

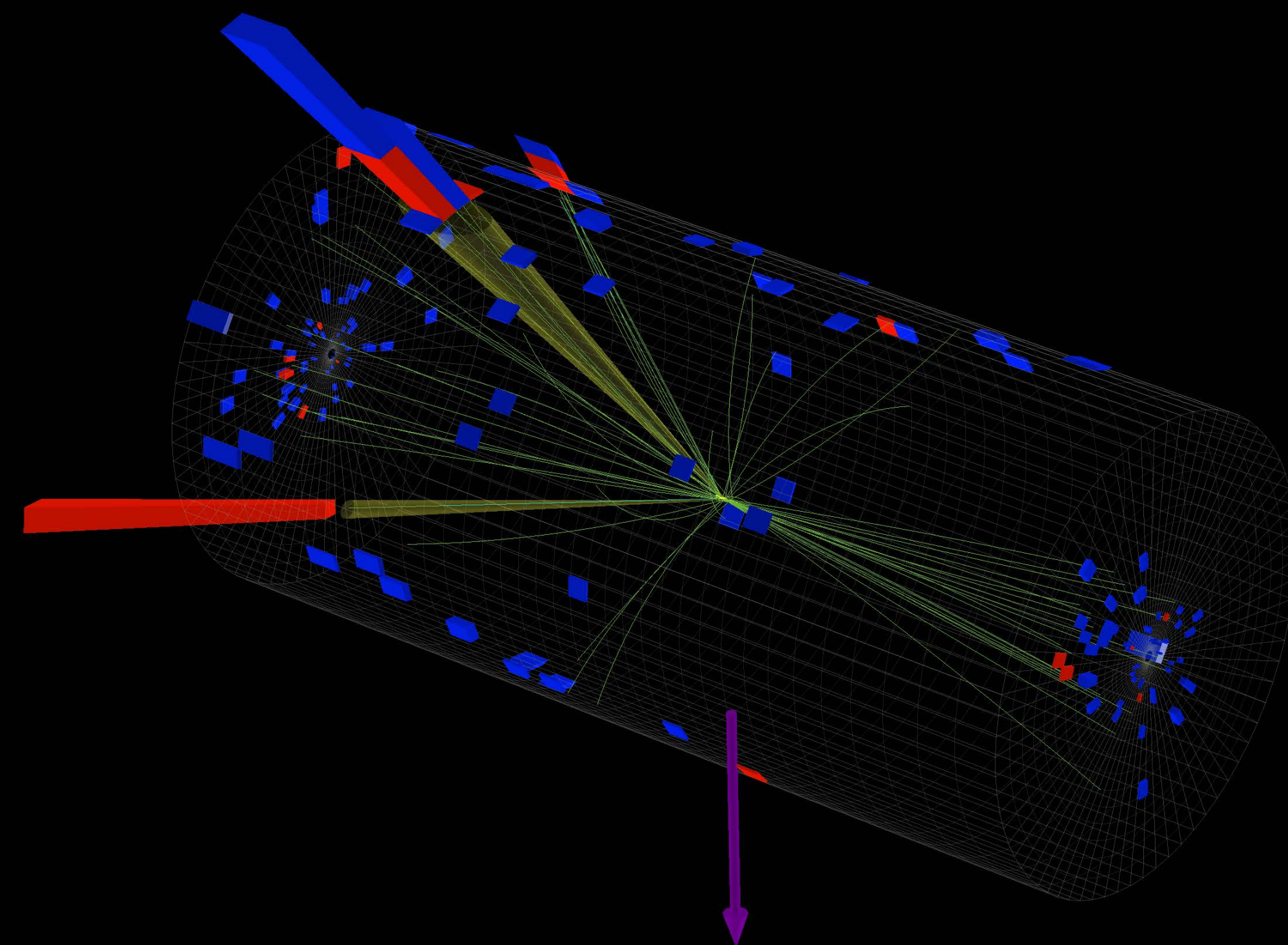
Graph Neural Networks for High Energy Physics

Huilin Qu (曲慧麟)

见微学术沙龙

中国科学技术大学

2023年12月8日

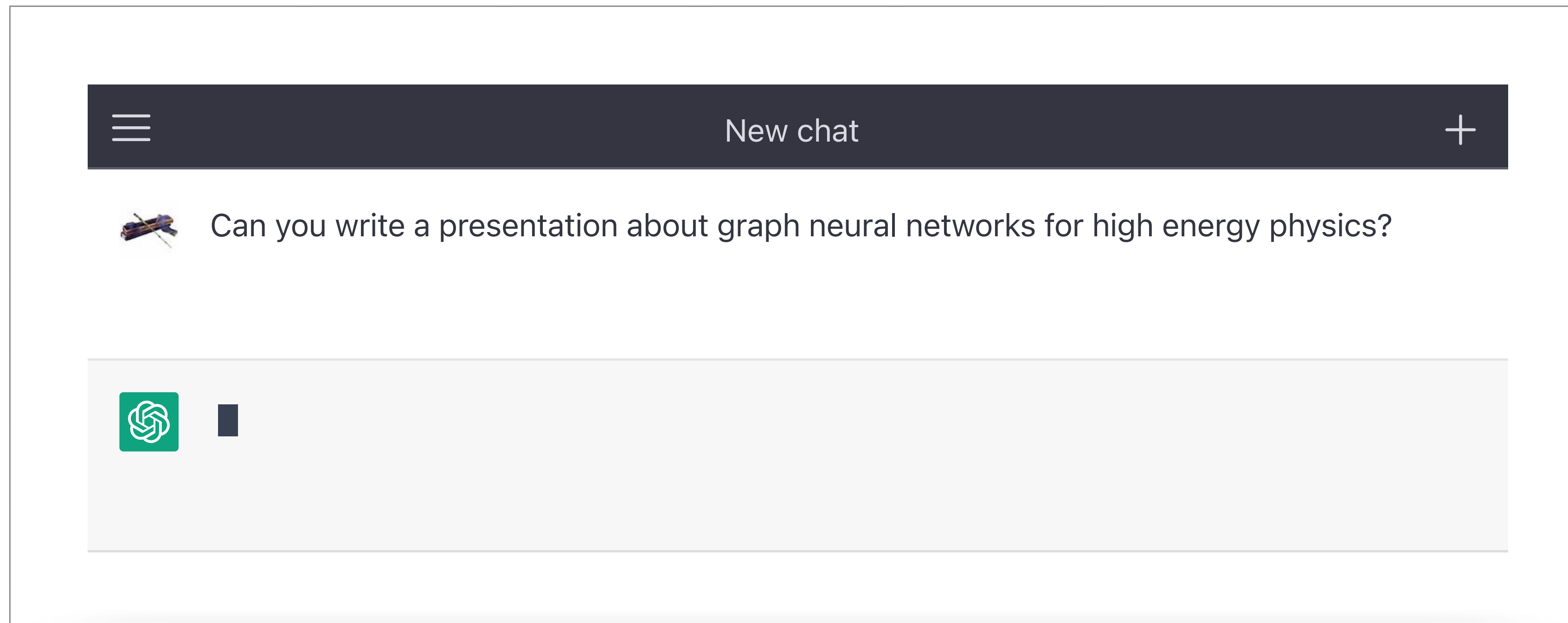


PRELUDE

Since I will be talking about Machine Learning and now is 2023...

PRELUDE

Since I will be talking about Machine Learning and now is 2023...



A few seconds later...

Can you write a presentation about graph neural networks for high energy physics?

Sure, here is a possible presentation about Graph Neural Networks for High Energy Physics:

Title: Graph Neural Networks for High Energy Physics

Introduction:

- Explain what High Energy Physics is and why it is important (e.g., studying fundamental particles and forces, exploring the origins of the universe, developing new technologies).
- Mention that High Energy Physics generates large amounts of data, and that analyzing this data is a major challenge.
- Introduce the concept of Graph Neural Networks (GNNs) as a promising tool for analyzing data that has a graph structure, such as particle collisions.

Body:

- Explain what GNNs are and how they work, using a simple example (e.g., predicting node labels in a social network).
- Emphasize that GNNs can be used to learn from the topology of a graph, as well as from the features associated with its nodes and edges.
- Show how GNNs can be applied to High Energy Physics, where the graph corresponds to a particle collision and the nodes and edges represent particles and their interactions.
- Describe some specific applications of GNNs in High Energy Physics, such as:
 - Particle tracking: using GNNs to reconstruct the paths of particles from detector measurements.
 - Event classification: using GNNs to distinguish between different types of particle collisions.
 - Anomaly detection: using GNNs to identify rare or unexpected events that may

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

- Event classification: using GNNs to distinguish between different types of particle collisions.
- Anomaly detection: using GNNs to identify rare or unexpected events that may indicate new physics.

• Discuss some of the challenges of using GNNs in High Energy Physics, such as:

- Dealing with noisy and incomplete data.
- Handling large graphs with many nodes and edges.
- Ensuring that the learned representations are physically meaningful.

• Present some recent advances in GNNs for High Energy Physics, such as:

- Graph Attention Networks (GATs): GNNs that use attention mechanisms to selectively focus on relevant nodes and edges.
- Message Passing Neural Networks (MPNNs): GNNs that use message passing to aggregate information from neighboring nodes and edges.
- Graph Convolutional Networks (GCNs): GNNs that apply convolutional operations on graphs to learn local patterns.

• Mention some ongoing research directions in GNNs for High Energy Physics, such as:

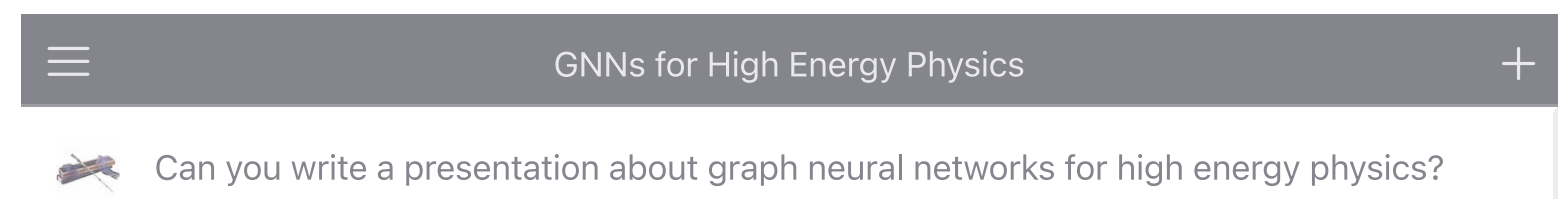
- Incorporating domain-specific knowledge into the GNN architecture.
- Combining GNNs with other machine learning techniques, such as reinforcement learning.
- Developing GNNs that can operate on heterogeneous graphs with multiple types of nodes and edges.

Conclusion:

- Summarize the main points of the presentation, highlighting the potential of GNNs for High Energy Physics.
- Emphasize the importance of continued research in this area, in order to address the challenges and fully exploit the capabilities of GNNs.
- Encourage the audience to explore GNNs further, either by reading related papers or by trying out existing GNN frameworks and libraries.

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

A few seconds later...



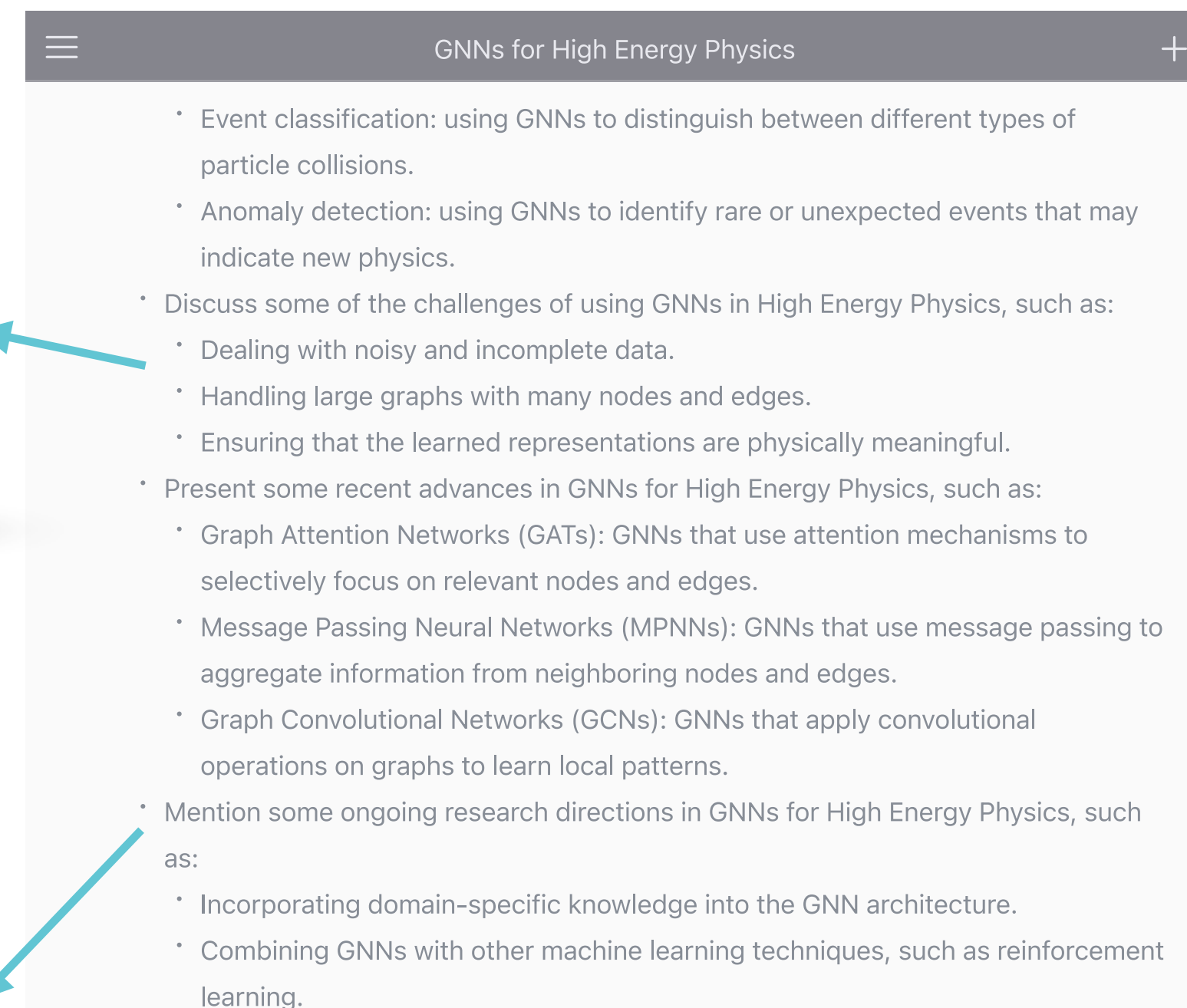
- Discuss some of the challenges of using GNNs in High Energy Physics, such as:
 - Dealing with noisy and incomplete data.
 - Handling large graphs with many nodes and edges.
 - Ensuring that the learned representations are physically meaningful.

- Explain what High Energy Physics is and why it is important (e.g., studying fundamental particles and forces, exploring the origins of the universe, developing new technologies).
- Mention that High Energy Physics generates large amounts of data, and that analyzing this data is a major challenge.
- Introduce the concept of Graph Neural Networks (GNNs) as a promising tool for analyzing data that has a graph structure, such as particle collisions.

Body:

- Explain what GNNs are and how they work, using a simple example (e.g., predicting node labels in a social network).
- Emphasize that GNNs can be used to learn from the relationships between nodes and edges from the features associated with its nodes and edges.
- Show how GNNs can be applied to High Energy Physics: a particle collision corresponds to a graph where nodes represent particles and edges represent their interactions.
- Describe some specific applications of GNNs in High Energy Physics:
 - Particle tracking: using GNNs to reconstruct the paths of particles from detector measurements.
 - Event classification: using GNNs to distinguish between different types of particle collisions.
 - Anomaly detection: using GNNs to identify rare or unexpected events that may indicate new physics.

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more useful and safer. Your feedback will help us improve.



- Mention some ongoing research directions in GNNs for High Energy Physics, such as:
 - Incorporating domain-specific knowledge into the GNN architecture.
 - Combining GNNs with other machine learning techniques, such as reinforcement learning.
 - Developing GNNs that can operate on heterogeneous graphs with multiple types of nodes and edges.



Sure, here is a possible presentation about Graph Neural Networks for High Energy Physics:

Title: Graph Neural Networks for High Energy Physics

Introduction:

- Explain what High Energy Physics is and why it is important (e.g., studying fundamental particles and forces, exploring the origins of the universe, developing new technologies).
- Mention that High Energy Physics generates large amounts of data, and that analyzing this data is a major challenge.
- Introduce the concept of Graph Neural Networks (GNNs) as a promising tool for analyzing data that has a graph structure, such as particle collisions.

Body:

- Explain what GNNs are and how they work, using a simple example (e.g., predicting node labels in a social network).
- Emphasize that GNNs can be used to learn from the topology of a graph, as well as from the features associated with its nodes and edges.
- Show how GNNs can be applied to High Energy Physics, where the graph corresponds to a particle collision and the nodes and edges represent particles and their interactions.
- Describe some specific applications of GNNs in High Energy Physics, such as:
 - Particle tracking: using GNNs to reconstruct the paths of particles from detector measurements.
 - Event classification: using GNNs to distinguish between different types of particle collisions.



- Anomaly detection: using GNNs to identify rare or unexpected events that may indicate new physics.
- Discuss some of the challenges of using GNNs in High Energy Physics, such as:
 - Dealing with noisy and incomplete data.
 - Handling large graphs with many nodes and edges.
 - Ensuring that the learned representations are physically meaningful.
- Present some recent advances in GNNs for High Energy Physics, such as:
 - Graph Attention Networks (GATs): GNNs that use attention mechanisms to selectively focus on relevant nodes and edges.
 - Message Passing Neural Networks (MPNNs): GNNs that use message passing to aggregate information from neighboring nodes and edges.
 - Graph Convolutional Networks (GCNs): GNNs that apply convolutional operations on graphs to learn local patterns.
- Mention some ongoing research directions in GNNs for High Energy Physics, such as:
 - Incorporating domain-specific knowledge into the GNN architecture.
 - Combining GNNs with other machine learning techniques, such as reinforcement learning.
 - Developing GNNs that can operate on heterogeneous graphs with multiple types of nodes and edges.

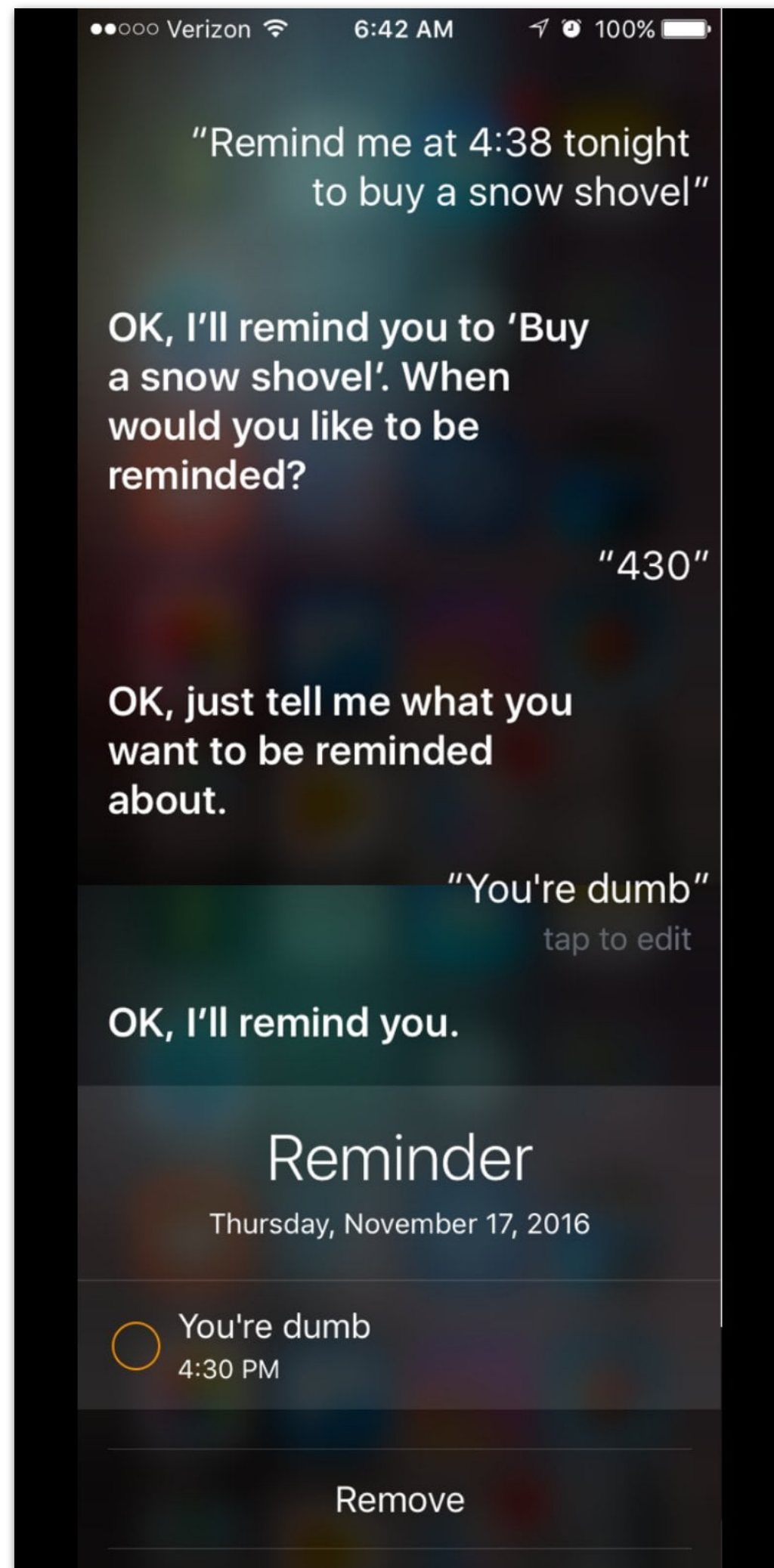
Conclusion:

- Summarize the main points of the presentation, highlighting the potential of GNNs for High Energy Physics.
- Emphasize the importance of continued research in this area, in order to address the challenges and fully exploit the capabilities of GNNs.
- Encourage the audience to explore GNNs further, either by reading related papers or by trying out existing GNN frameworks and libraries.

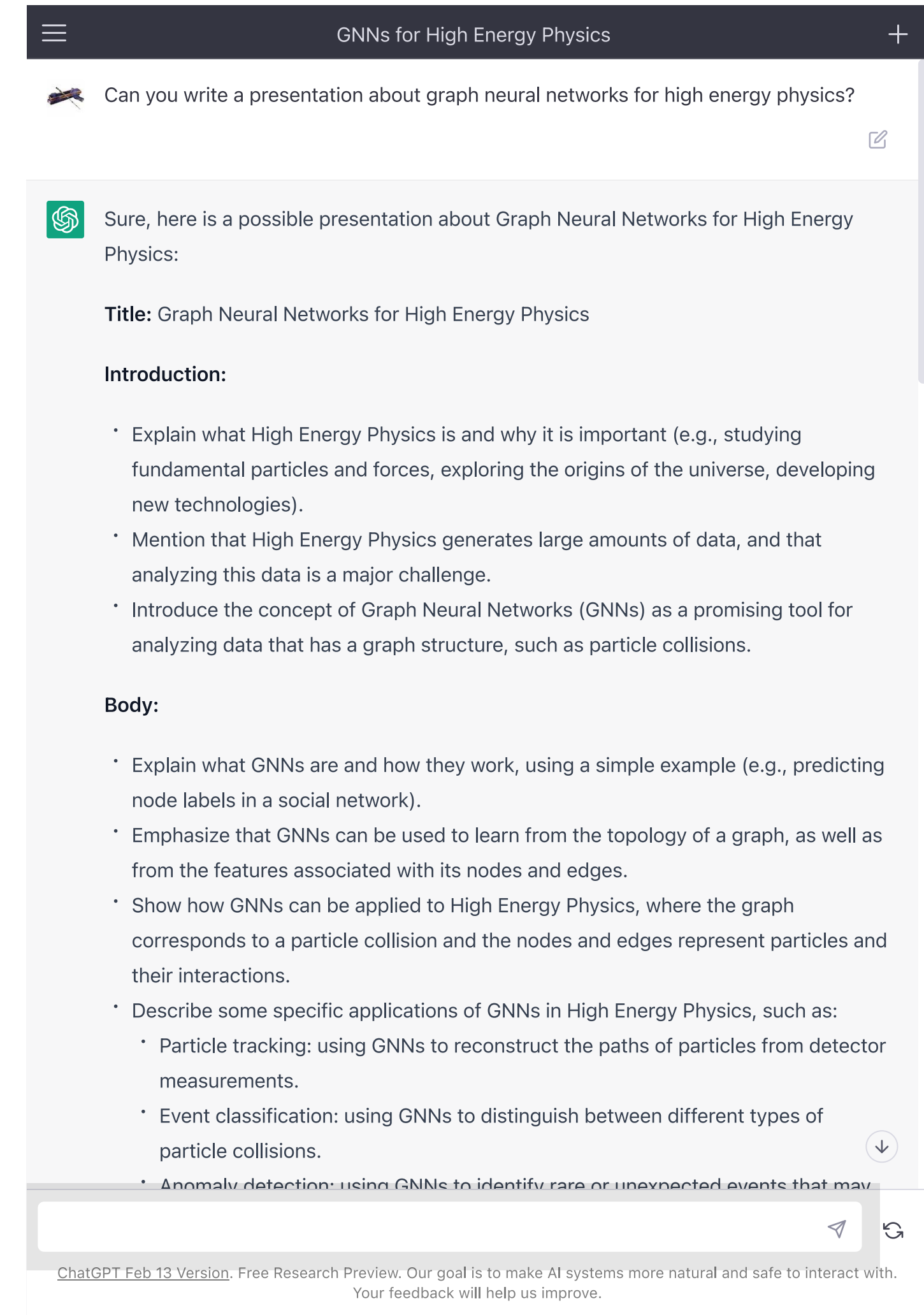


That's all for today.
Thank you!

THE EVOLUTION OF ARTIFICIAL INTELLIGENCE

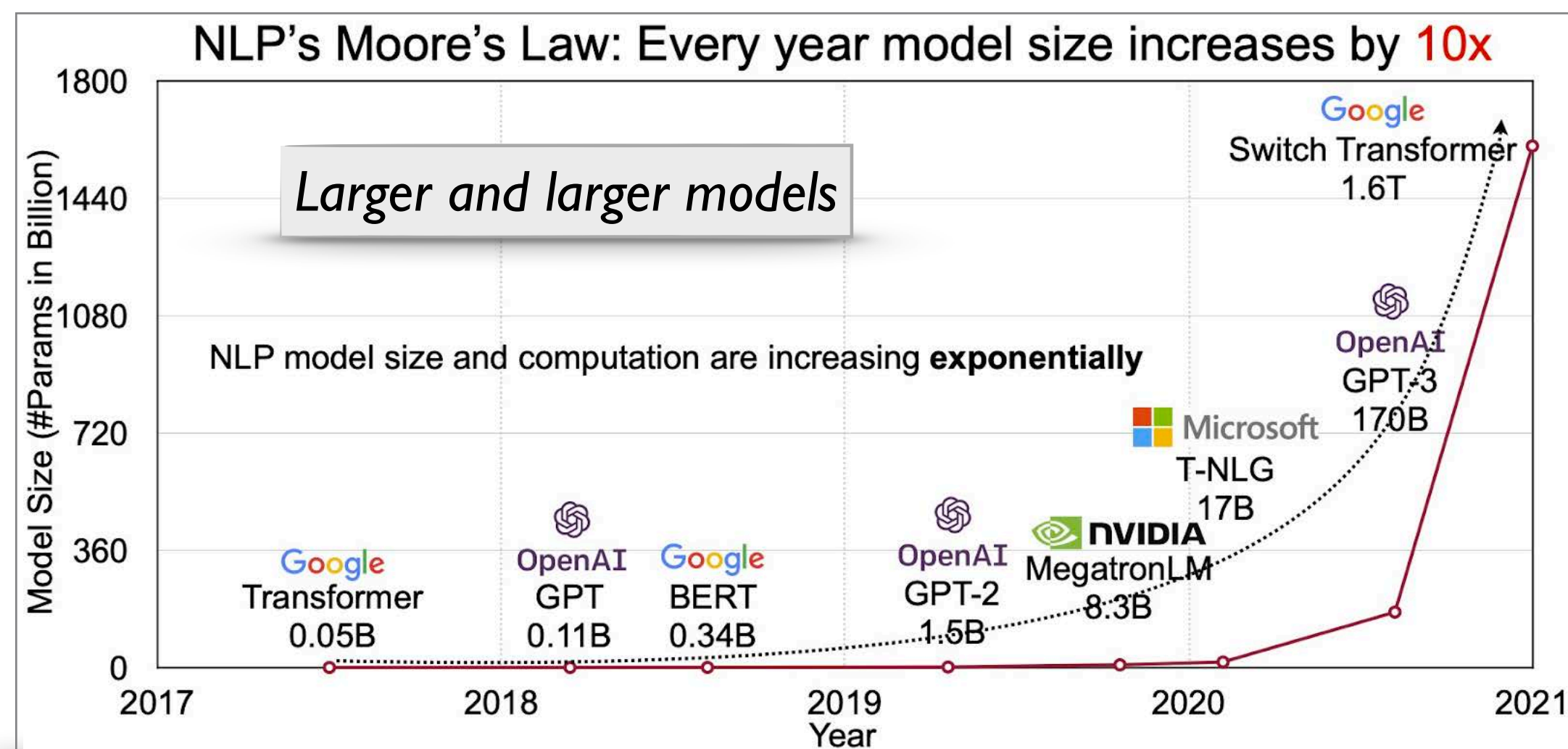
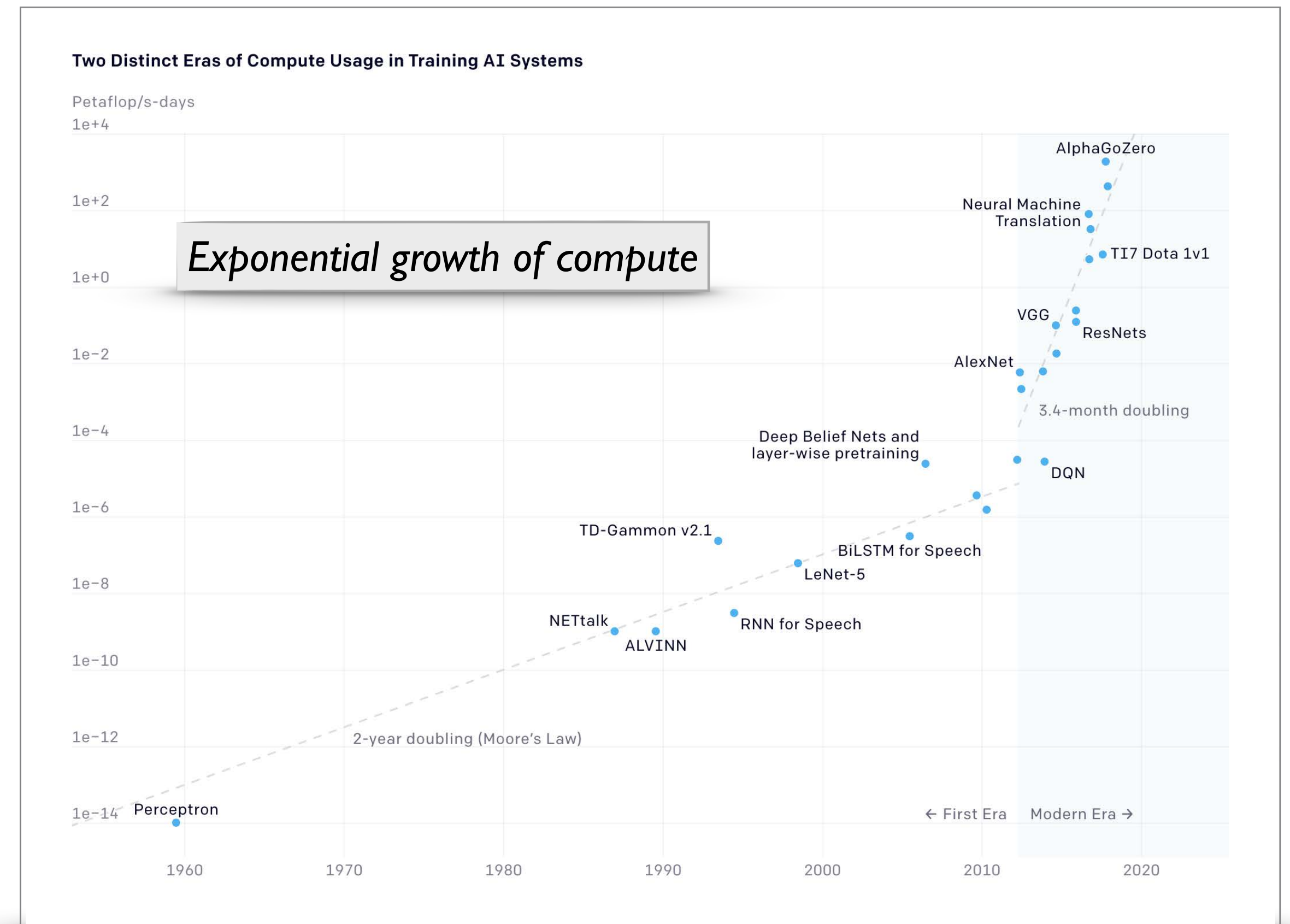
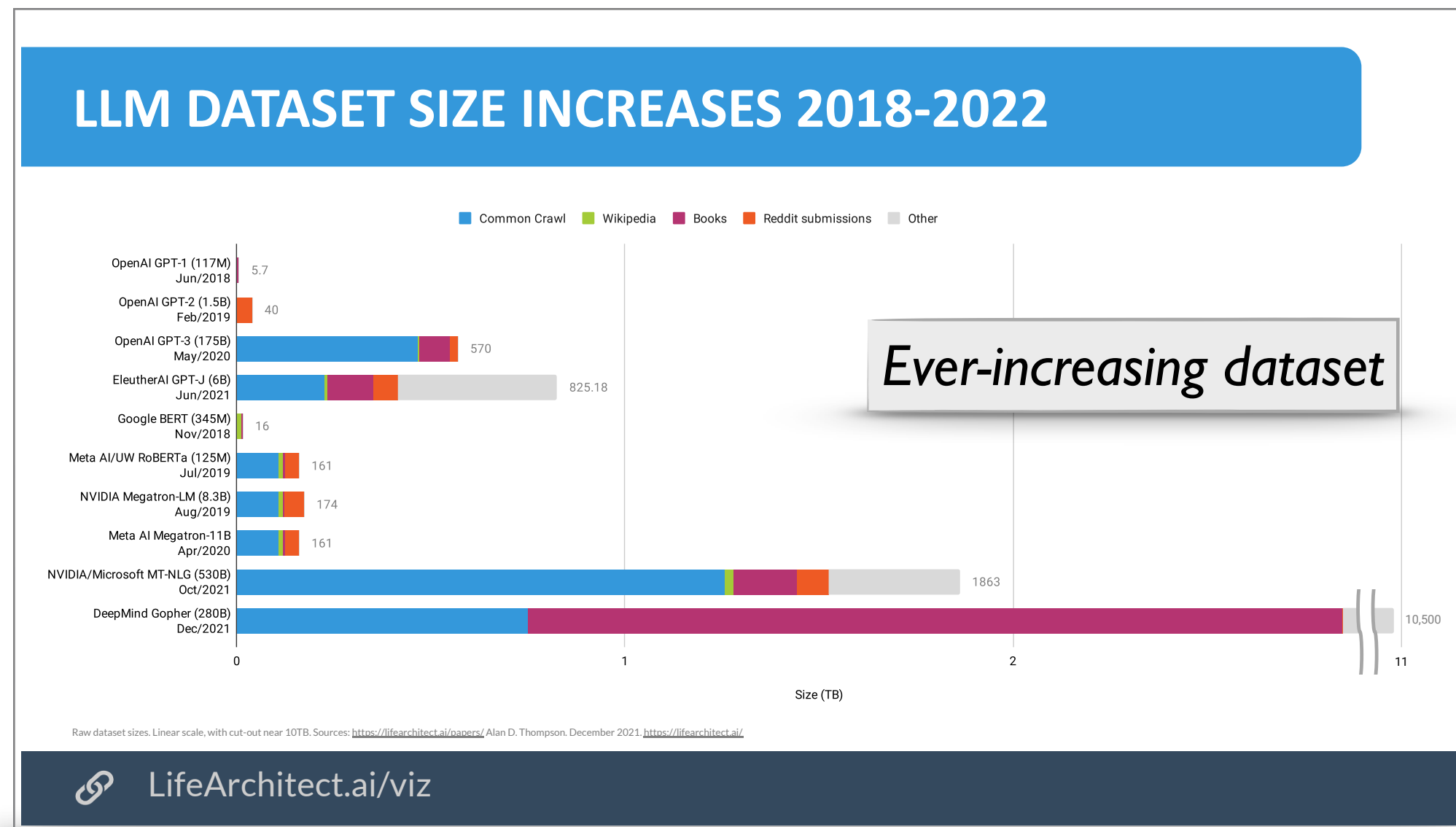


Siri (Apple, 2011)



ChatGPT (OpenAI, 2022)

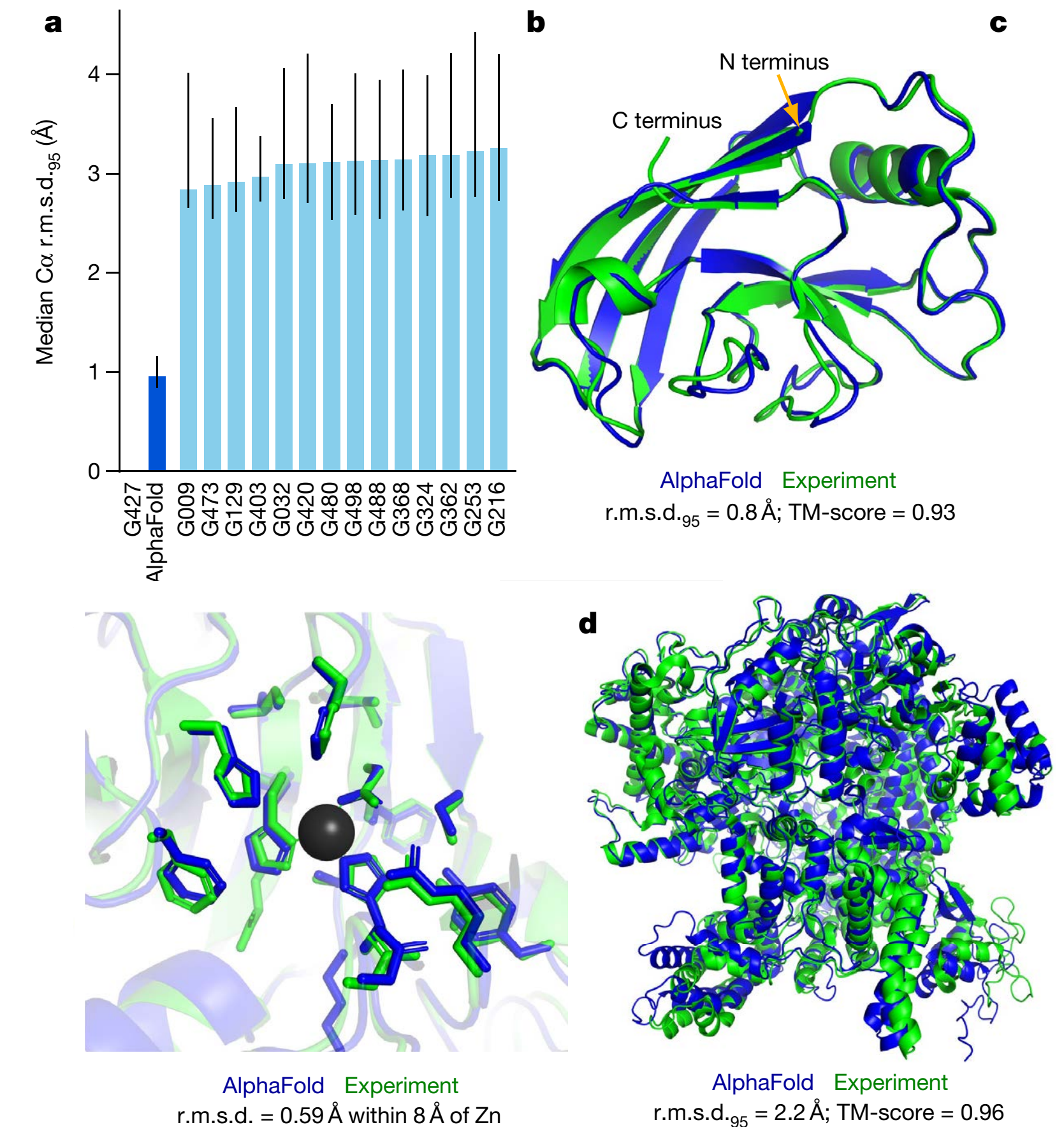
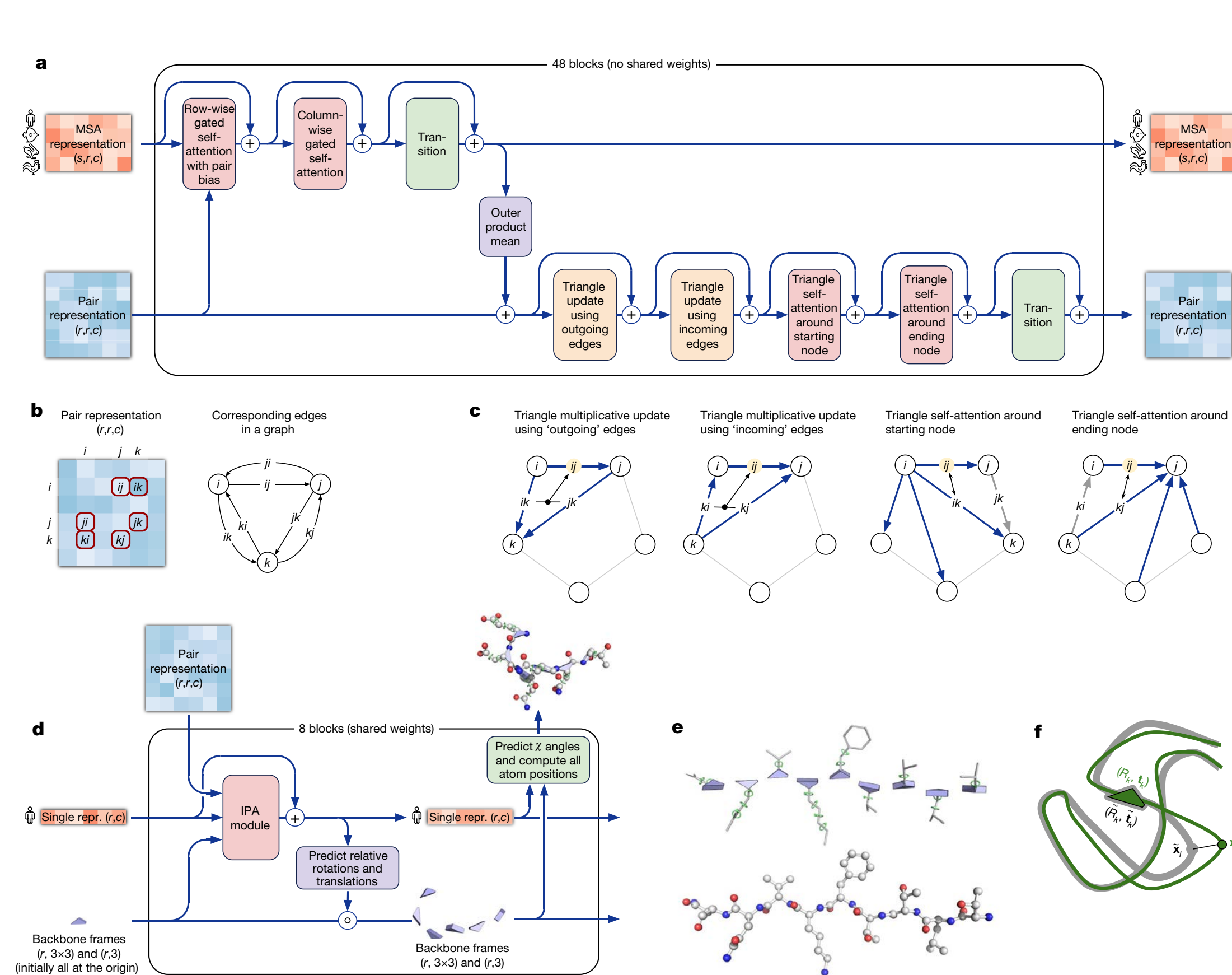
THE EVOLUTION OF ARTIFICIAL INTELLIGENCE



THE EVOLUTION OF AI FOR SCIENCE

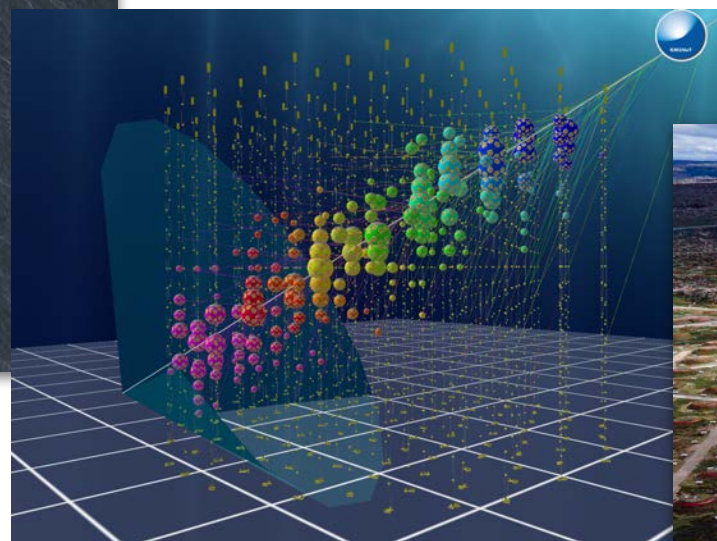
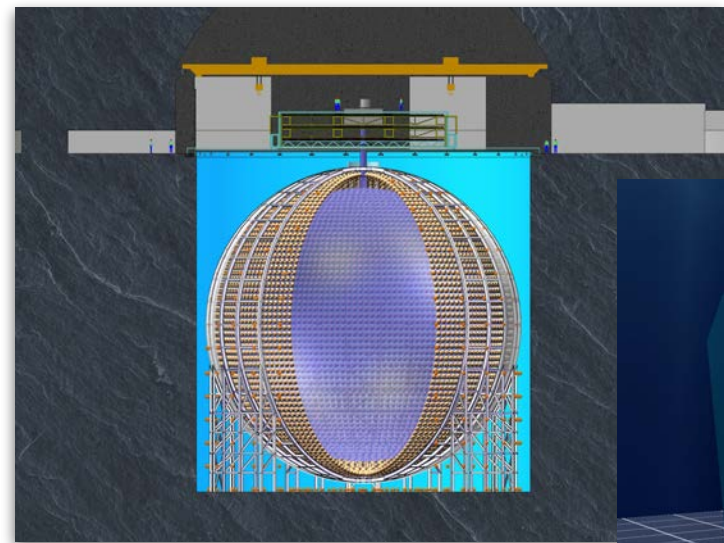
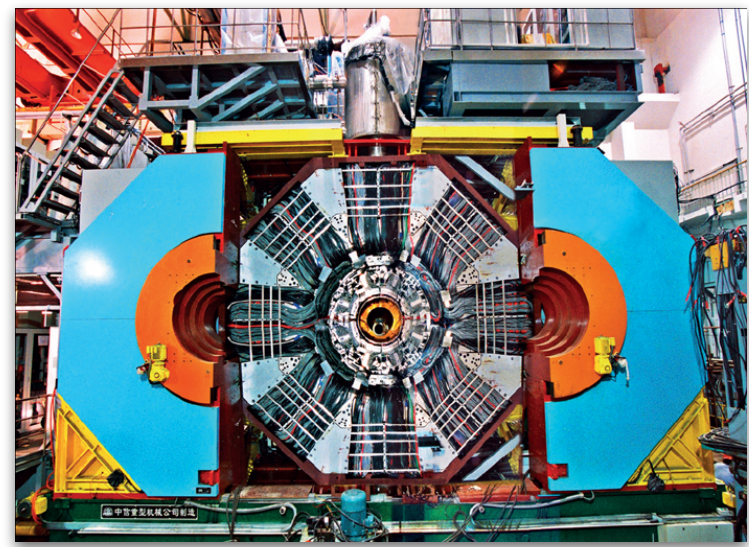
- AlphaFold2: reaching atomic accuracy on protein structure prediction
- based on Transformer models – the ML technology behind ChatGPT

