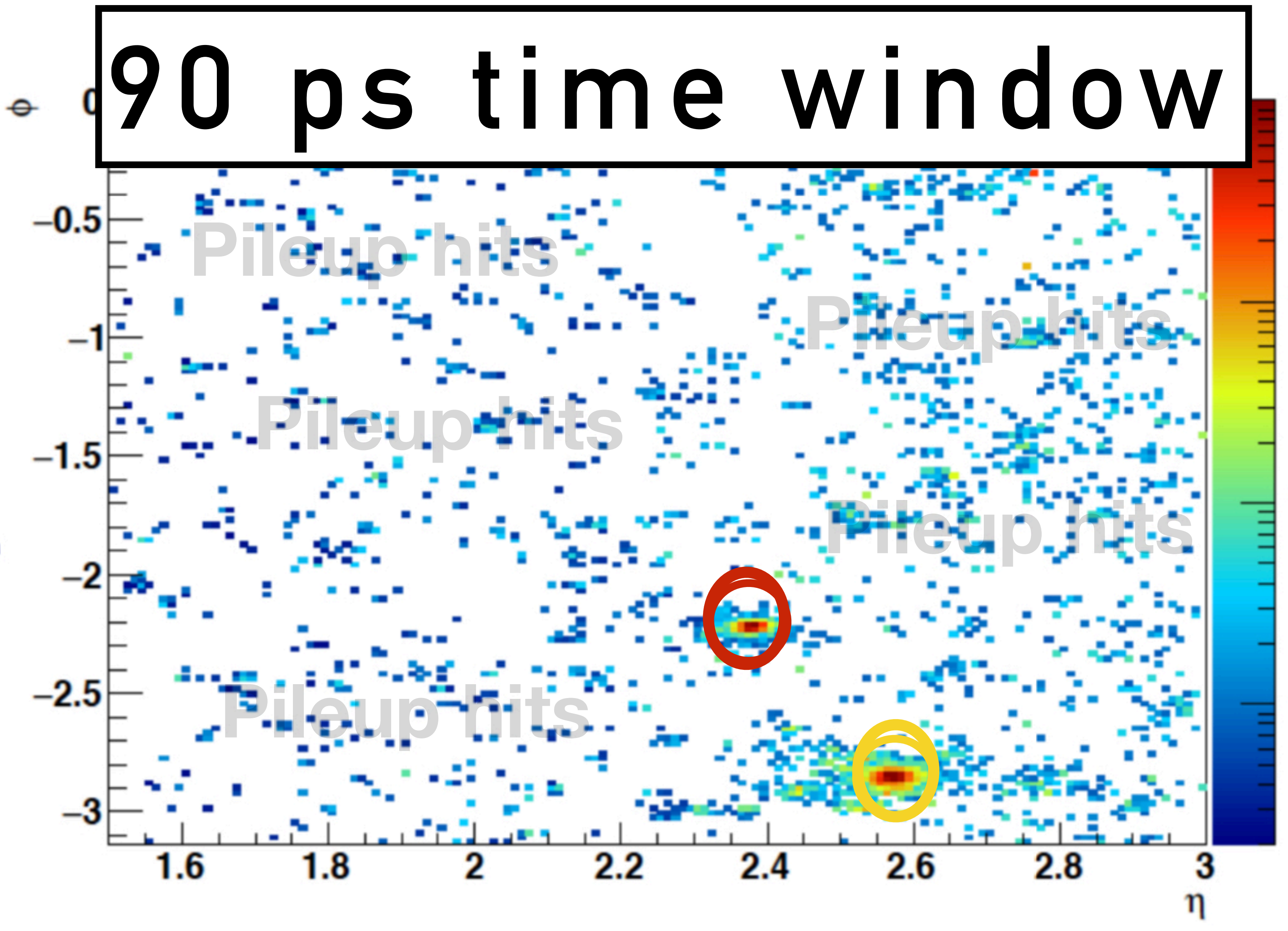


VBF H (γγ)

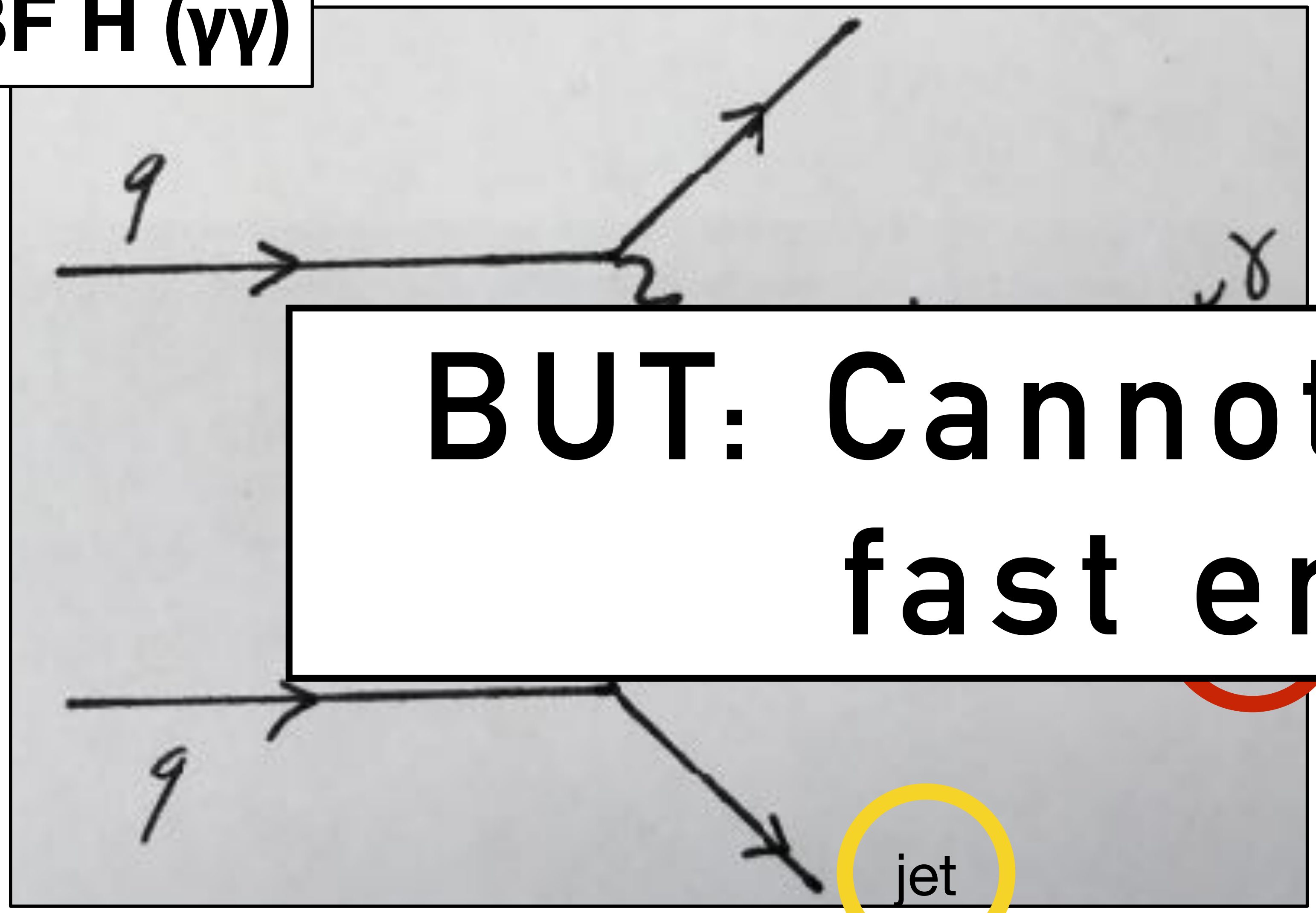


+

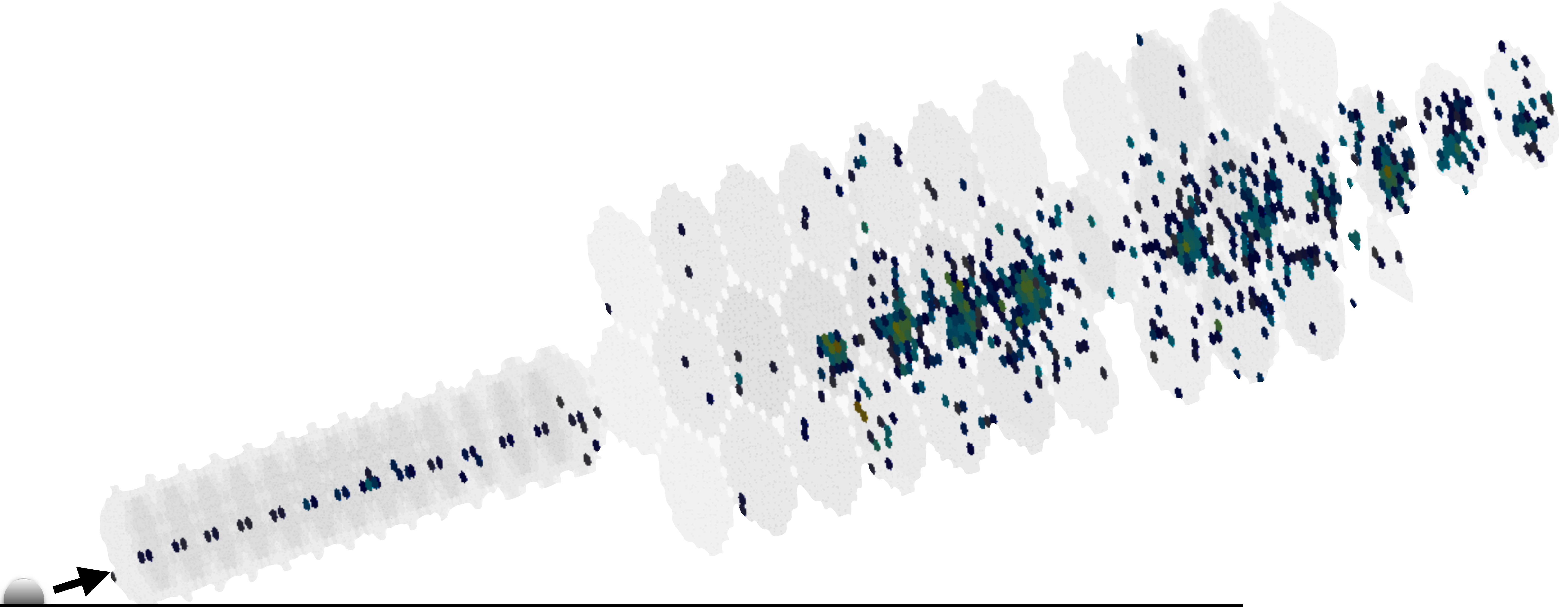

No timing cut


90 ps time window

VBF H (γγ)

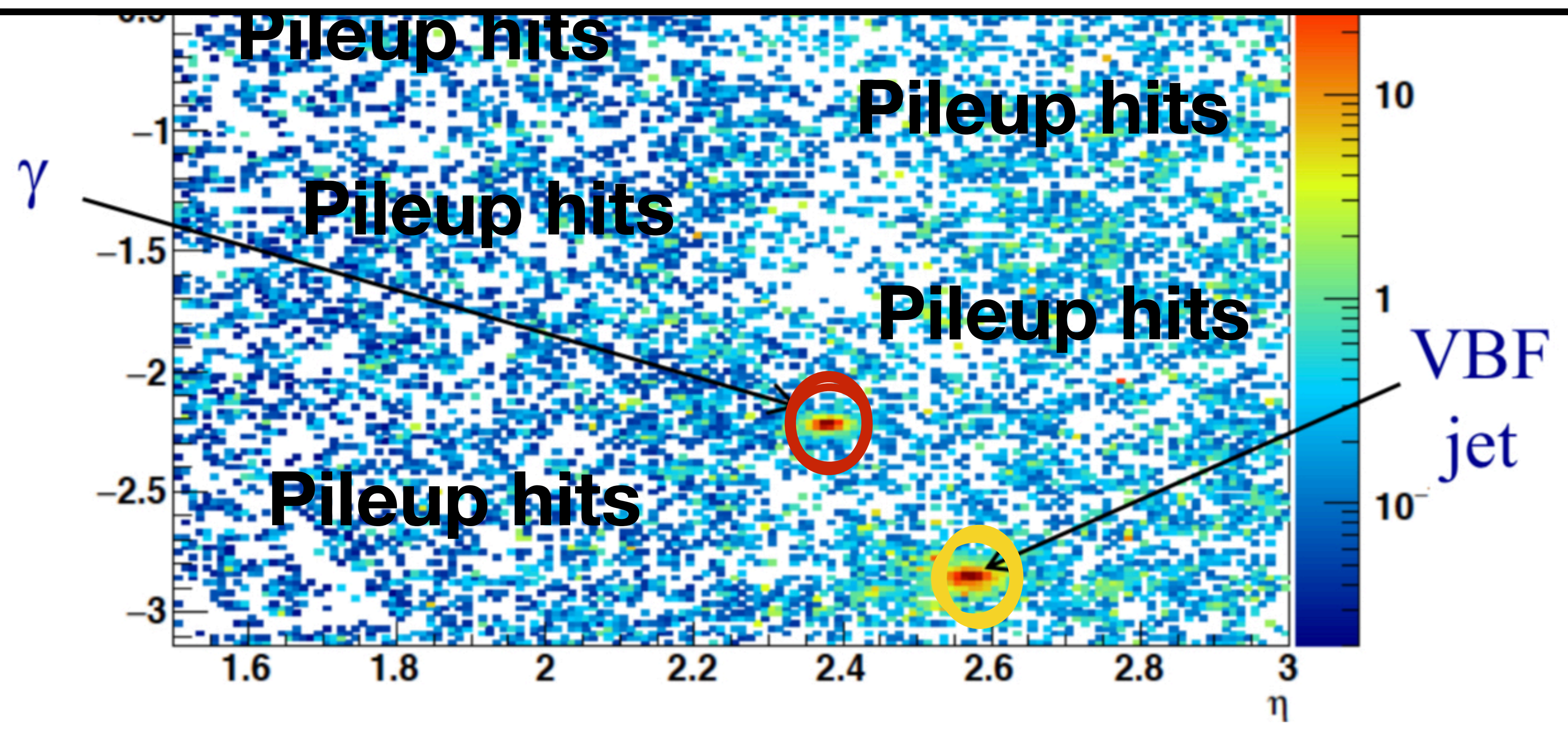


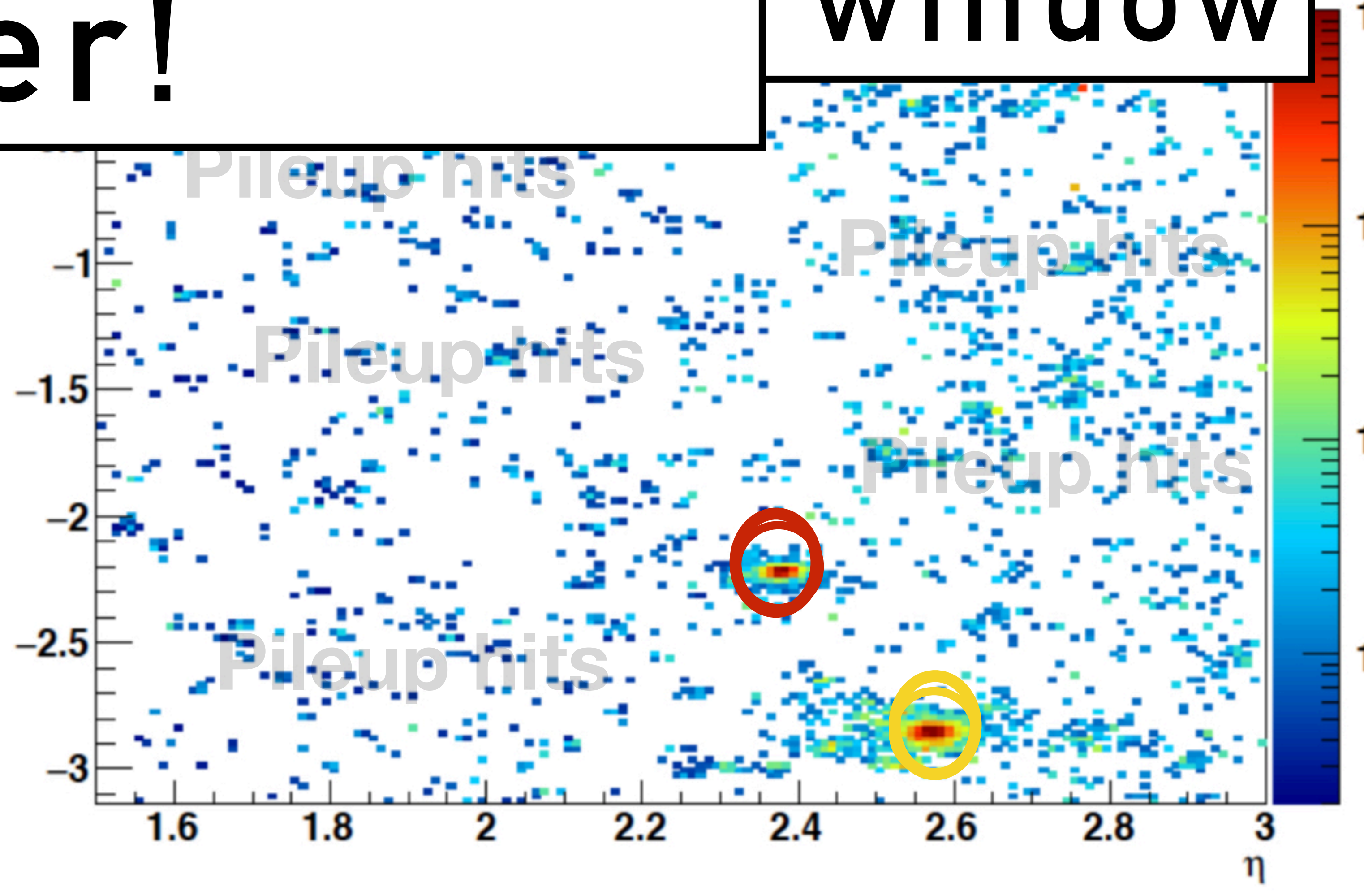
+

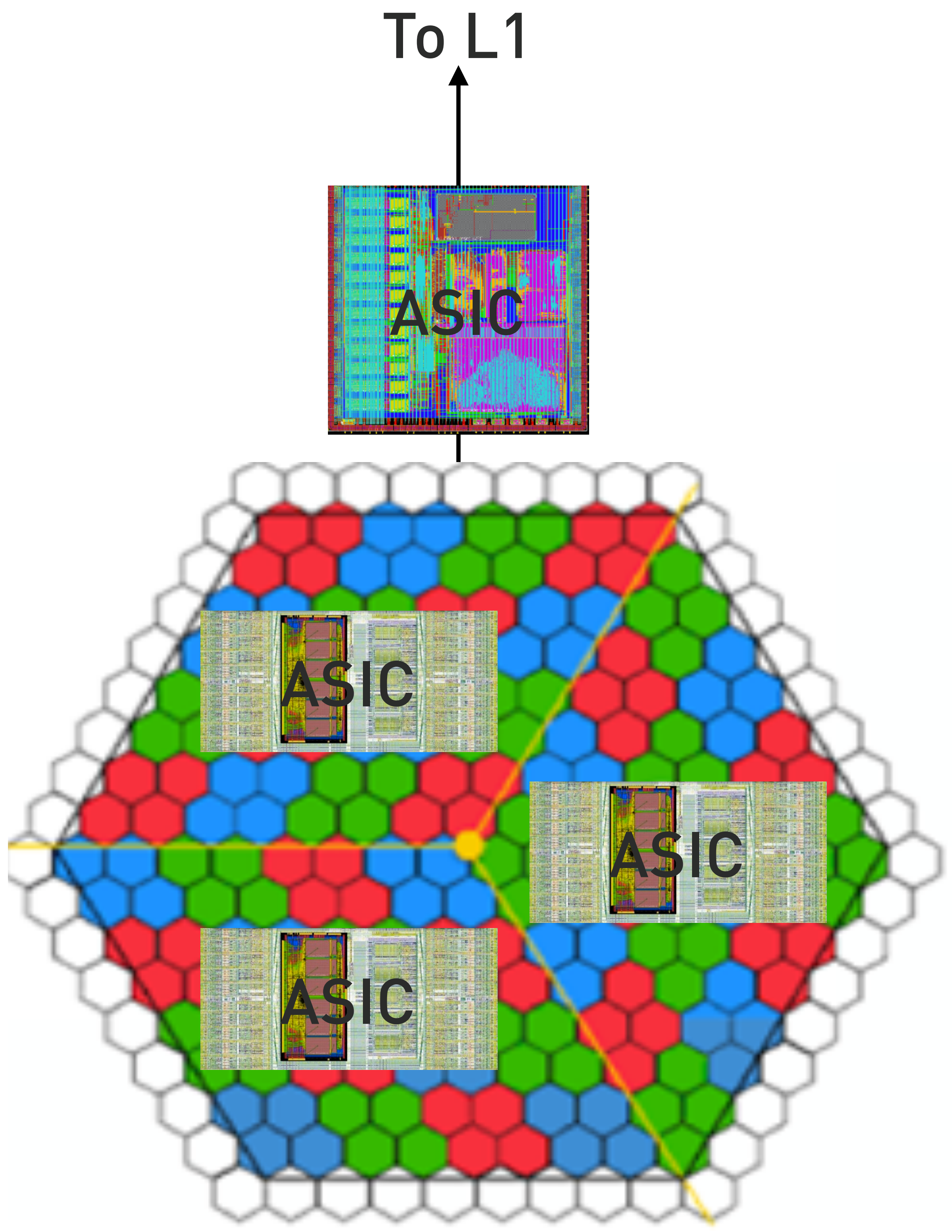


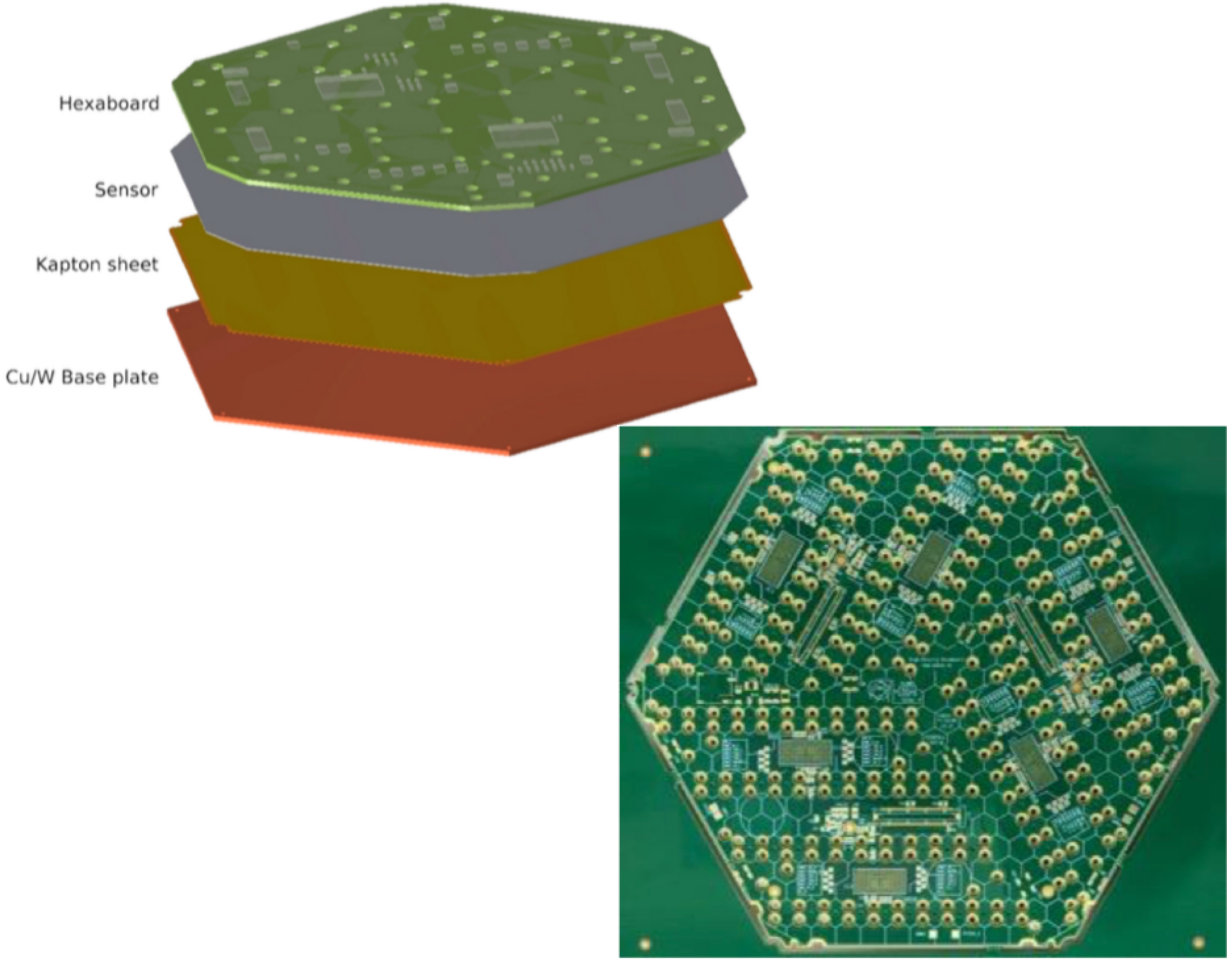
BUT: Cannot read out all these channels fast enough for L1 to trigger!

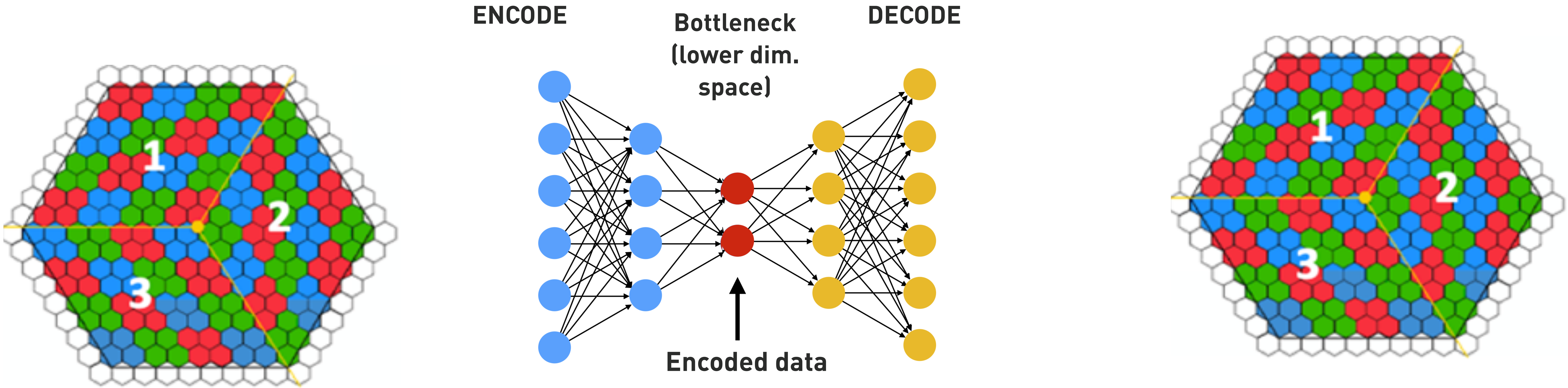
window






To L1
ASIC
ASIC
ASIC
ASIC

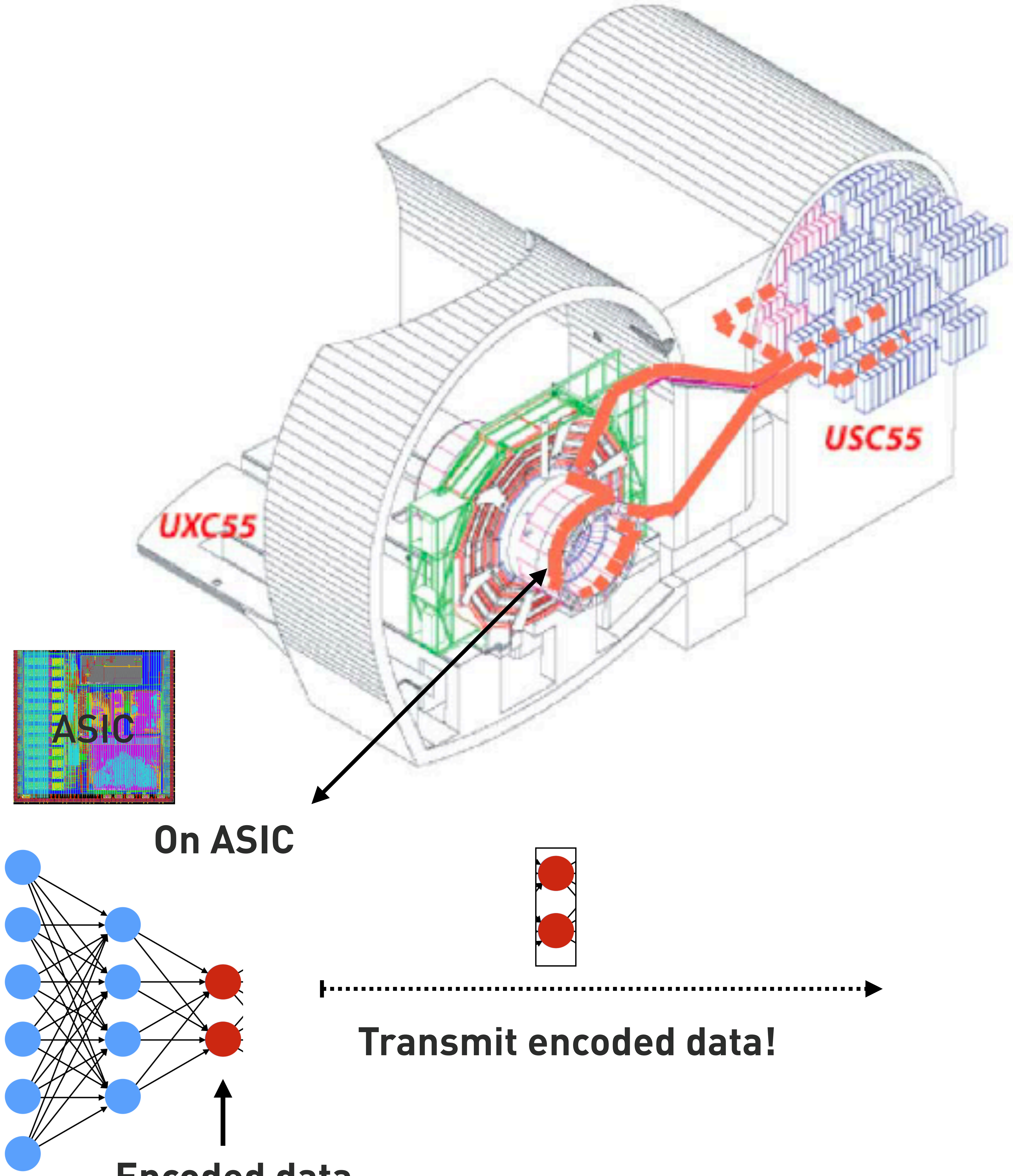

Hexaboard
Sensor
Kapton sheet
Cu/W Base plate



Variational Autoencoder

ECON-T, D. Noonan

AEs for compression also at LHCb!



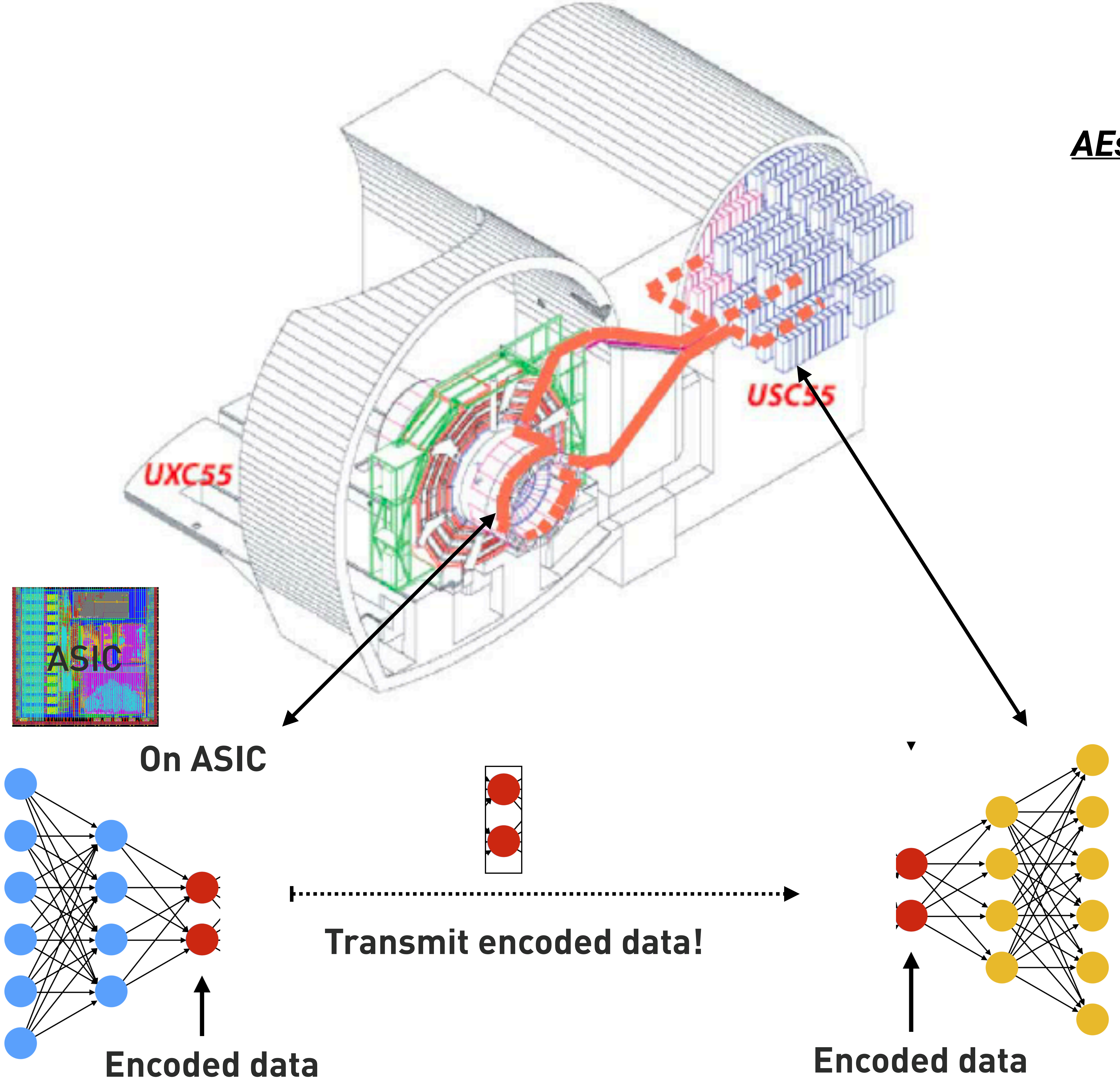
On ASIC

Transmit encoded data!

Encoded data

ECON-T, D. Noonan

AEs for compression also at LHCb!

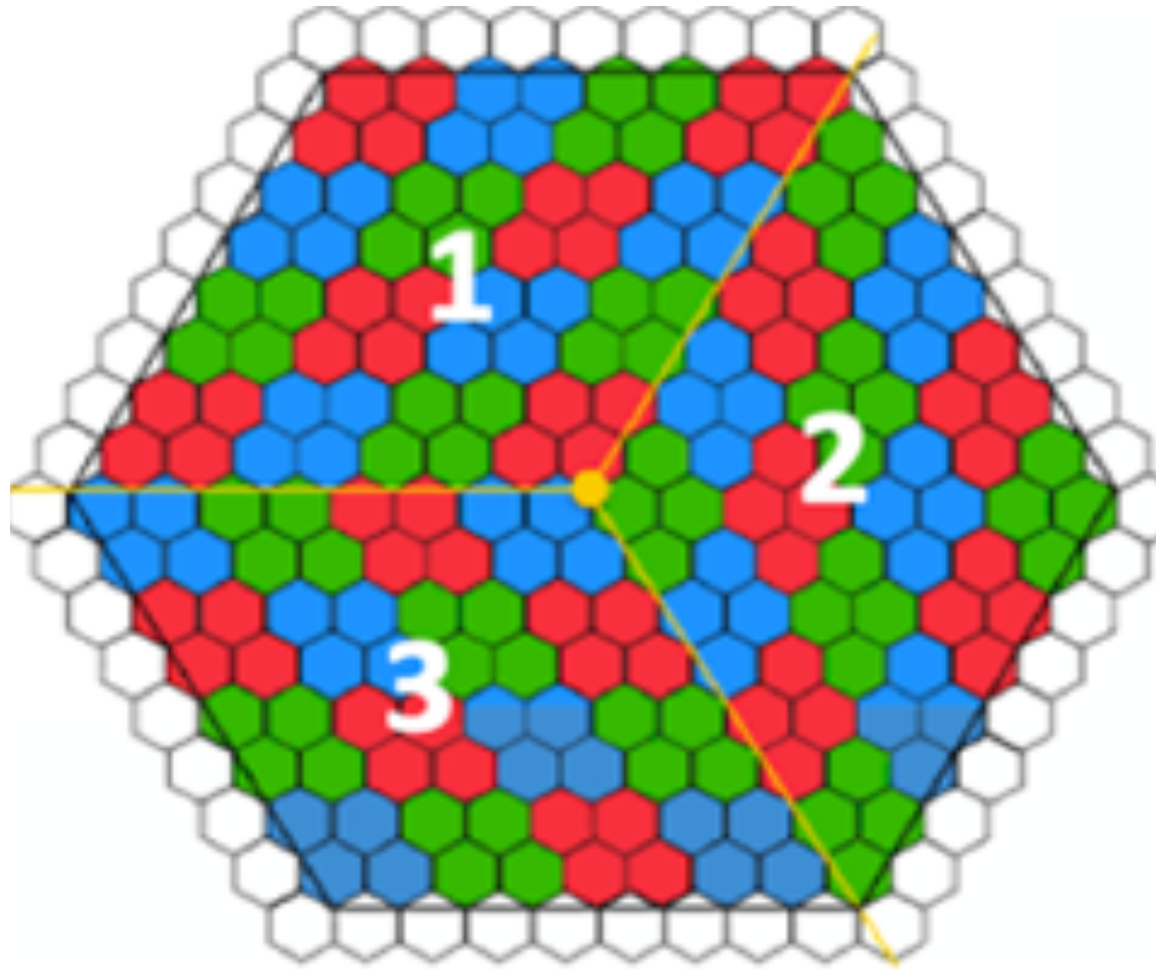


On ASIC

Transmit encoded data!

Encoded data

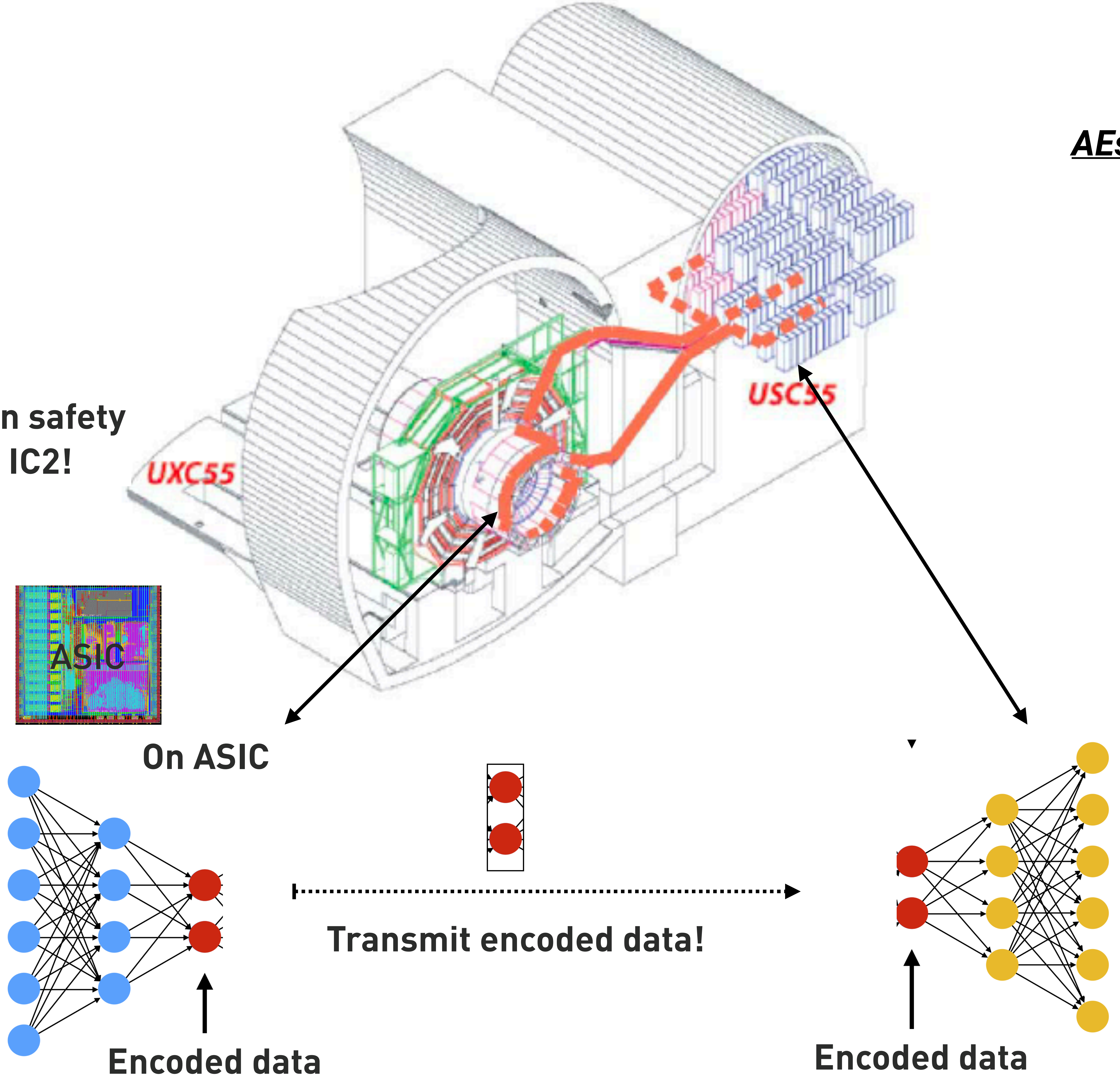
Encoded data

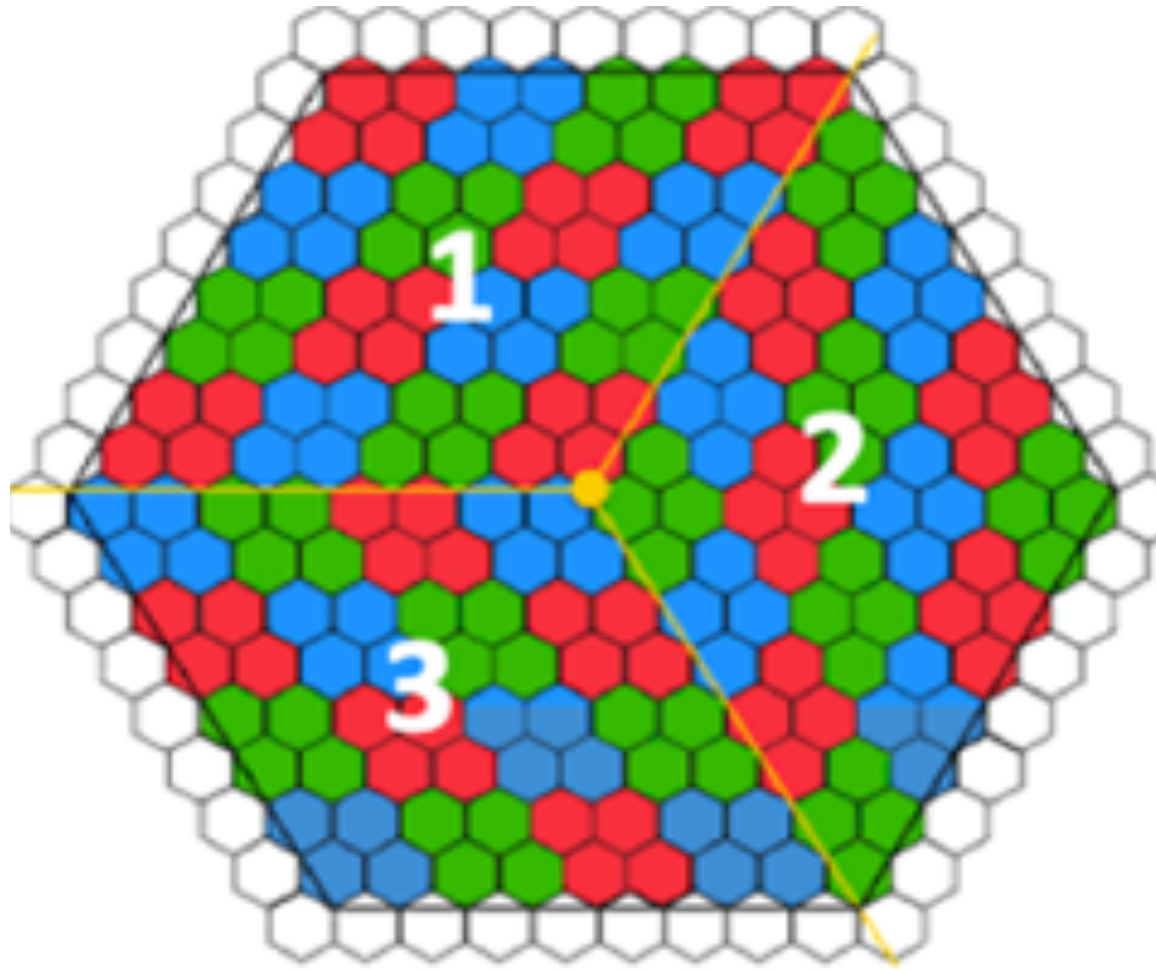


ECON-T, D. Noonan

AEs for compression also at LHCb!

- **75-100 mW**
- **Triplicated w/b for radiation safety**
  Reprogrammable w/b over IC2!





ECON-T, D. Noonan

AEs for compression also at LHCb!

- 75-100 mW
- Triplicated w/b for radiation safety
  Reprogrammable w/b over IC2!

On ASIC

Transmit encoded data!

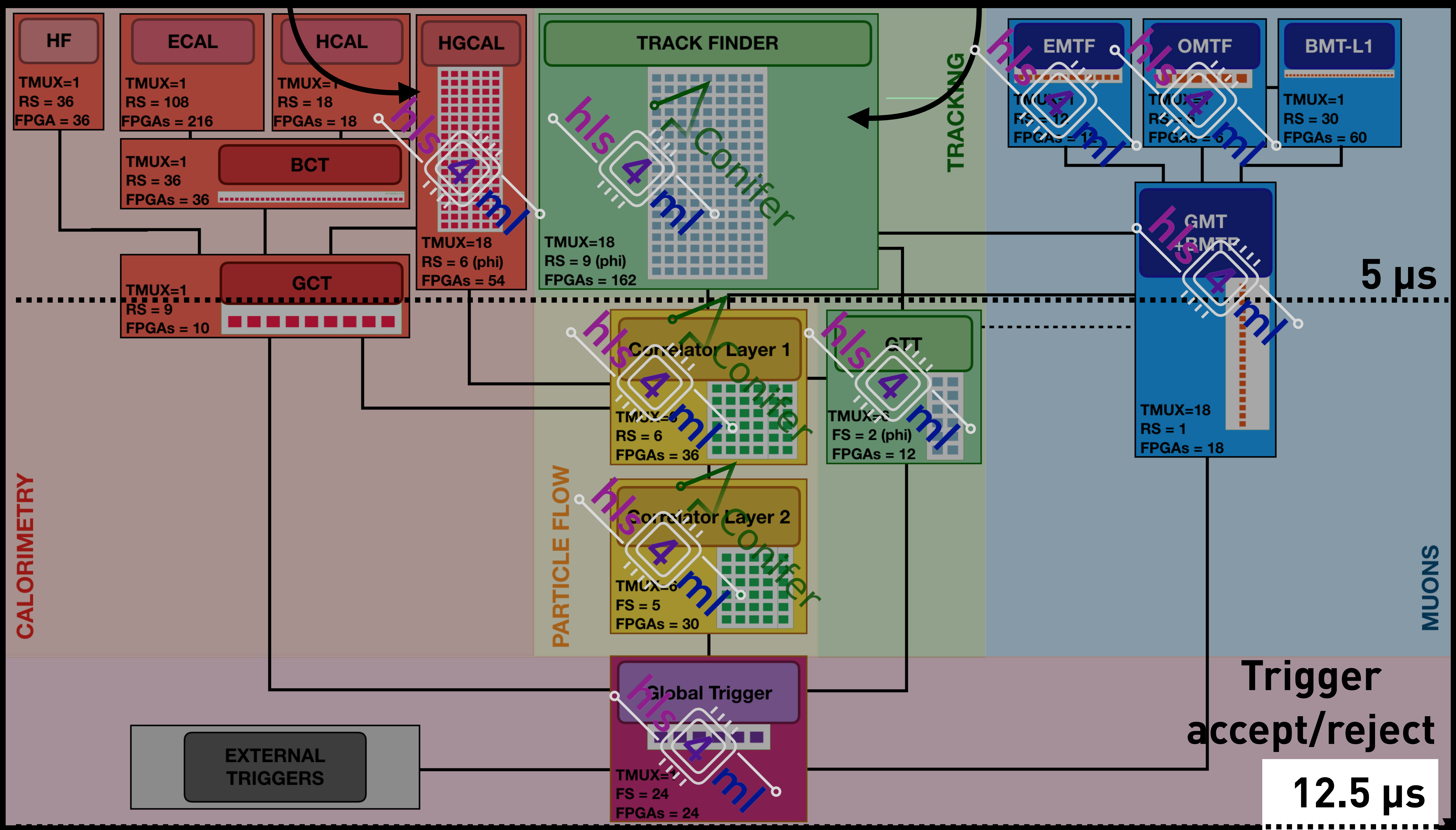
Encoded data

Encoded data

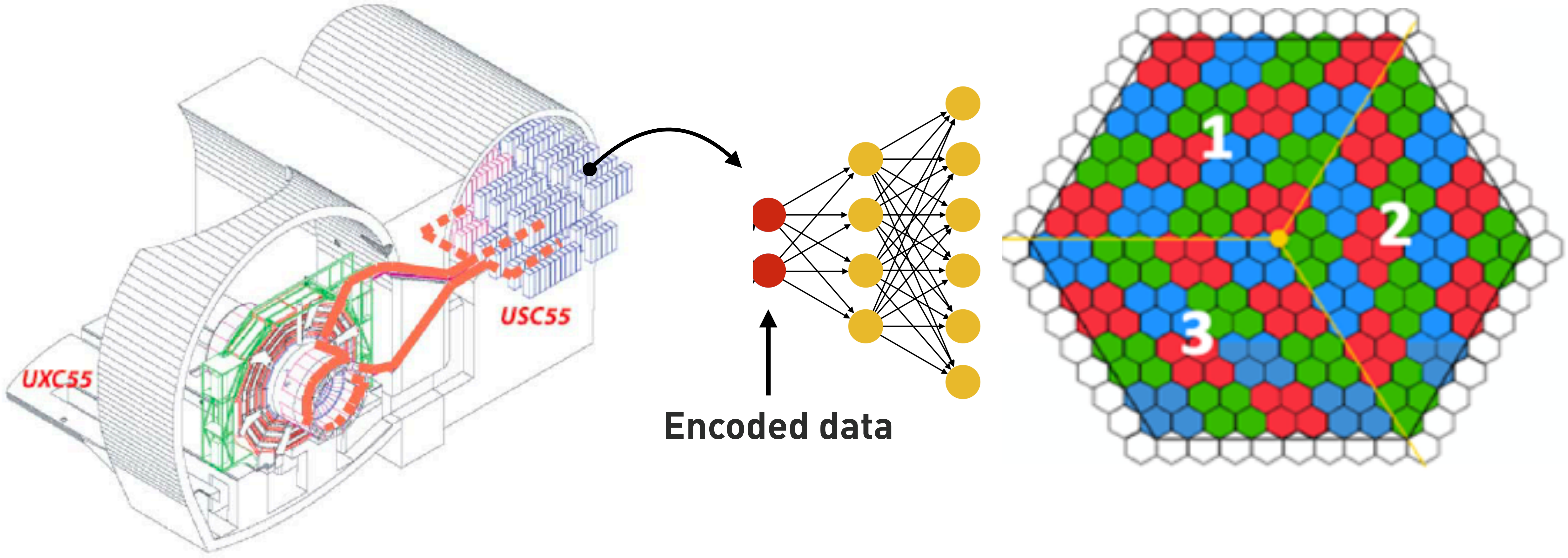
Nanosecond ML inference on FPGAs!

40 billion inferences/s during HL-LHC

HEP developed libraries for fast ML on FPGAs



ML for reconstruction



On FPGA

Encoded data

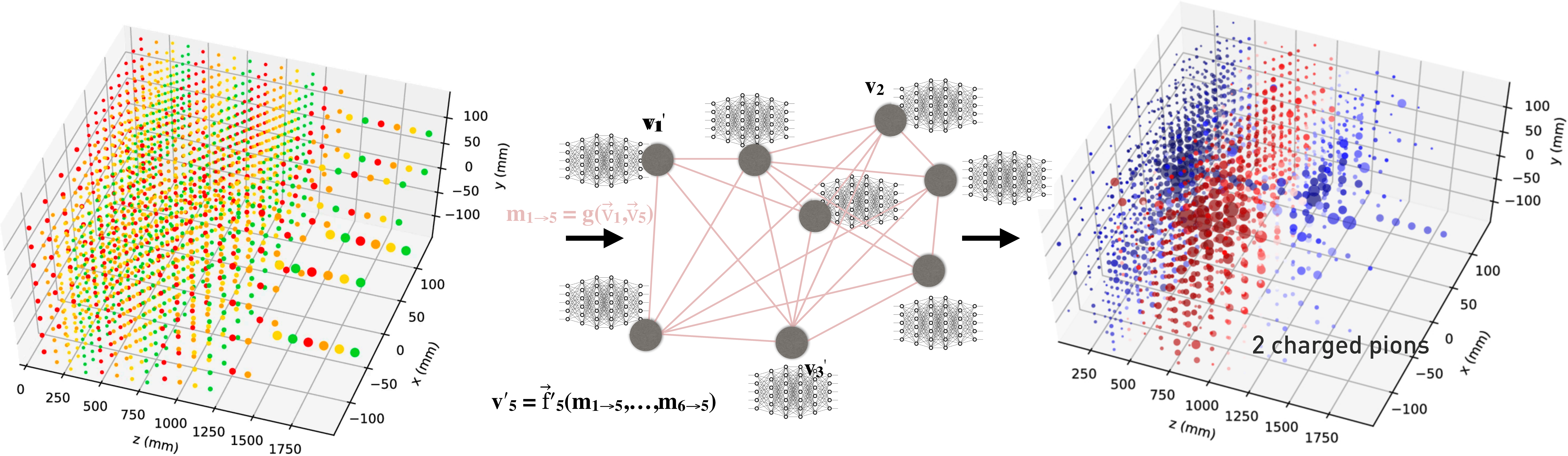
ML for reconstruction

On FPGA

Encoded data

EPJC Vol 79 608 (2019)

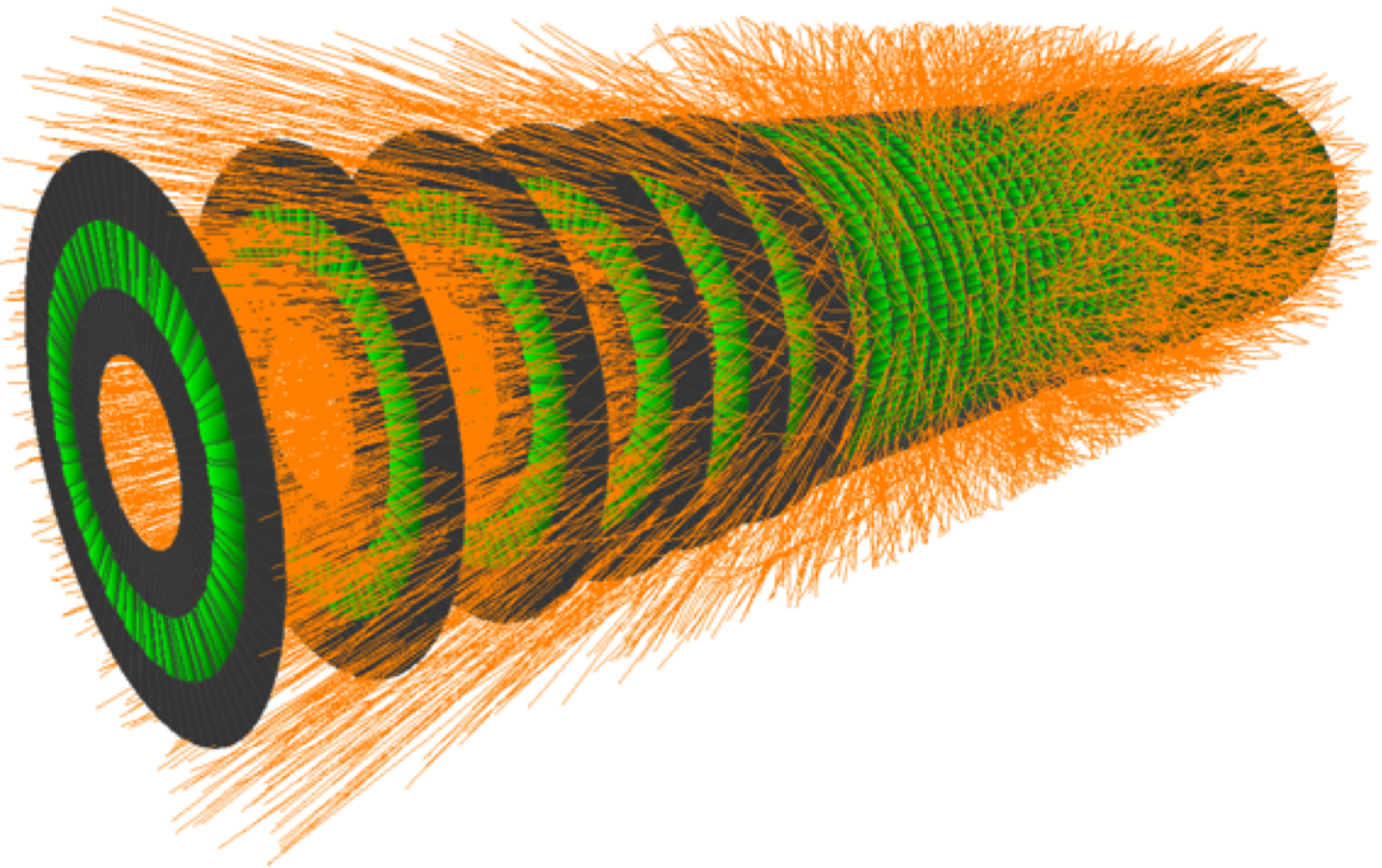
ML for reconstruction



On FPGA: 3.5 μs to cluster energy deposits into disentangled showers from individual particles

DOI:10.3389/fdata.2022.828666

ML 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

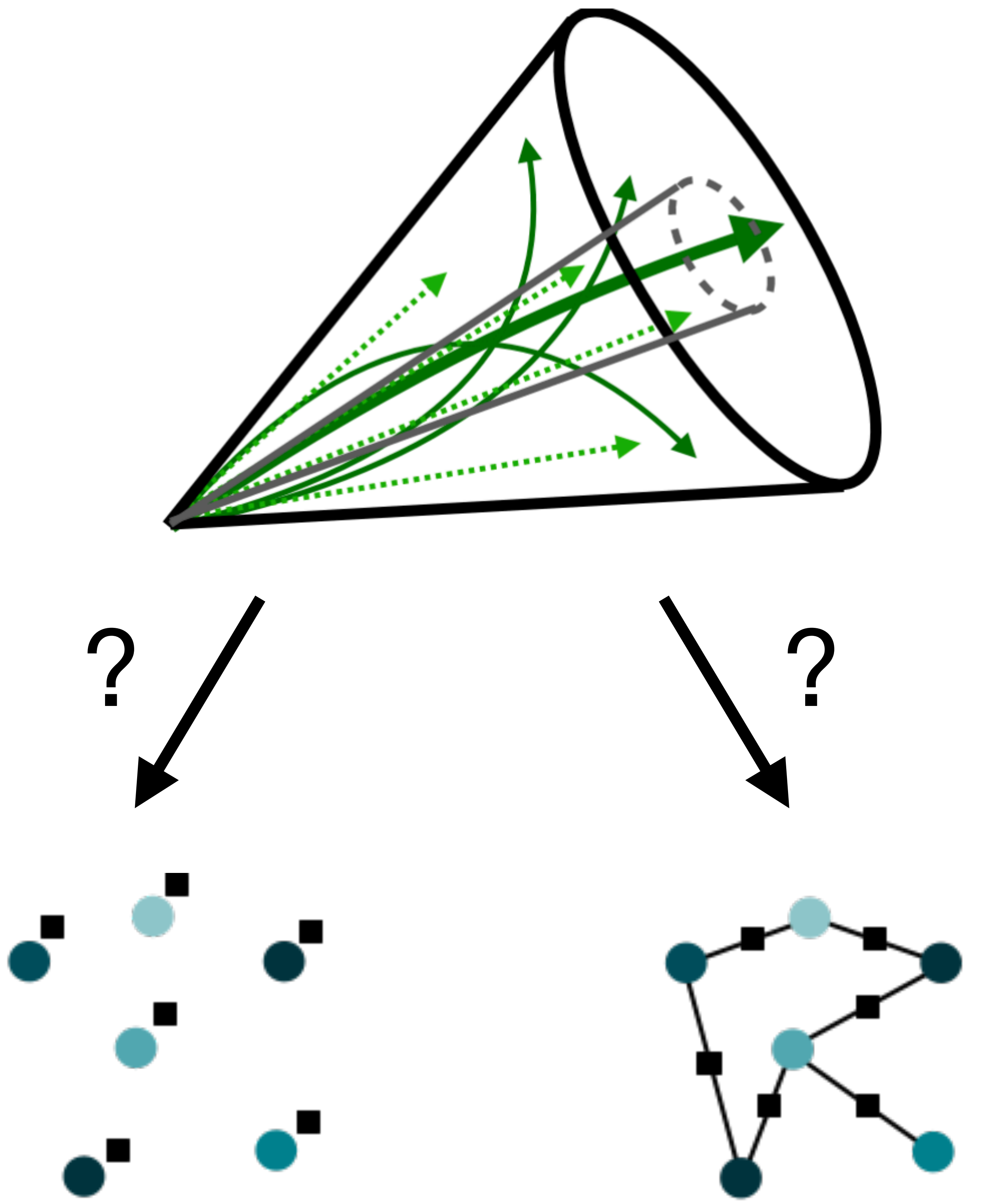
Graph Neural Networks for fast charged particle tracking

Throughput-optimized for L1 applications,
resource-optimised for co-processing

Design	(n_{nodes}, n_{edges})	RF	Precision	Latency [cycles]	II [cycles]	DSP [%]	LUT [%]	FF [%]	BRAM [%]
Throughput-opt.	(28, 56)	1	`ap_fixed<14,7>`	59 295 ns	1	99.9	66.0	11.7	0.7
Resource-opt.	(28, 56)	1	`ap_fixed<14,7>`	79 395 ns	28	56.6	17.6	3.9	13.1

cds.cern.ch/record/2814728/

Fast jet tagging

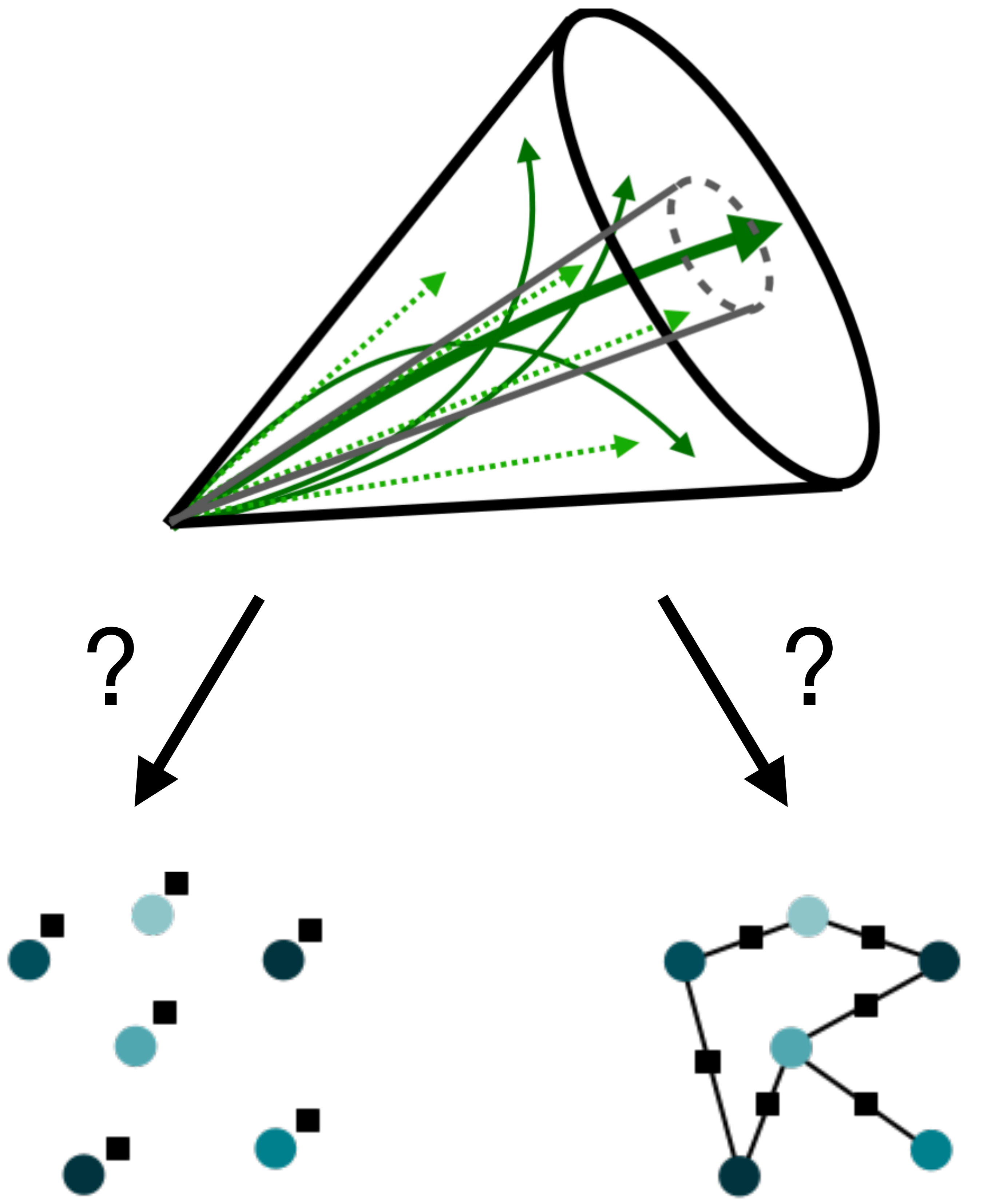


Sets: Information is only assigned to individual nodes.

Graphs: Information is assigned to edges, i.e., pairs of nodes.

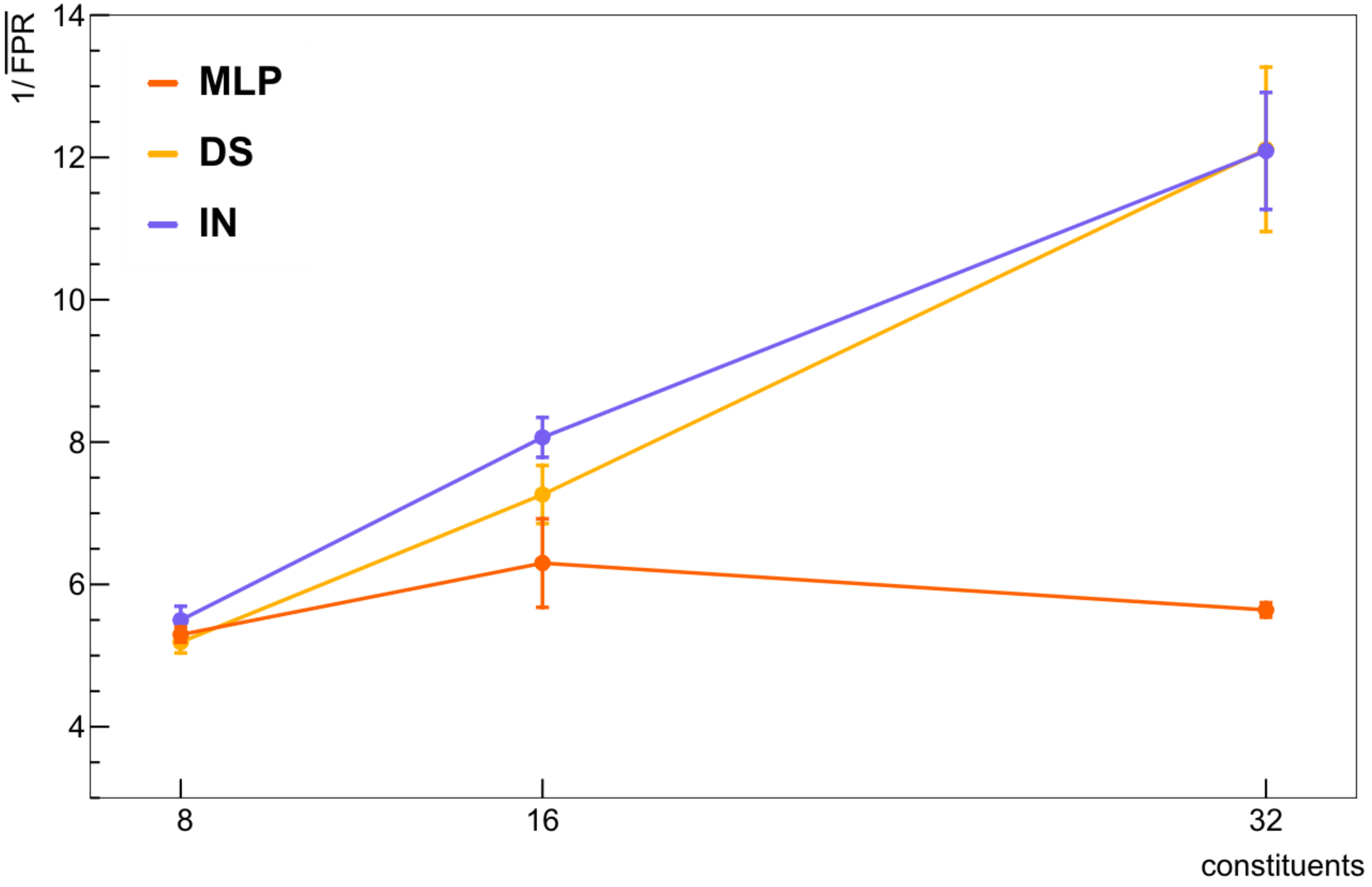
cds.cern.ch/record/2814728/

Fast jet tagging



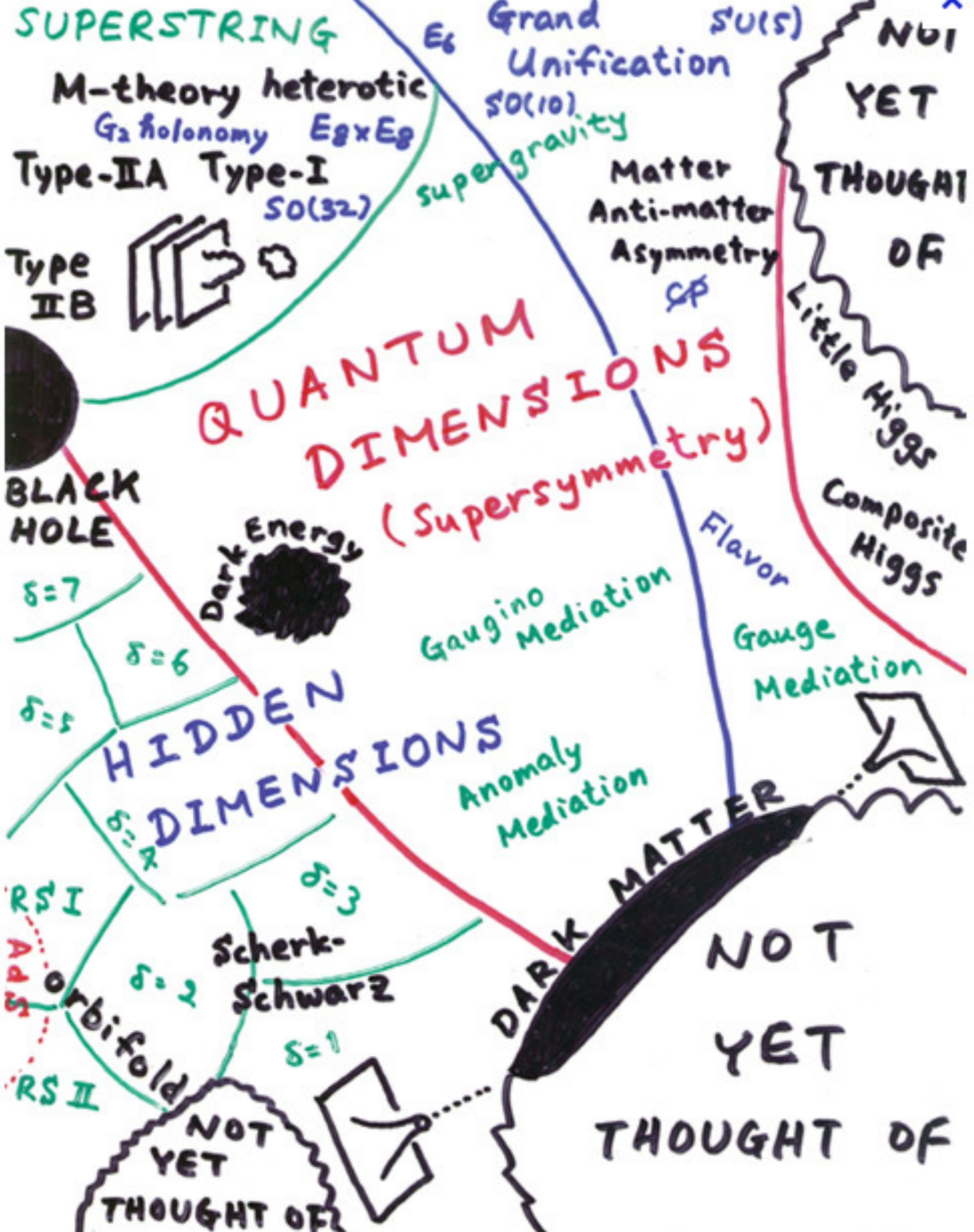
Sets: Information is only assigned to individual nodes.

Graphs: Information is assigned to edges, i.e., pairs of nodes.



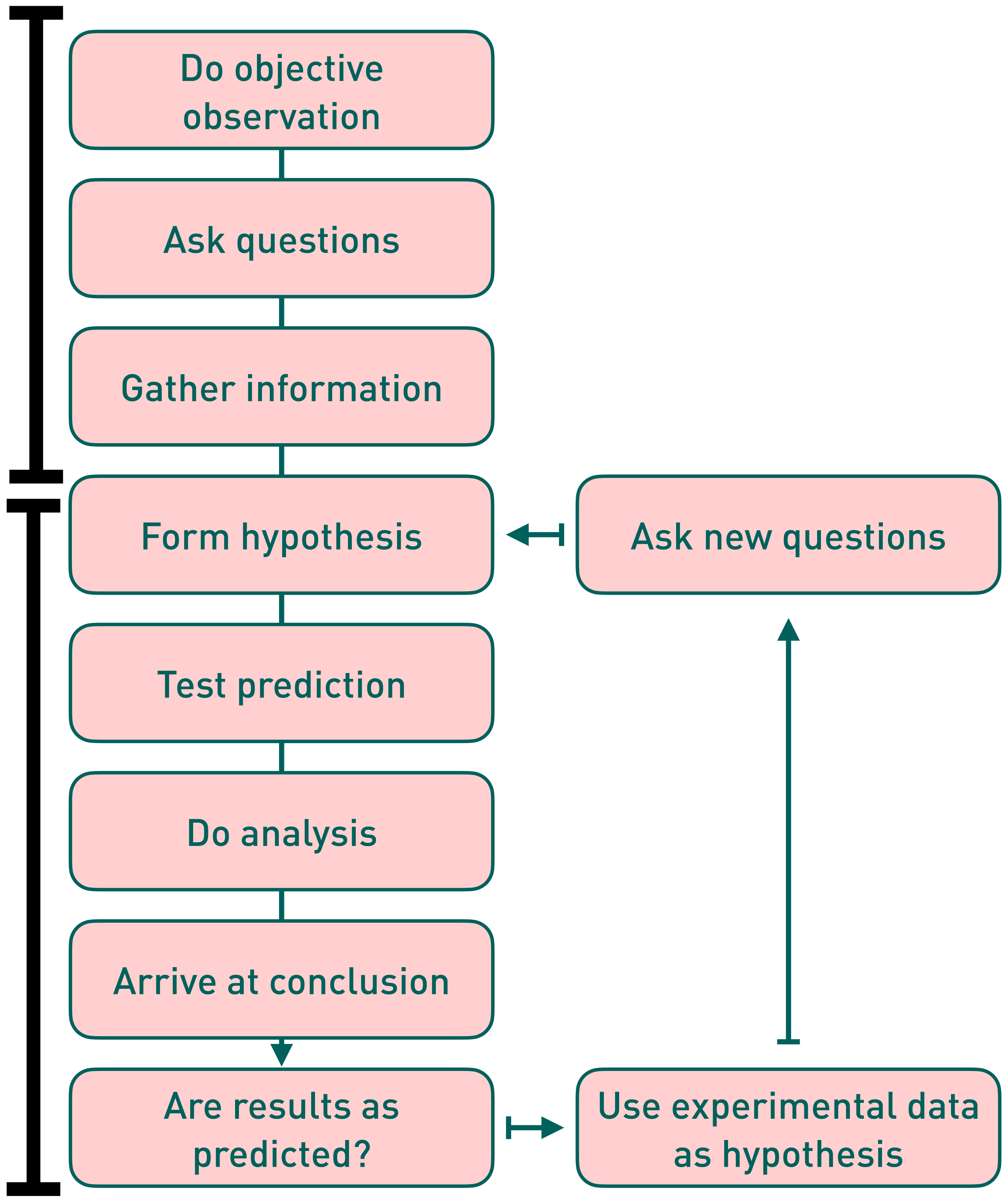
(Can also do 90 ns transformers for jet tagging!)


CMS Preliminary LHCP 2017
Vector mediator
Dark matter mass m_{DM} [GeV]
Mediator mass M_{med} [GeV]
EXCLUDED
(a) CMS
DM Simplified Model Exclusions ATLAS Preliminary March 2017
35.9 fb⁻¹ (13 TeV)
CMS
Vector med, Dirac DM, g_q = 0.25, g_{DM} = 1
Axial med, Dirac DM, g_q = 0.25, g_{DM} = 1
EXCLUDED
EXCLUDED
(a) CMS
ATLAS
EXCLUDED
EXCLUDED
(b) ATLAS
CMS monojet
2016 diboson combination
CMS
$m_{W'}$ (GeV)
EXCLUDED
CMS Preliminary LHCP 2020
Axial-vector mediator
Dirac DM
EXCLUDED
ATLAS
EXCLUDED
EXCLUDED
EXCLUDED
(a) CMS
CMS Preliminary
Baryonic Z'
EXCLUDED
CMS Preliminary
EXCLUDED
(a) CMS
Fall 2020 132-140 fb⁻¹ (13 TeV)
CMS Preliminary
$c\tau_0$ (mm)
EXCLUDED
Pseudoscalar, Dirac, g_q=1, g_χ=1
EXCLUDED
EXCLUDED
ATLAS
Axial-vector, Dirac, g_q = 0.25, g_χ = 1.0
m_{med} [GeV]
EXCLUDED
(b) ATLAS
ATLAS
EXCLUDED
ATLAS
EXCLUDED
CMS
Z' mass (GeV)
EXCLUDED
(a) CMS
ATLAS
EXCLUDED
(b) ATLAS
CMS Preliminary LHCP 2020
$Z' \to q\bar{q}$
$M_{Z'}$ [GeV]
EXCLUDED


SUPERSTRING
M-theory
heterotic
G2 holonomy
E8×E8
Type-IIA
Type-I
SO(32)
Type IIB
E6
Grand Unification
SU(5)
SO(10)
supergravity
Matter Anti-matter Asymmetry
CP
NOT YET THOUGHT OF
Little Higgs
Composite Higgs
BLACK HOLE
QUANTUM DIMENSIONS (Supersymmetry)
Dark Energy
Flavor
Gaugino Mediation
Gauge Mediation
δ=7
δ=6
δ=5
δ=4
δ=3
δ=2
δ=1
HIDDEN DIMENSIONS
Anomaly Mediation
DARK MATTER
RS I
AdS5
orbifold
RS II
Scherk-Schwarz
NOT YET THOUGHT OF
NOT YET THOUGHT OF

ARE WE LOOKING IN THE WRONG WAY?

The scientific method



Look at nature

Form and test hypothesis

Do objective observation

Ask questions

Gather information

Form hypothesis

Test prediction

Do analysis

Arrive at conclusion

Are results as predicted?

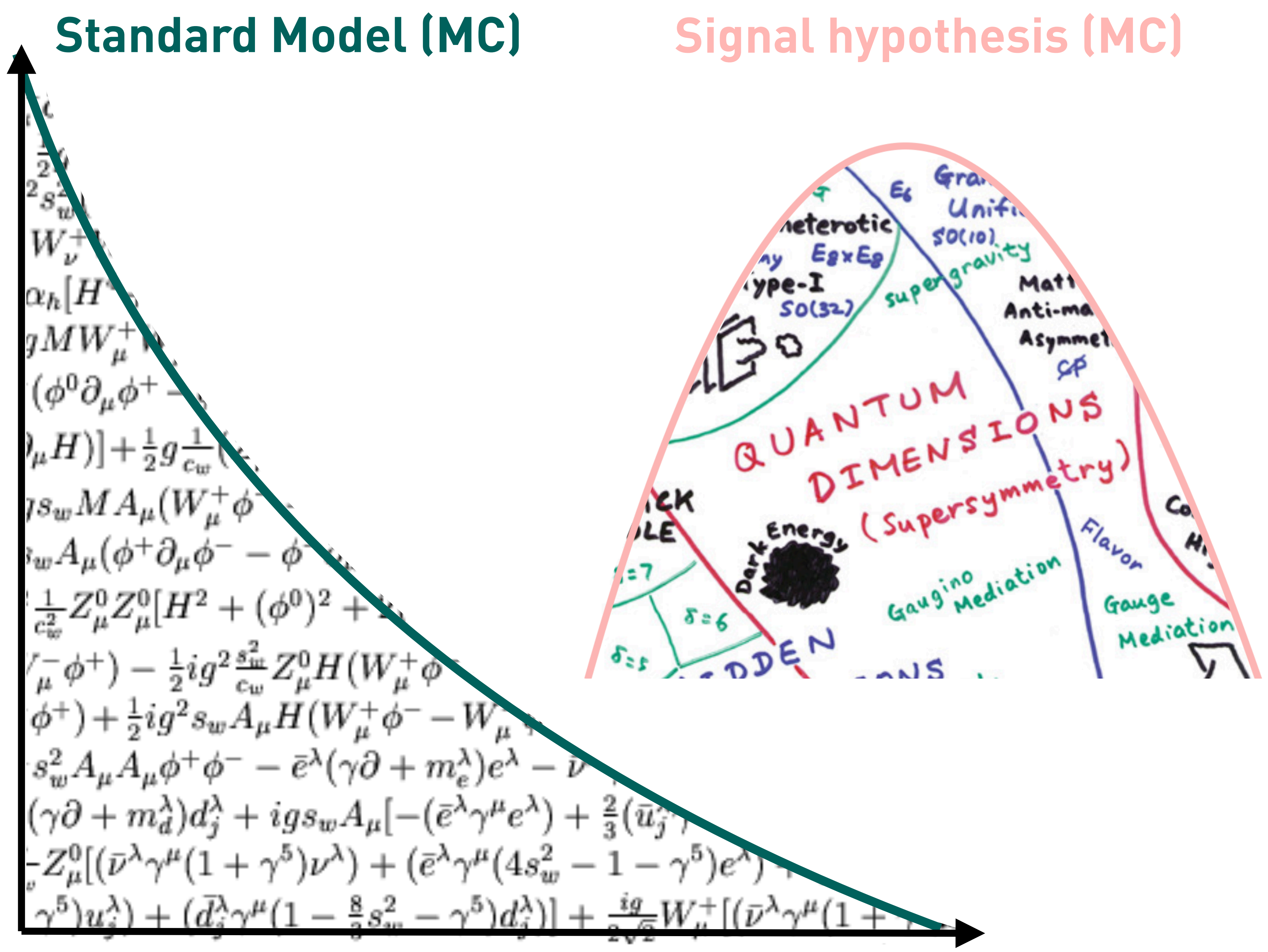
Use experimental data as hypothesis

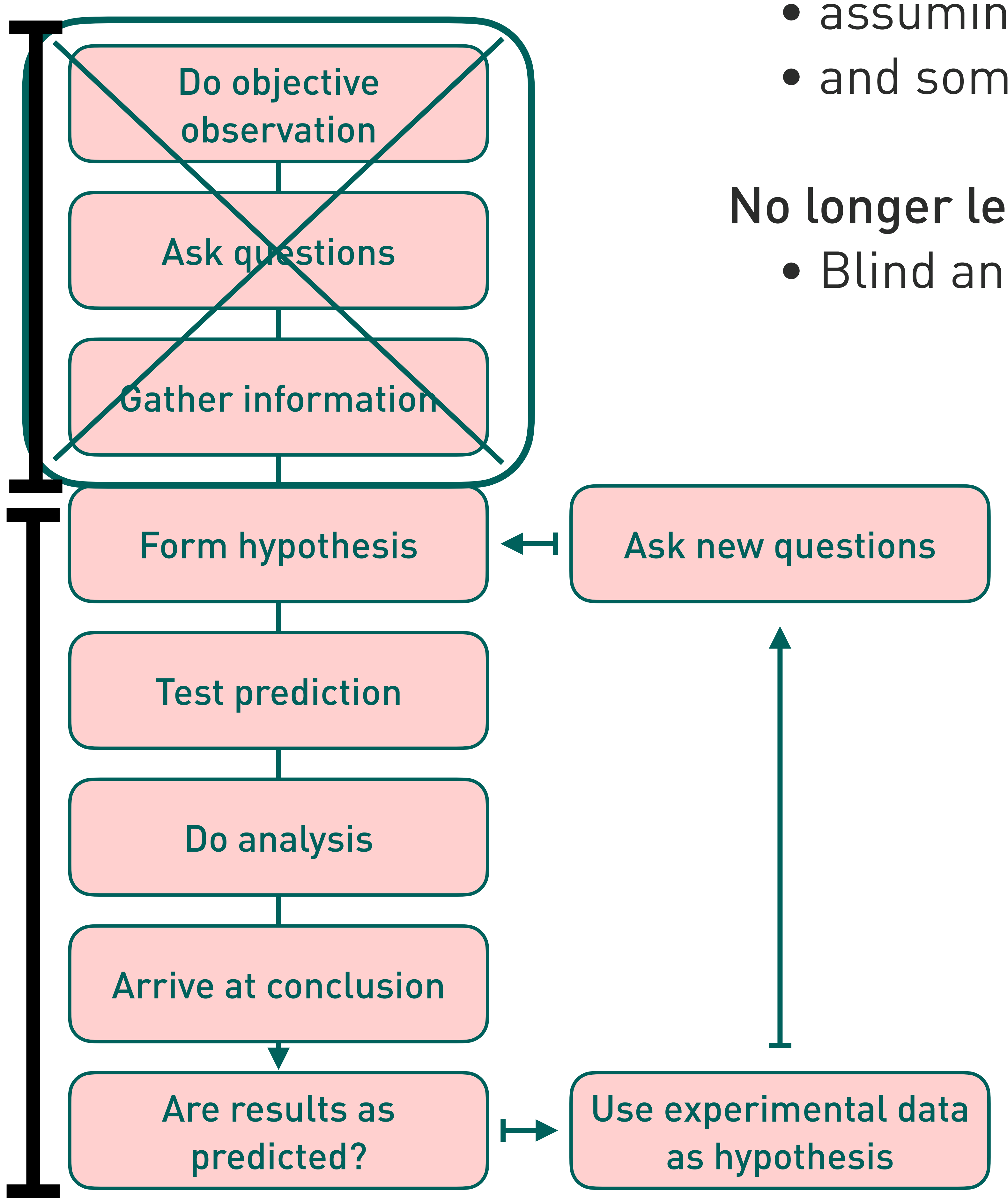
Ask new questions

Searches at LHC

Whatever you do,
don’t look at nature

Replaced by: ←




~~Do objective observation~~

~~Ask questions~~

~~Gather information~~

Form hypothesis

Test prediction

Do analysis

Arrive at conclusion

Are results as predicted?

Use experimental data as hypothesis

Ask new questions

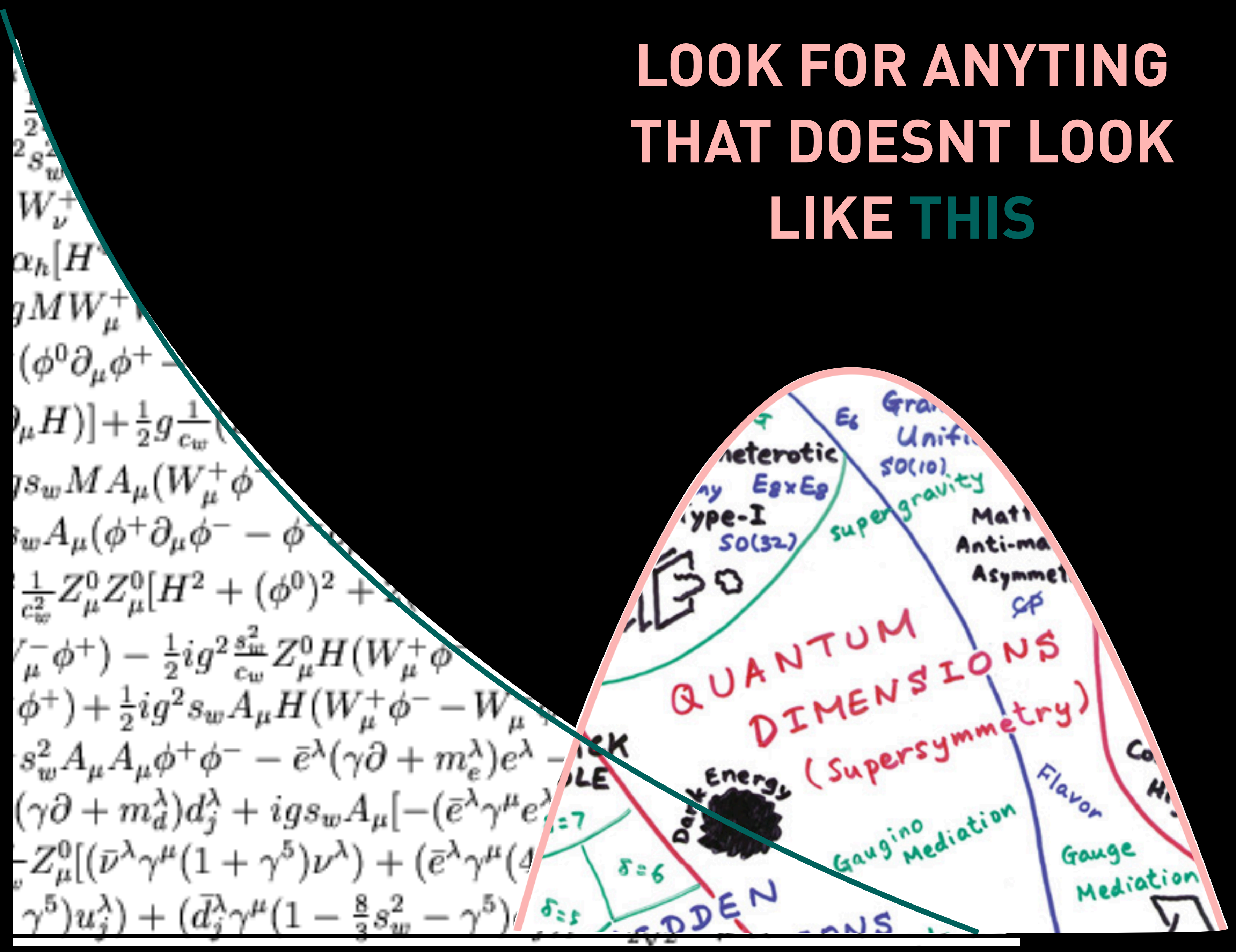
Searches at LHC (almost) always start with by

- assuming Standard Model
- and some signal hypothesis

No longer learn from observation

- Blind analysis only way we perform searches

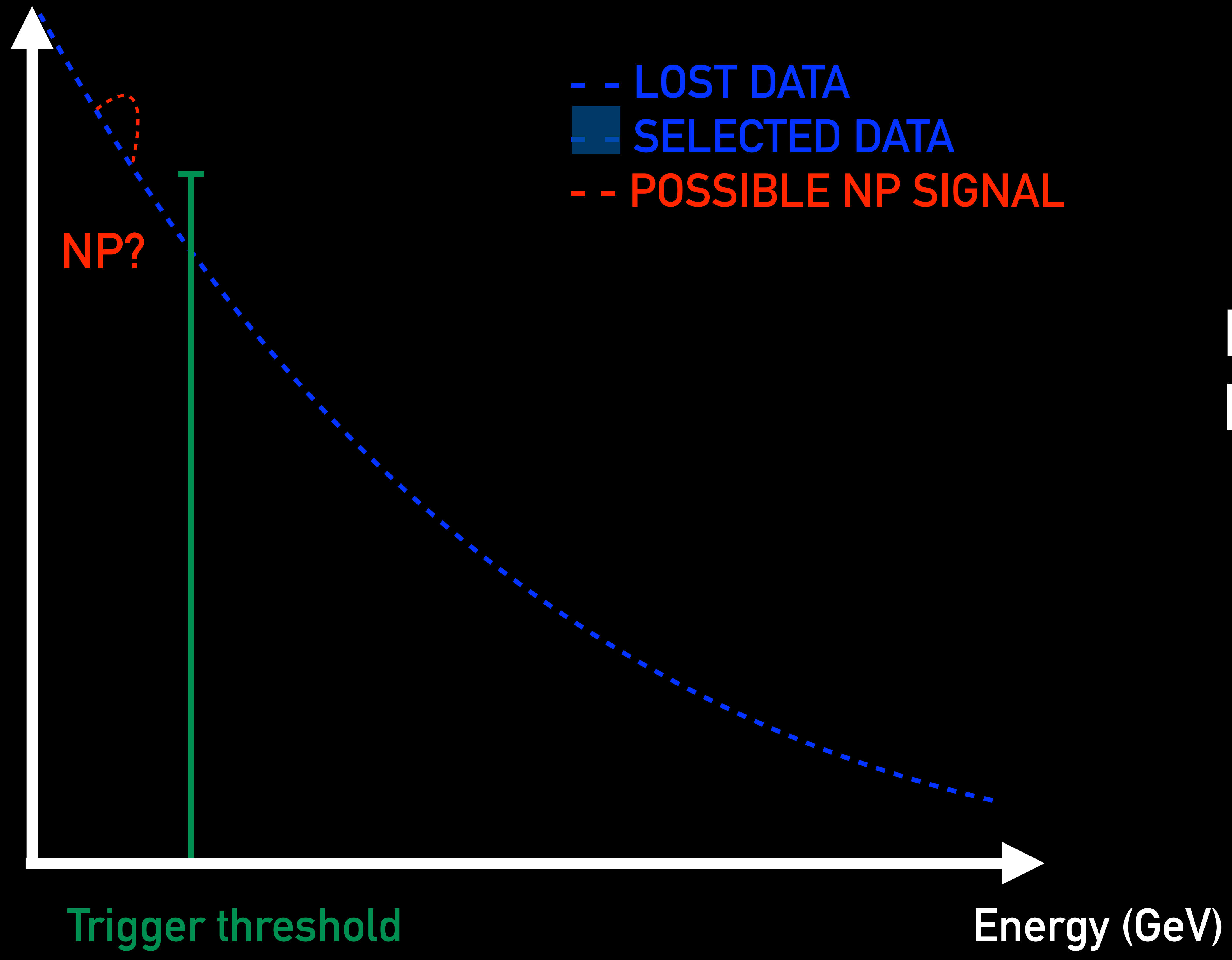
Anomaly detection for New Physics searches



LEARN THIS FROM DATA

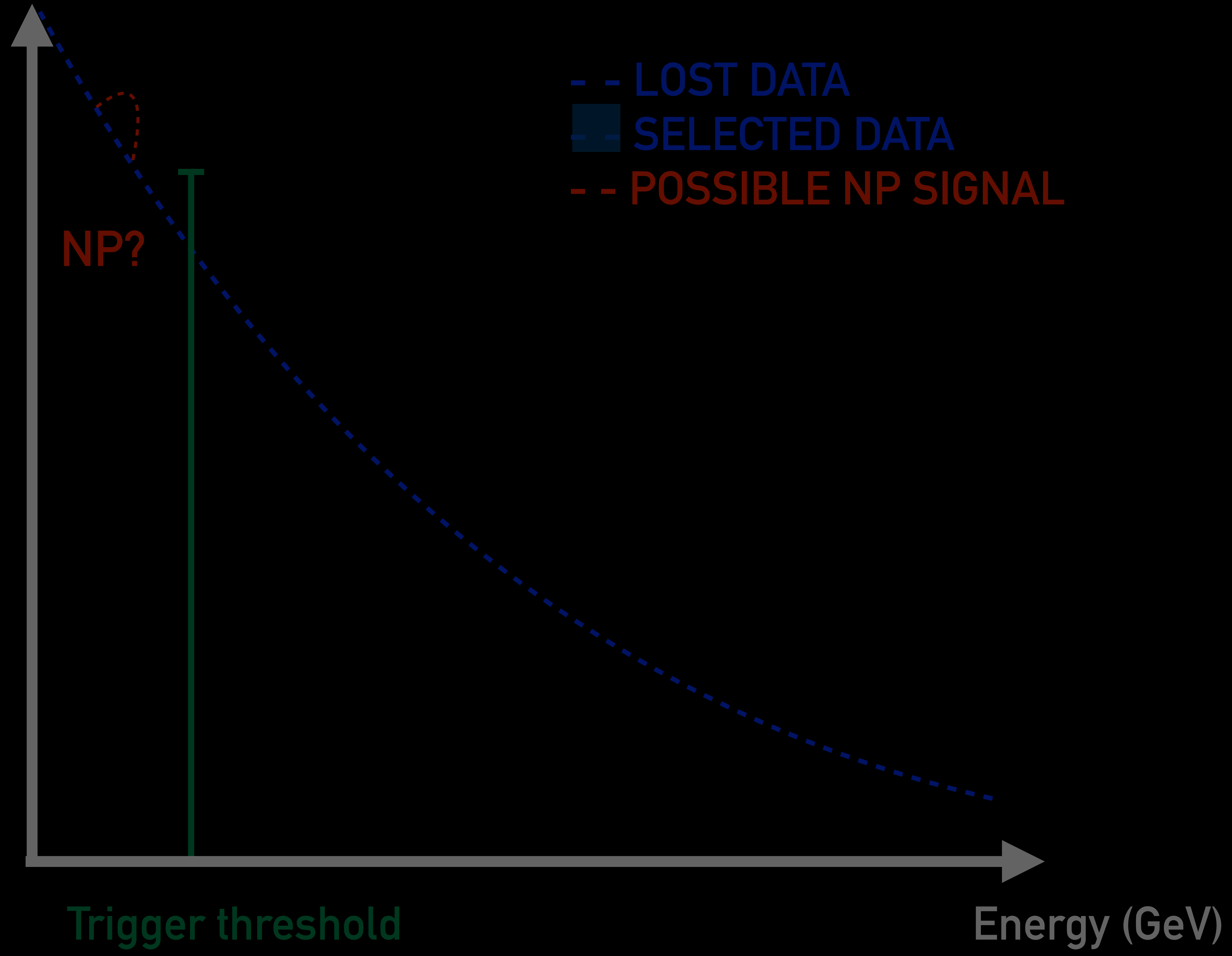
LOOK FOR ANYTING THAT DOESNT LOOK LIKE THIS

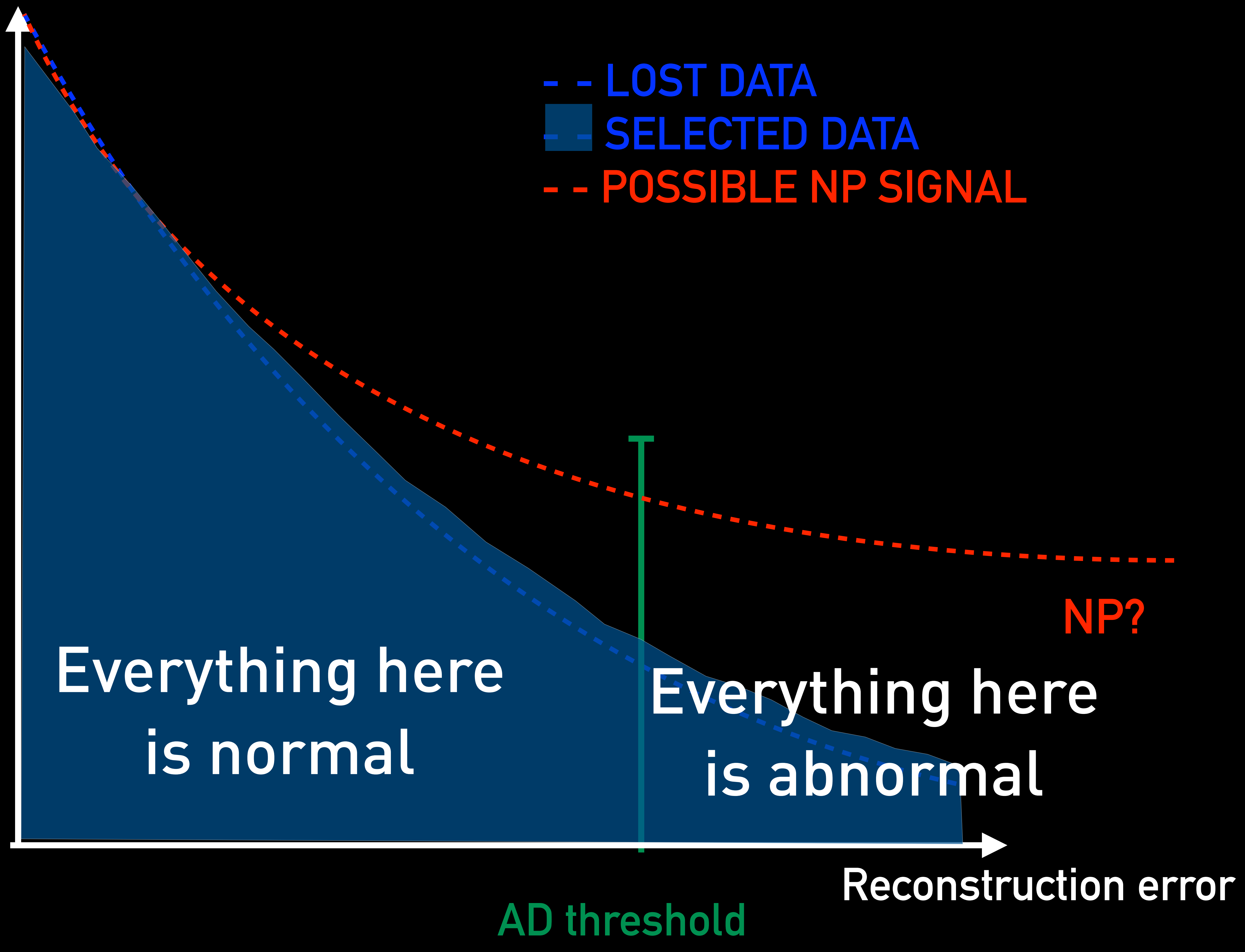
Anomaly Detection triggers



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

Anomaly Detection triggers

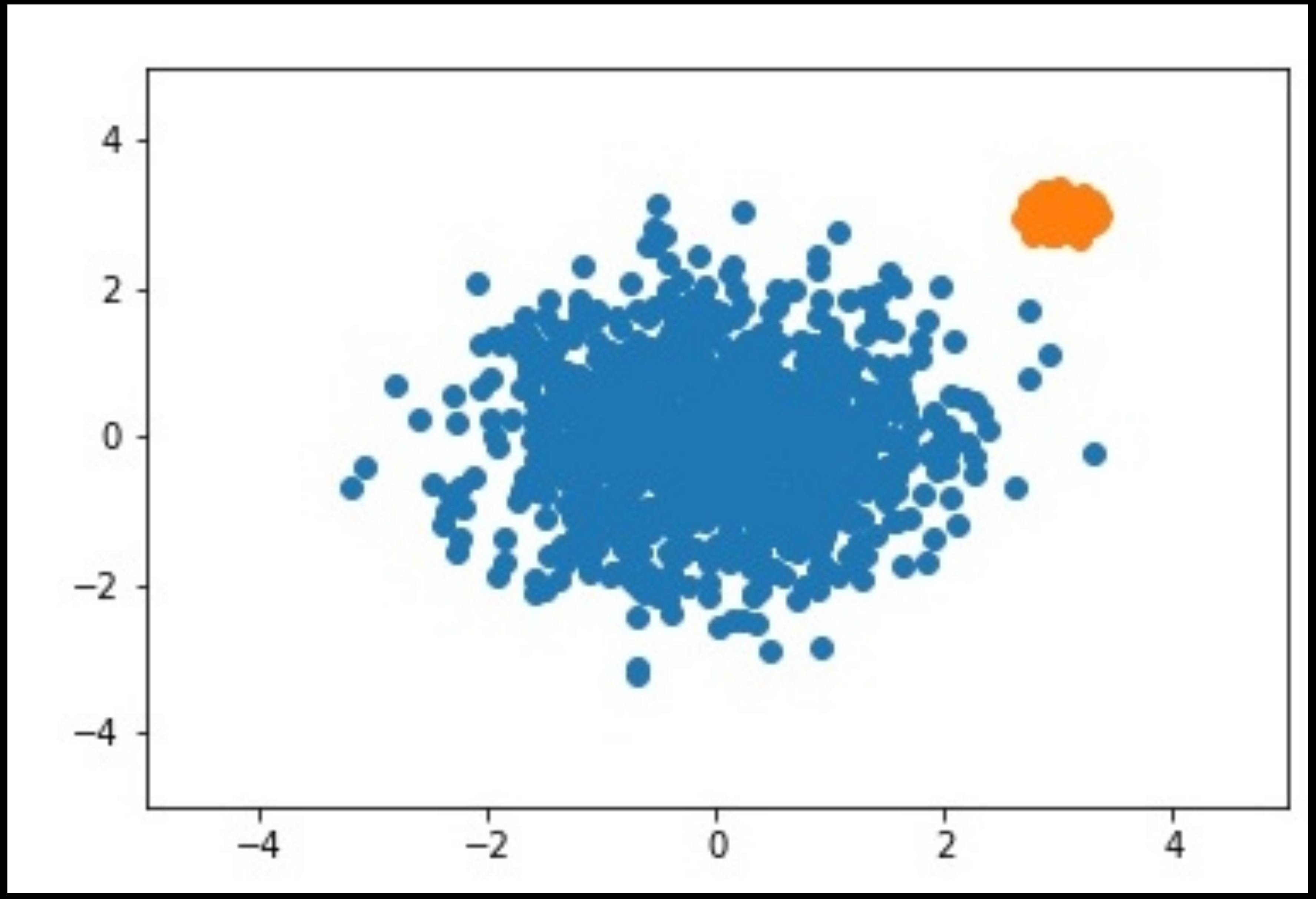




Types of anomaly detection

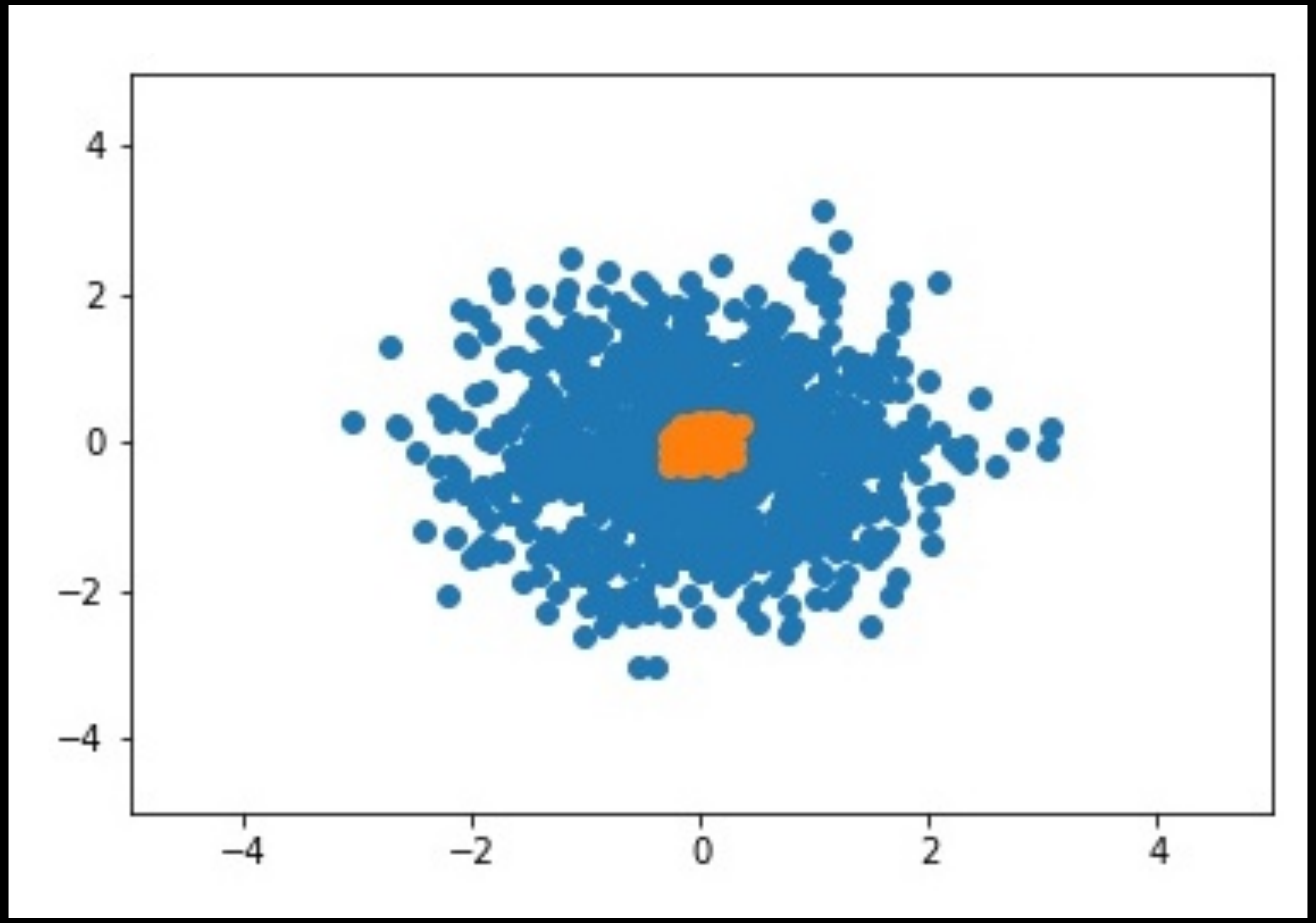
Outlier detection

Find (non-resonant) out-of-distribution datapoints



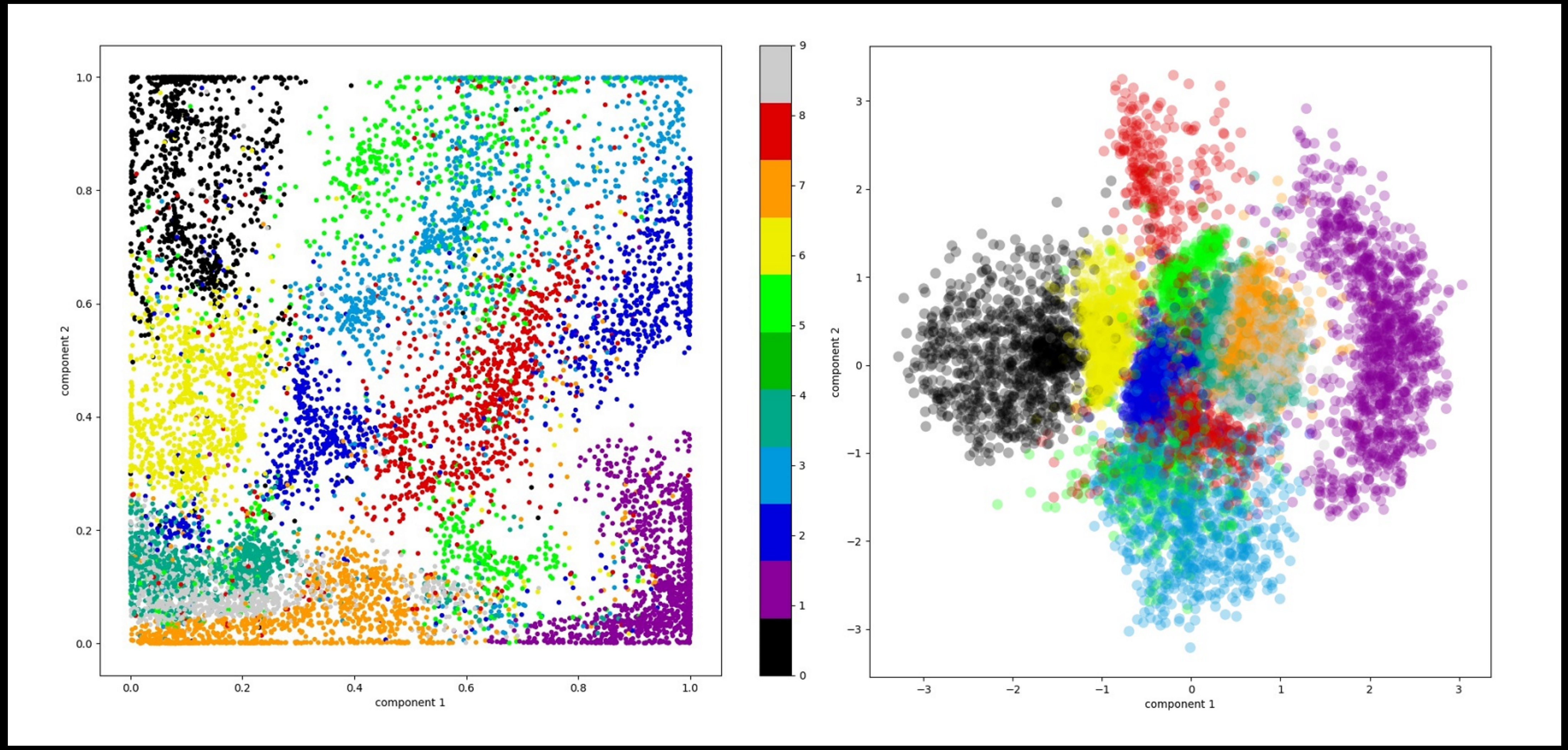
Detecting overdensities

Find (resonant) overdensities in distributions



Types of anomaly detection

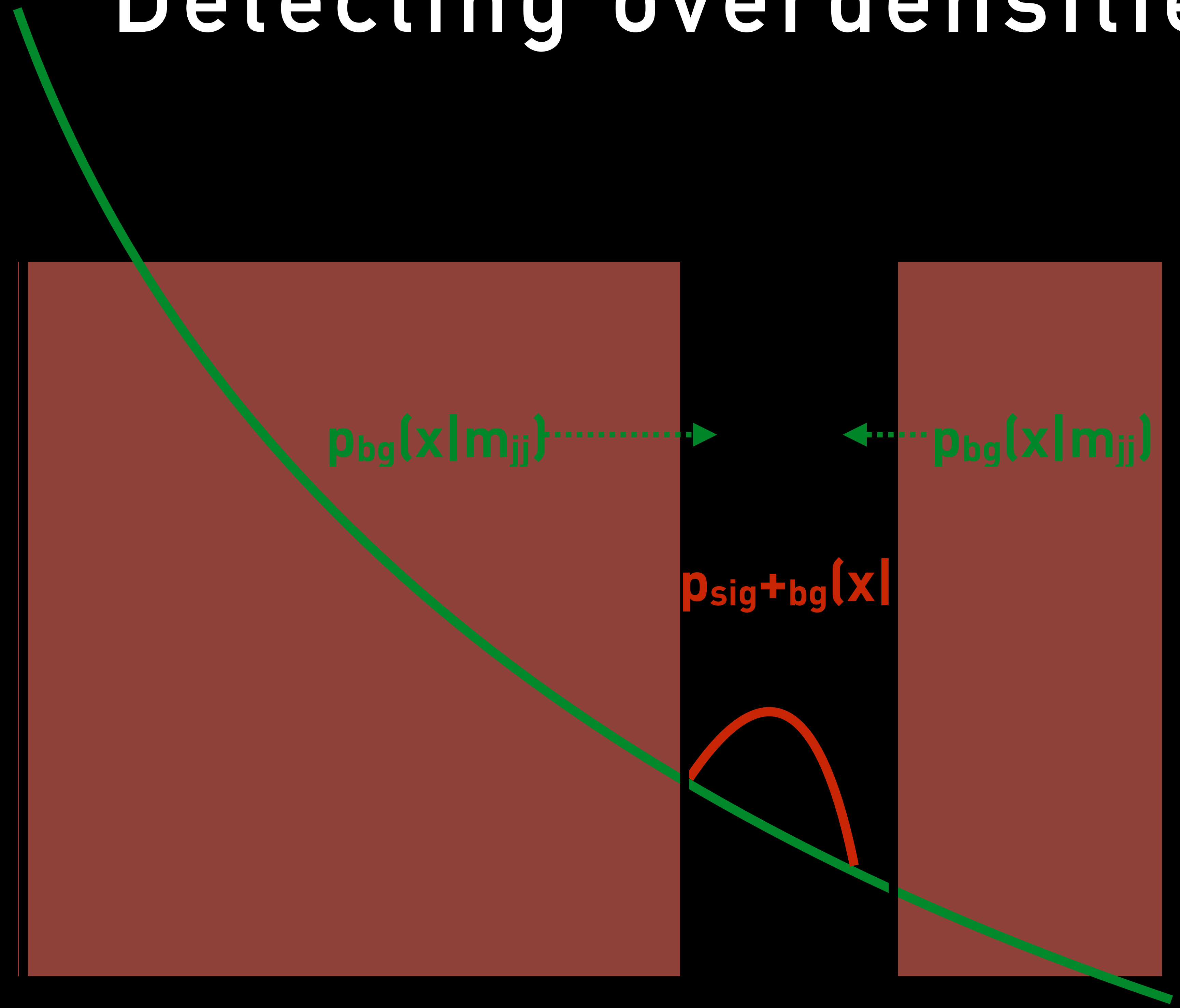
Outlier detection



Non-resonant, tail of distributions
- Often (variational) auto-encoders
- Useful for triggering!

Caveats
- What's a good metric for optimisation?
- How to use selected events in analysis?