*J. Jumper, R. Evans, A. Pritzel et al.
[Nature 596, 583–589 (2021)]*



THE DATA CHALLENGE IN HIGH ENERGY PHYSICS

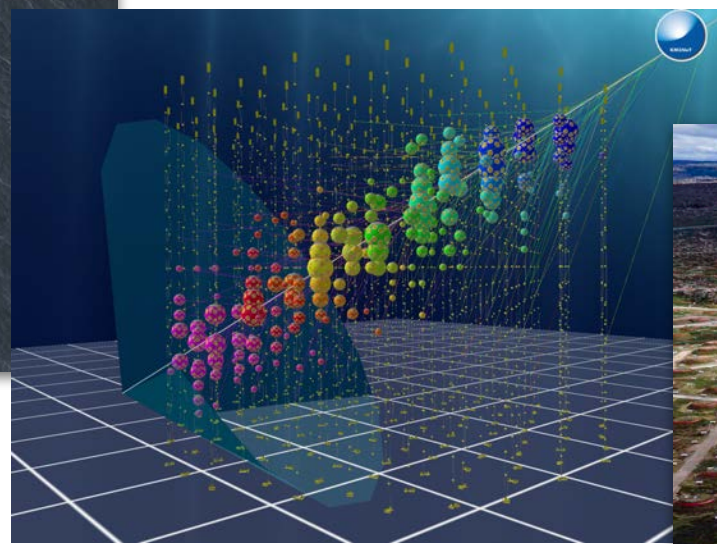
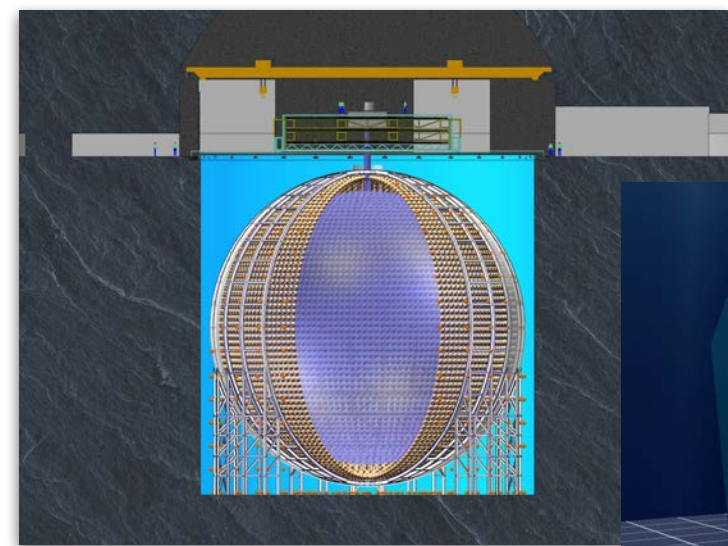
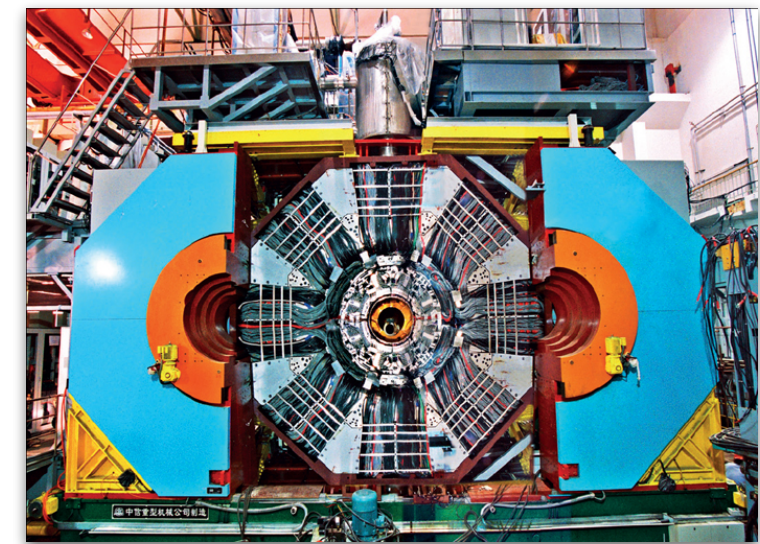
HEP



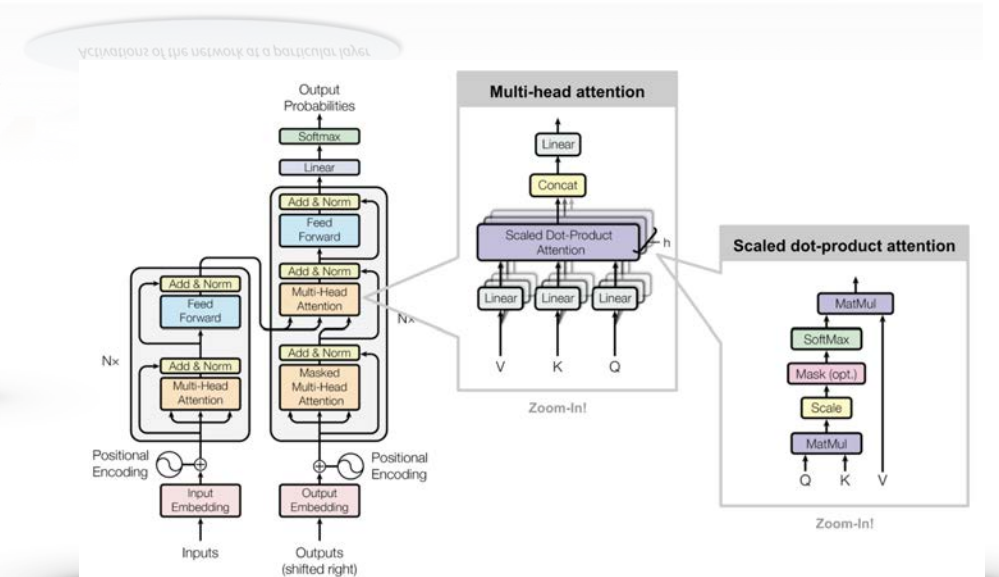
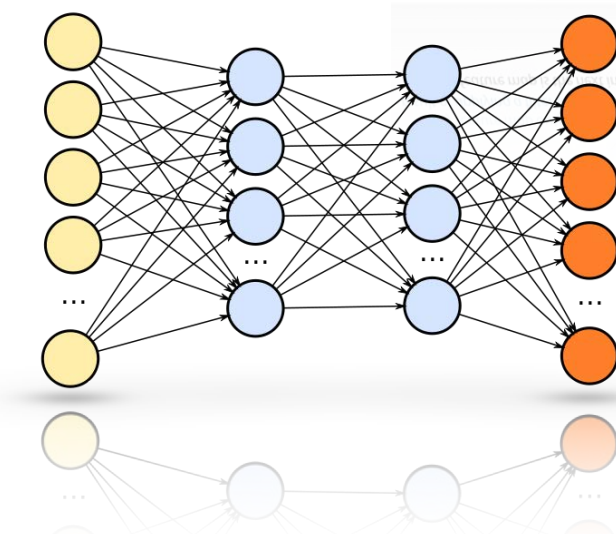
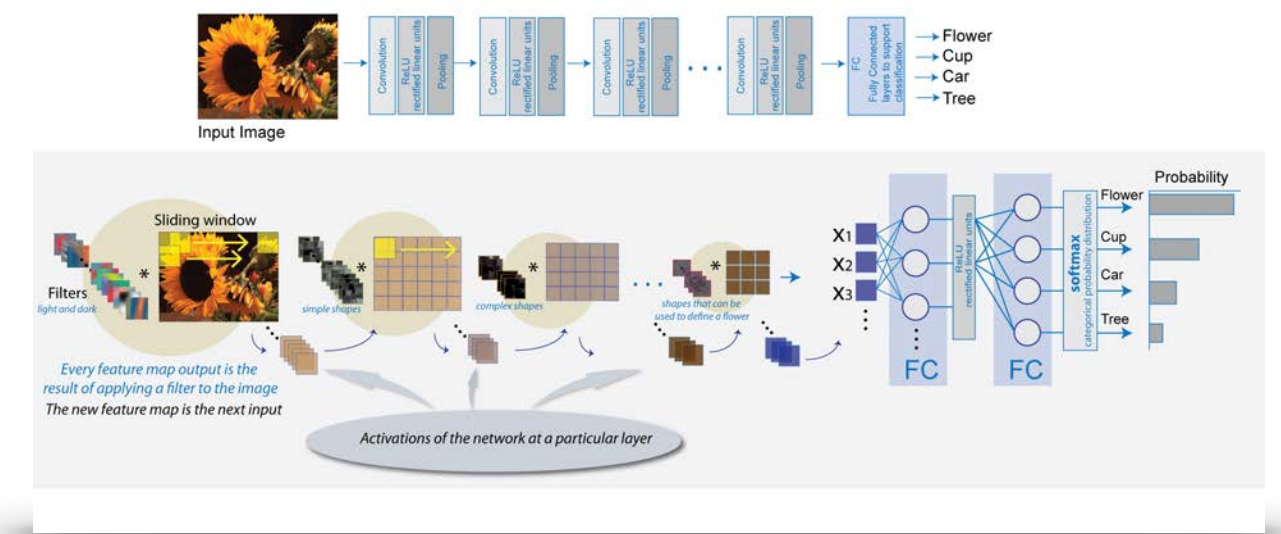
Large volume of data, complex topology, ...

AI + HEP: AT THE COLLISION POINT

HEP



Large volume of data, complex topology, ...



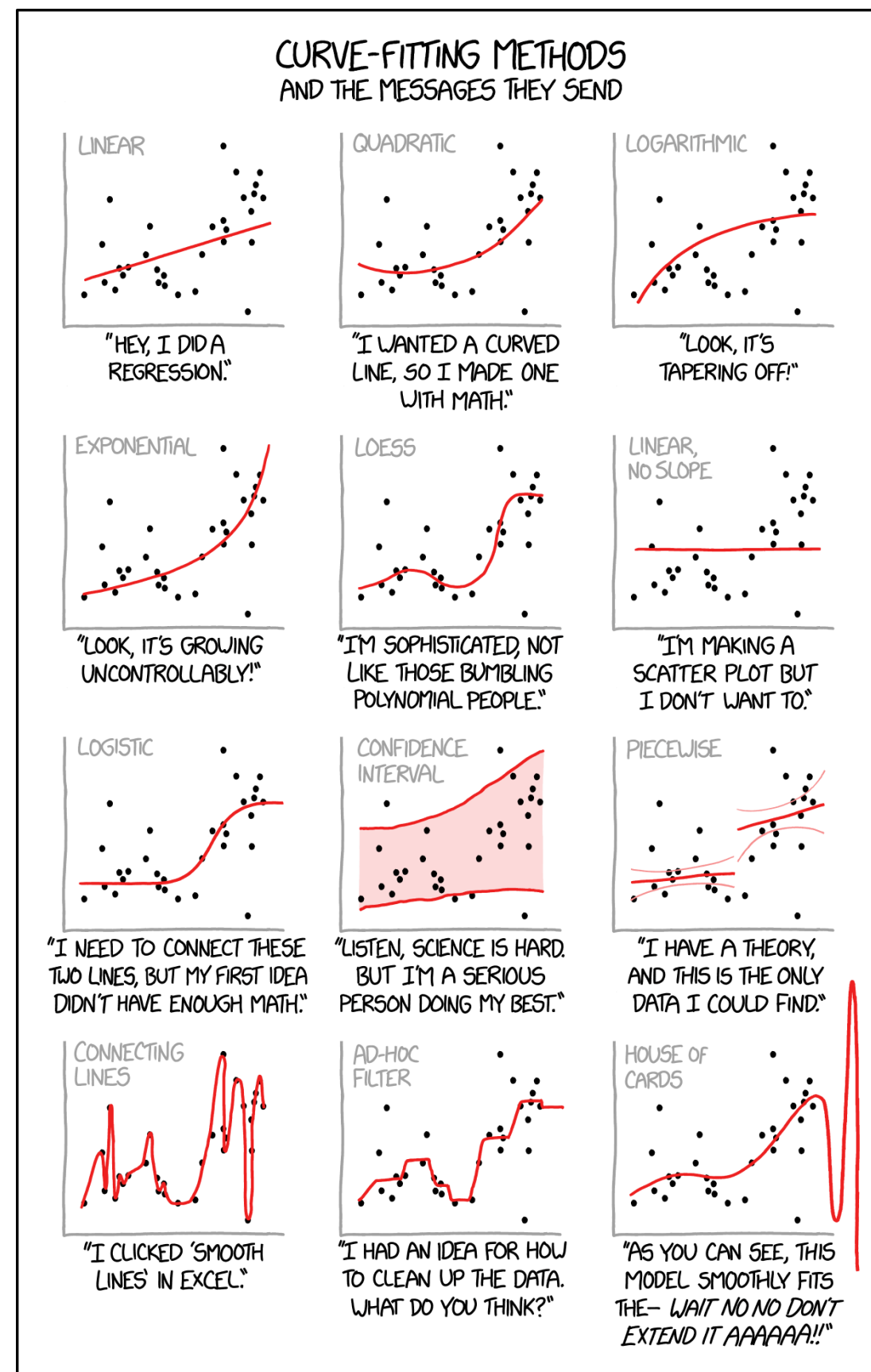
AI

Collimate HEP and AI to make them collide!

A JOURNEY THROUGH GRAPH NEURAL NETWORKS

MACHINE LEARNING 101

- “Machine learning is just curve fitting”
 - but...



Input data: $\{x_i\}$

Model: $f_{\theta}(x)$

Optimization algorithm

Curve fitting

Low dimensional (1D/2D)
Few points $O(10-100)$

Simple functions with few parameters

Minimization of chi square / likelihood / etc.

Machine learning

(esp. Deep Neural Networks)

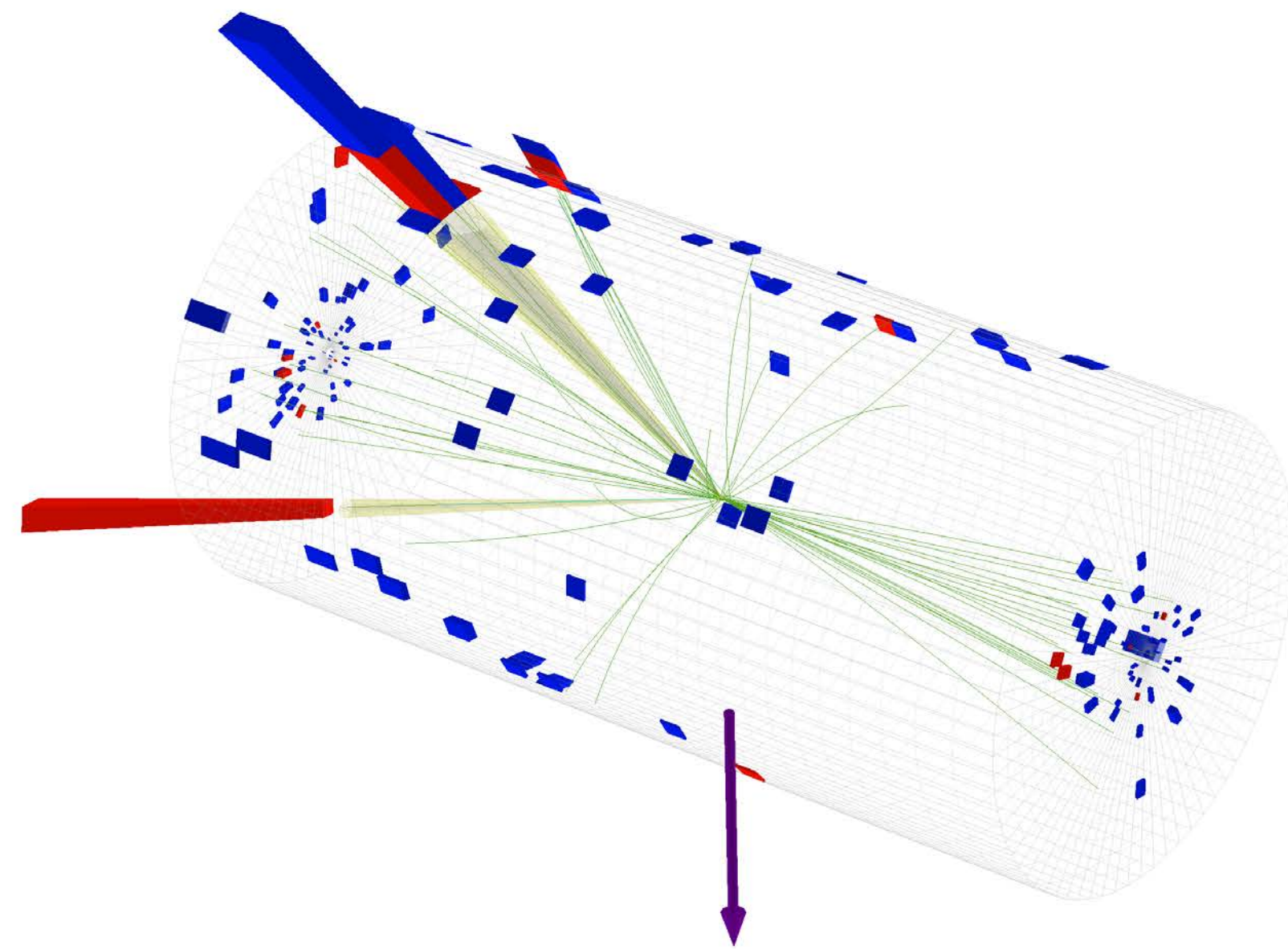
High dimensional ($\gg 100$)
Huge data sample ($\gg 1M$)

Complex functions with highly special structures (CNNs, RNNs, Transformers, etc.) and a large number of parameters (10^3 to 10^9)

Minimization of task-specific loss functions (e.g., cross entropy for classification, MSE/MAE for regression, etc.)

DATA REPRESENTATION

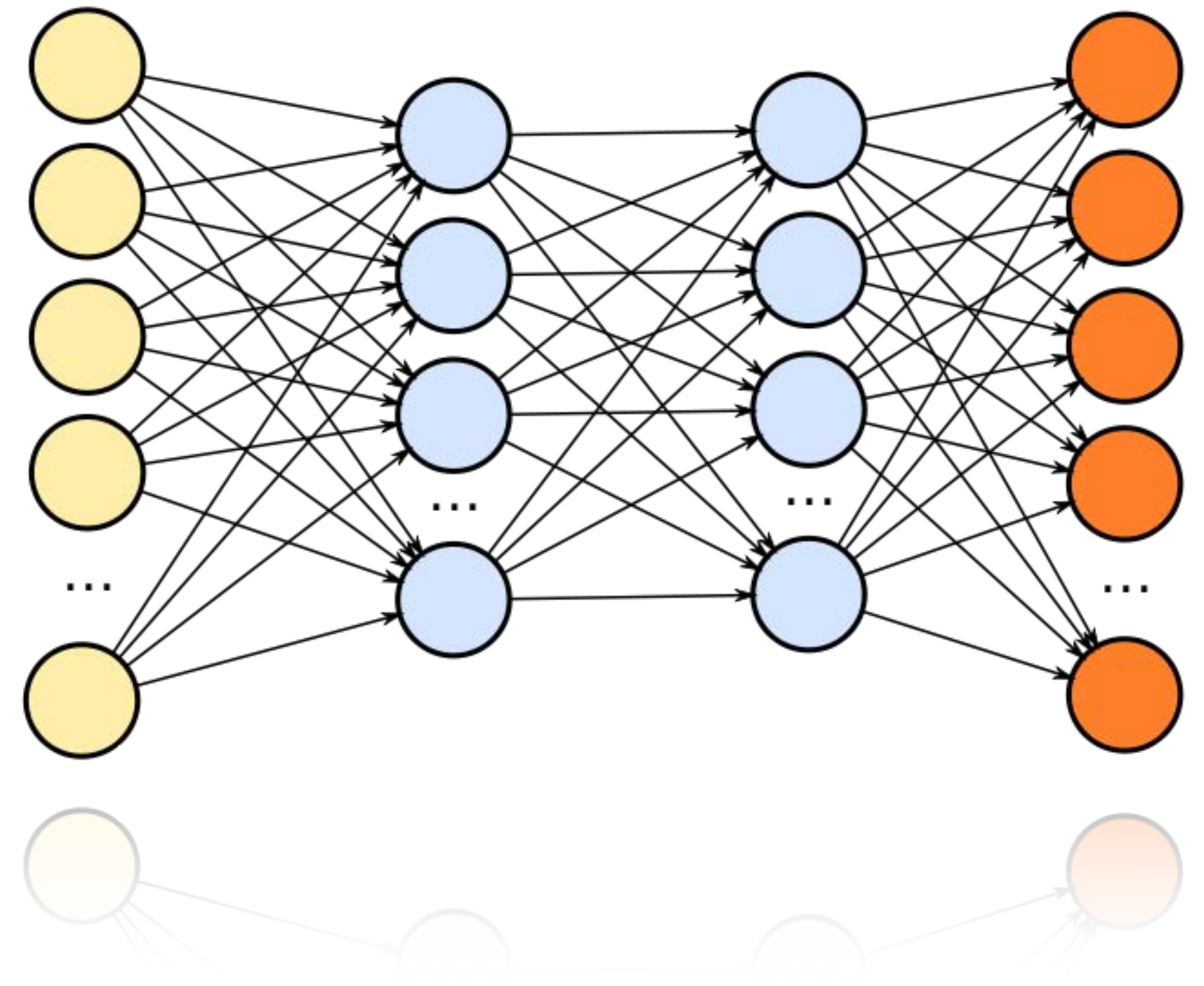
HEP



Collision events, detector hits, sensor arrays, ...

×

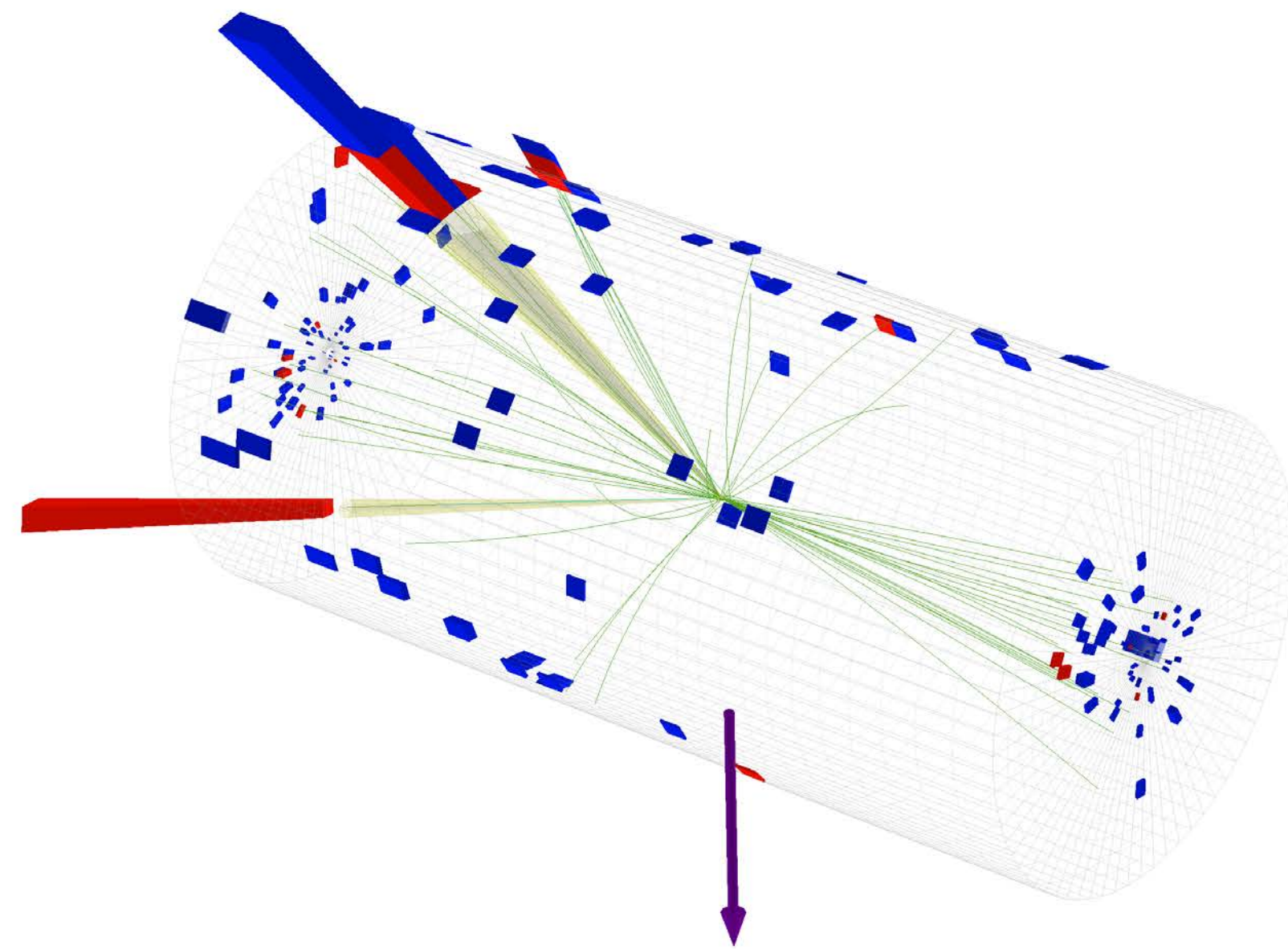
ML



First and foremost:
How to represent the data?

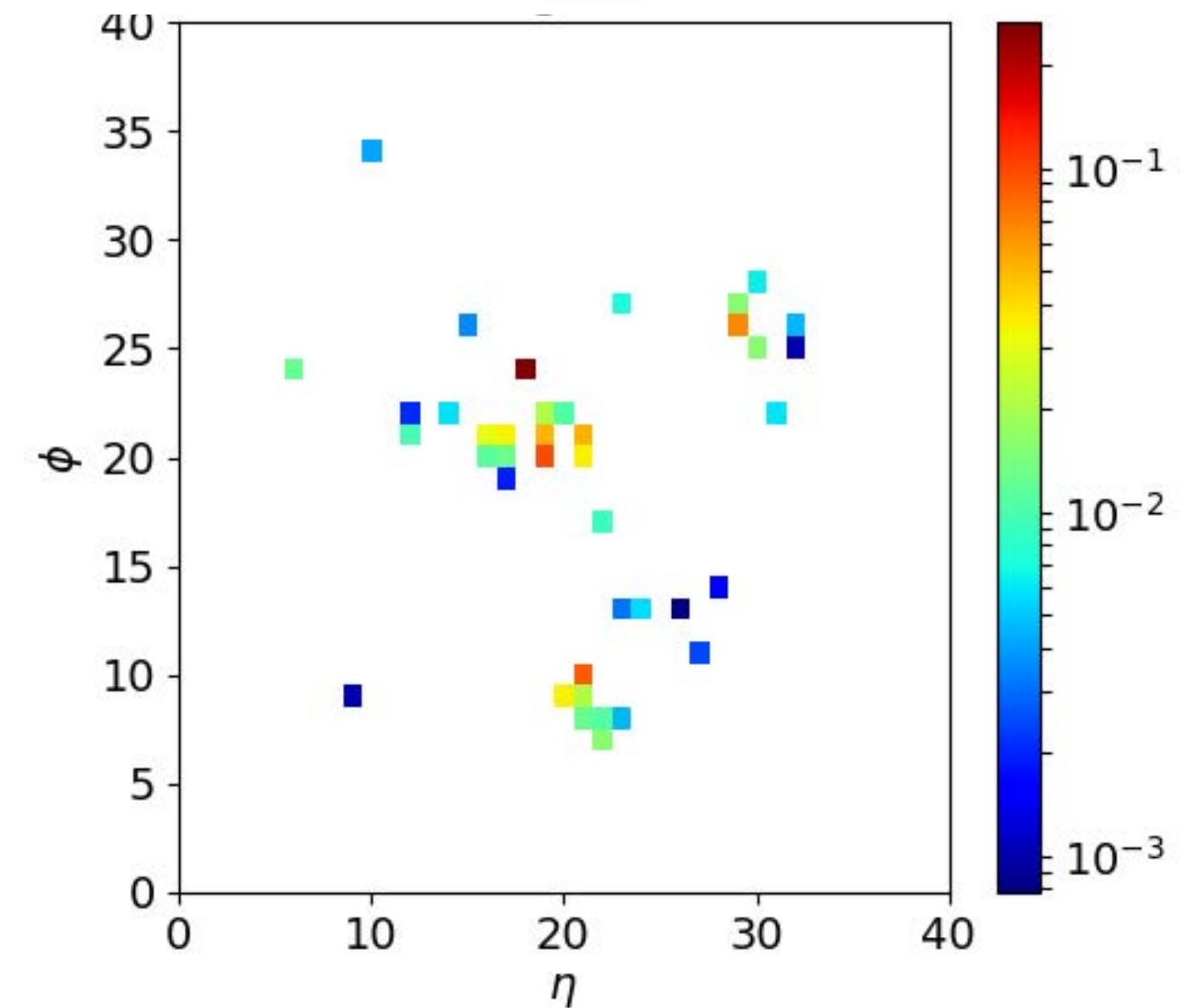
DATA REPRESENTATION: IMAGE

HEP



Collision events, detector hits, sensor arrays, ...

Image

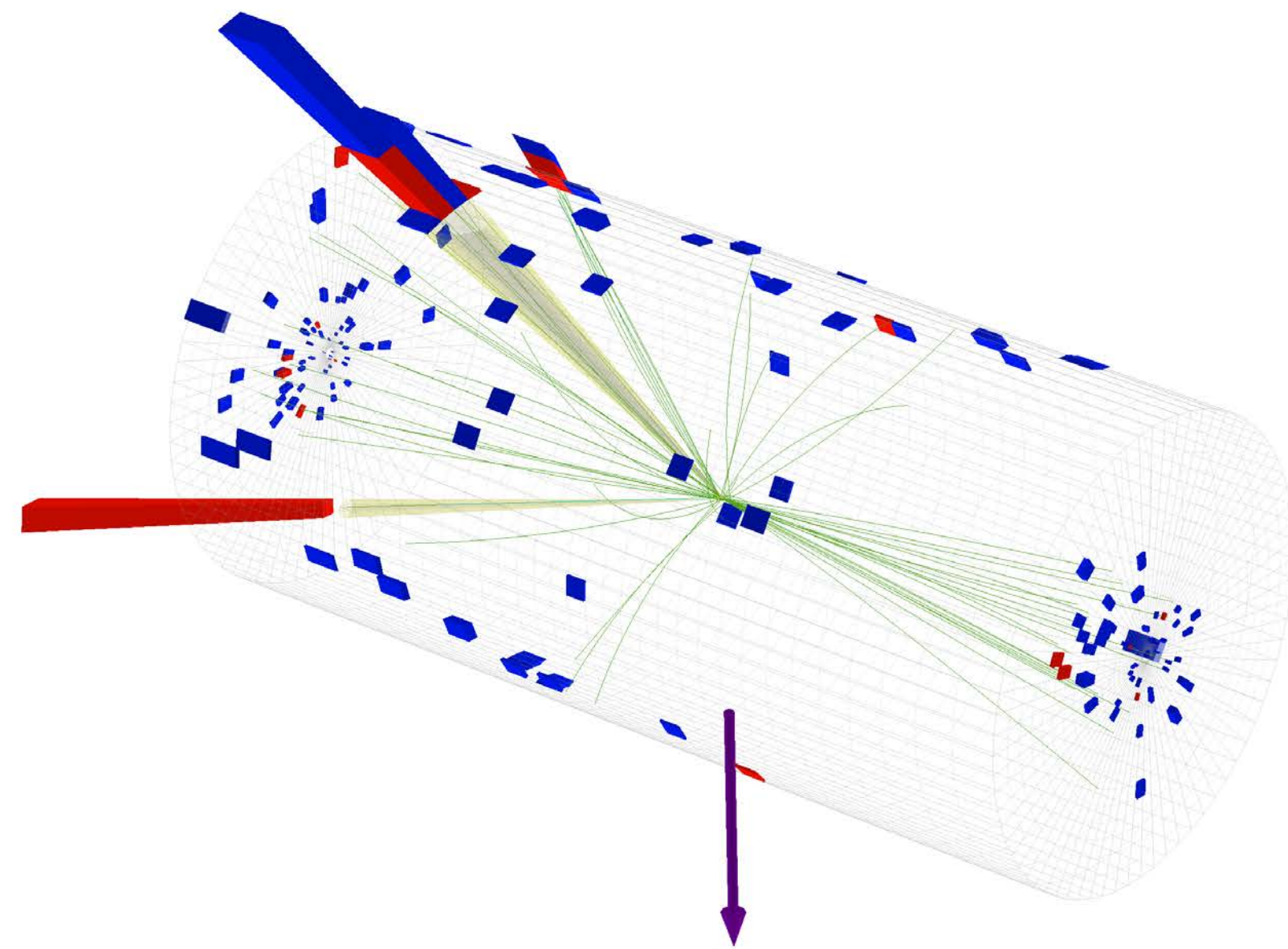


e.g., review in Kagan, arXiv:2012.09719

- Convert to 2D/3D image => **Computer vision**
 - then use convolutional neural networks (CNNs)
 - but:
 - inhomogeneous geometry, high sparsity, ...

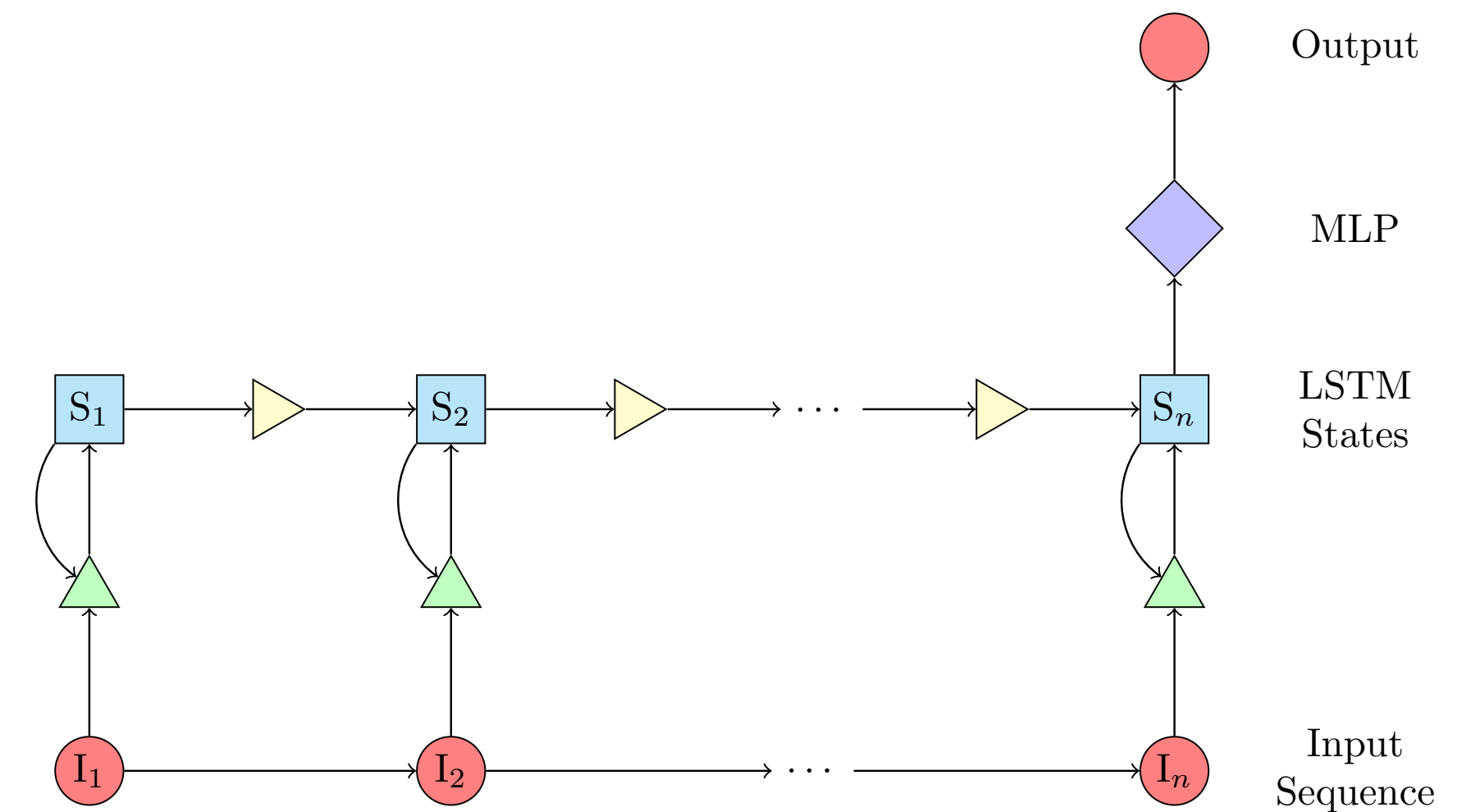
DATA REPRESENTATION: SEQUENCE

HEP



Collision events, detector hits, sensor arrays, ...

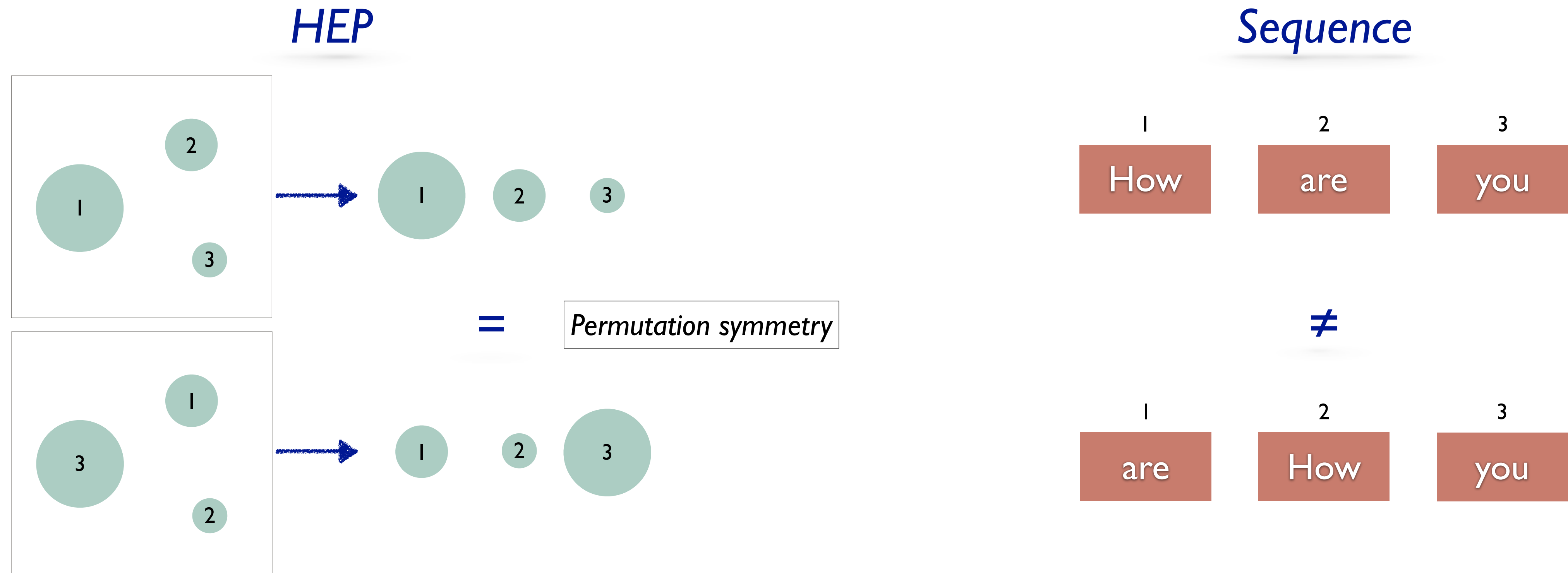
Sequence



e.g., Guest, Collado, Baldi, Hsu, Urban, Whiteson
arXiv: 1607.08633

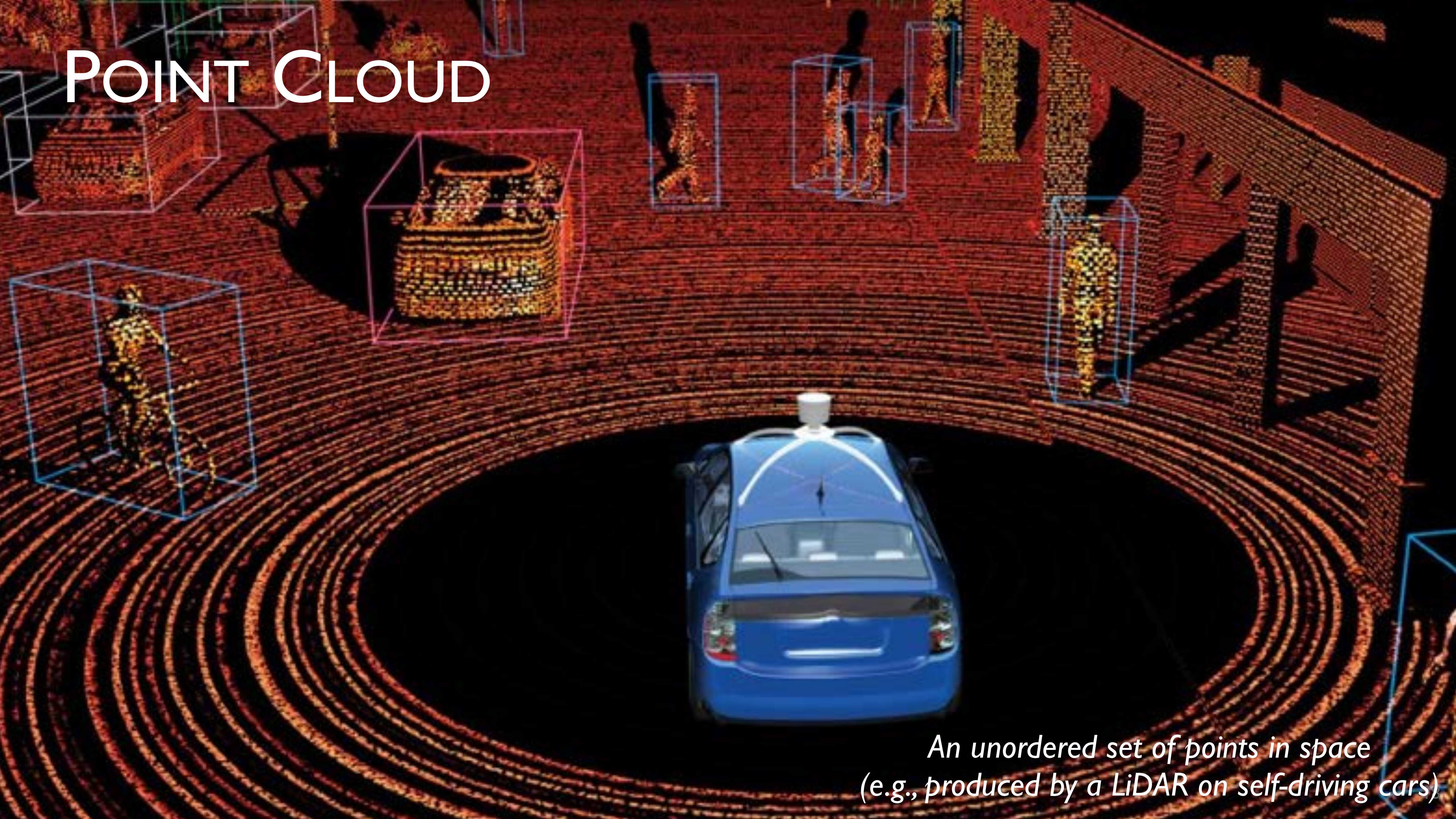
- Convert to a sequence => **Natural language processing (NLP)**
 - recurrent neural network (RNN), e.g., GRU/LSTM; 1D CNNs; etc.

DATA REPRESENTATION: SEQUENCE?