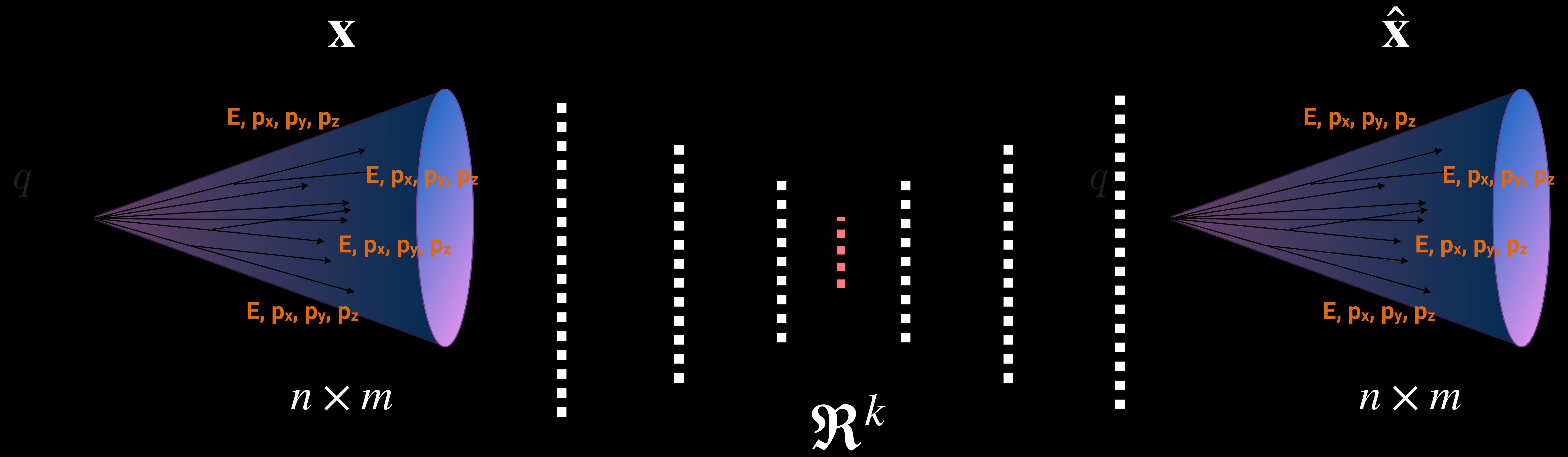
Detecting overdensities



Resonant, similar to a bump hunt
- Density estimation methods
- Useful for offline analysis

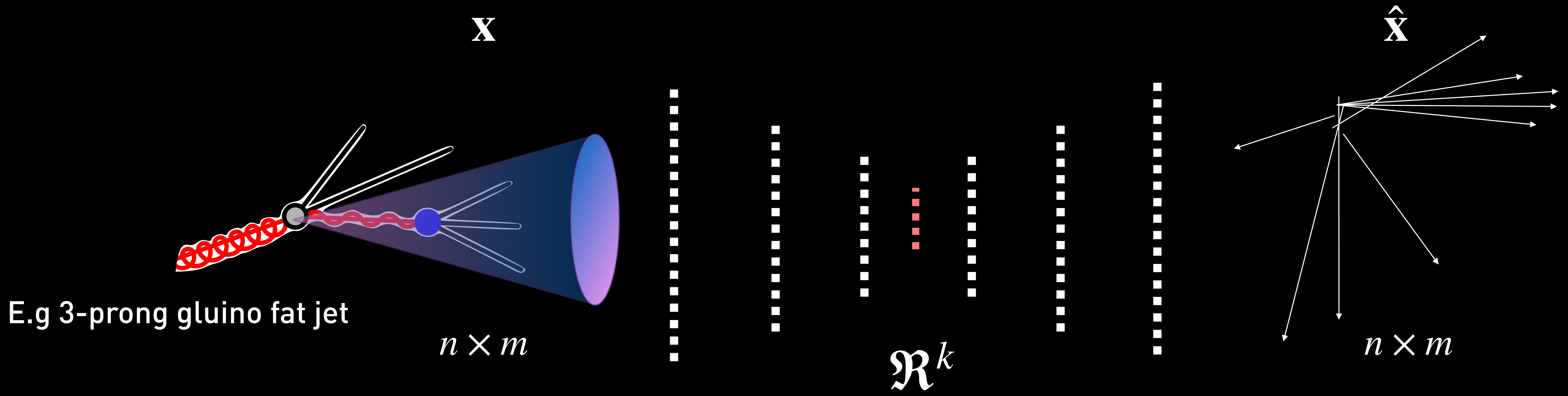
Caveats
- Relies on a definition of "sideband" and a sizeable signal

Outlier detection



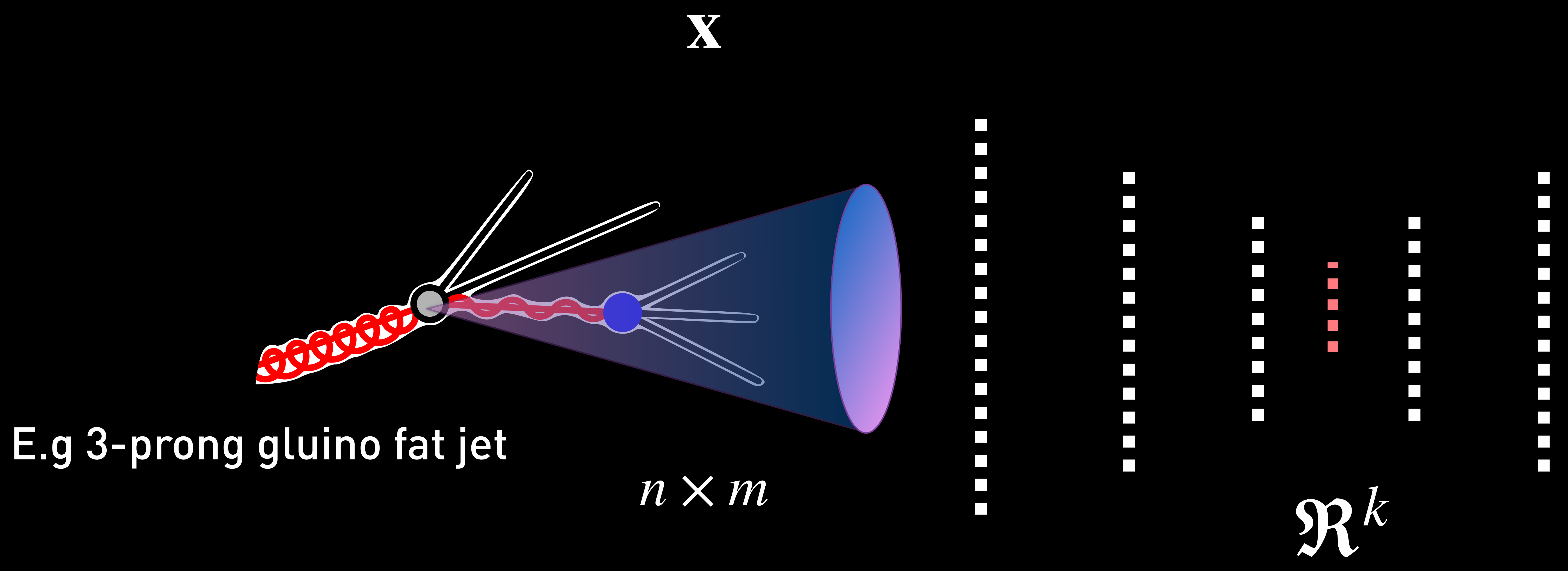
Compressed representation of x.
Latent space $\Re^k$, $k < m \times n$
prevents memorisation of input, must learn

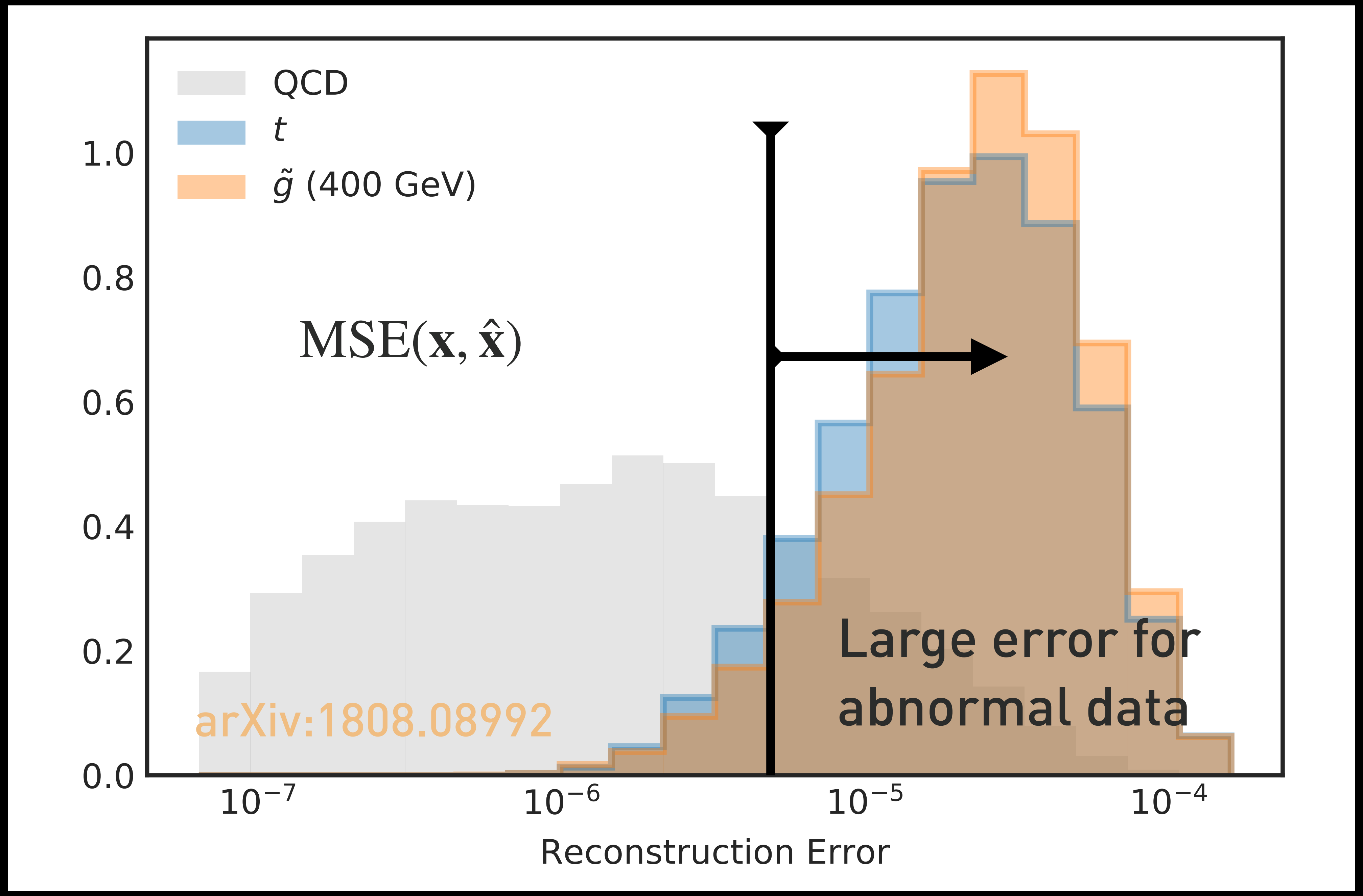
Outlier detection



$\mathscr{L}(\mathbf{x}, \hat{\mathbf{x}})$ is Mean Squared Error$(\mathbf{x}, \hat{\mathbf{x}})$, “high error events” proxy for “degree of abnormality”

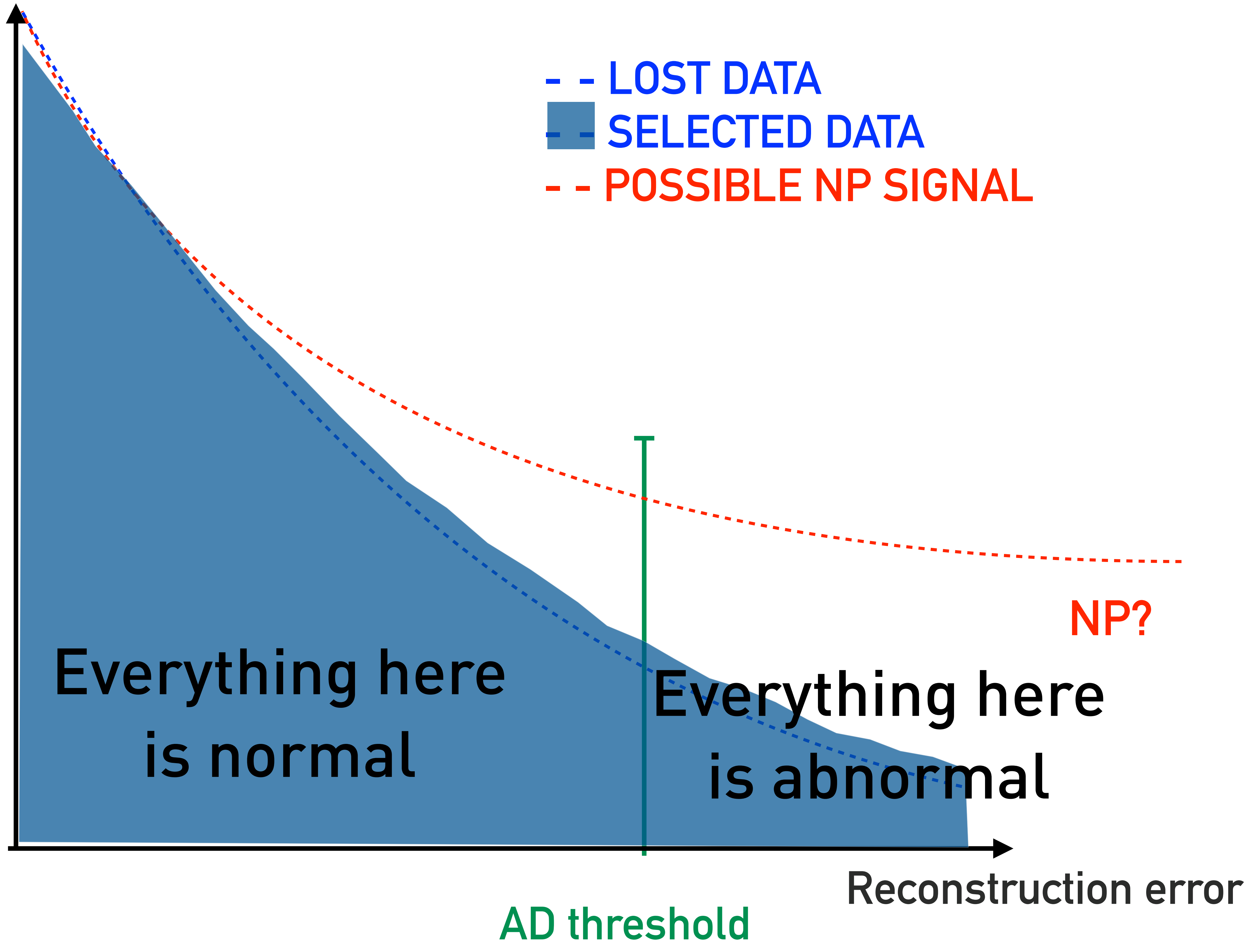
Outlier detection





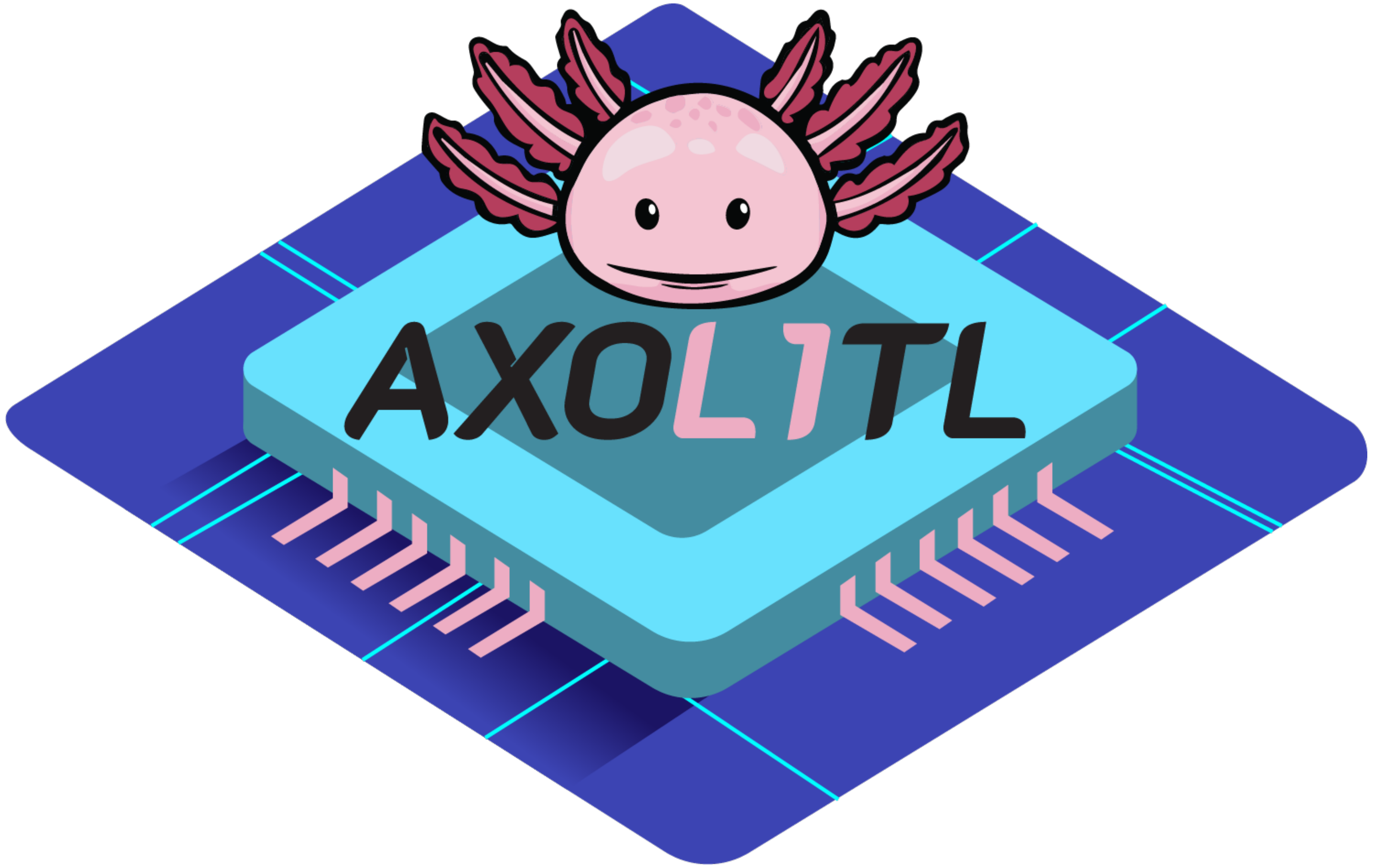
Large error for abnormal data

arXiv:1808.08992



Everything here is normal

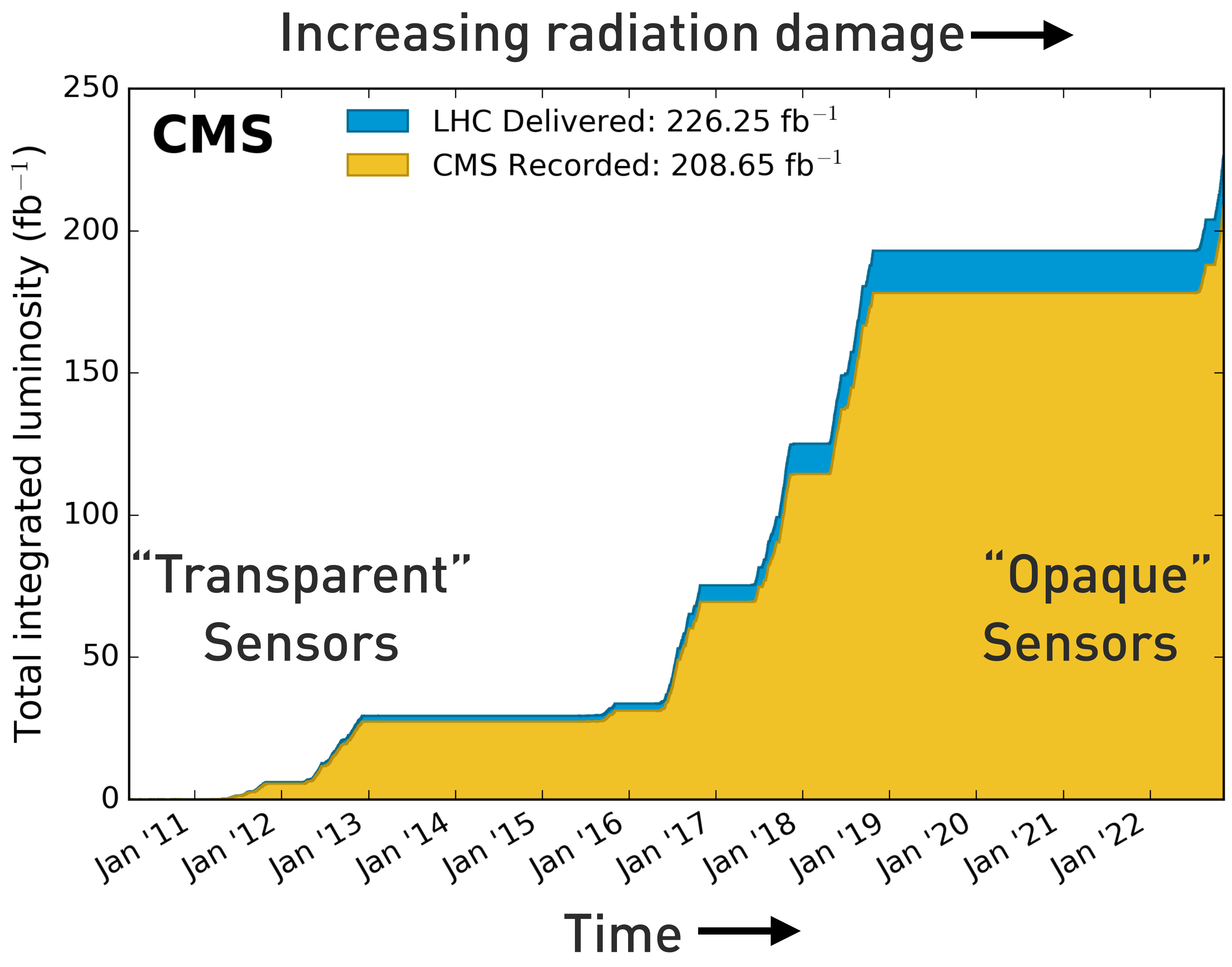
Everything here is abnormal



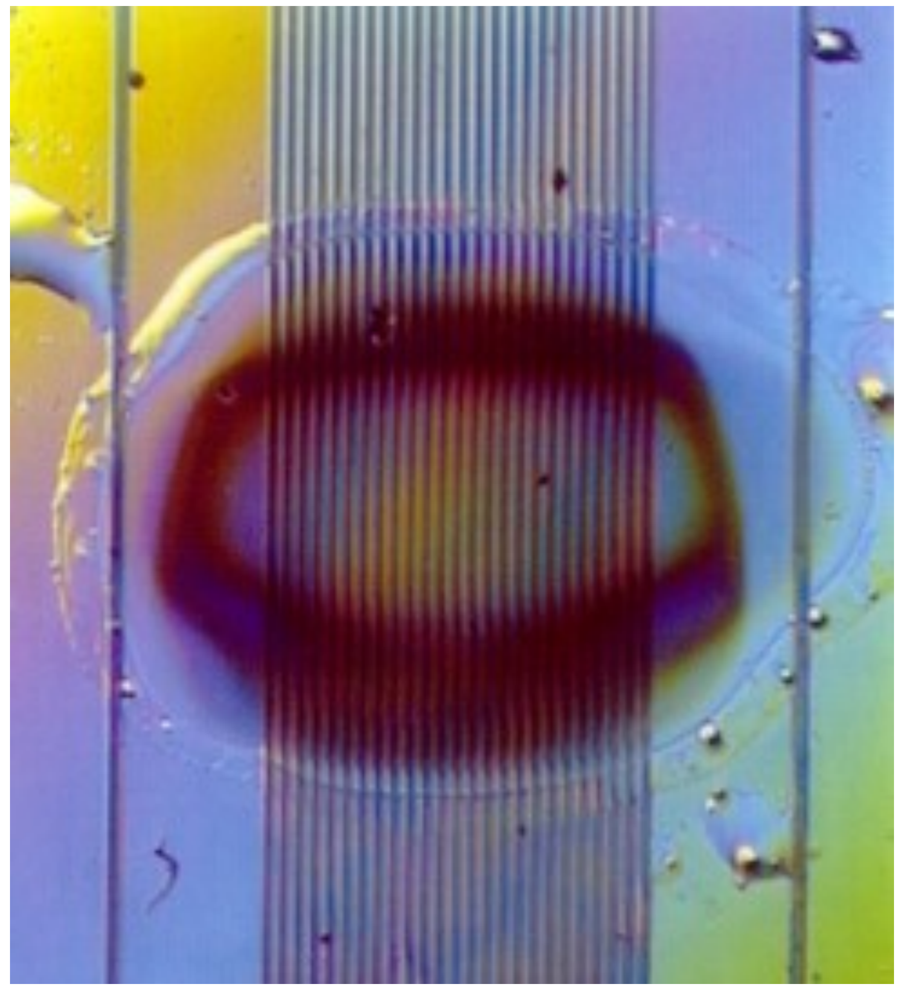
....in 50 nanoseconds!
Currently taking data in CMS!

https://cds.cern.ch/record/2859651

Continual learning



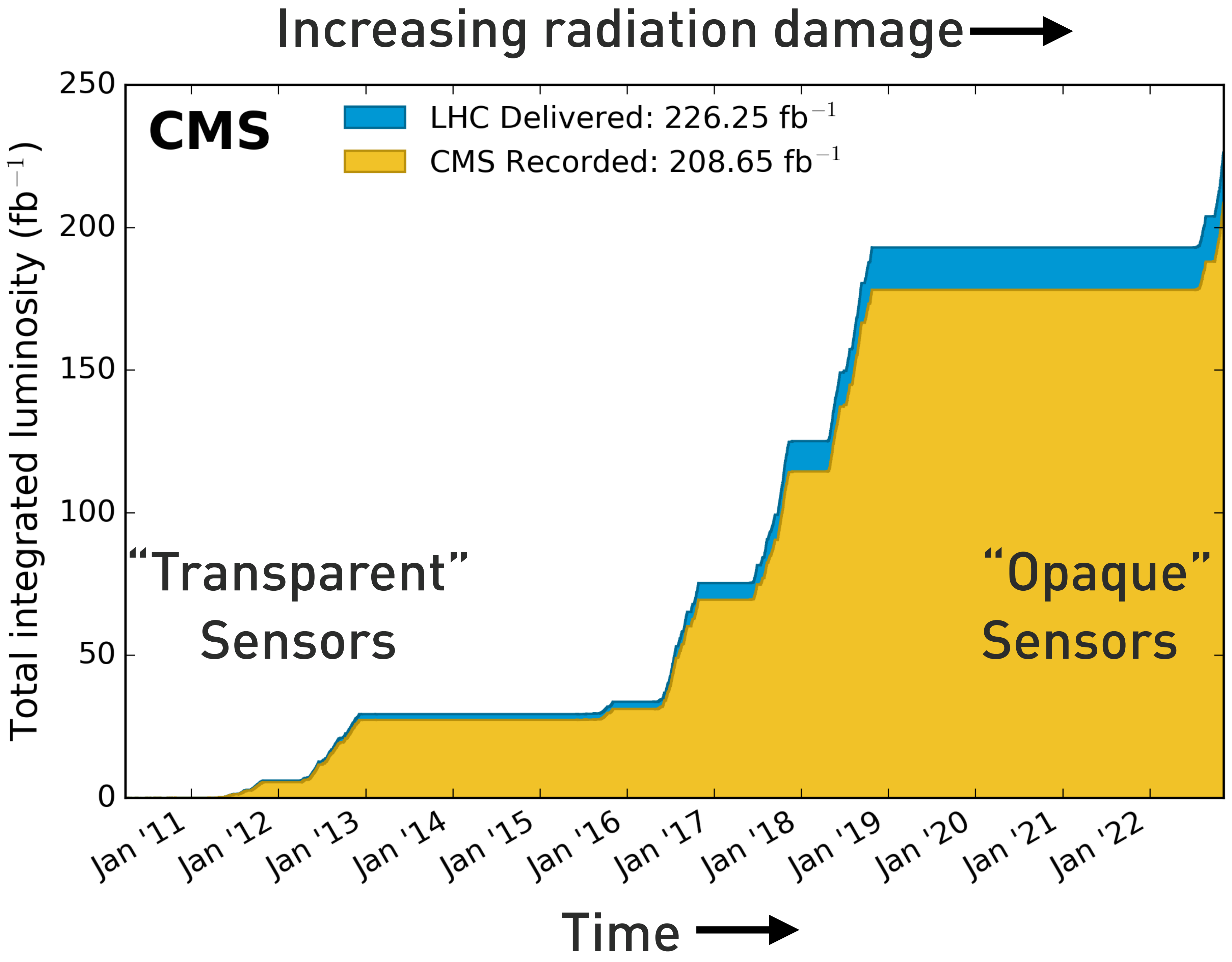
Radiation damage of silicon detector

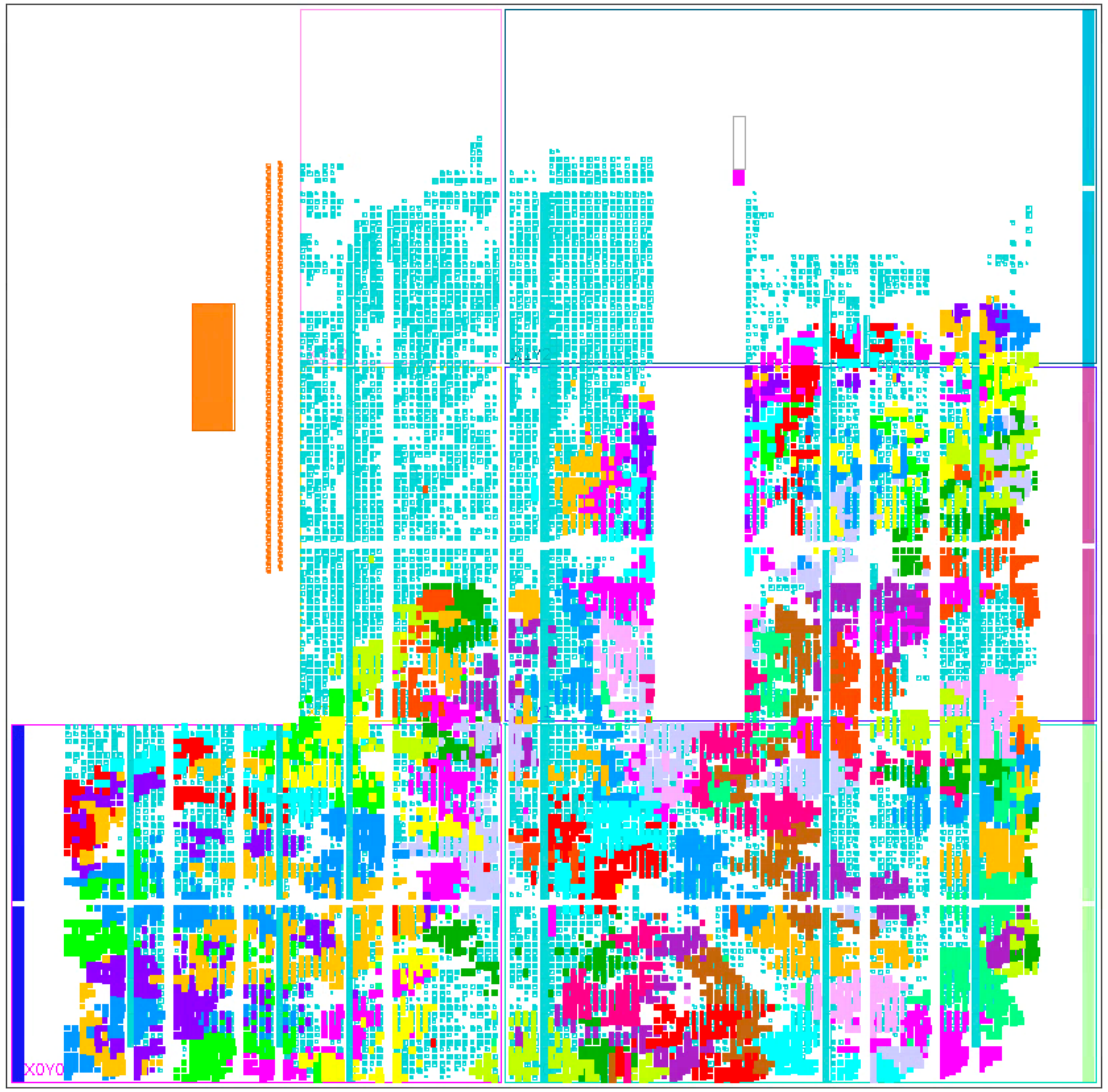


Many reasons for changing conditions

- Detector position slightly changes
- Radiation damage

Continual learning



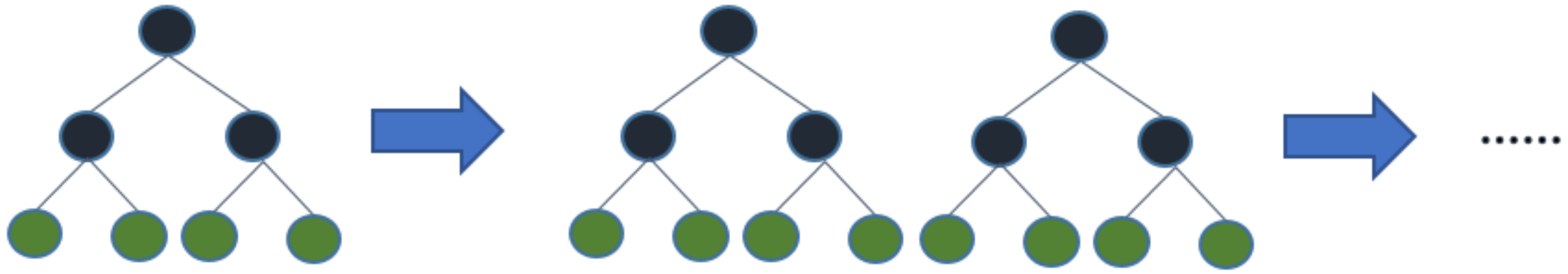


Many reasons for changing conditions

- Detector position slightly changes
- Radiation damage

Continual learning to the aid for self-supervised training?

- Avoid re-training on TBs of data, adapt to gradual changes!



F. Capel et al.

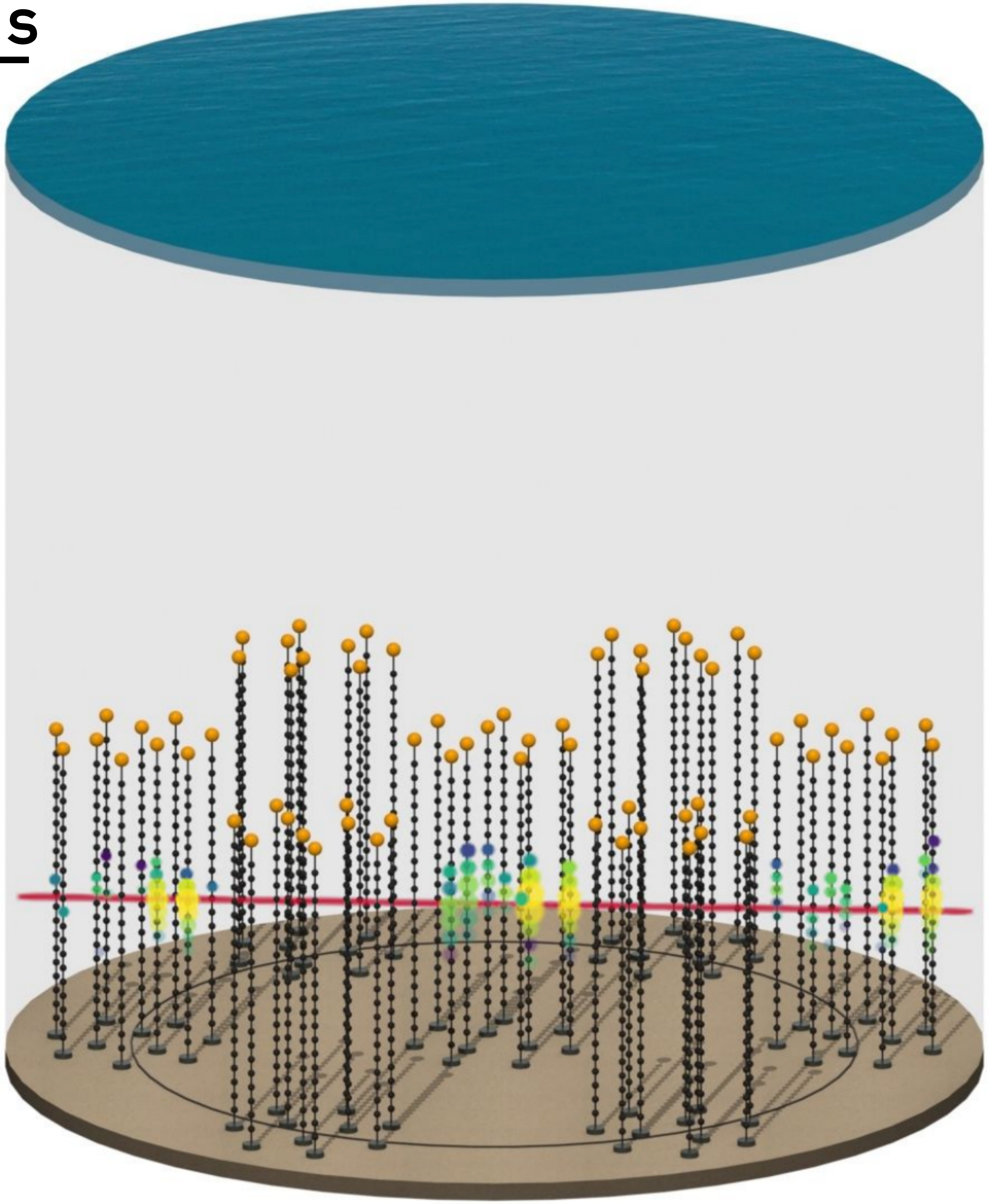
Real-time ML in other experiments

CERNCOURIER

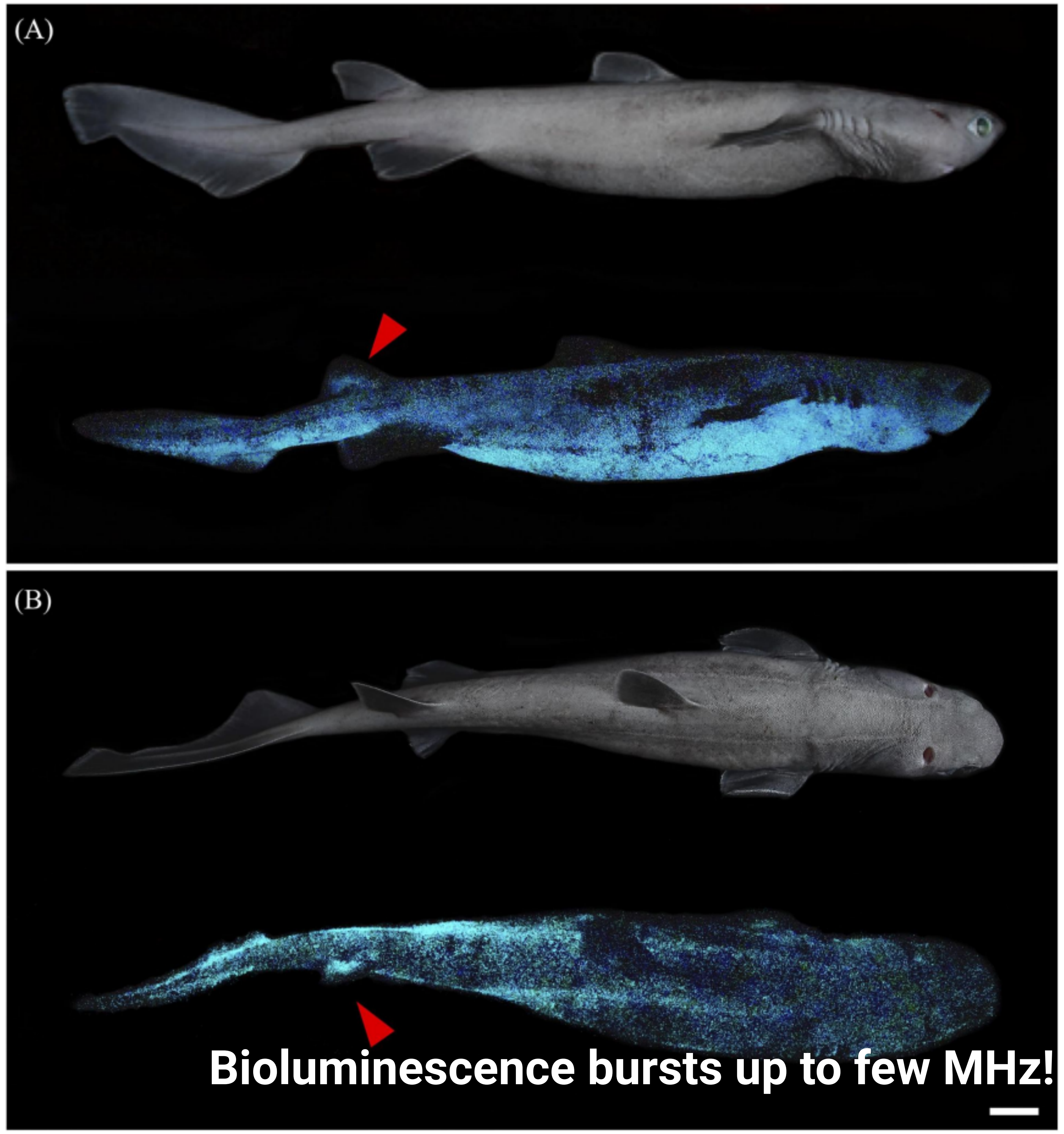
November/December 2022 cerncourier.com Reporting on international high-energy physics

NEUTRINOS OUT OF THE BLUE

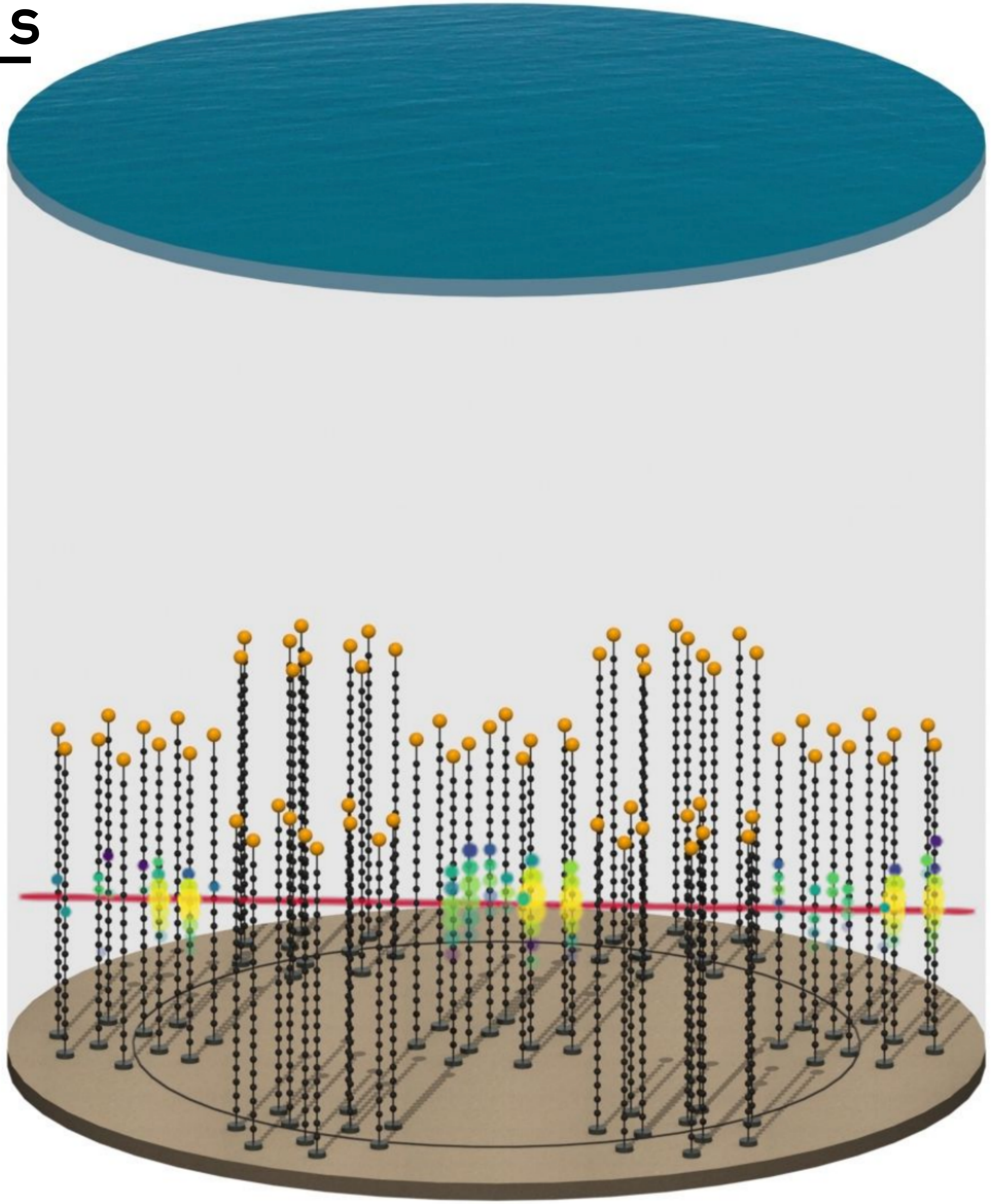
HL-LHC civil engineering complete
Snowmass: the full report
Taking plasma accelerators to market



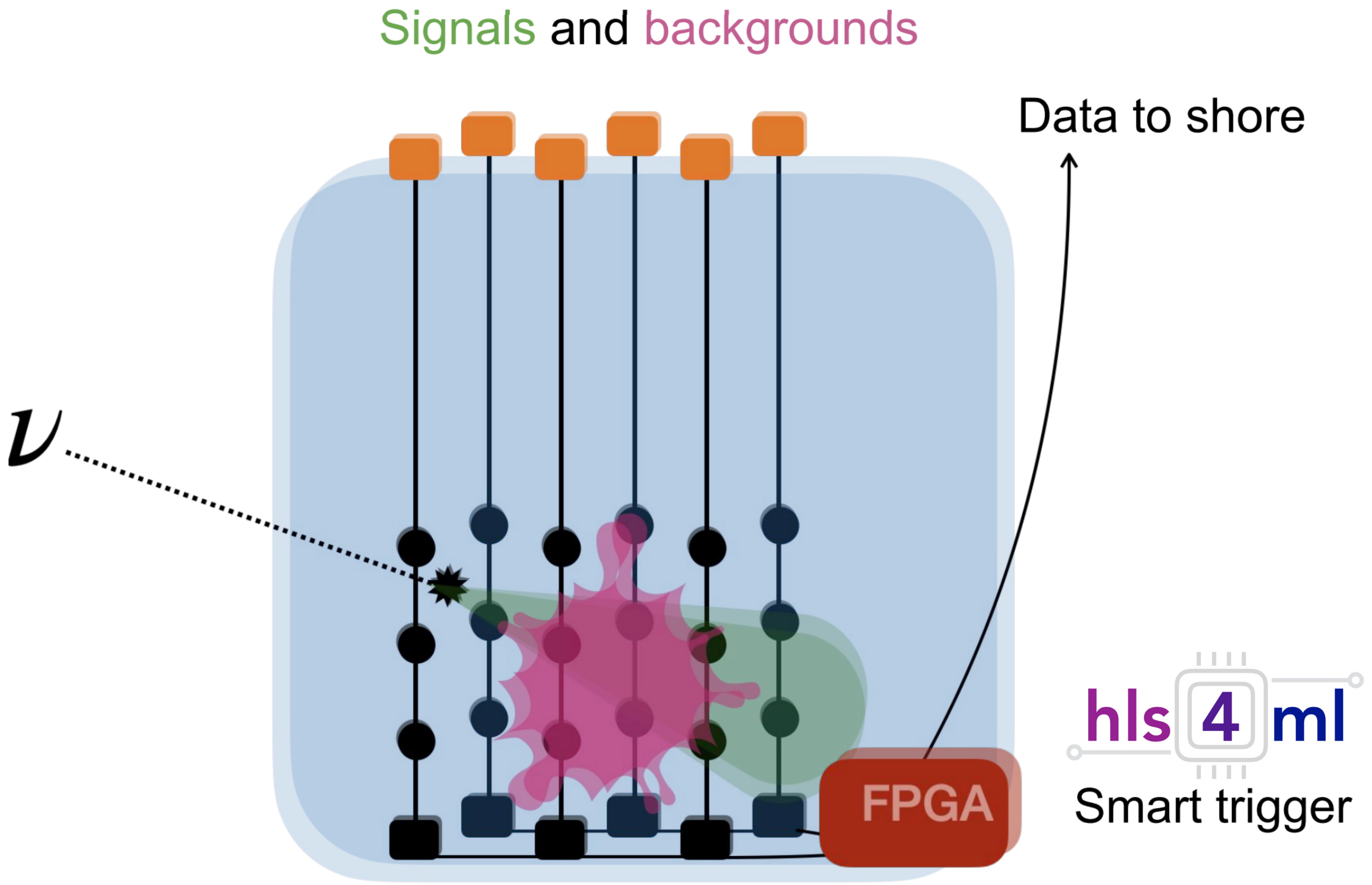
Real-time ML in other experiments



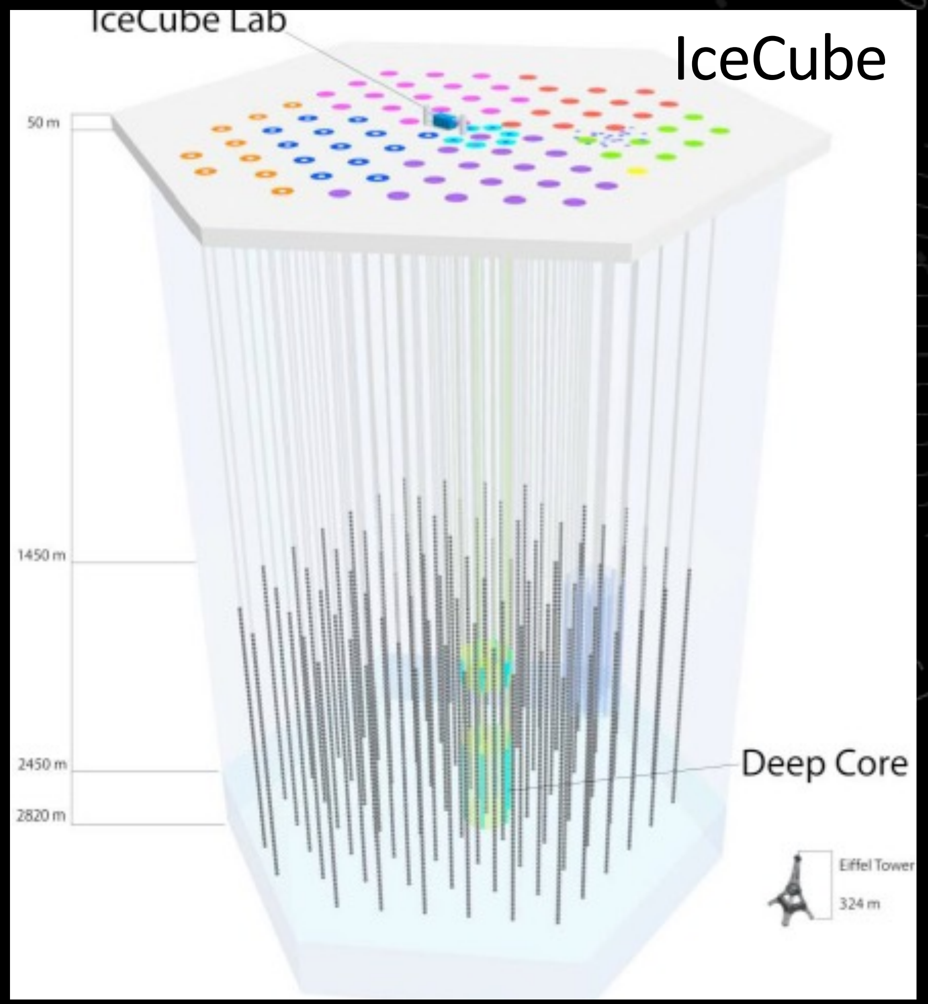
Bioluminescence bursts up to few MHz!



Real-time ML in other experiments



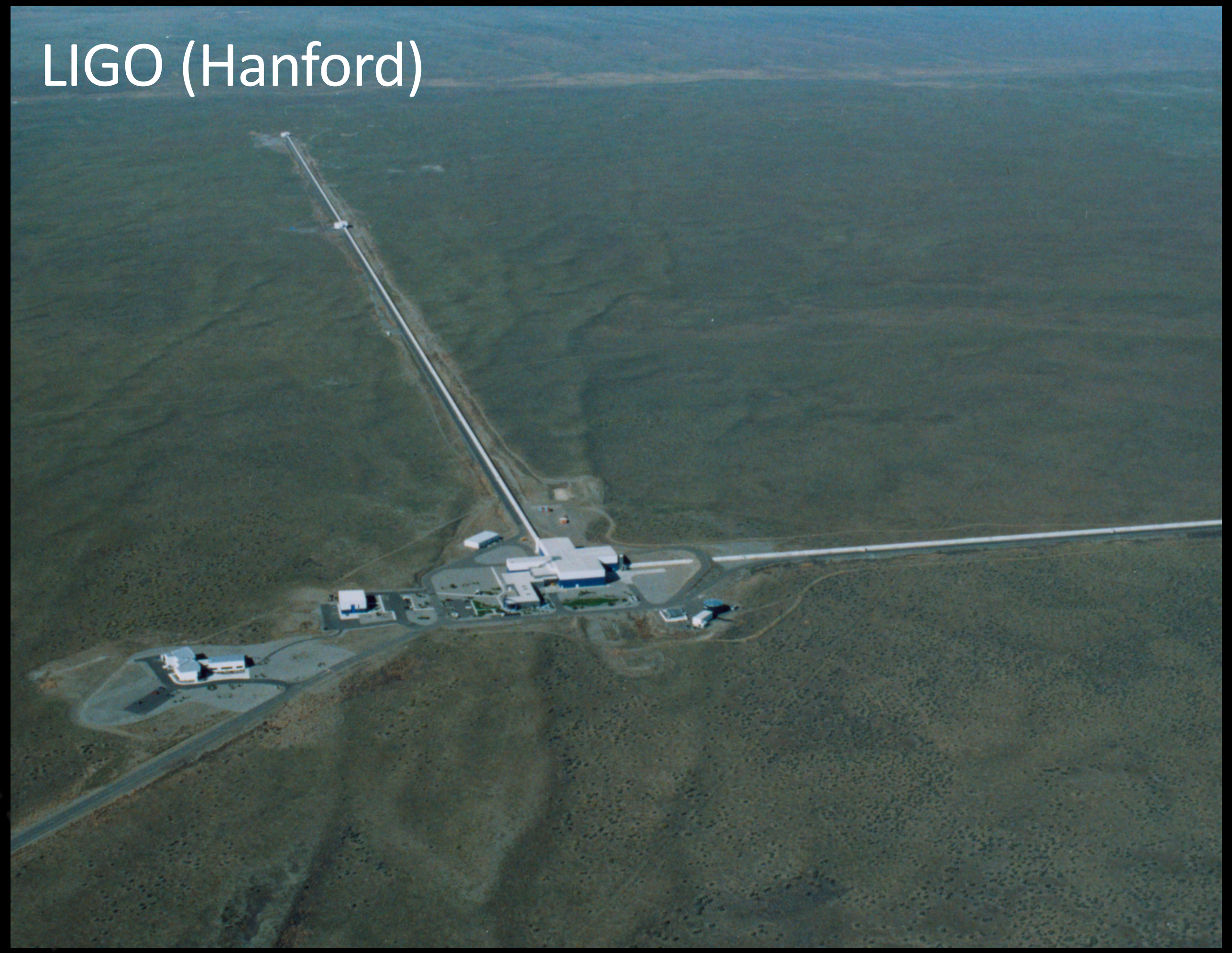
Electromagnetic waves



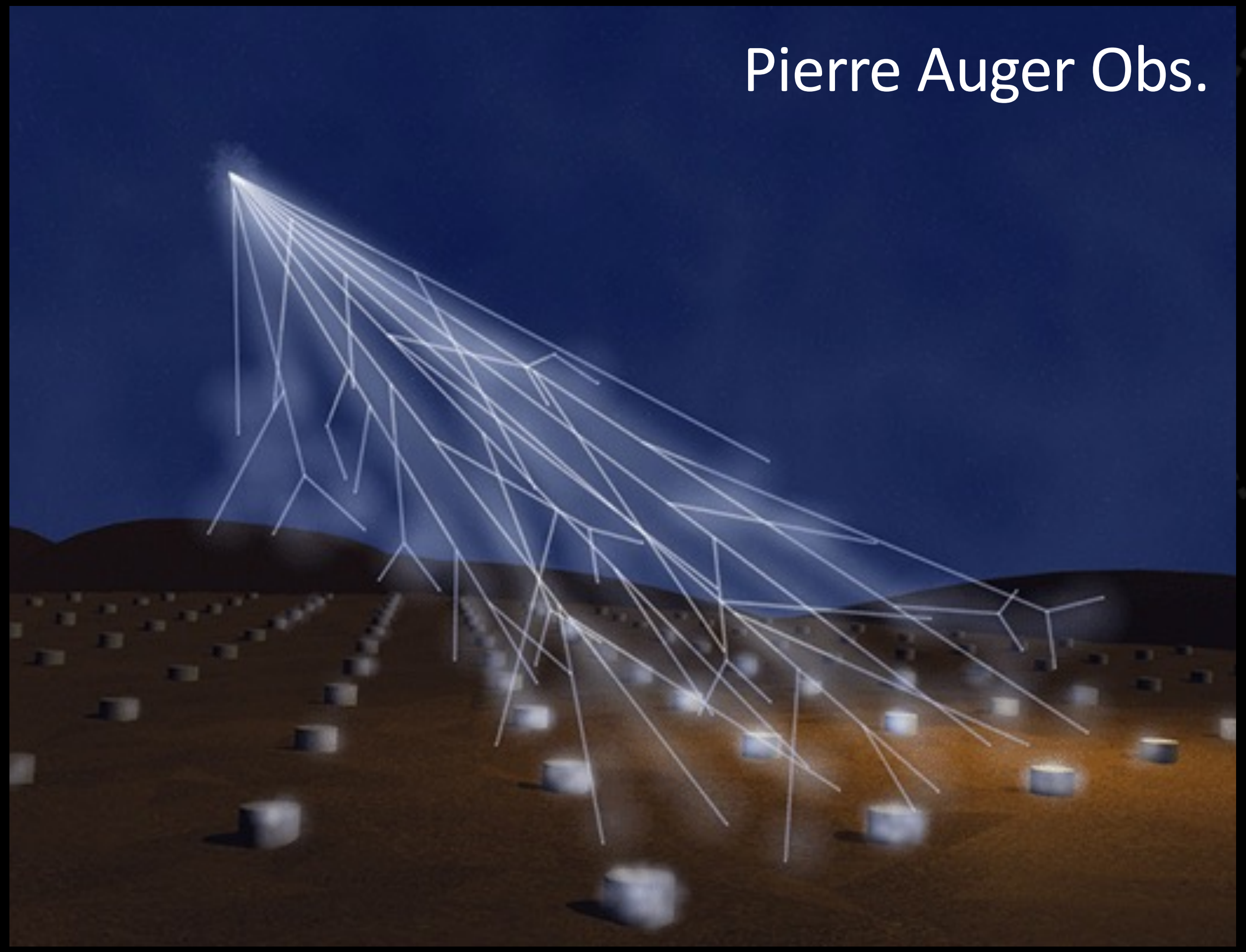
Neutrinos

MULTI-MESSENGER ASTRONOMY
(MMA)

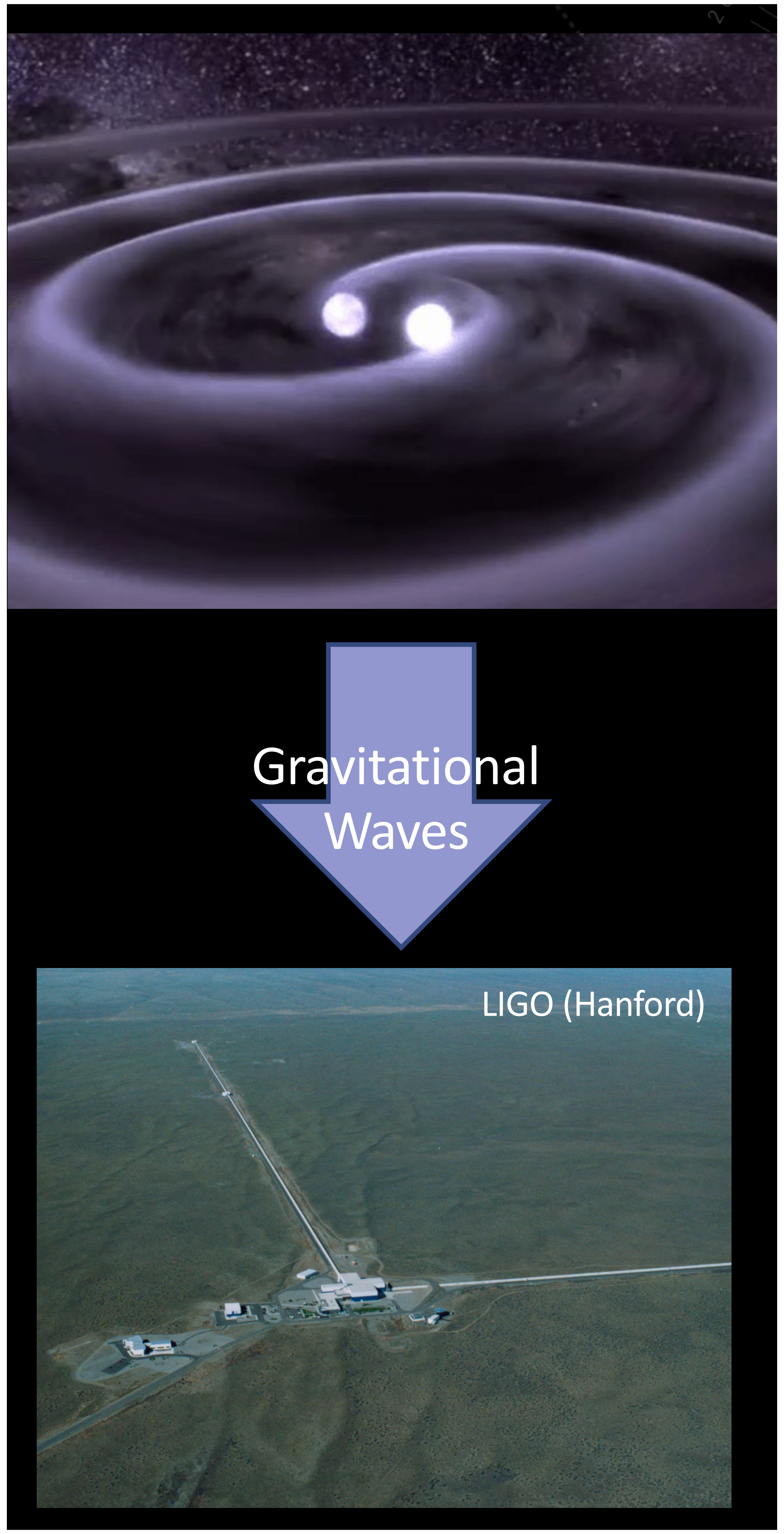
Gravitational Waves



Cosmic Rays

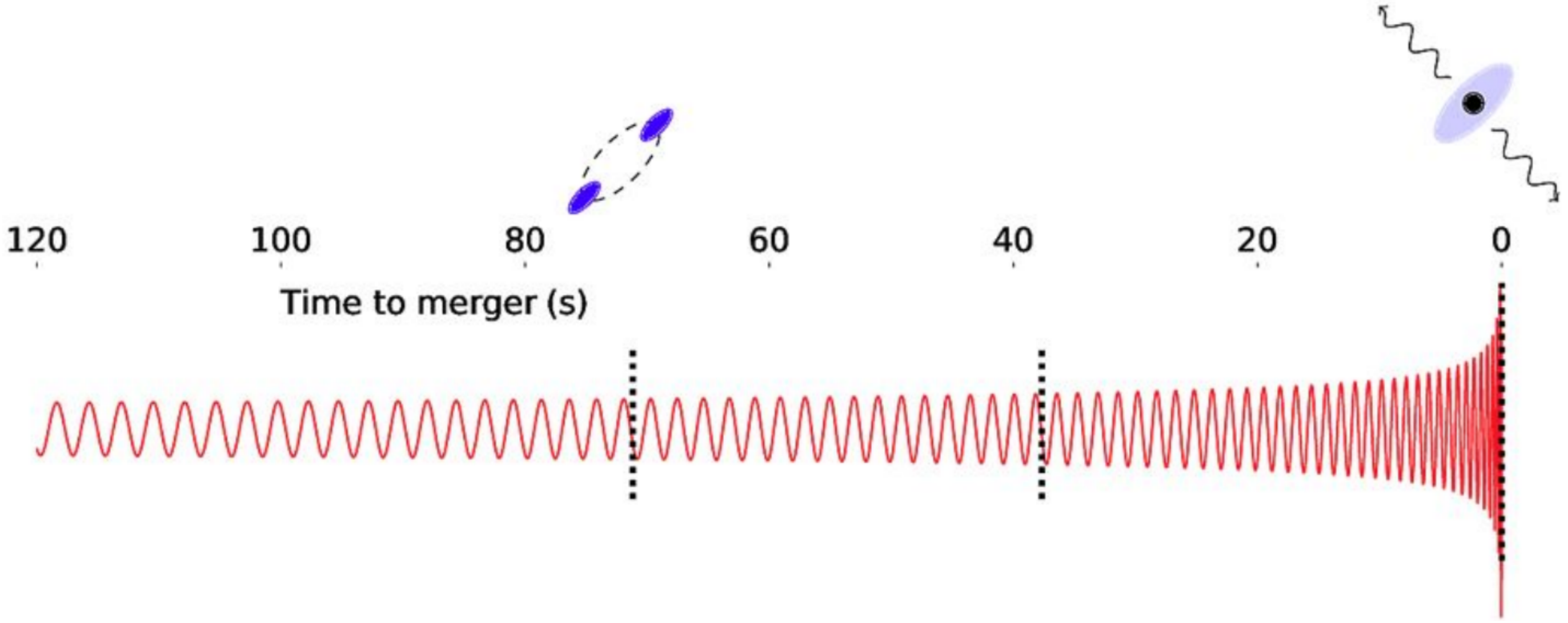


Real-time gravitational wave detection



Using gravitational waves, we can detect neutron star merger (kilo novae) before it occurs!

- Use real time ML for event cleaning, parameter estimation and anomaly detection!



Rapid communication of gravitational-wave date on

- source location
- properties

Crucial to facilitate multi-messenger follow-up of sources !

LSST TAKES 20TB OF IMAGES PER NIGHT

Rubin Observatory

10 million transient alerts per night!

Transient searches have relied on human eyes for alerts ***→real-time ML!***

How can we expect to get lucky with such high volumes of data?

...and outside of HEP

Semantic segmentation for autonomous vehicles

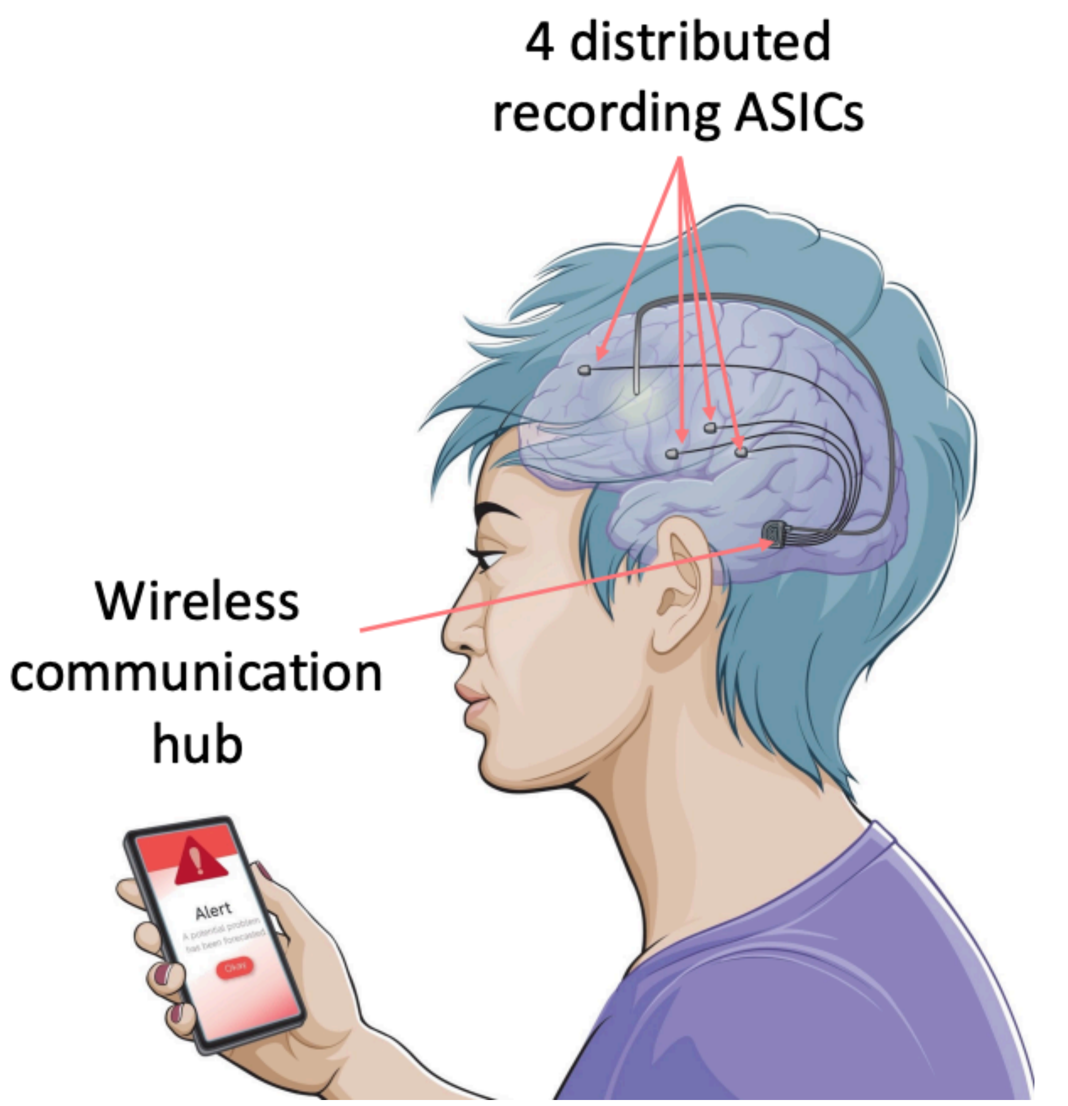
N. Ghielmetti et al.

...and outside of HEP

Semantic segmentation for autonomous vehicles

N. Ghielmetti et al.

Seizure Predicting Brain Implant



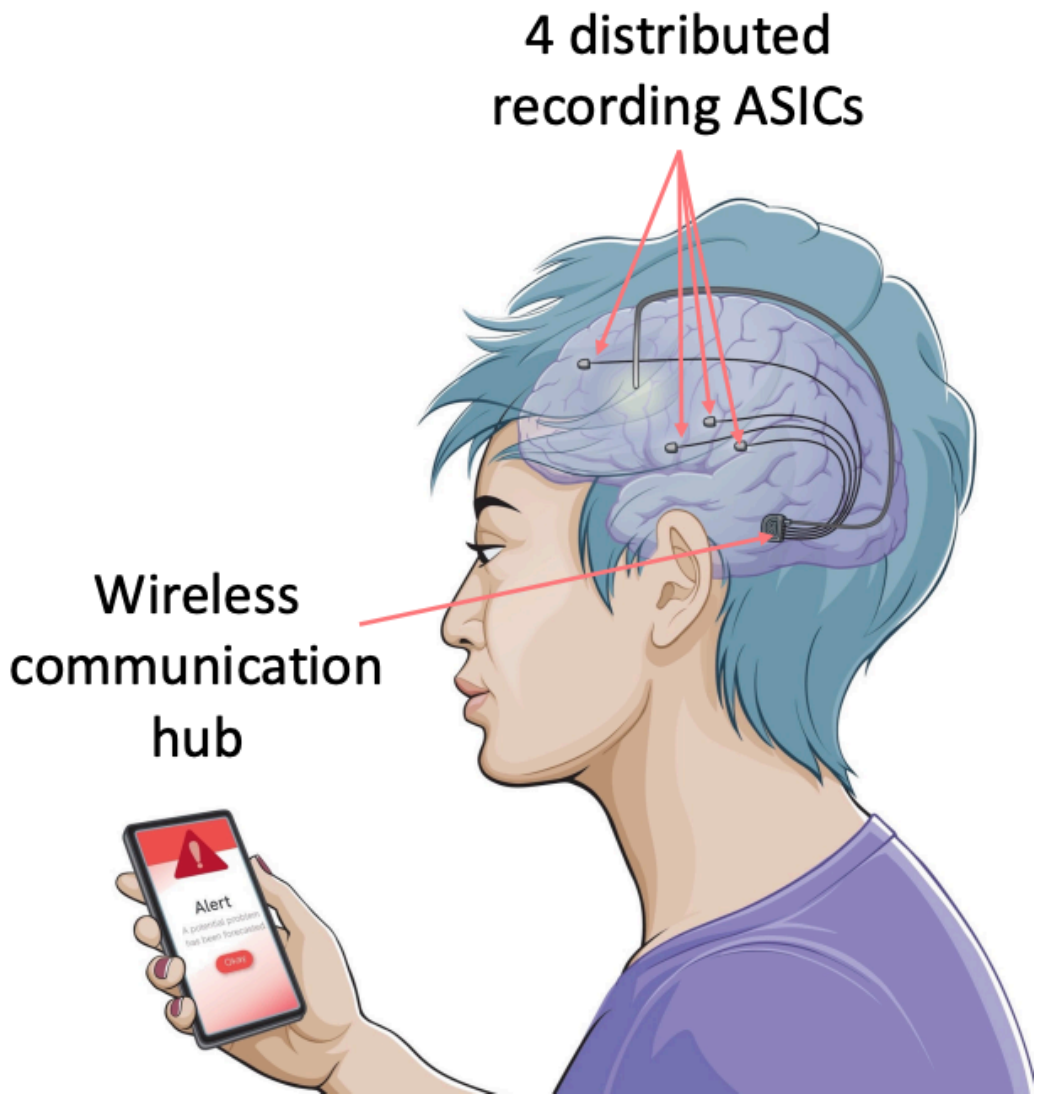
W. Lemaire et al.

...and outside of HEP

Semantic segmentation for autonomous vehicles

N. Ghielmetti et al.

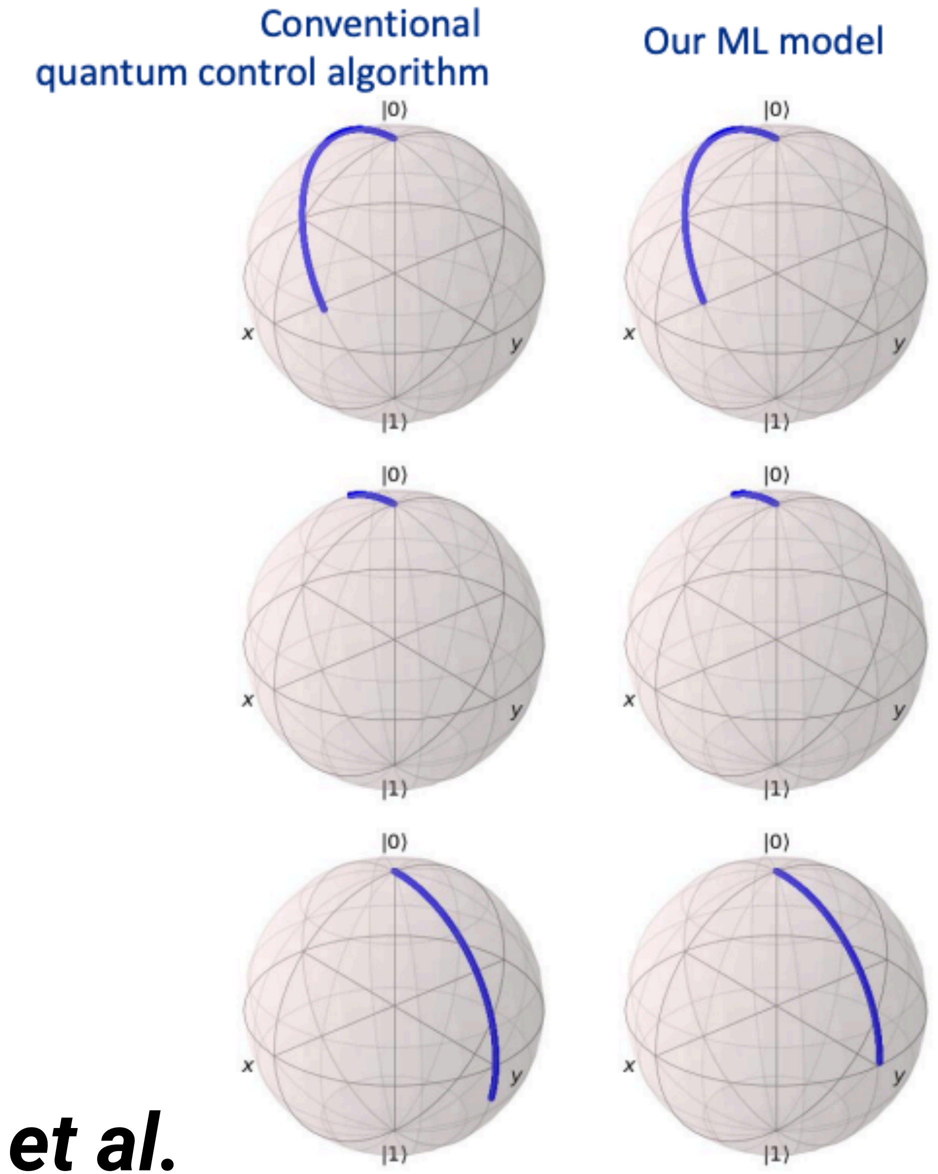
Seizure Predicting Brain Implant



W. Lemaire et al.

NN accelerator for quantum control

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



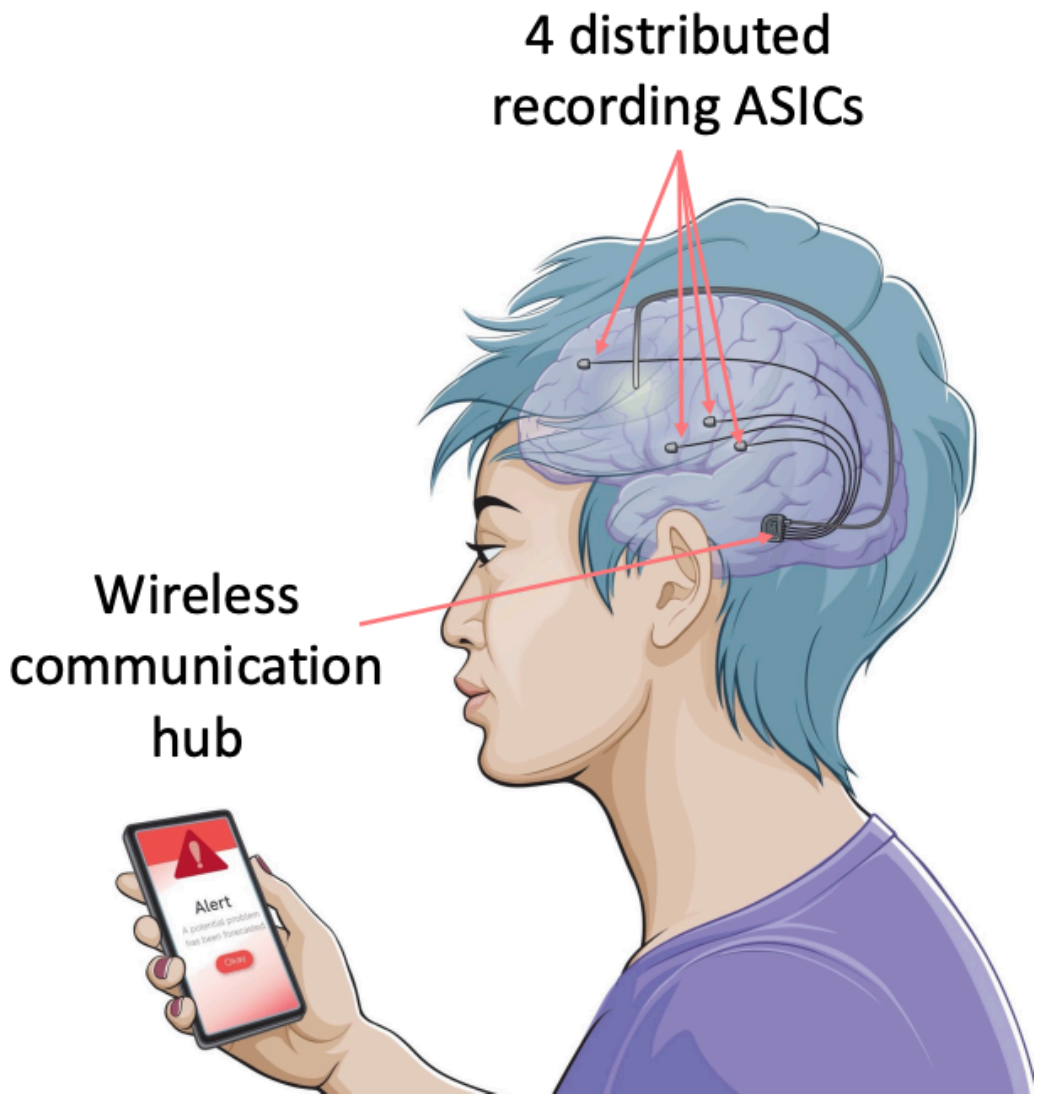
D Xu et al.

...and outside of HEP

Semantic segmentation for autonomous vehicles

N. Ghielmetti et al.

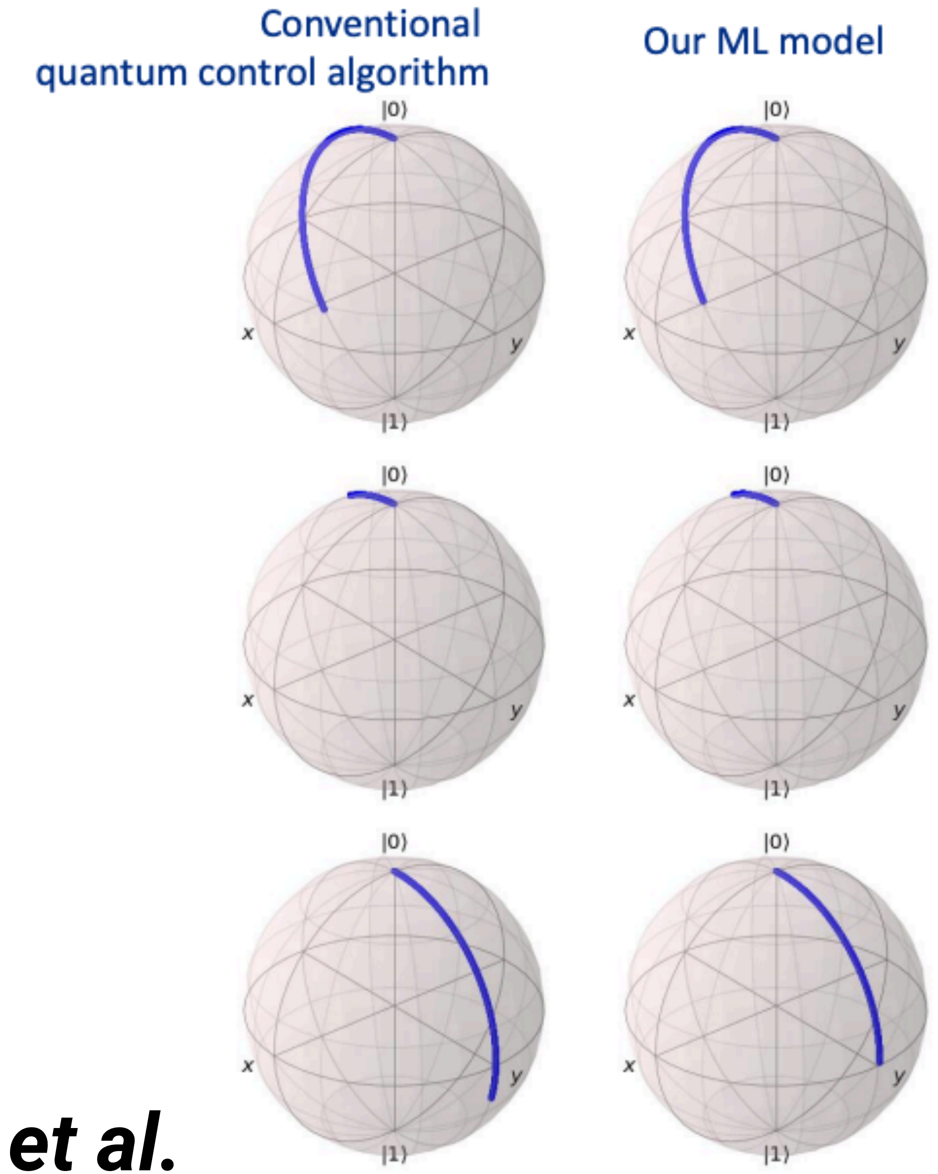
Seizure Predicting Brain Implant



W. Lemaire et al.

NN accelerator for quantum control

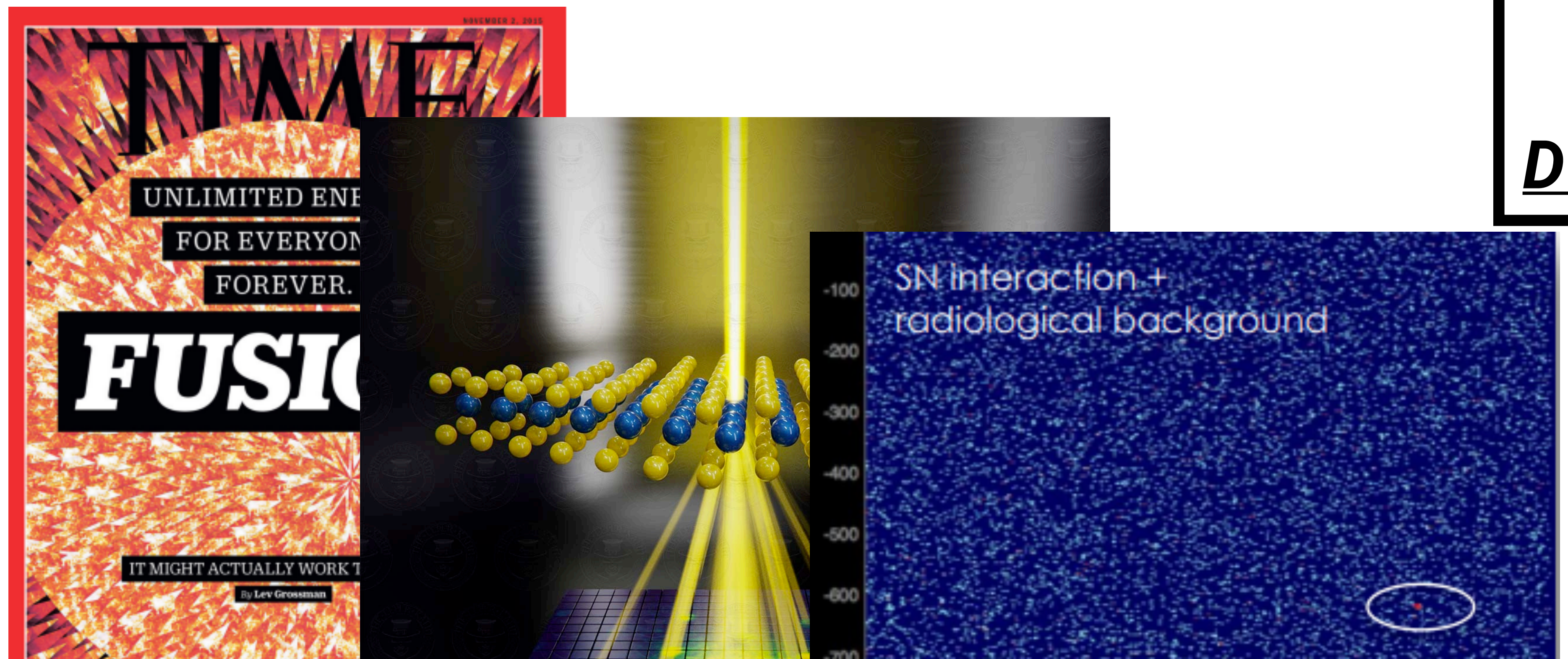
- Putting control in cryostat (e.g optimal pulse parameters)

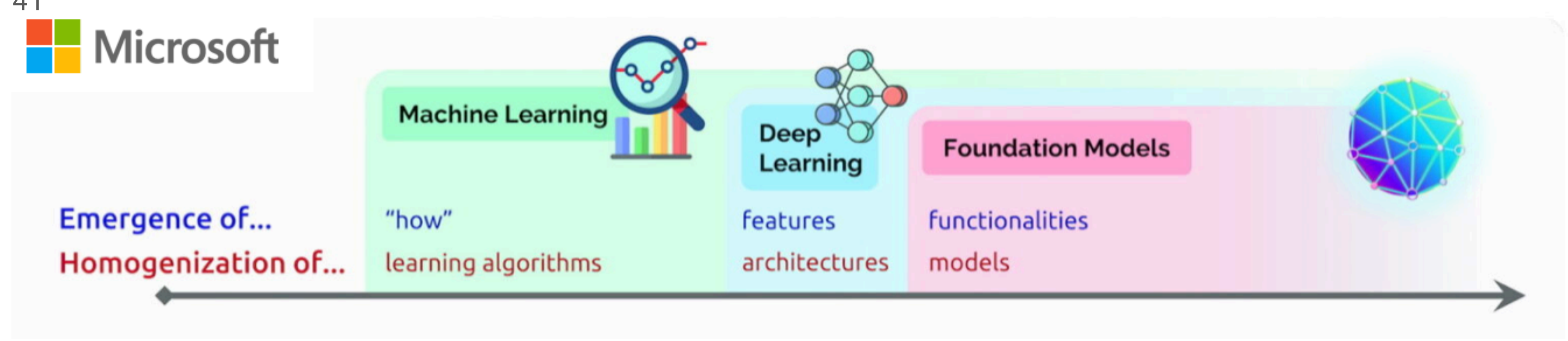


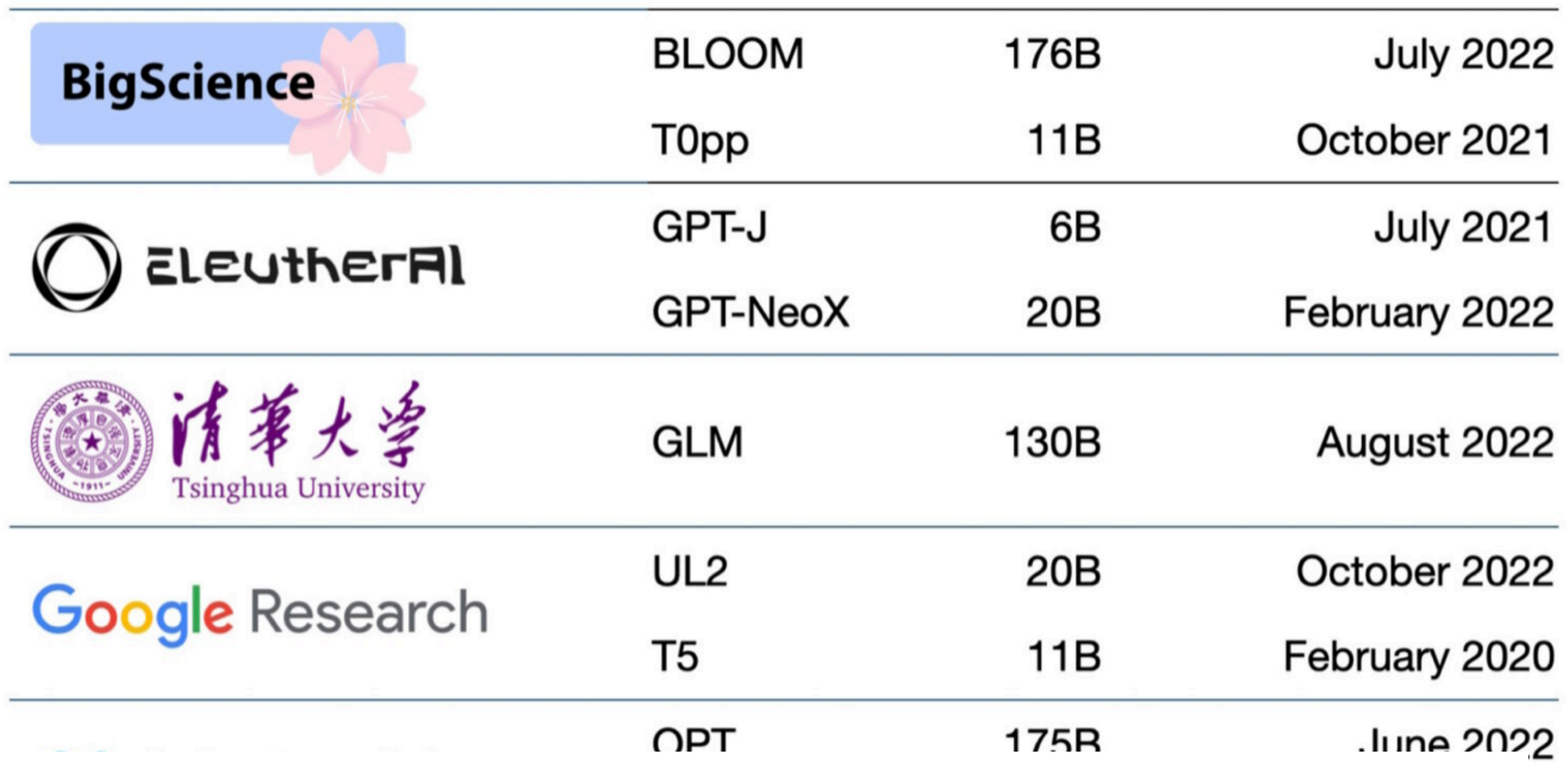
D Xu et al.

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….







BigScience	BLOOM	176B	July 2022
	T0pp	11B	October 2021
EleutherAI	GPT-J	6B	July 2021
	GPT-NeoX	20B	February 2022
清華大學 Tsinghua University	GLM	130B	August 2022
Google Research	UL2	20B	October 2022
	T5	11B	February 2020
	OPT	175B	June 2022

The New York Times

A.I. and Chatbots › How the A.I. Race Began One Year of ChatGPT Key Figures in A.I. How A.I. Could Be Regulated

THE SHIFT

Maybe We Will Finally Learn More About How A.I. Works

Stanford researchers have ranked 10 major A.I. models on how openly they operate.

AI Explainer: Foundation models and the next era of AI

Published March 23, 2023

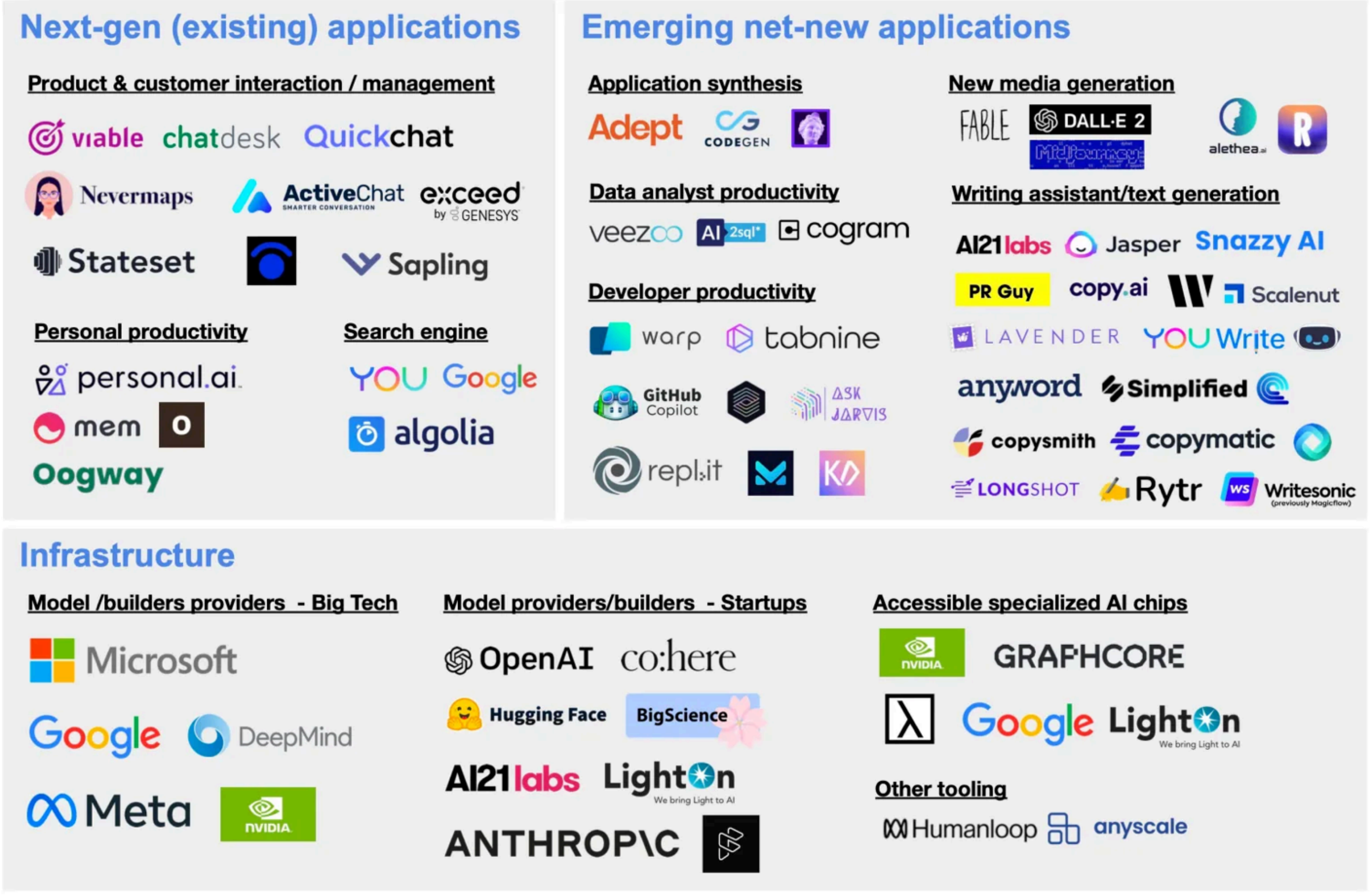
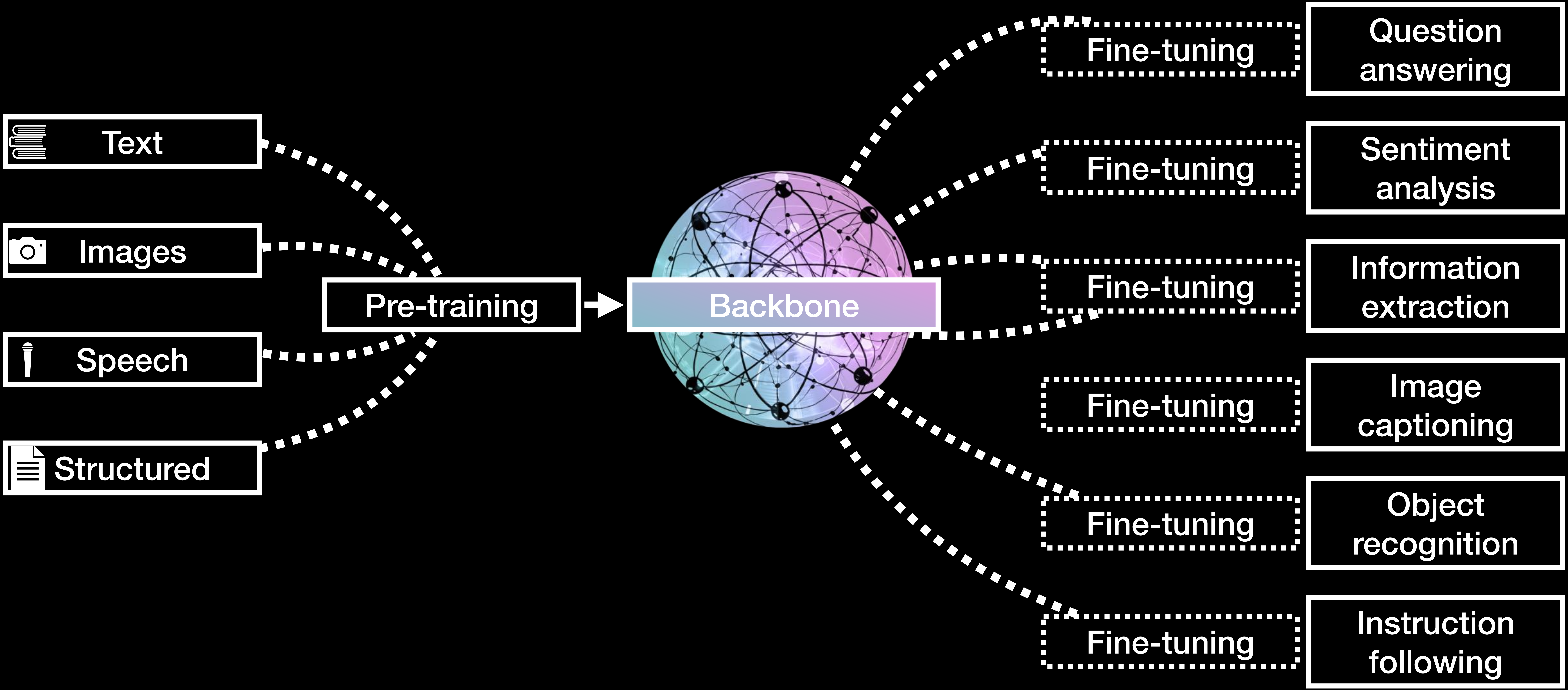


Figure 5: Representative sample of companies that have publicly stated that they are using, building, or enabling


Live


Live

Foundation Models



Text

Images

Speech

Structured

Pre-training

Backbone

Fine-tuning

Question answering

Fine-tuning

Sentiment analysis

Fine-tuning

Information extraction

Fine-tuning

Image captioning

Fine-tuning

Object recognition

Fine-tuning

Instruction following

Heterogeneous detector
Multi-modal input!

Pre-training

Backbone

Fine-tuning
Jet reconstruction

Fine-tuning
Electron reconstruction

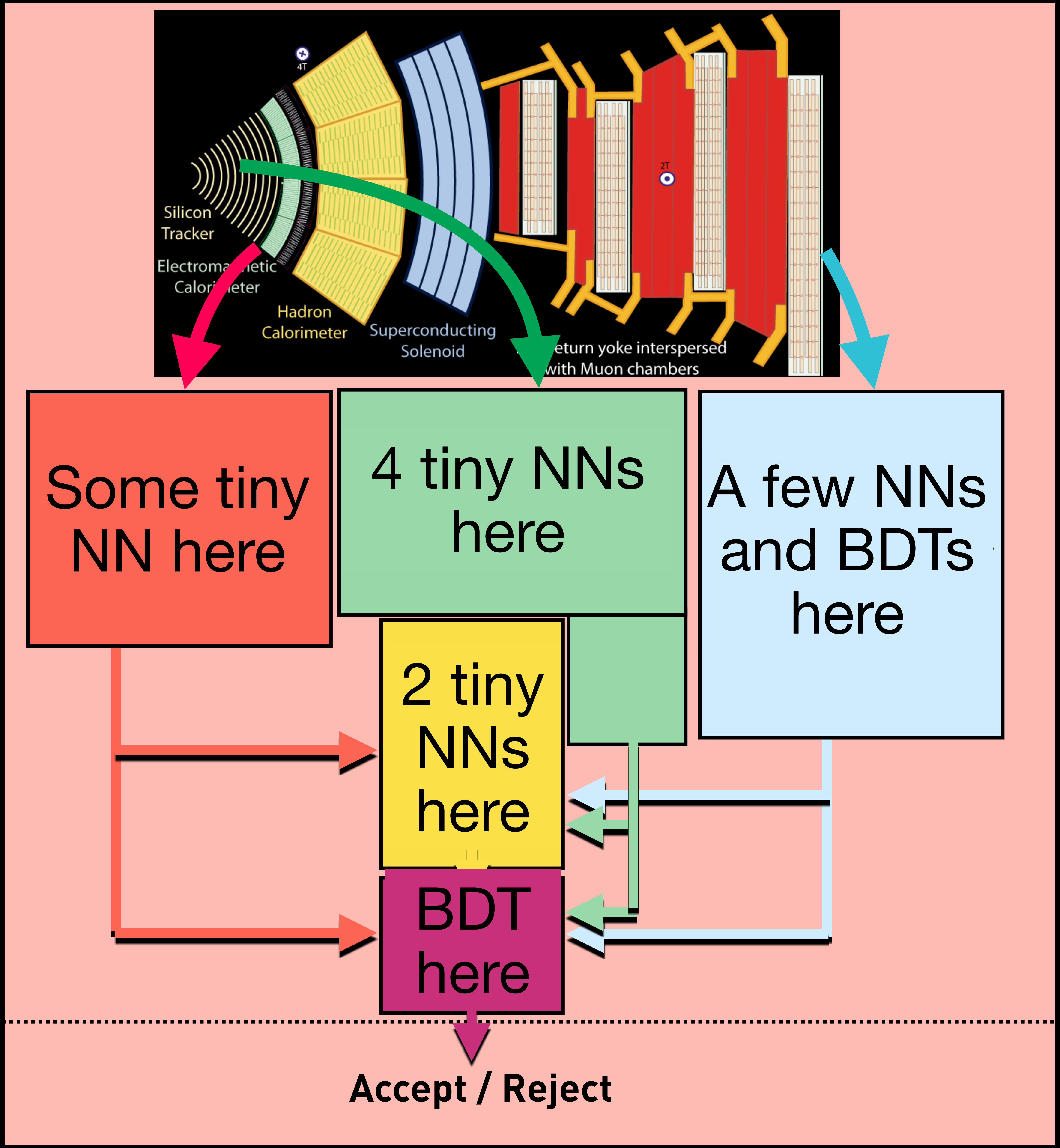
Fine-tuning
Pile-up removal

Fine-tuning
Missing energy computation

Fine-tuning
Anomaly Detection

Fine-tuning
0/1?

(Generate simulation?)



Some tiny NN here

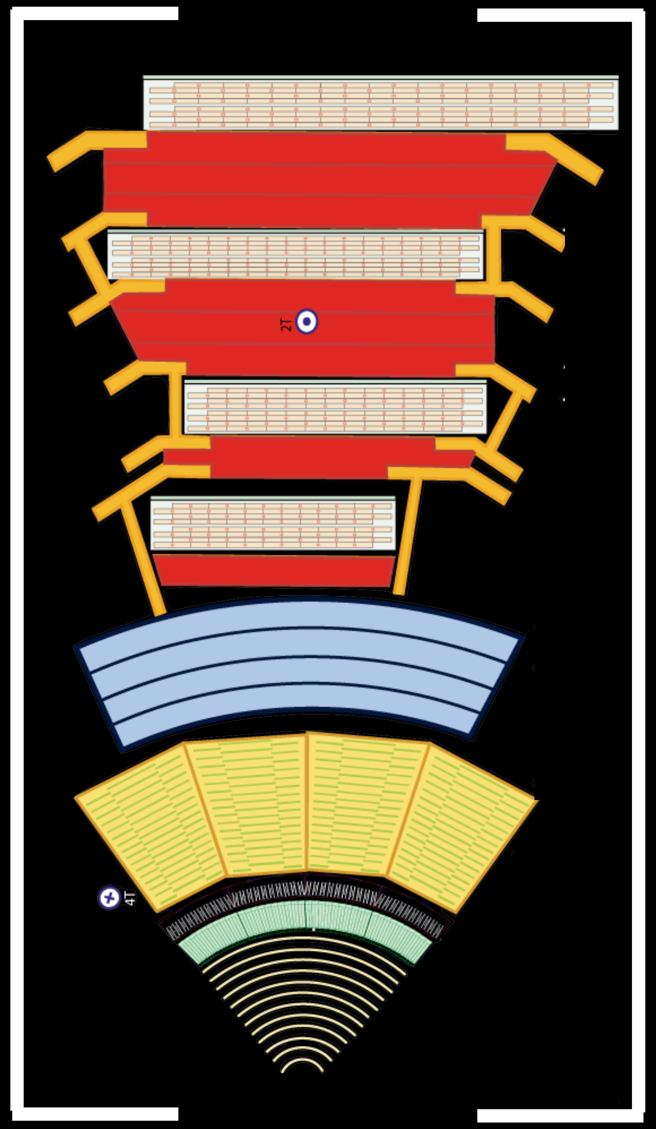
4 tiny NNs here

A few NNs and BDTs here

2 tiny NNs here

BDT here

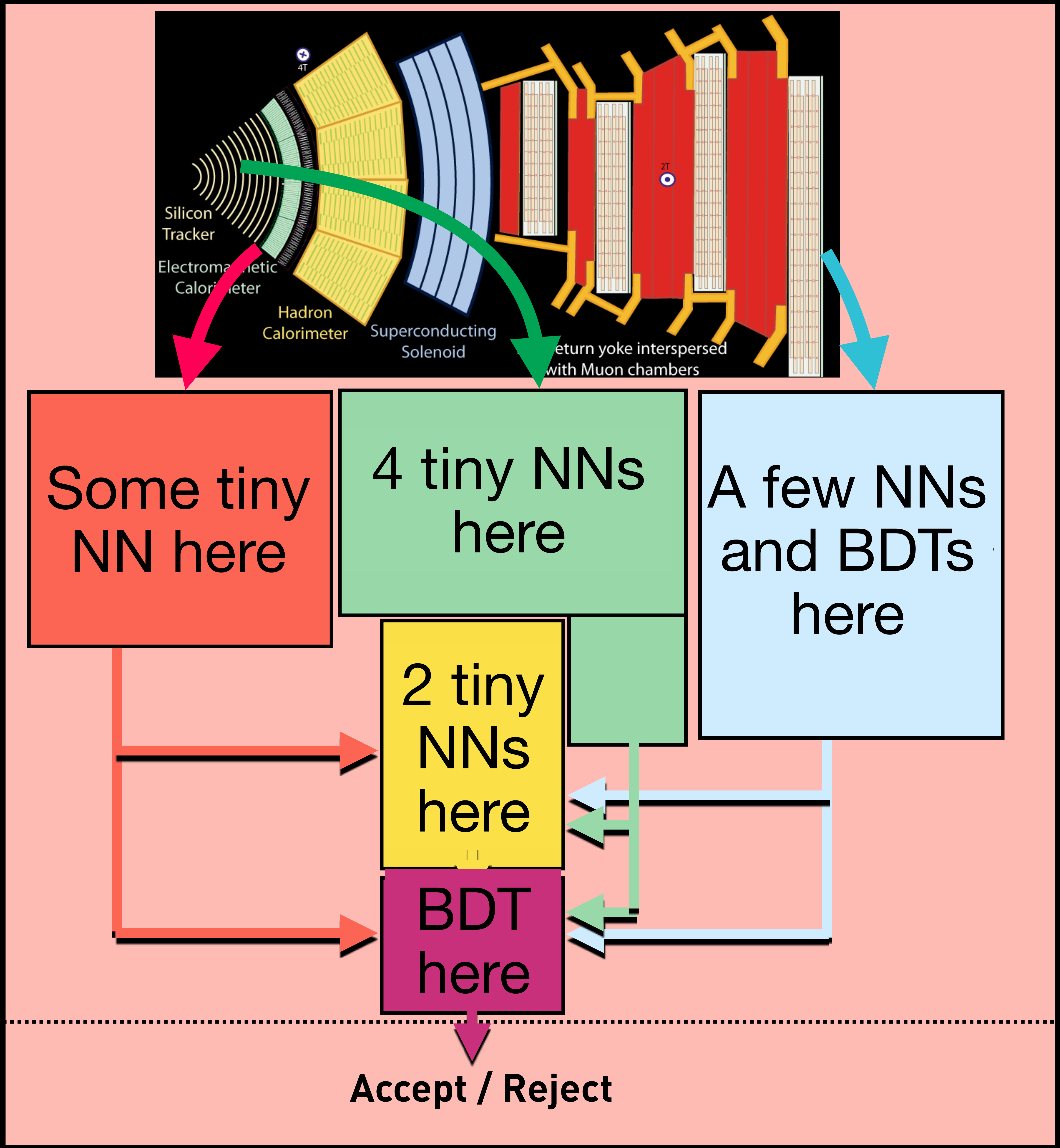
Accept / Reject



$$\Rightarrow f(x; w^*) \Rightarrow \hat{y}$$

$$x = (x_1, x_2, \ldots,)$$

Too many models, too little learning?



Some tiny NN here

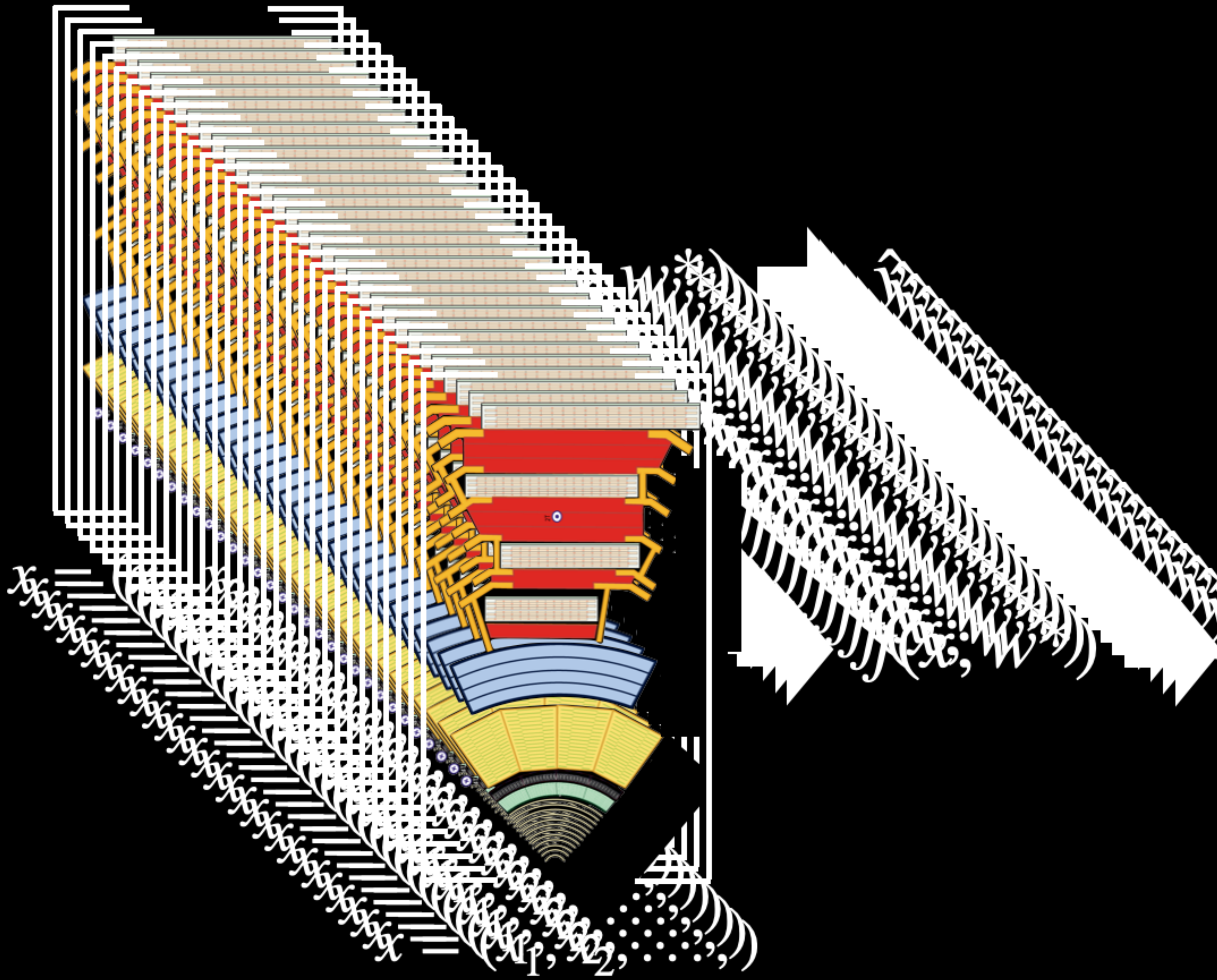
4 tiny NNs here

A few NNs and BDTs here

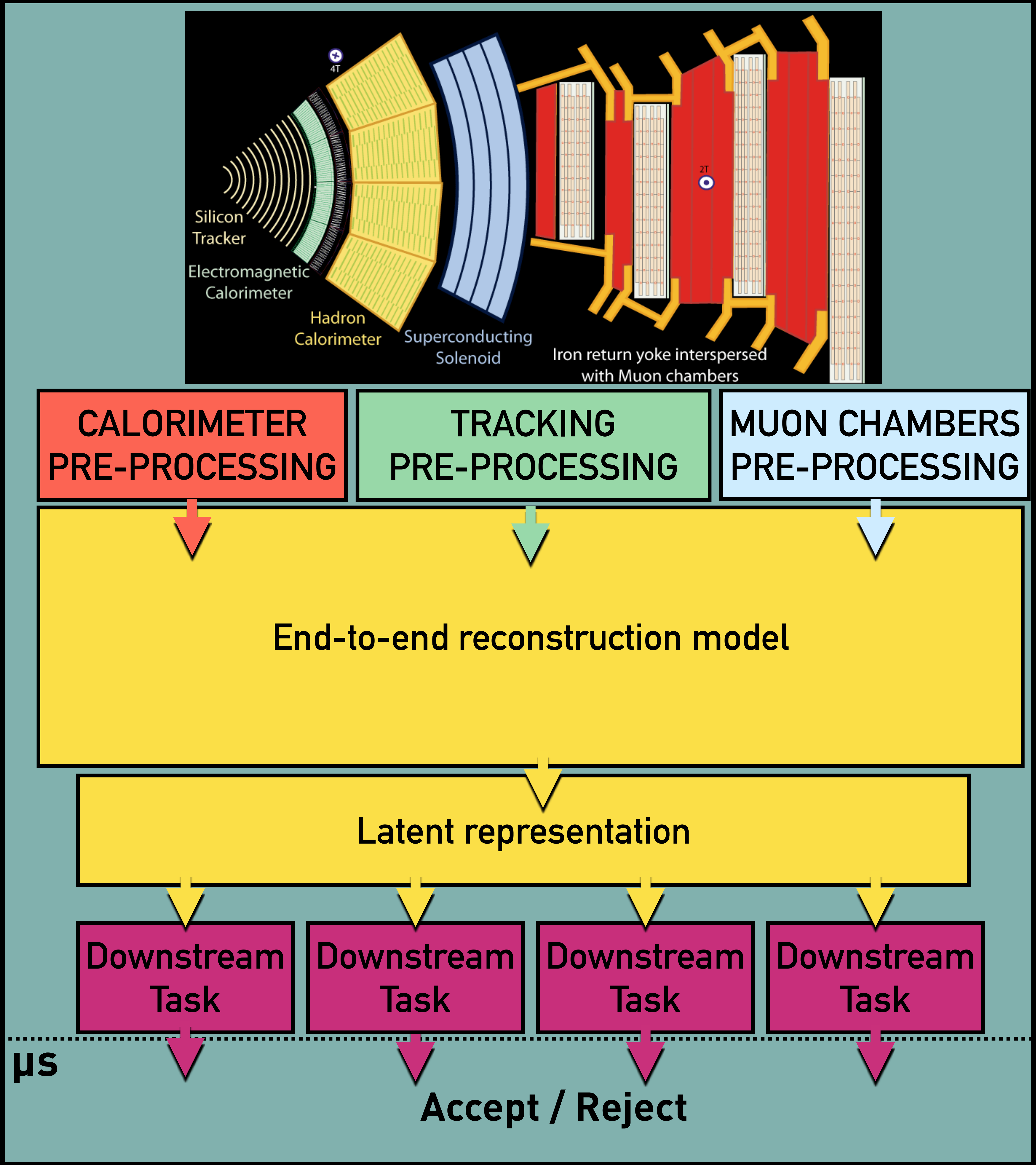
2 tiny NNs here

BDT here

Accept / Reject



One model, learn neural embedding?



CALORIMETER PRE-PROCESSING

TRACKING PRE-PROCESSING

MUON CHAMBERS PRE-PROCESSING

End-to-end reconstruction model

Latent representation

Downstream Task

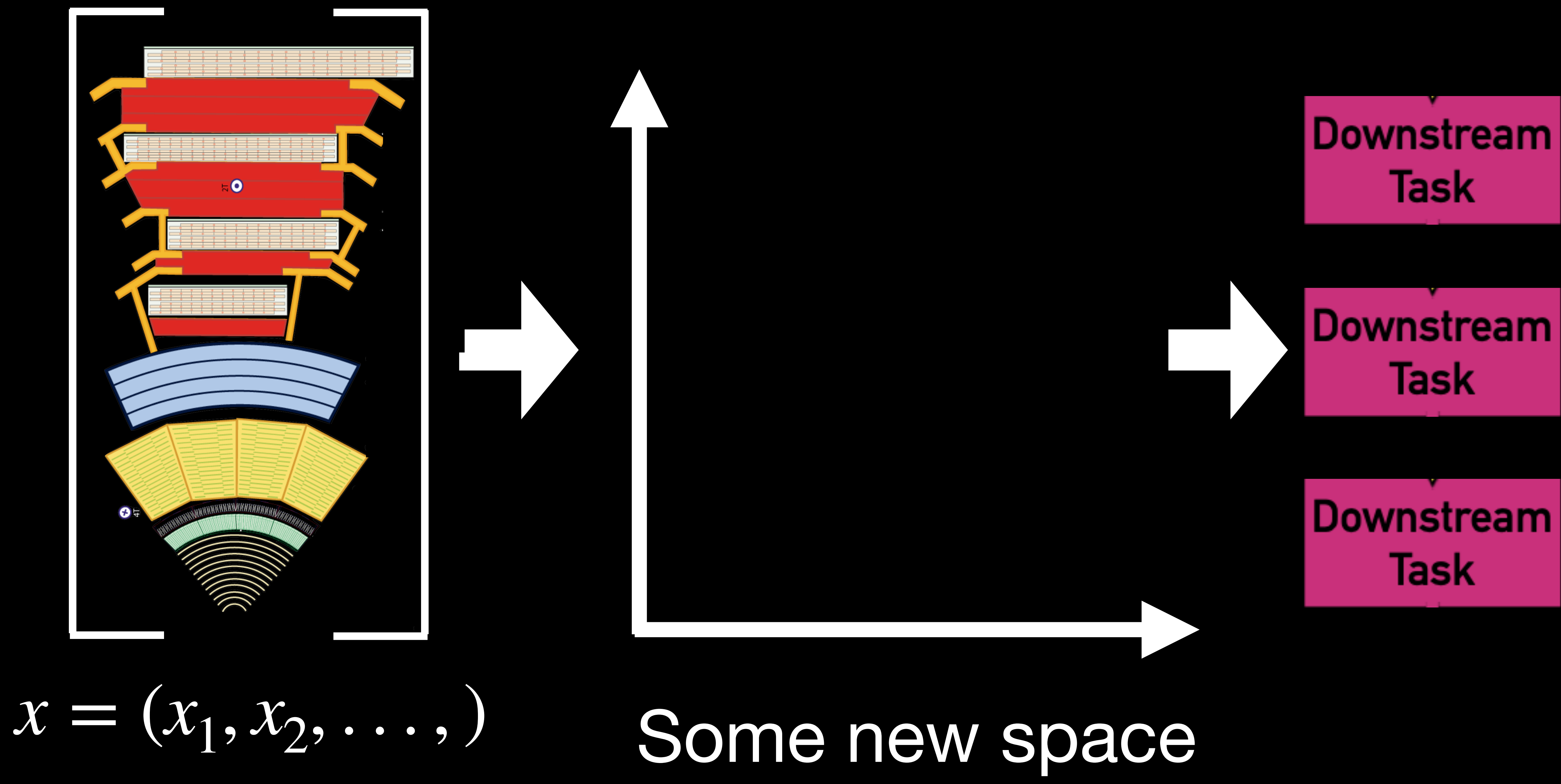
Downstream Task

Downstream Task

Downstream Task

μs

Accept / Reject



$$x = (x_1, x_2, \ldots,)$$

Some new space

Downstream Task

Downstream Task

Downstream Task


A3D3 Institute
hls4ml
Streaming data rate [B/s]
10^{19}
10^{17}
10^{15}
10^{13}
10^{11}
FPGA/ASIC
1 TB/yr
1 PB/yr
1 EB/yr
CPU/GPU
LHC HLT
Google Cloud
LHC L1T
DUNE
Neuro
Netflix 4K UHD
10^{-8}
10^{-6}
10^{-4}
10^{0}
10^{4}
10^{6}
Latency requirement [s]
Anchor
Negative Samples
QCD
Z(qq)+jets
W(qq)+jets
ttbar
Augmentation
QCD
QCD (Changed shower setting)
Dijet invariant mass (GeV)
CMS
CMS Simulation Preliminary
$t\bar{t}$ + PU, $\sqrt{s}$ = 14 TeV
Machine-Learned Particle Flow reconstruction
Charged hadrons
HFEM
Neutral hadrons
Electrons
Photons
Muons
HFHAD
AXOL1T
Signals and backgrounds
FPGA

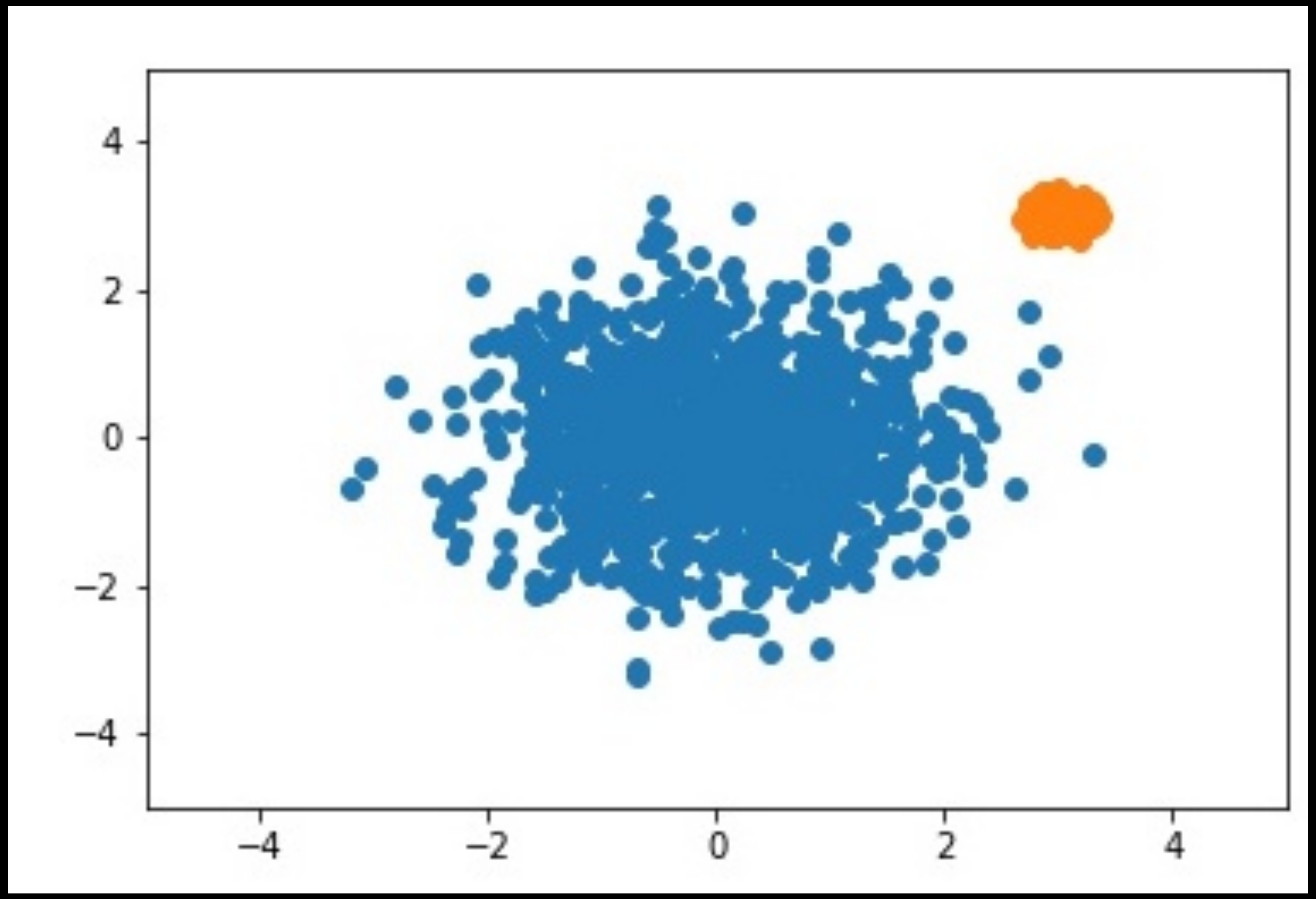
thea.aarrestad@cern.ch

Backup

Types of anomaly detection

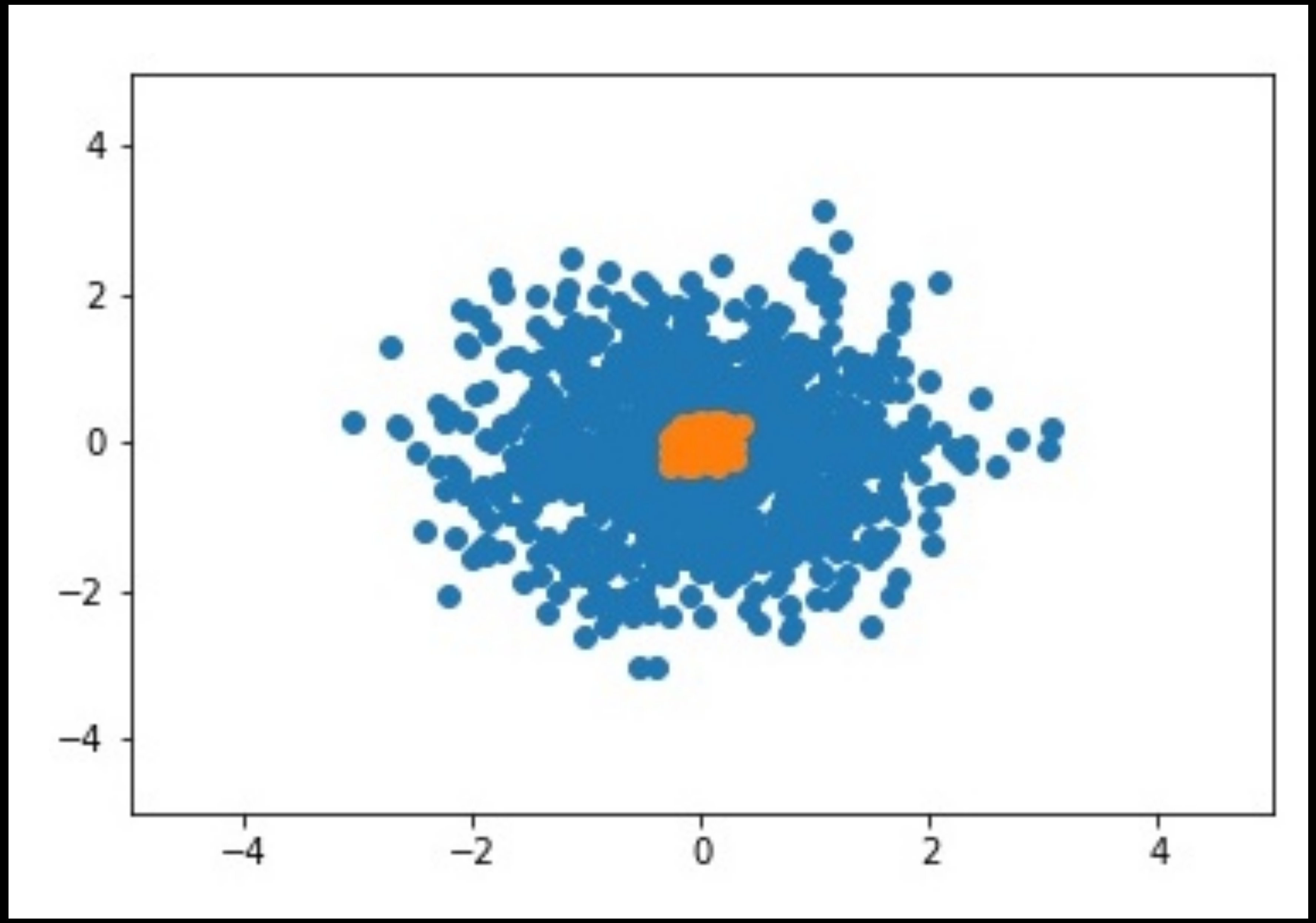
Outlier detection

Find (non-resonant) out-of-distribution datapoints



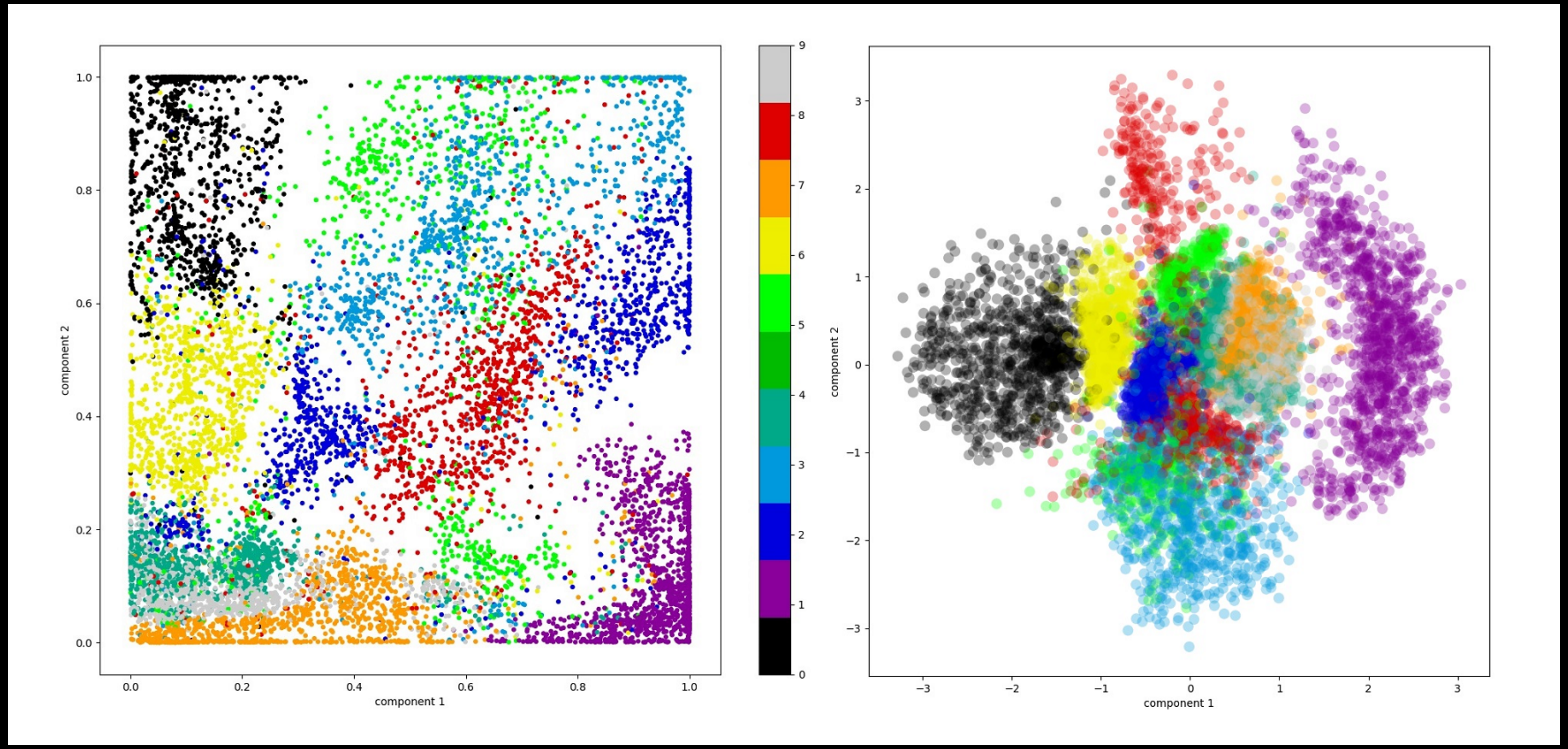
Detecting overdensities

Find (resonant) overdensities in distributions



Types of anomaly detection

Outlier detection



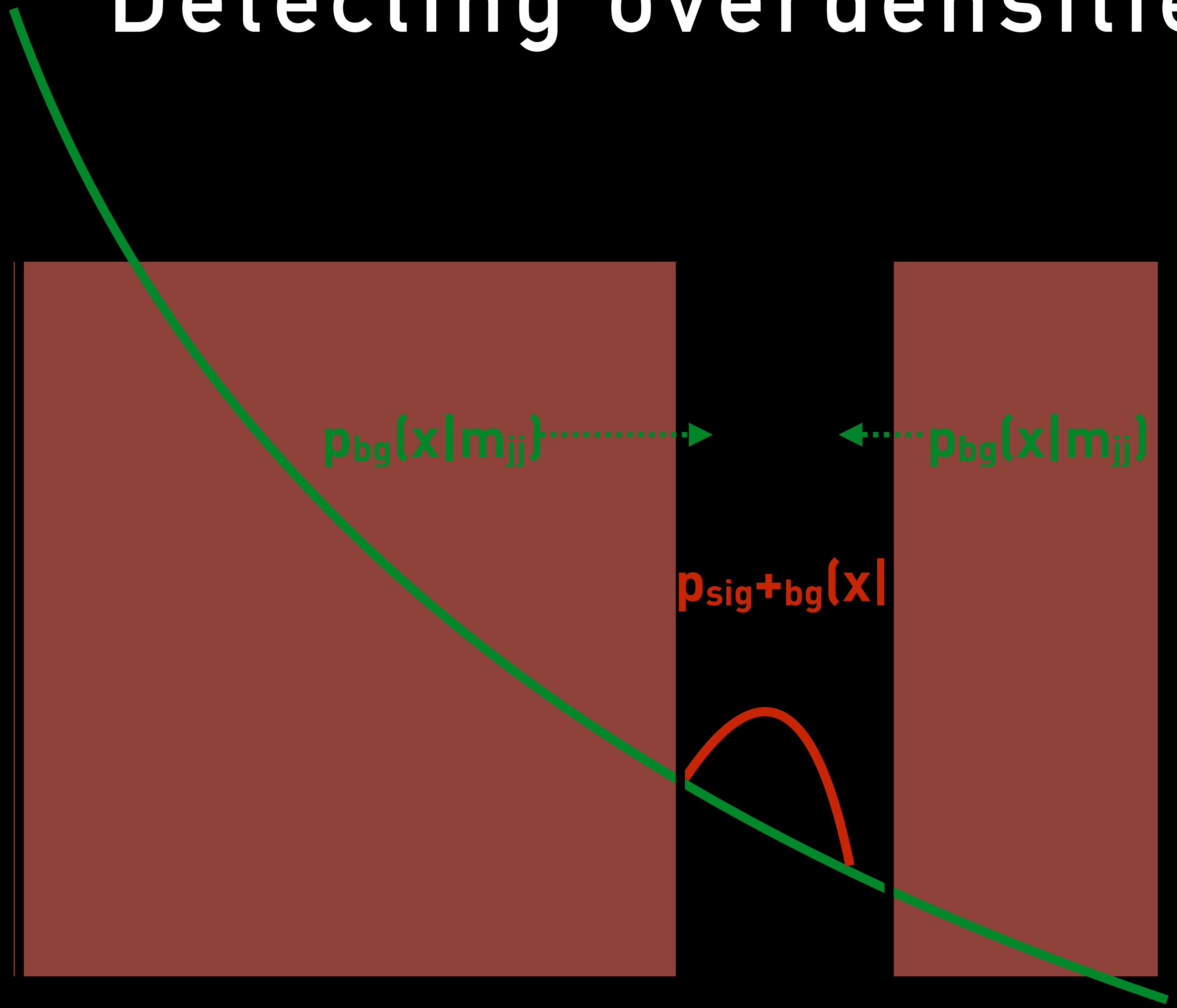
Non-resonant, tail of distributions

- Often (variational) auto-encoders
- Useful for triggering!