- Convert to a sequence => **Natural language processing (NLP)**
 - recurrent neural network (RNN), e.g., GRU/LSTM; 1D CNNs; etc.
 - but:
 - must impose an **ordering** on the particles/hits, which can limit the learning performance

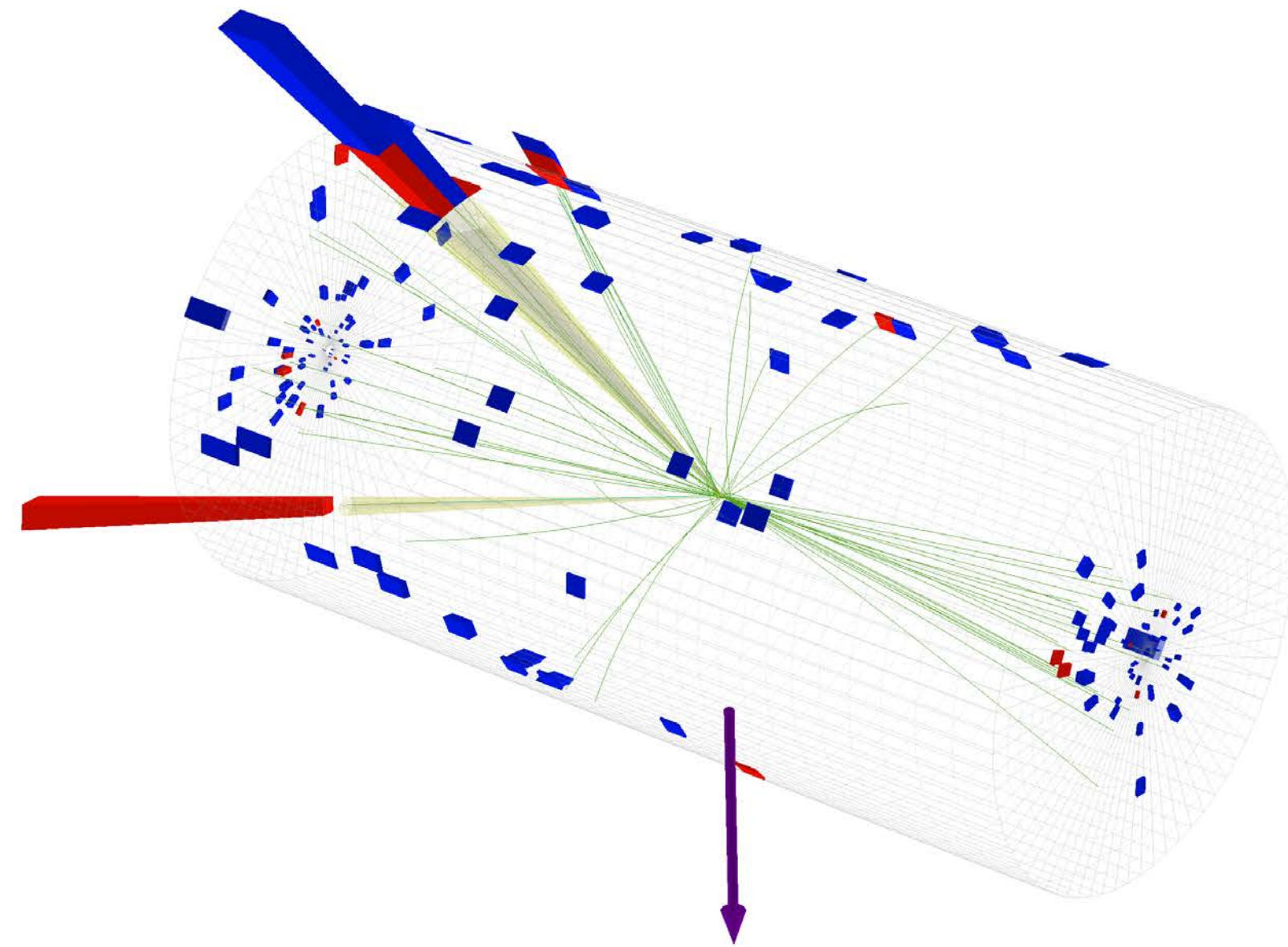
POINT CLOUD



*An unordered set of points in space
(e.g., produced by a LiDAR on self-driving cars)*

DATA REPRESENTATION: POINT CLOUD

HEP



Collision events, detector hits, sensor arrays, ...

Point cloud



- HEP data as a point cloud

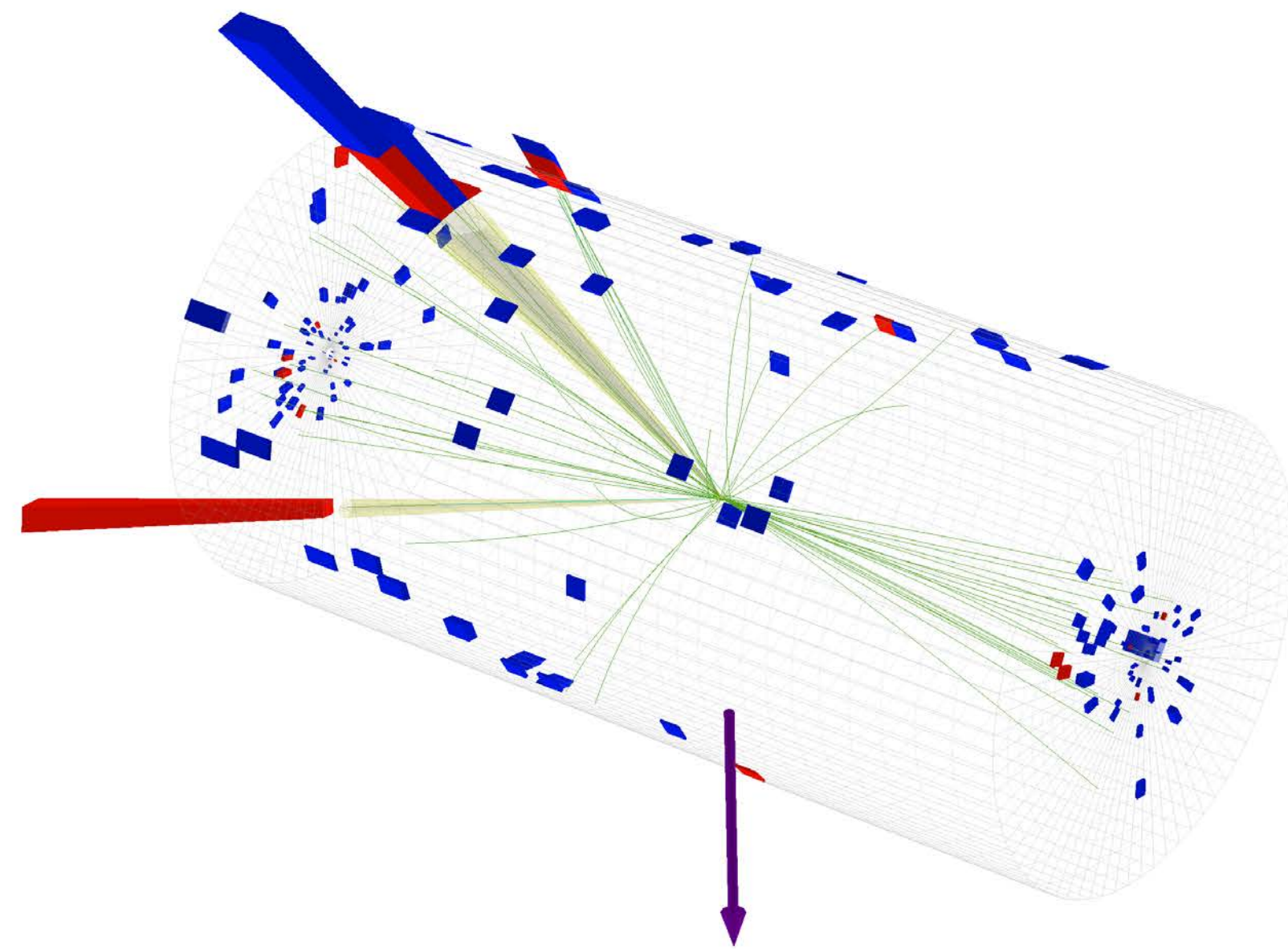
- each particle / detector cell is a point in the cloud

- for each point: (spatial) coordinates + any additional properties (energy/momentum, detector response, ...)

- key feature: ***permutation symmetry***

LEARNING ON POINT CLOUDS

HEP



Collision events, detector hits, sensor arrays, ...



Point cloud



ϕ

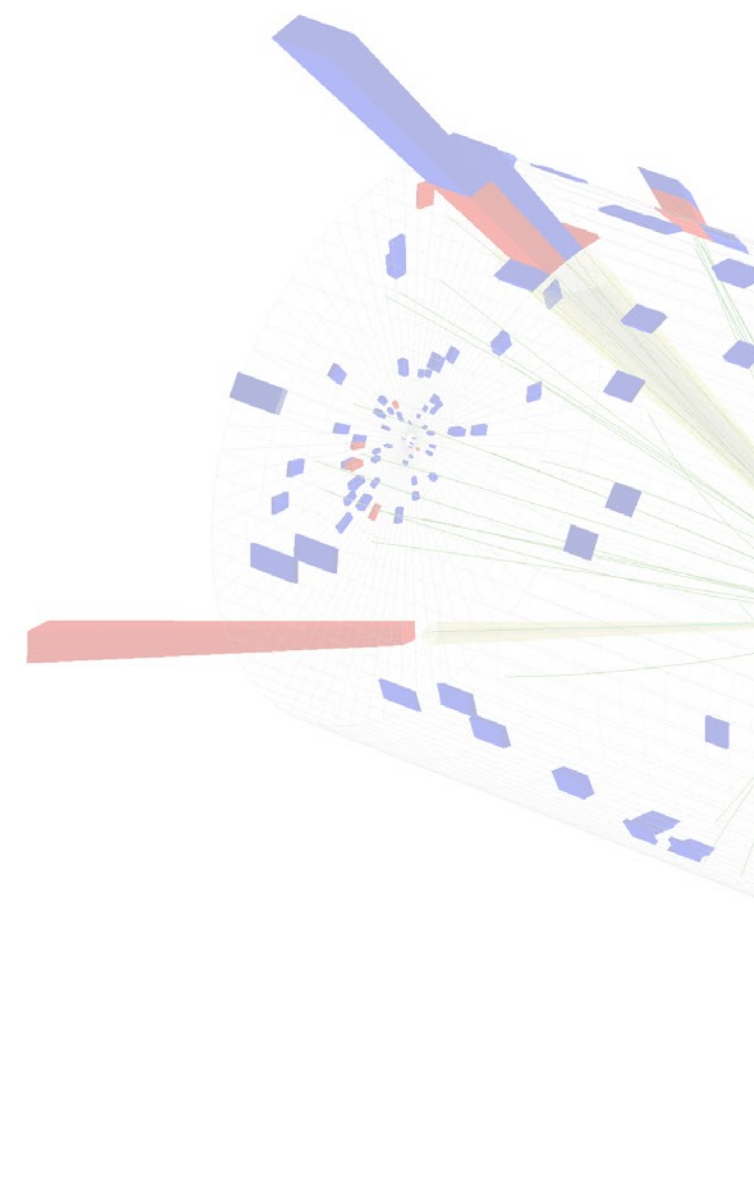
η

- Desired algorithms for learning on point cloud data
 - symmetry-preserving: the outputs should be invariant under permutation of the points
 - high expressiveness: capable of fully exploiting the correlations between points
 - low computational cost: scalable from $O(10)$ to $O(1000)$ points, and even up to $O(1M)$ points in some cases

LEARNING ON POINT CLOUDS

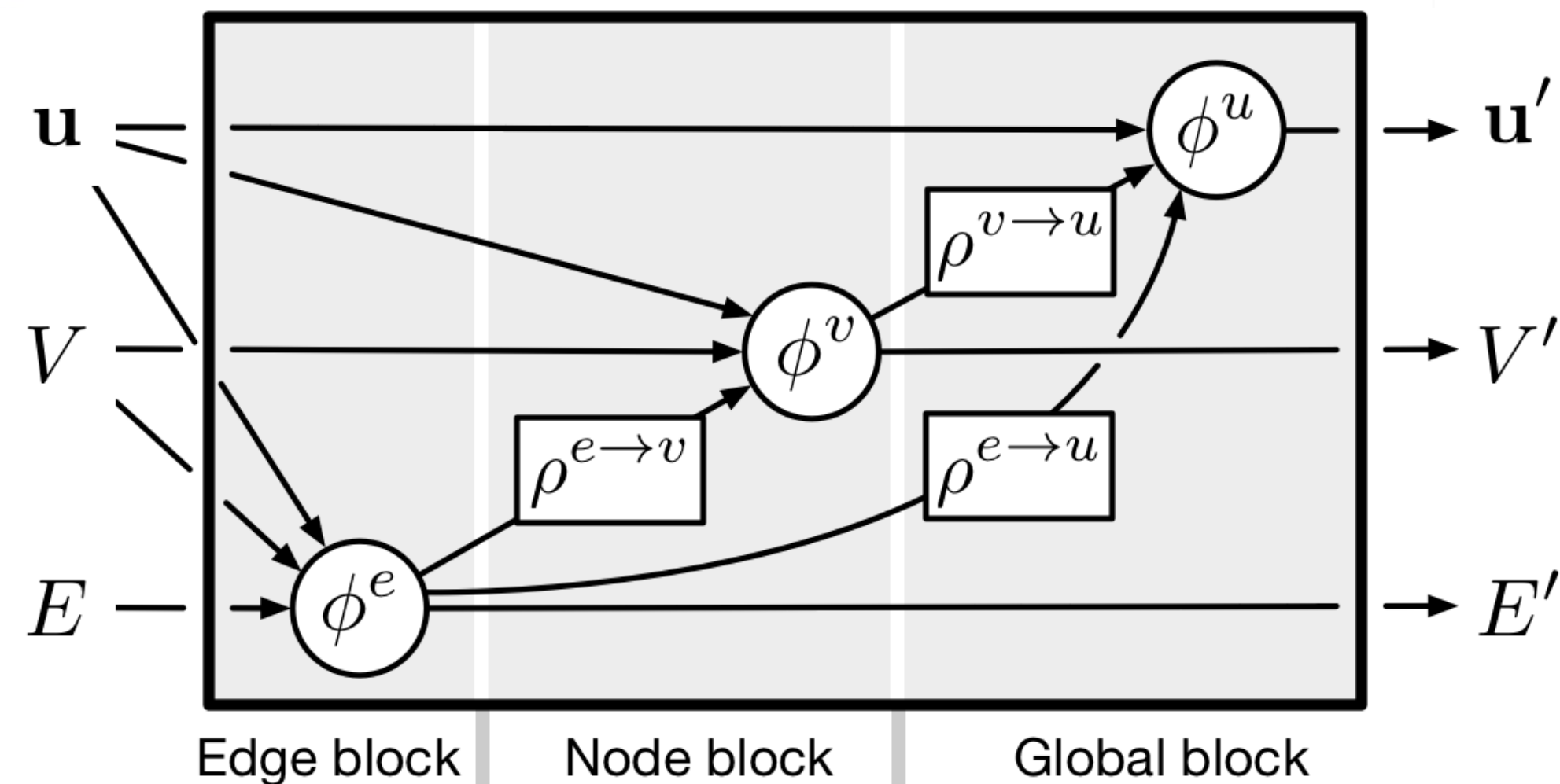
HEP

Point cloud



Collision events, detector hit

Graph neural network - A unified framework



η

- Desired algorithms for learning on point cloud data Review in Shlomi, Battaglia, Vlimant, arXiv:2007.13681
 - symmetry-preserving: the outputs should be invariant under permutation of the points
 - high expressiveness: capable of fully exploiting the correlations between points
 - low computational cost: scalable from $O(10)$ to $O(1000)$ points, and even up to $O(1M)$ points in some cases

WHAT IS A GRAPH?

Graph level attributes
Graph: $G = (\mathbf{u}, V, E)$ with N_v vertices and N_e edges

Vertices (nodes)

$$V = \{\mathbf{v}_i\}_{i=1:N_v}$$

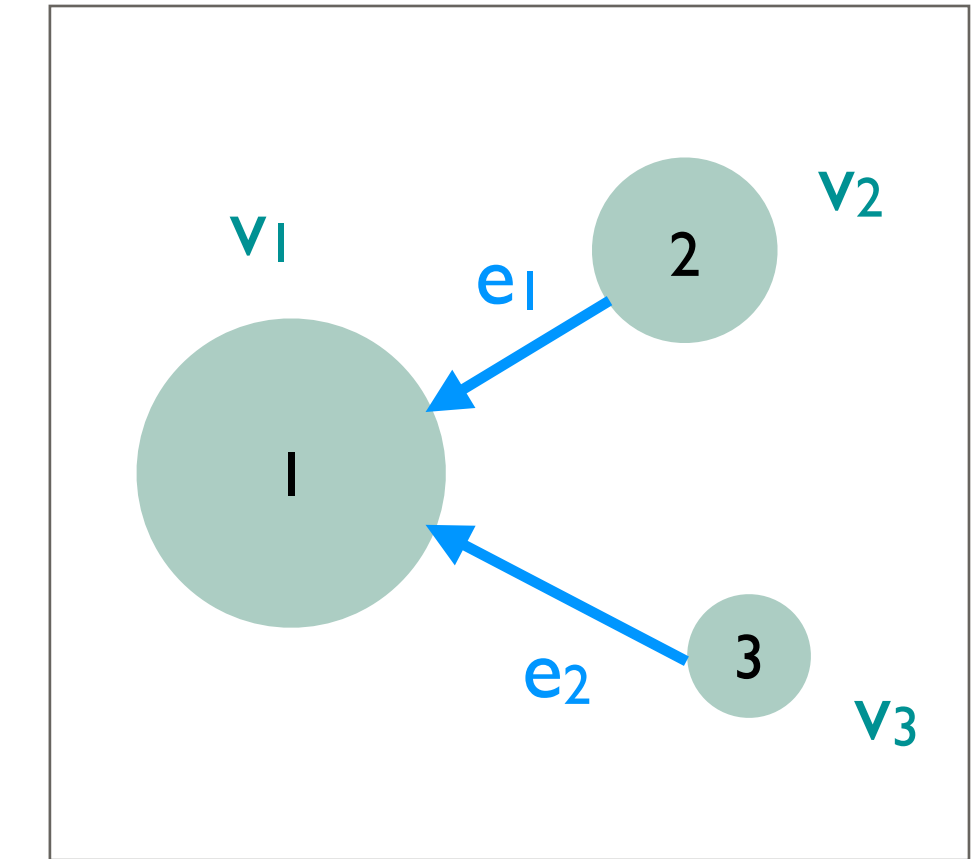
attributes of the i-th node

Edges (links)

$$E = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N_e}$$

attributes of the k-th edge

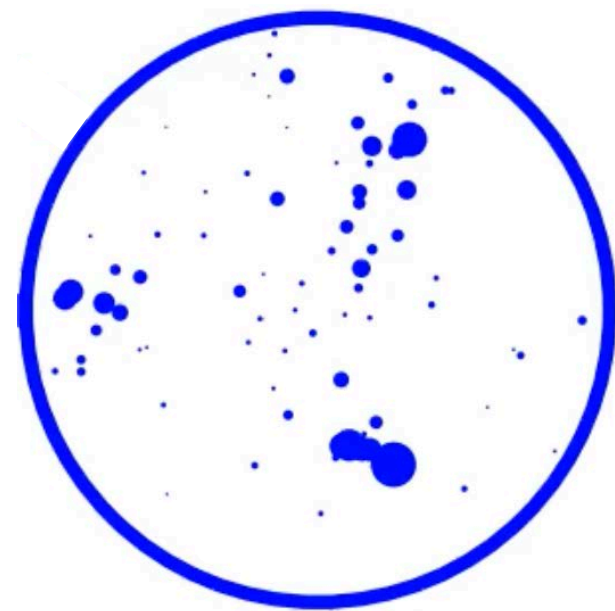
*indices of the two nodes (receiver and sender)
connected by the k-th edge*



HOW TO BUILD THE GRAPH?

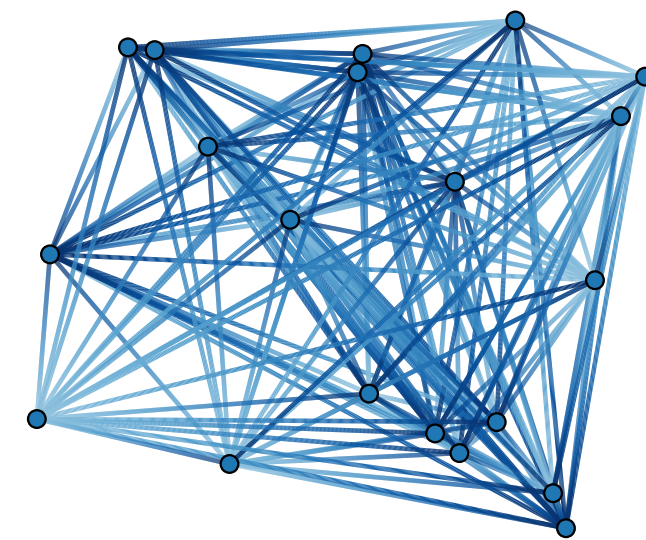
- From point clouds to graphs:
 - points (particles/hits/sensors) naturally become the **nodes** of the graph
 - but how to define the **edges**?

Set: no edges



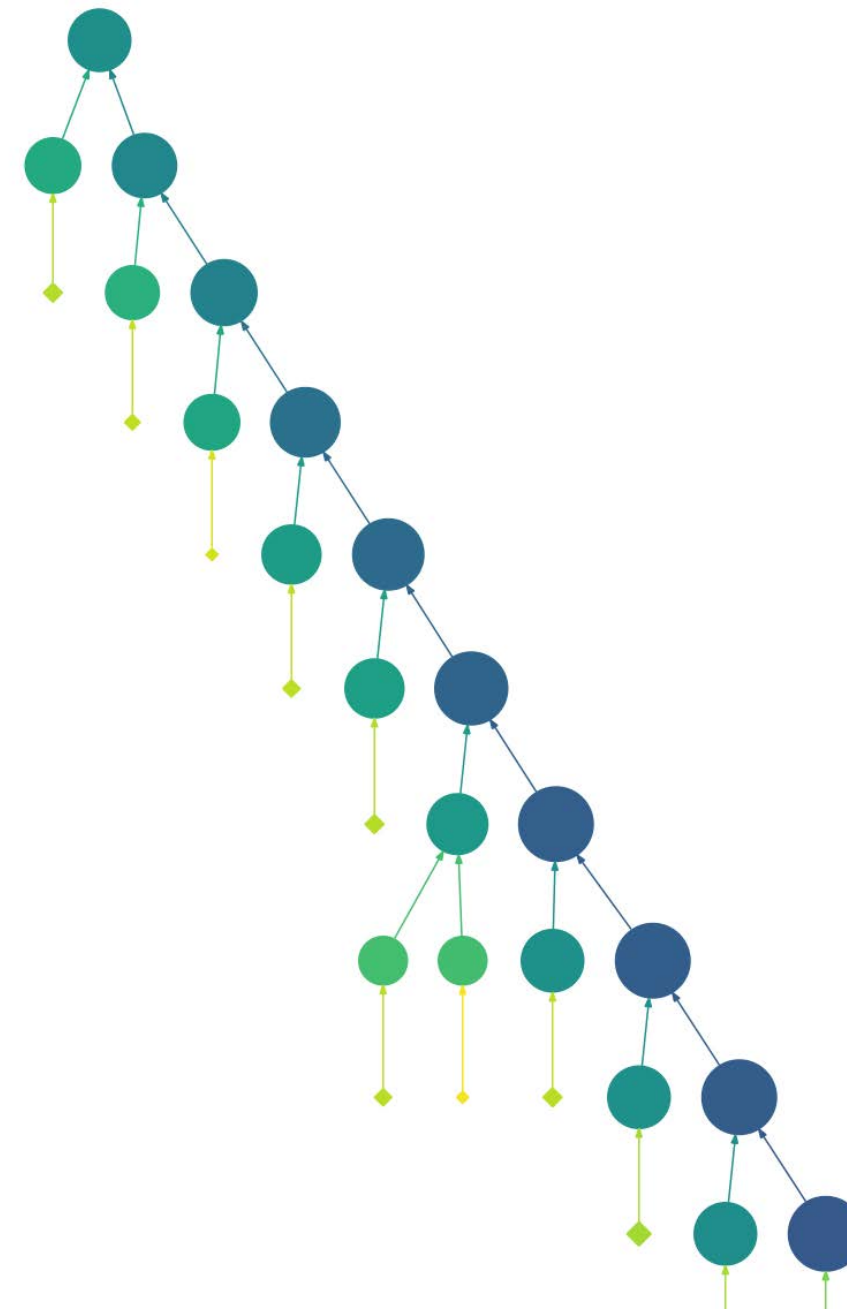
Fully connected graph

- i.e., connect each node to all other nodes



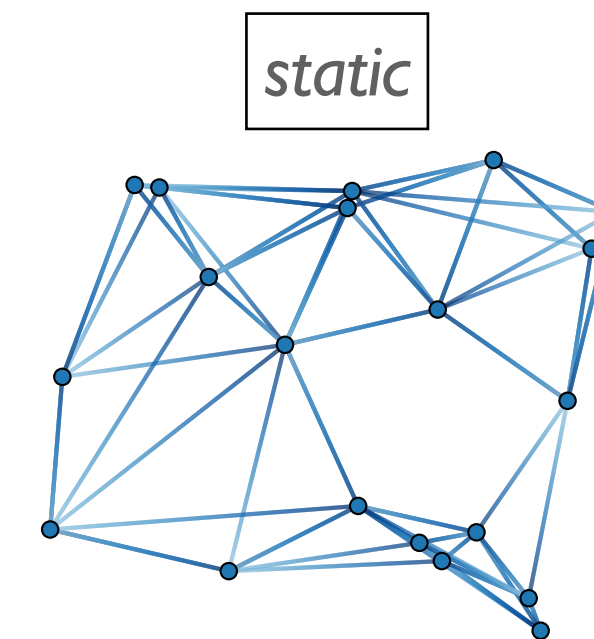
Hierarchical trees:

- decay chain
- jet clustering history

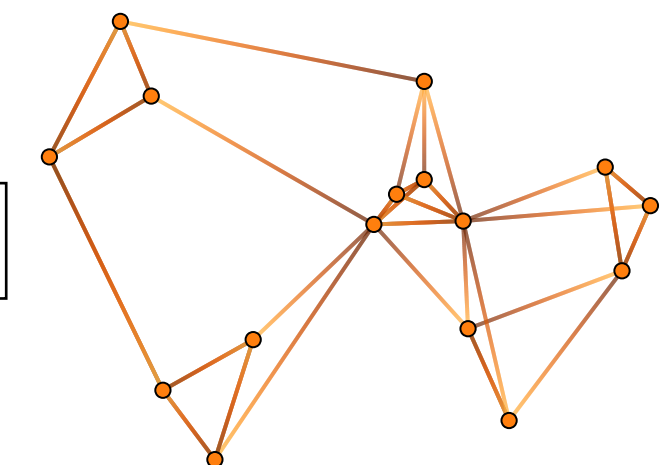


Locally connected graph

- i.e., connect each node only to neighbor nodes
 - k-nearest neighbors
 - fixed radius

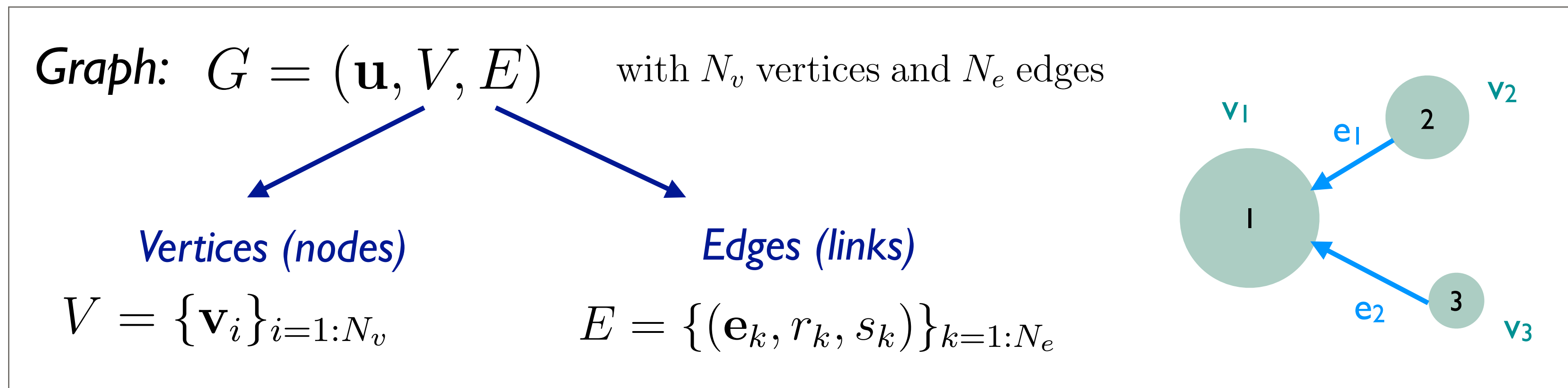


(dynamically) learned



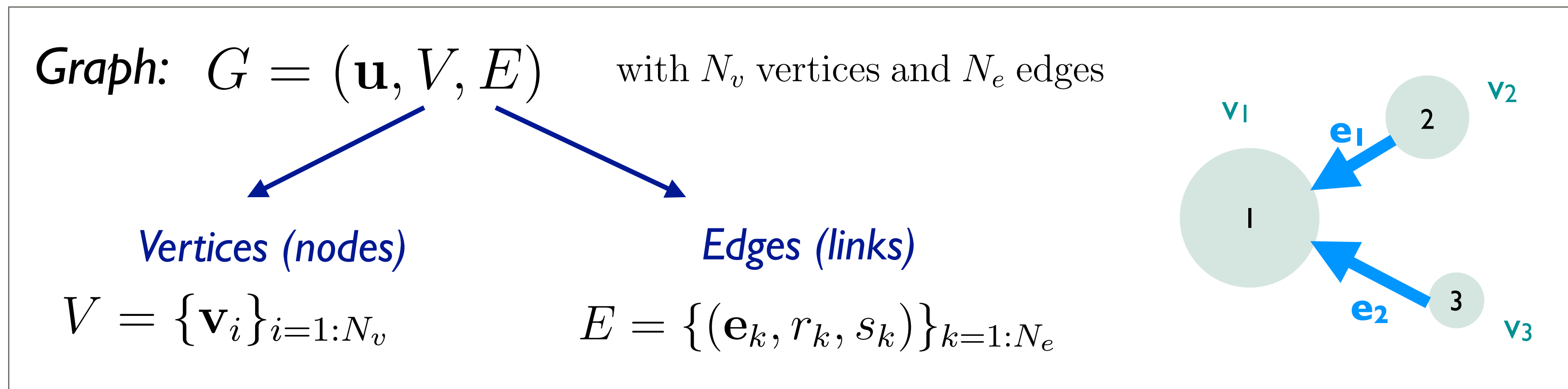
GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework



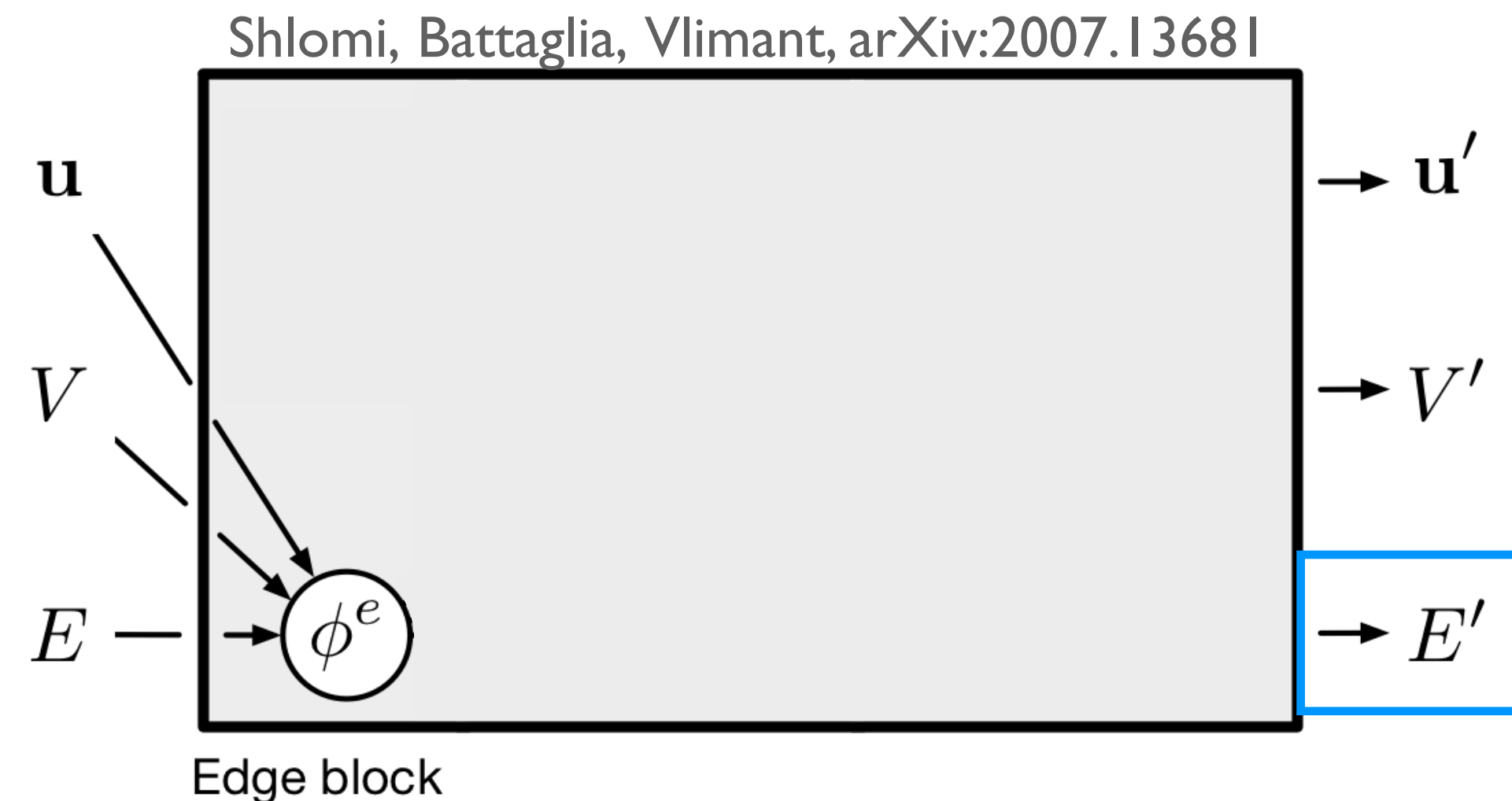
GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework



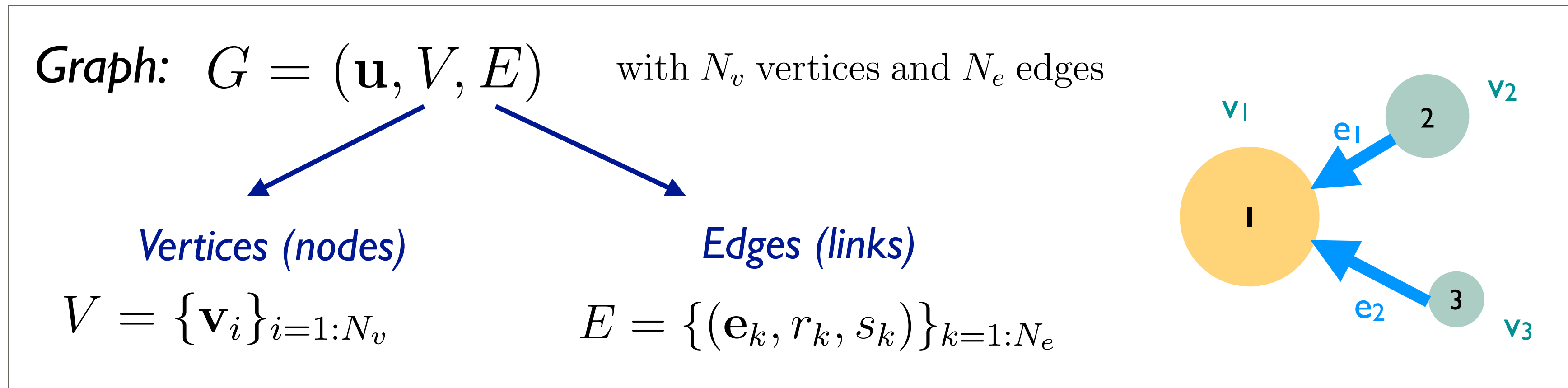
e'_k : message computed for edge k connecting nodes r_k, s_k

$$e'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u})$$



GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework



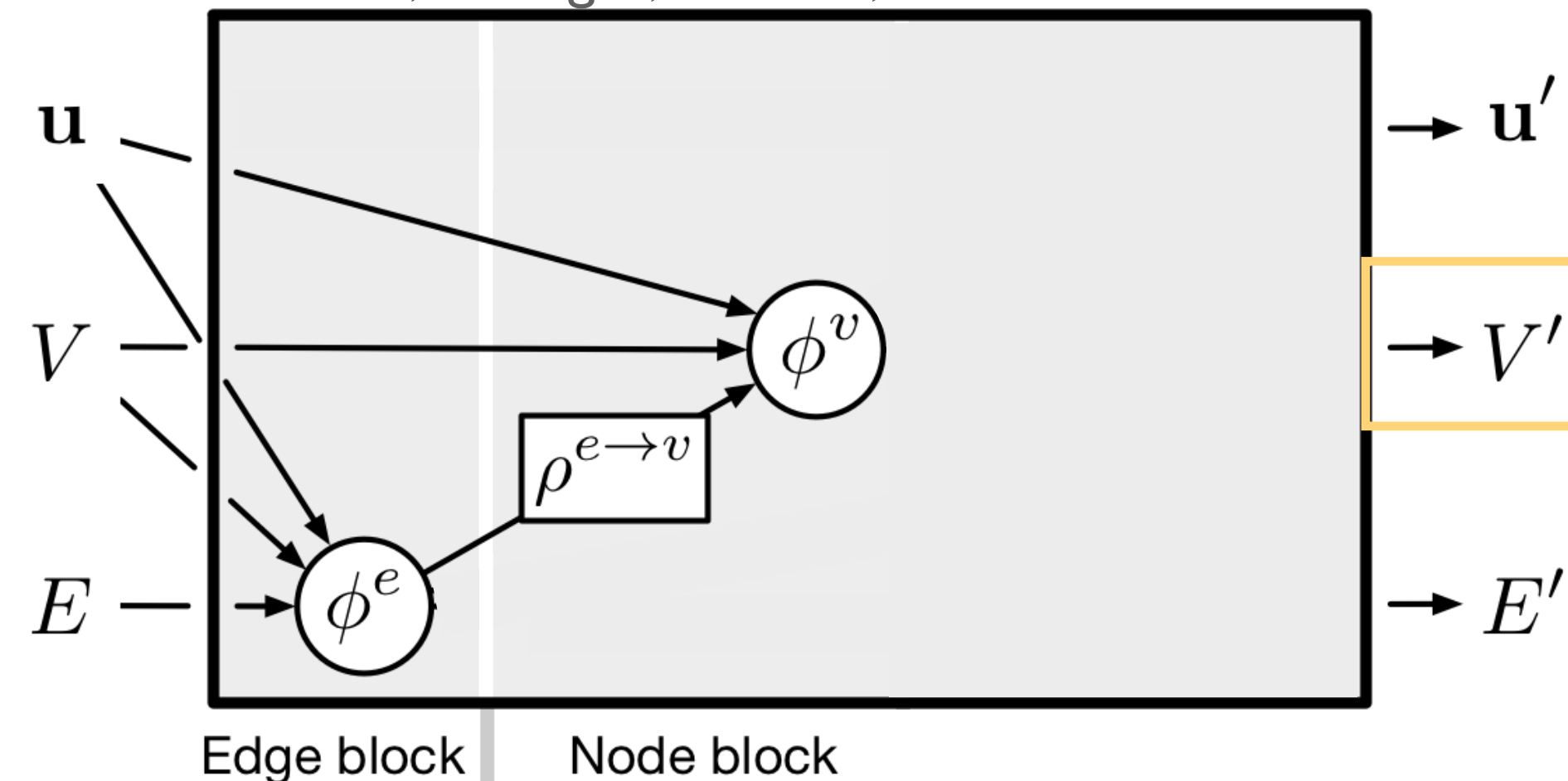
e'_k : message computed for edge k connecting nodes r_k, s_k

v'_i : node feature update based on aggregated messages and previous features

$$e'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}) \quad \bar{e}'_i = \rho^{e \rightarrow v}(E'_i)$$

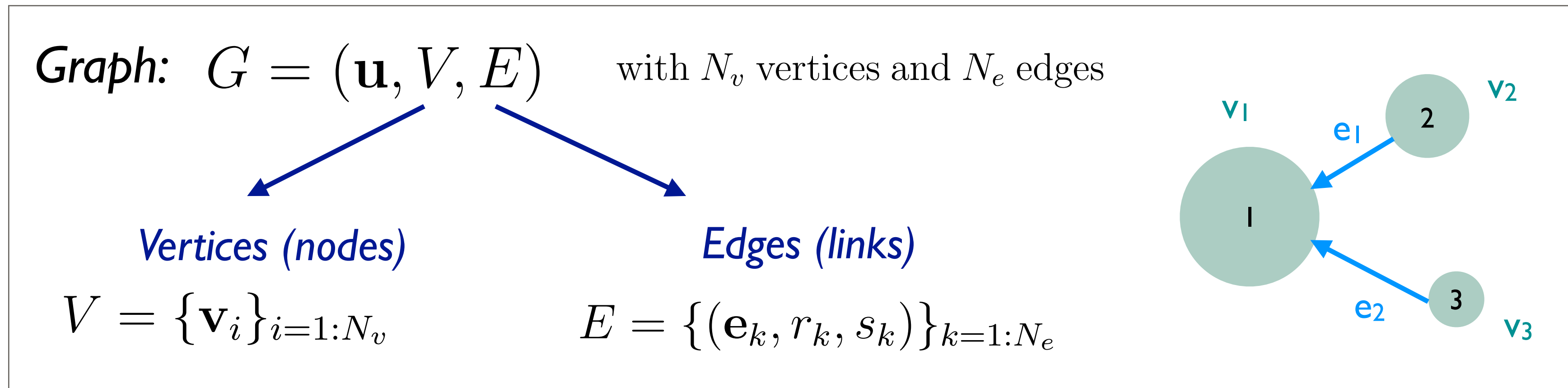
$$\mathbf{v}'_i = \phi^v(\bar{e}'_i, \mathbf{v}_i, \mathbf{u})$$

Shlomi, Battaglia, Vlimant, arXiv:2007.13681



GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework



\mathbf{e}'_k : message computed for edge k connecting nodes r_k, s_k

\mathbf{v}'_i : node feature update based on aggregated messages and previous features

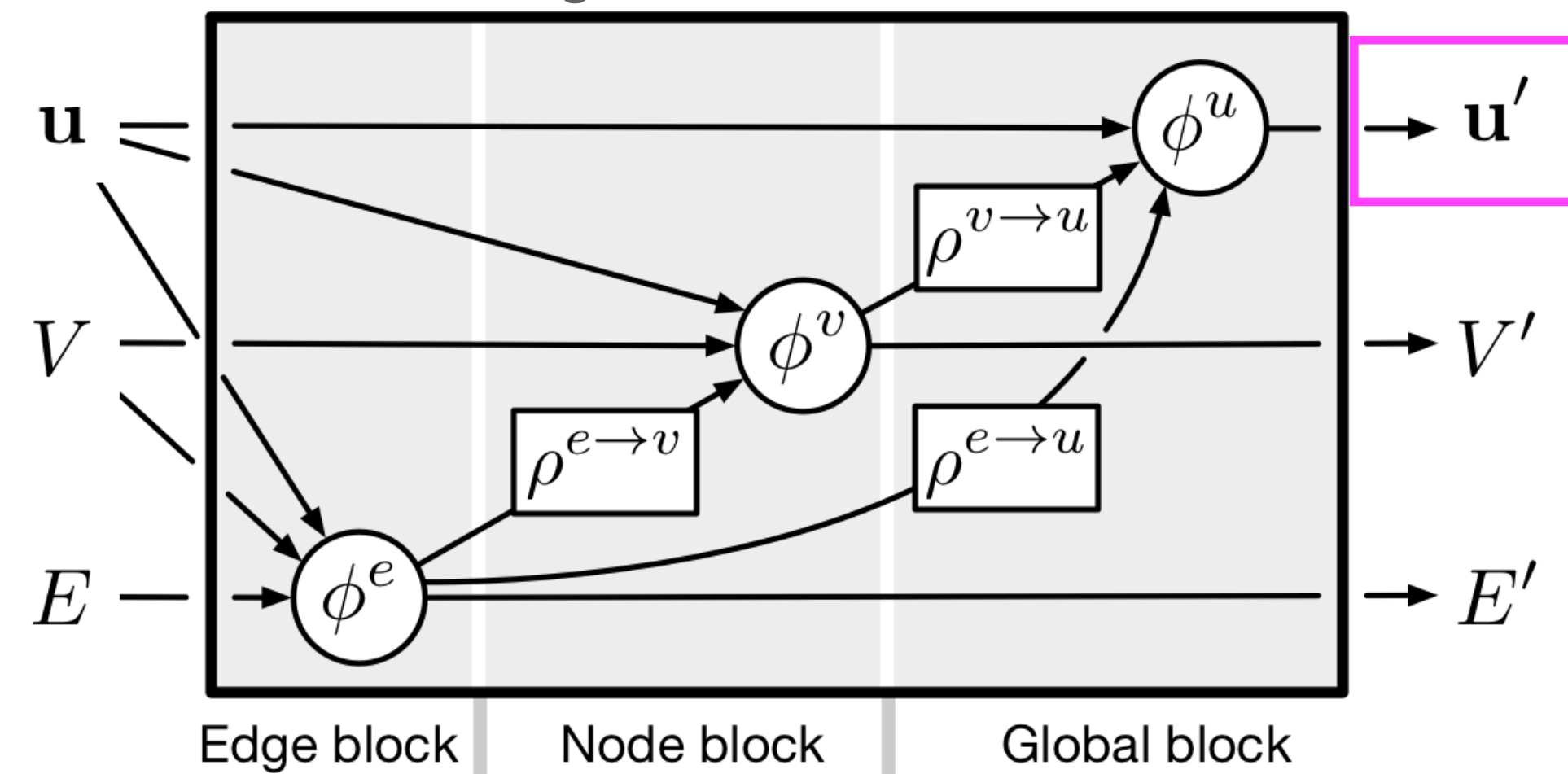
\mathbf{u}' : global feature update based on aggregated, updated node and edge features

$$\mathbf{e}'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}) \quad \bar{\mathbf{e}}'_i = \rho^{e \rightarrow v}(E'_i)$$

$$\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}) \quad \bar{\mathbf{e}}' = \rho^{e \rightarrow u}(E')$$

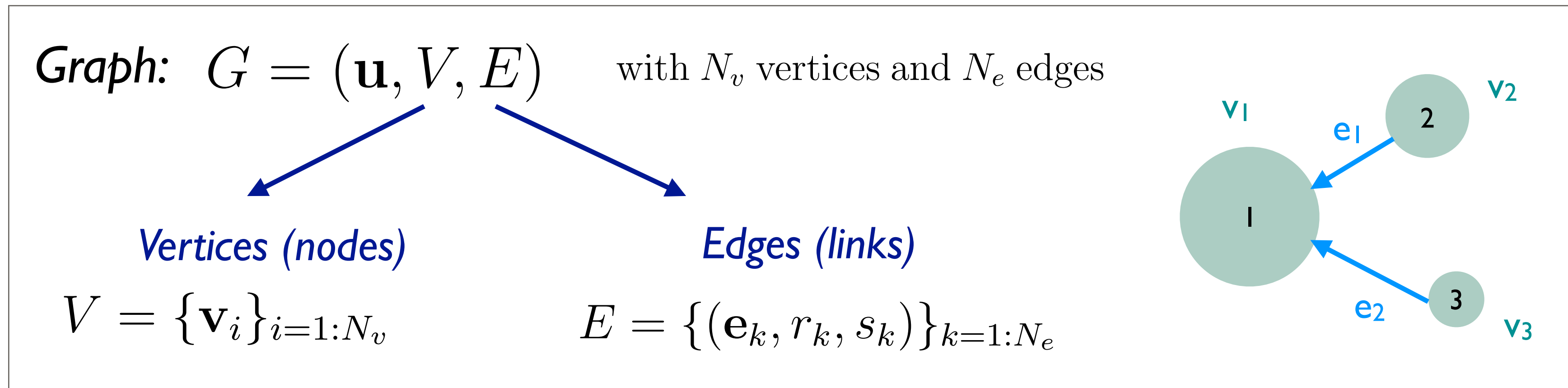
$$\mathbf{u}' = \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}) \quad \bar{\mathbf{v}}' = \rho^{v \rightarrow u}(V')$$