Caveats

- What's a good metric for optimisation?
- How to use selected events in analysis?

Detecting overdensities



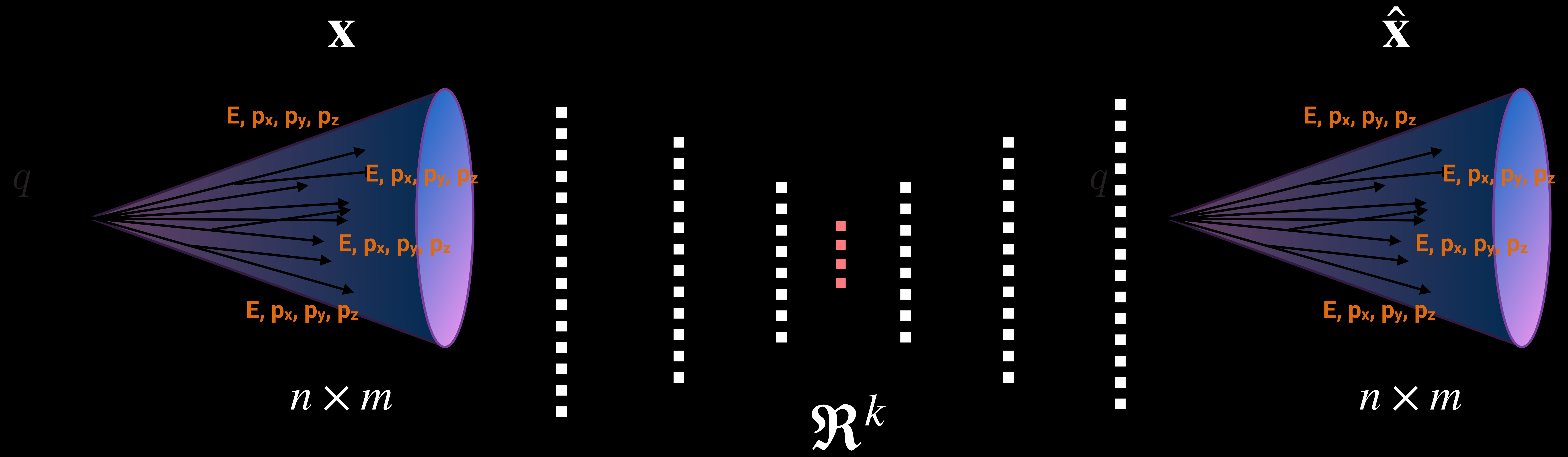
Resonant, similar to a bump hunt

- Density estimation methods
- Useful for offline analysis

Caveats

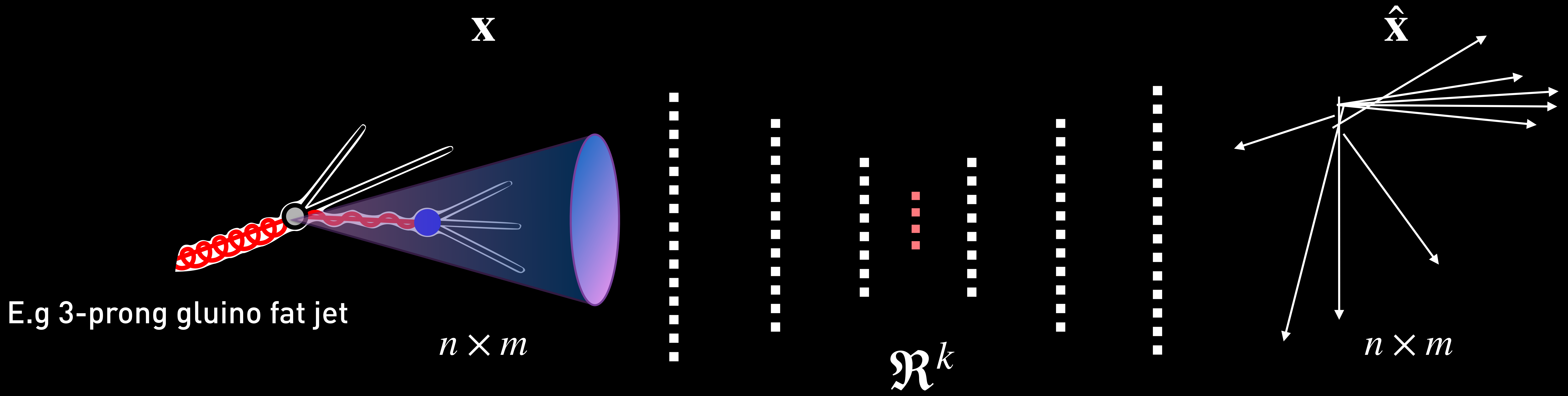
- Relies on a definition of "sideband" and a sizeable signal

Outlier detection



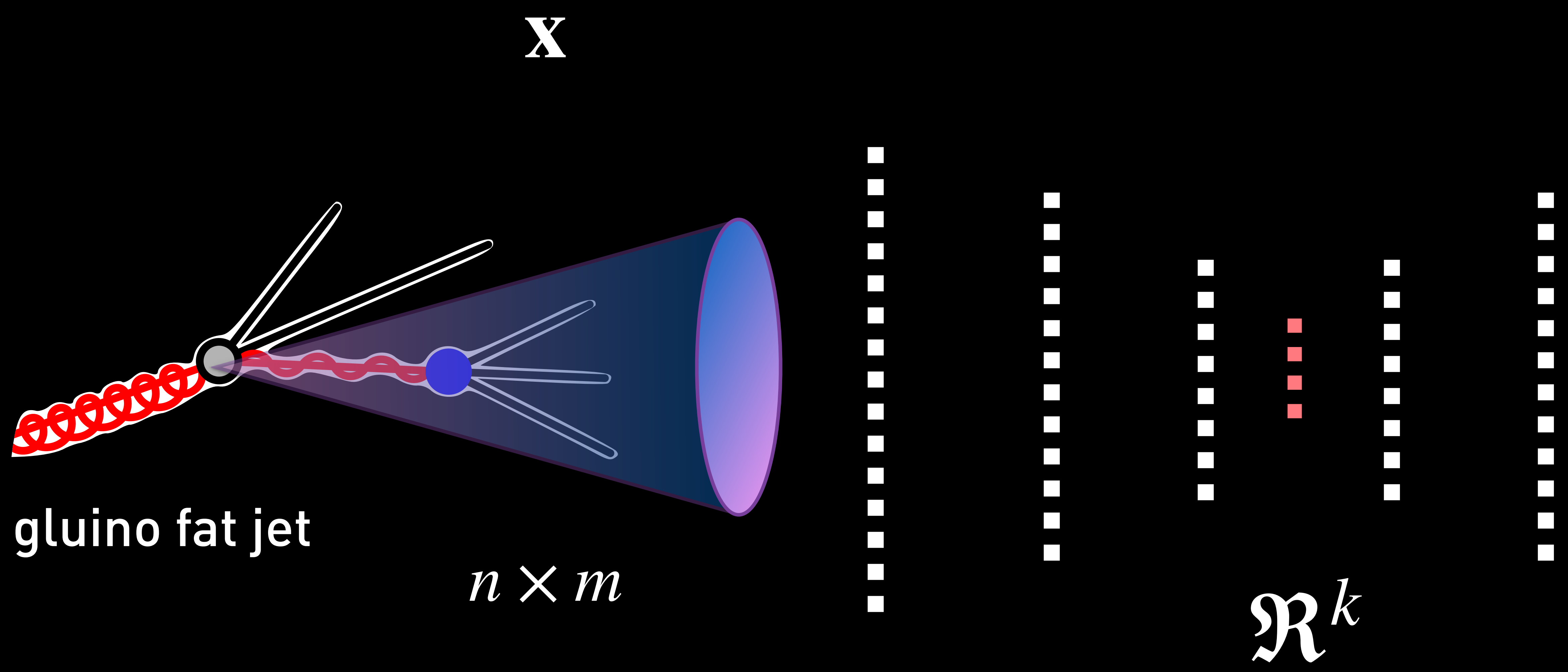
Compressed representation of x.
Latent space $\mathfrak{R}^k$, $k < m \times n$
prevents memorisation of input, must learn

Outlier detection

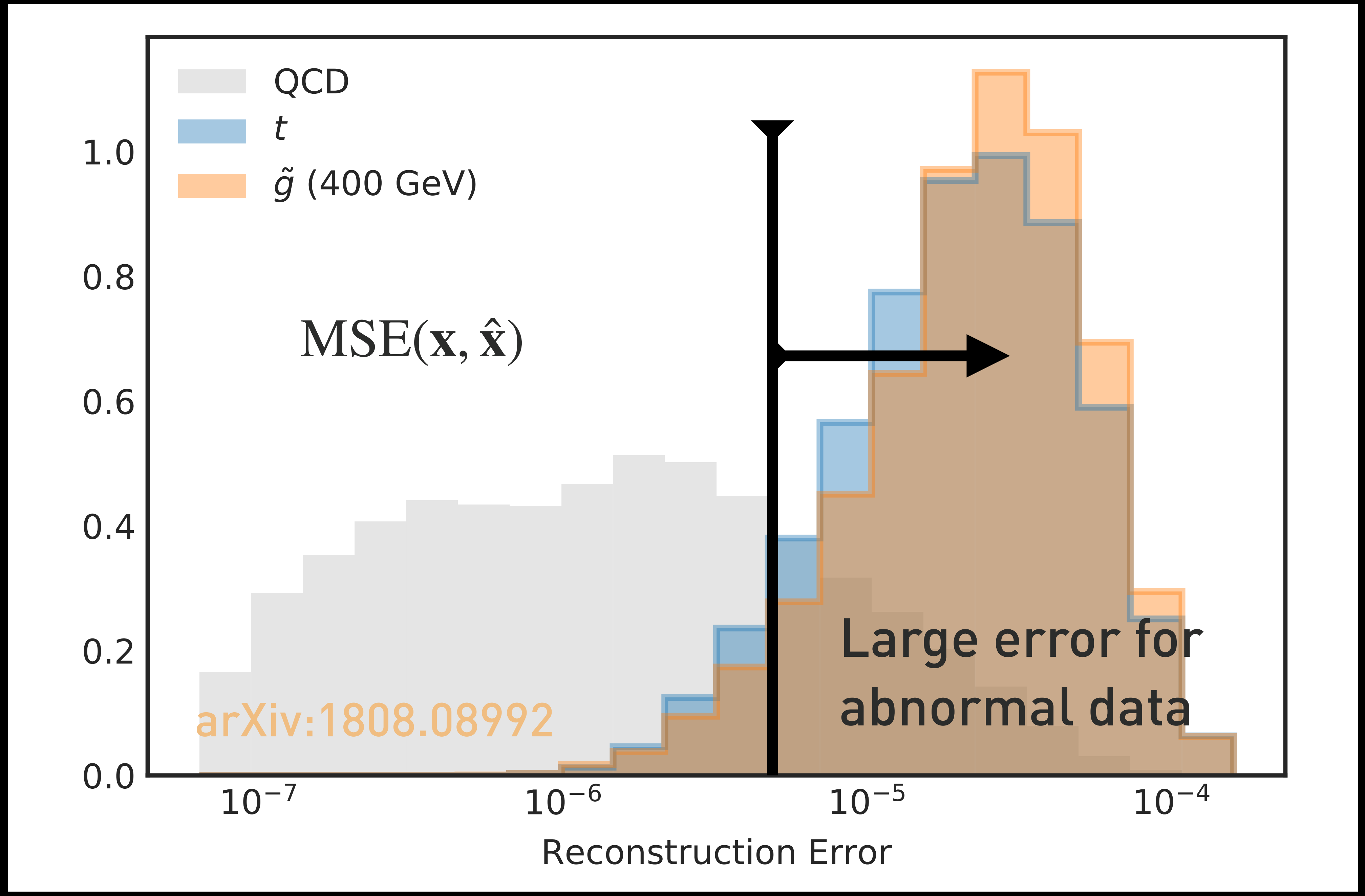


$\mathscr{L}(\mathbf{x}, \hat{\mathbf{x}})$ is Mean Squared Error$(\mathbf{x}, \hat{\mathbf{x}})$, “high error events” proxy for “degree of abnormality”

Outlier detection



E.g 3-prong gluino fat jet

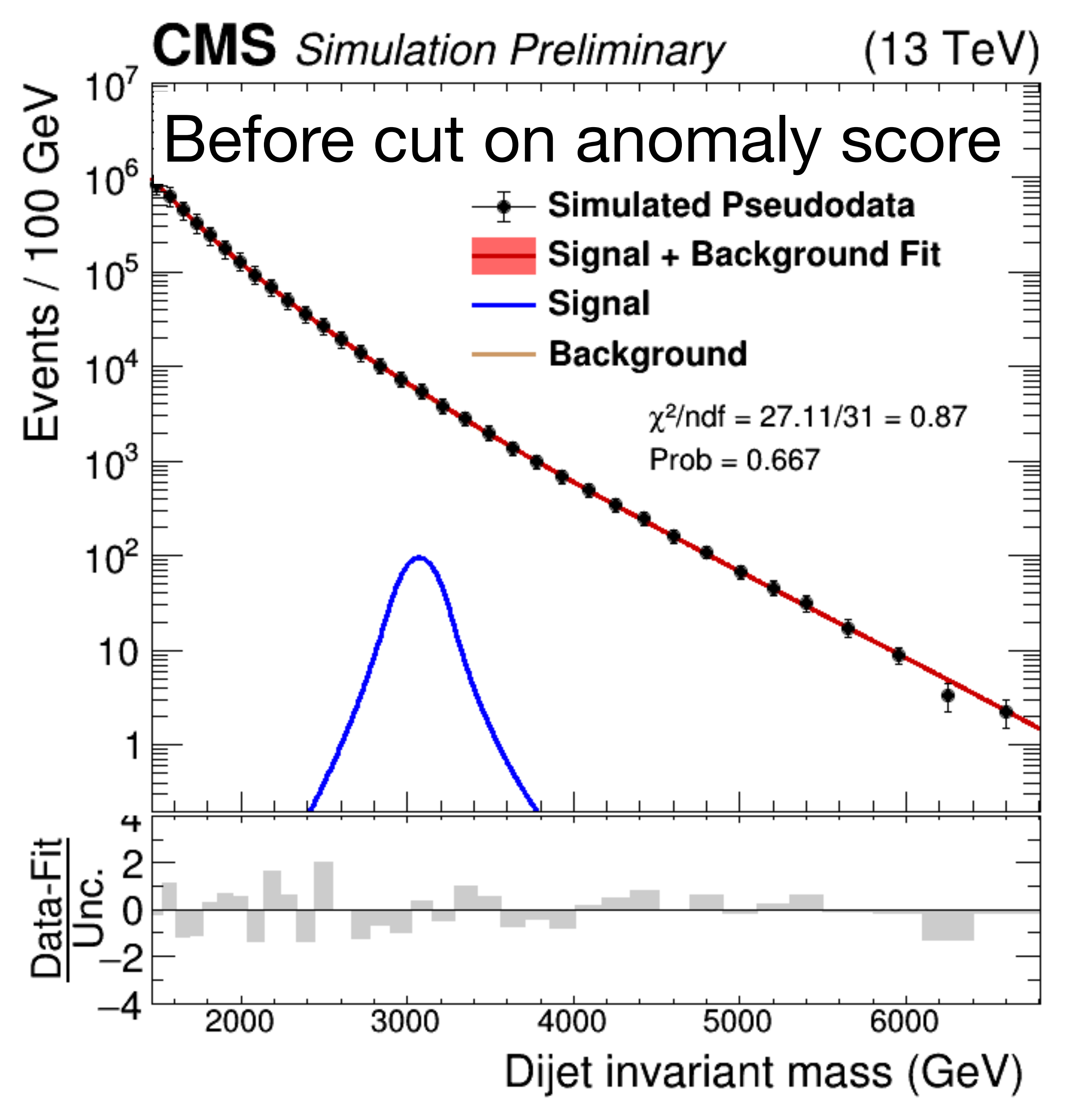


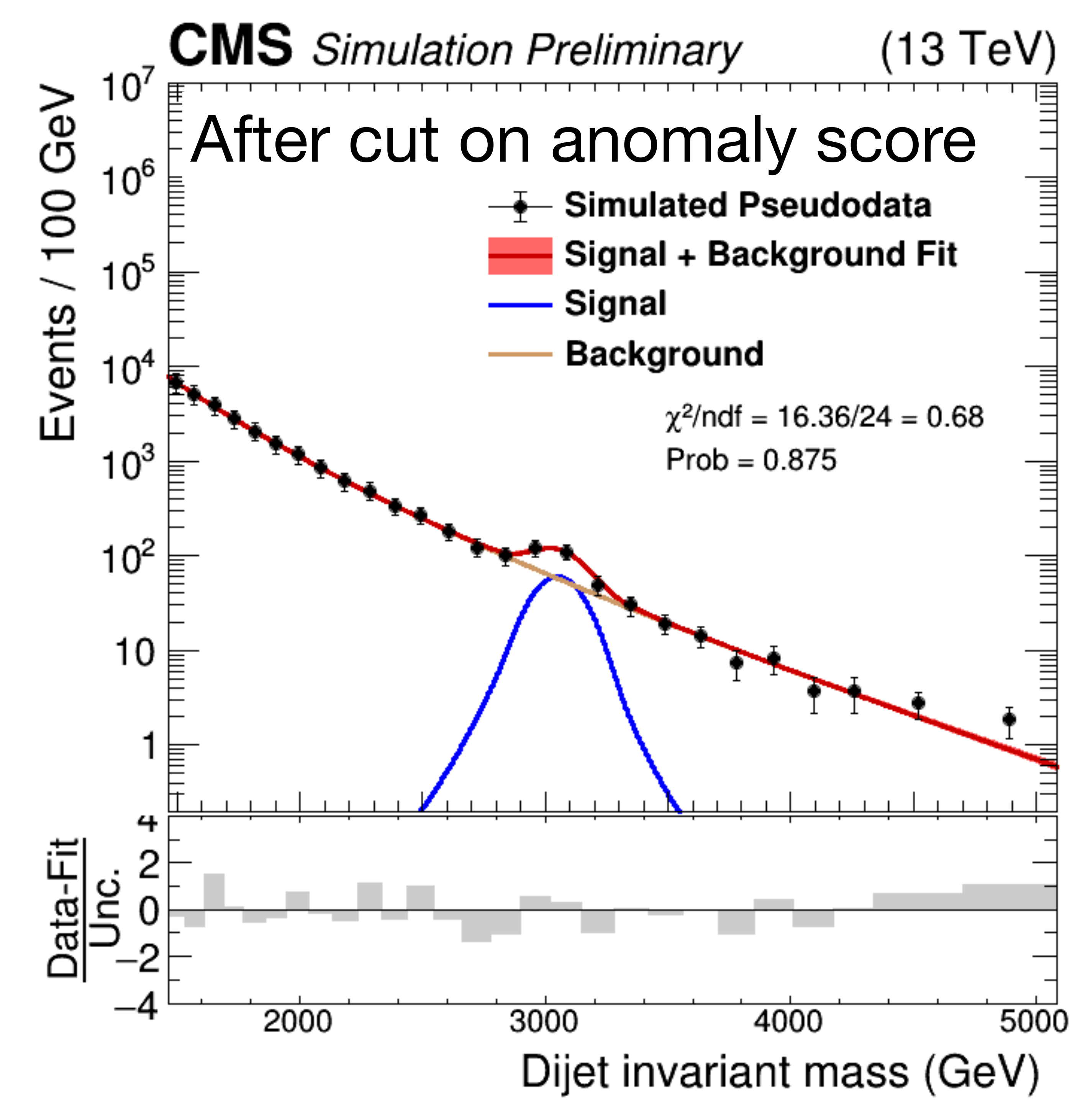
Large error for abnormal data

arXiv:1808.08992

Outlier detection in analysis

E.g CASE

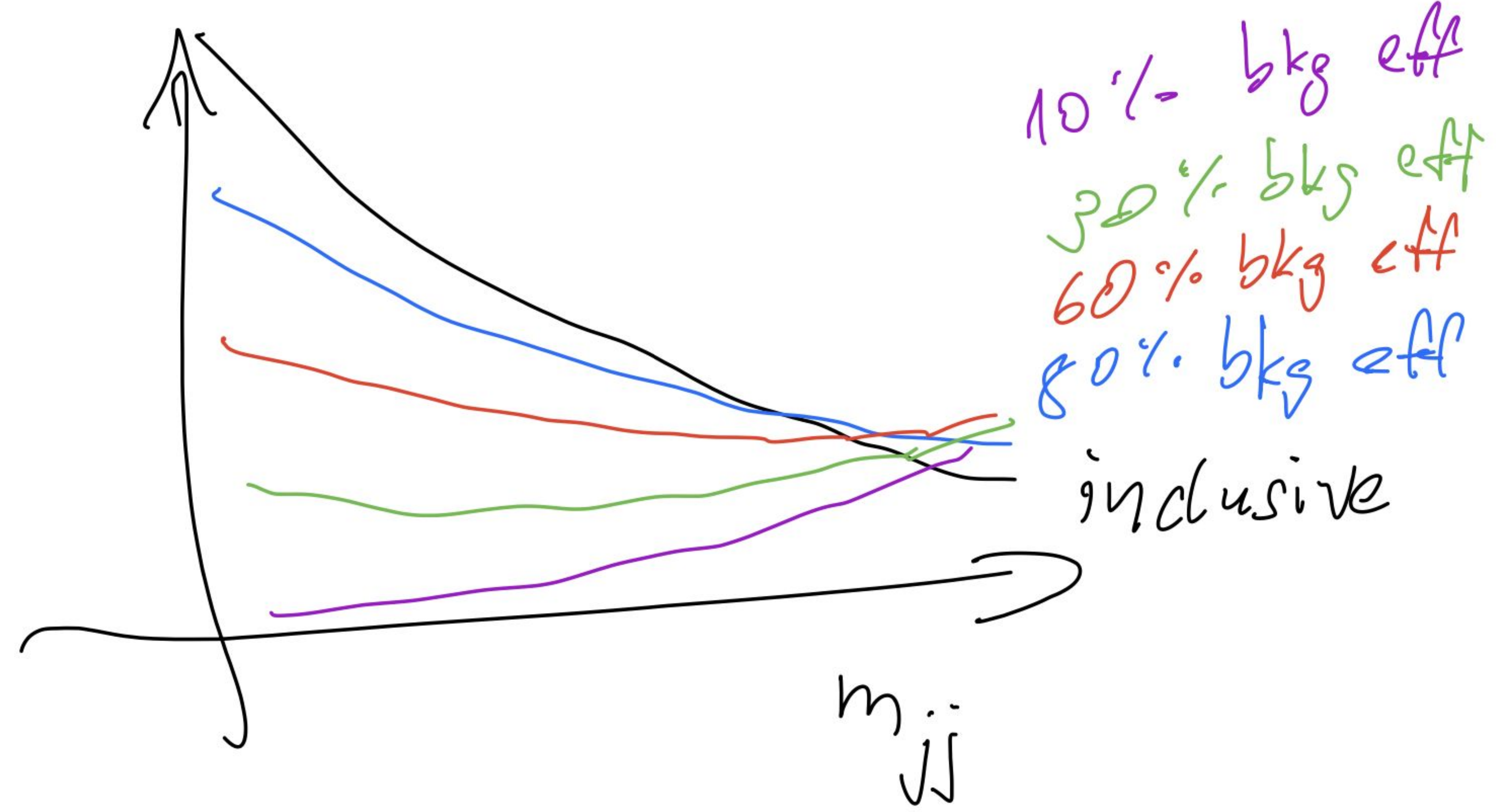




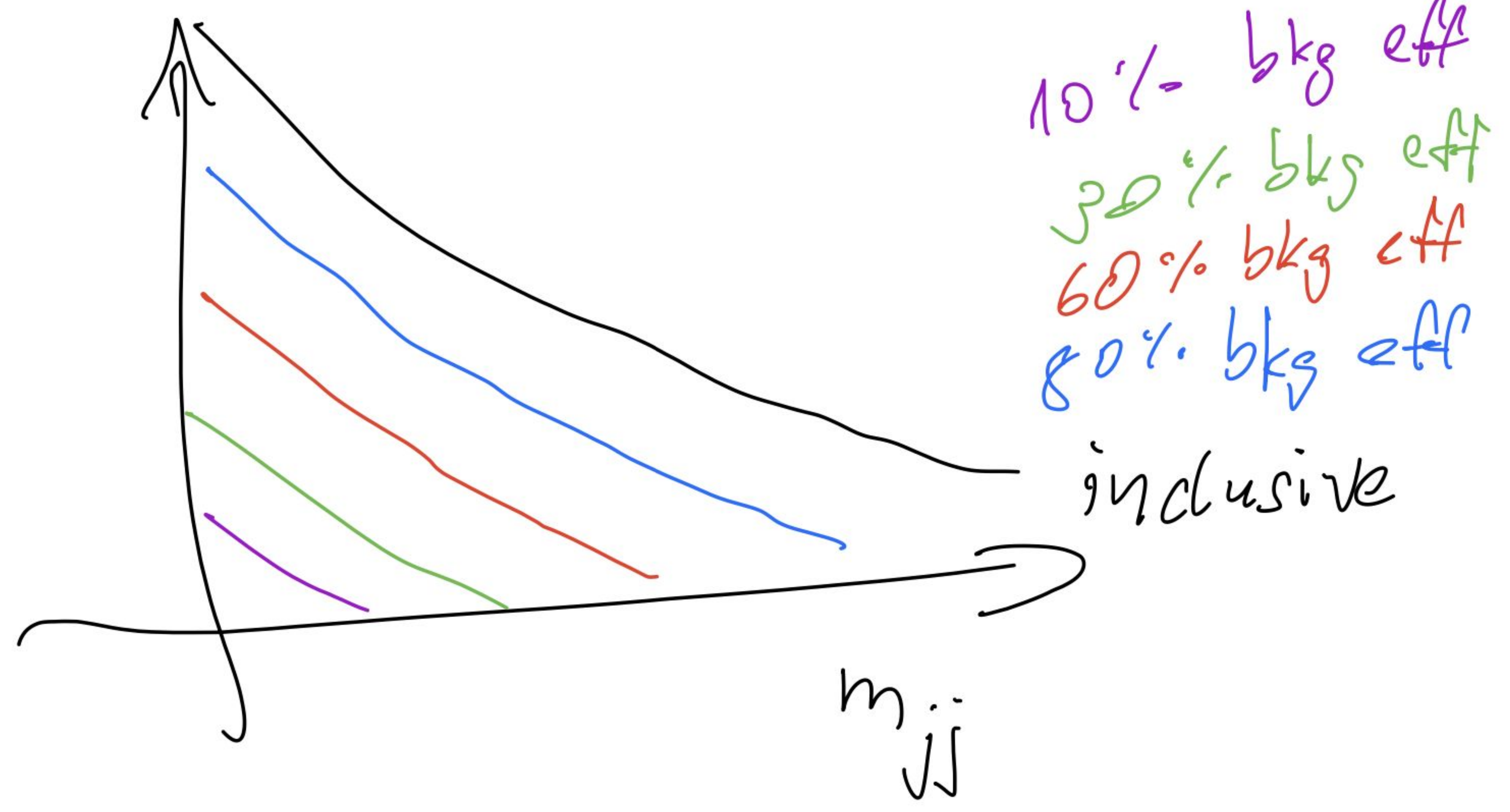
Outlier detection in analysis

E.g CASE

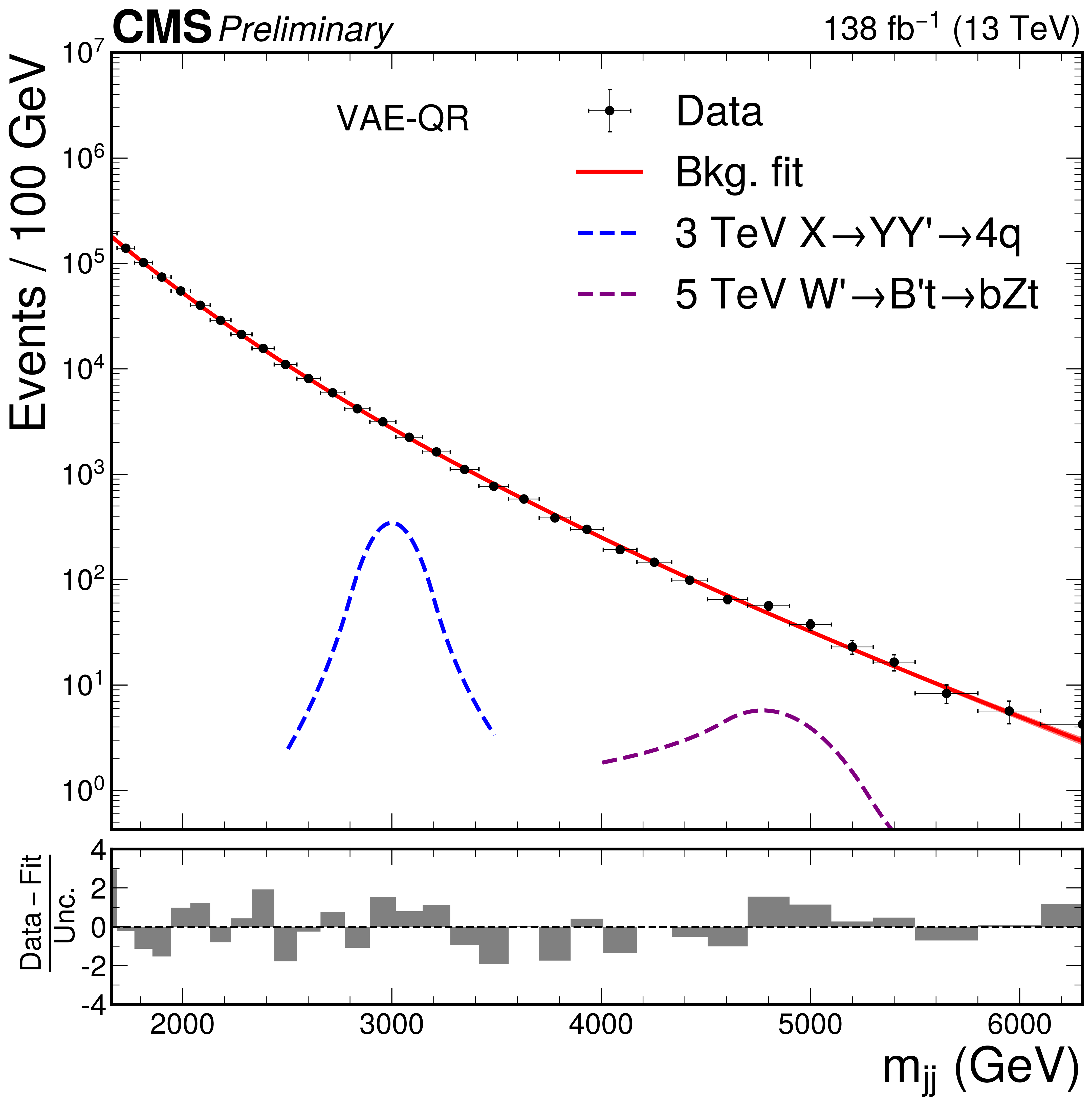
Careful! Cut on score can sculpt spectrum



Can fix using quantile regression



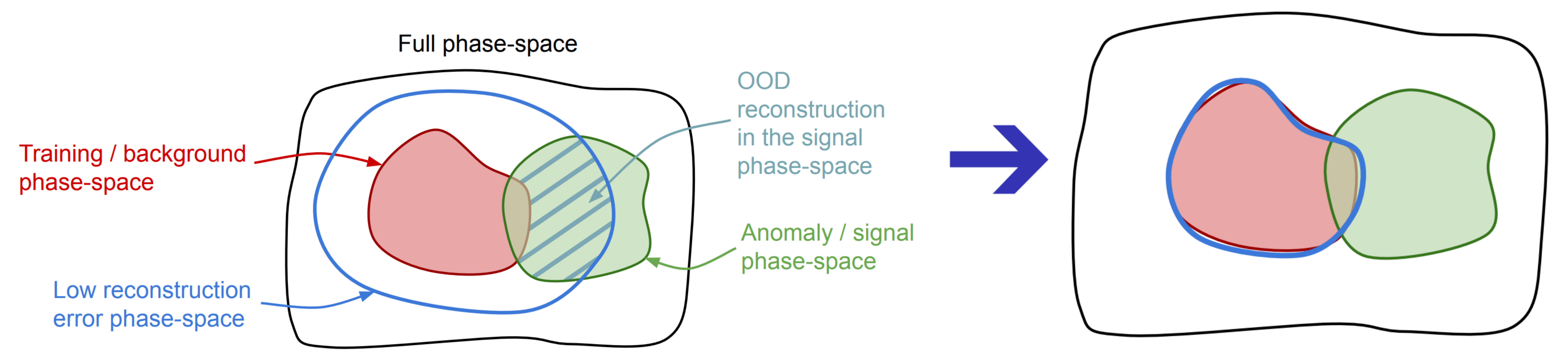
Outlier detection in analysis

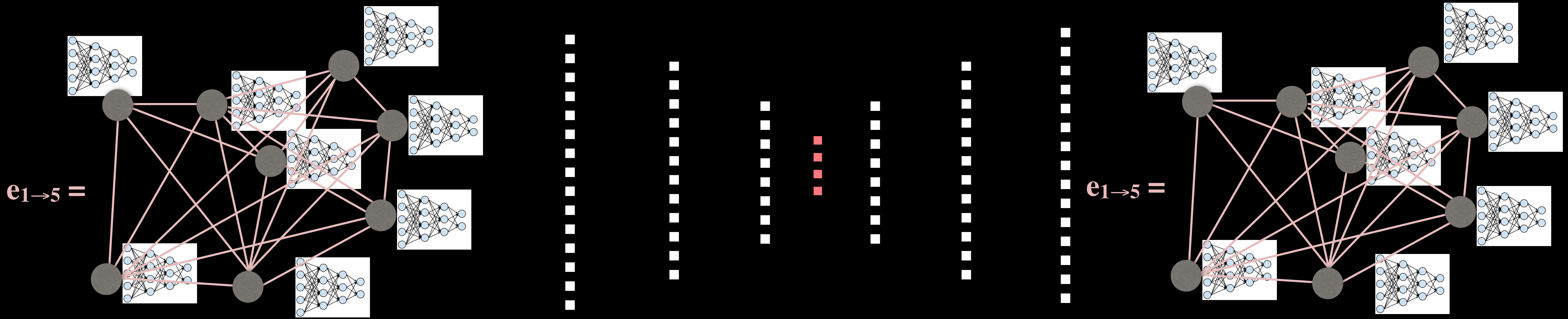


Example for semi-visible jets

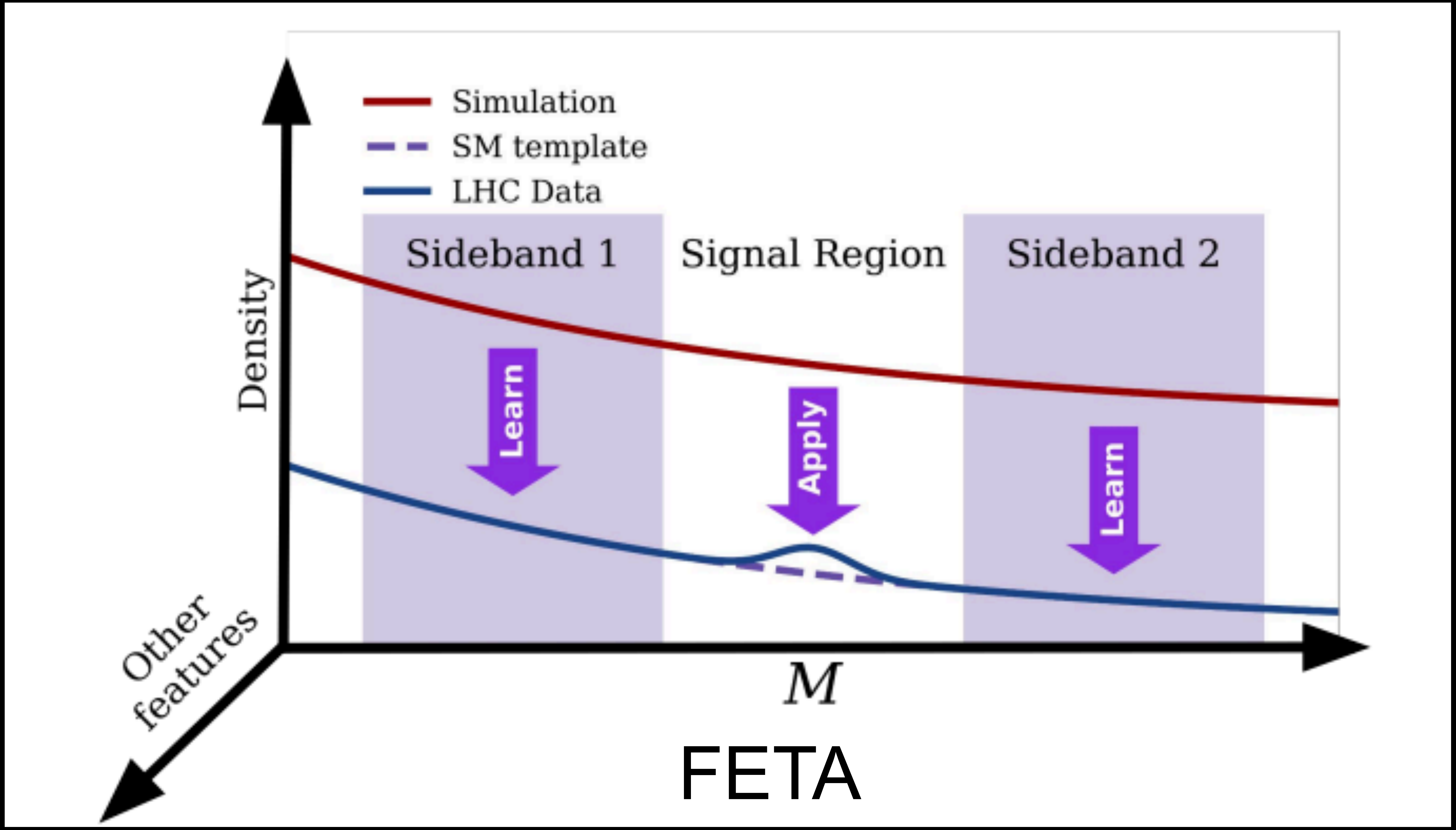
F. Eble: Normalized autoencoders

R. Seidita: Lund Graph autoencoders



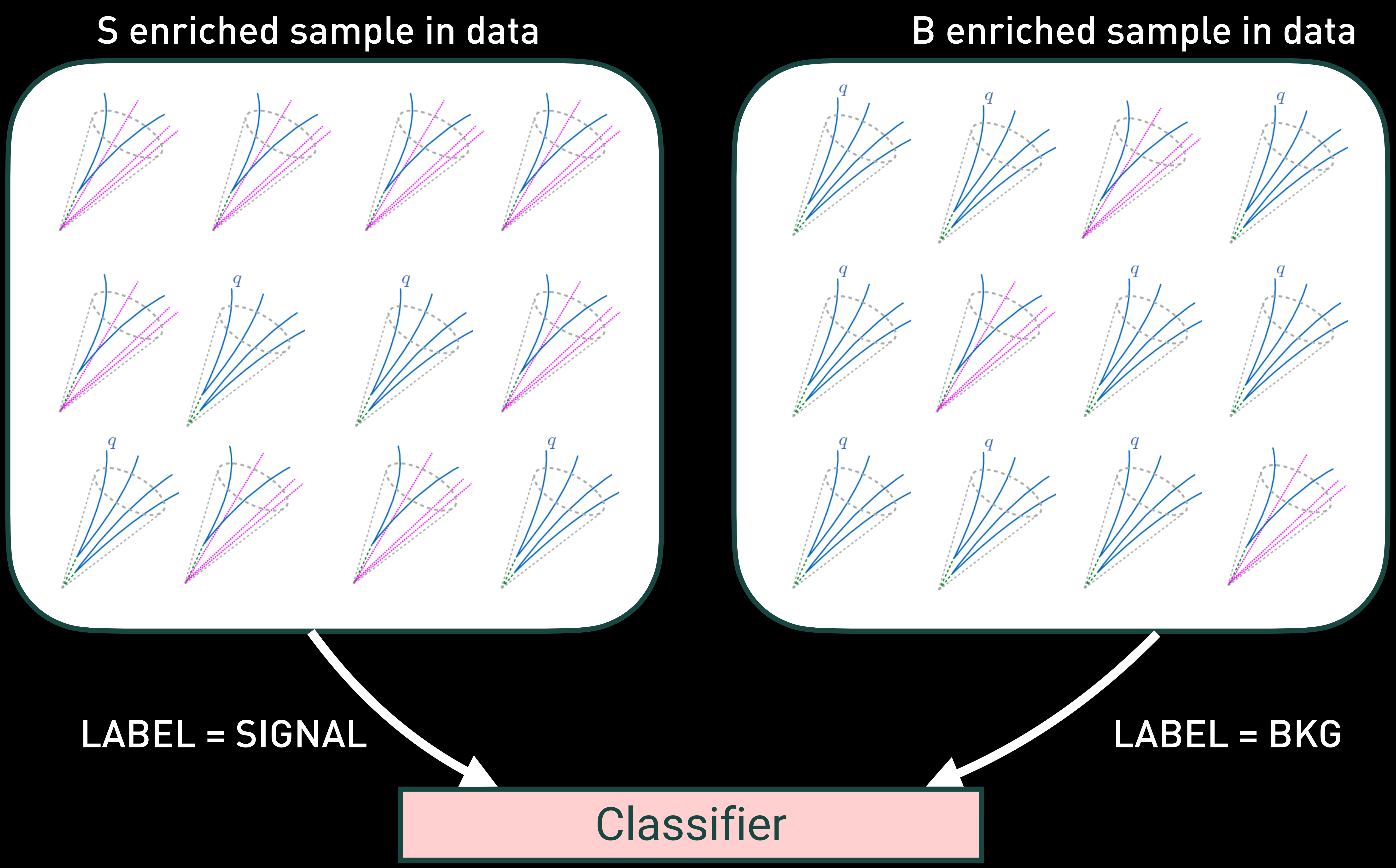


Finding overdensities



FETA

Finding overdensities - CWoLa bumphunt



S enriched sample in data

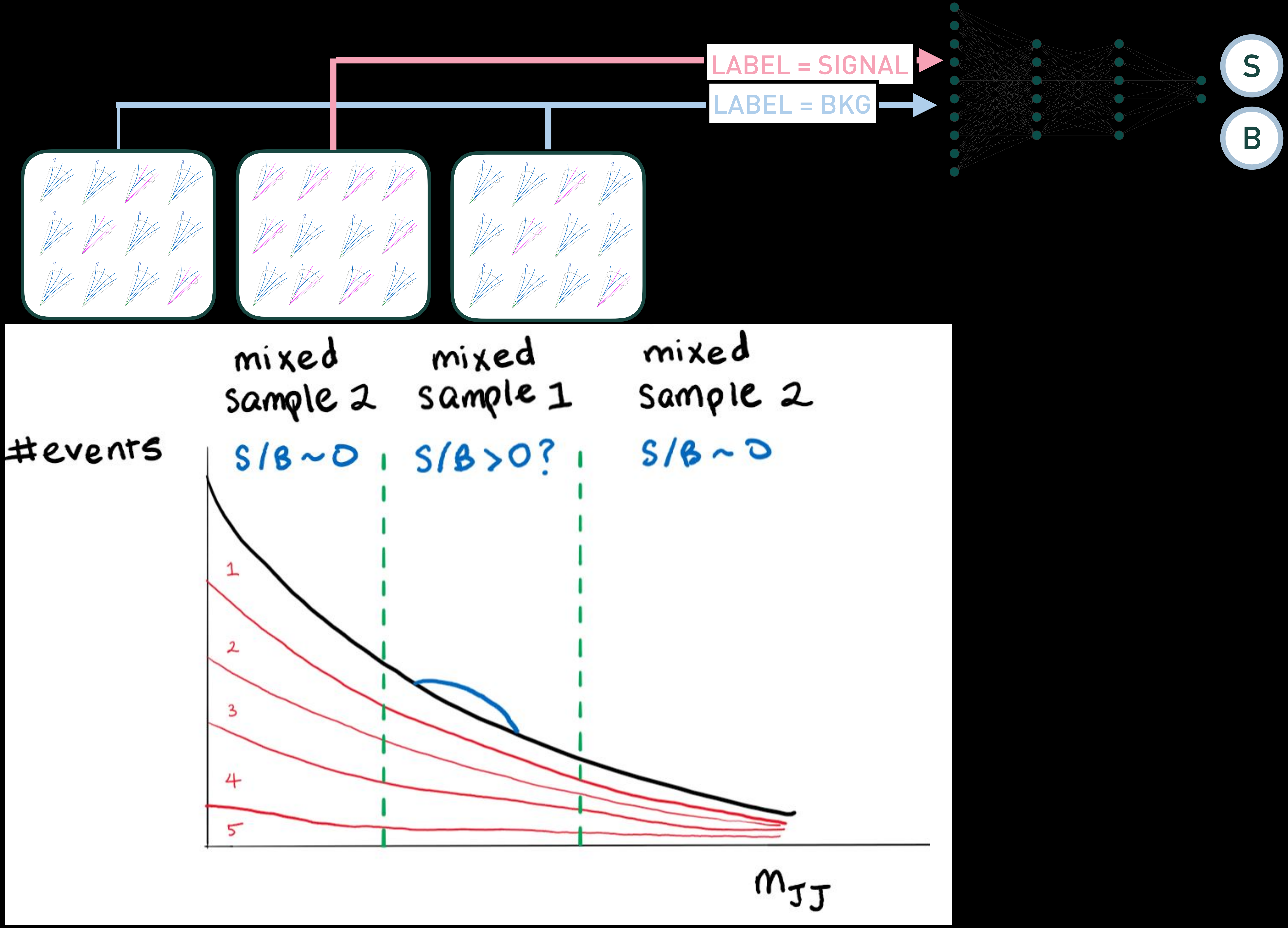
B enriched sample in data

LABEL = SIGNAL

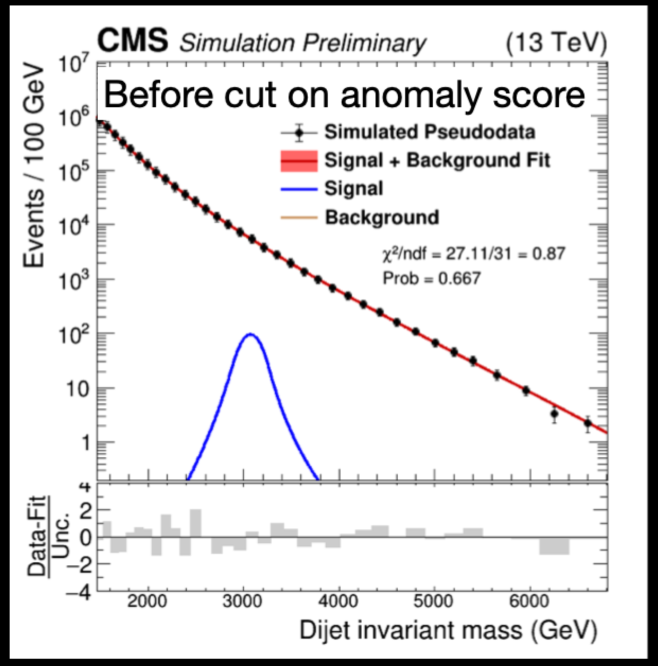
LABEL = BKG

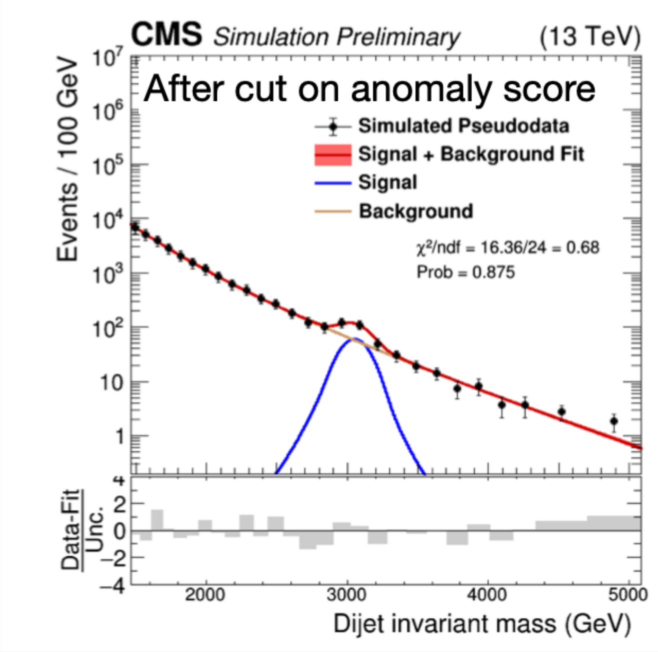
Classifier

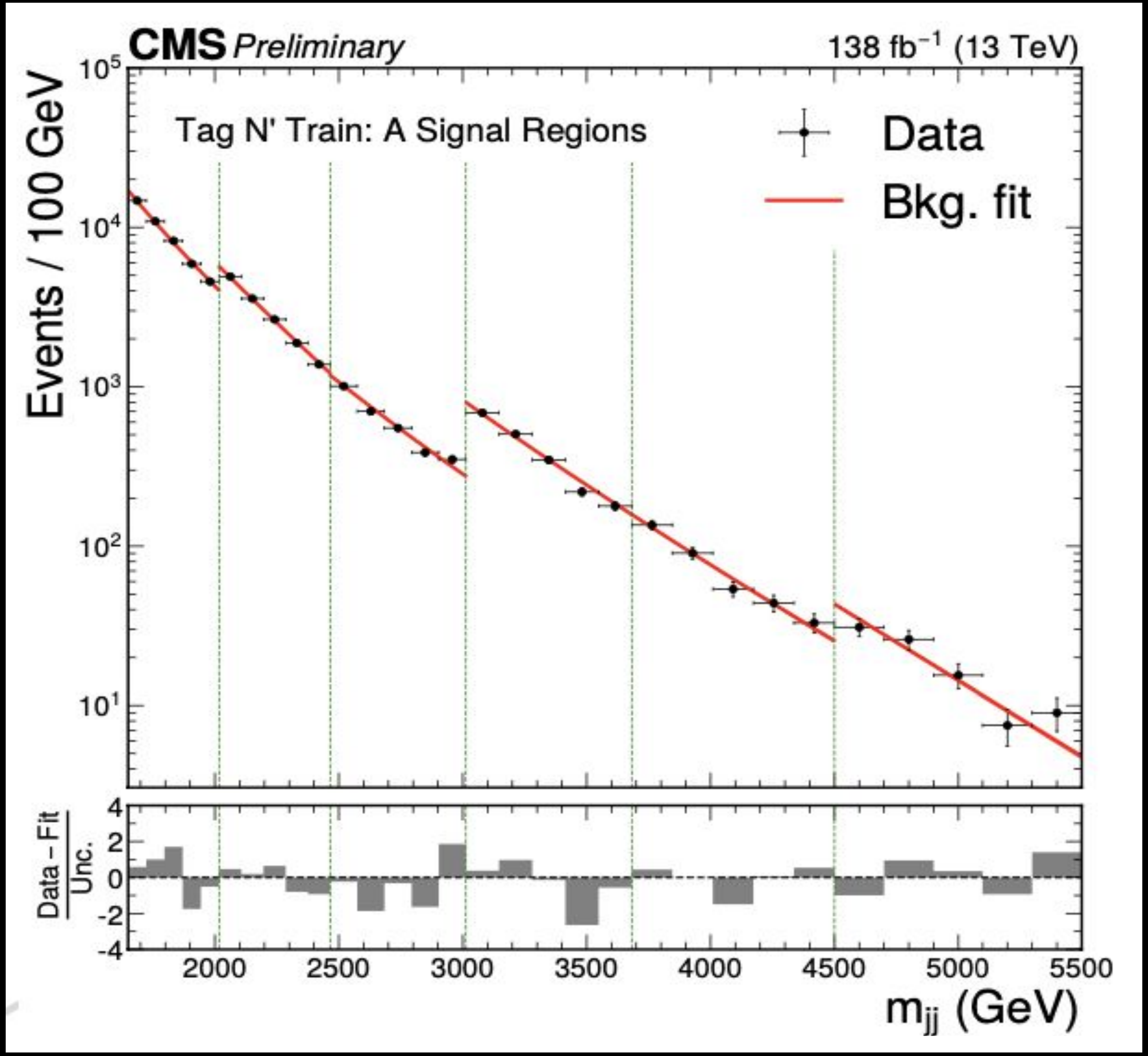
E.g CASE


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}

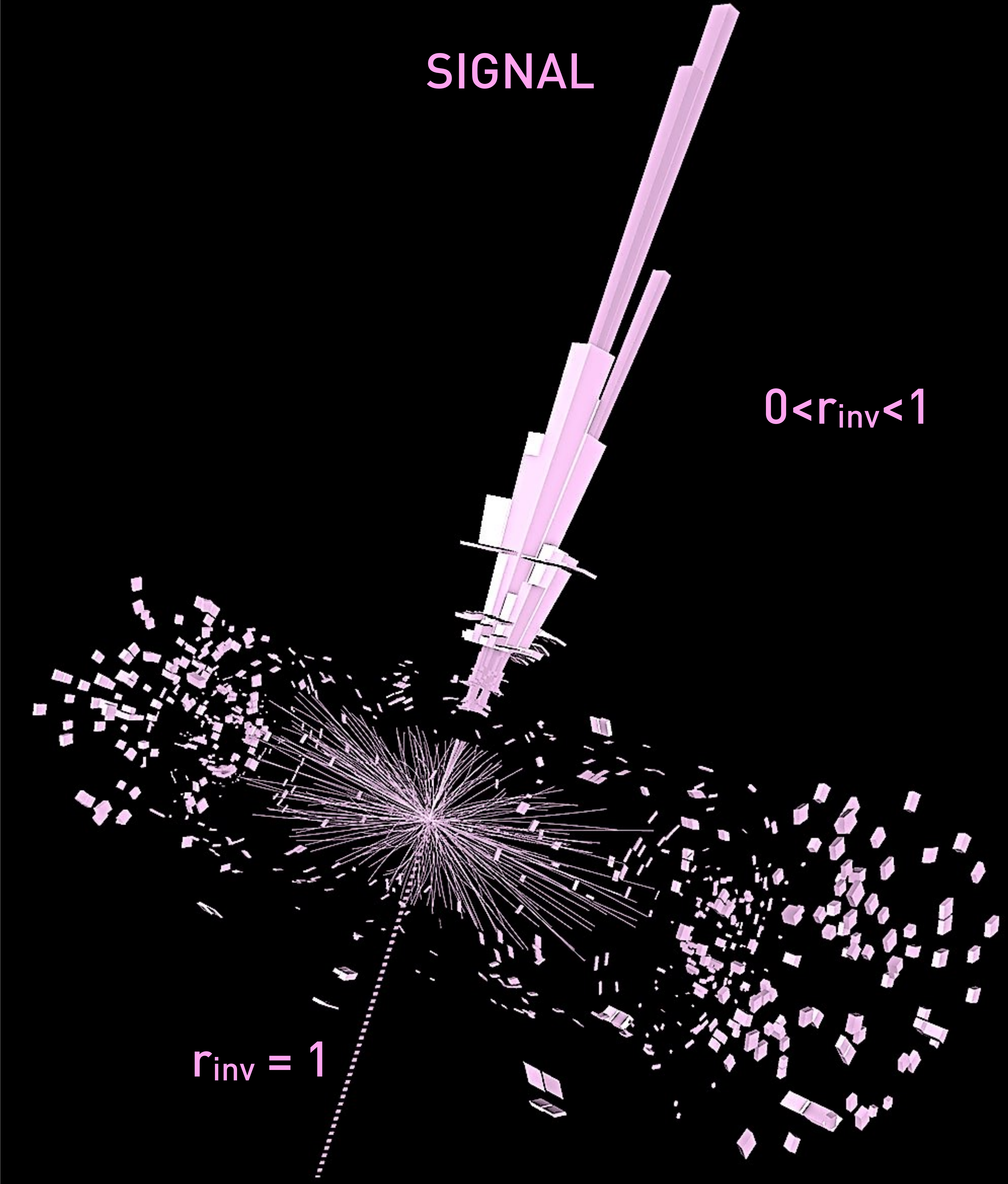
E.g CASE


CMS Simulation Preliminary
(13 TeV)
Before cut on anomaly score
Simulated Pseudodata
Signal + Background Fit
Signal
Background
χ²/ndf = 27.11/31 = 0.87
Prob = 0.667
Events / 100 GeV
Data-Fit / Unc.
Dijet invariant mass (GeV)