Shlomi, Battaglia, Vlimant, arXiv:2007.13681



GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework

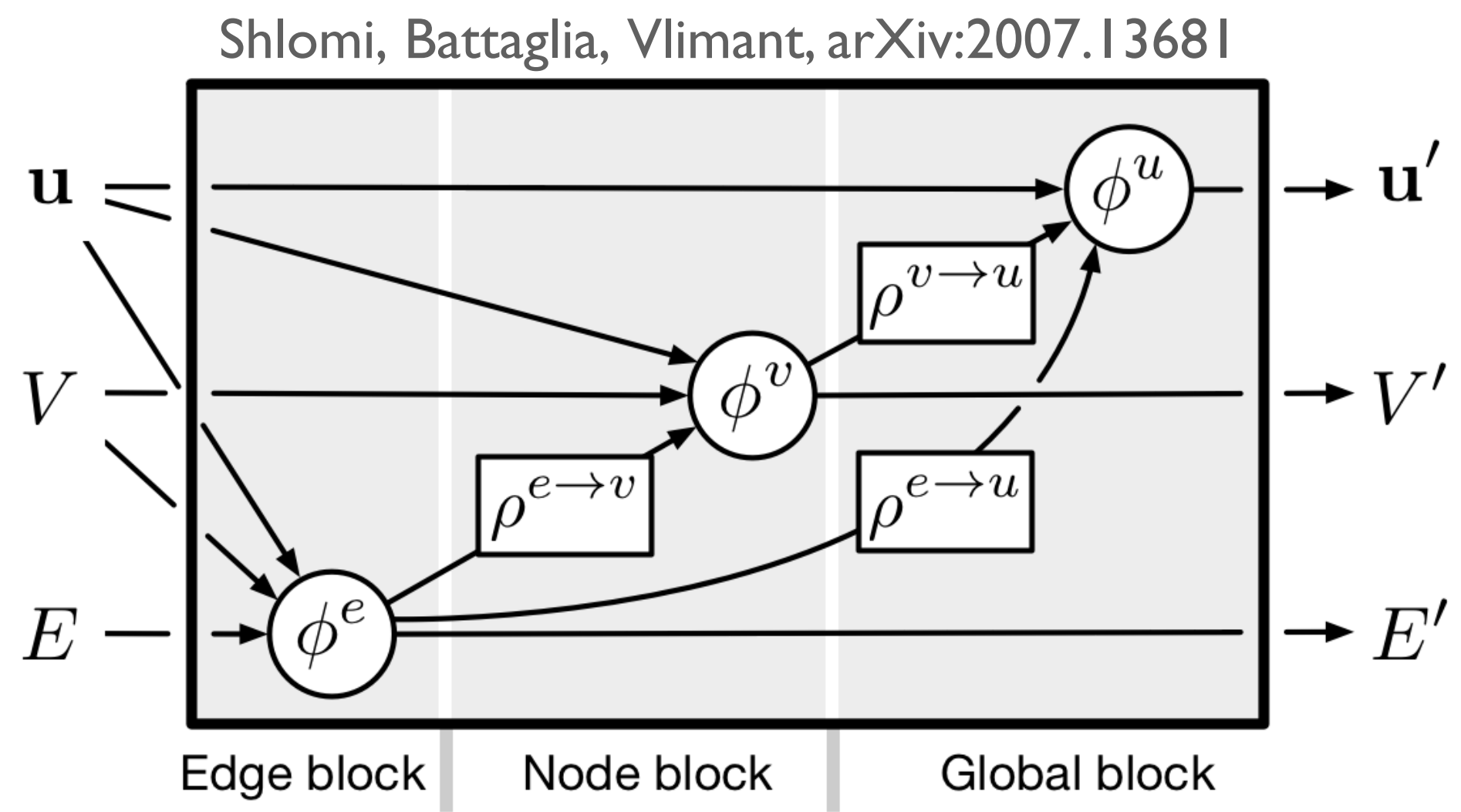


\mathbf{e}'_k : message computed for edge k connecting nodes r_k, s_k

\mathbf{v}'_i : node feature update based on aggregated messages and previous features

\mathbf{u}' : global feature update based on aggregated, updated node and edge features

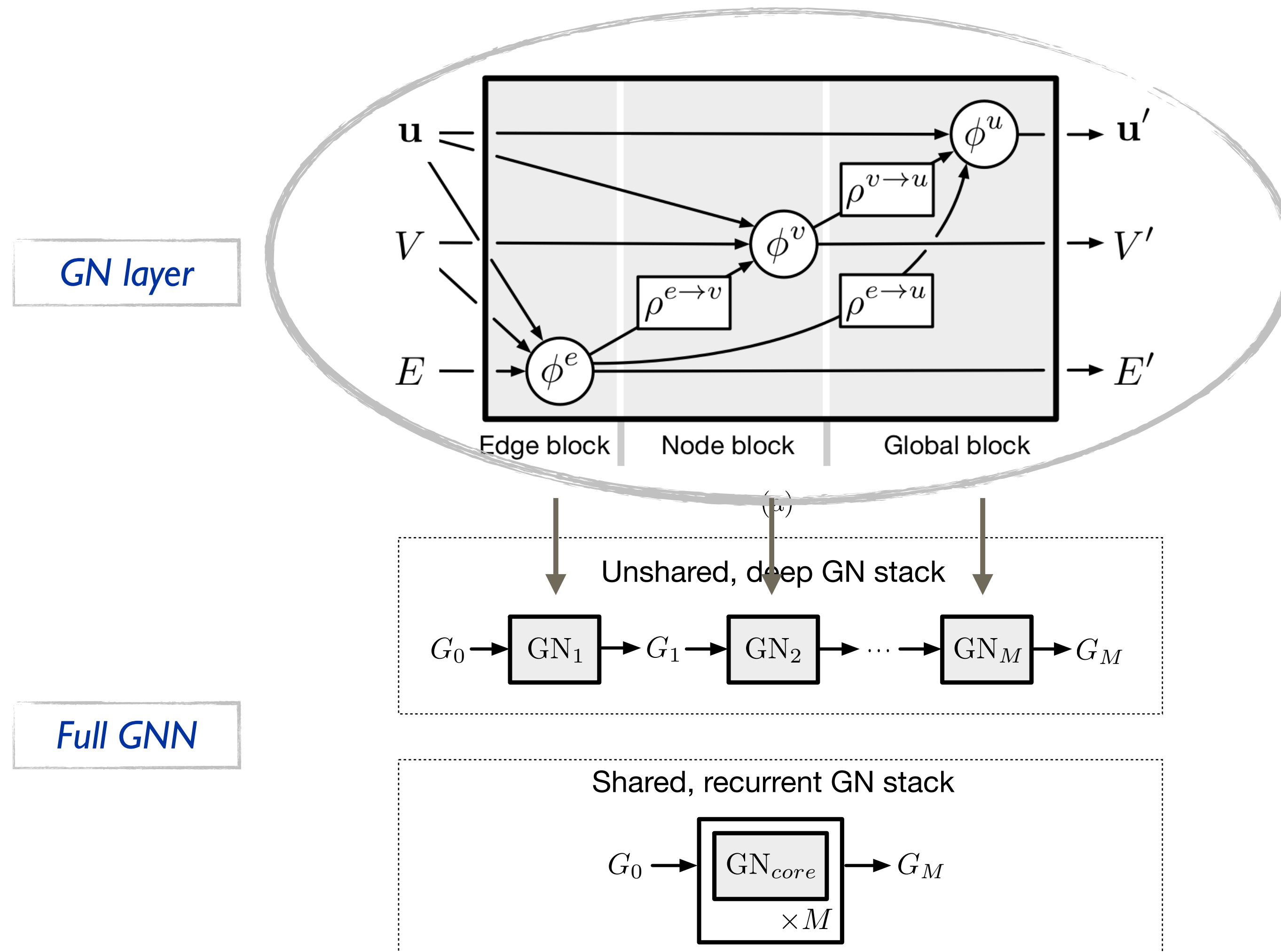
$\mathbf{e}'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u})$ $\bar{\mathbf{e}}'_i = \rho^{e \rightarrow v}(E'_i)$
 $\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u})$ $\bar{\mathbf{e}}' = \rho^{e \rightarrow u}(E')$
 $\mathbf{u}' = \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u})$ $\bar{\mathbf{v}}' = \rho^{v \rightarrow u}(V')$



Shared-weight NN Symmetric functions (e.g., sum, mean, max, etc.) \rightarrow Permutation invariance

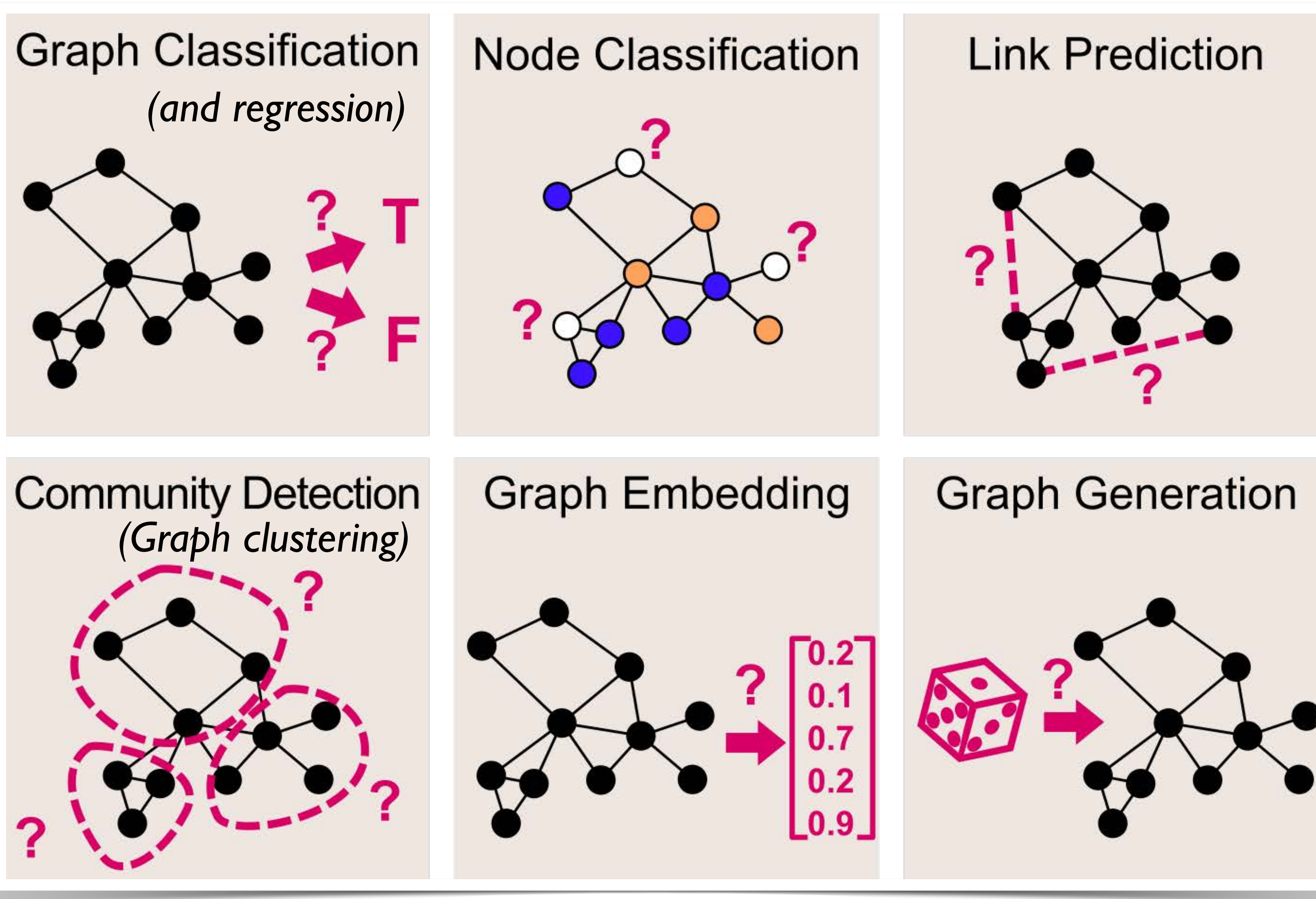
GRAPH NETWORK FORMALISM

- Typical graph neural networks (GNNs) can be described in the “Message Passing” framework

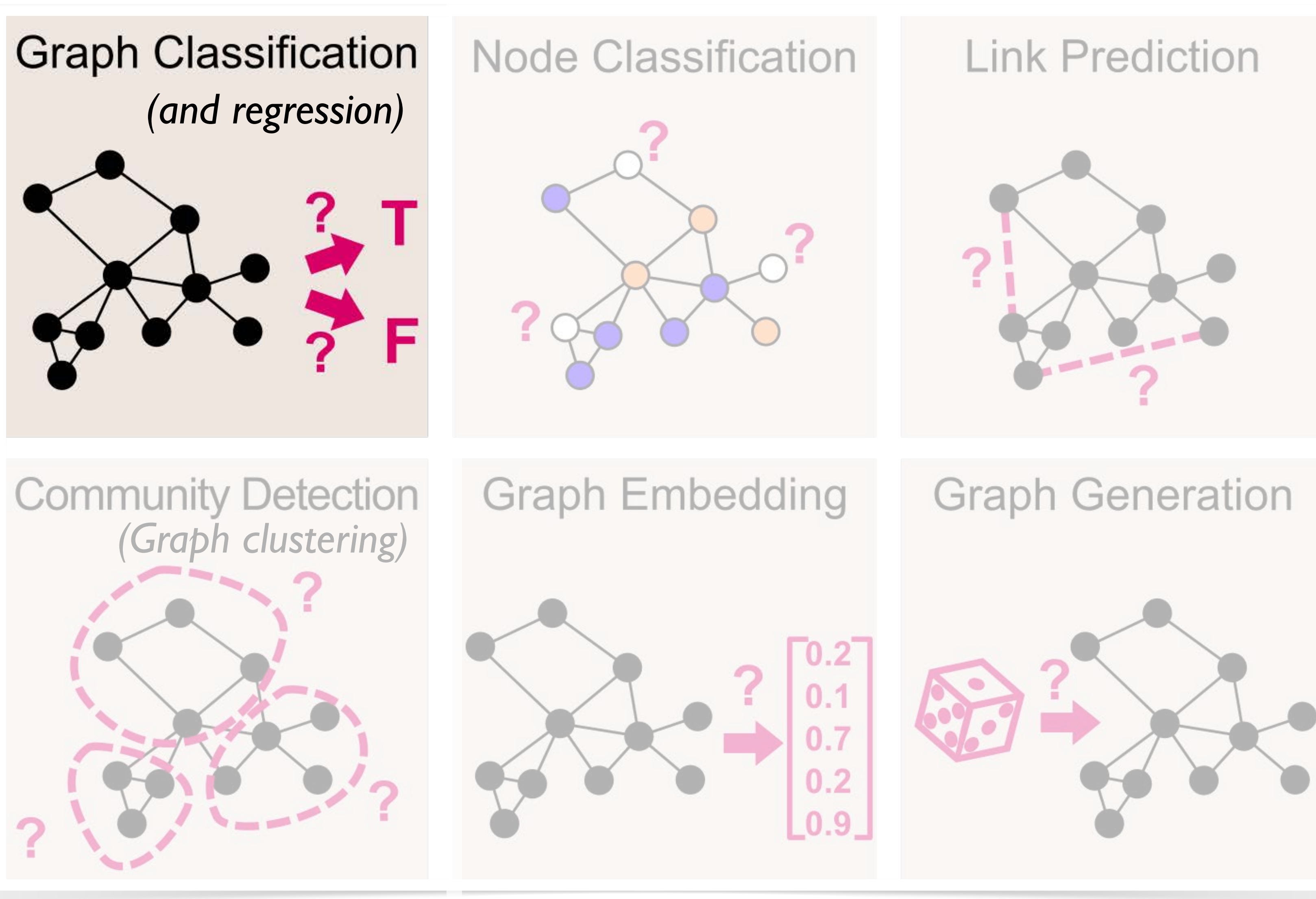


GRAPH NEURAL NETWORKS IN ACTION

GRAPH ML TASKS



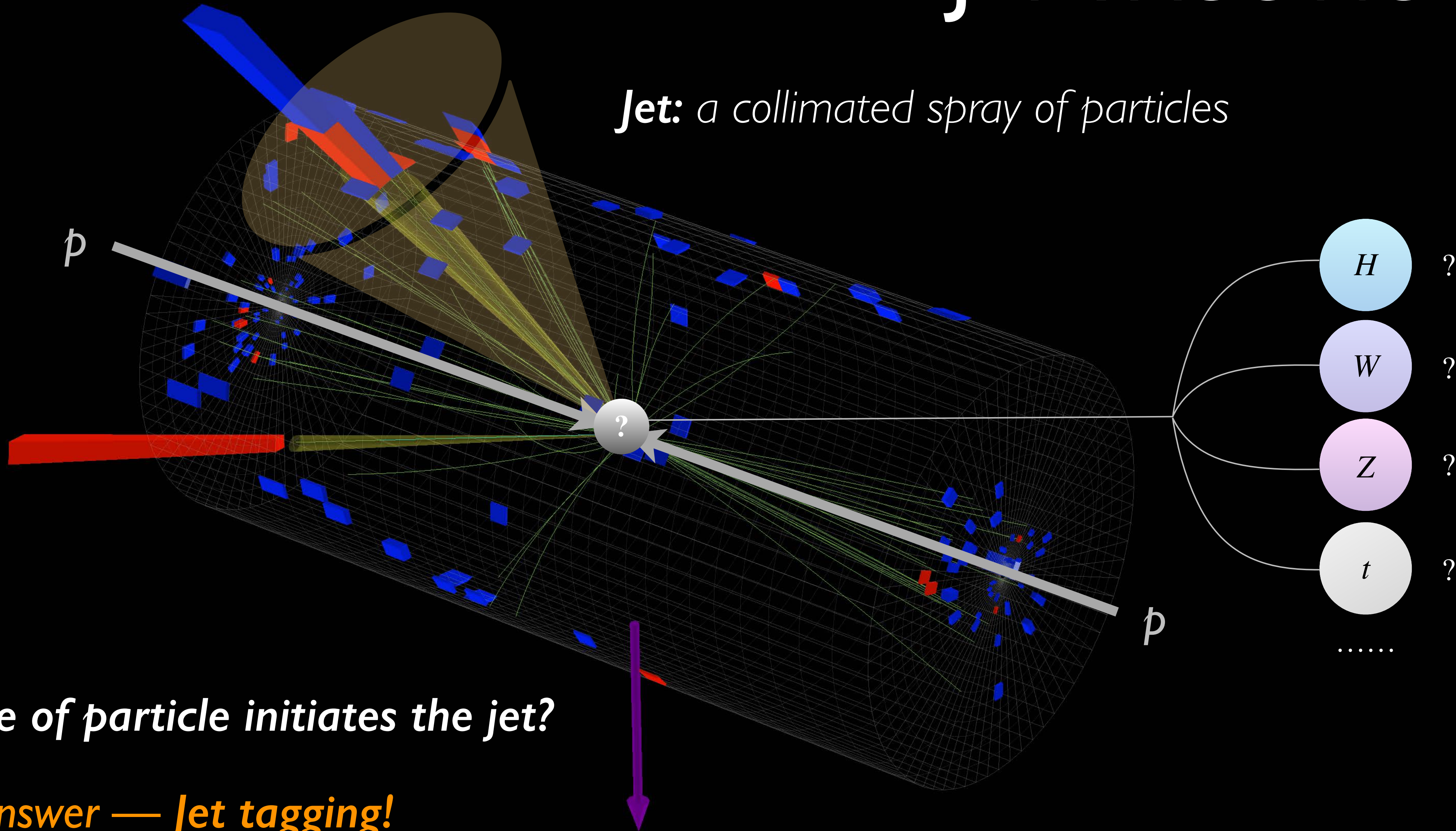
GRAPH ML TASKS





JET TAGGING

Jet: a collimated spray of particles



Key question:

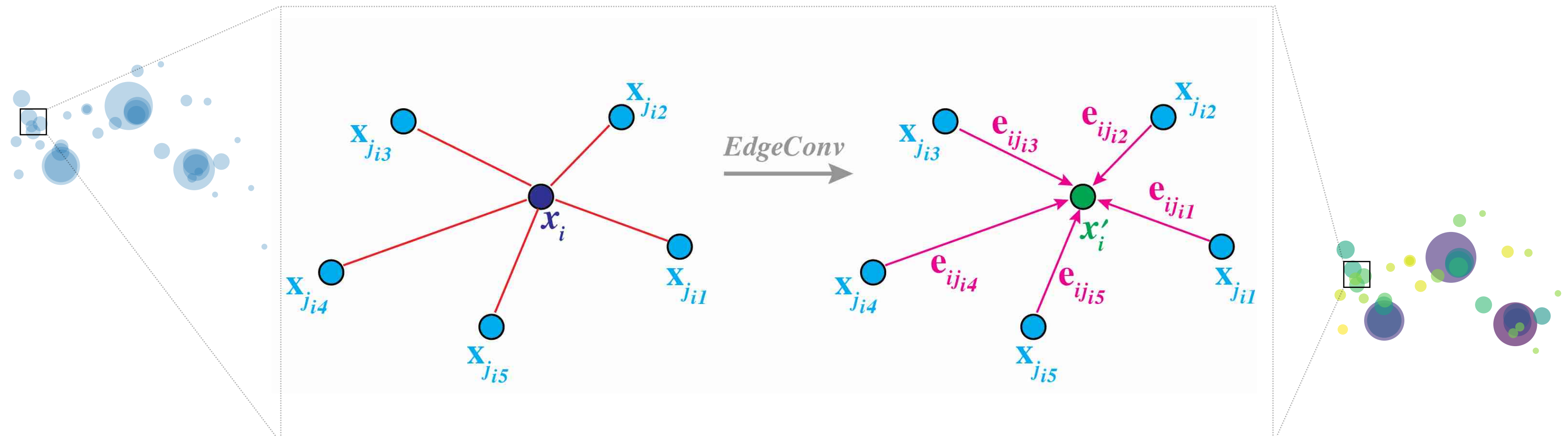
What type of particle initiates the jet?

The answer — Jet tagging!

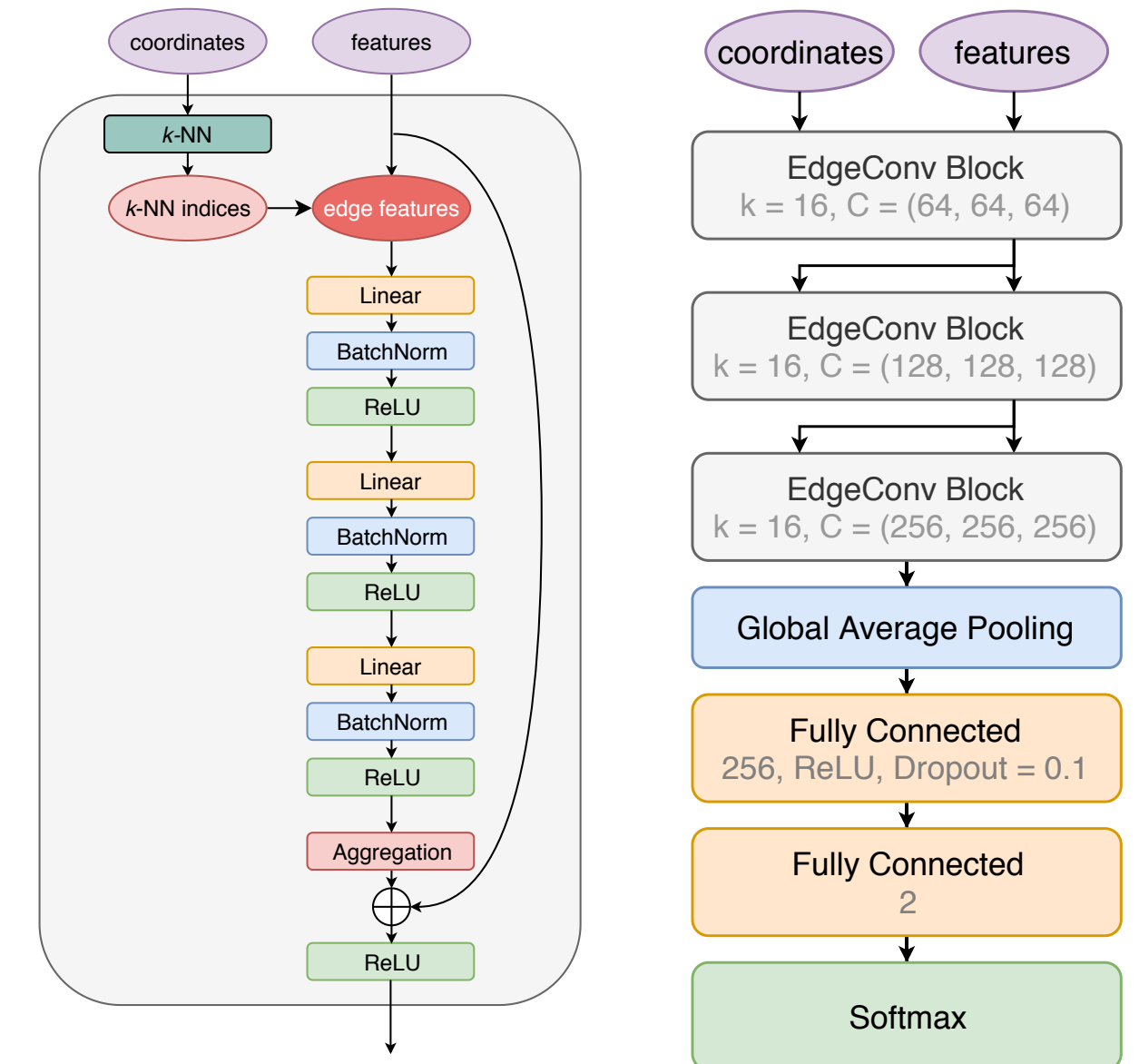
PARTICLENET

HQ and L. Gouskos
Phys.Rev.D 101 (2020) 5, 056019

- ParticleNet: jet tagging via particle clouds
 - treating a jet as an **unordered set of particles**, distributed in the $\eta - \phi$ space
 - **graph neural network architecture**, adapted from Dynamic Graph CNN [arXiv:1801.07829]
 - treating a point cloud as a graph: each point is a vertex
 - for each point, a local patch is defined by finding its k-nearest neighbors
 - designing a permutation-invariant “convolution” function
 - define “edge feature” for each center-neighbor pair: $e_{ij} = \text{MLP}(x_i, x_j)$
 - aggregate the edge features in a symmetric way: $x'_i = \text{mean}_j e_{ij}$

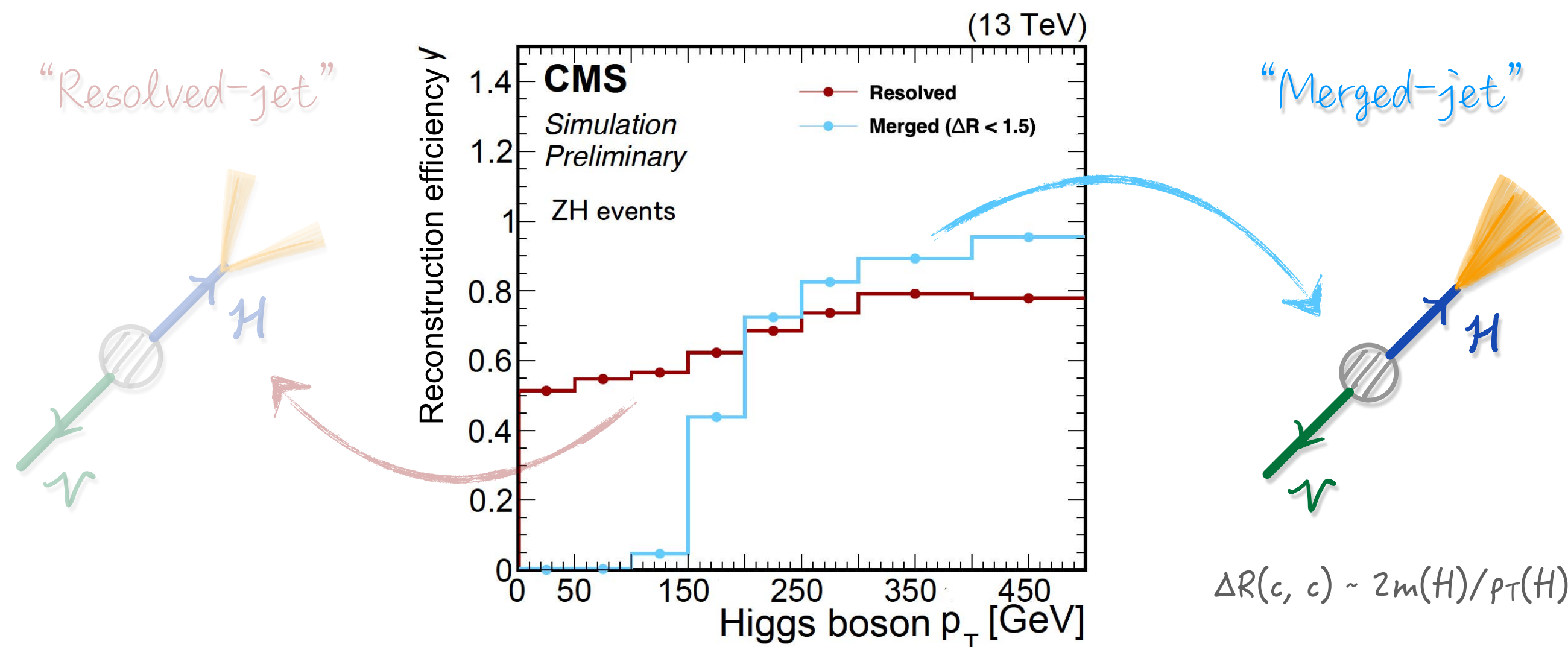
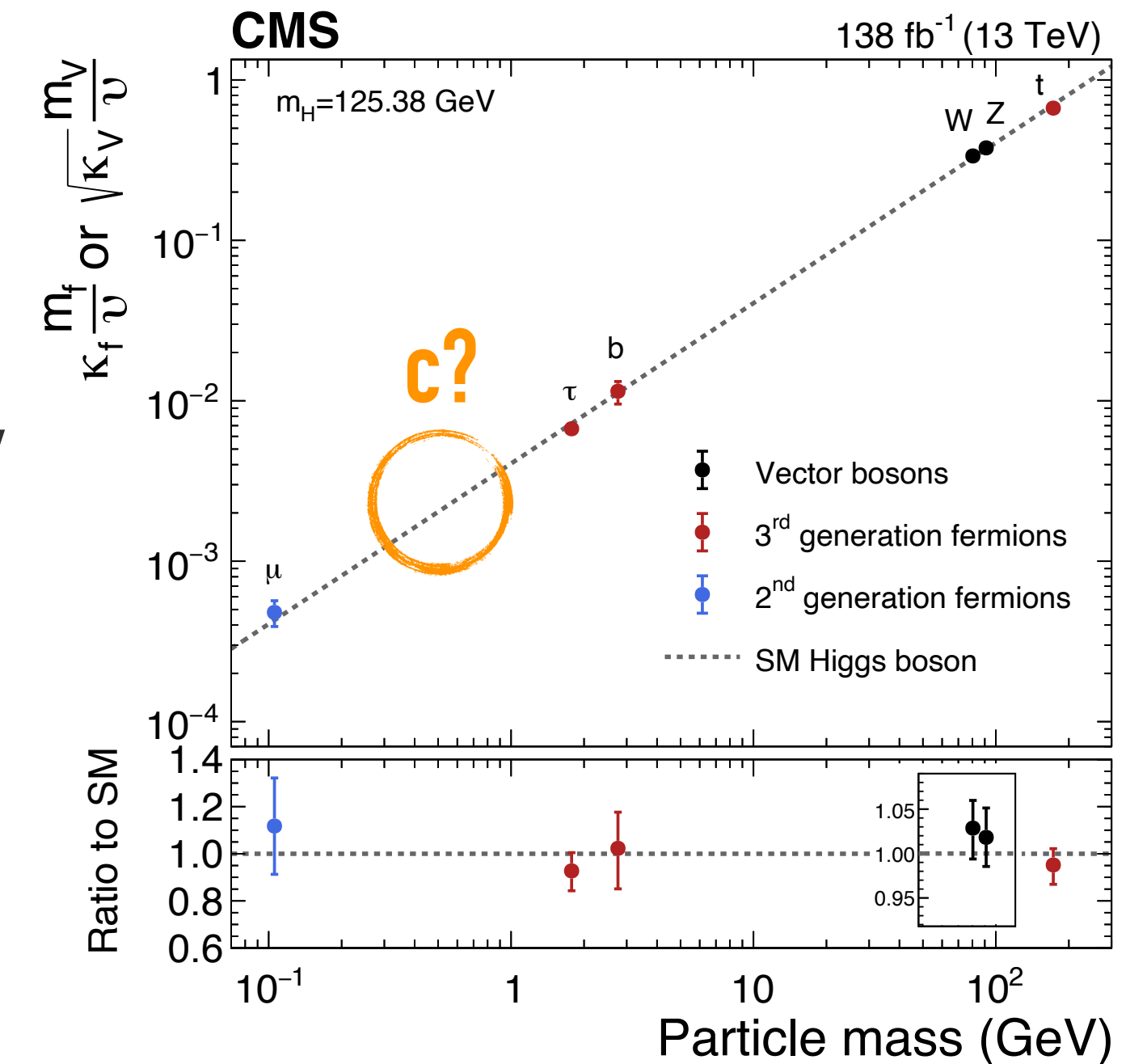


ParticleNet architecture



PARTICLENET IN ACTION: $H \rightarrow CC$ SEARCH

- **Higgs-charm coupling: next milestone in Higgs physics**
 - a crucial test of fermion mass generation mechanism in SM
 - $H \rightarrow cc$: extremely challenging search at the LHC
 - small branching fraction ($\sim 3\%$) vs enormous backgrounds – **charm tagging** is the key
- **Innovative approach: search for $VH(H \rightarrow cc)$ in the “merged-jet” topology**
 - reconstructs $H \rightarrow cc$ decay with one large- R jet ($R=1.5$)
 - then: exploits advanced ML for $H \rightarrow cc$ identification

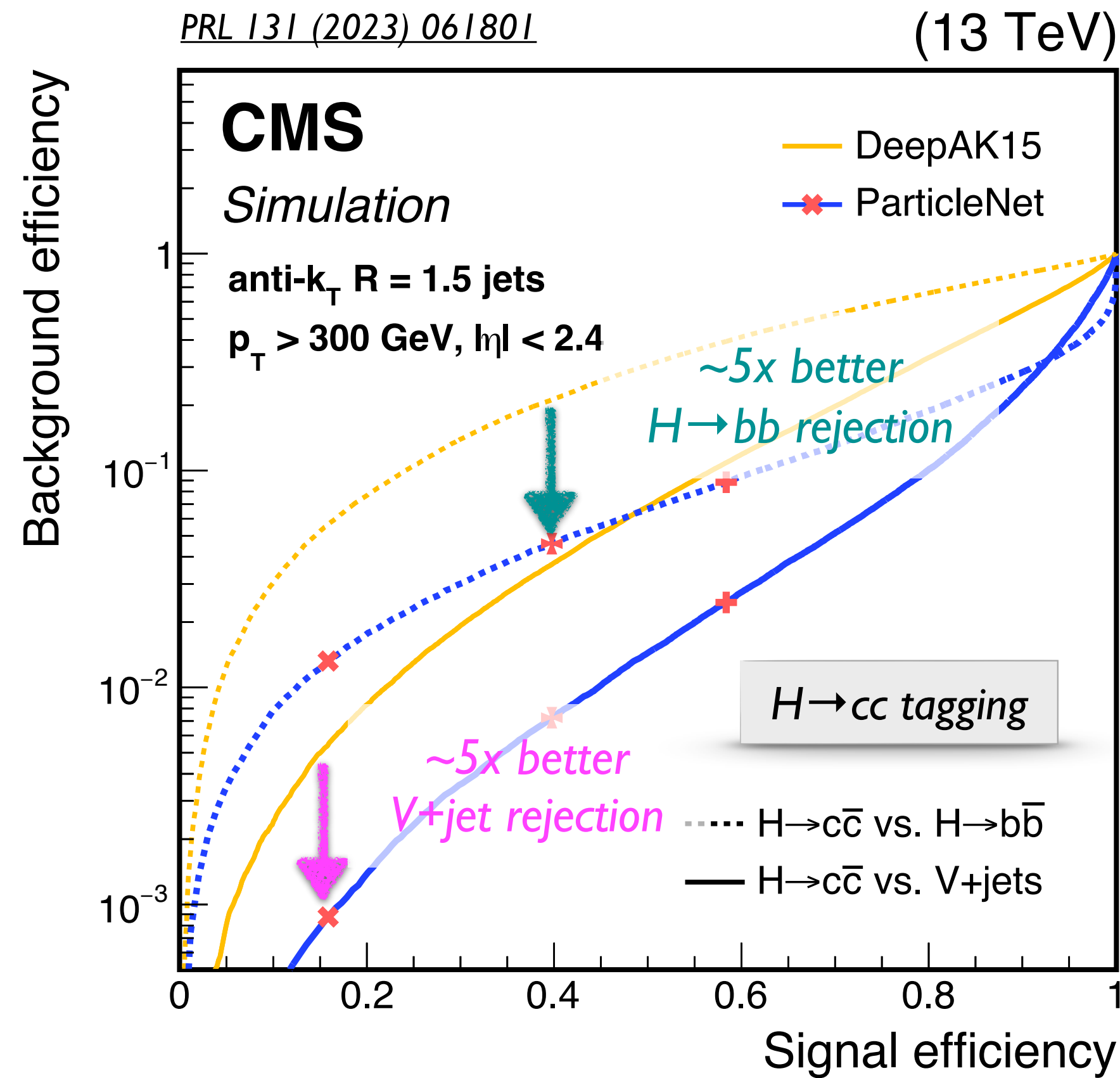


Why merged-jet topology?

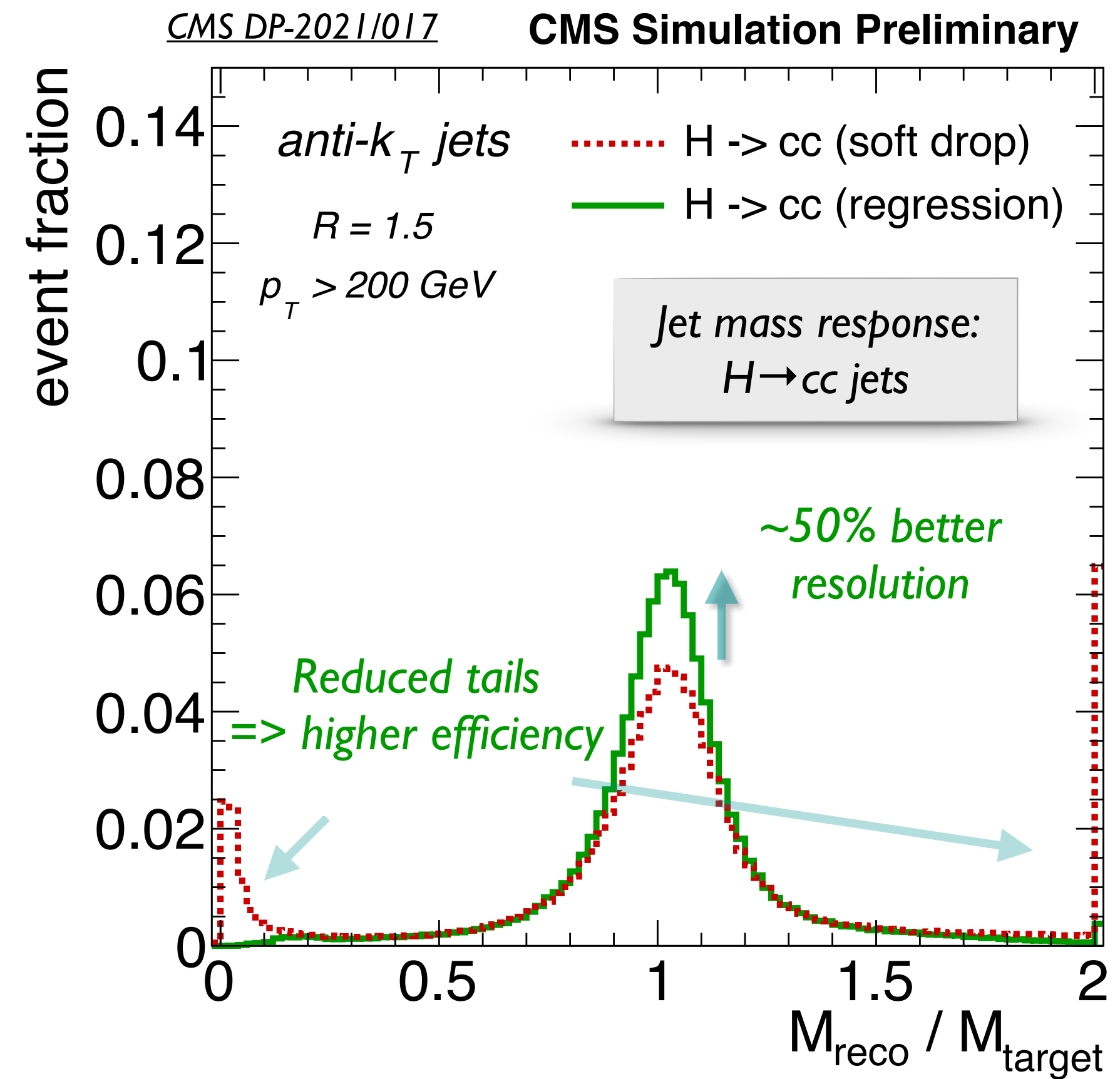
- better signal purity at higher p_T
- higher reconstruction efficiency with large- R jets
- better exploiting correlations between the two charm quarks — especially with deep learning

PARTICLENET IN ACTION: $H \rightarrow CC$ SEARCH

- ParticleNet for $H \rightarrow cc$ jet tagging and mass reconstruction: substantial improvements



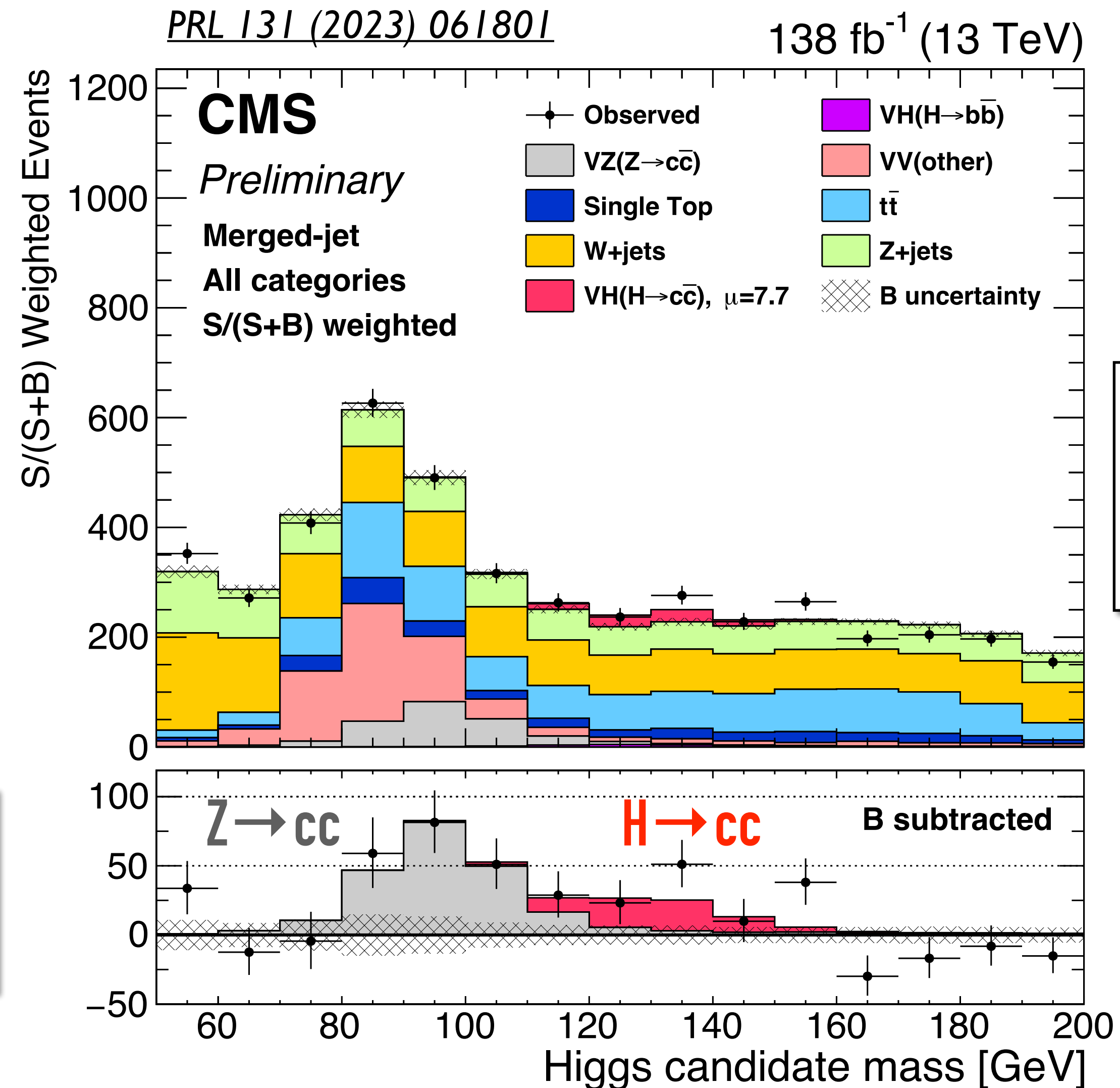
ParticleNet tagger for $H \rightarrow cc$ tagging
>2x improvement in final sensitivity