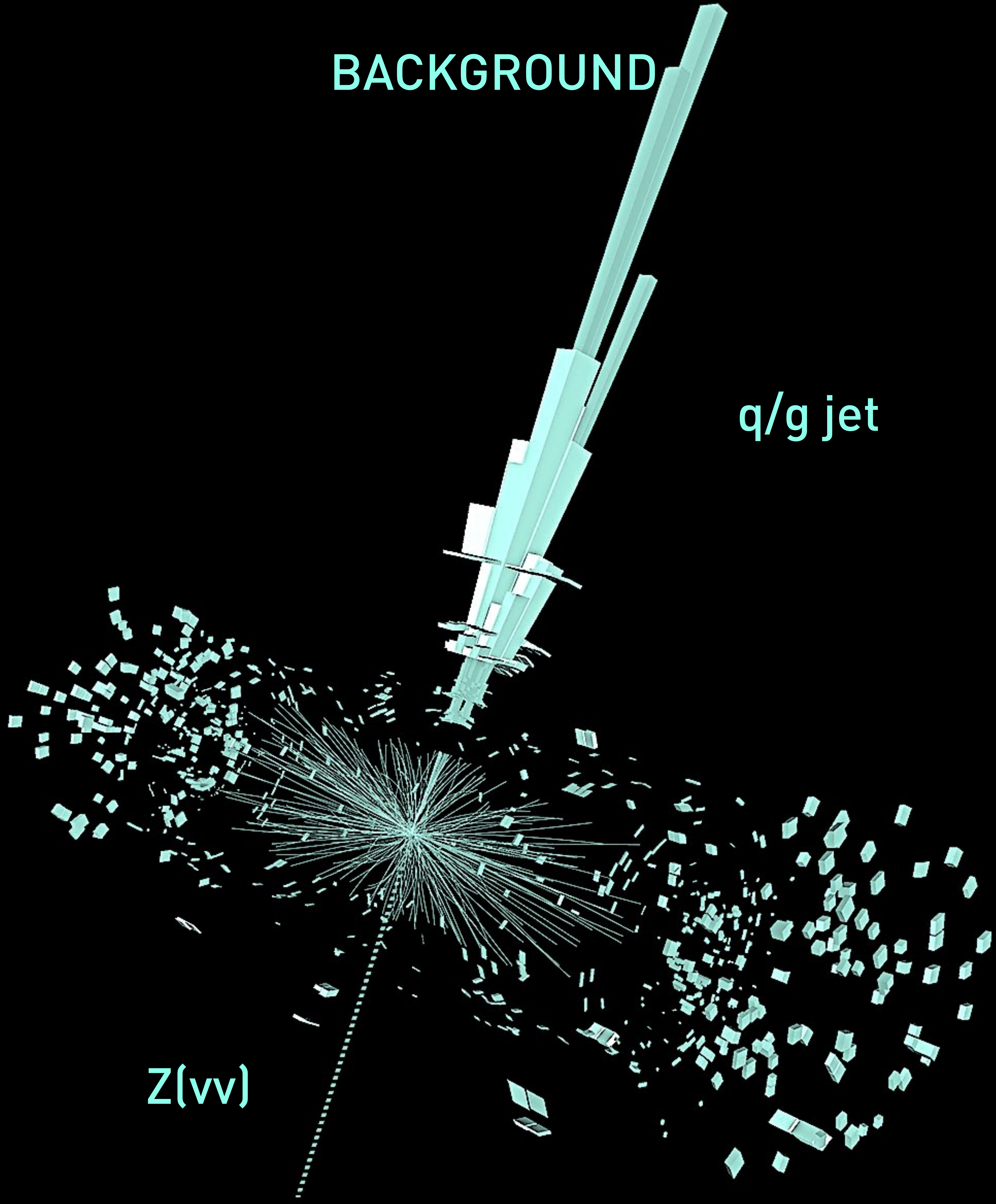

CMS Simulation Preliminary
(13 TeV)
After cut on anomaly score
Simulated Pseudodata
Signal + Background Fit
Signal
Background
χ²/ndf = 16.36/24 = 0.68
Prob = 0.875
Events / 100 GeV
Data-Fit / Unc.
Dijet invariant mass (GeV)


CMS Preliminary
138 fb⁻¹ (13 TeV)
Tag N' Train: A Signal Regions
Data
Bkg. fit
Events / 100 GeV
Data − Fit / Unc.
2000
2500
3000
3500
4000
4500
5000
5500
m_jj (GeV)

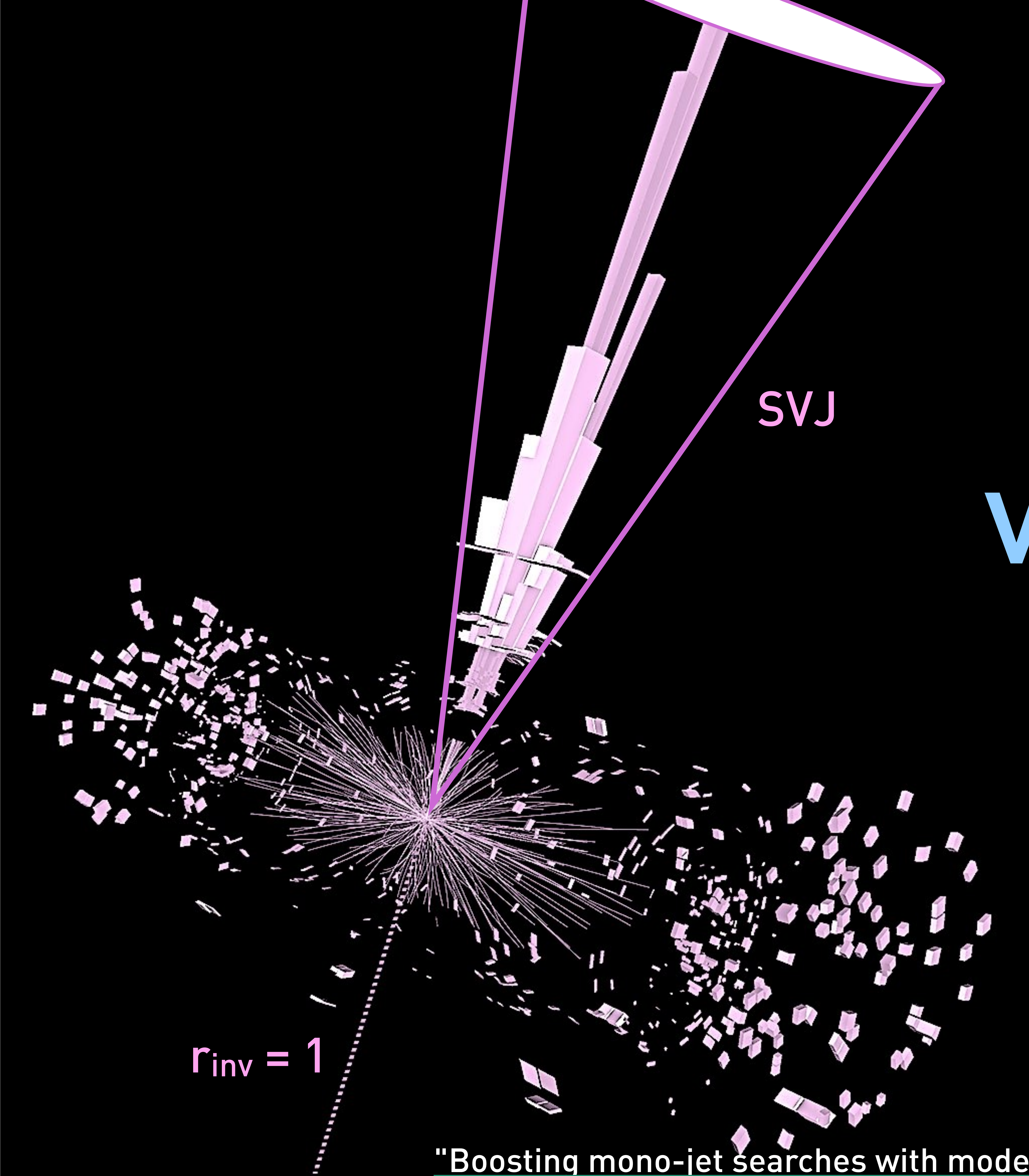
SIGNAL



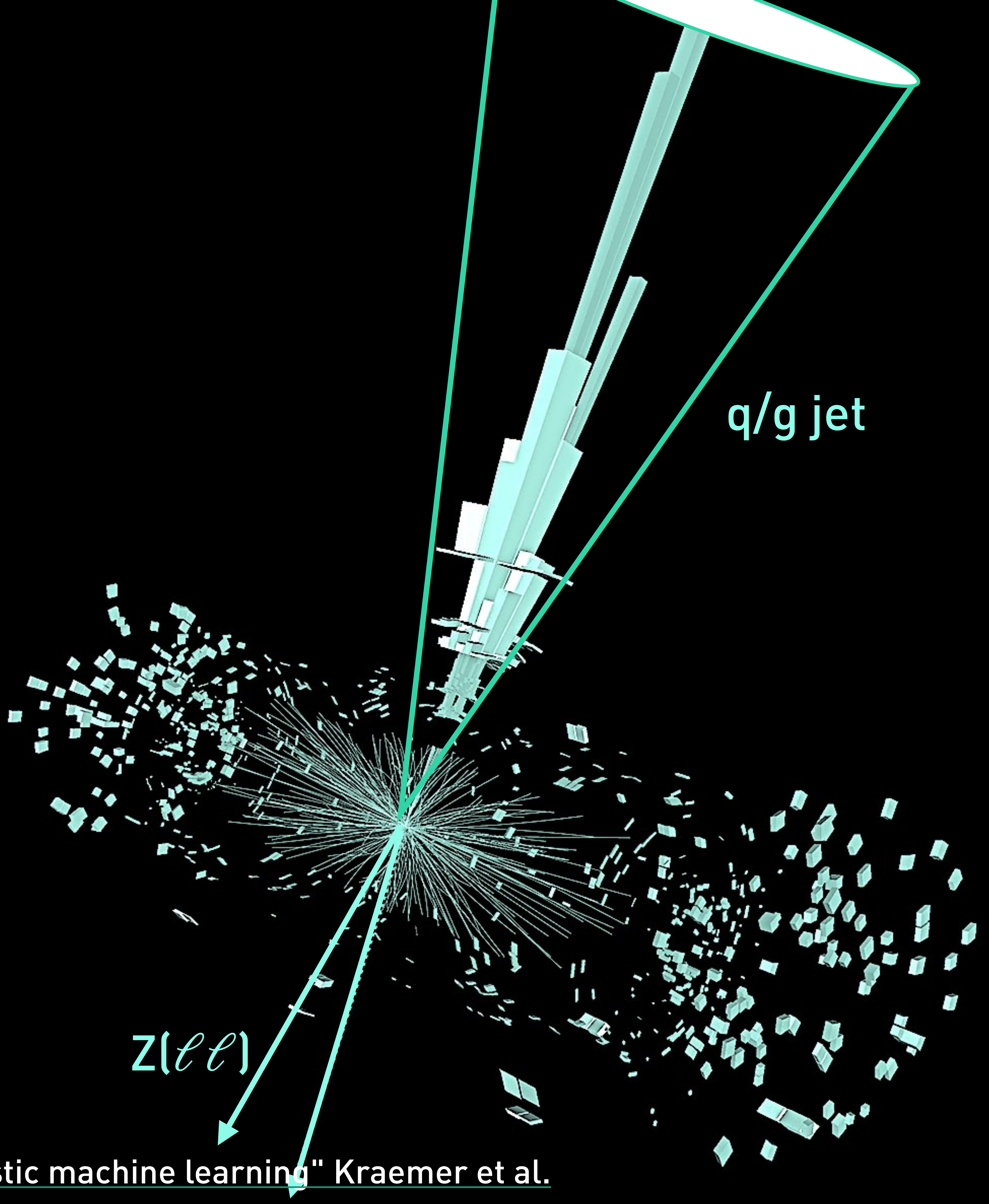
BACKGROUND




$0<r_{inv}<1$
$r_{inv} = 1$
VS.
q/g jet
$Z(\nu\nu)$



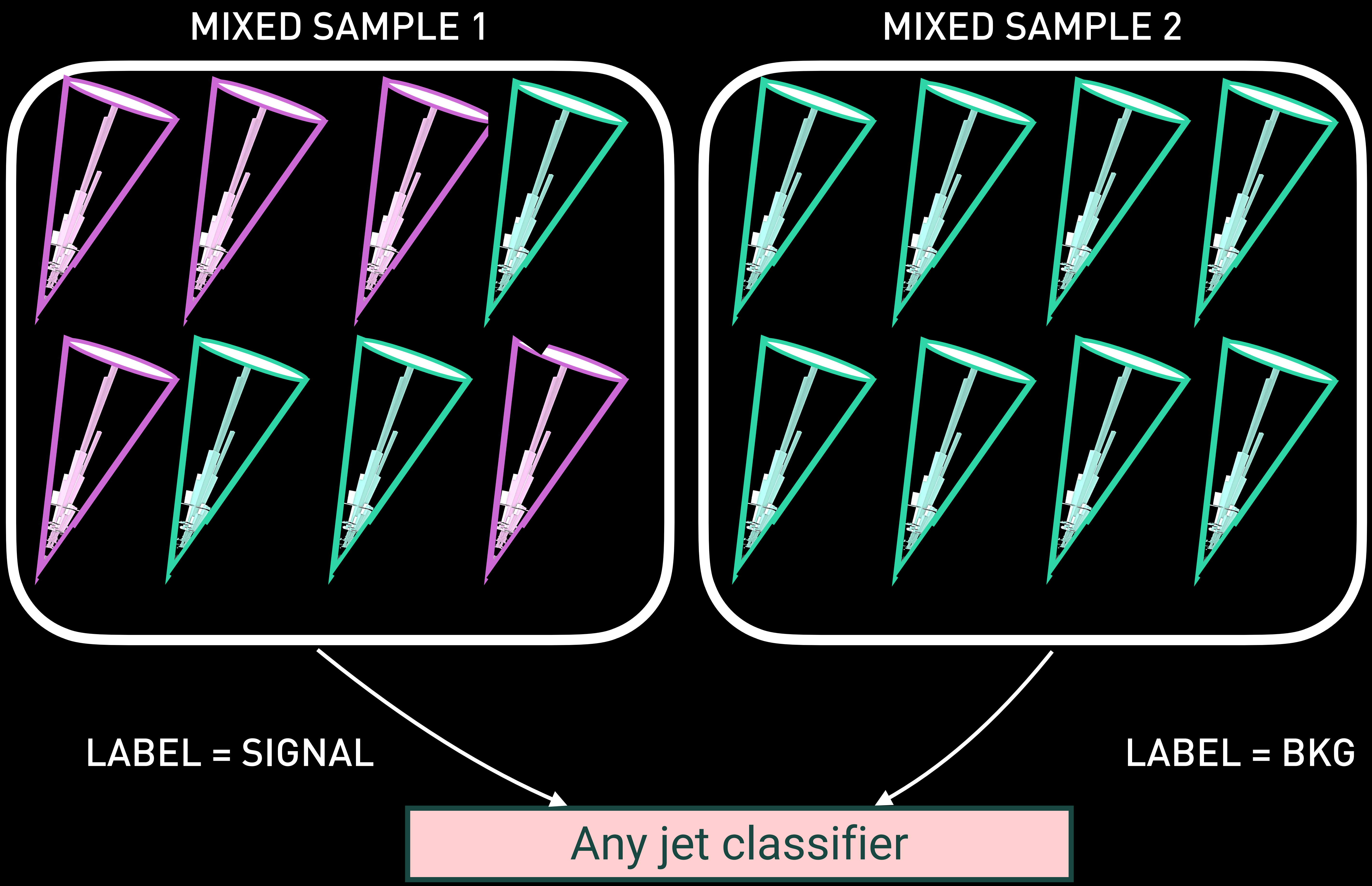
VS.



"Boosting mono-jet searches with model-agnostic machine learning" Kraemer et al.

JETS FROM
MET+JET TOPOLOGY
→ SIGNAL REGION

JETS FROM
$\ell\ell$+JET TOPOLOGY
→ SIGNAL NOT EXPECTED HERE



MIXED SAMPLE 1

MIXED SAMPLE 2

LABEL = SIGNAL

LABEL = BKG

Any jet classifier

Density estimation

Various methods

ML-based interpolation from sidebands to signal region:

ANODE: interpolates densities from sidebands to the signal-region & constructs likelihood ratio

CATHODE: samples from the background model in signal region after interpolating and estimates likelihood ratio with classifier

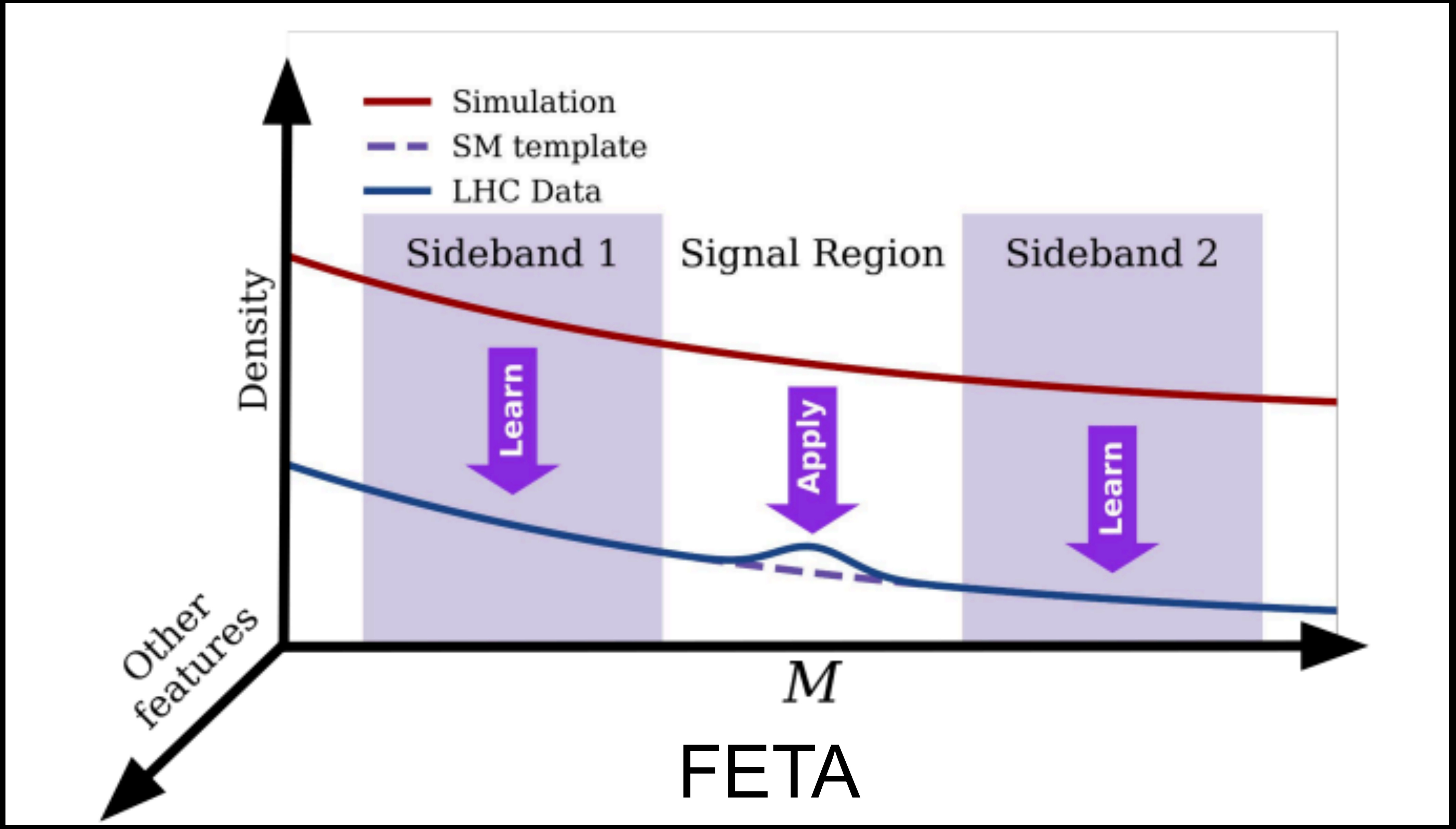
LaCATHODE: Use a in flow to perform CATHODE in latent space

CURTAINS: Train invertible NN conditioned on mass to map between sidebands

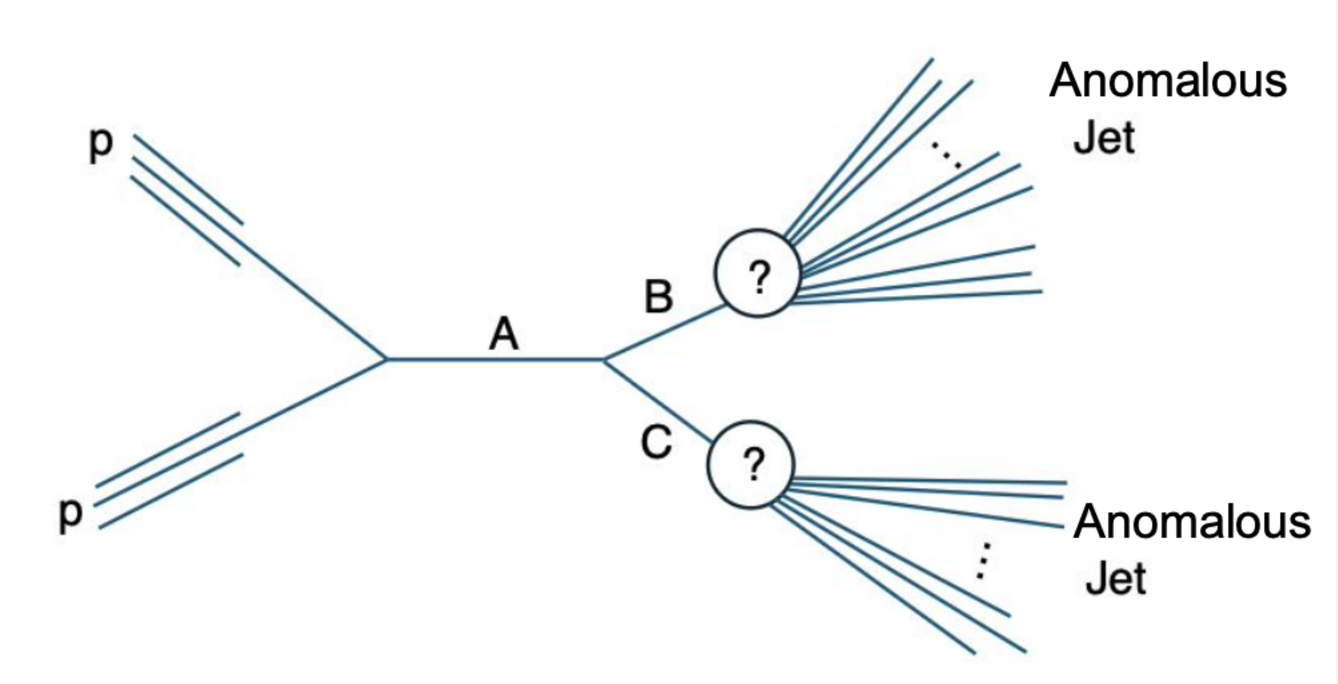
ML-based MC reweighting:

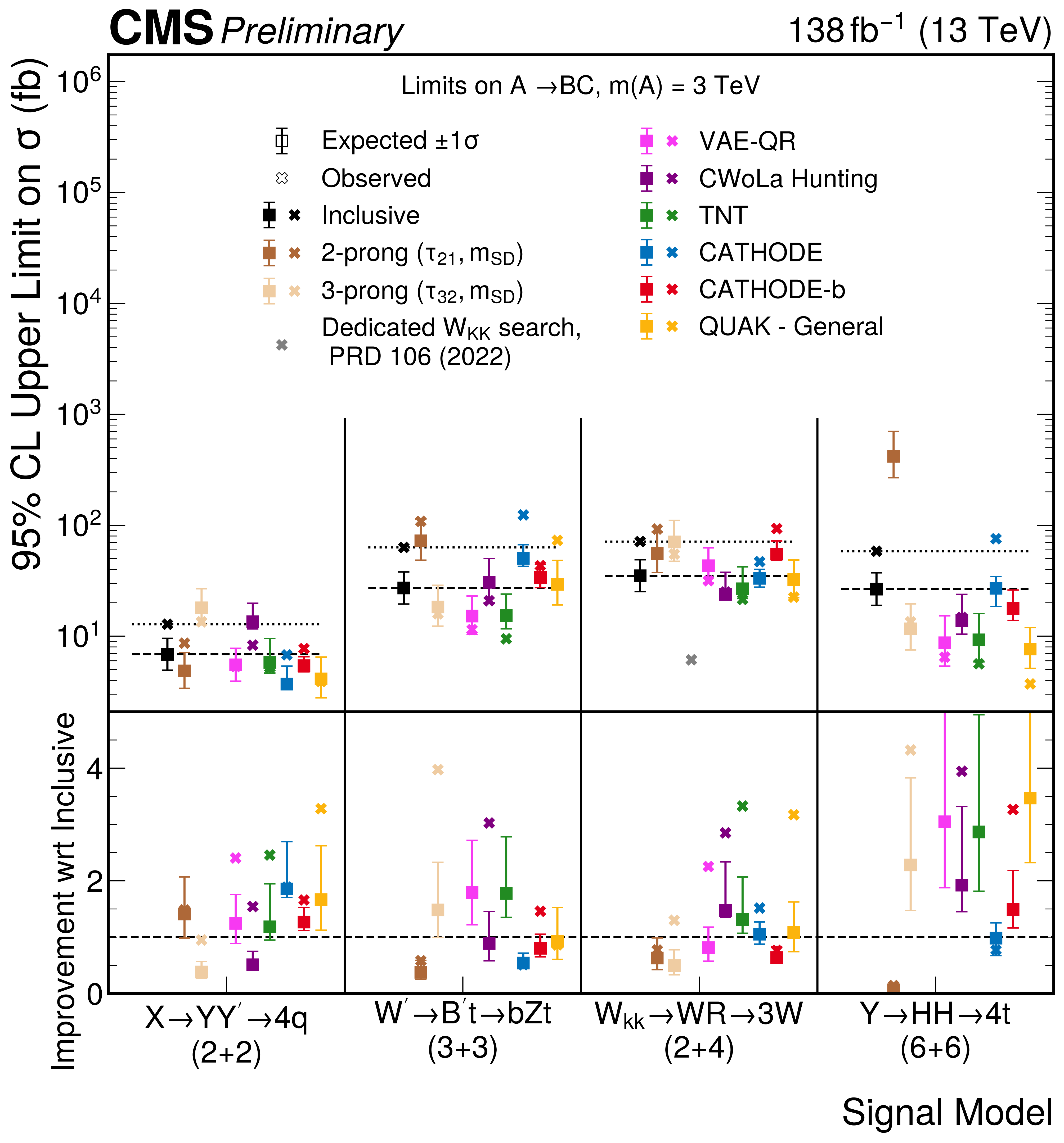
SALAD: Reweight simulation to match sideband, interpolate into the signal region and use a second classifier to get the likelihood

FETA: Map simulation to data in sidebands, then compare to SR data



Why these methods are good for DM searches



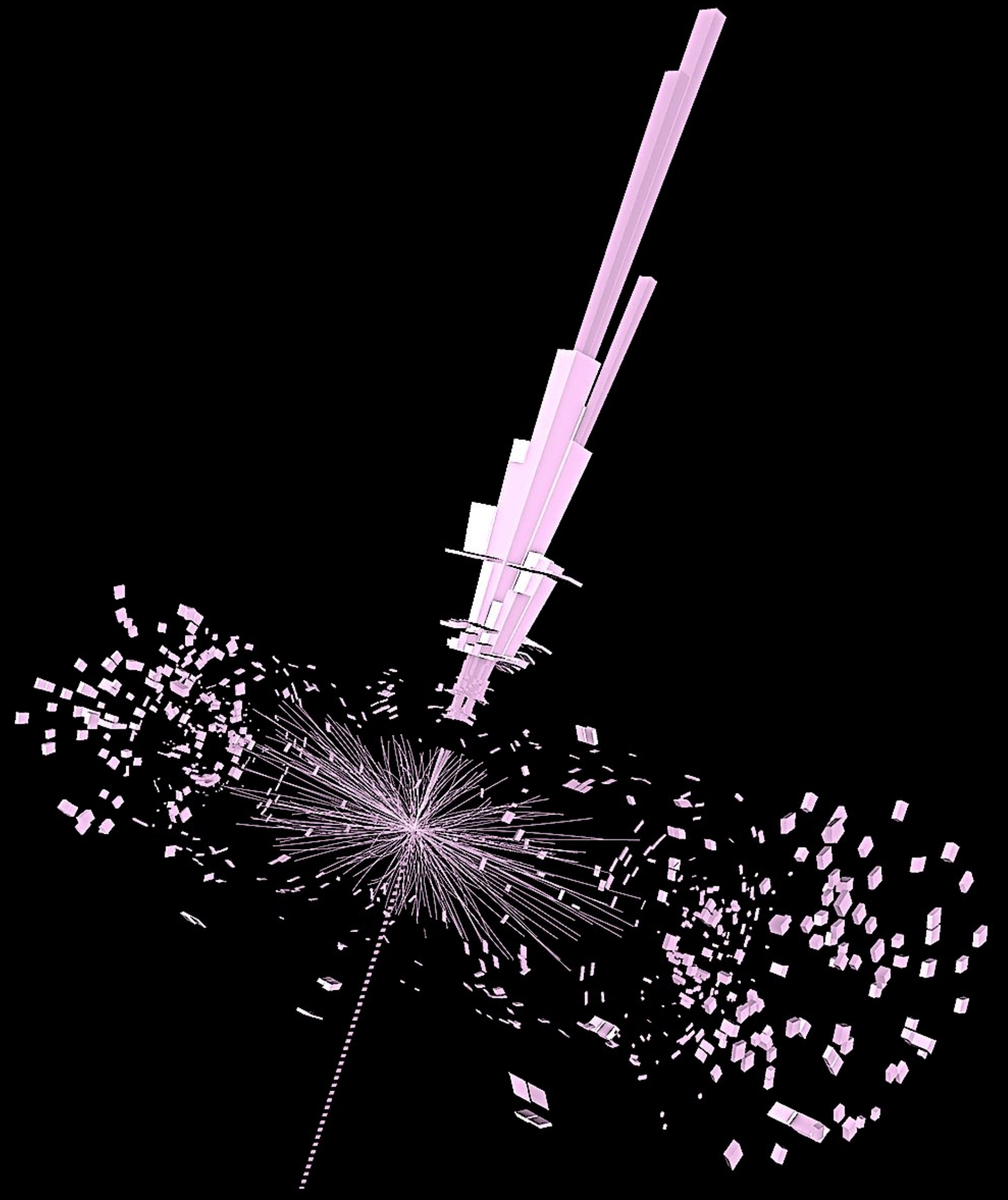


We could cast a huge net to catch a broad range of signals in a single search!

CASE

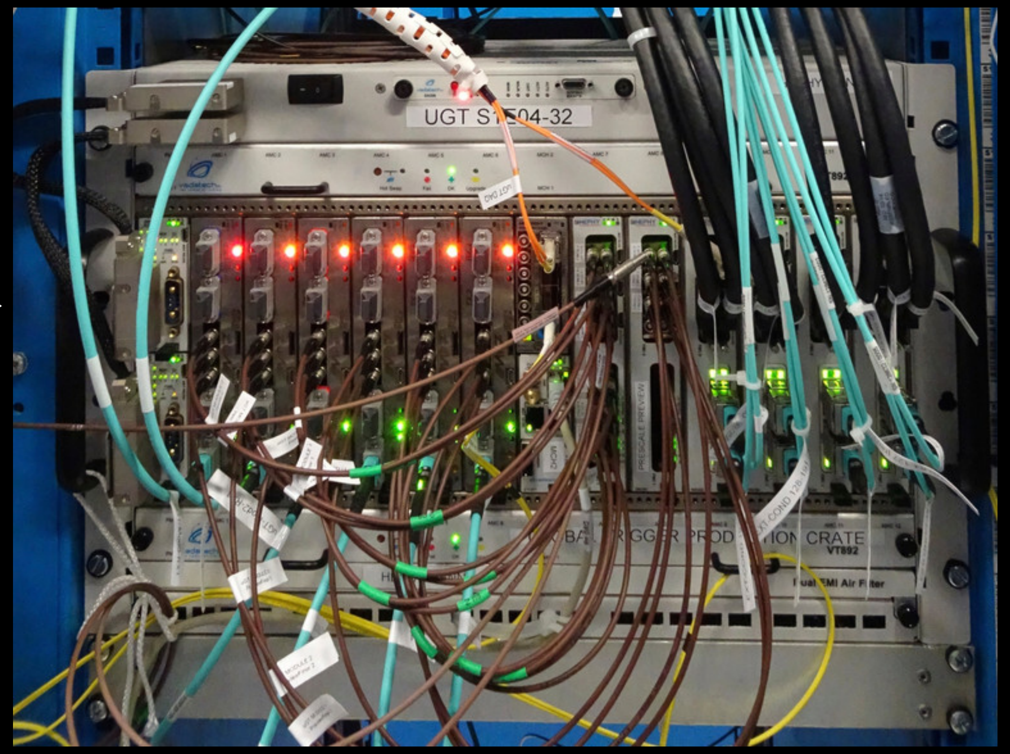
Do physics with 0.0025% of collision events, the rest is discarded!

Detector
- 100% of events left



40 MHz

Level-1 hardware trigger
- 0.3% of events left

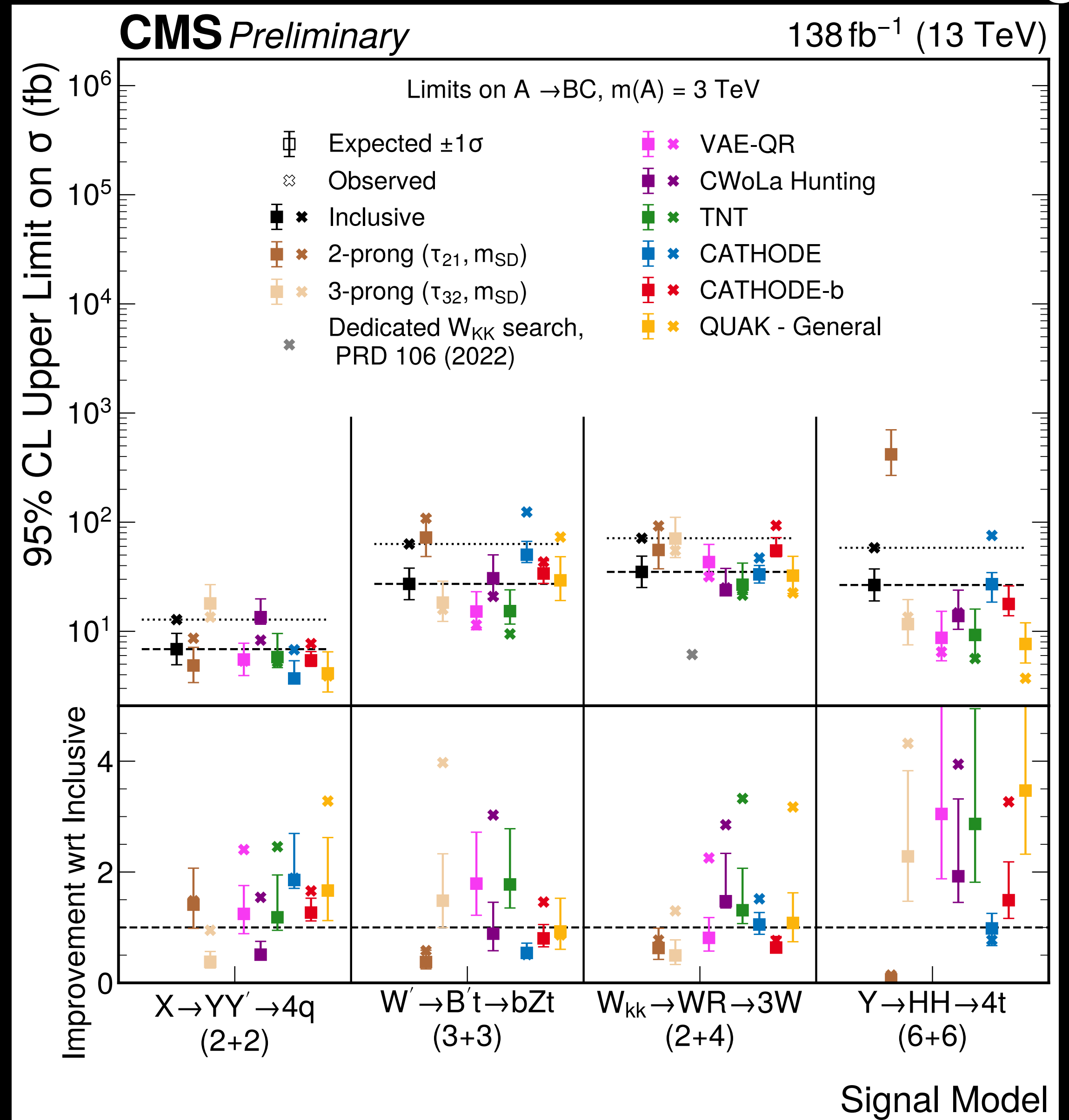


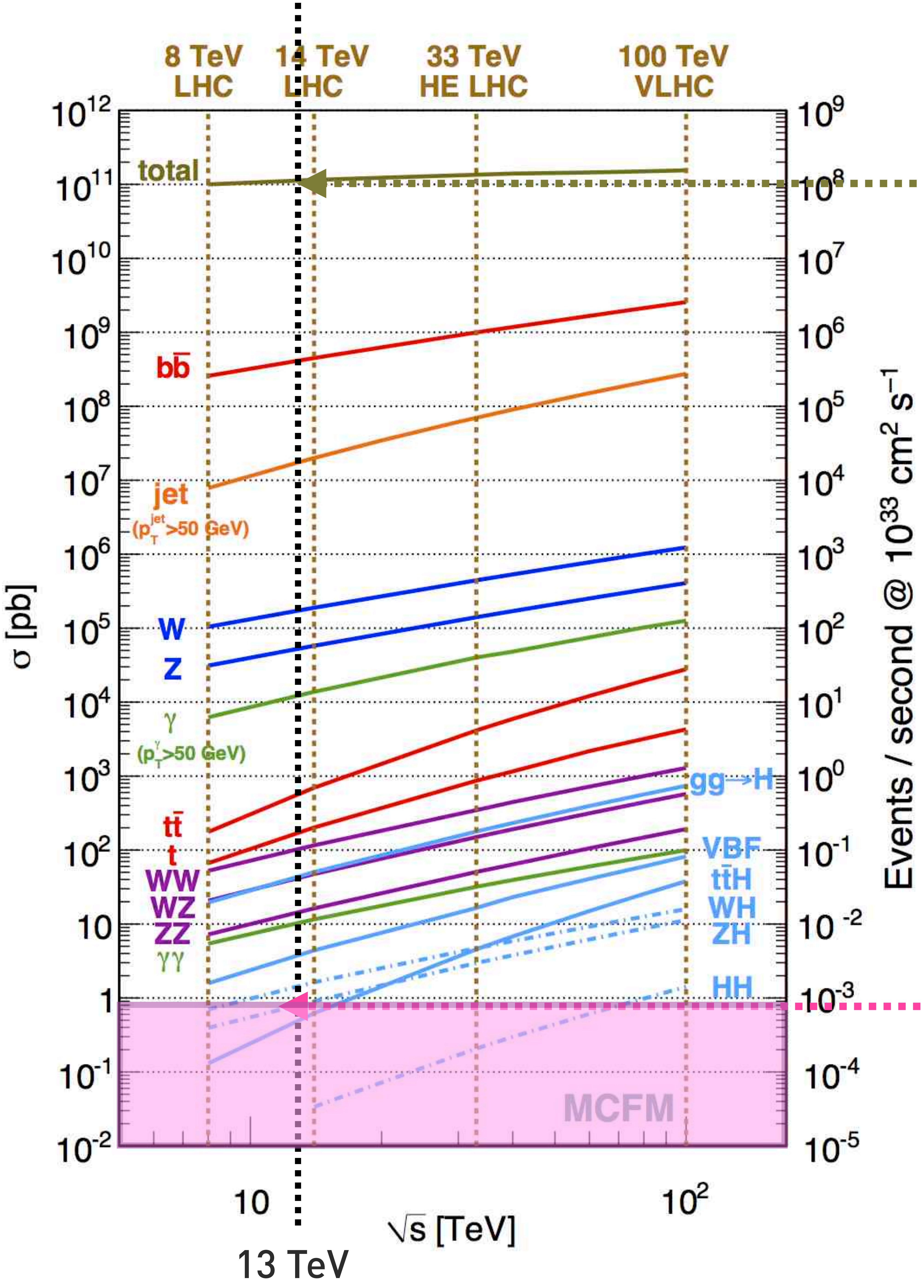
110 kHz

High Level Trigger CPU farm
- 0.0025% of events left

1 kHz

Offline reconstruction and storage





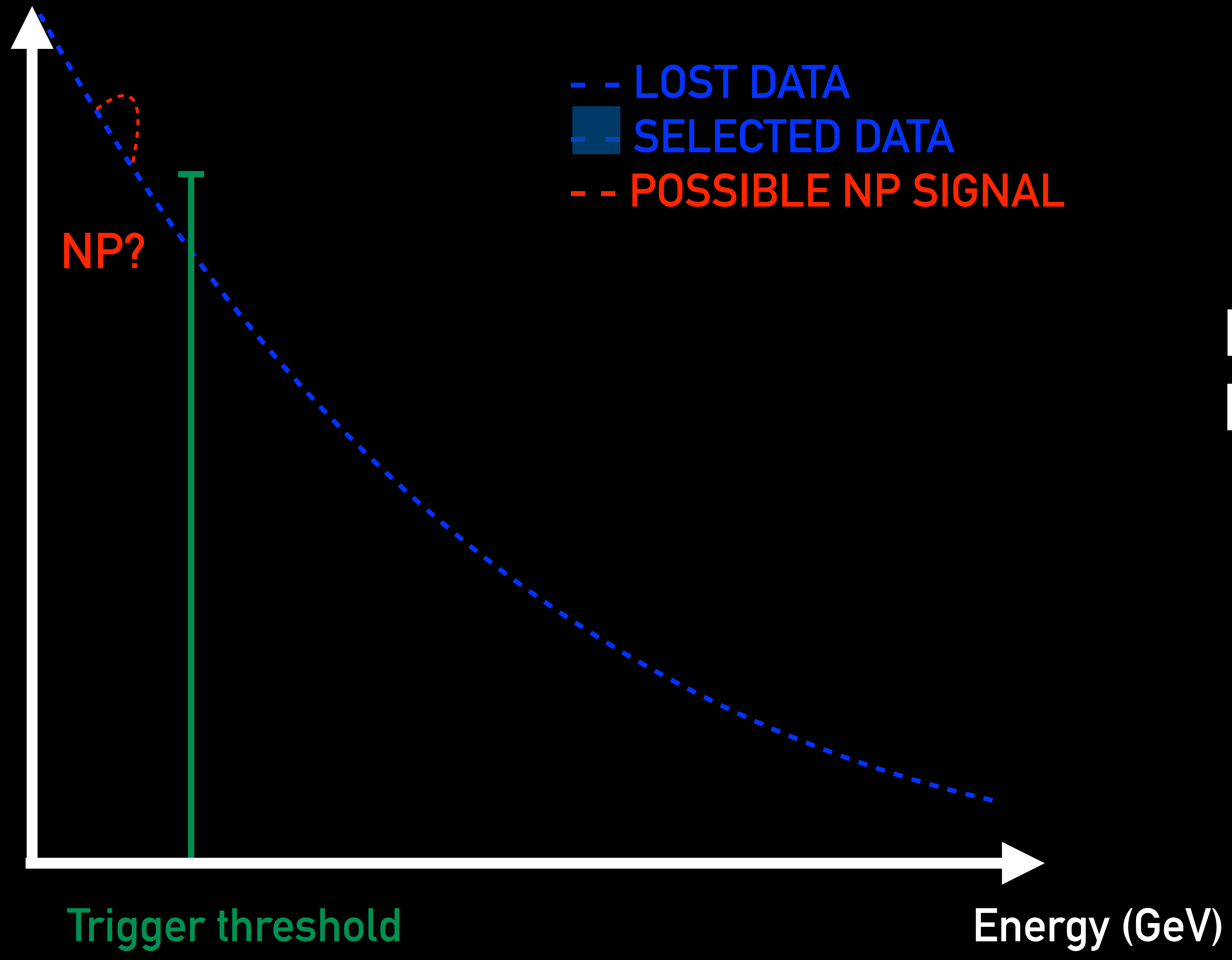
"Probability" of producing "anything"

Mono-jet search limits 95% CL
0.3 - 736 fb

Probing smaller and smaller couplings, lower and lower masses

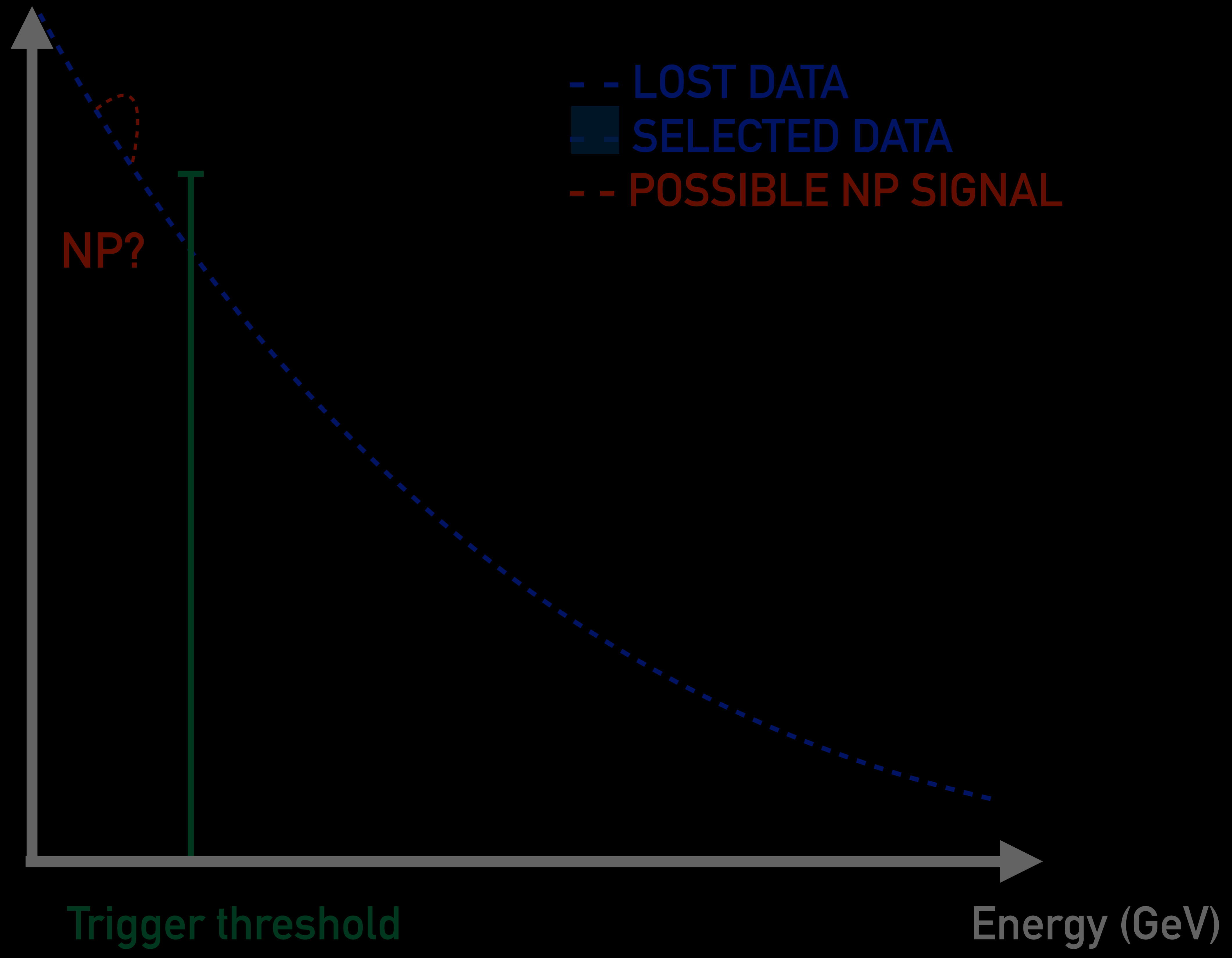
Need <u>more</u> statistics!

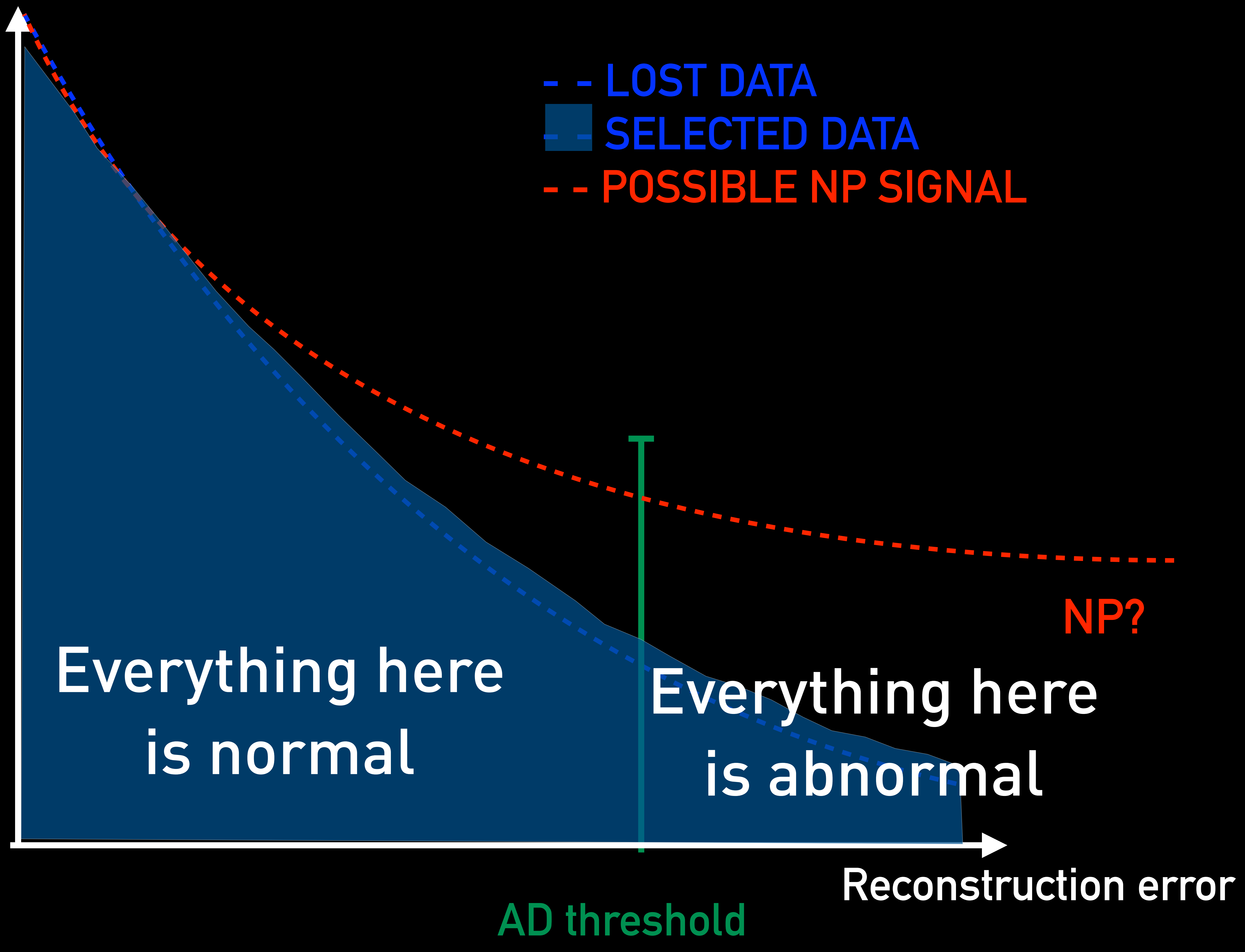
Anomaly Detection triggers



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

Anomaly Detection triggers

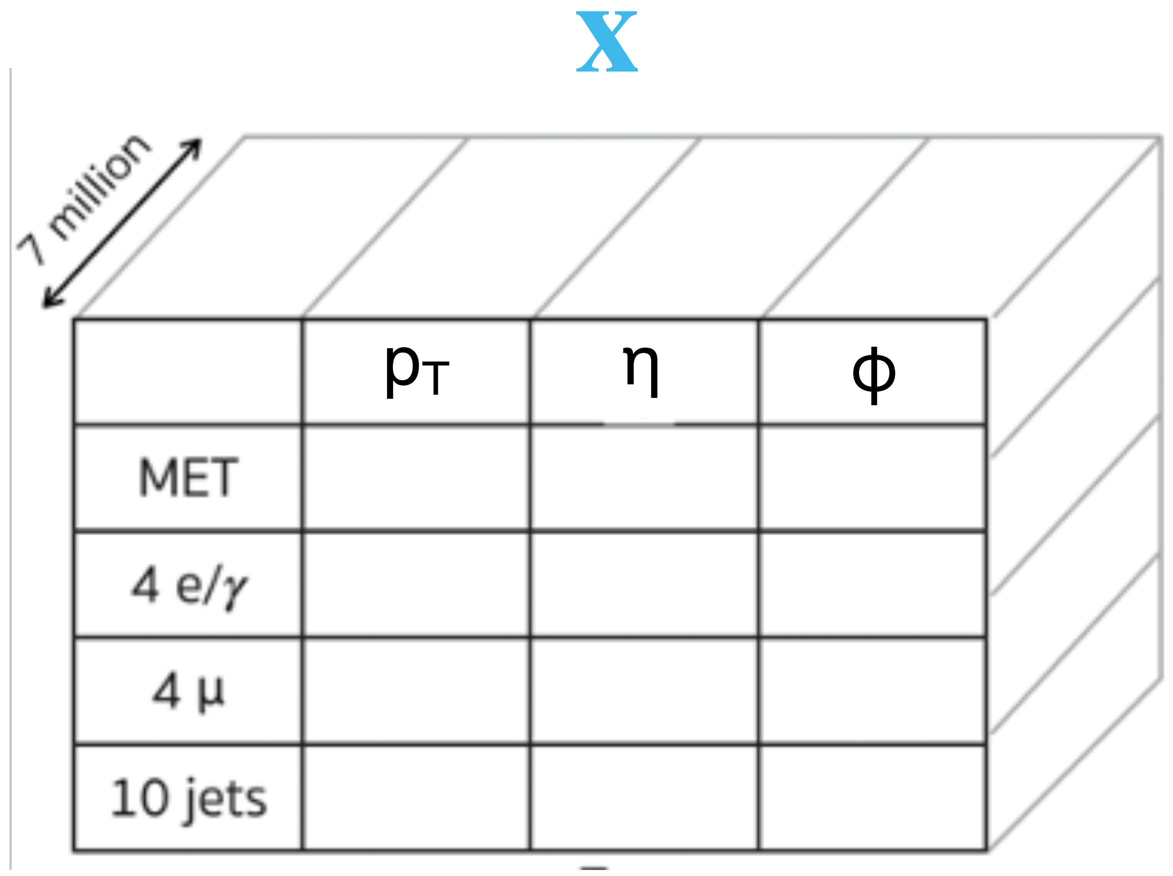




Anomaly Detection in the CMS Level 1 μGT for Run3!