ParticleNet-based jet mass regression
~20-25% improvement in final sensitivity

PARTICLENET IN ACTION: $H \rightarrow CC$ SEARCH

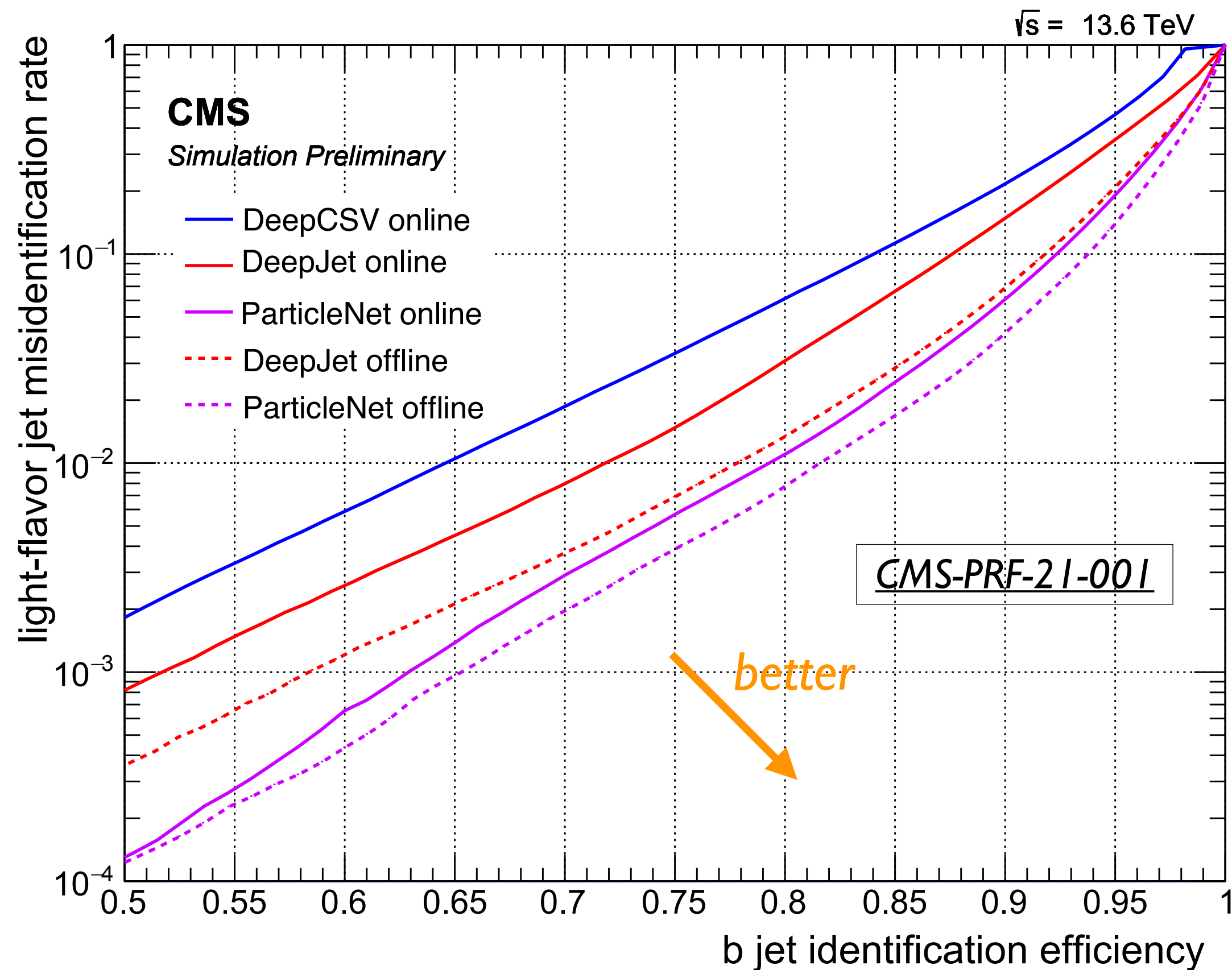
- ParticleNet for $H \rightarrow cc$ jet tagging and mass reconstruction: substantial improvements



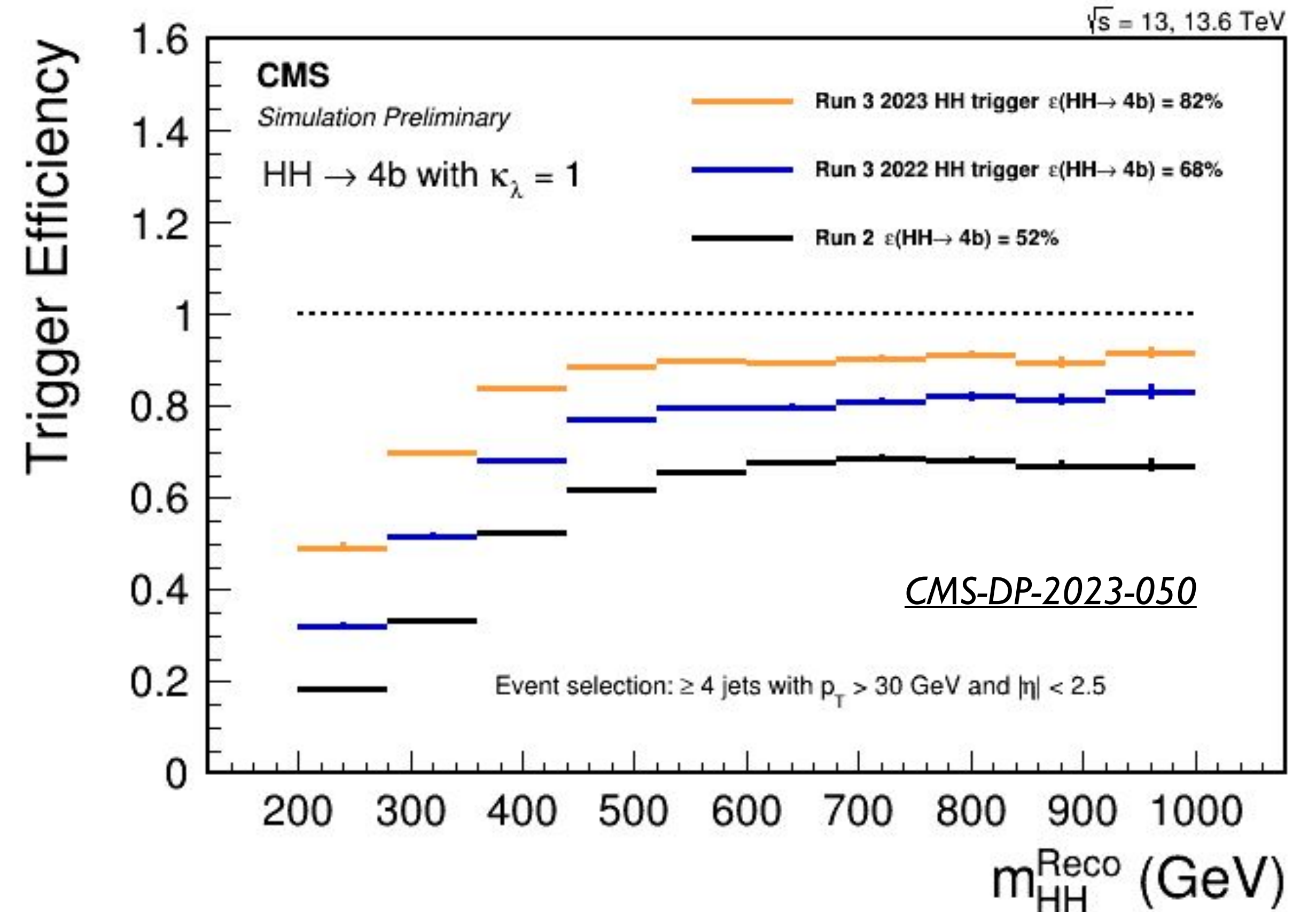
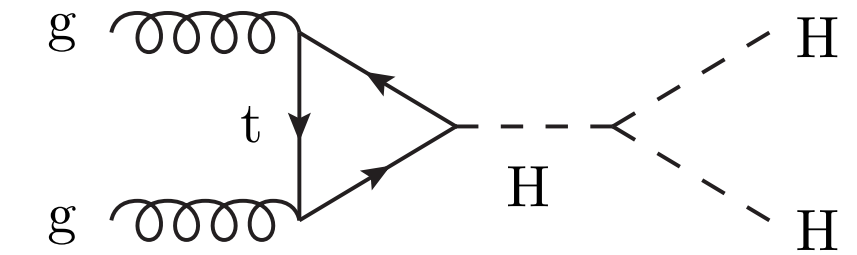
PARTICLENET IN ACTION: ONLINE EVENT SELECTION

- ParticleNet also deployed at the CMS High-Level Trigger (HLT) system for online event selection since Run3
 - substantial improvement in trigger efficiency for e.g., di-Higgs searches

b-tagging performance @ HLT



Trigger efficiency for $HH \rightarrow 4b$

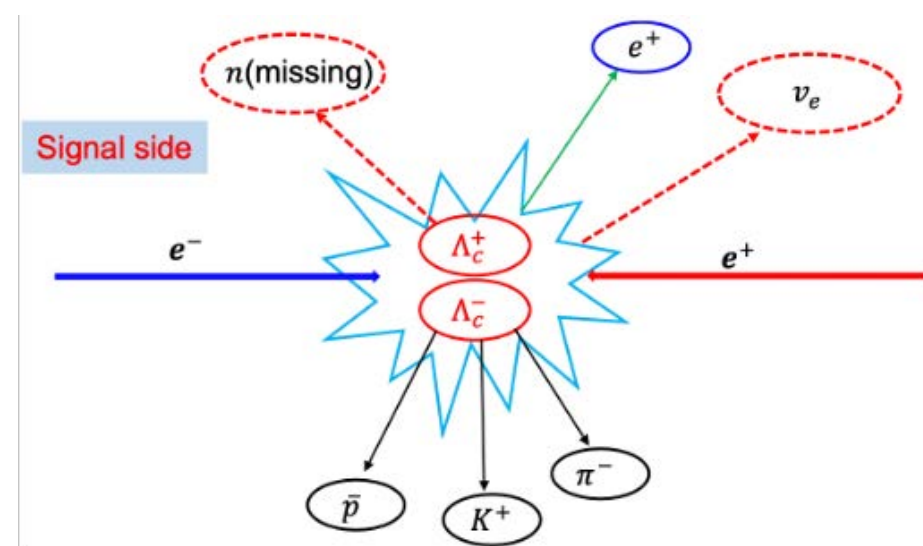


PARTICLENET IN ACTION: BEYOND JETS



$\Lambda_c^+ \rightarrow ne^+\nu$ search

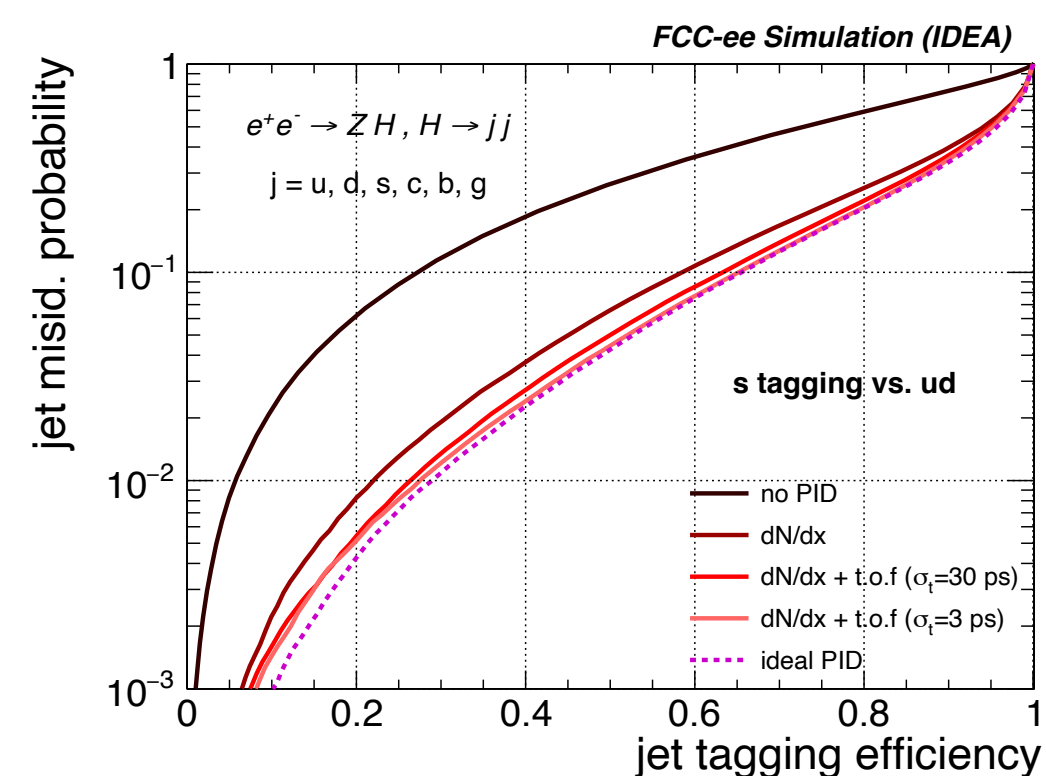
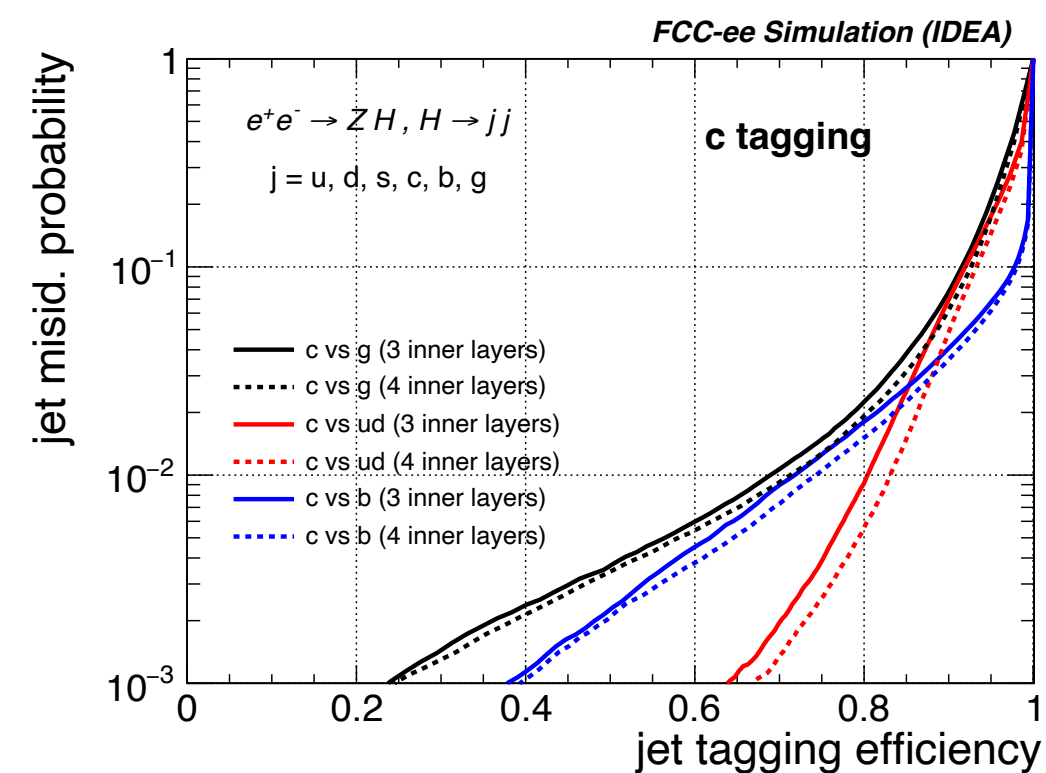
Yunxuan Song, Yangu Li et al.



Particle identification

Eur.Phys.J.Plus 137 (2022) 1, 39

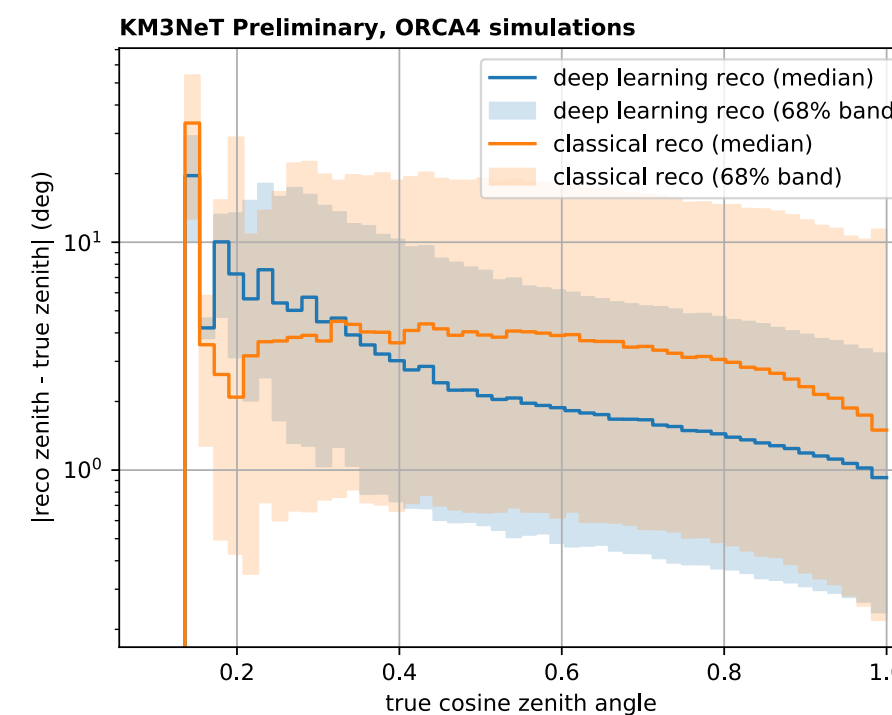
Eur.Phys.J.C 82 (2022) 7, 646



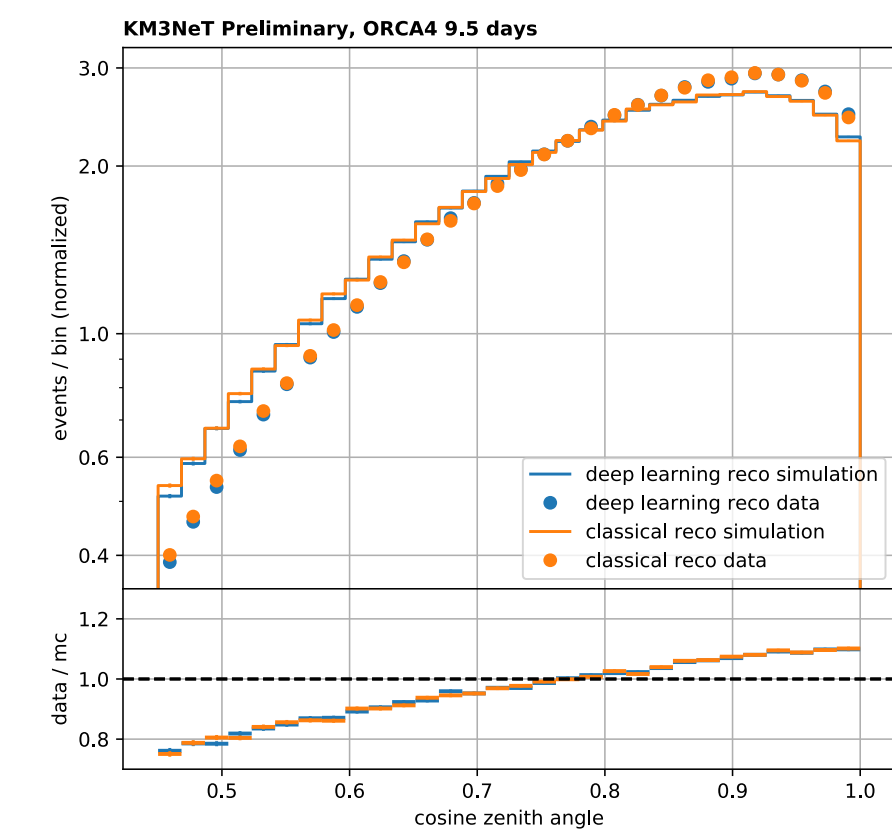
Muon bundle reconstruction

JINST 16 (2021) 10, C10011

PoS ICRC2021 (2021) 1048

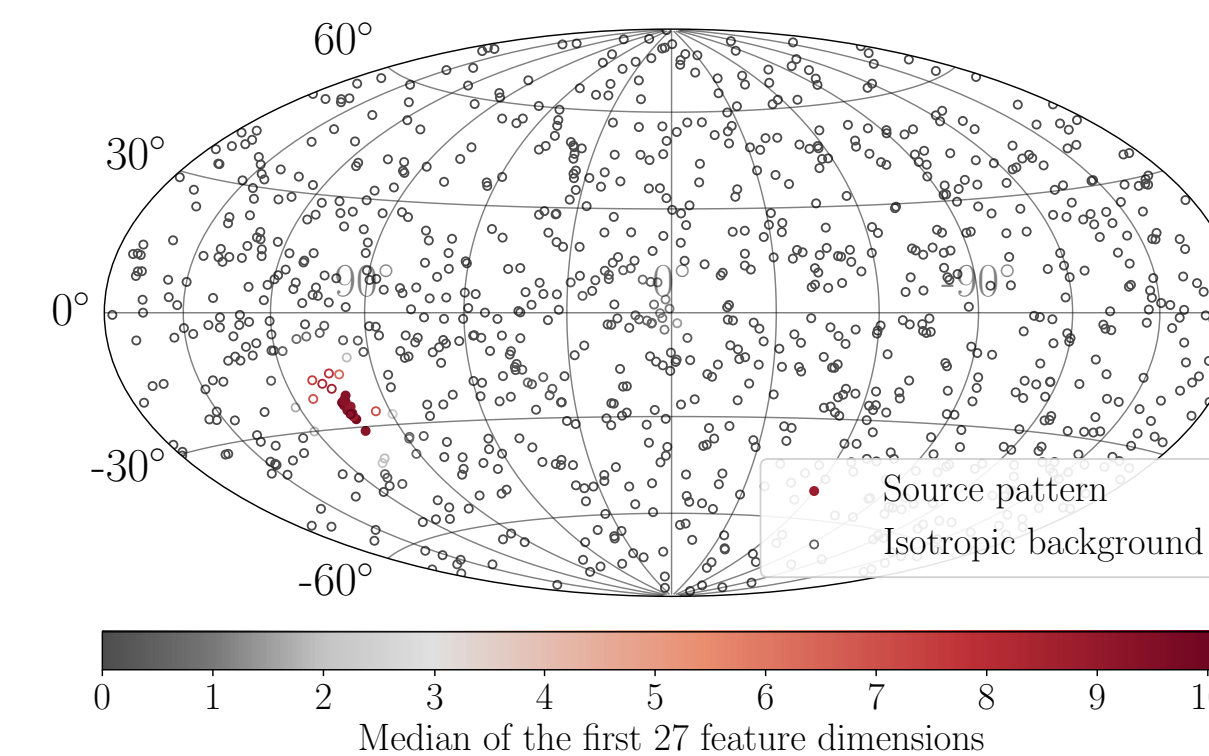
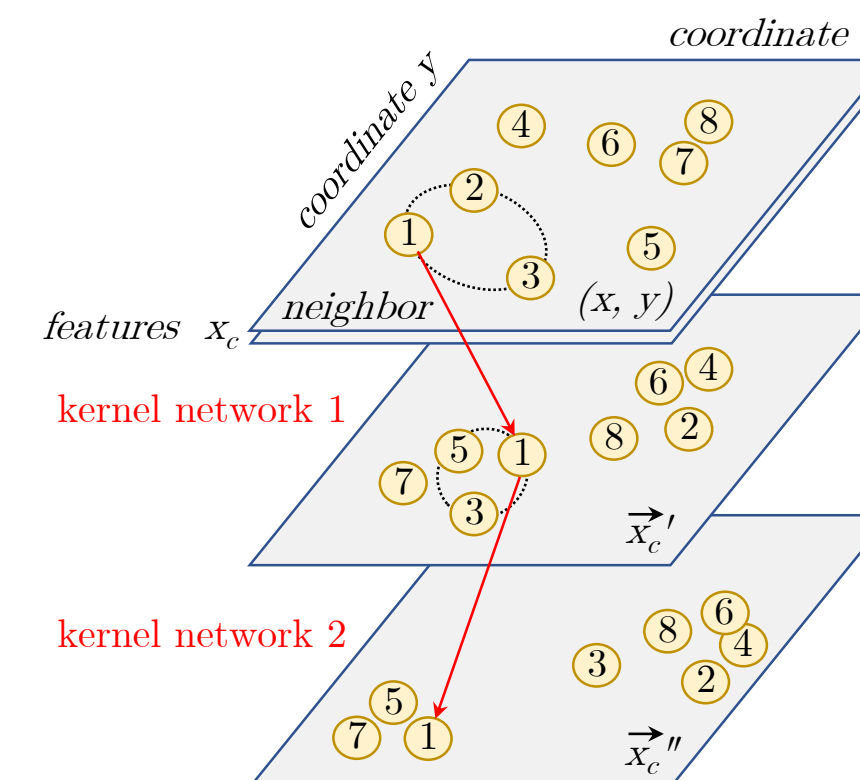


(b) events with two or more muons



Cosmic ray pattern identification

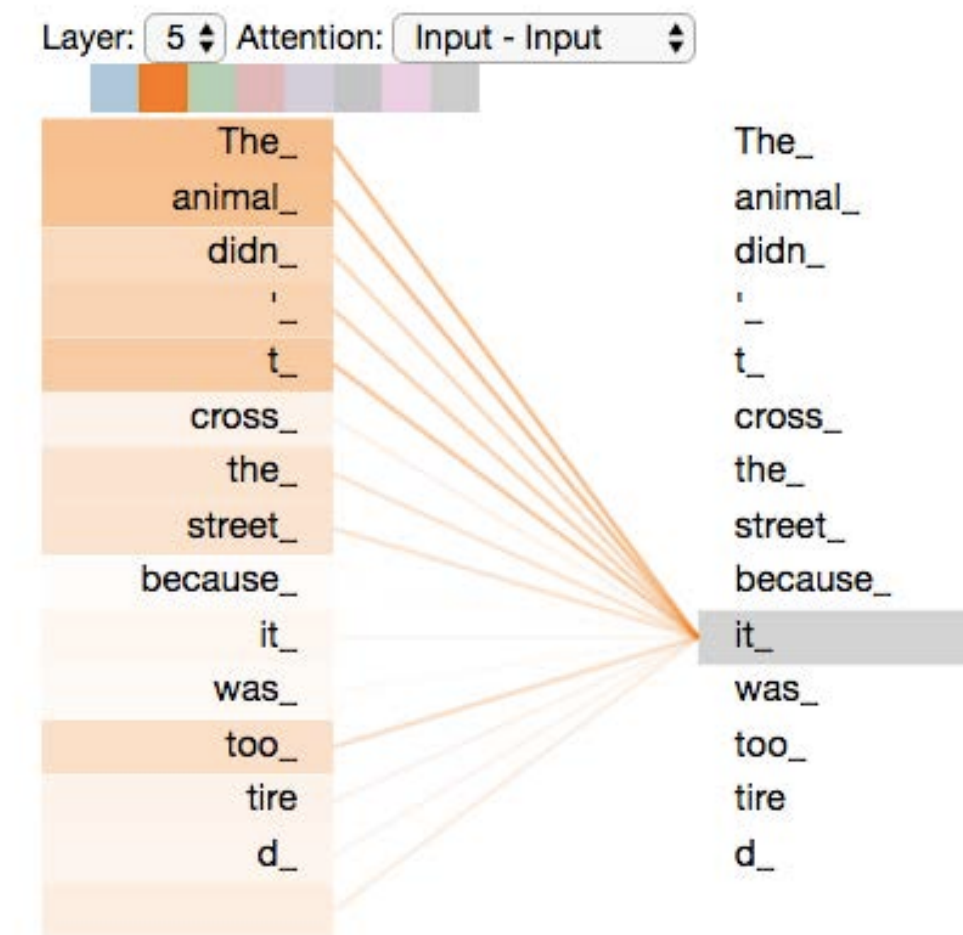
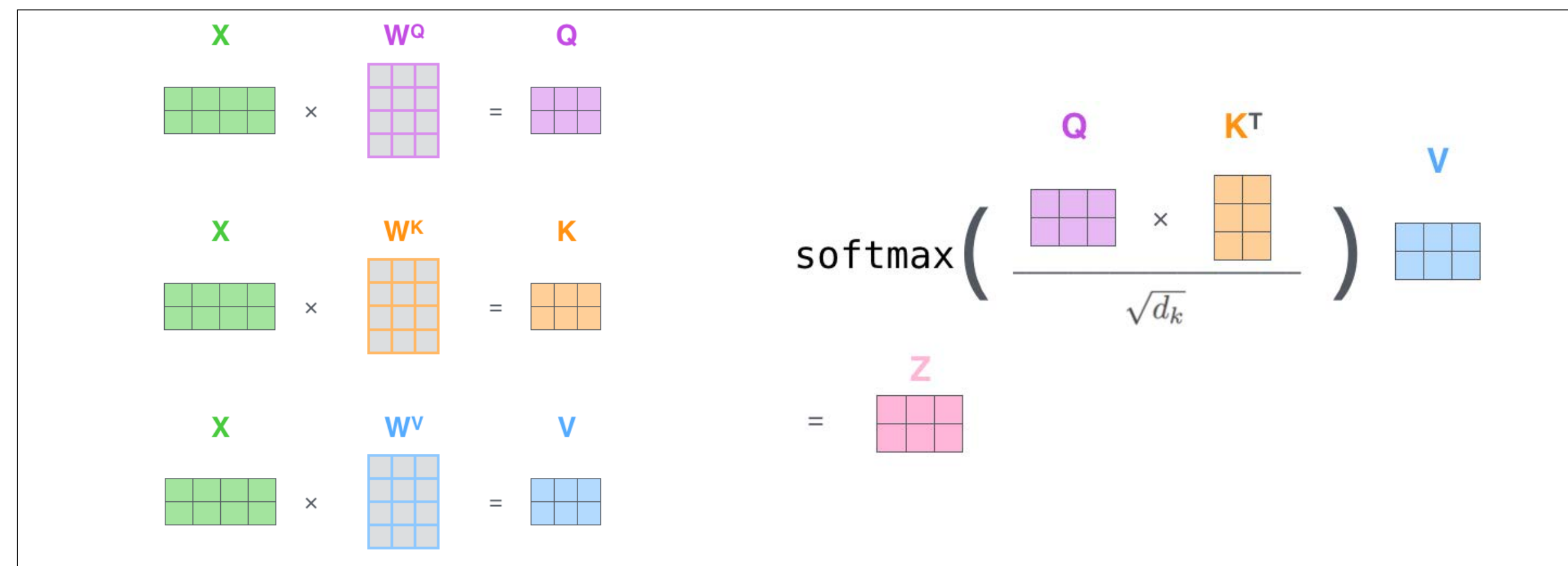
Astropart.Phys. 126 (2021) 102527



BEYOND PARTICLENET

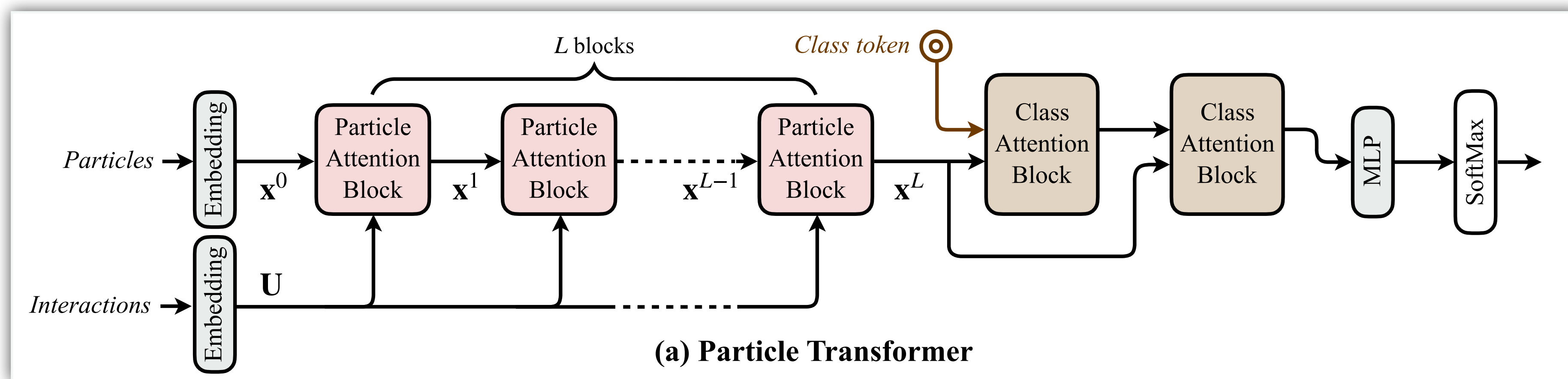
- **Transformers:** the new state-of-the-art architecture in ML – foundation of LLM like BERT/GPT

- core concept: self-attention mechanism

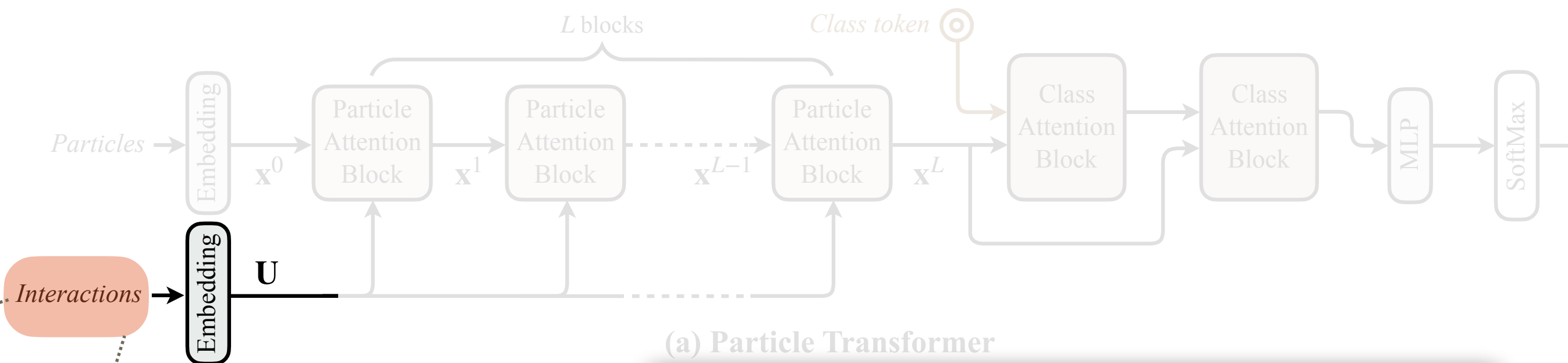
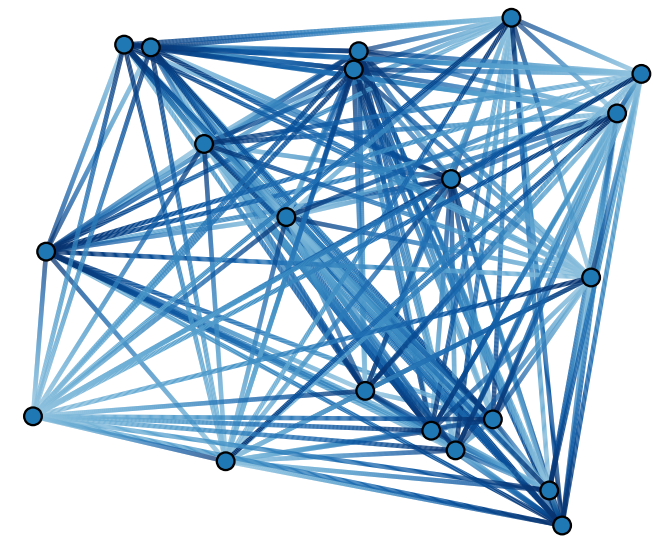


- **Particle Transformer (ParT):** Transformer model **tailored for particle physics**

HQ, C. Li, S. Qian,
ICML 2022



PARTICLE TRANSFORMER: ARCHITECTURE



HQ, C. Li, S. Qian,
ICML 2022

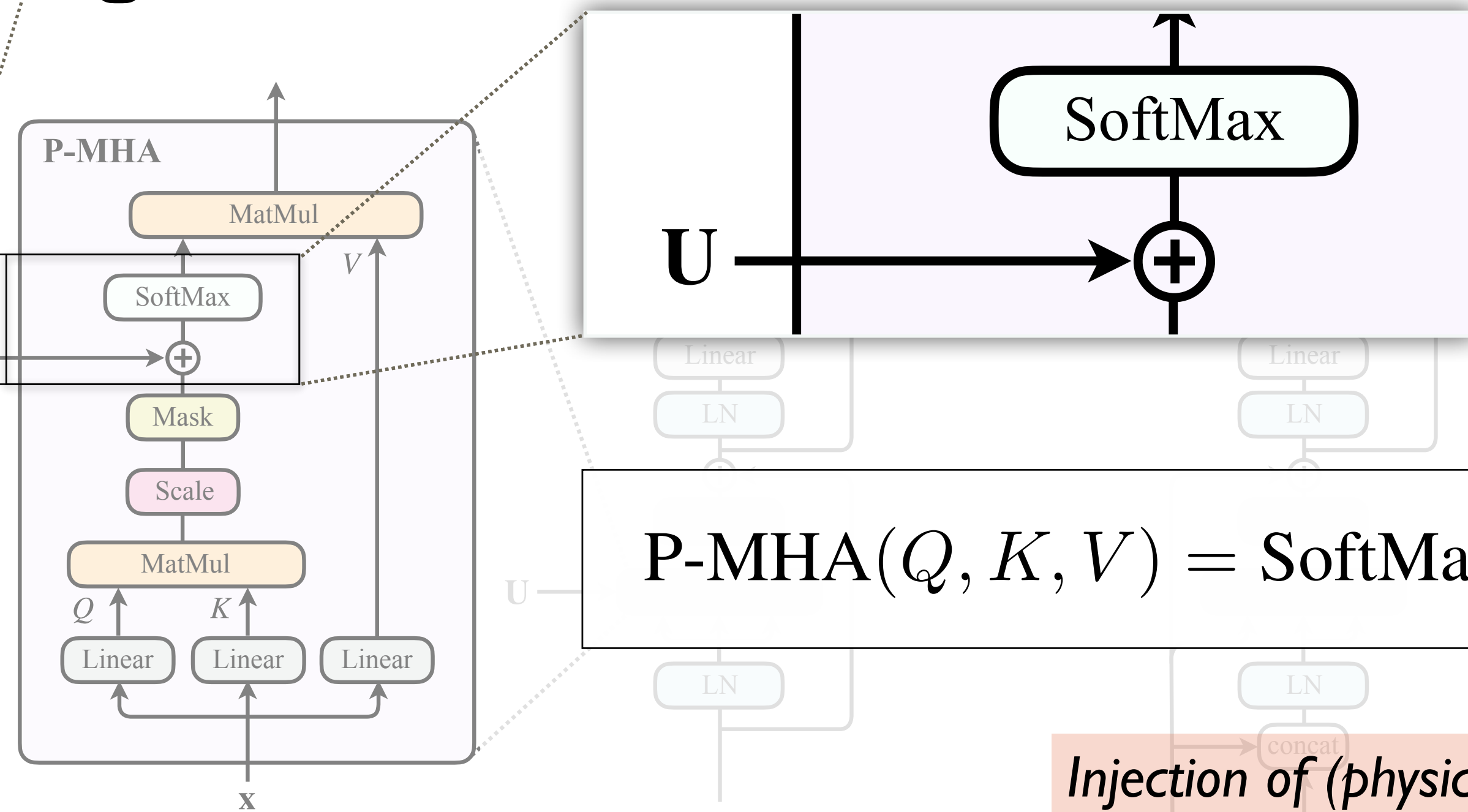
$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2},$$

$$k_T = \min(p_{T,a}, p_{T,b}) \Delta,$$

$$z = \min(p_{T,a}, p_{T,b}) / (p_{T,a} + p_{T,b}),$$

$$m^2 = (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2,$$

and many other possible pairwise features...



$$\text{P-MHA}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d_k} + \mathbf{U})V,$$

Injection of (physics-inspired) pairwise features to "bias" the dot-product self-attention

(b) Particle Attention Block

(c) Class Attention Block

PARTICLE TRANSFORMER: PERFORMANCE

	All classes		$H \rightarrow b\bar{b}$	$H \rightarrow c\bar{c}$	$H \rightarrow gg$	$H \rightarrow 4q$	$H \rightarrow \ell\nu qq'$	$t \rightarrow bqq'$	$t \rightarrow bl\nu$	$W \rightarrow qq'$	$Z \rightarrow q\bar{q}$
	Accuracy	AUC	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{99%}	Rej _{50%}	Rej _{99.5%}	Rej _{50%}	Rej _{50%}
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT	0.861	0.9877	10638	4149	123	1864	5479	32787	15873	543	402
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

JETCLASS dataset (100M jets)

- Particle Transformer (ParT): significant performance improvement!
 - compared to the existing state-of-the-art, ParticleNet
 - 1.7% increase in accuracy
 - up to 3x increase in background rejection (Rej_{X%})

$$\text{Rej}_{X\%} \equiv 1/\text{FPR at TPR} = X\%,$$

PARTICLE TRANSFORMER: PERFORMANCE

	All classes		$H \rightarrow b\bar{b}$	$H \rightarrow c\bar{c}$	$H \rightarrow gg$	$H \rightarrow 4q$	$H \rightarrow \ell\nu qq'$	$t \rightarrow bqq'$	$t \rightarrow bl\nu$	$W \rightarrow qq'$	$Z \rightarrow q\bar{q}$
	Accuracy	AUC	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{99%}	Rej _{50%}	Rej _{99.5%}	Rej _{50%}	Rej _{50%}
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT	0.861	0.9877	10638	4149	123	1864	5479	32787	15873	543	402
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

JETCLASS dataset (100M jets)

- Particle Transformer (ParT): significant performance improvement!
 - compared to the existing state-of-the-art, ParticleNet
 - 1.7% increase in accuracy
 - **up to 3x increase in background rejection (Rej_{x%})**
- ParT (plain): plain Transformer w/o interaction features
 - 1.2% drop in accuracy compared to full ParT
 - **Physics-driven modification of self-attention plays a key role!**

Model complexity

	Accuracy	# params	FLOPs
PFN	0.772	86.1 k	4.62 M
P-CNN	0.809	354 k	15.5 M
ParticleNet	0.844	370 k	540 M
ParT	0.861	2.14 M	340 M
ParT (plain)	0.849	2.13 M	260 M

PARTICLE TRANSFORMER: PRE-TRAINING + FINE-TUNING

- The large Transformer-based model enables new training paradigm
 - (supervised) pre-training on a large dataset (e.g., JETCLASS) & fine-tuning to downstream tasks
 - significantly outperforms existing models

Top quark tagging benchmark ($\sim 2M$ jets) [SciPost Phys. 7 (2019) 014]

	Accuracy	AUC	Rej _{50%}	Rej _{30%}
P-CNN	0.930	0.9803	201 ± 4	759 ± 24
PFN	—	0.9819	247 ± 3	888 ± 17
ParticleNet	0.940	0.9858	397 ± 7	1615 ± 93
JEDI-net (w/ $\sum O$)	0.930	0.9807	—	774.6
PCT	0.940	0.9855	392 ± 7	1533 ± 101
LGN	0.929	0.964	—	435 ± 95
rPCN	—	0.9845	364 ± 9	1642 ± 93
LorentzNet	0.942	0.9868	498 ± 18	2195 ± 173
ParT	0.940	0.9858	413 ± 16	1602 ± 81
ParticleNet-f.t.	0.942	0.9866	487 ± 9	1771 ± 80
ParT-f.t.	0.944	0.9877	691 ± 15	2766 ± 130

Quark-gluon tagging benchmark ($\sim 2M$ jets) [JHEP 01 (2019) 121]

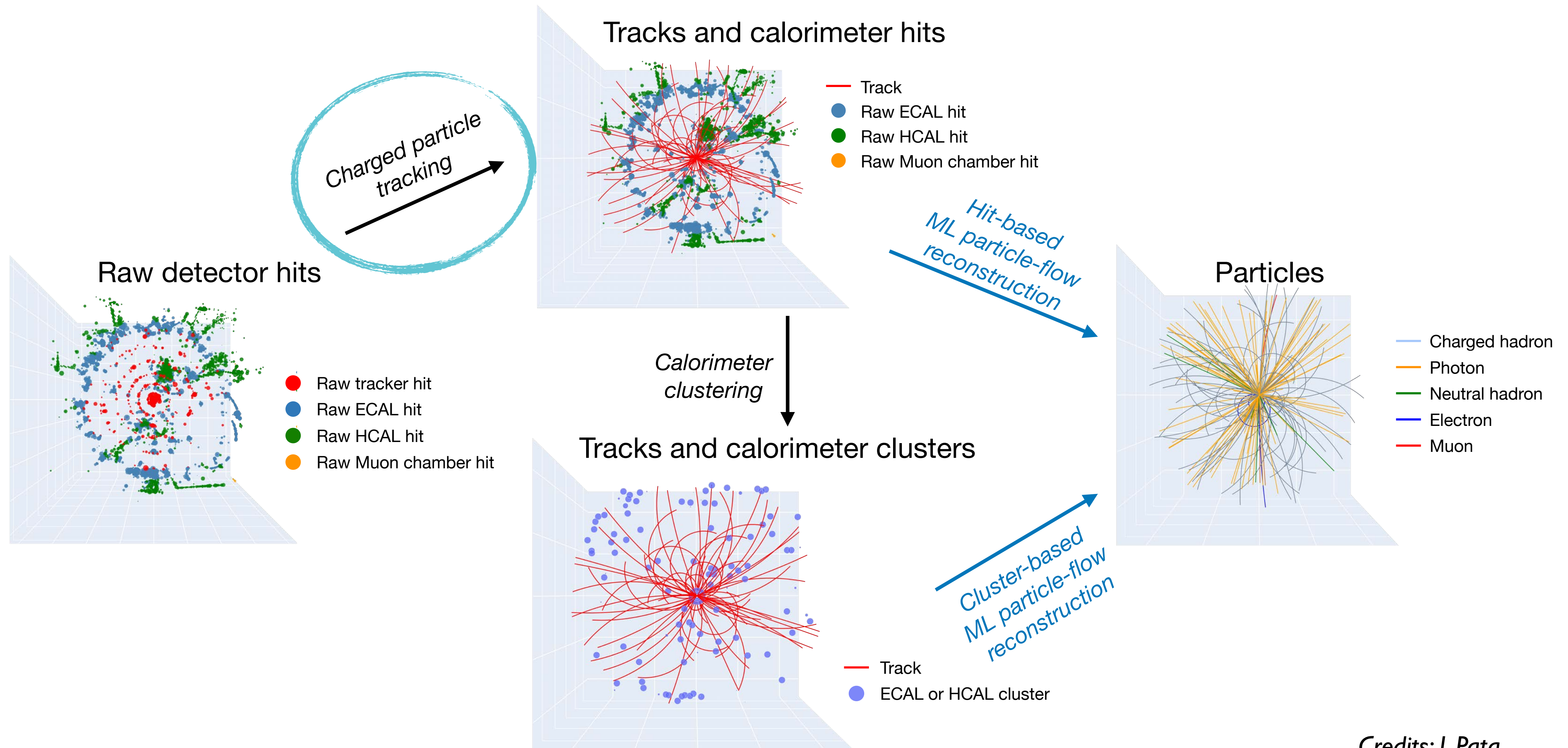
	Accuracy	AUC	Rej _{50%}	Rej _{30%}
P-CNN _{exp}	0.827	0.9002	34.7	91.0
PFN _{exp}	—	0.9005	34.7 ± 0.4	—
ParticleNet _{exp}	0.840	0.9116	39.8 ± 0.2	98.6 ± 1.3
rPCN _{exp}	—	0.9081	38.6 ± 0.5	—
ParT _{exp}	0.840	0.9121	41.3 ± 0.3	101.2 ± 1.1
ParticleNet-f.t. _{exp}	0.839	0.9115	40.1 ± 0.2	100.3 ± 1.0
ParT-f.t._{exp}	0.843	0.9151	42.4 ± 0.2	107.9 ± 0.5
PFN _{full}	—	0.9052	37.4 ± 0.7	—
ABCNet _{full}	0.840	0.9126	42.6 ± 0.4	118.4 ± 1.5
PCT _{full}	0.841	0.9140	43.2 ± 0.7	118.0 ± 2.2
LorentzNet _{full}	0.844	0.9156	42.4 ± 0.4	110.2 ± 1.3
ParT _{full}	0.849	0.9203	47.9 ± 0.5	129.5 ± 0.9
ParT-f.t._{full}	0.852	0.9230	50.6 ± 0.2	138.7 ± 1.3

RECONSTRUCTION



*How to build physics objects
from low-level detector information?*

RECONSTRUCTION

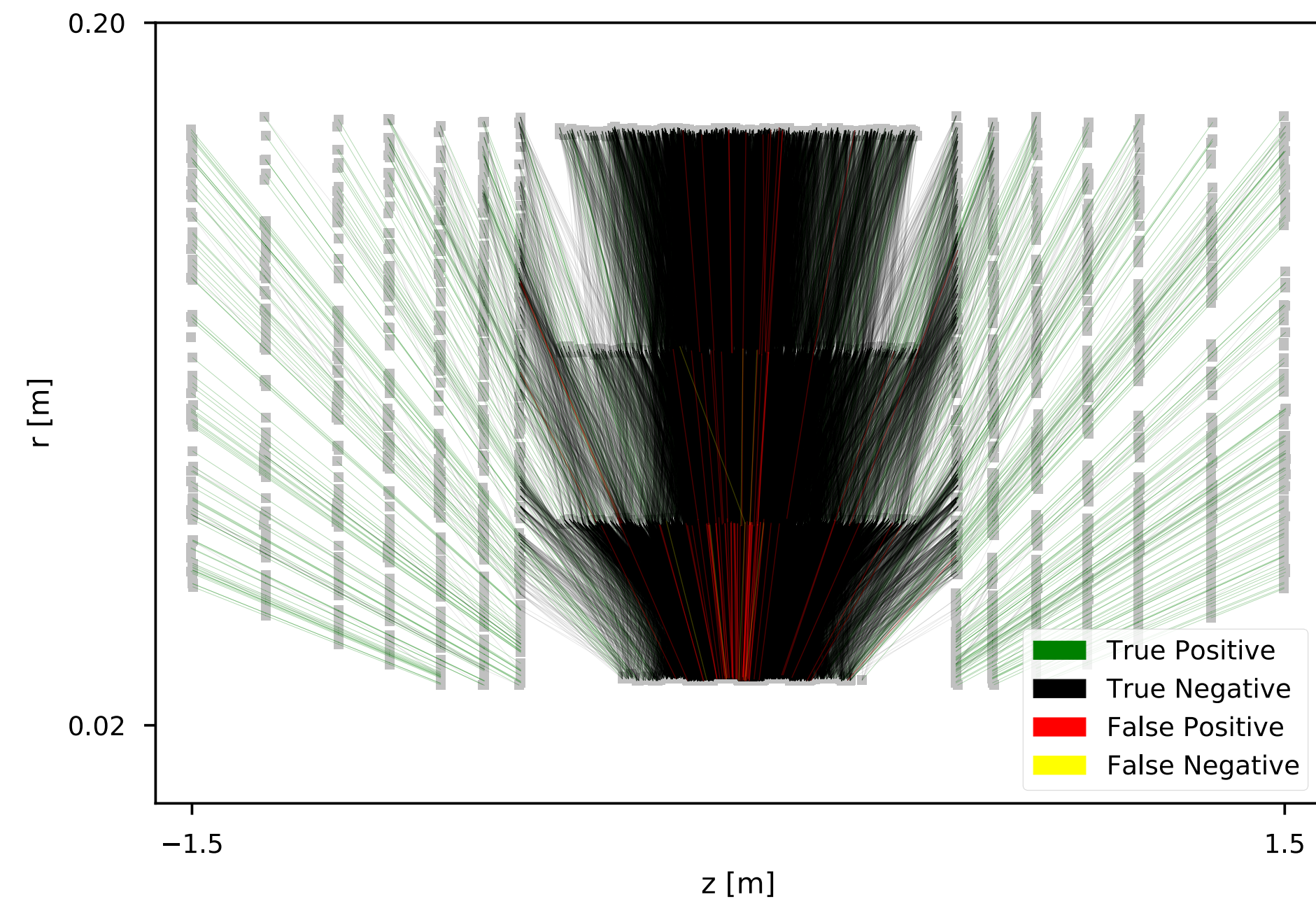
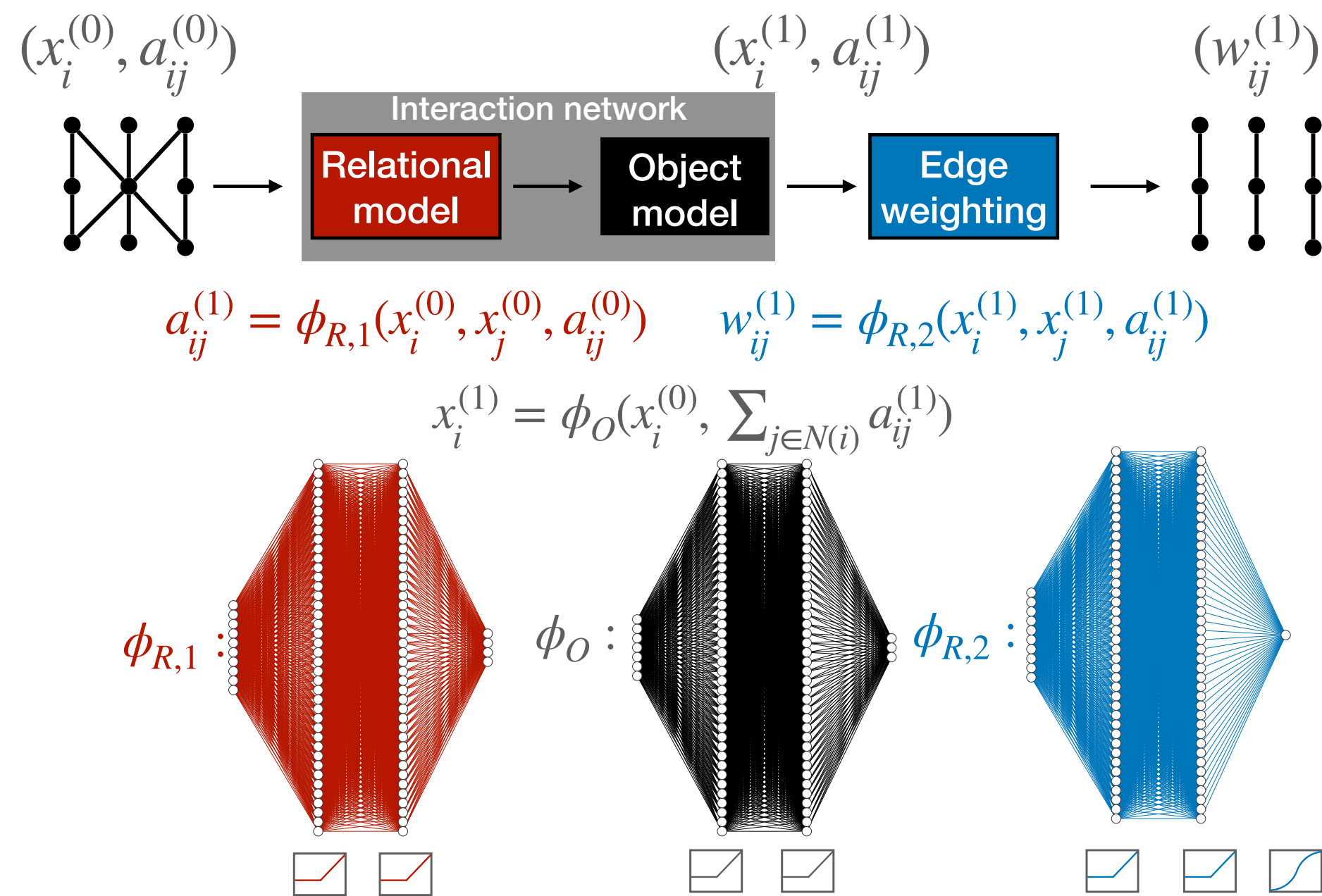


Credits: J. Pata

CHARGED PARTICLE TRACKING

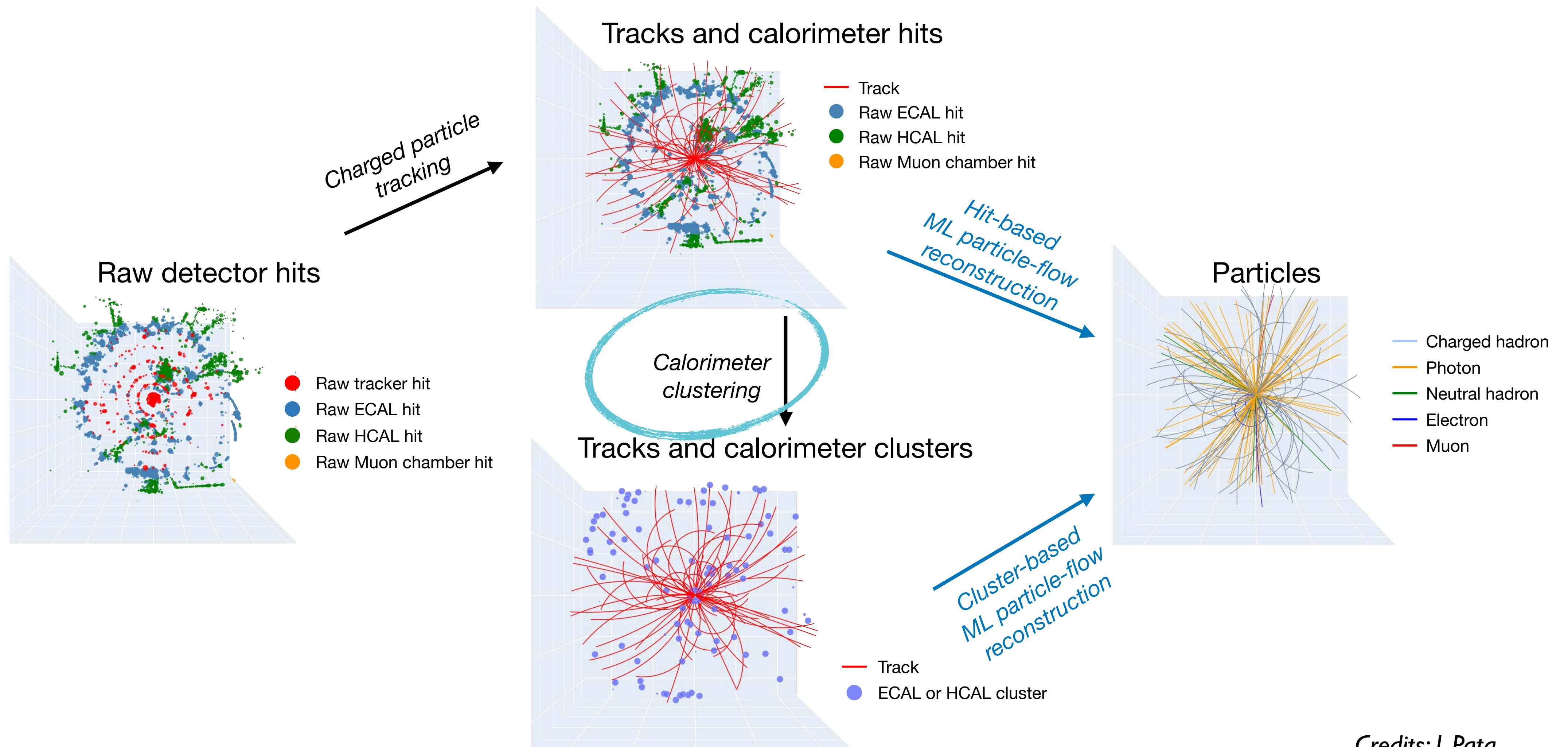
- Charged particle tracking via **edge classification** with GNNs
 - each hit is a node of the graph
 - edges constructed between pairs of hits with geometrically plausible relations
 - classify whether each edge connects hits belonging to the same track or not

G. DeZoort et al.
 [Comput. Softw. Big Sci. 5, 26 (2021)]



See also: S. Farrell et al. [1810.06111]; X. Ju et al. [2003.11603];
 C. Biscarat, S. Caillou, C. Rougier, J. Stark and J. Zahreddine [2103.00916]; X. Ju et al. [2103.06995]; etc.

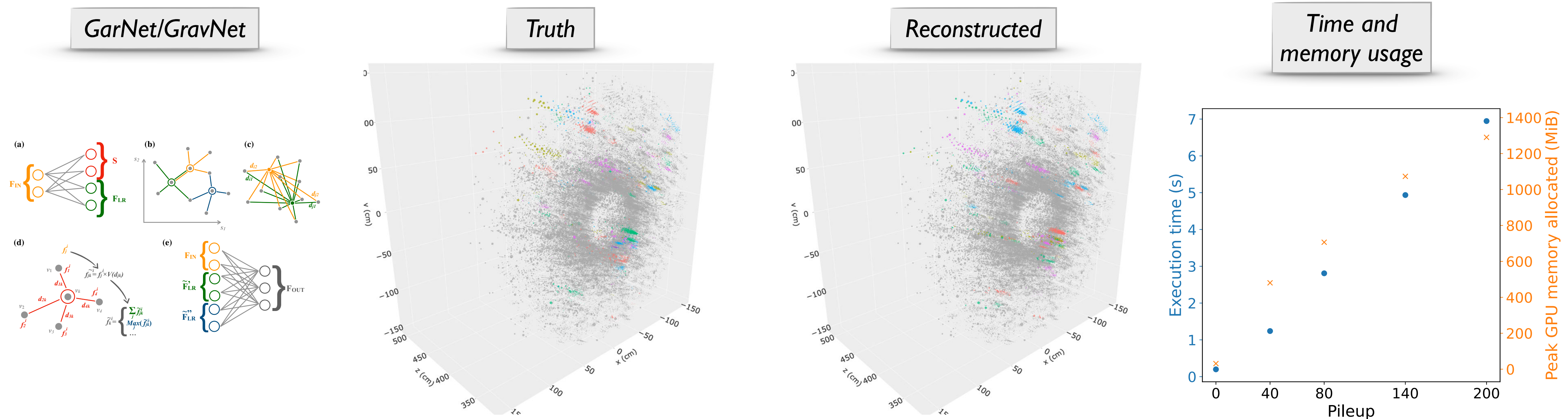
RECONSTRUCTION



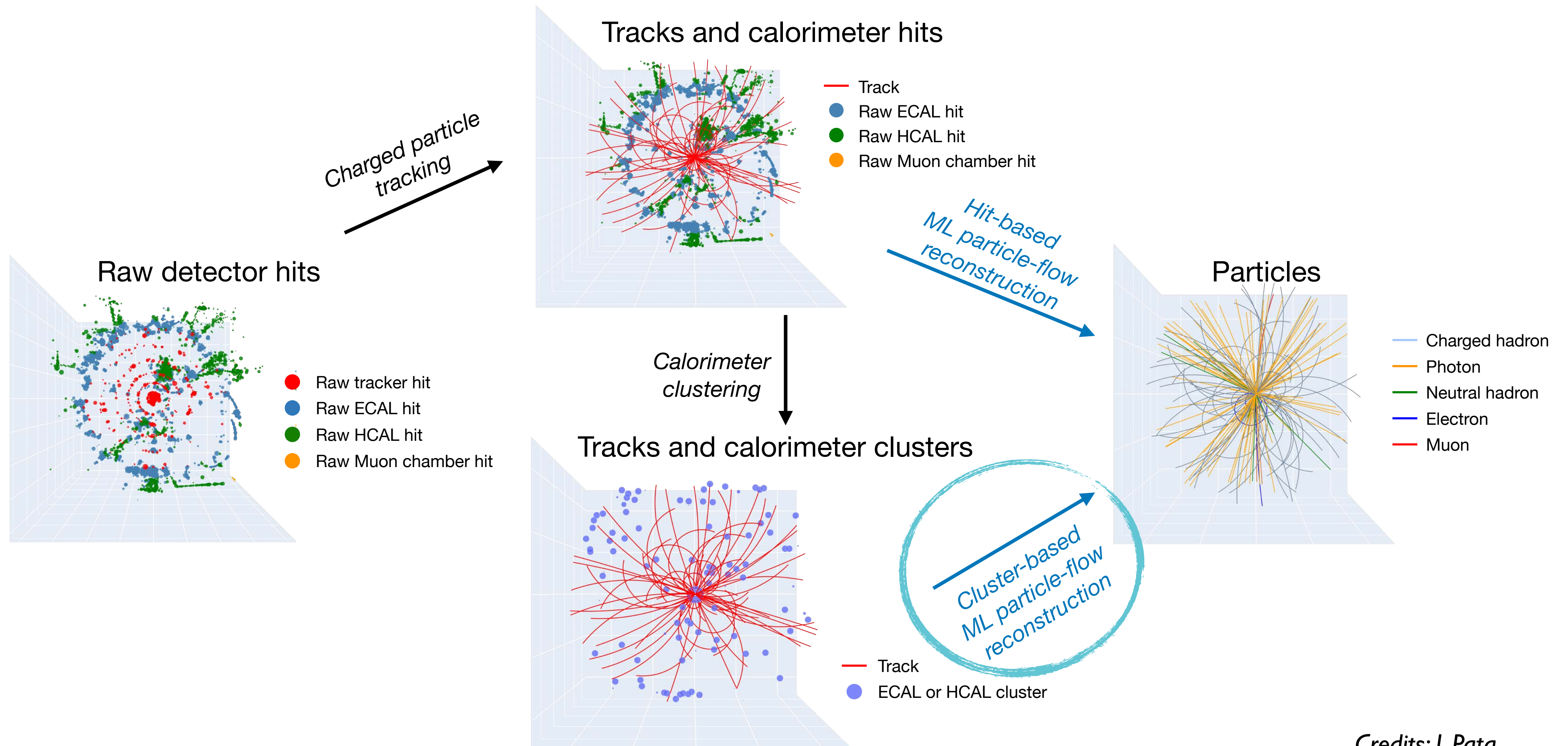
Credits: J. Pata

CALORIMETER CLUSTERING

- End-to-end reconstruction for a high granularity calorimeter
- Object condensation: one-stage multi-object reconstruction
 - supervised clustering of hits belonging to a shower to a “condensation point” by using attractive/repulsive potentials in the loss
 - simultaneously predict the number of showers and their properties



RECONSTRUCTION

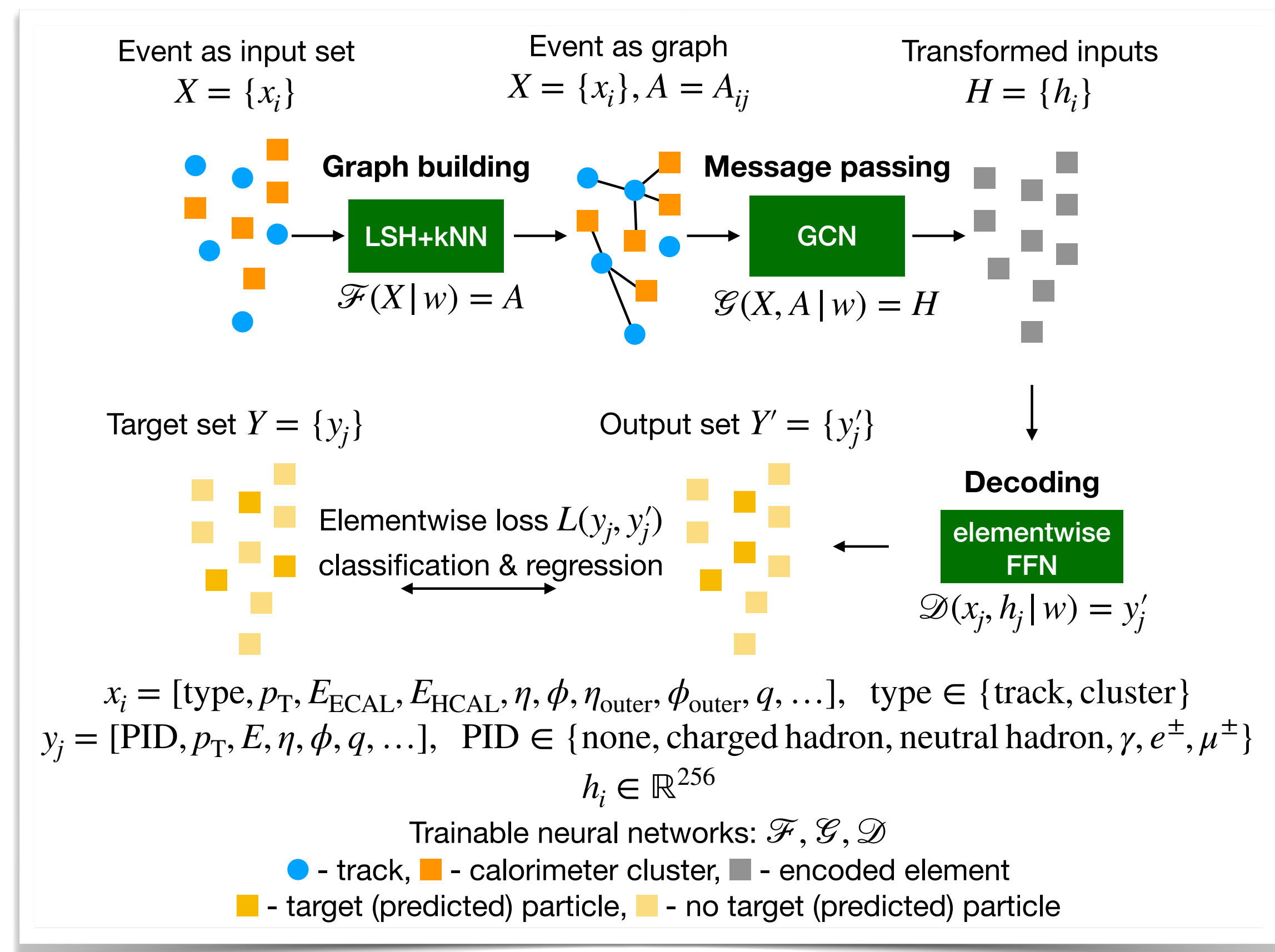
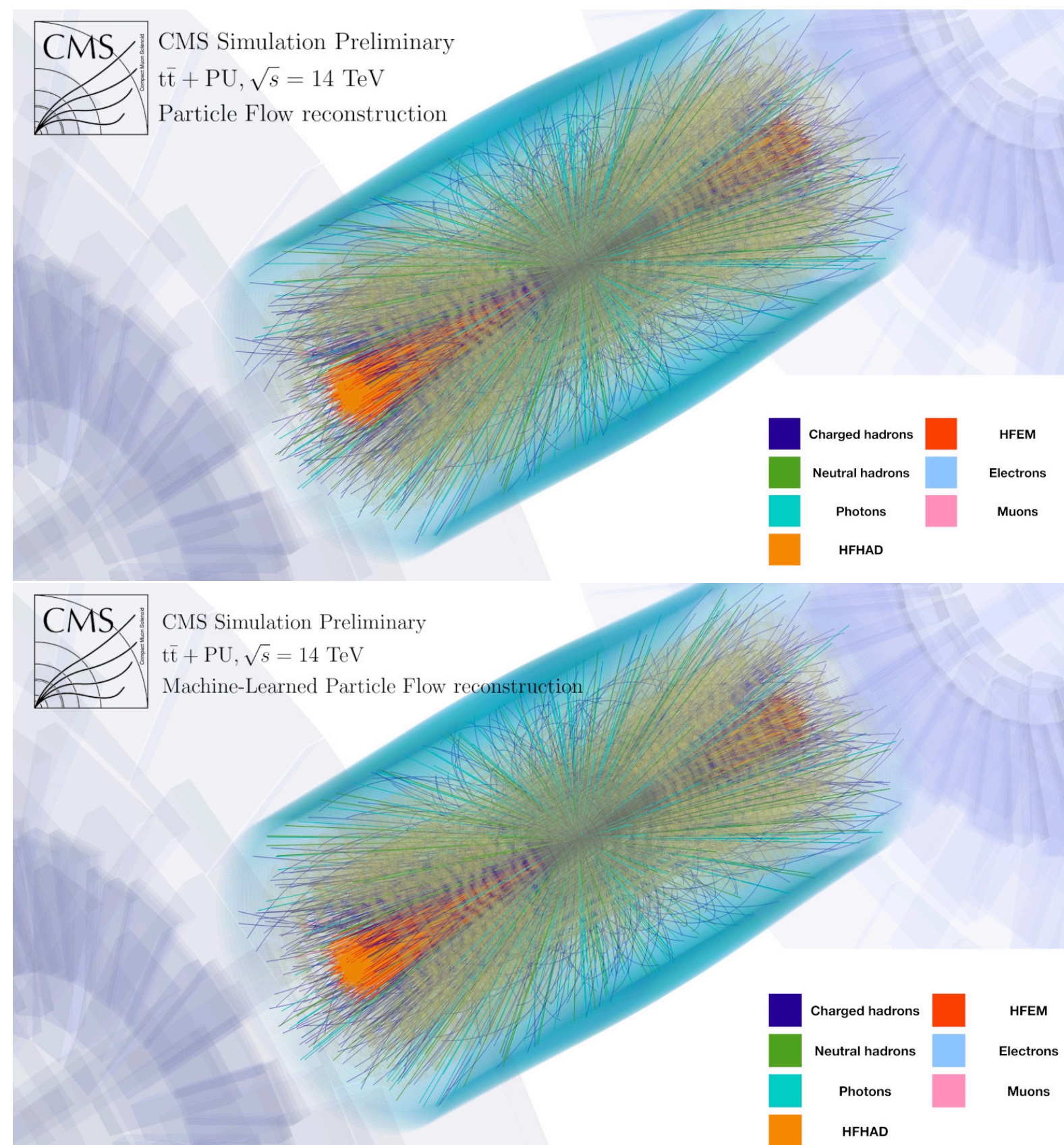


Credits: J. Pata

PARTICLE-FLOW RECONSTRUCTION: MLPF

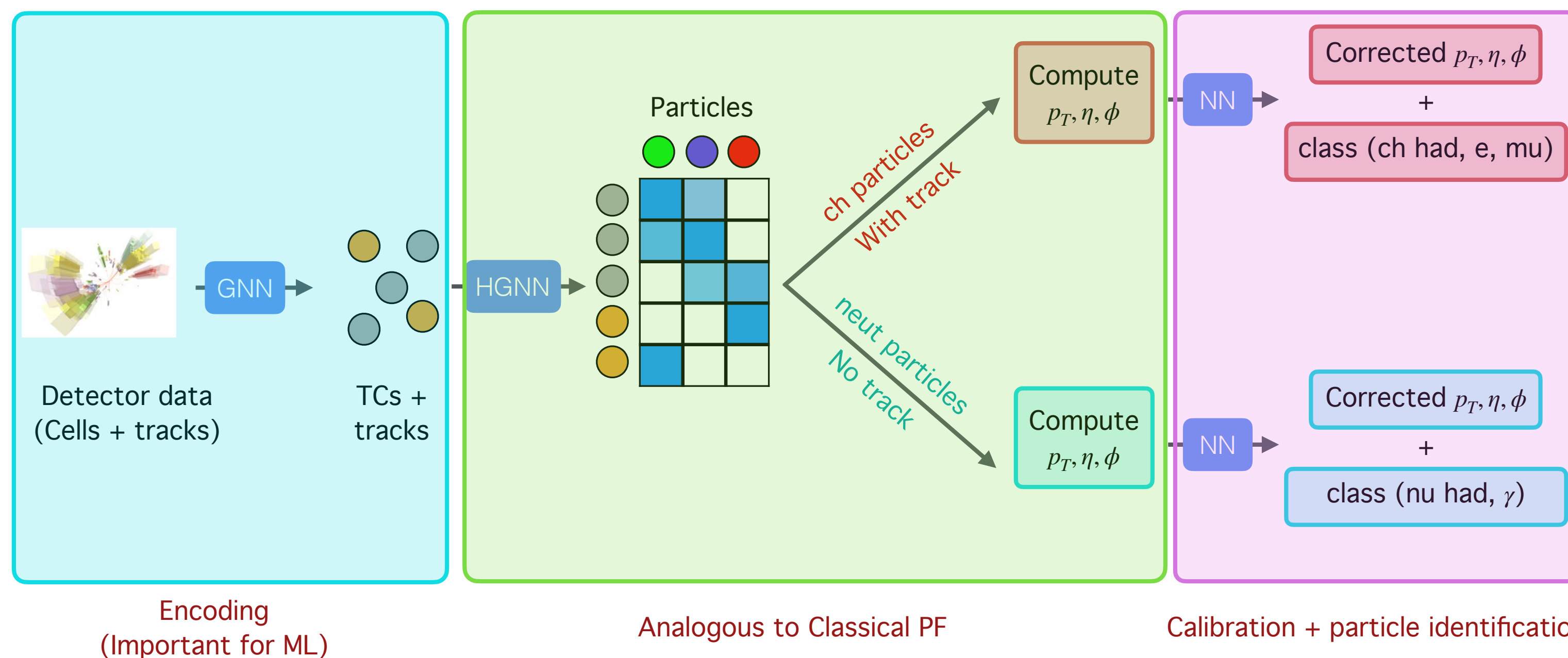
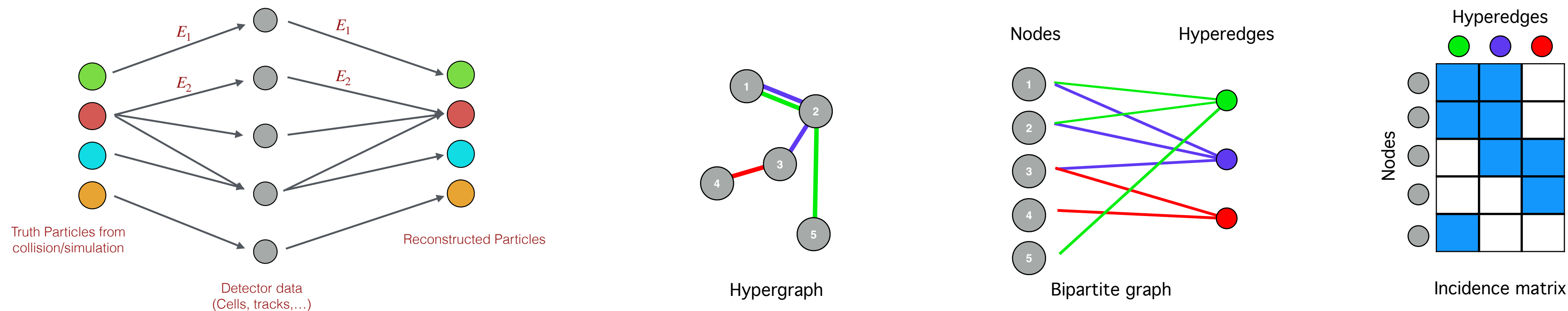
- Global particle-flow reconstruction via node classification and regression using GNNs
 - comparable performance to the baseline rule-based PF algorithm at CMS
 - runtime scales linearly with input size, no quartic explosion

*J. Pata et al.,
ACAT 2021, 2203.00330*



PARTICLE-FLOW RECONSTRUCTION: HGPFLOW

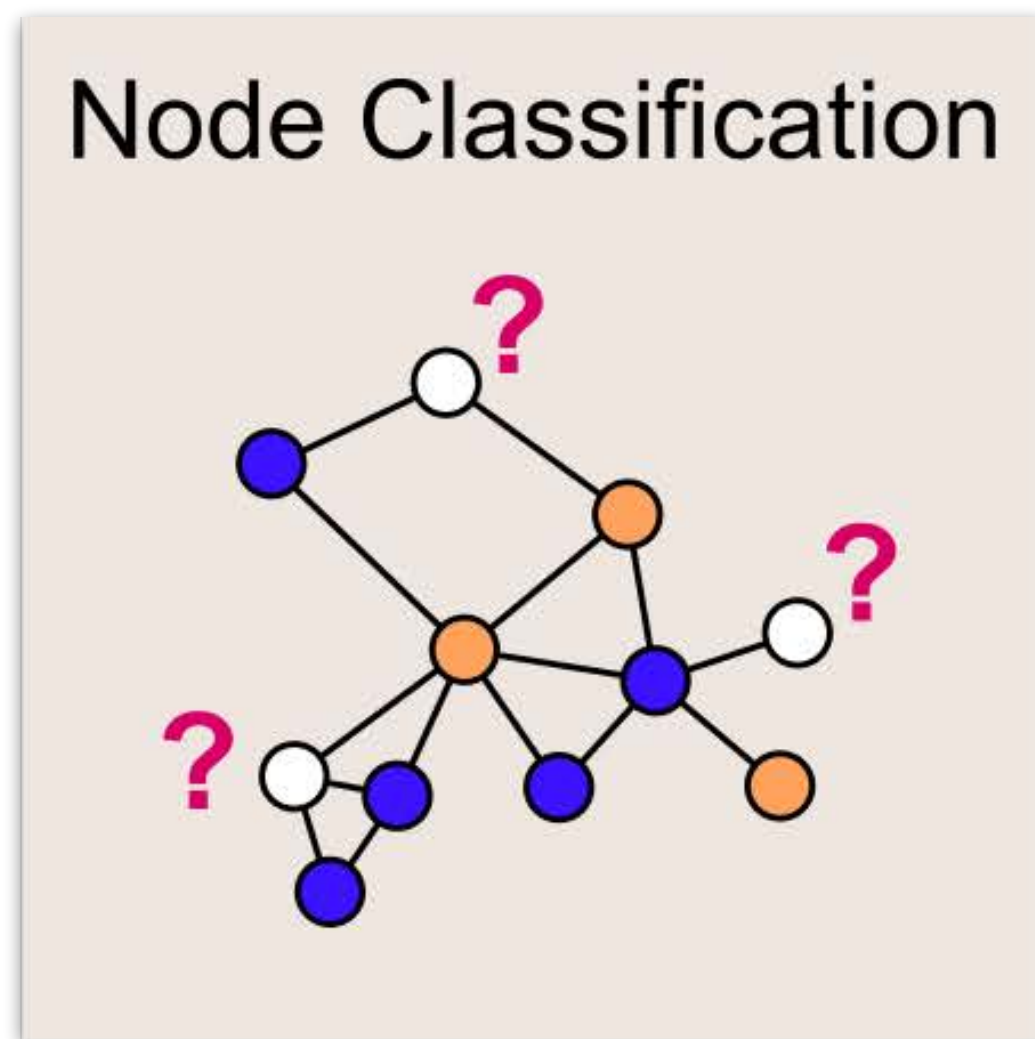
- What if multiple clusters should be associated to multiple particles? ==> Hypergraph



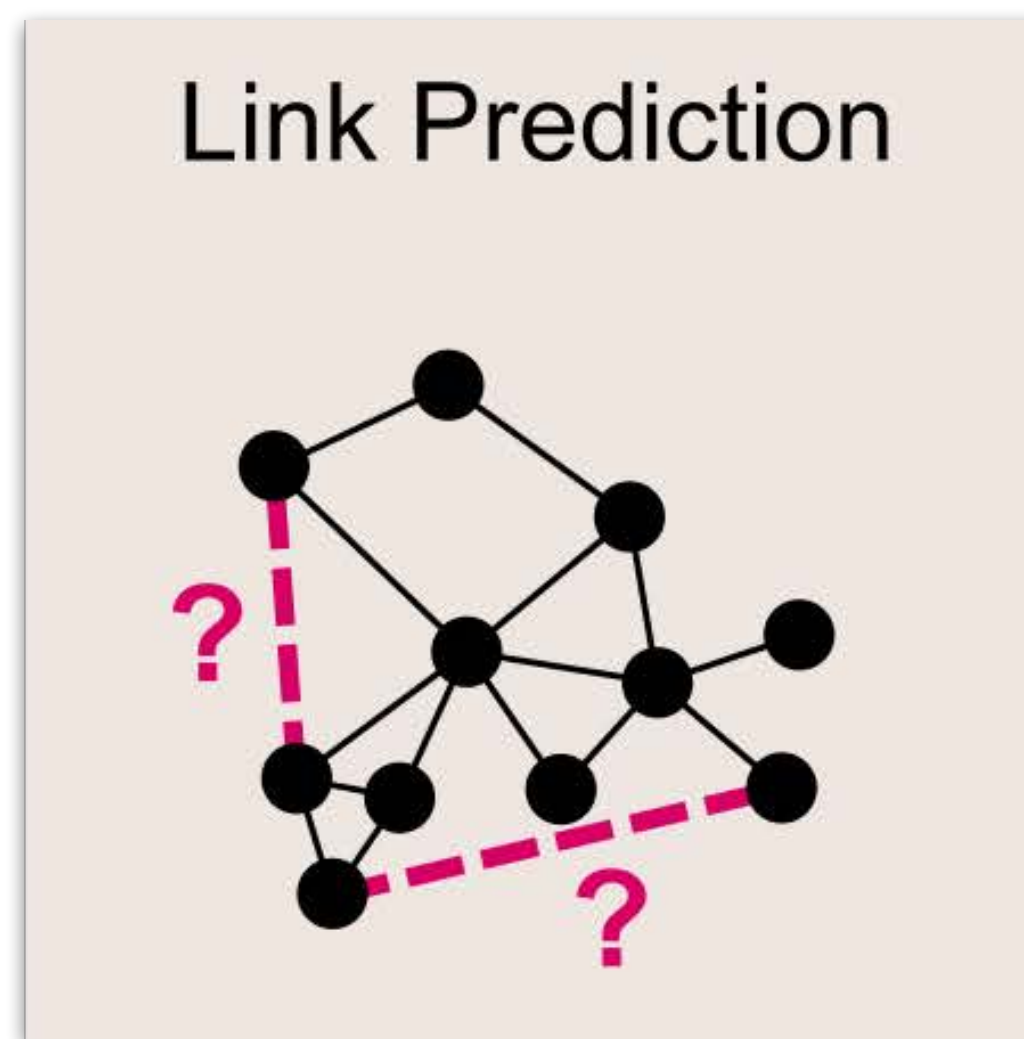
See talk by N. Kakati at ML4jets2023

GNNs FOR RECONSTRUCTION

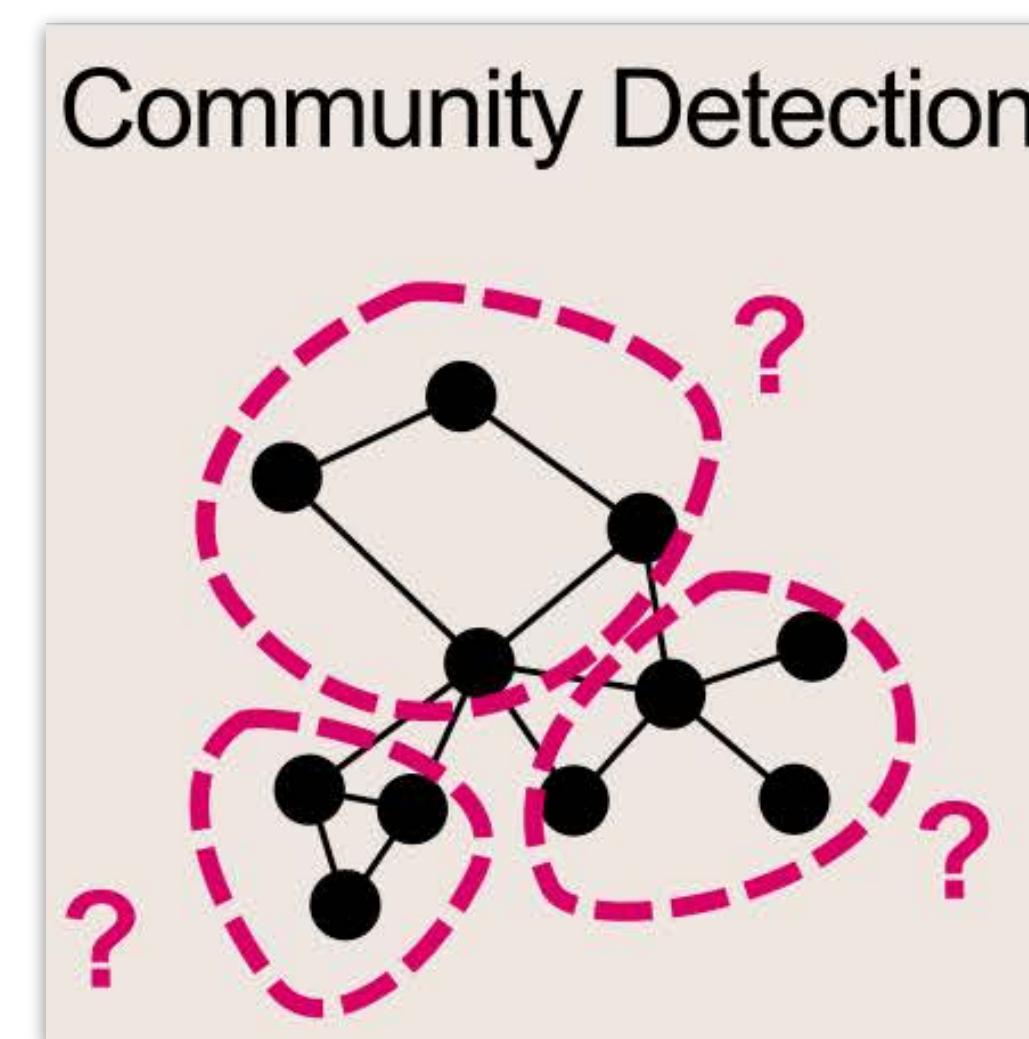
- Almost all ML reconstruction models are based on point clouds and using graph neural networks
 - however, with varying approaches for different tasks



MLPF



*Tracking,
HGPflow*



*Calo clustering
(object condensation)*

THE ROAD AHEAD

THE ROAD AHEAD

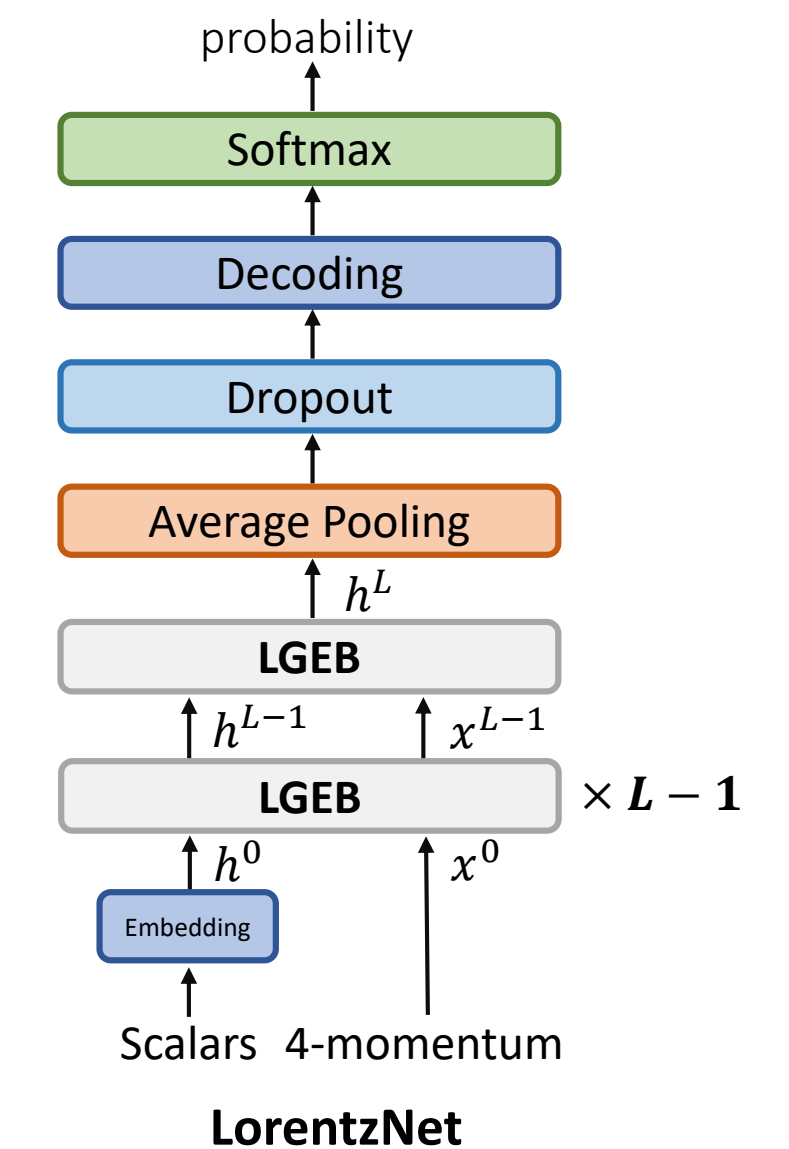
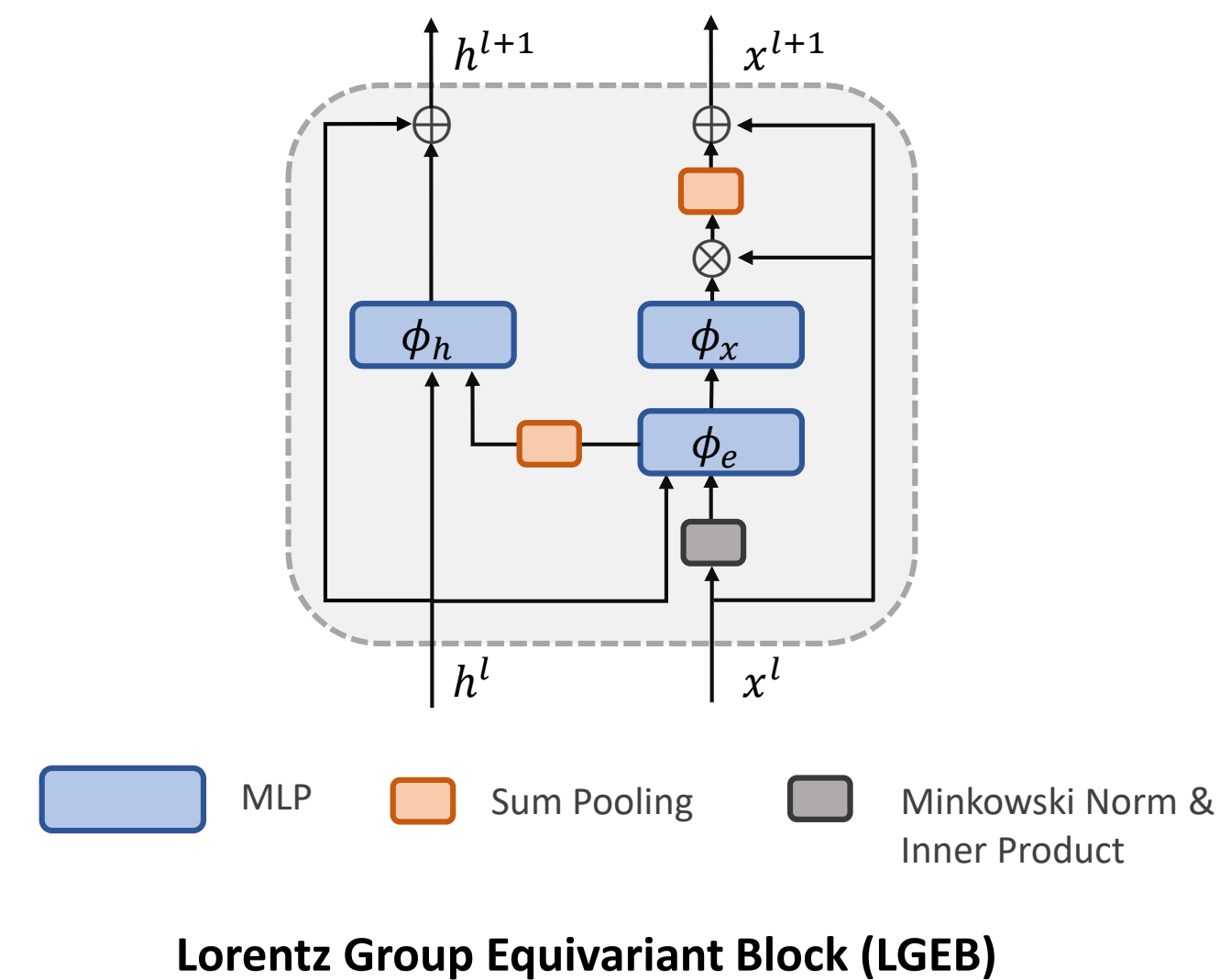
- Can we better incorporate physics knowledge into the network design?
 - physics aware data representation, symmetry group equivariant architecture, ...