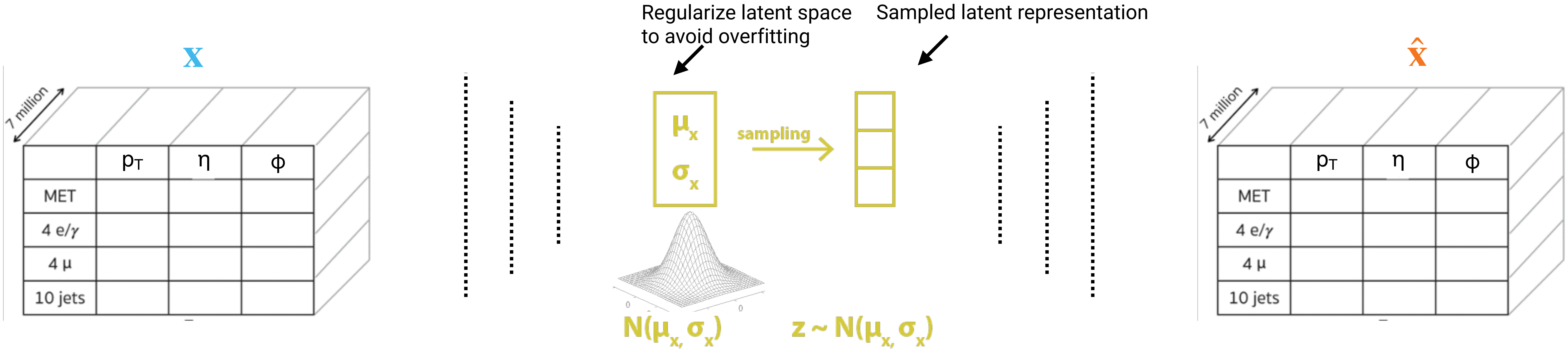
Input from Run 3 μGT quantities:

- (p_T, η, ϕ) hardware integer inputs from: 1 MET, 4 e/γ, 4 μ, and 10 jet objects



	p_T	η	ϕ
MET			
4 e/γ			
4 μ			
10 jets			

AXOL1TL



Regularize latent space to avoid overfitting

Sampled latent representation

x

7 million

	p_T	η	ϕ
MET			
4 e/γ			
4 μ			
10 jets			

$\hat{x}$

7 million

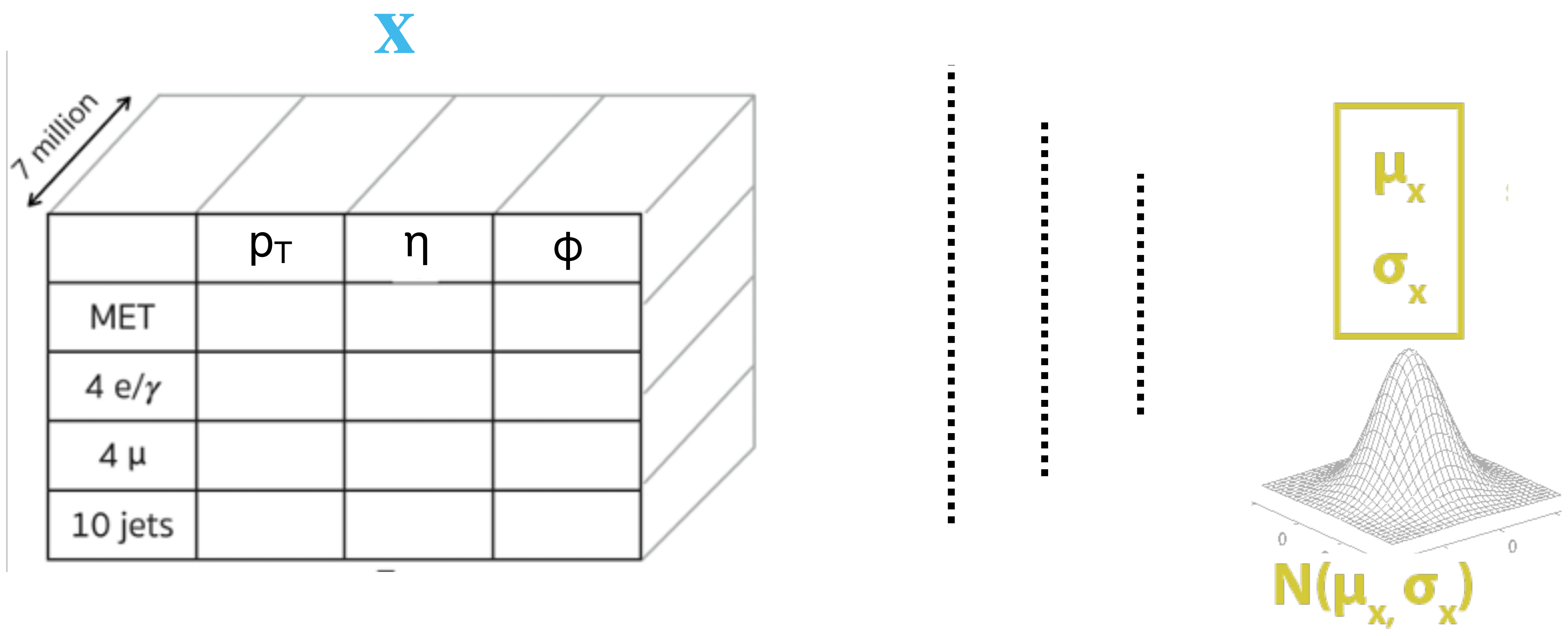
	p_T	η	ϕ
MET			
4 e/γ			
4 μ			
10 jets			

$$\text{loss} = ||x - \hat{x}||^2 + \text{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,]$$

AXOL1TL

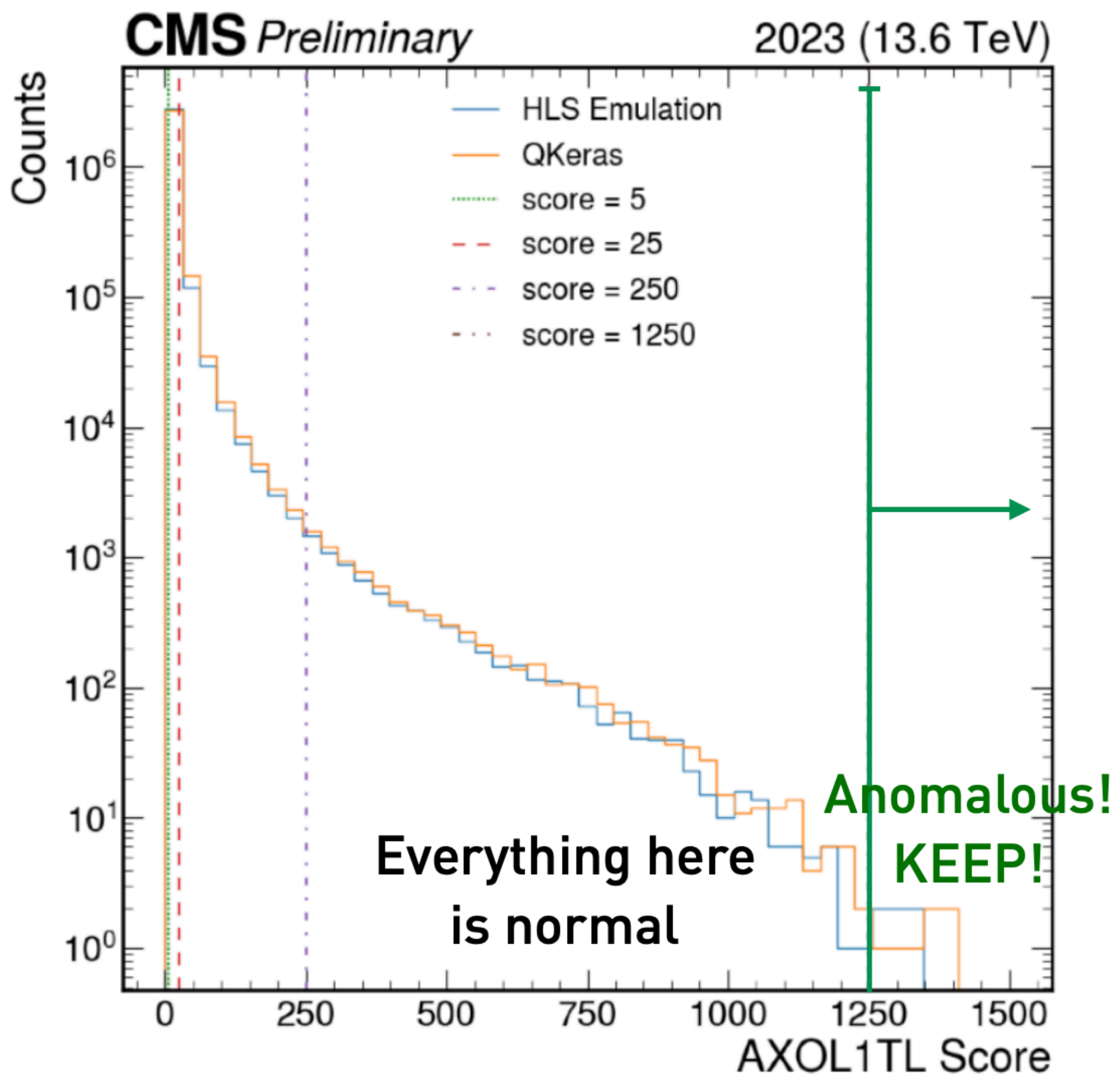
Only deploy encoder, compute degree of abnormality from patent space only

- Do not need to keep input around for MSE
- Half network size and latency!



	p_T	η	ϕ
MET			
4 e/γ			
4 μ			
10 jets			

$$\text{loss} = \|x - \hat{x}\|^2 + \boxed{\mathrm{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,]}$$



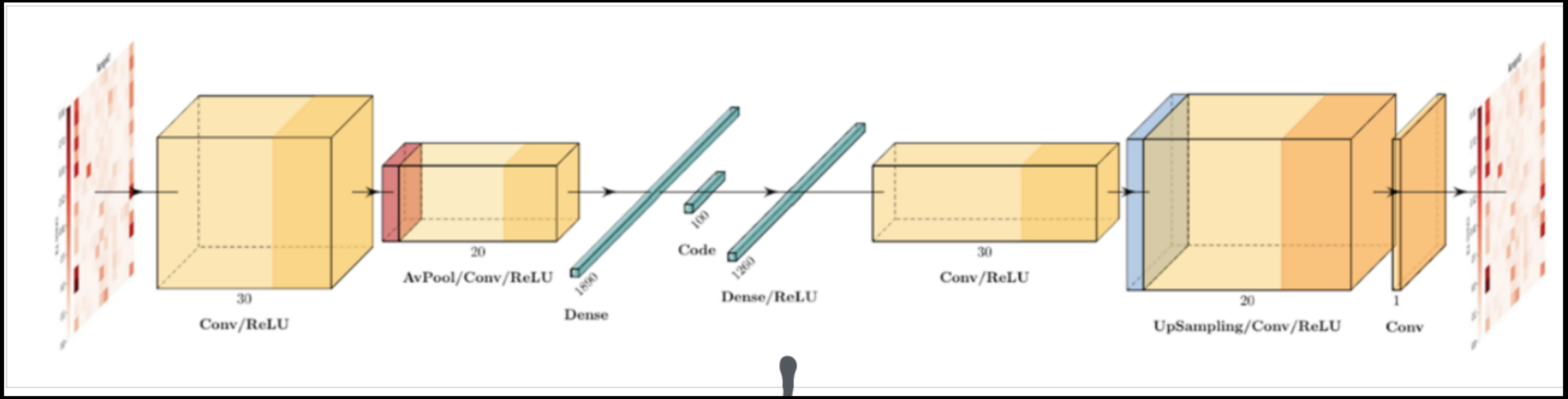
Everything here is normal

Anomalous! KEEP!

CICADA

CNN in Level-1 Calorimeter Trigger!

Represent calorimeter tower as image and use CNN auto encoder



AXOL1TL

E.g Higgs → A(15 GeV) A(15 GeV) → 4b

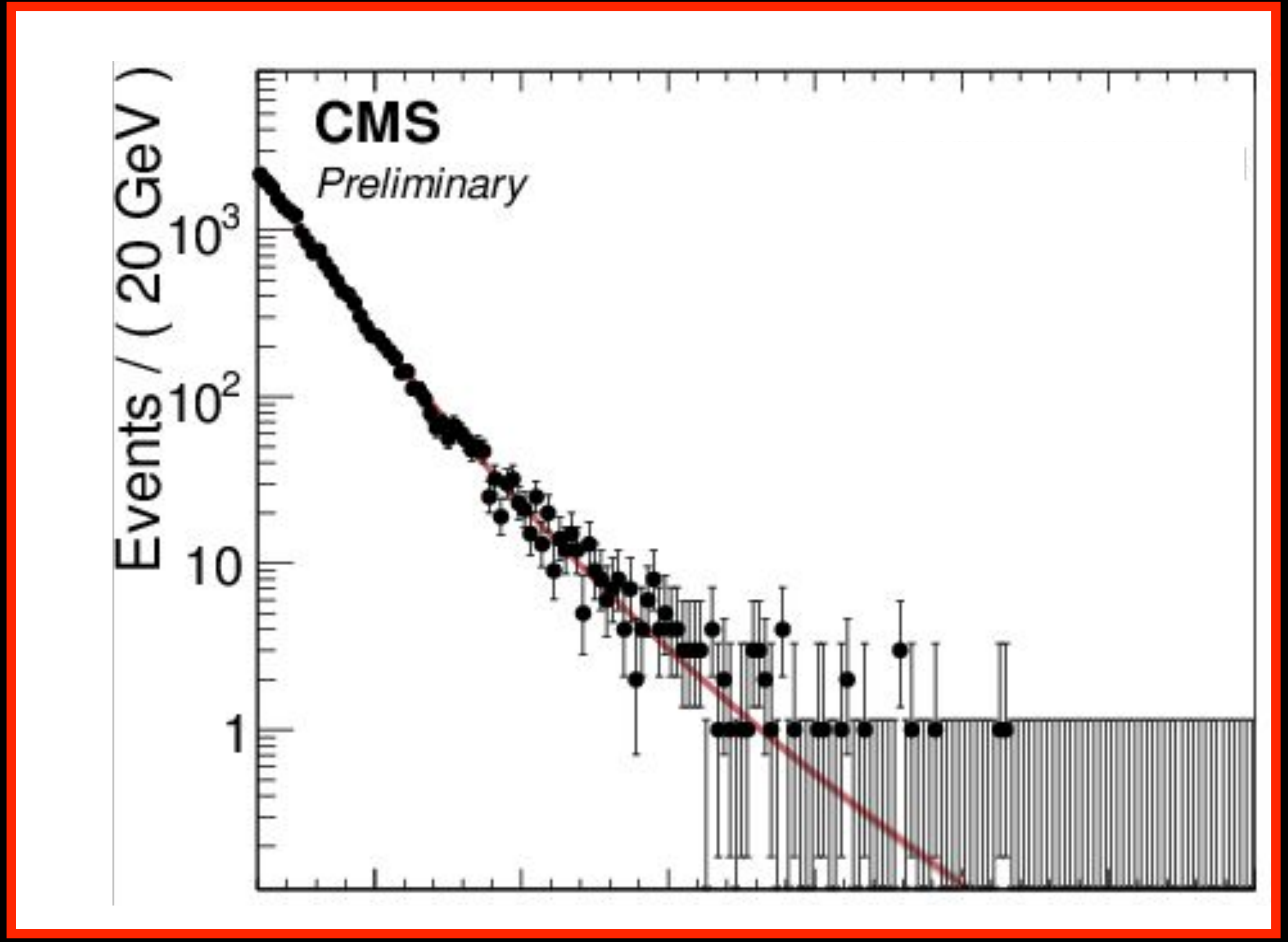
AXOL1TL Rate	1 kHz	5 kHz	10 kHz
Signal Efficiency Gain	46%	100%	133%

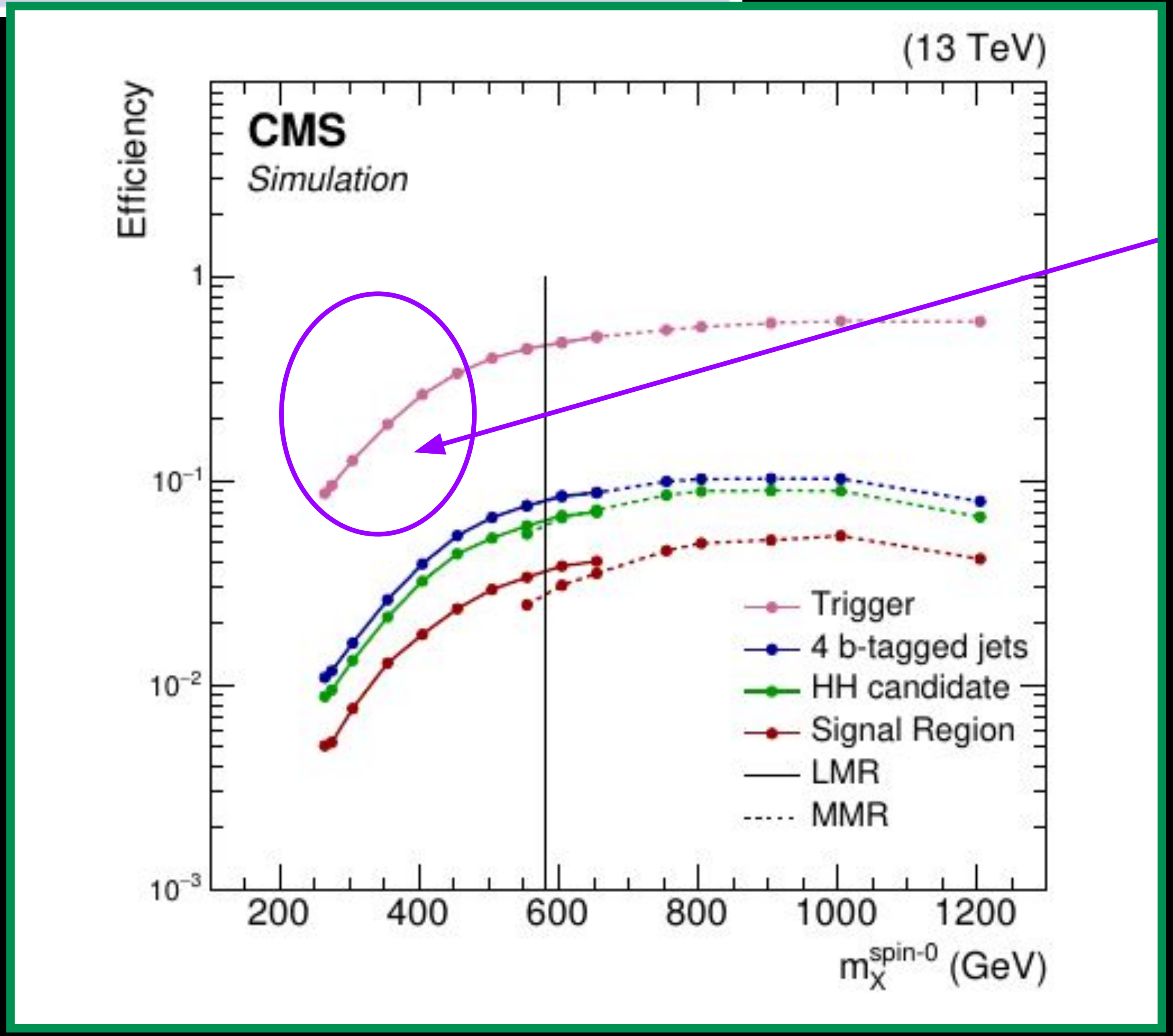
AXOL1TL

E.g Higgs → A(15 GeV) A(15 GeV) → 4b

<u>We can do both of these efficiently, model-agnostic and datadriven!</u>

Signal Efficiency Gain	46%	100%	133%





End-to-end-approach: NPLM

Alternative approach: End-to-end DNN search

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

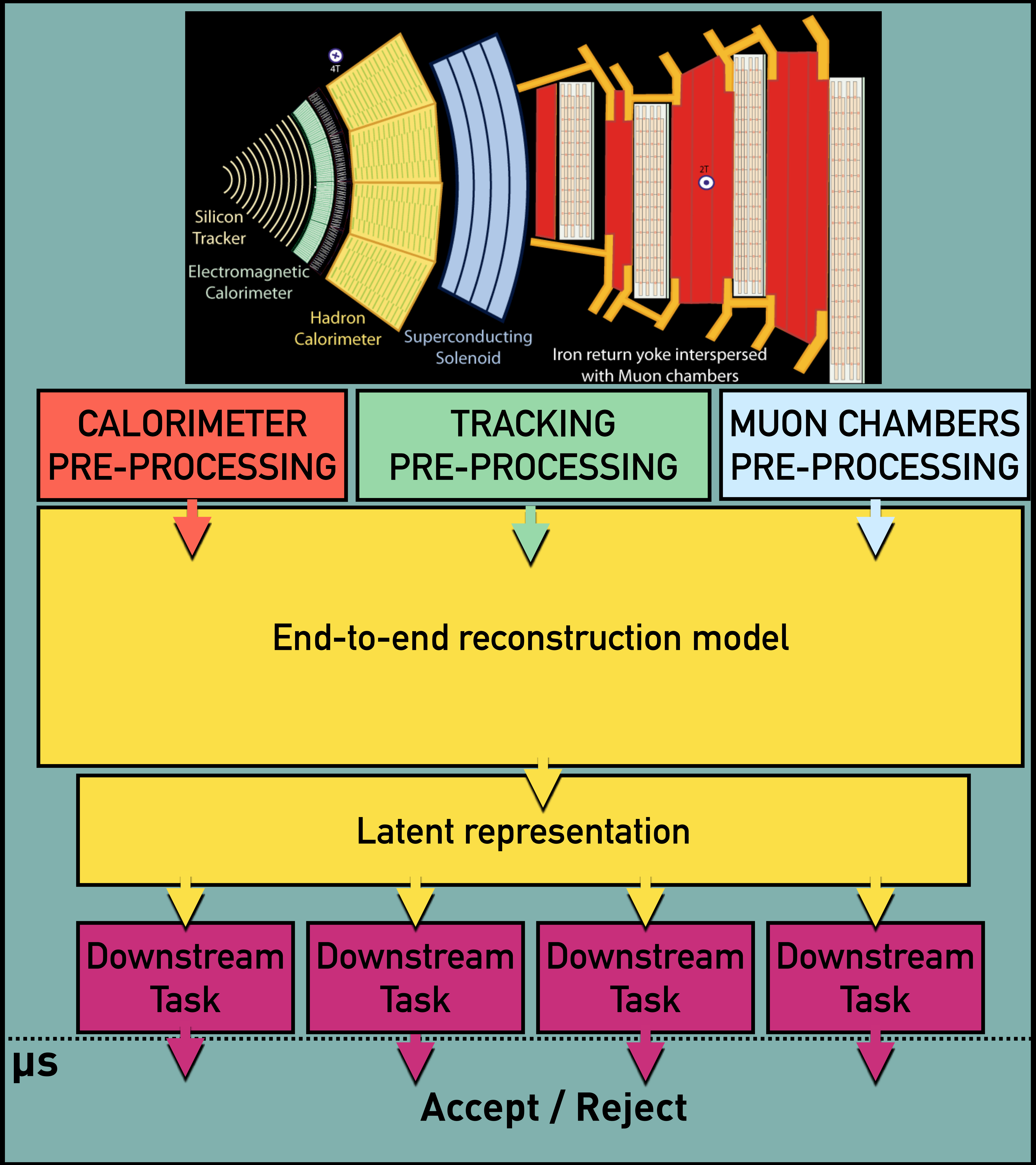
Idea: Assume alternate model n(x|w) can be parametrised in terms of reference model n(x|R)

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

- Let DNN parametrise alternative model

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

One model, learn neural embedding?



CALORIMETER PRE-PROCESSING

TRACKING PRE-PROCESSING

MUON CHAMBERS PRE-PROCESSING

End-to-end reconstruction model

Latent representation

Downstream Task

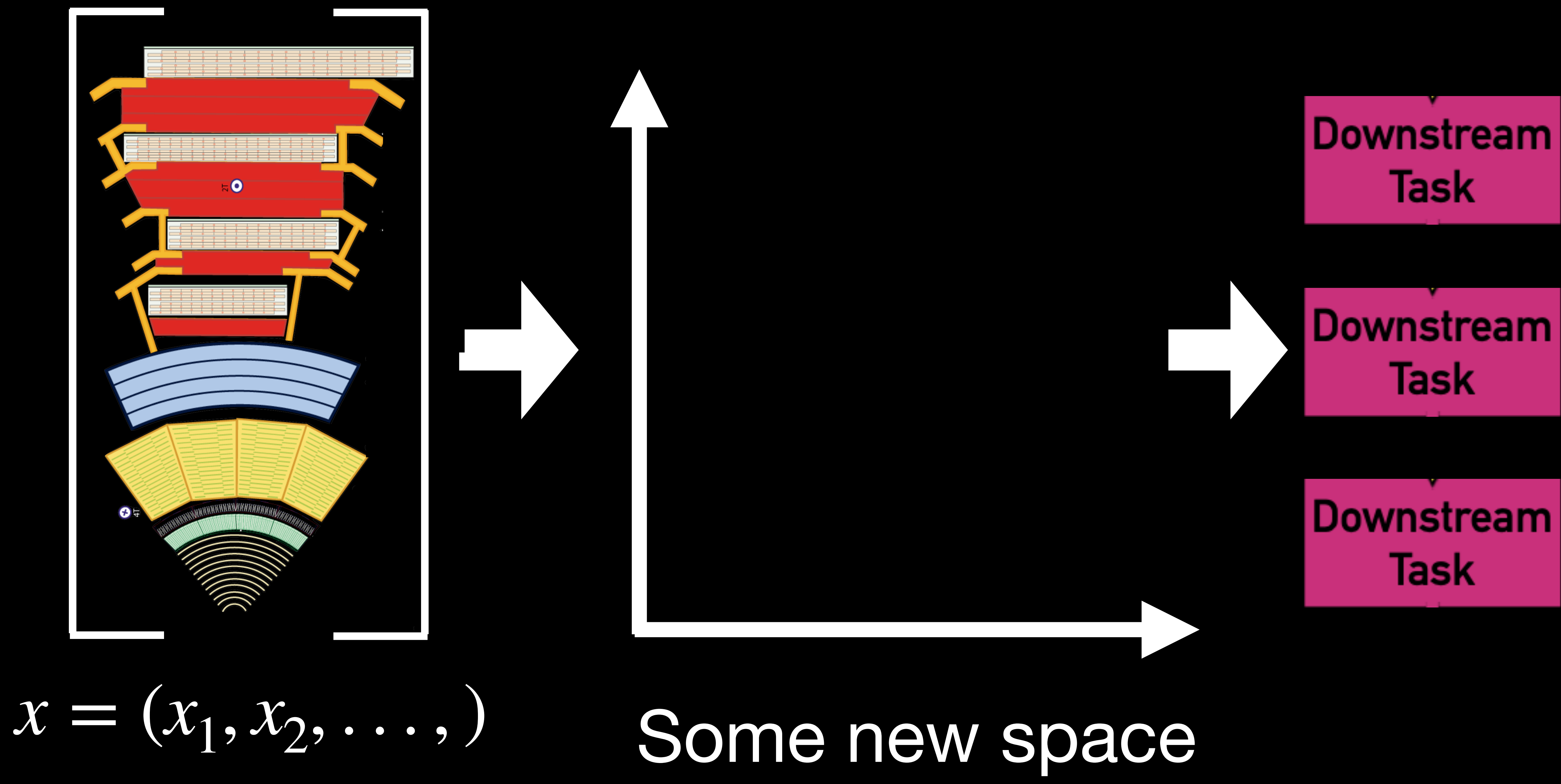
Downstream Task

Downstream Task

Downstream Task

μs

Accept / Reject



$x = (x_1, x_2, \ldots,)$

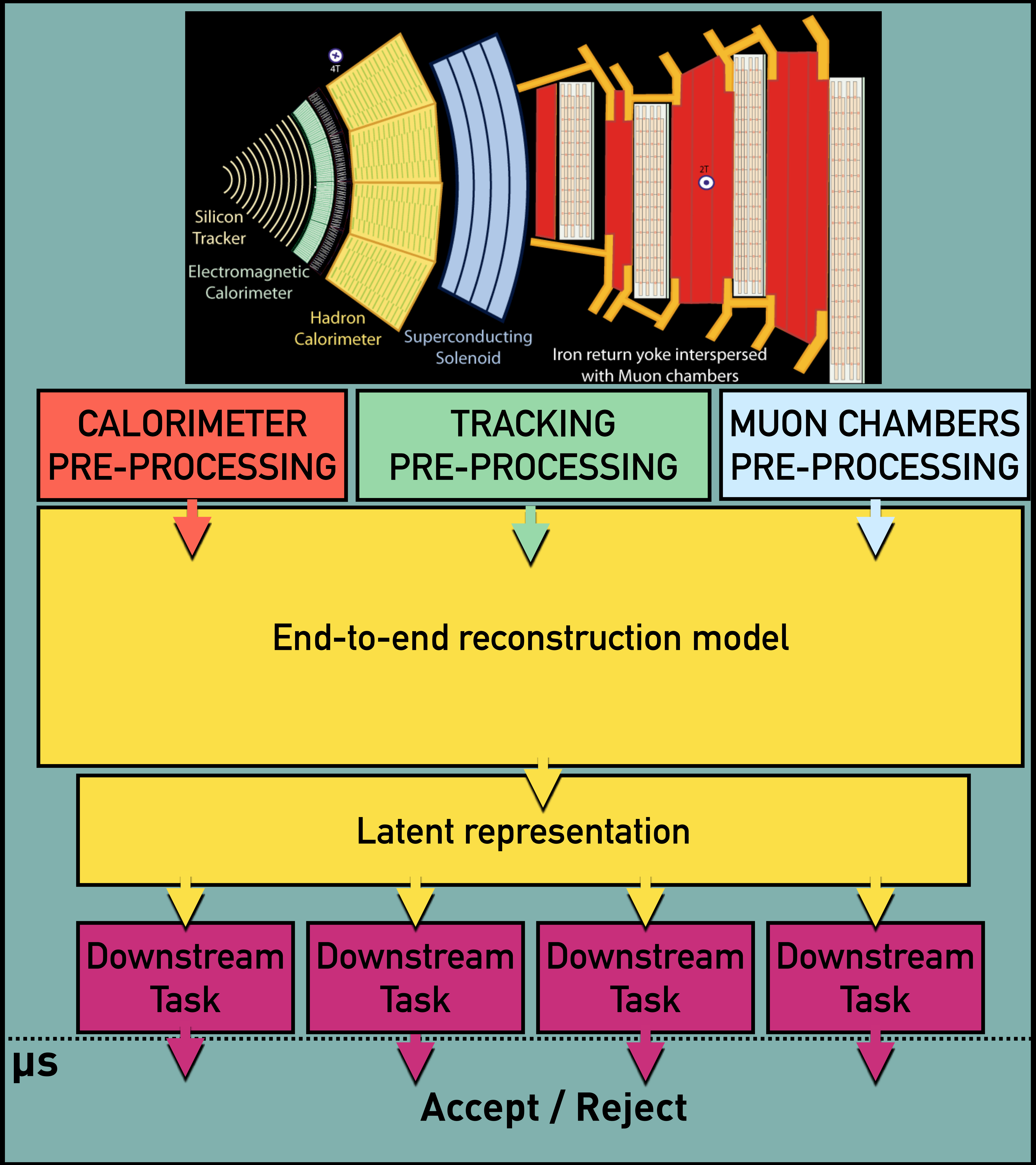
Some new space

Downstream Task

Downstream Task

Downstream Task

One model, learn neural embedding?



CALORIMETER PRE-PROCESSING

TRACKING PRE-PROCESSING

MUON CHAMBERS PRE-PROCESSING

End-to-end reconstruction model

Latent representation

Downstream Task

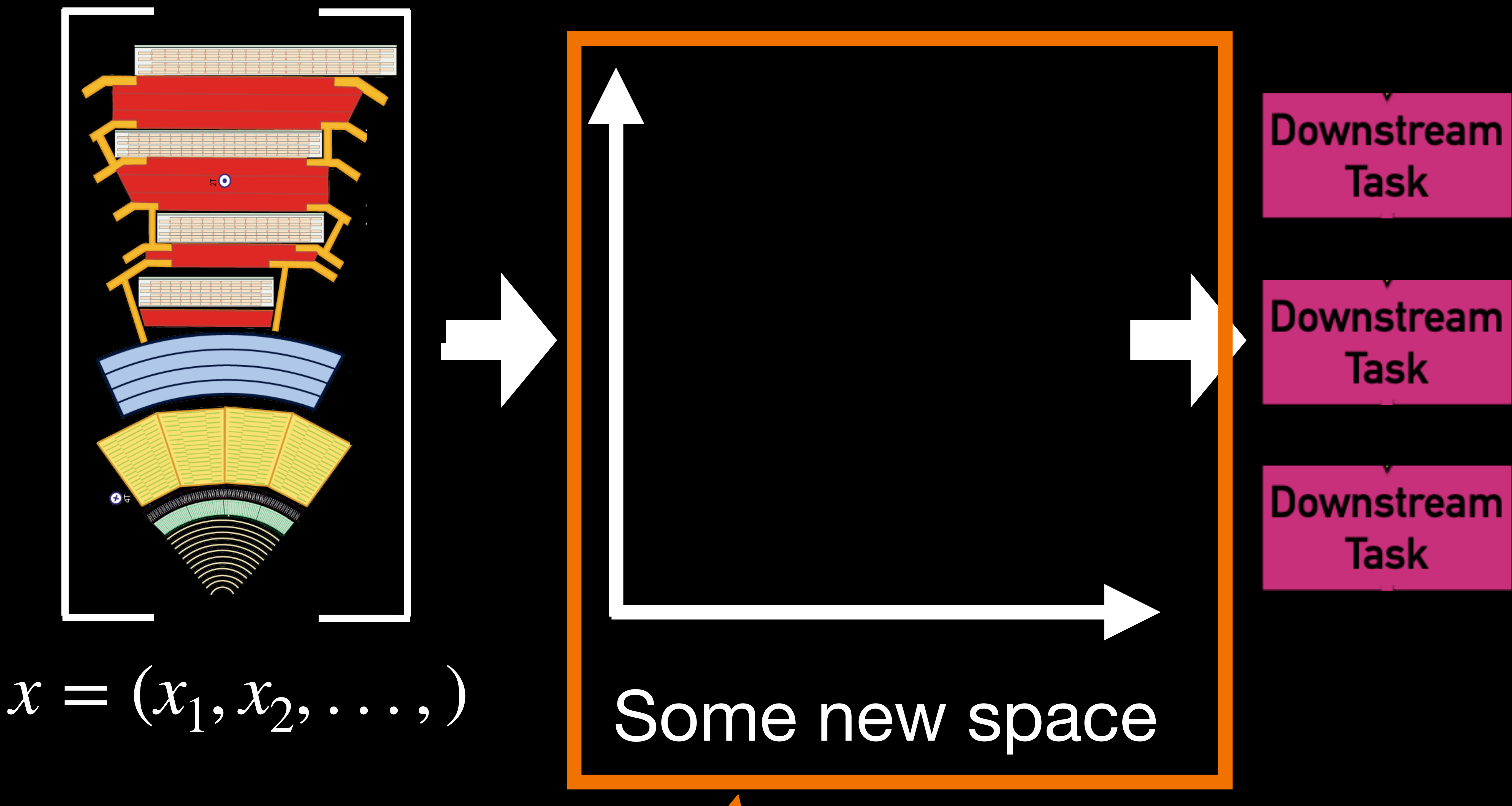
Downstream Task

Downstream Task

Downstream Task

μs

Accept / Reject



$x = (x_1, x_2, \ldots,)$

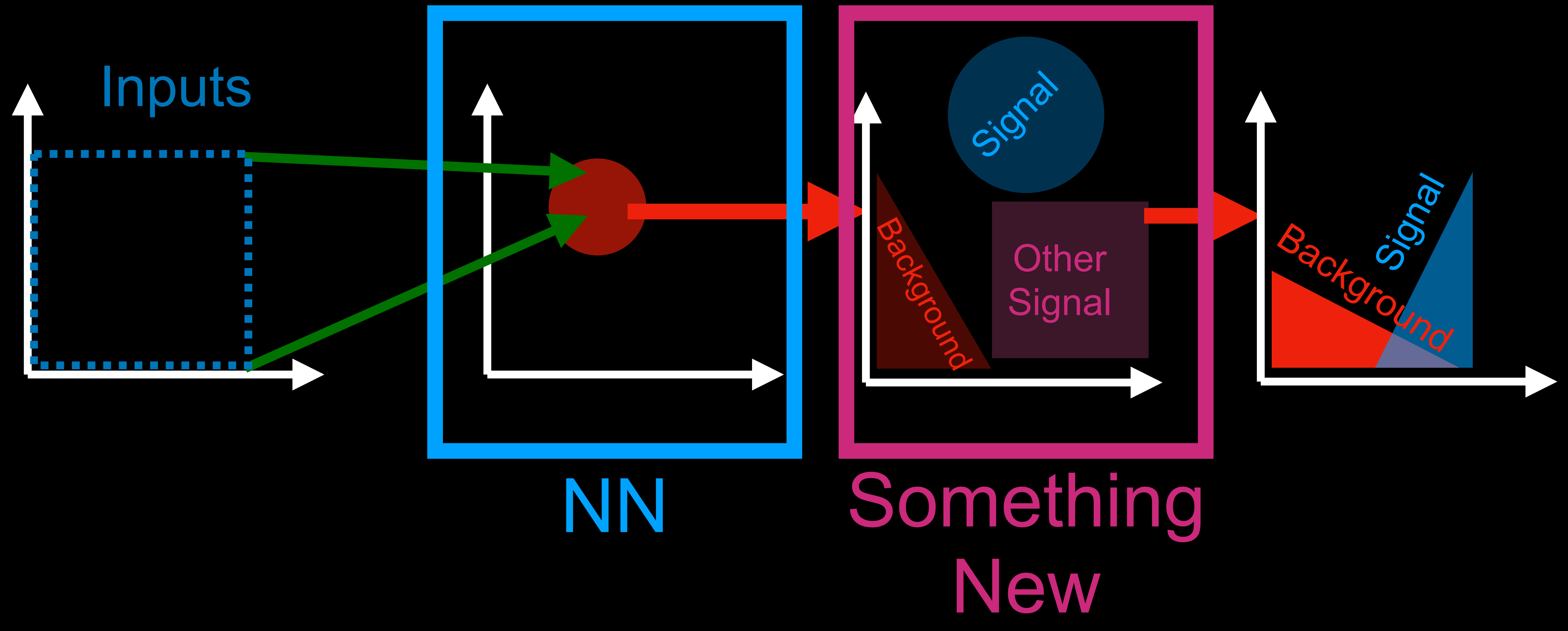
Some new space

Downstream Task

Downstream Task

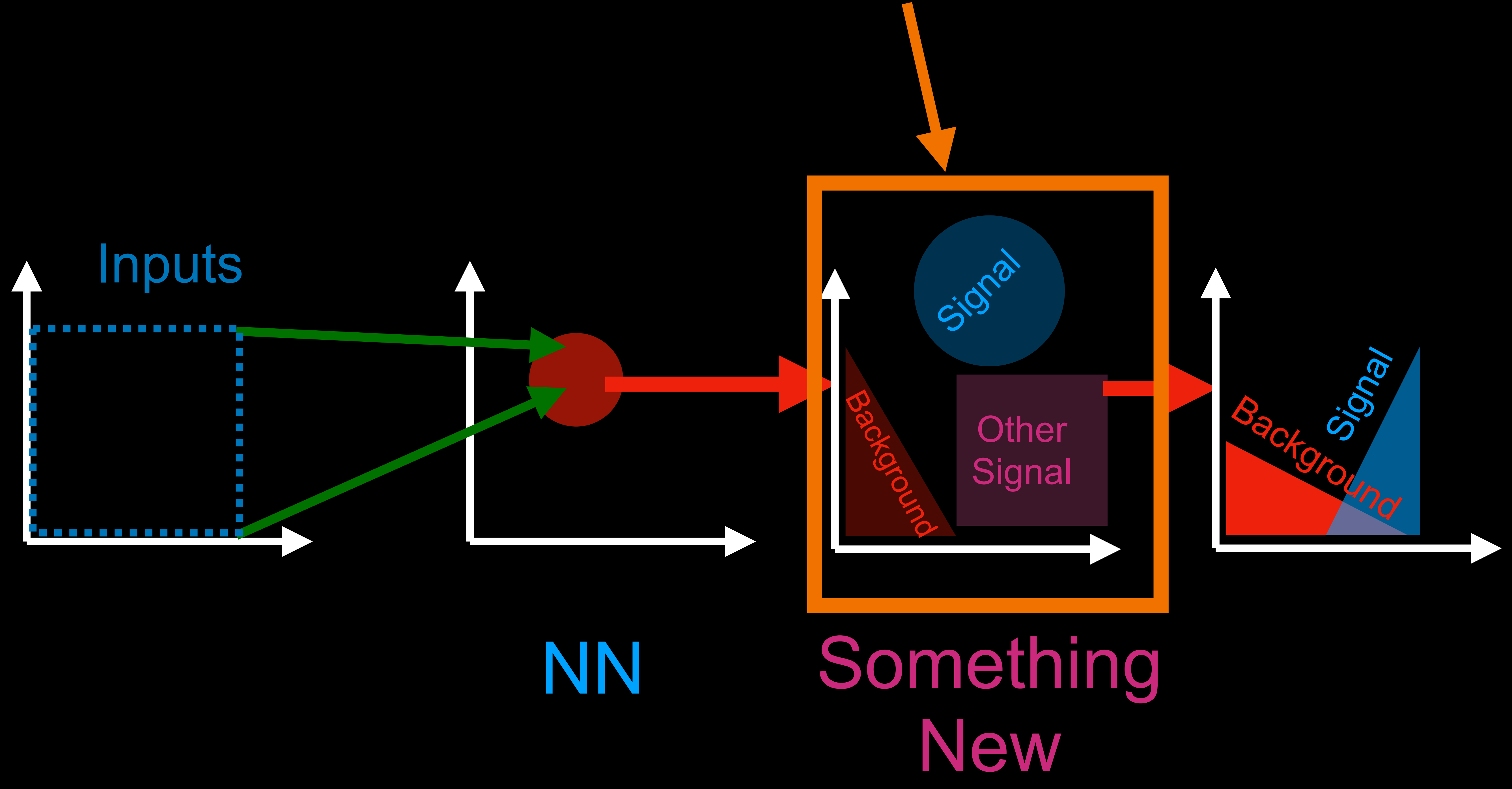
Downstream Task

Let's build this space!



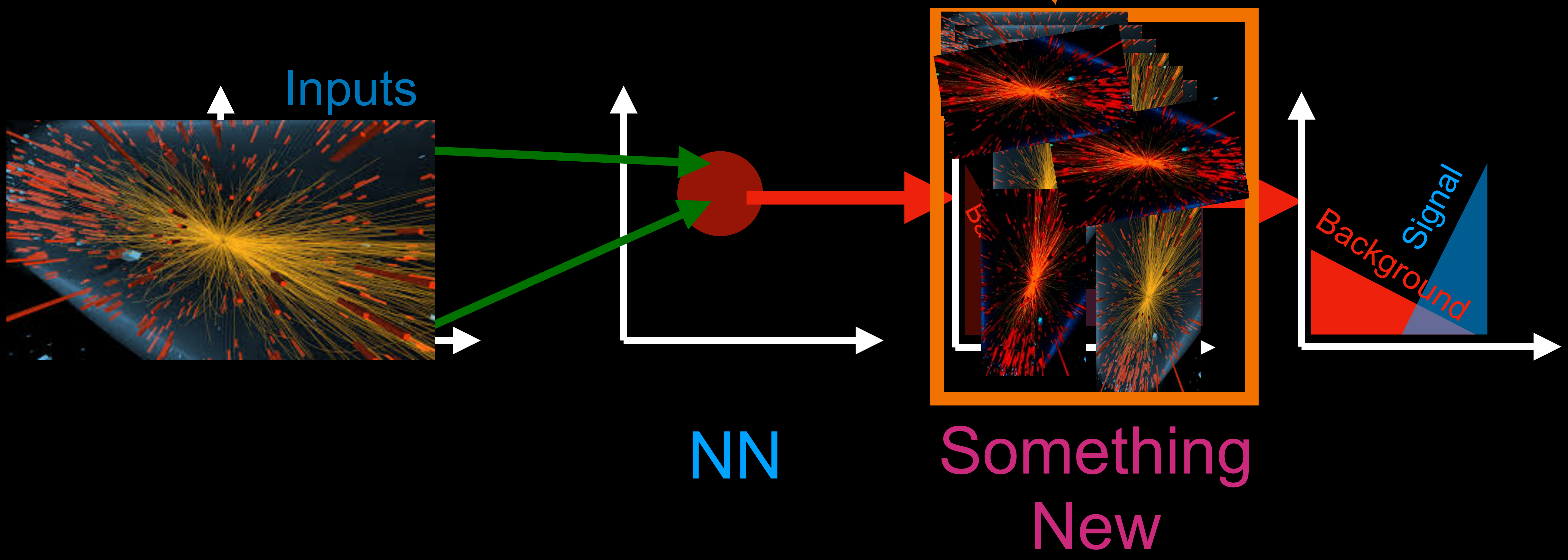
Metric Learning

What if we really try to focus on this space



Neural embedding

What if we really try to focus on this space



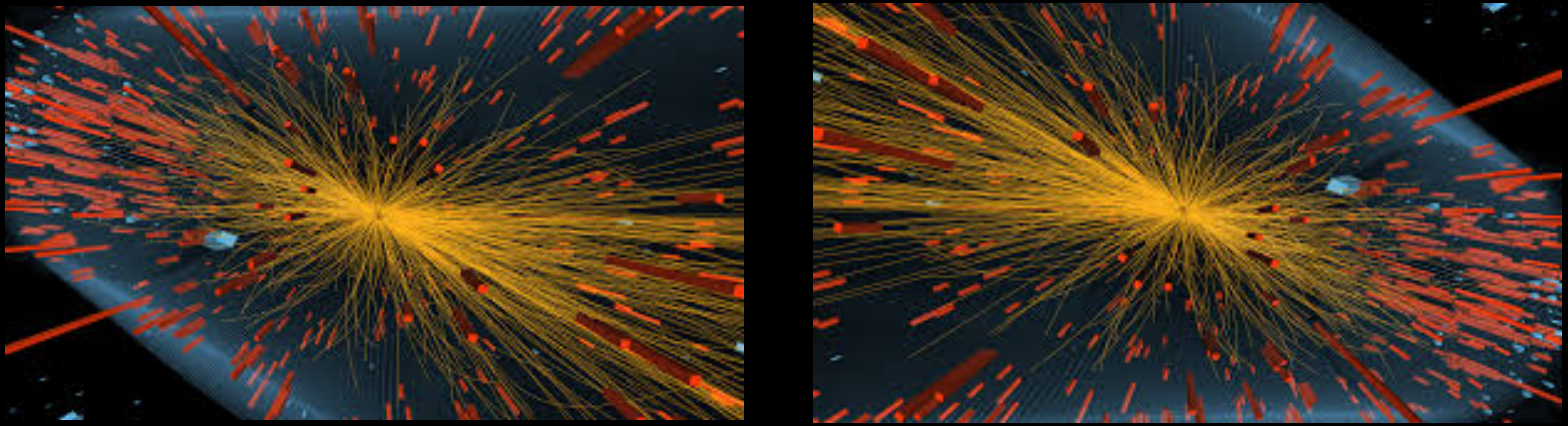
Inputs

NN

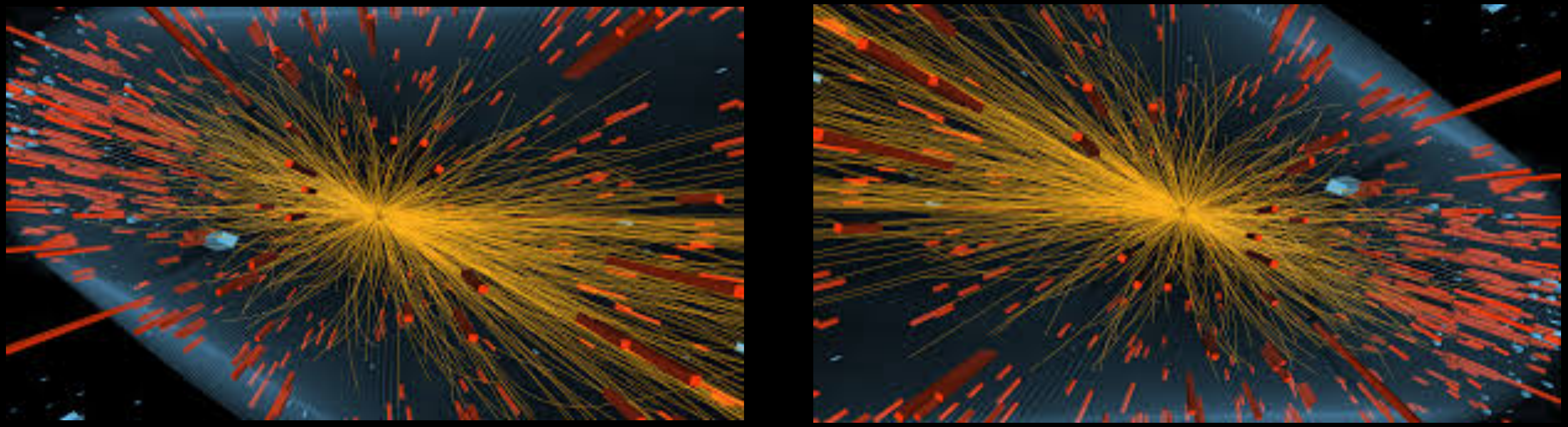
Something New

Neural embedding

Learning the space



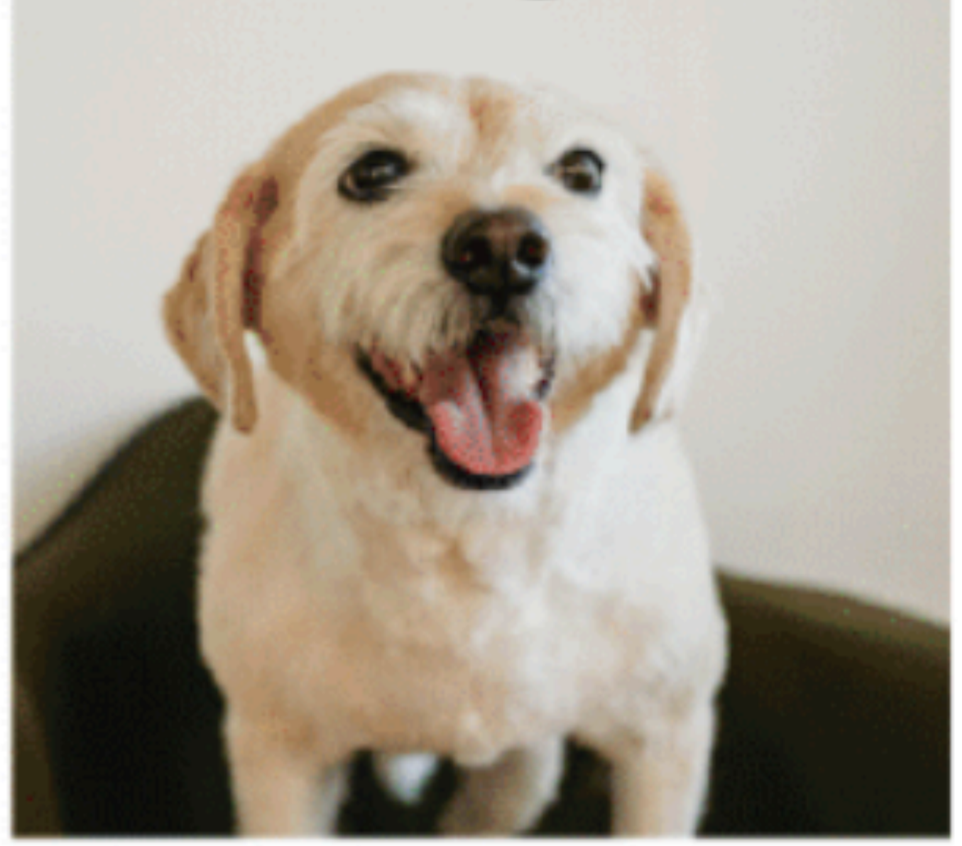
Learning the space



- By looking at data, we can learn a lot
  - Go over input piece by piece
  - Analyze every aspect
  - Compare every feature
- Find distinctive style of the input
  - can be done e.g by looking for a deviation

Cat A

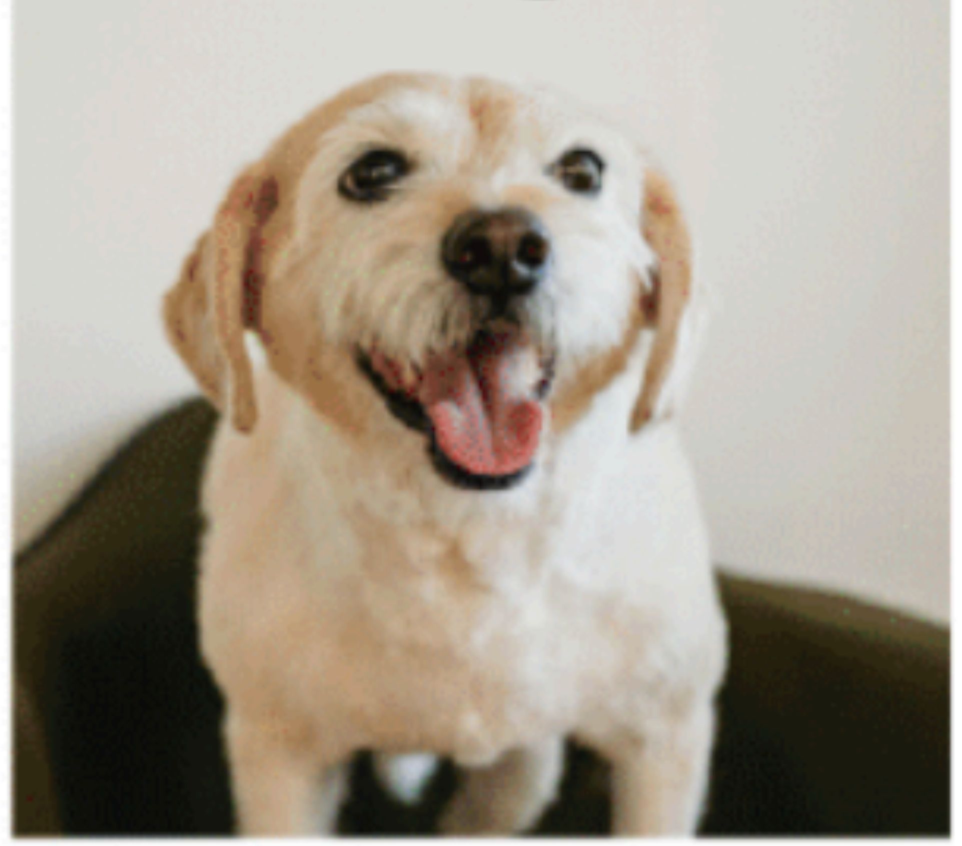
Dog A



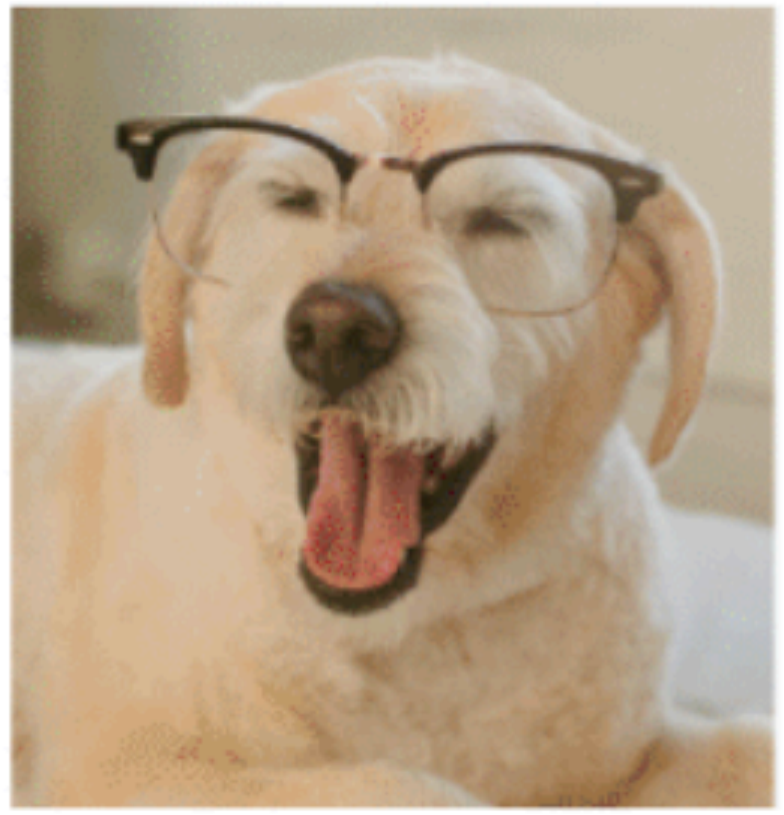
Cat A

Augmented Cat A

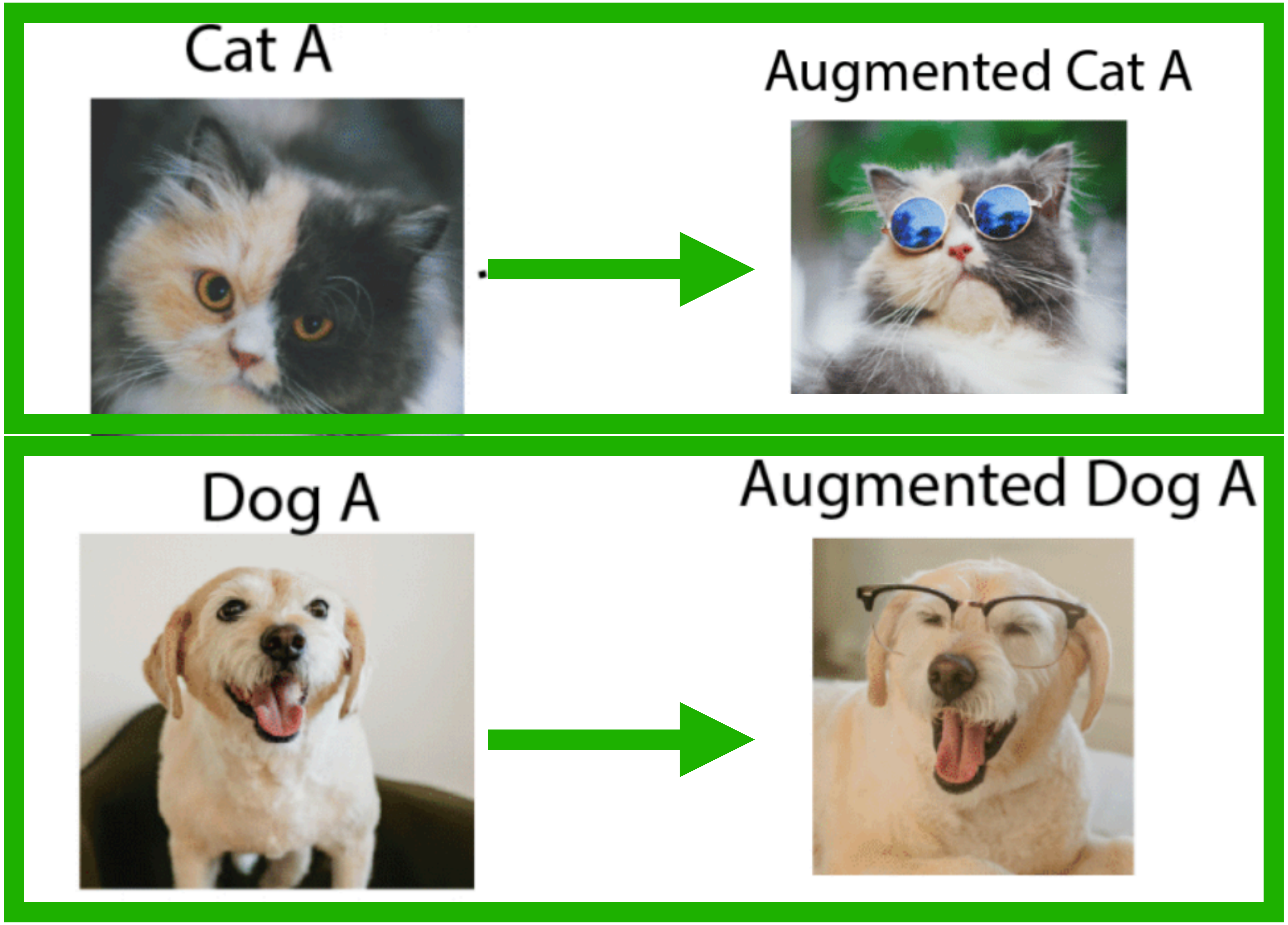
Dog A



Augmented Dog A



Minimize



Cat A

Augmented Cat A

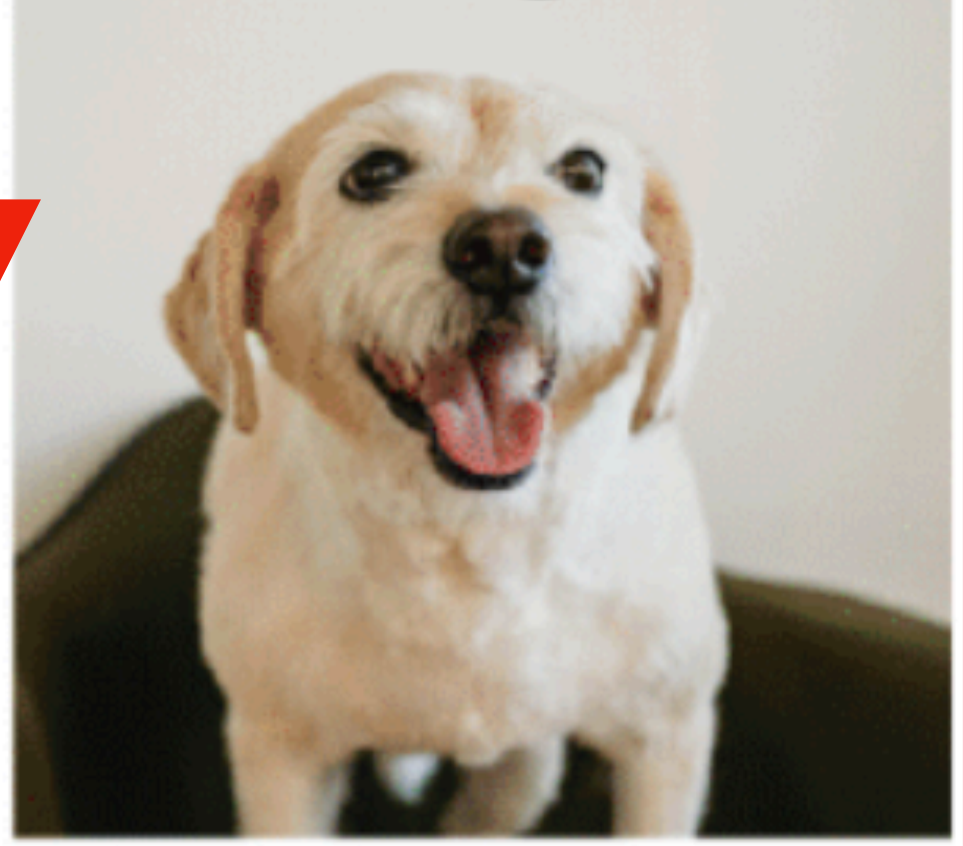
Dog A

Augmented Dog A

Maximize

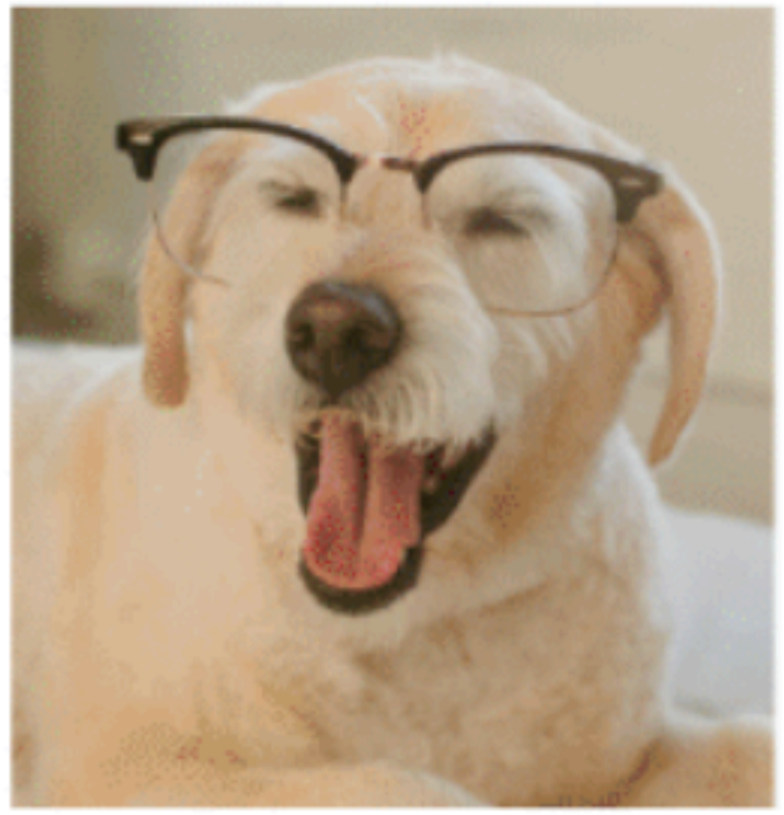
Cat A

Dog A

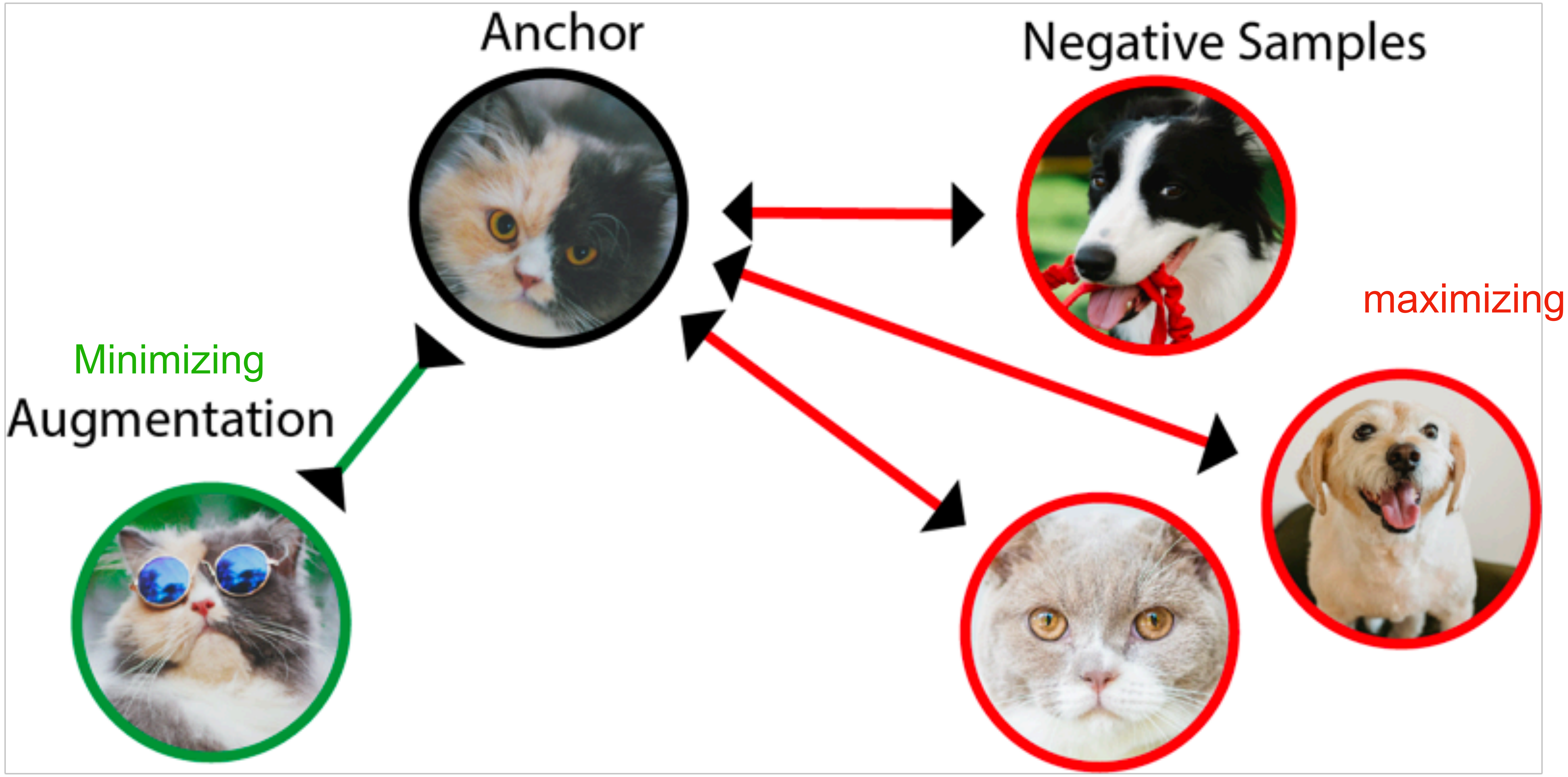


Augmented Cat A

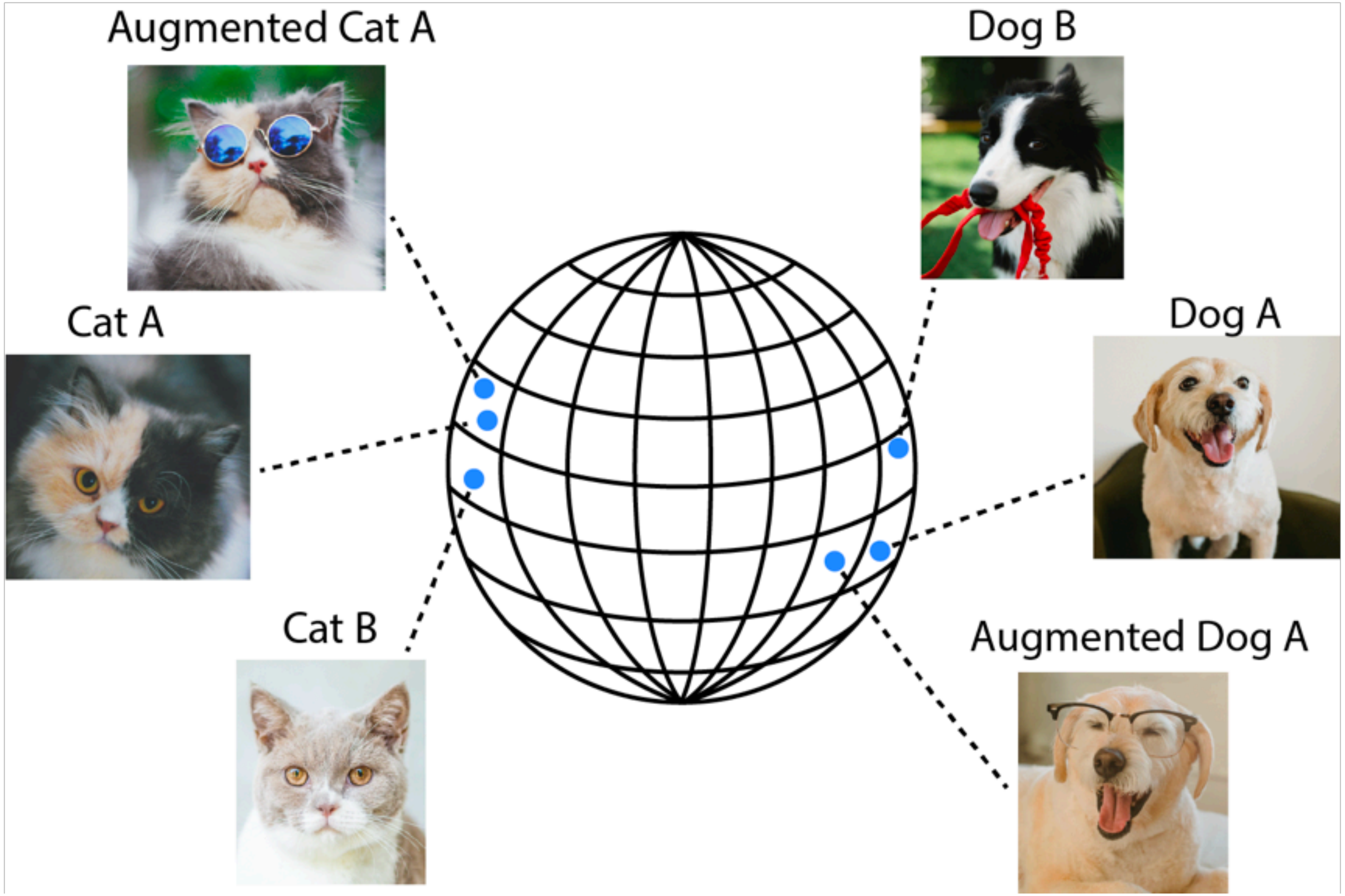
Augmented Dog A



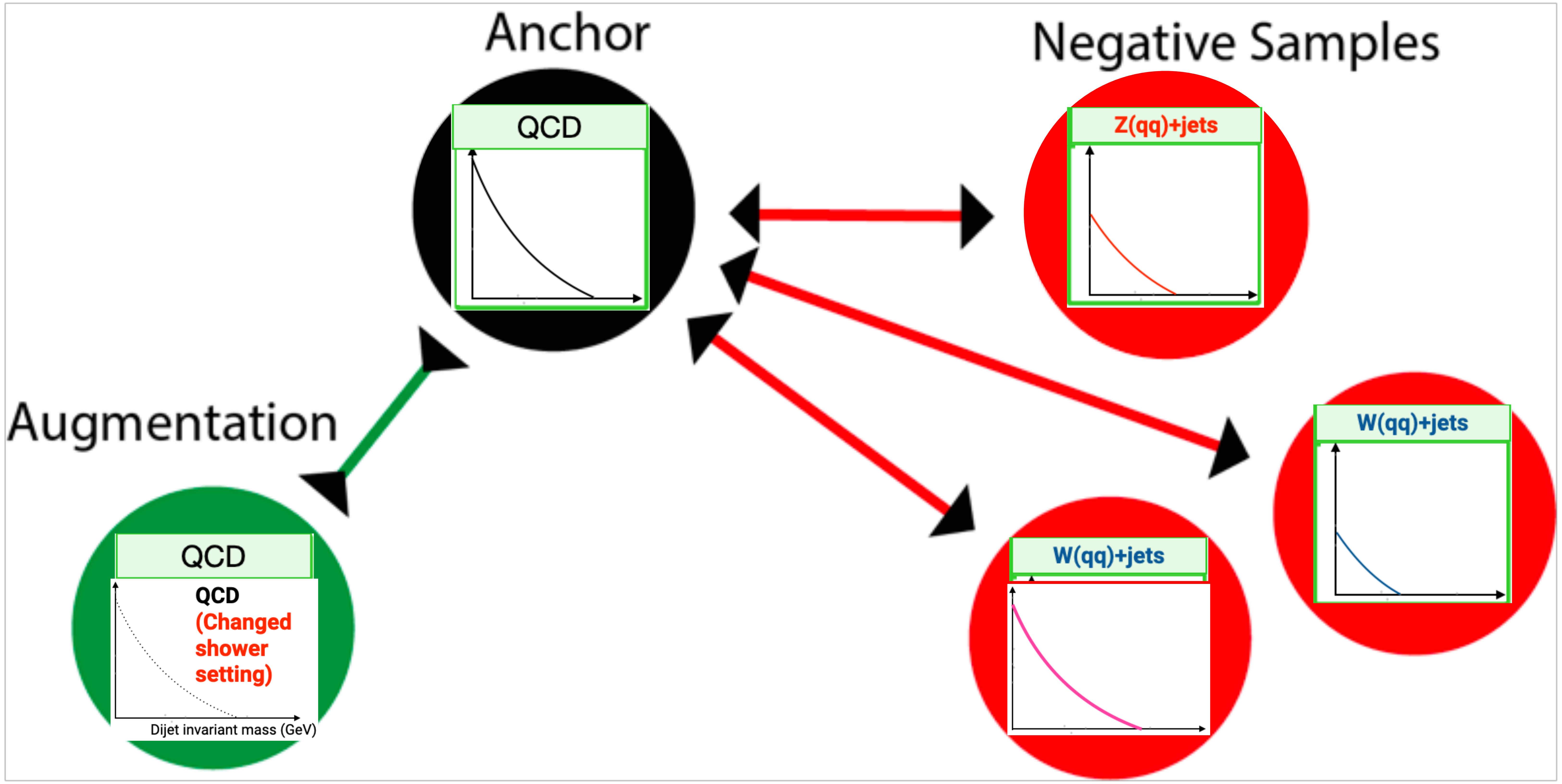
Physically motivated augmentations?