LORENTZNET

- Incorporating Lorentz symmetry into graph neural network architecture

S. Gong, Q. Meng, J. Zhang, HQ, C. Li, S. Qian,
W. Du, Z. M. Ma and T.Y. Liu,
JHEP 07 (2022) 030

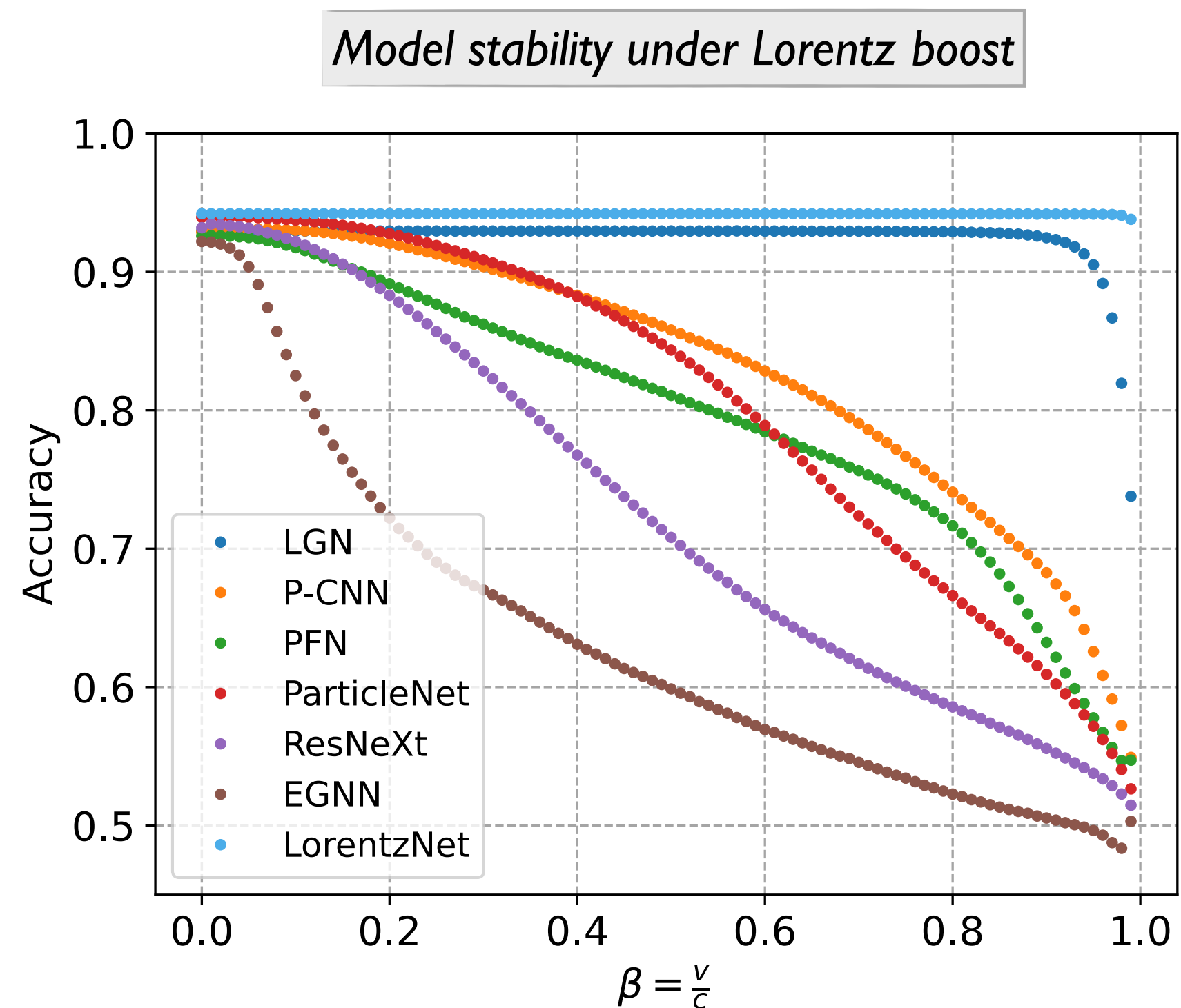
Coordinate input:	x^0	Lorentz 4-vector Lorentz scalar
Feature input:	h_i^0	
Message:	$m_{ij}^l = \phi_e \left(\underbrace{h_i^l, h_j^l}_{\text{Scalars}}, \underbrace{\psi(\ x_i^l - x_j^l\ ^2), \psi(\langle x_i^l, x_j^l \rangle)}_{\text{Pairwise Lorentz invariants}} \right)$	
Coordinate update:	$x_i^{l+1} = x_i^l + c \sum_{j \in [N]} \phi_x(m_{ij}^l) \cdot x_j^l$	
Feature update:	$h_i^{l+1} = h_i^l + \phi_h \left(h_i^l, \sum_{j \in [N]} w_{ij} m_{ij}^l \right)$	



cf. A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller and R. Kondor, [arXiv: 2006.04780](https://arxiv.org/abs/2006.04780) ["LGN"];
A. Bogatskiy, T. Hoffman, D. W. Miller, J. T. Offermann and X. Liu, [2307.16506](https://arxiv.org/abs/2307.16506) ["PELICAN"];
I. Batatia, M. Geiger, J. Munoz, T. Smidt, L. Silberman and C. Ortner, [arXiv: 2306.00091](https://arxiv.org/abs/2306.00091) ["lie-nn"];

LORENTZNET: BENEFITS FROM SYMMETRY

- Benefits from the symmetry preservation
 - model response invariant under Lorentz transformation
 - sample efficiency: incorporation of Lorentz symmetry allows to train with very few samples



Performance when trained on a fraction of the top-tagging dataset

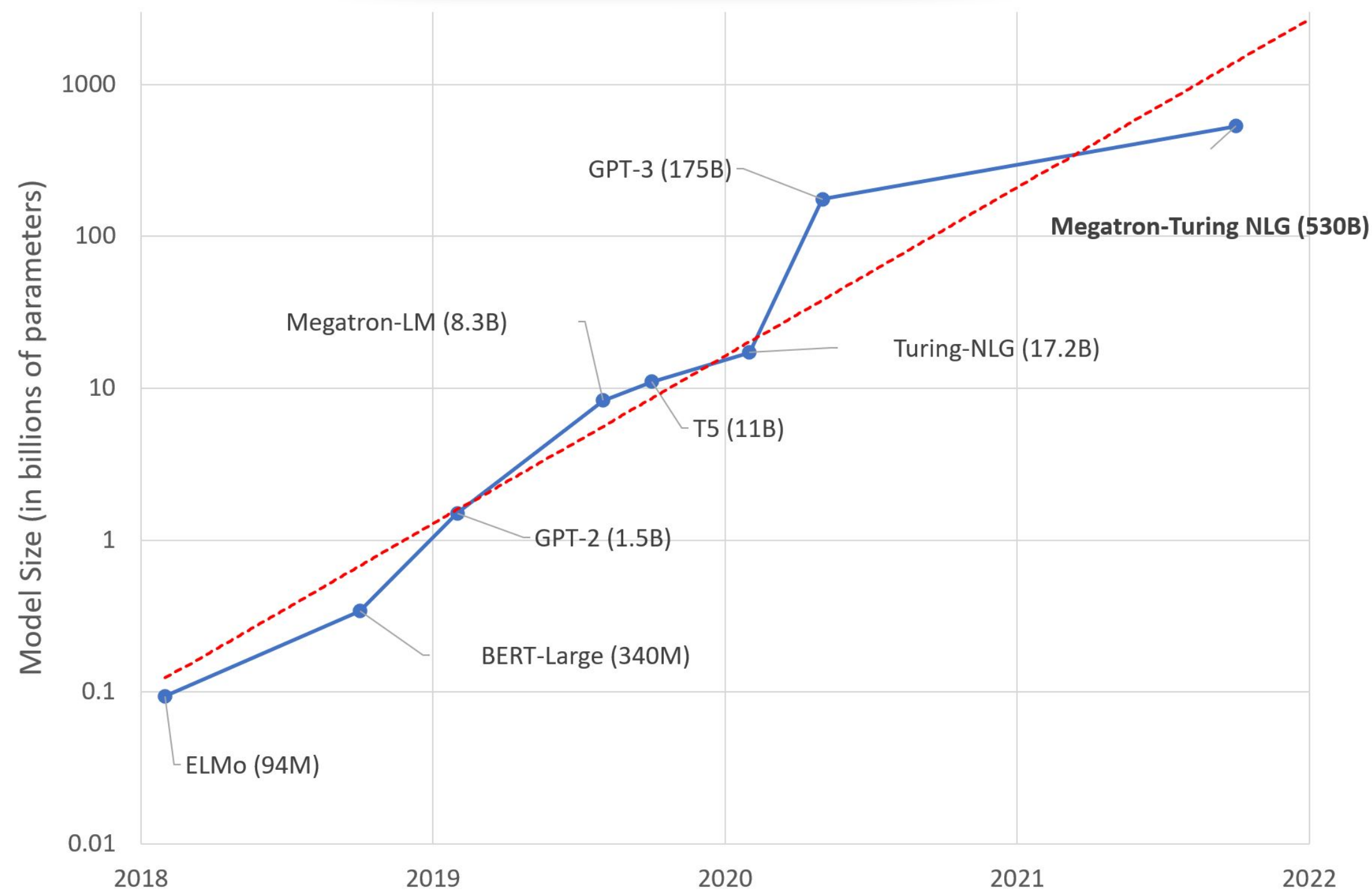
Training Fraction	Model	Accuracy	AUC	$1/\varepsilon_B$ ($\varepsilon_S = 0.5$)	$1/\varepsilon_B$ ($\varepsilon_S = 0.3$)
0.5% (~6k jets)	ParticleNet	0.913	0.9687	77 ± 4	199 ± 14
	LorentzNet	0.929	0.9793	176 ± 14	562 ± 72
1%	ParticleNet	0.919	0.9734	103 ± 5	287 ± 19
	LorentzNet	0.932	0.9812	209 ± 5	697 ± 58
5%	ParticleNet	0.931	0.9807	195 ± 4	609 ± 35
	LorentzNet	0.937	0.9839	293 ± 12	1108 ± 84

THE ROAD AHEAD

- Can we better incorporate physics knowledge into the network design?
 - physics aware data representation, symmetry group equivariant architecture, ...
- Can we scale up to a large model for HEP?
 - large datasets, pre-training, multi-modal learning, ...

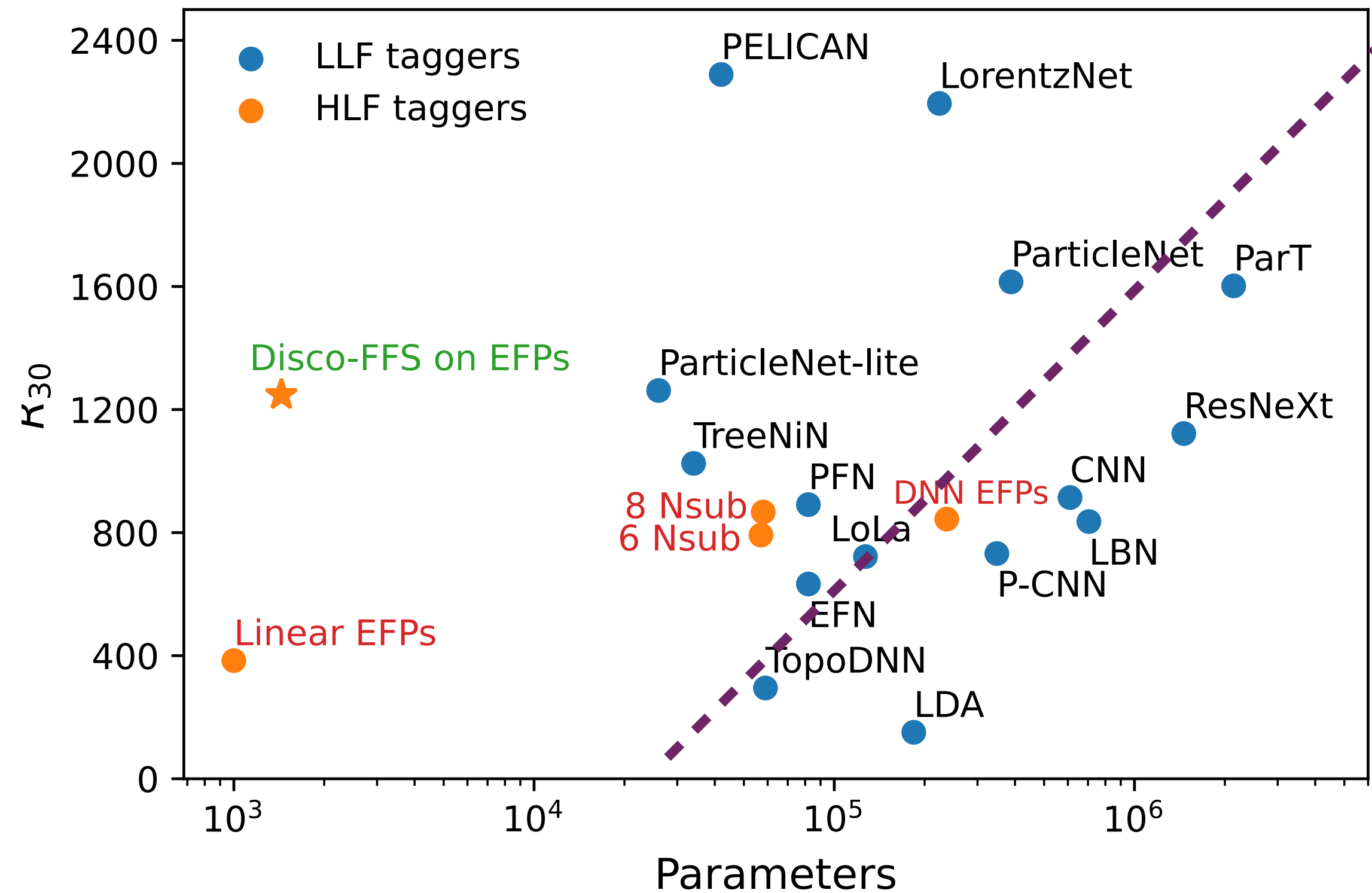
LARGE PHYSICS MODEL?

Natural language models



<https://huggingface.co/blog/large-language-models>

HEP models (jet tagging)



R. Das, G. Kasieczka and D. Shih, arXiv: 2212.00046

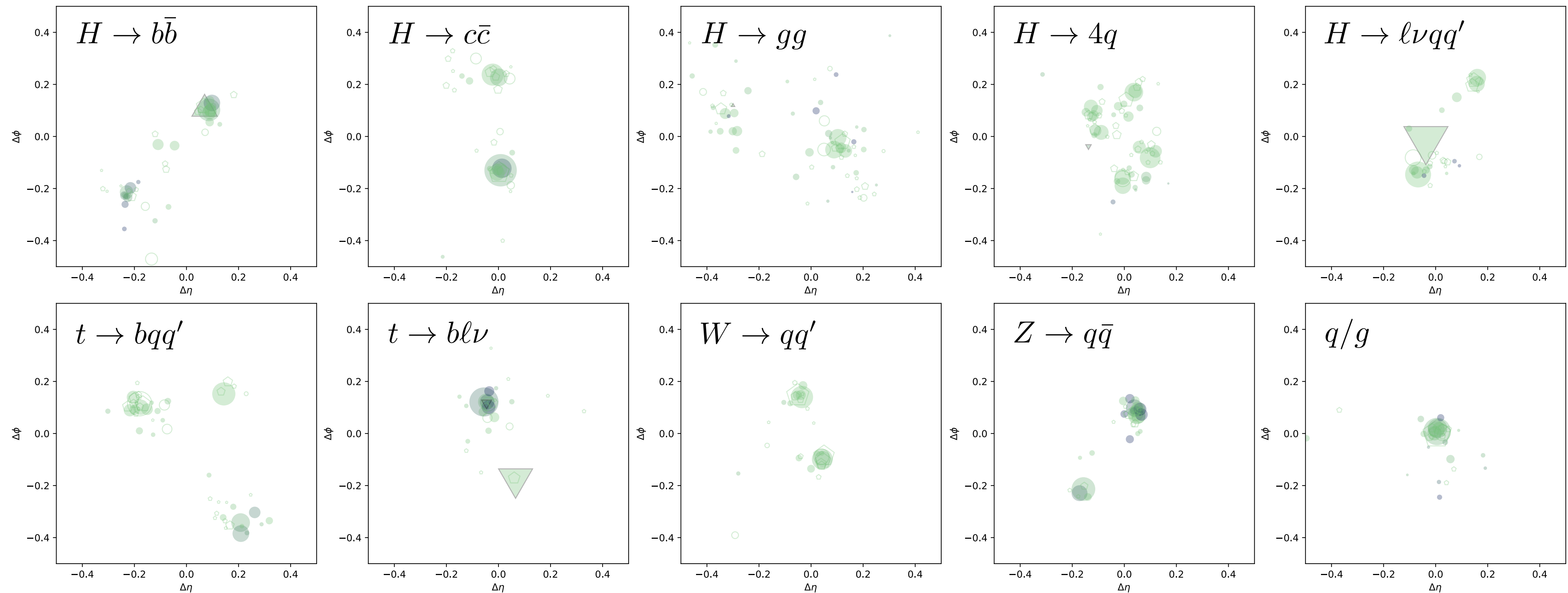
**Large Language Models (like GPT) has transformed NLP.
What if a Large Physics Model?**

A FIRST STEP



HQ, C. Li, S. Qian,
ICML 2022

- **JETCLASS**: a new large and comprehensive jet simulation dataset
- 100M jets in 10 classes: ~two orders of magnitude larger than existing public datasets



We invite the community to explore and experiment with this dataset and extend the boundary of deep learning and HEP even further.

THE ROAD AHEAD

- Can we better incorporate physics knowledge into the network design?
 - physics aware data representation, symmetry group equivariant architecture, ...
- Can we scale up to a large model for HEP?
 - large datasets, pre-training, multi-modal learning, ...
- Can we improve the computational efficiency of GNNs?
 - emerging specialized libraries for GNN training and inference (PyG, DGL, TF-GNN, ...)
 - accelerated inference on specialized ASICs / FPGAs (e.g., for triggering), software hardware co-design, ...
- Can we improve the robustness of GNNs (e.g., data/simulation difference)?
 - domain adaption? calibration? uncertainty aware training? ...
- Can we improve the interpretability and explainability of GNNs?

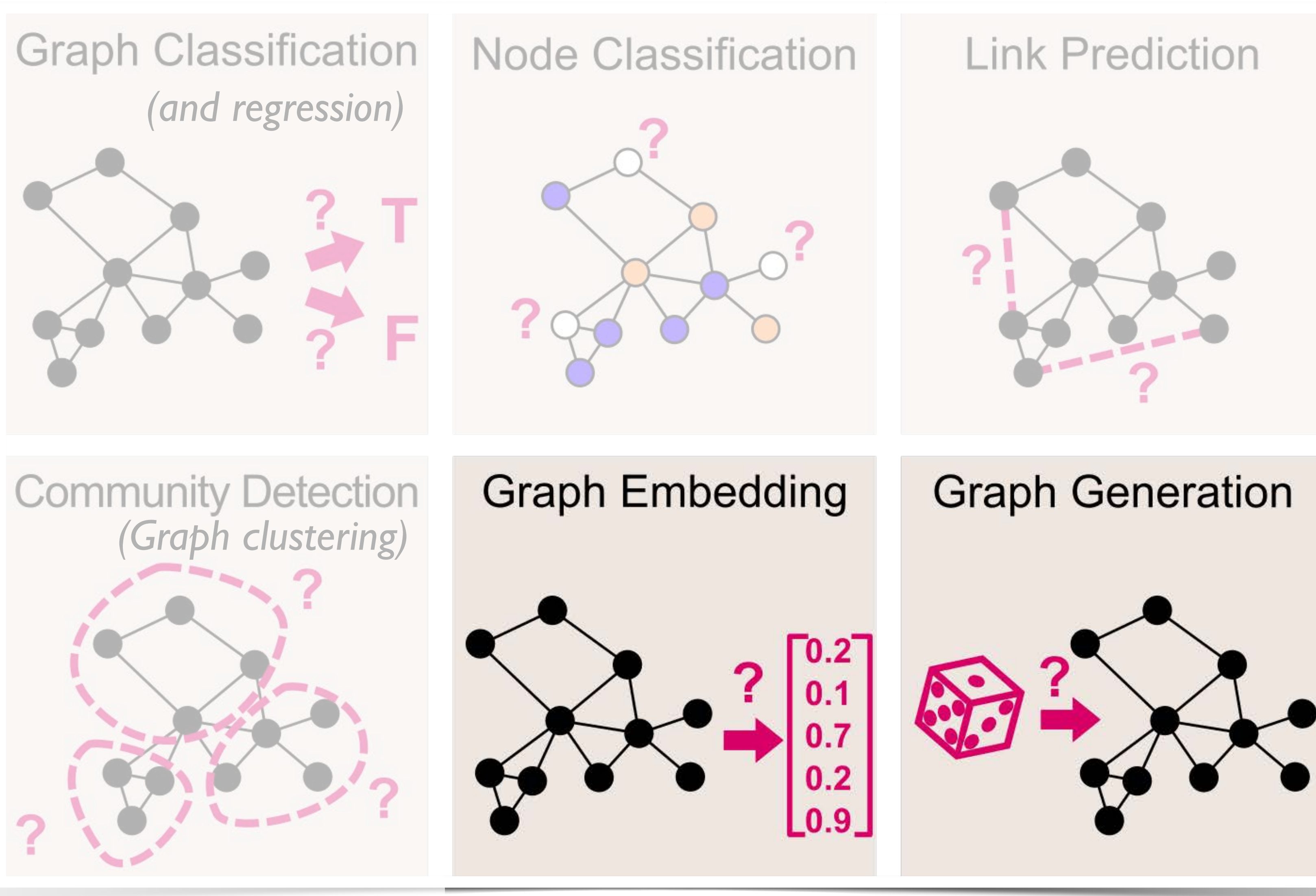
THE ROAD AHEAD

- Can we better incorporate physics knowledge into the network design?
 - physics aware data representation, symmetry group equivariant architecture, ...
- Can we scale up to a large model for HEP?
 - large datasets, pre-training, multi-modal learning, ...
- Can we improve GNN training and inference?
 - emerging specialized hardware for GNN training and inference (PyG, DGL, TF-GNN, ...)
 - accelerated inference on specialized ASICs / FPGAs (e.g., for triggering), software hardware co-design, ...
- Can we improve the robustness of GNNs (e.g., data/simulation difference)?
 - domain adaption? calibration? uncertainty aware training? ...
- Can we improve the interpretability and explainability of GNNs?

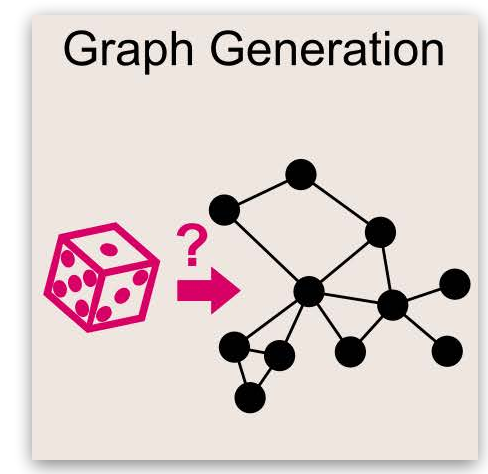
Your innovation and creativity can make a big difference!

EXTRAS

GRAPH GENERATIVE MODELS

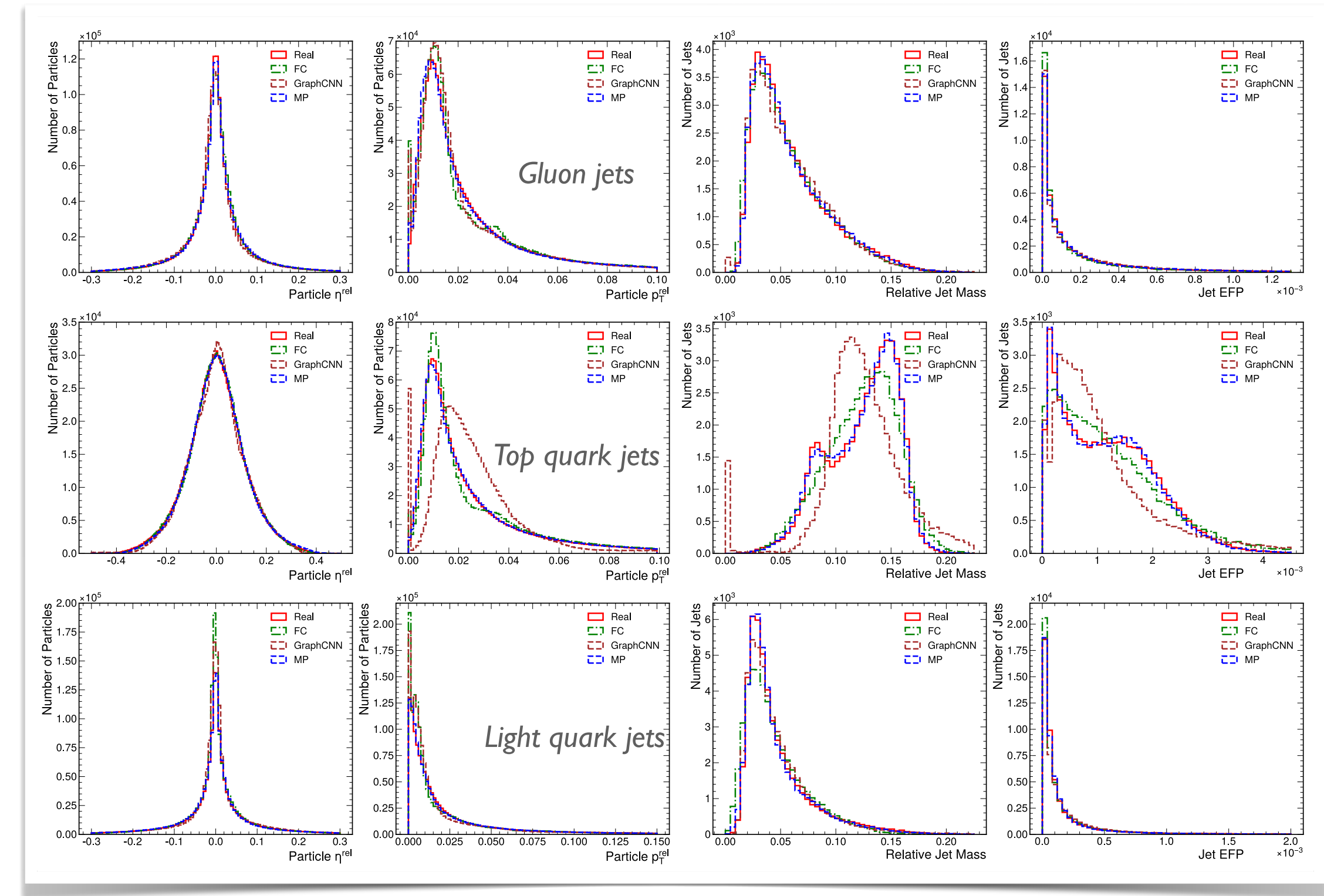
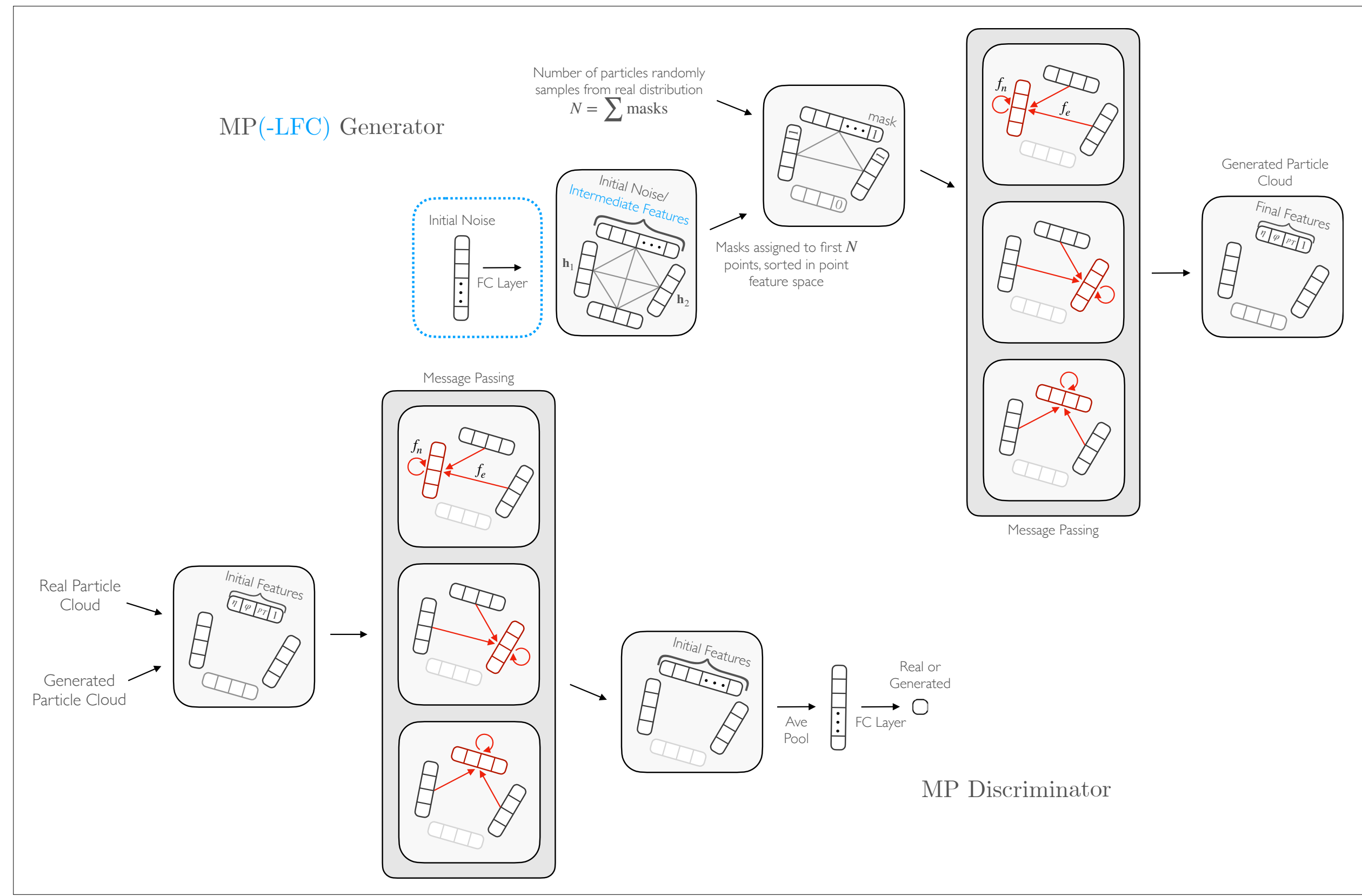


PARTICLE CLOUD GENERATION



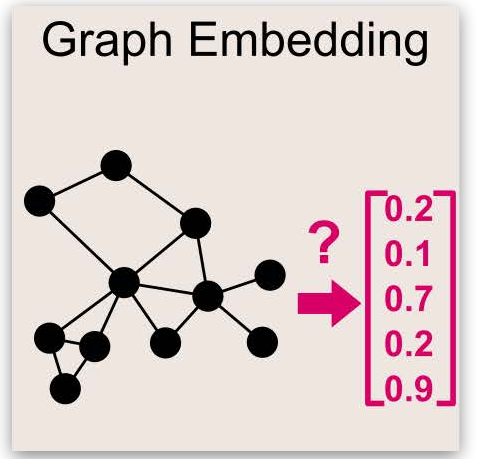
- Exploit GNNs for “particle cloud” generation
 - enables fast detector simulation

R. Kansal, J. Duarte, H. Su, B. Orzari, T. Tomei, M. Pierini, M. Touranakou, J. R. Vlimant and D. Gunopulos
 [NeurIPS 2021]

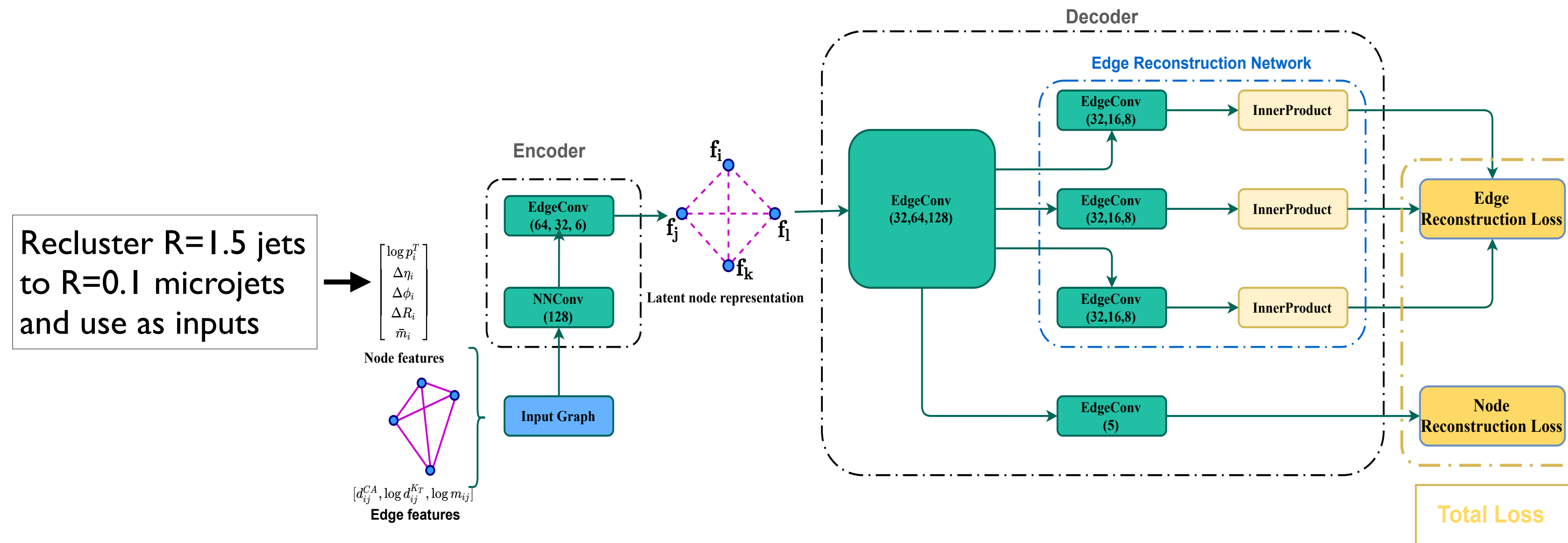


ANOMALY DETECTION

O. Atkinson, A. Bhardwaj, C. Englert, V. S. Ngairangbam and M. Spannowsky
 [JHEP 08 (2021) 080]



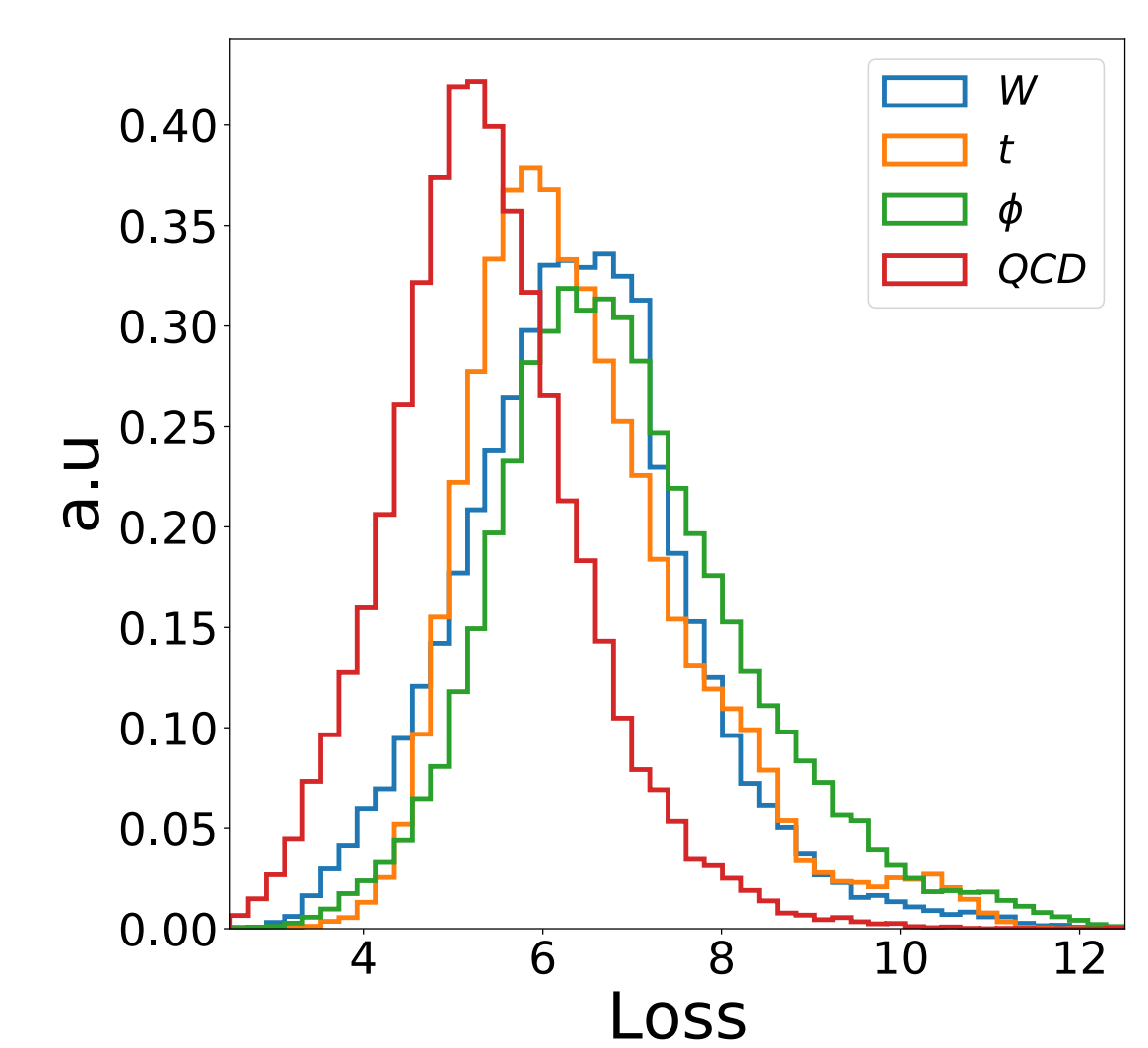
- GNN based autoencoders for anomaly detection
 - enables automated and model-agnostic new physics search



Recluster R=1.5 jets to R=0.1 microjets and use as inputs

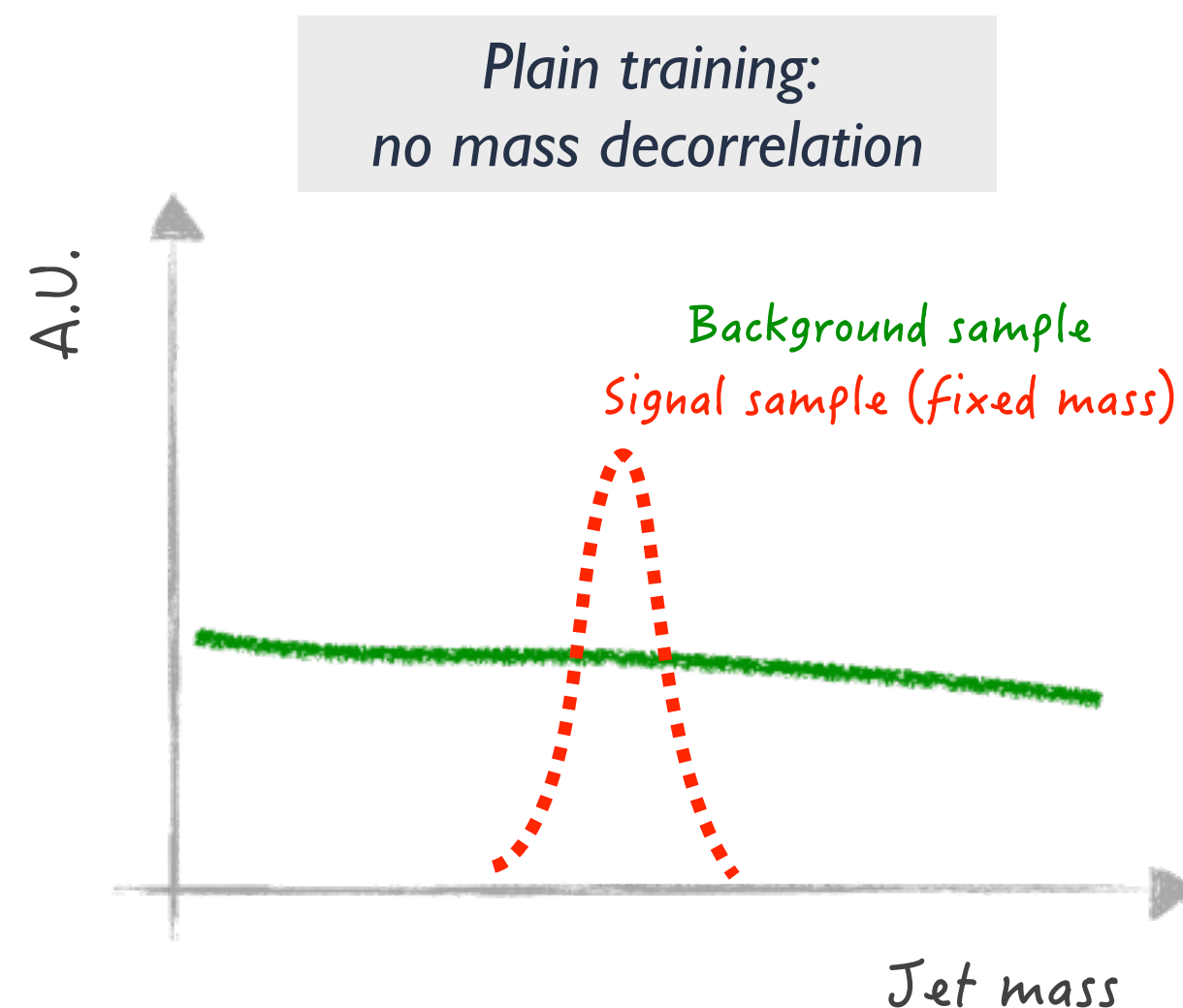
$$L_{node} = \sqrt{\sum_{ia} \frac{(\hat{x}_i^a - x_i^a)^2}{N \times 5}}, \quad L_{edge} = \sum_a \sqrt{\sum_{ij} \frac{(\hat{A}_{ij}^a - A_{ij}^a)^2}{N \times N}}$$

$$L_{auto} = \lambda_{node} L_{node} + \lambda_{edge} L_{edge}$$

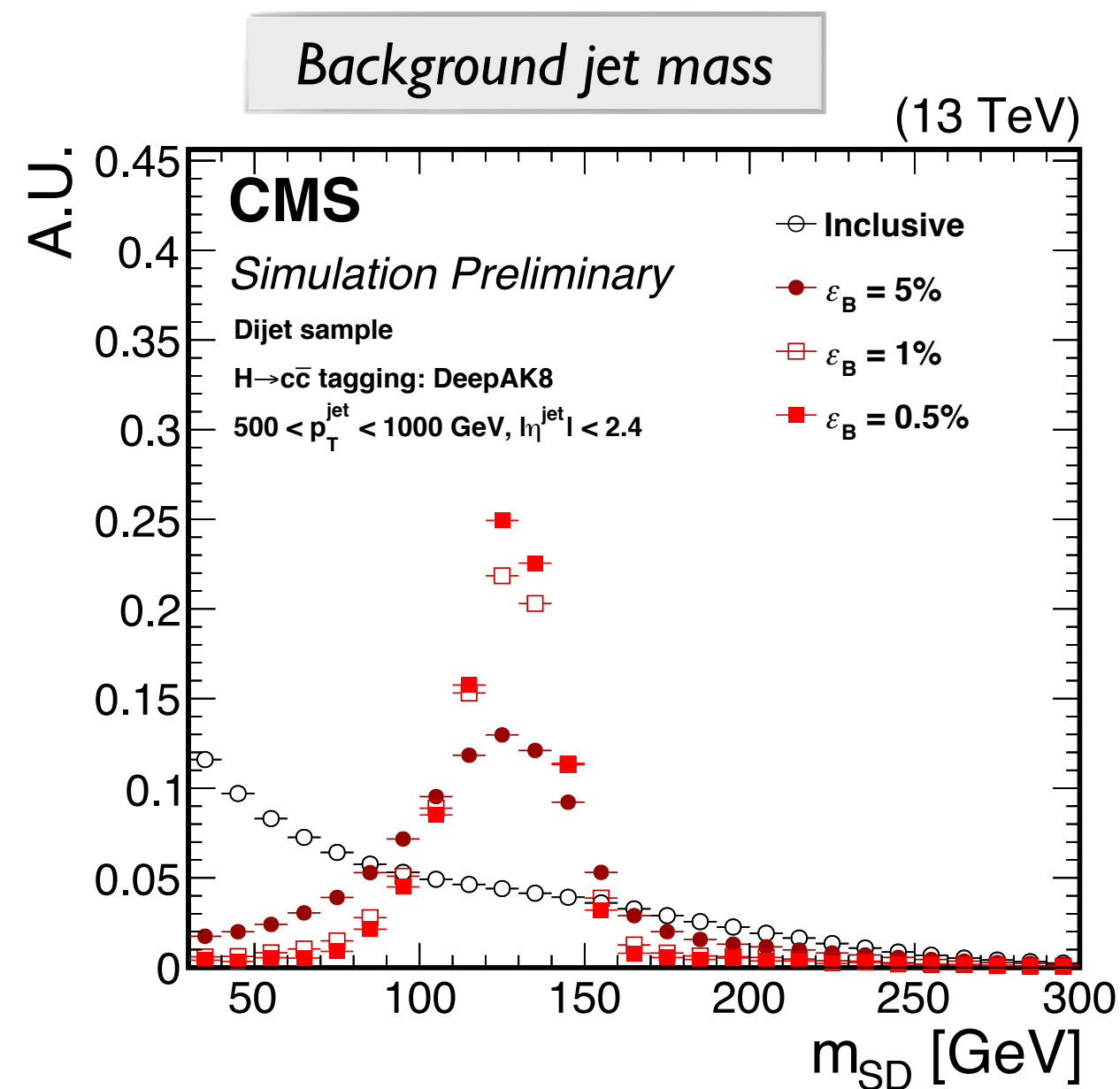


CORRELATION WITH THE JET MASS

CMS DP-2020/002



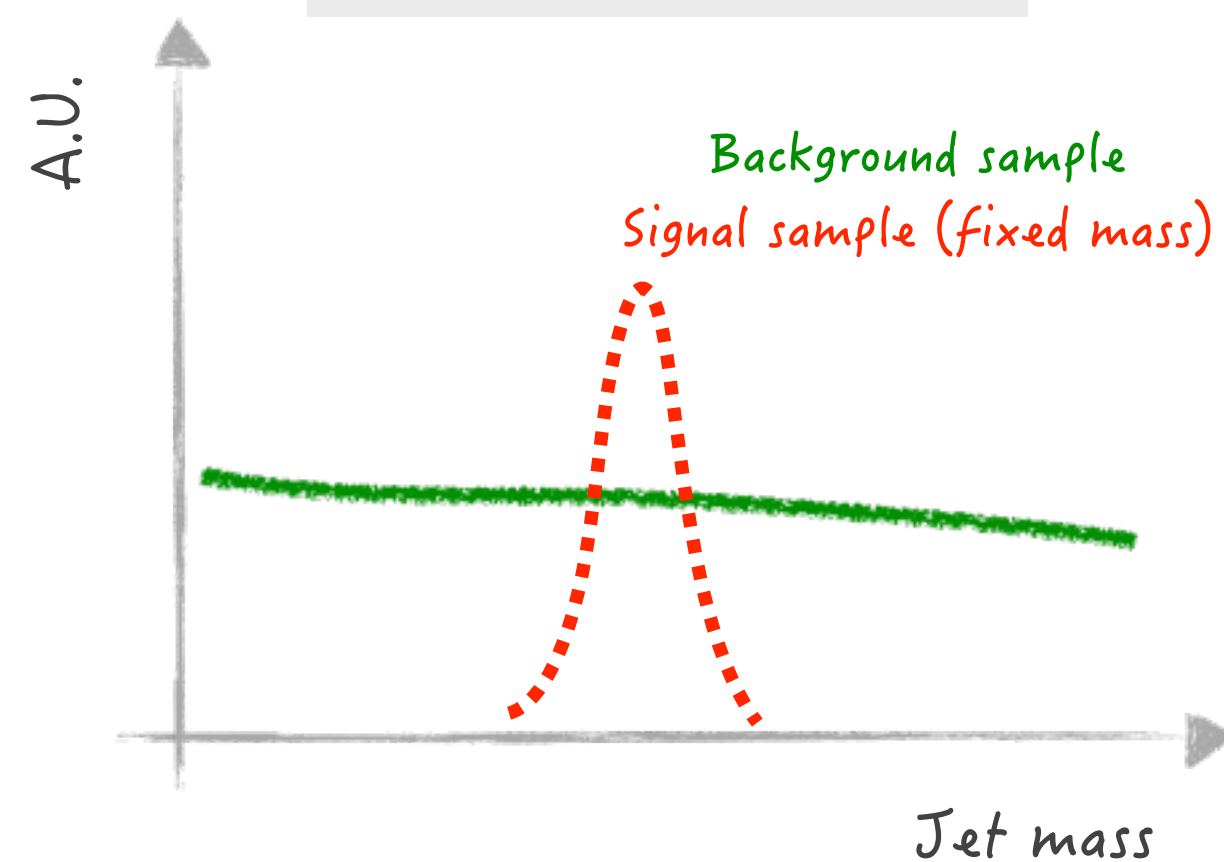
- One feature of these taggers is the correlation with the jet mass
 - jet mass shape of the background becomes similar to that of the signal after selection with the tagger: “**mass sculpting**”
 - not necessarily a problem, but a mass-independent tagger is often more desirable:
 - allows to use the mass variable to further separate signal and background
 - enables tagging signal jets with an unknown mass
 - ...