- Minimizing and maximizing distances learns a space

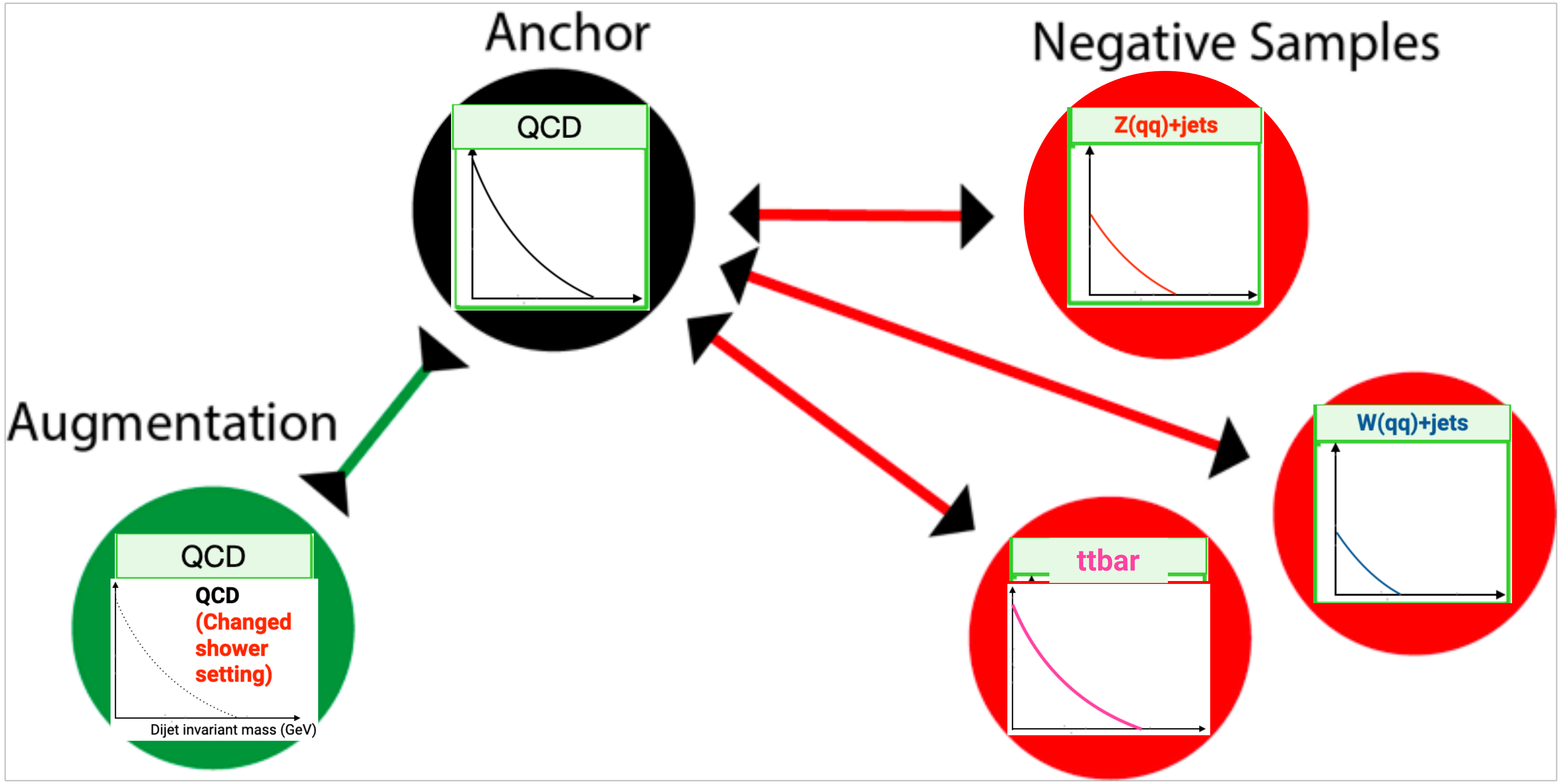

Augmented Cat A
Dog B
Cat A
Dog A
Cat B
Augmented Dog A

Physically motivated augmentations?

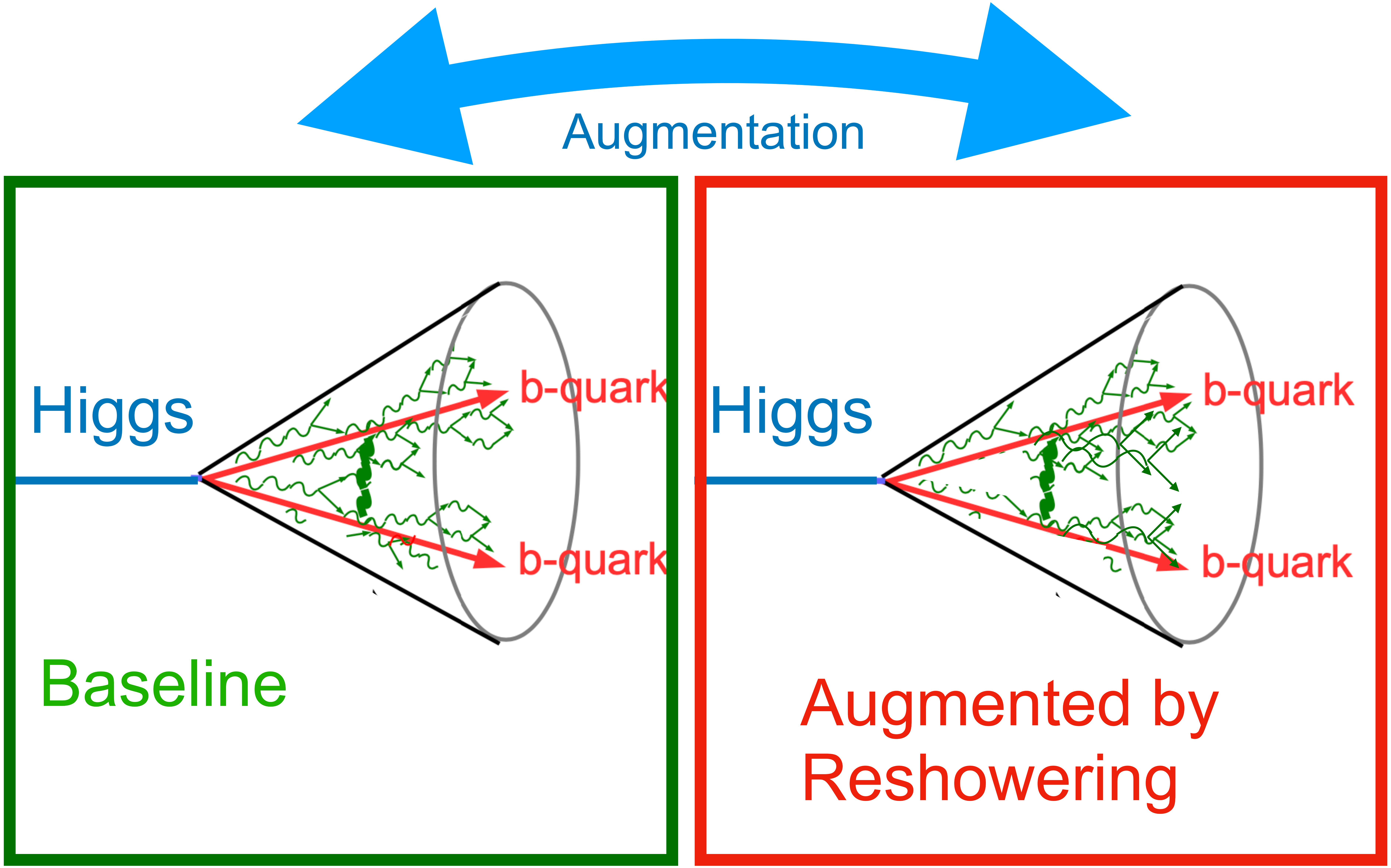


No class labels used in training! How do we augment detector data?

Physically motivated augmentations?



No class labels used in training! How do we augment detector data?



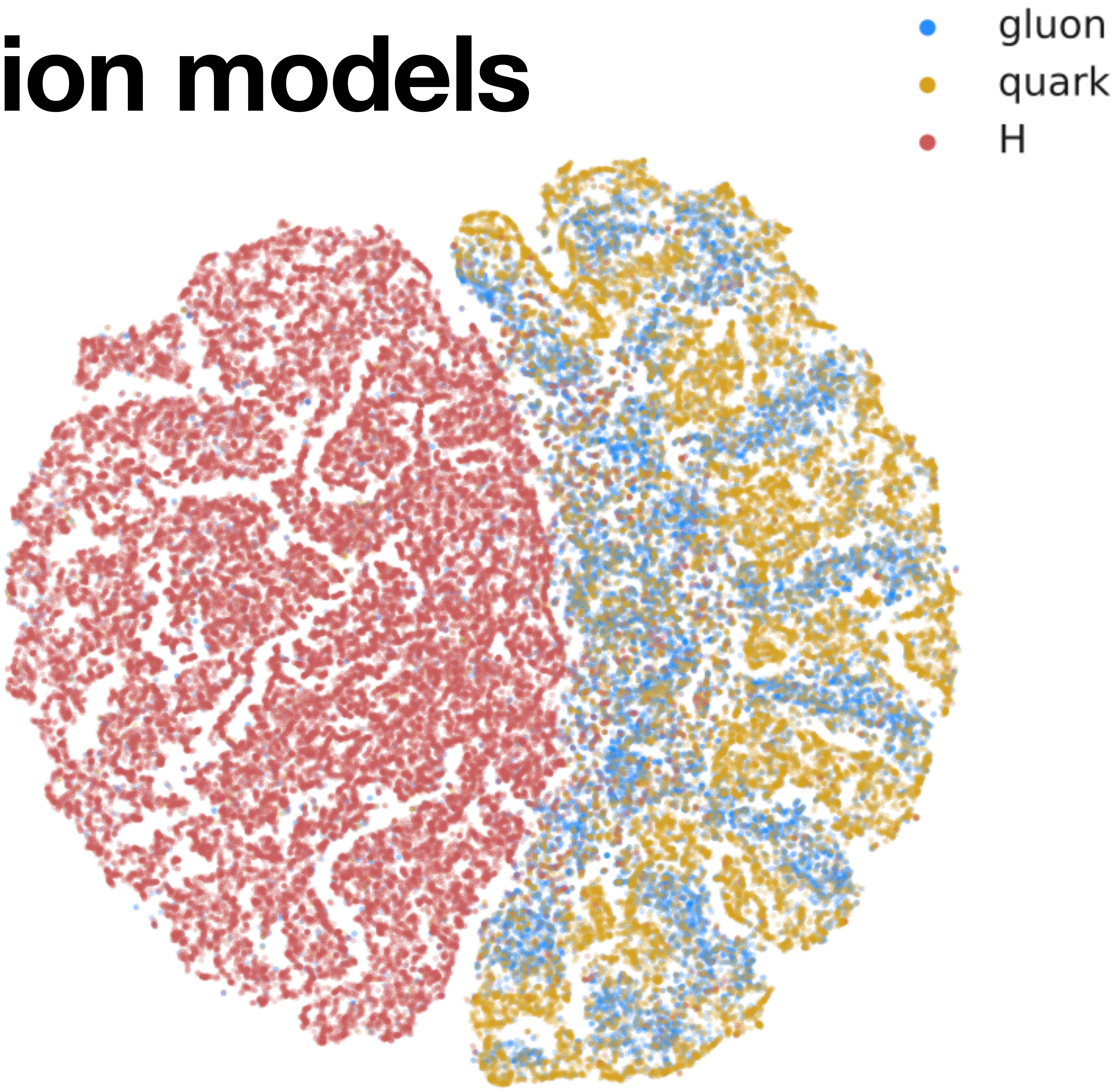
Augmentation

Baseline

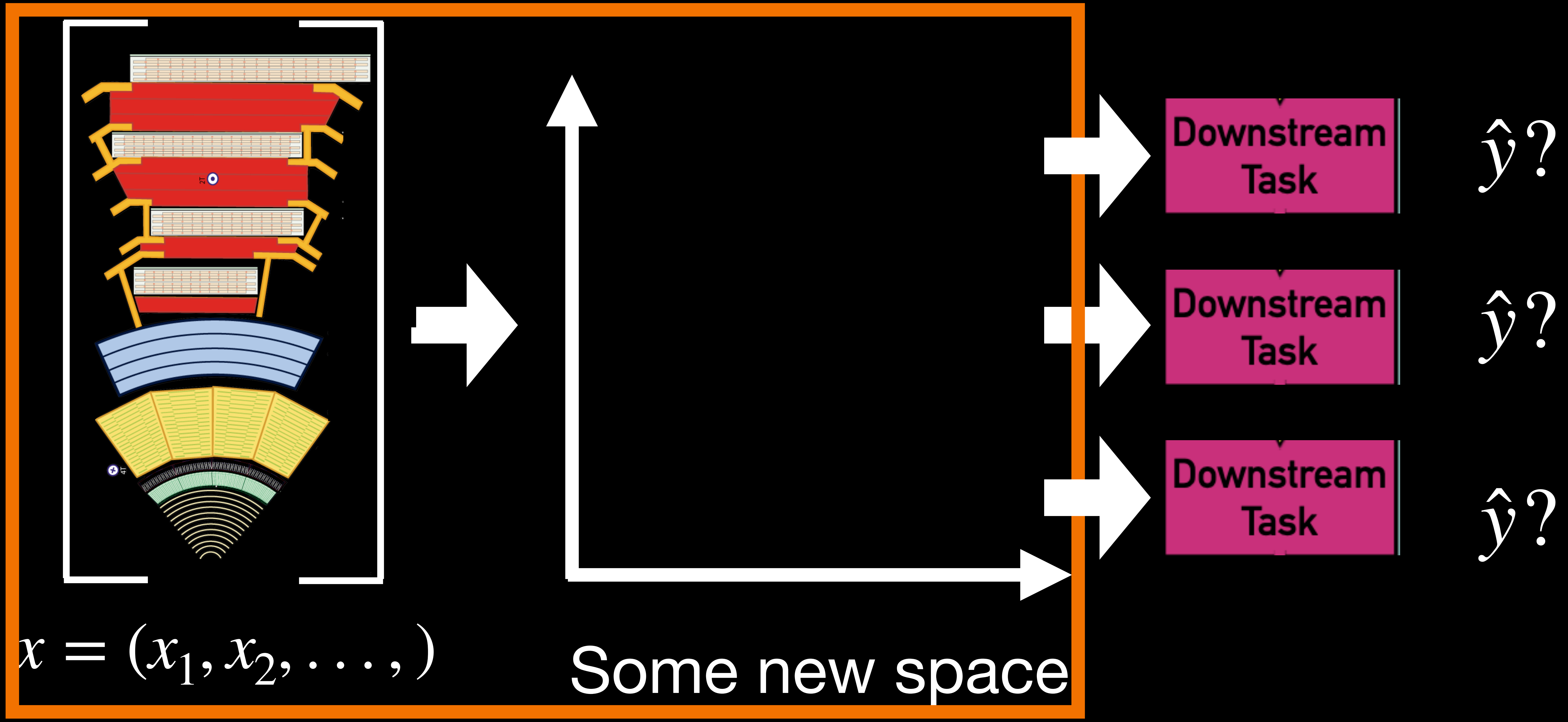
Augmented by Reshowering

Embedded Space can use any NN to embed

QM foundation models



→ embedding quantum mechanics into AI algorithm



$x = (x_1, x_2, \ldots,)$

Some new space

Downstream Task

$\hat{y}?$

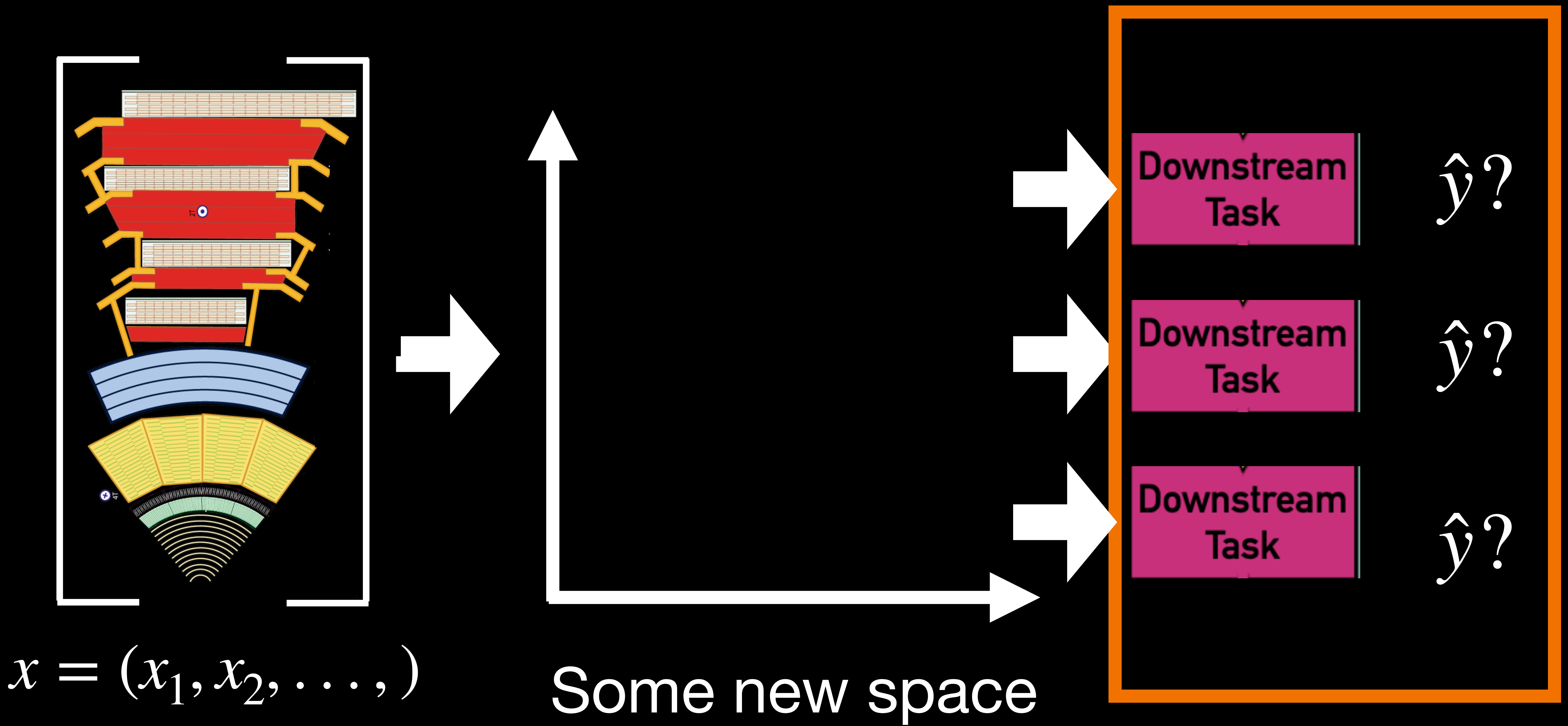
Downstream Task

$\hat{y}?$

Downstream Task

$\hat{y}?$

Training 1: Learn neural embedding
(on a lot of data, for a long time)
On simulation? On data?



$x = (x_1, x_2, \ldots,)$

Some new space

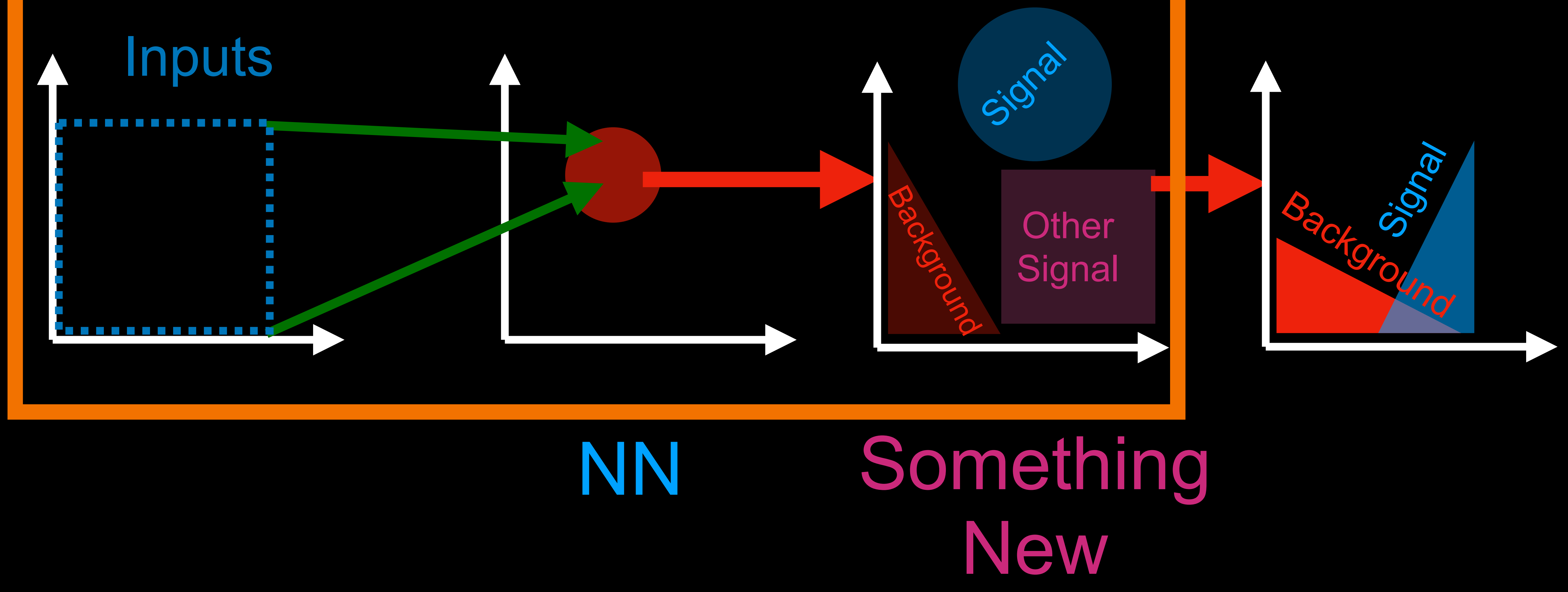
Downstream Task $\hat{y}$?

Downstream Task $\hat{y}$?

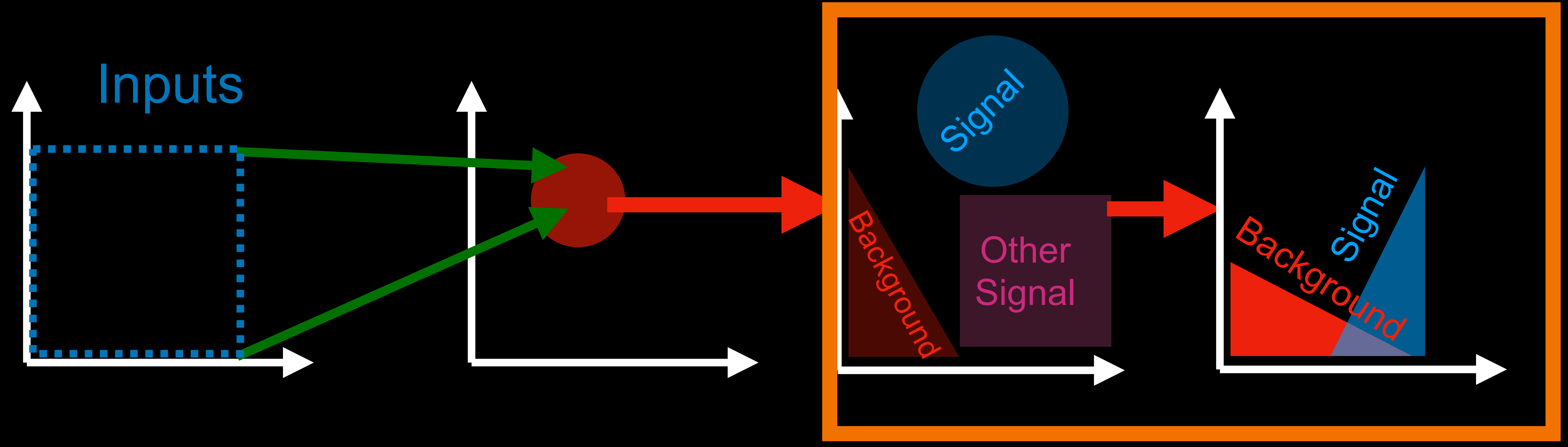
Downstream Task $\hat{y}$?

Training 2: Fine tune for specific task (fast, small dataset, simulation)

Training number one: learn embedding



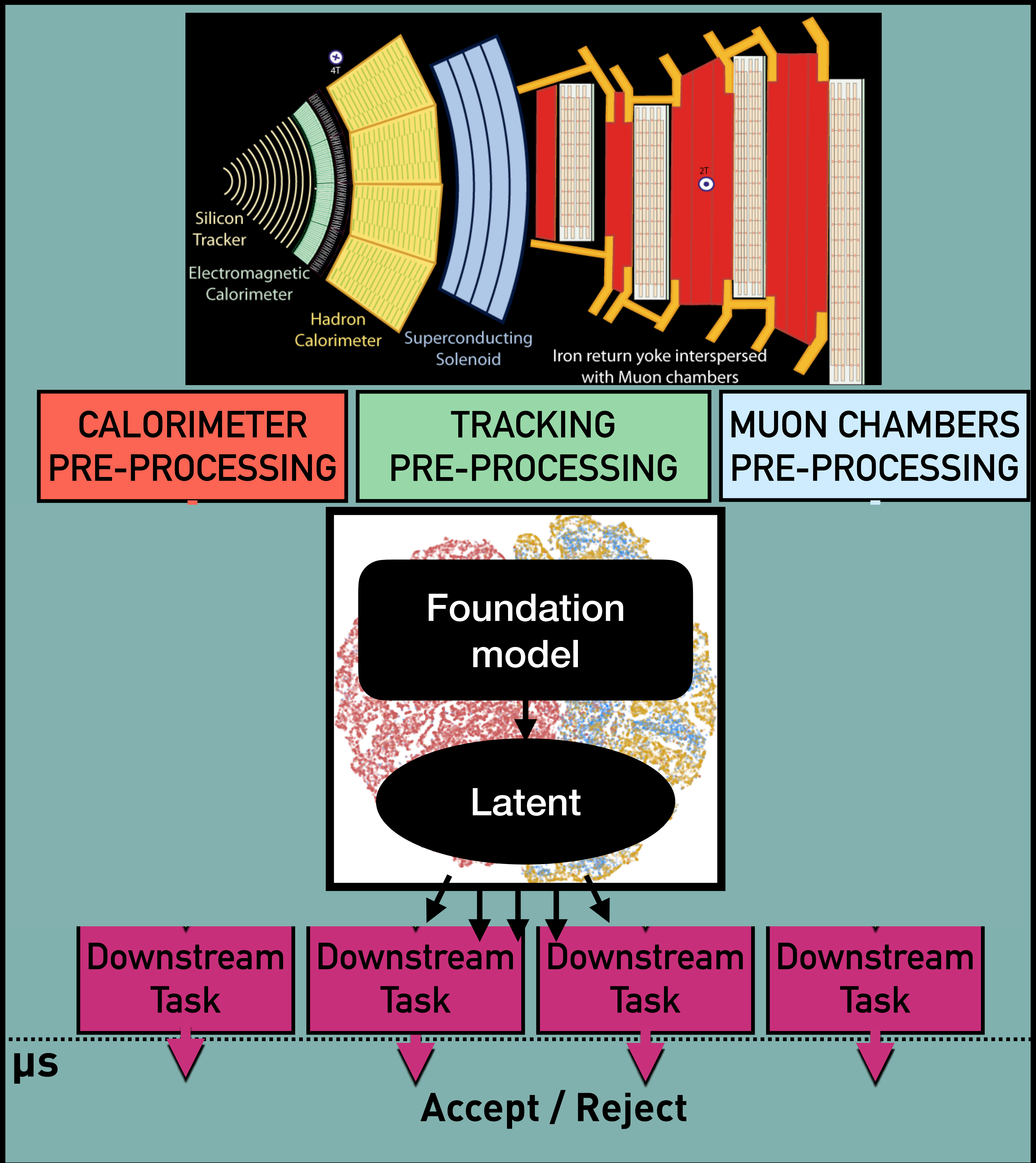
Training number two: fine tuning



NN

Something New

Foundation model of the Level-1 trigger



CALORIMETER PRE-PROCESSING

TRACKING PRE-PROCESSING

MUON CHAMBERS PRE-PROCESSING

Foundation model

Latent

Downstream Task

Downstream Task

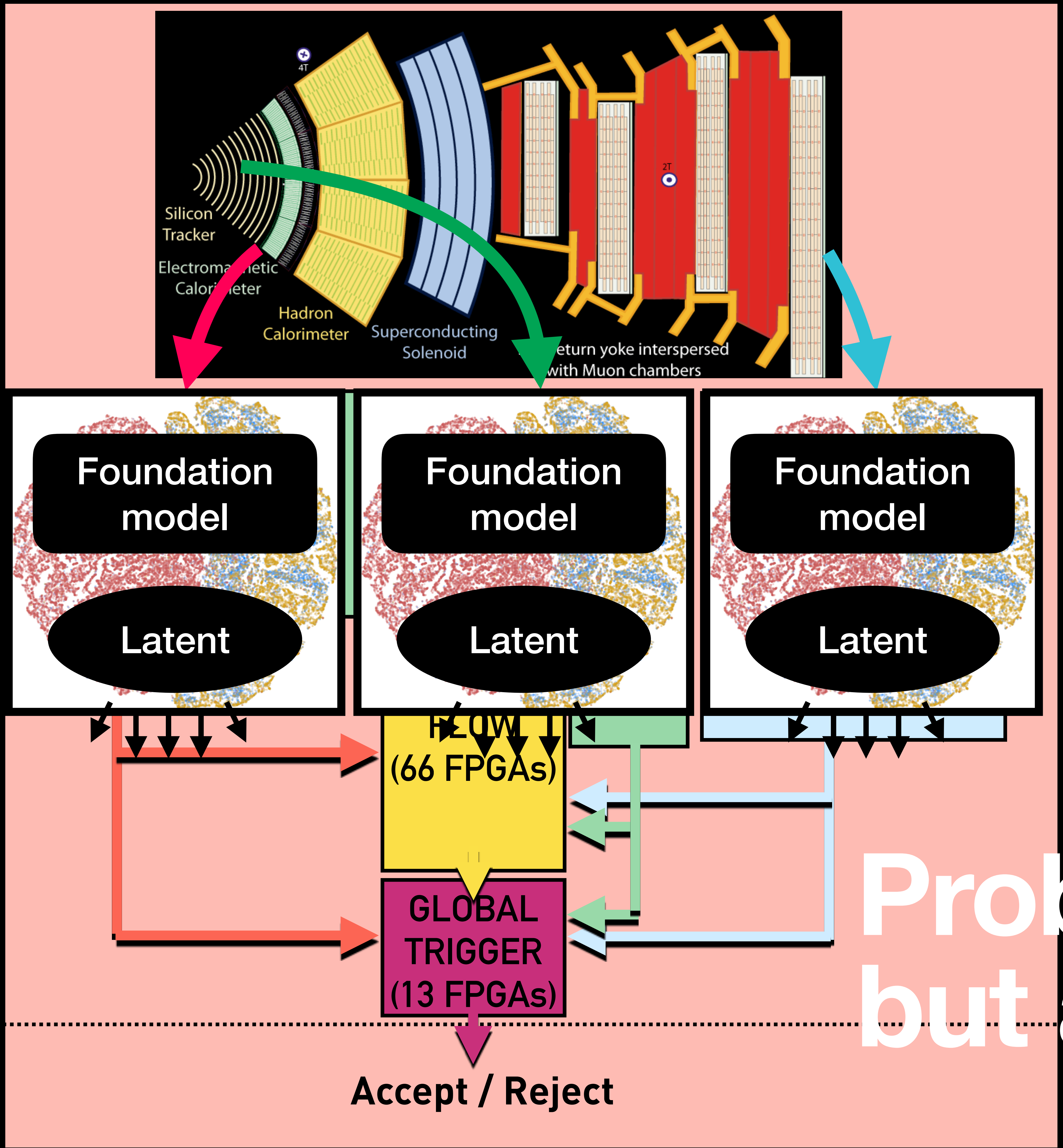
Downstream Task

Downstream Task

μs

Accept / Reject

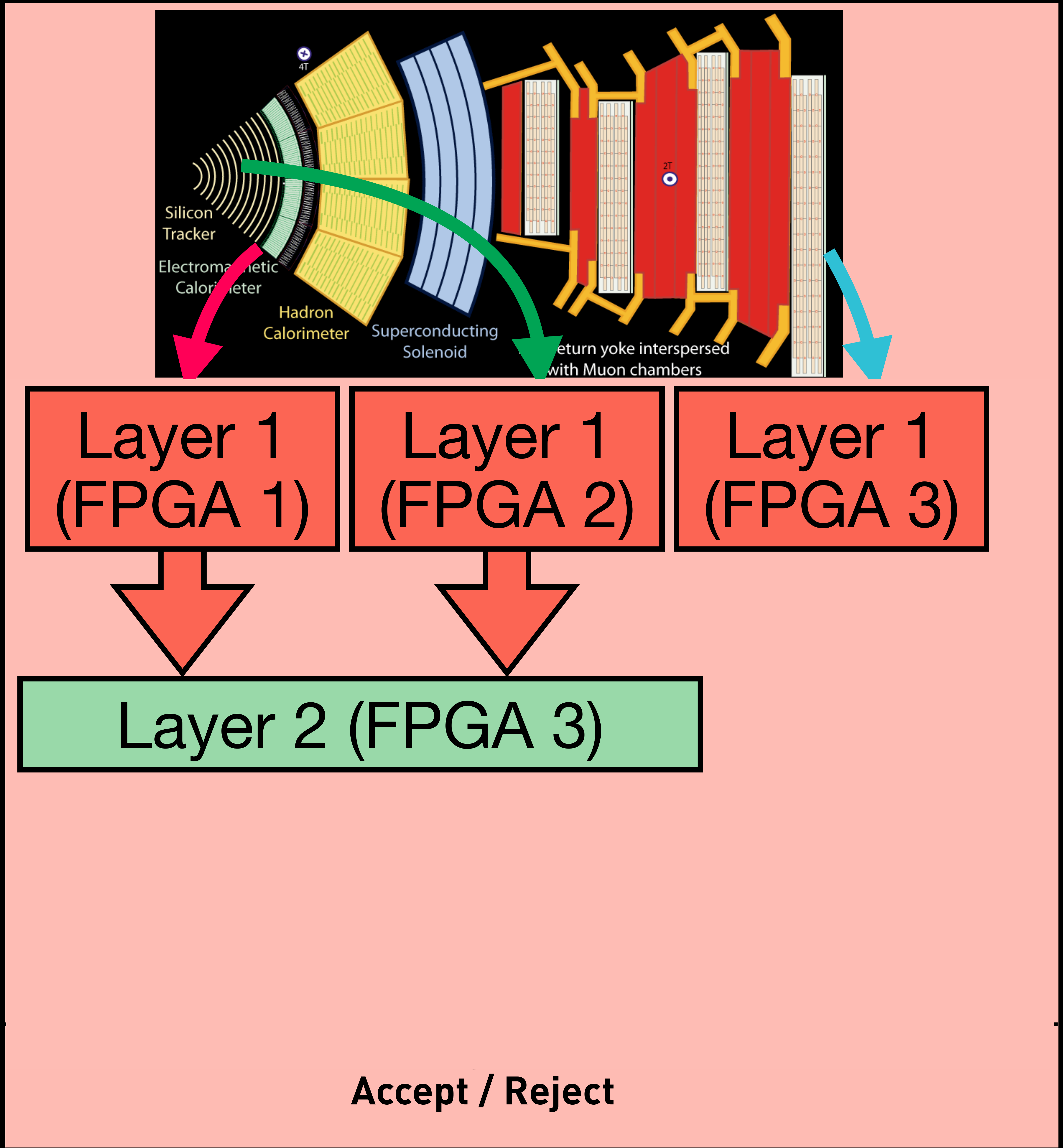
Do I really think this will be possible?



Probably not,
but at some scale

Careful software-hardware co-design

O(1M) parameter model on 1000 FPGAs and do inference in O(1)µs?



Layer 1 (FPGA 1)

Layer 1 (FPGA 2)

Layer 1 (FPGA 3)

Layer 2 (FPGA 3)

Accept / Reject

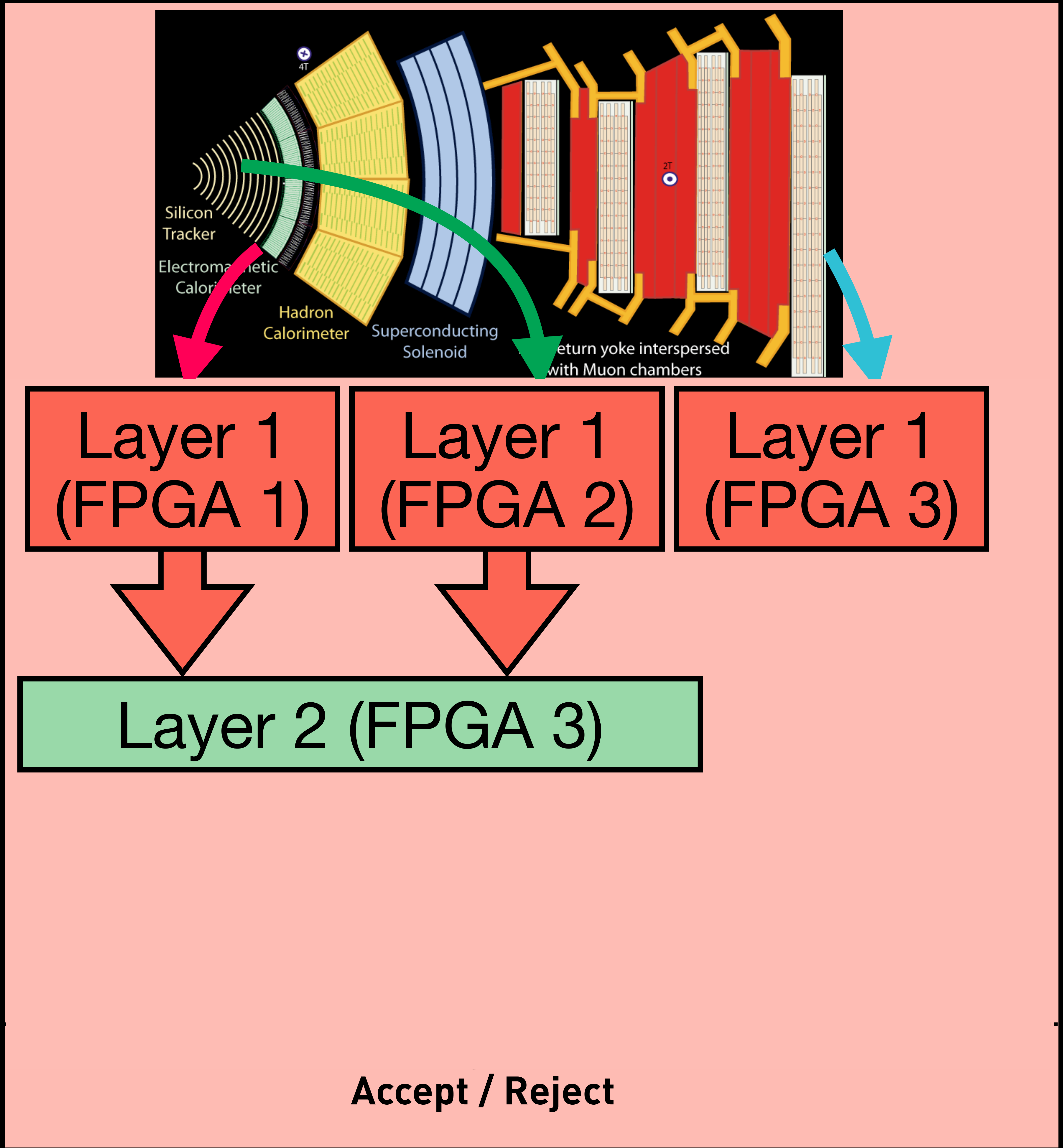
Similar for GPT-4, layers carefully map onto hardware

Algean

Careful software-hardware co-design

Designed our own protocol to make boards talk to each other fast enough

(25 Gbs to transfer data LHC-synchronously between boards)



Layer 1 (FPGA 1)

Layer 1 (FPGA 2)

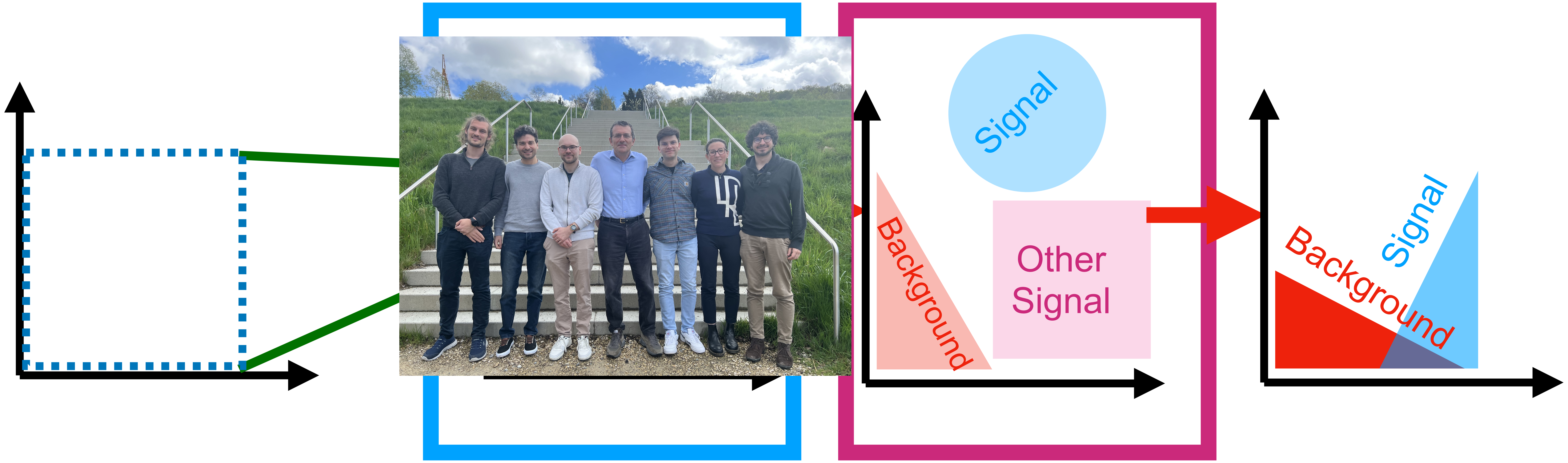
Layer 1 (FPGA 3)

Layer 2 (FPGA 3)

Accept / Reject

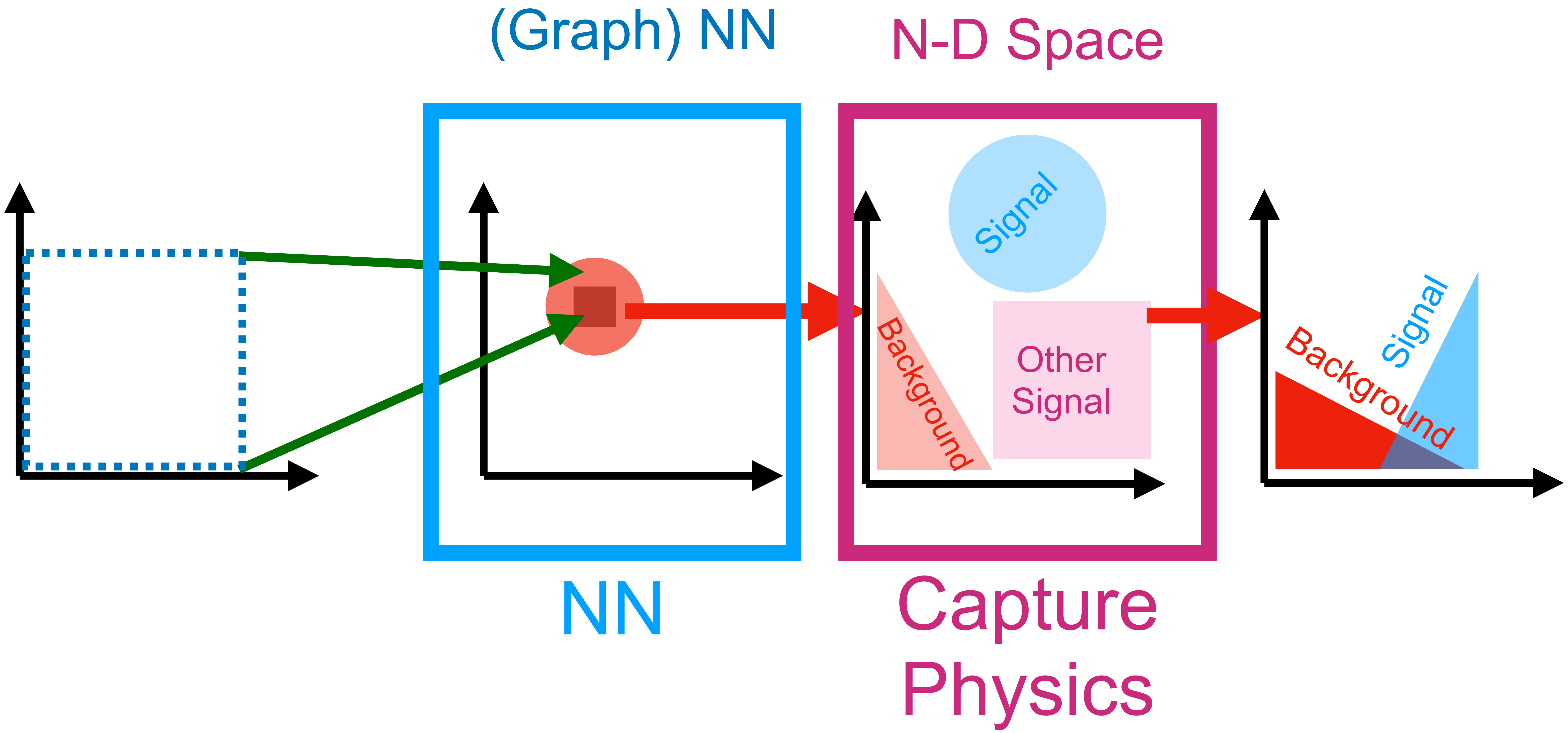
Theorists

N-D Space



Capture Physics

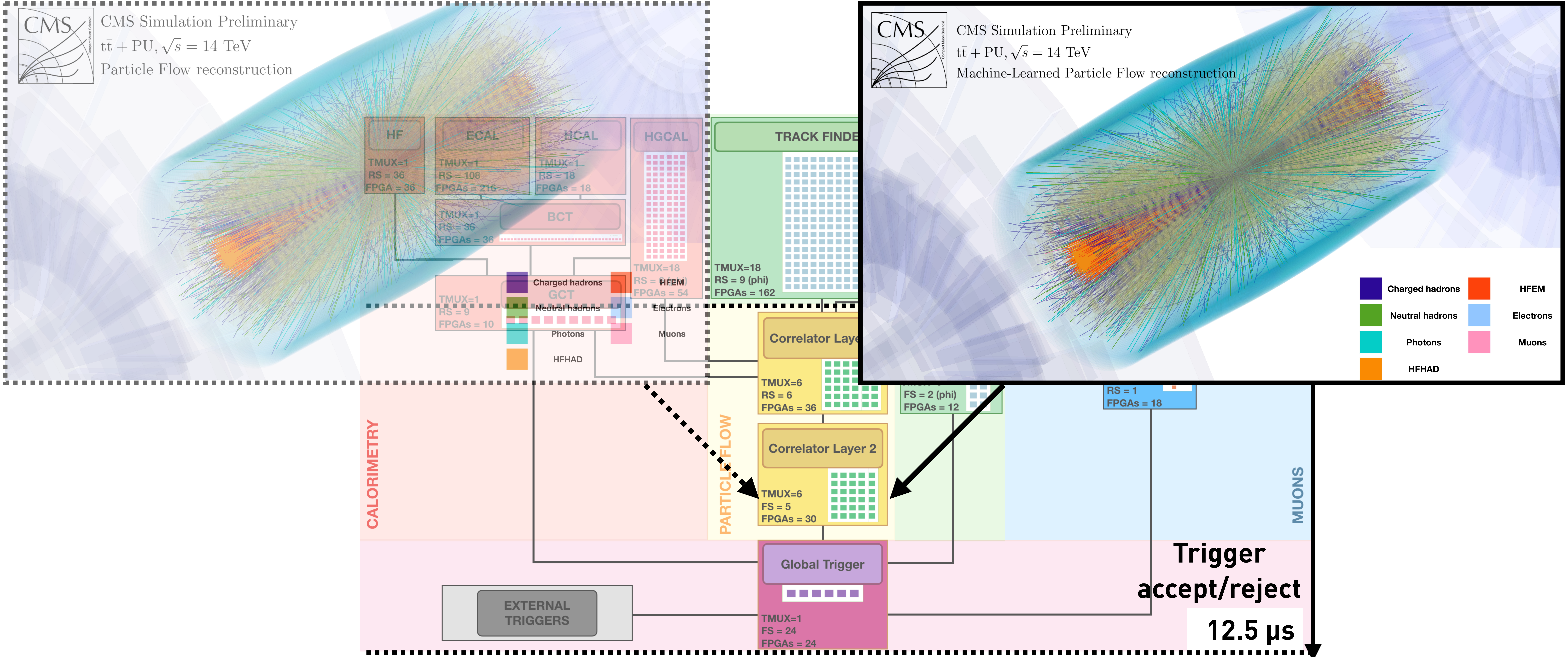
We can replace the QCD theorist with a NN
(And it works better)



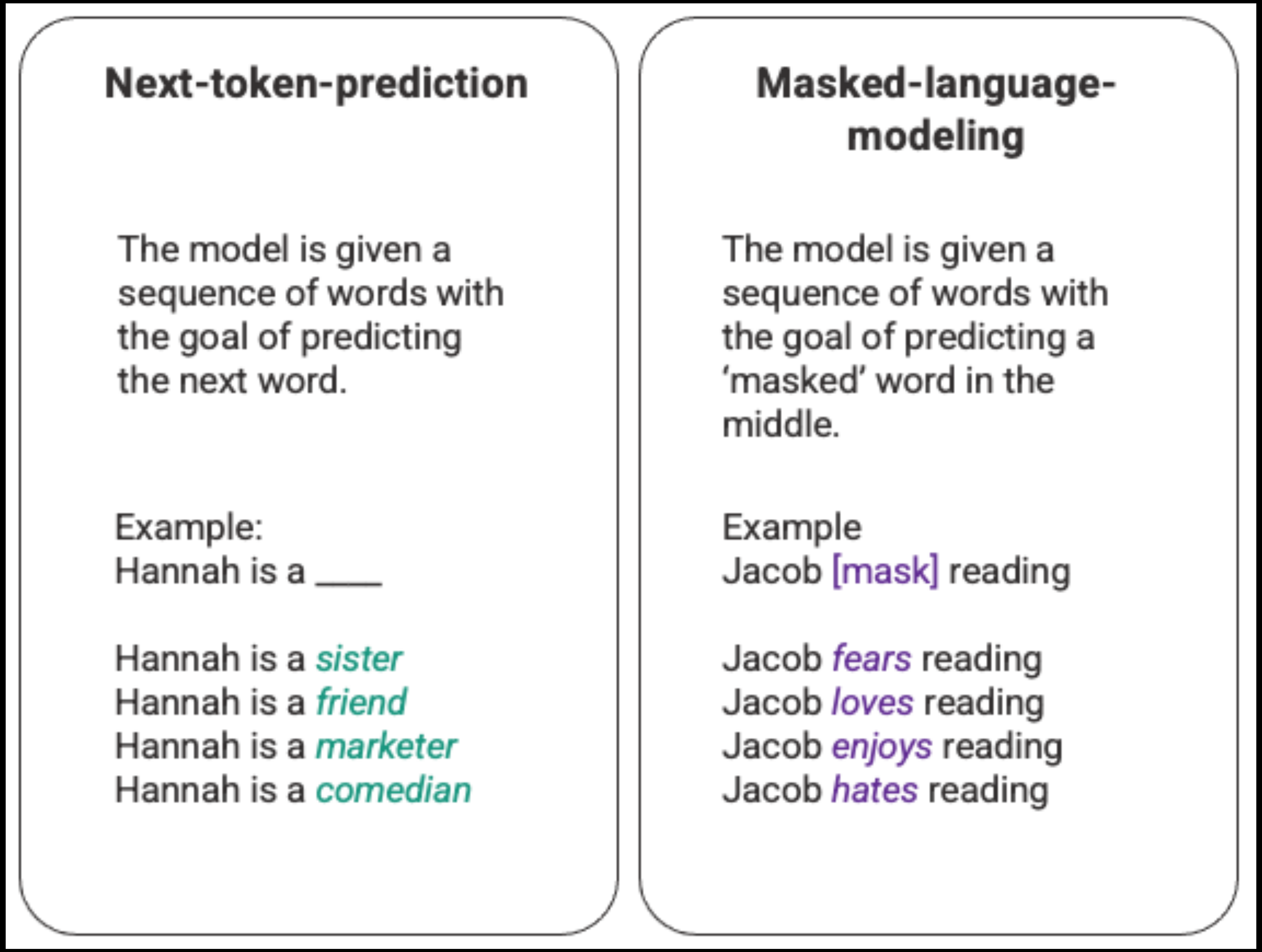
Towards end-to-end reconstruction

Classical Particle Flow

Graph Neural Network



Masked language modelling



Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

Masked-language-modeling

The model is given a sequence of words with the goal of predicting a 'masked' word in the middle.

Example
Jacob [mask] reading

Jacob *fears* reading
Jacob *loves* reading
Jacob *enjoys* reading
Jacob *hates* reading

Self-supervised pre-training