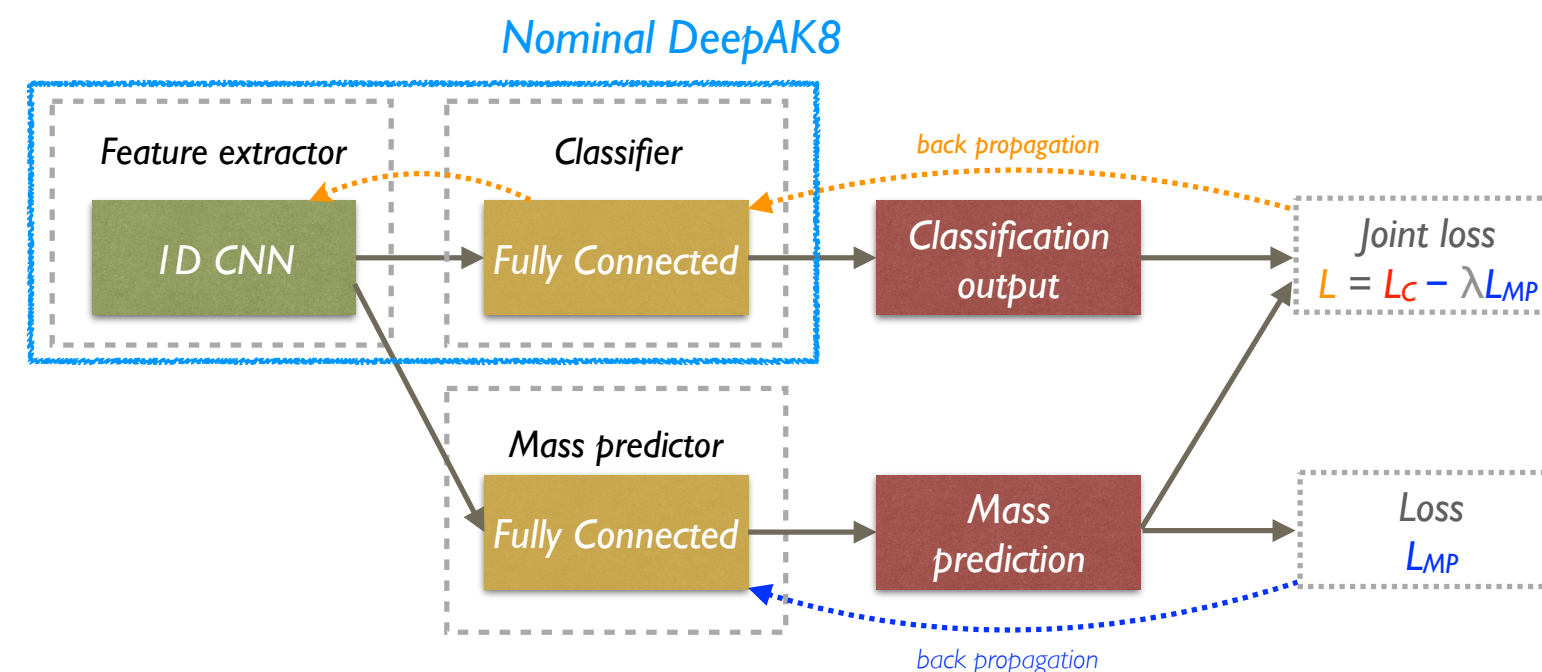
DECORRELATION WITH THE JET MASS

CMS DP-2020/002

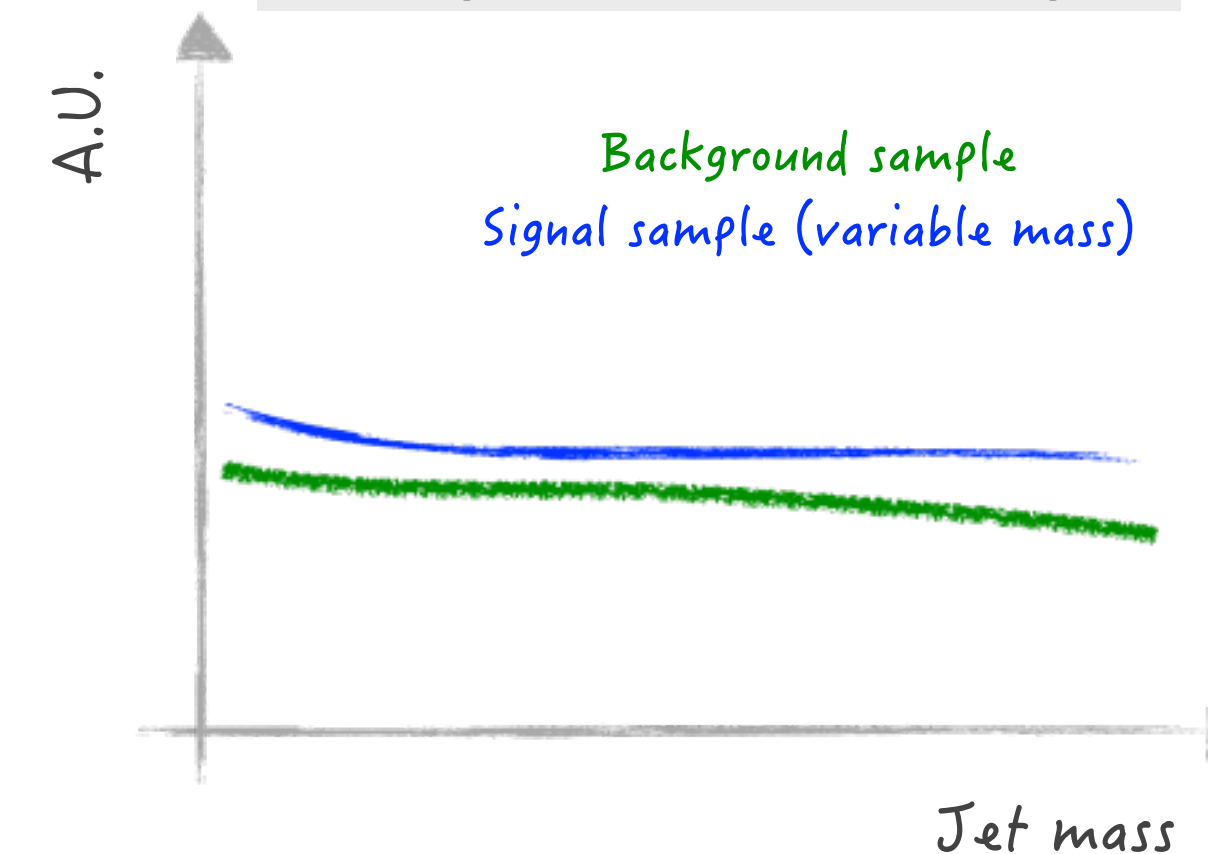
Plain training:
no mass decorrelation



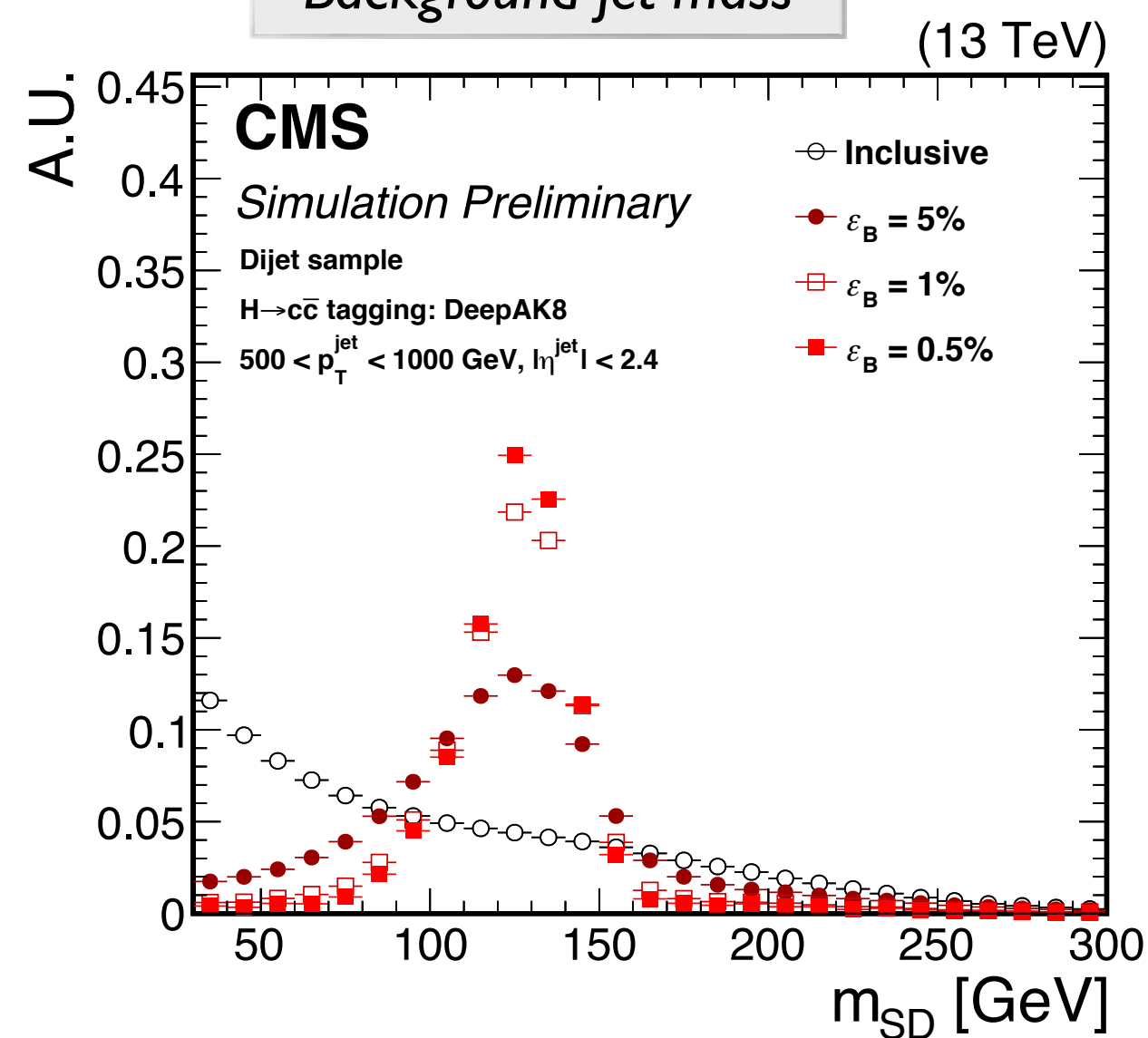
Mass-decorrelated DeepAK8:
“adversarial training”



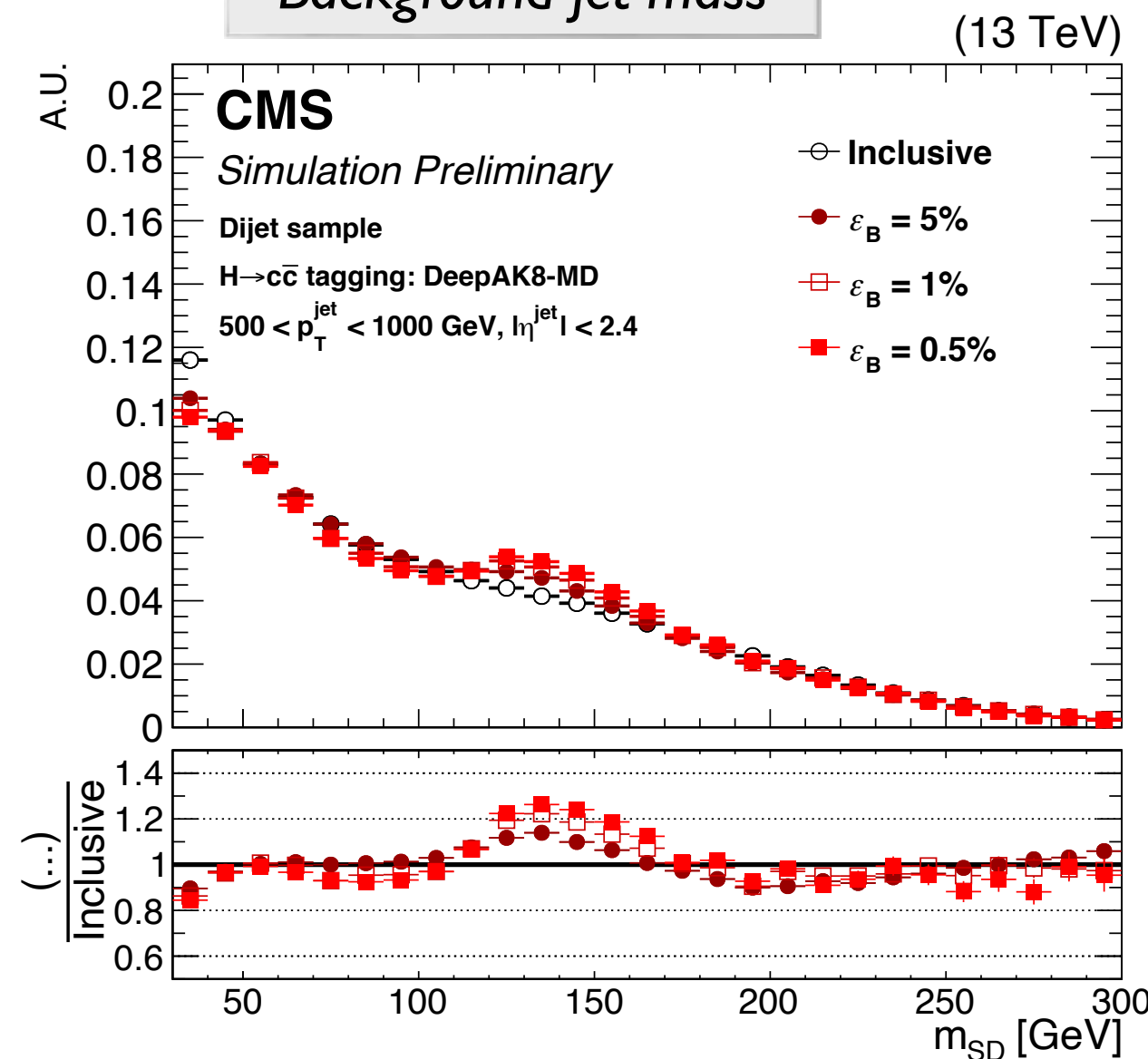
Mass-decorrelated ParticleNet:
training with variable-mass signal



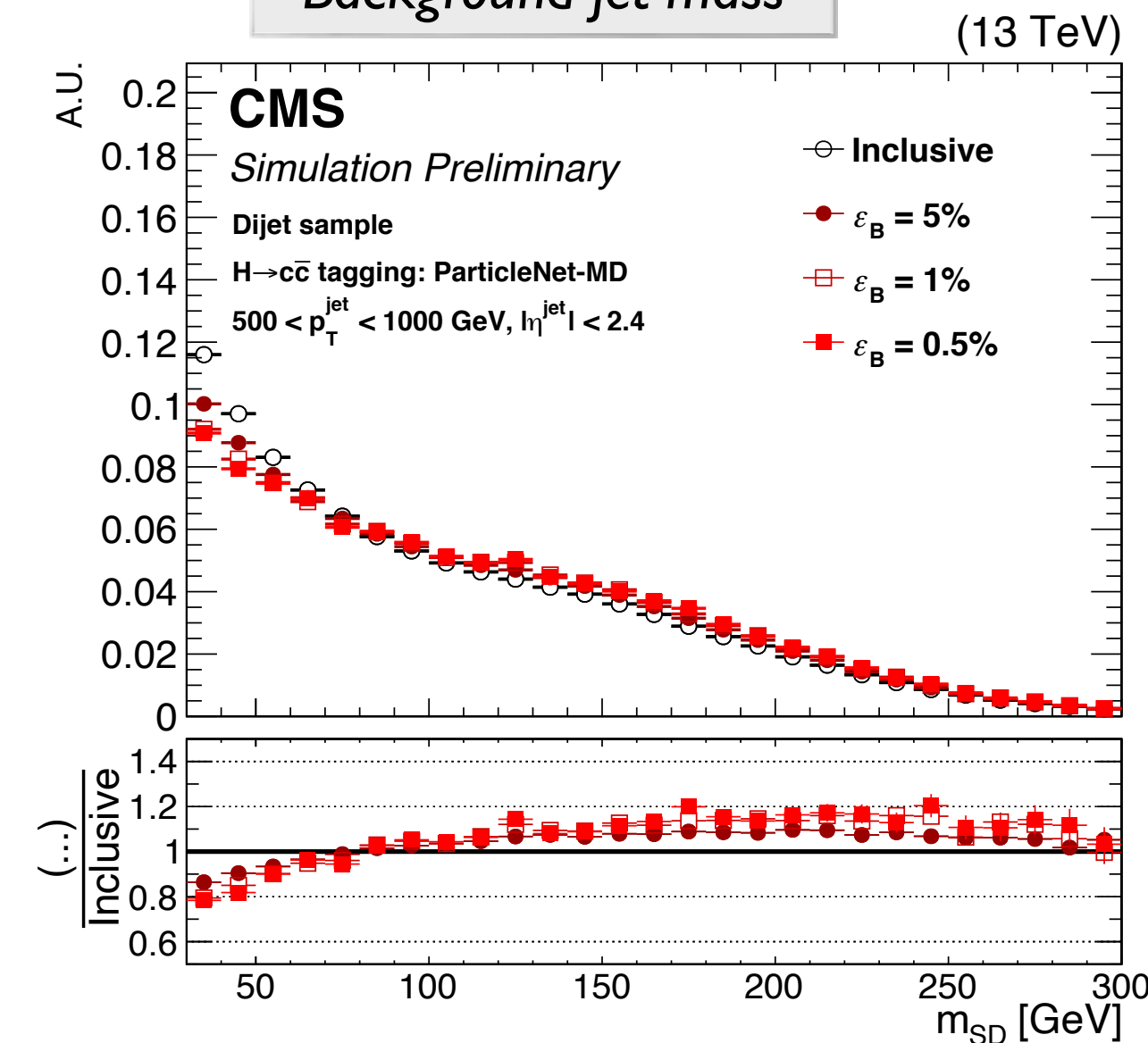
Background jet mass



Background jet mass

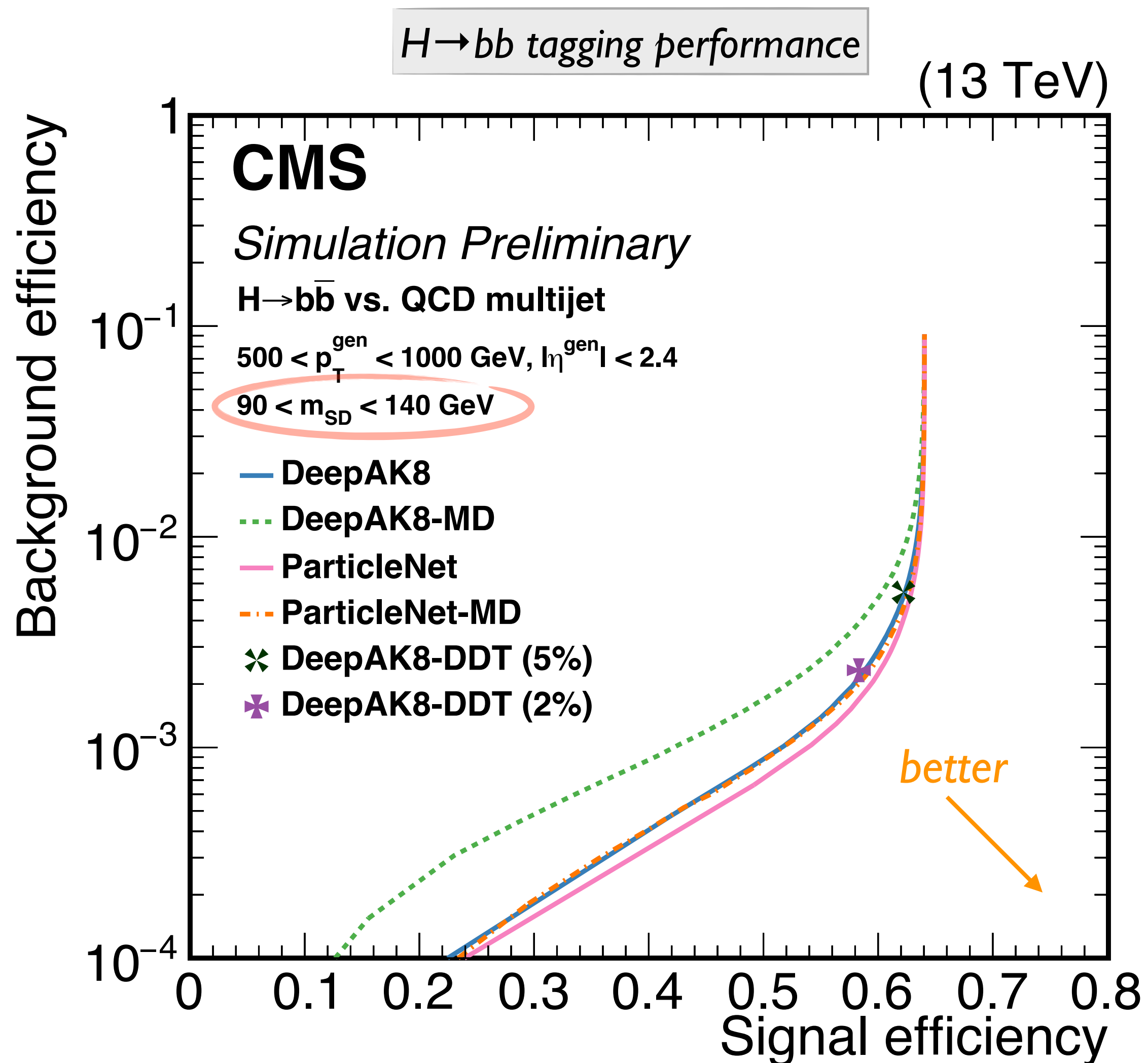


Background jet mass



PERFORMANCE COMPARISON

CMS DP-2020/002



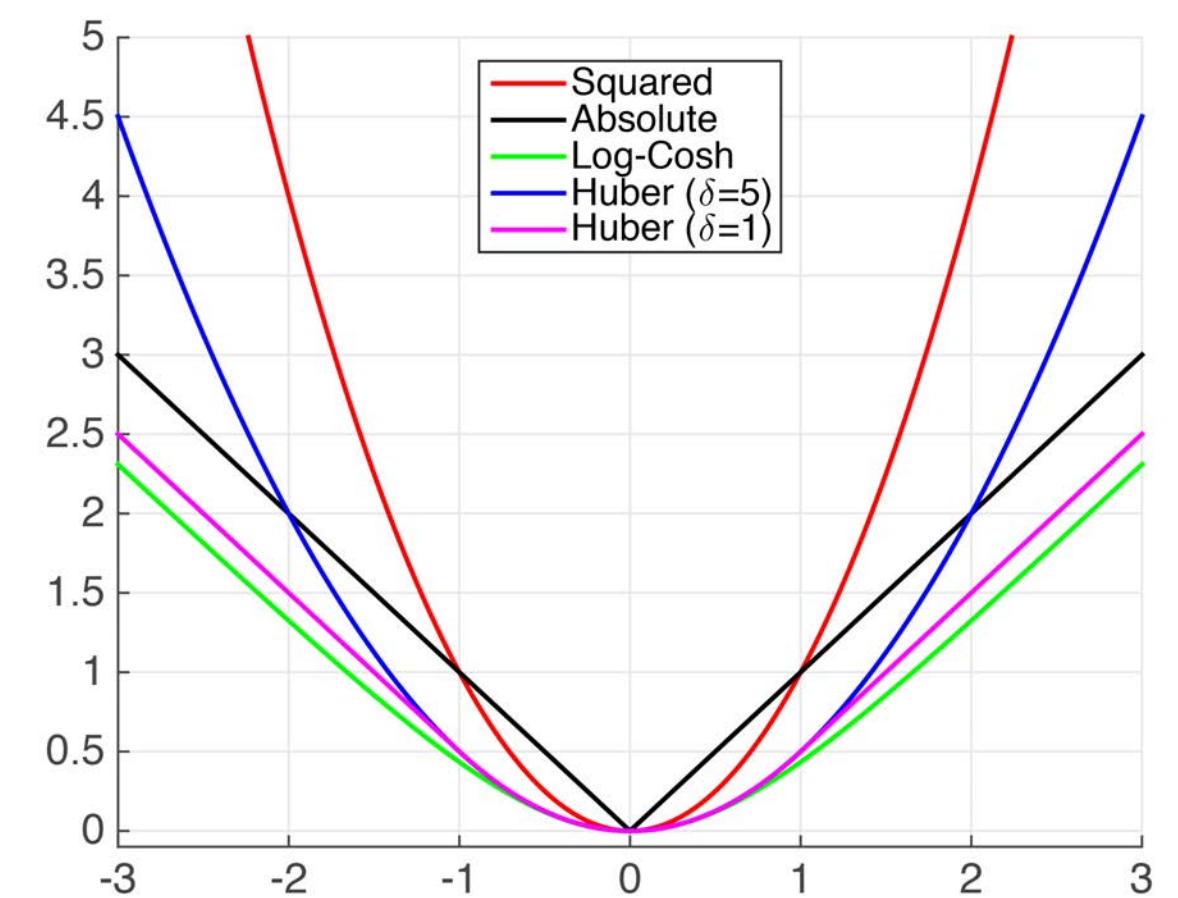
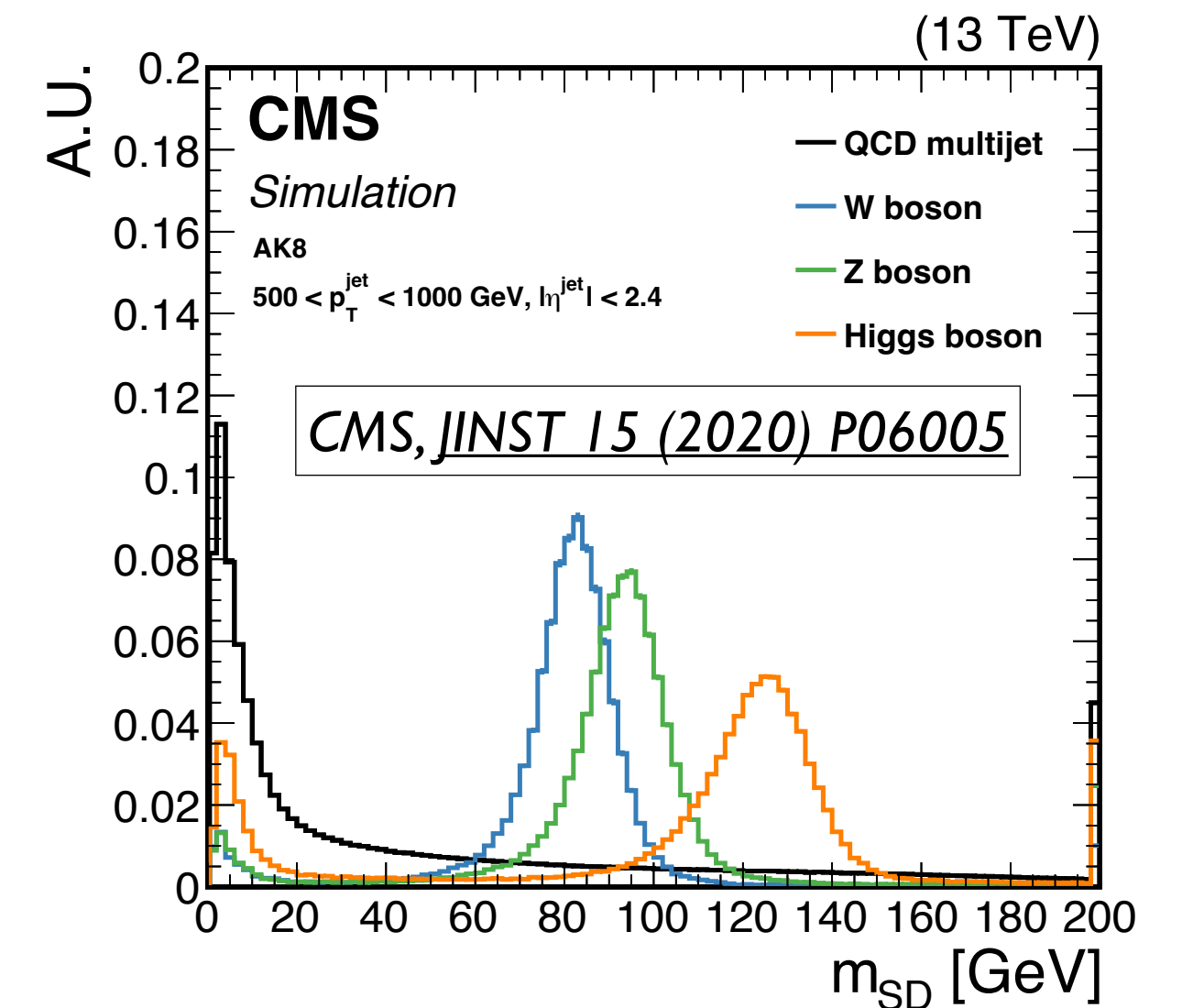
- ParticleNet-MD
 - using a special signal sample for training
 - hadronic decays of a spin-0 particle *X*
 - $X \rightarrow bb, X \rightarrow cc, X \rightarrow qq$
 - not a fixed mass, but a flat mass spectrum
 - $m(X) \in [15, 250] \text{ GeV}$
 - allows to easily reweight both signal and background to a \sim flat 2D distribution in (p_T , mass) for the training
- ParticleNet-MD shows the best performance
 - \sim 3-4x better background rejection compared to DeepAK8-MD (based on “adversarial training”)
 - only slight performance loss compared to the nominal version w/o mass decorrelation

MASS REGRESSION

- Jet mass: one of the most powerful observables for boosted jet tagging
 - characteristic mass peak for top/W/Z/H jets v.s. continuum for QCD jets
- Mass regression:
 - exploit deep learning to reconstruct jet mass with the highest possible resolution
 - training setup similar to the ParticleNet tagger
 - but: predict the jet mass directly from the jet constituents
- Regression target:
 - signal ($X \rightarrow bb/cc/qq$): generated particle mass of X [flat spectrum in 15 – 250 GeV]
 - background (QCD) jets: soft drop mass of the generated particle-level jet

Loss function

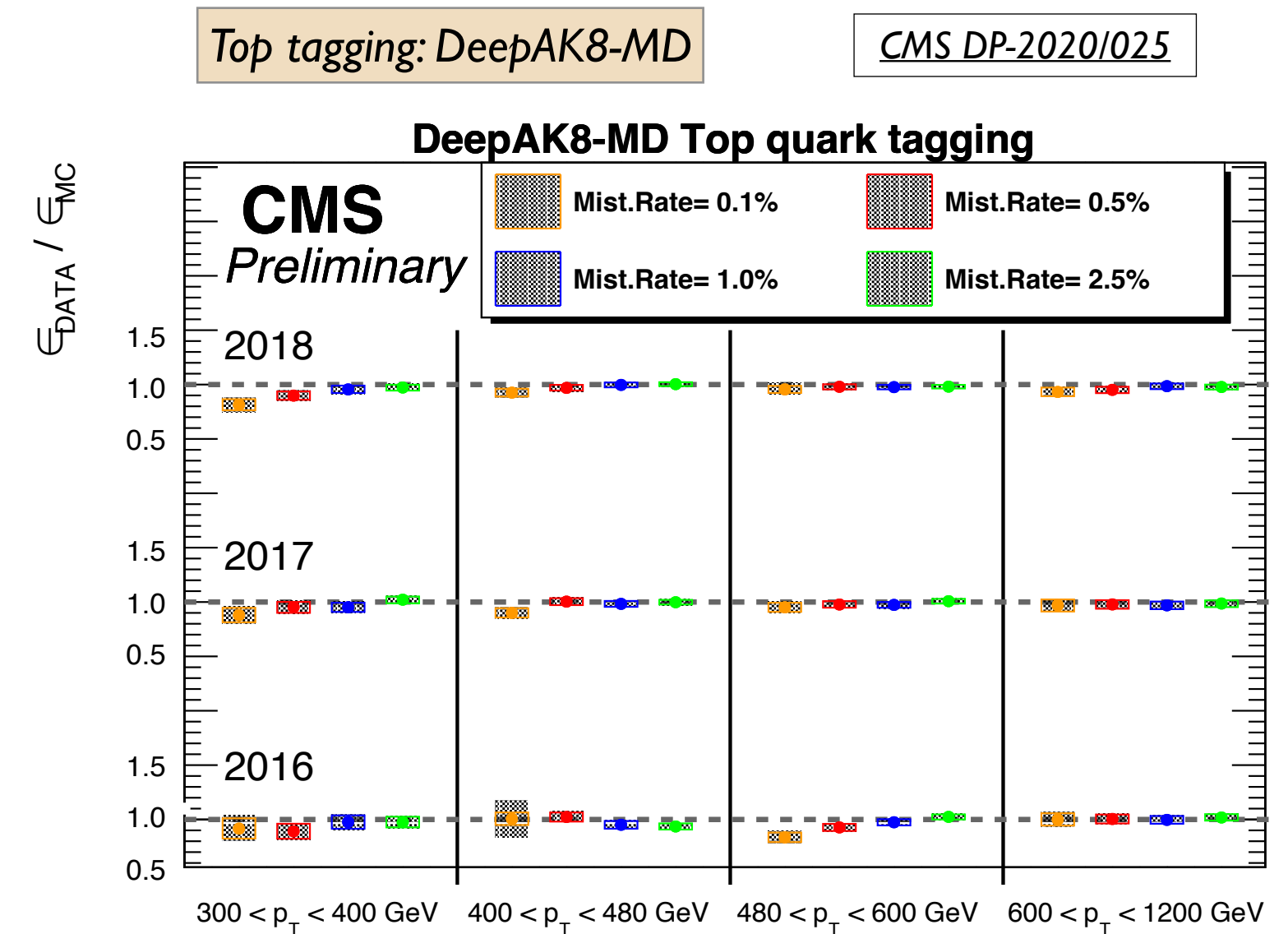
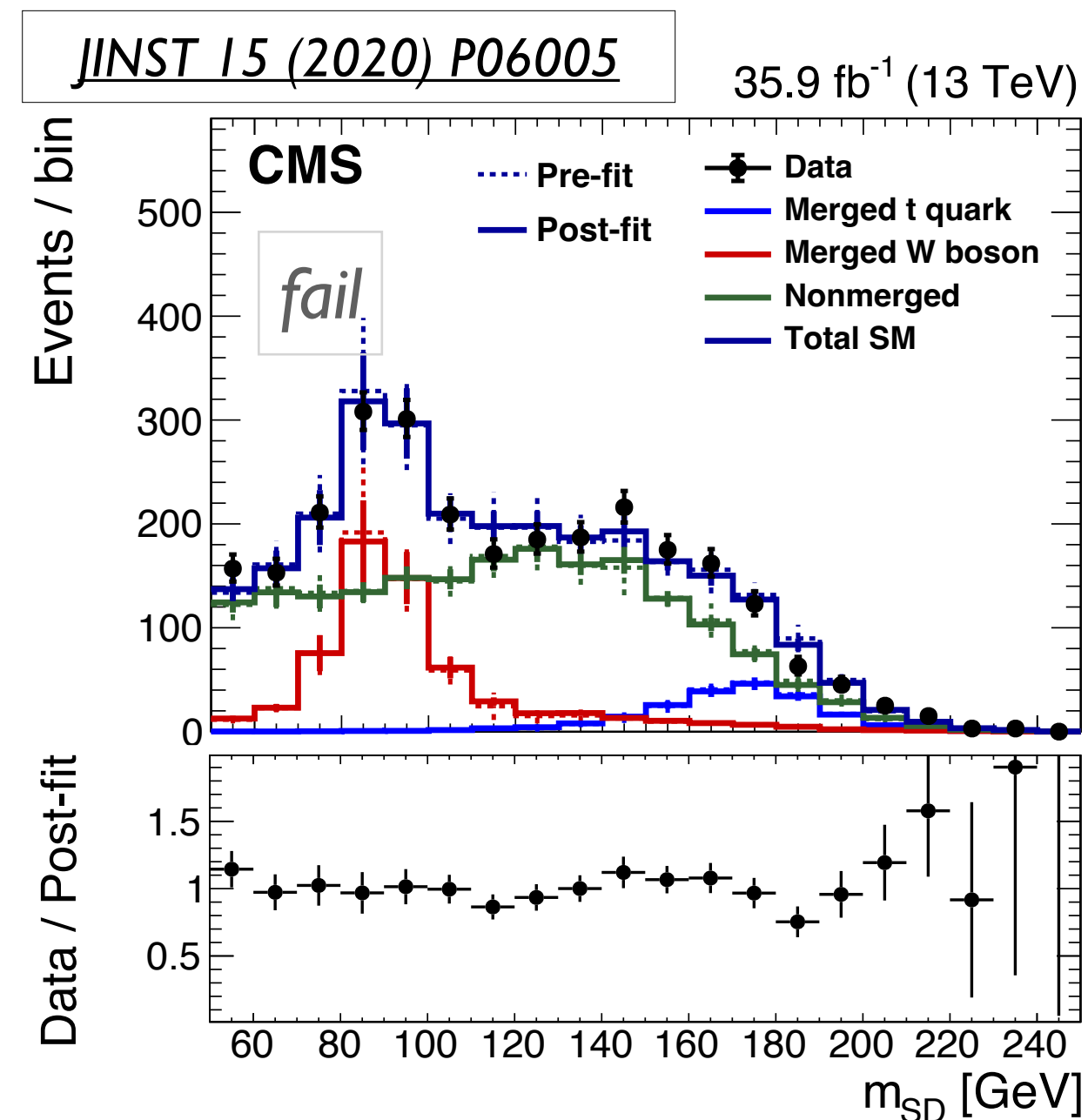
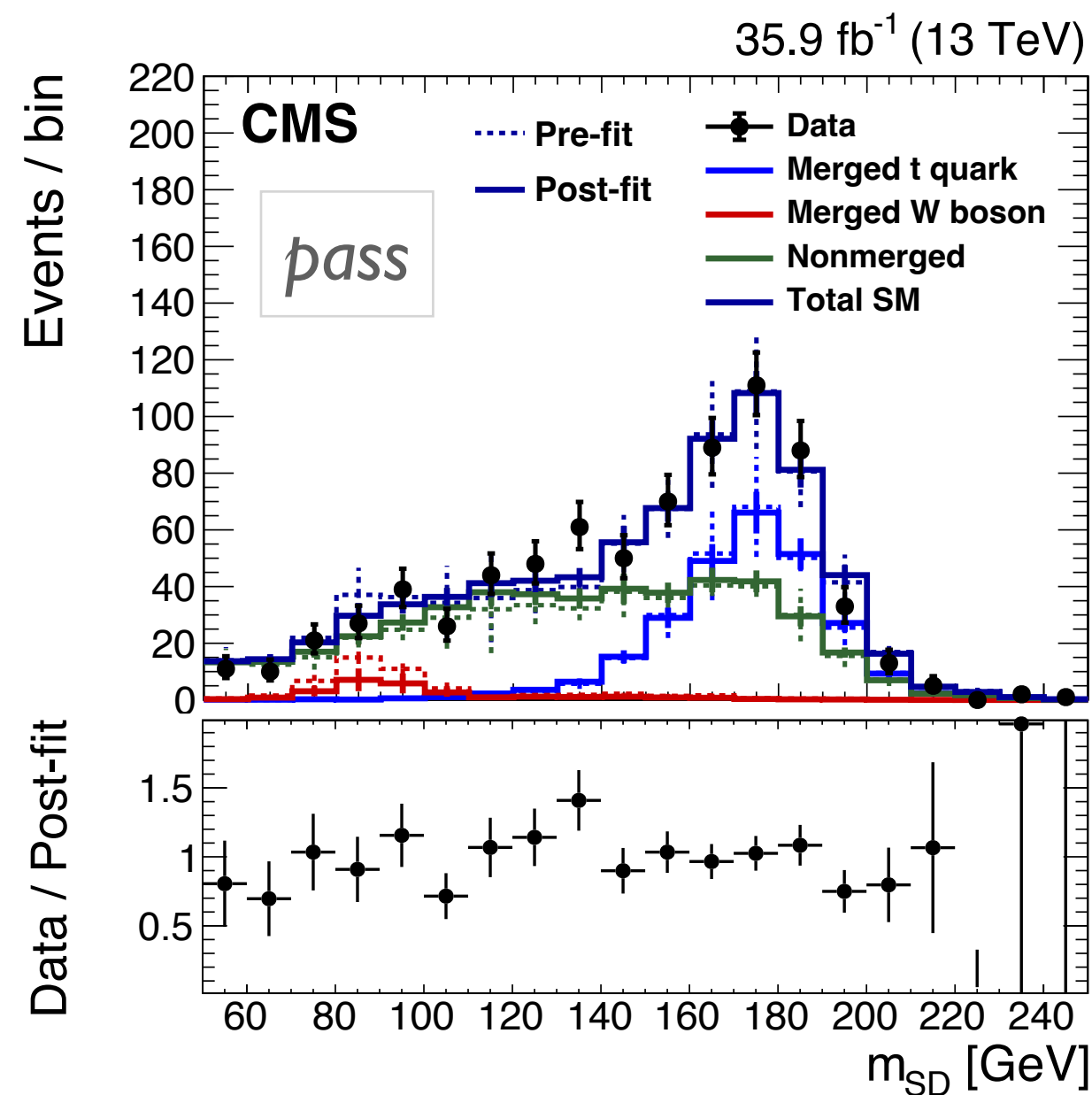
- LogCosh:
$$L(y, y^p) = \sum_{i=1}^n \log(\cosh(y_i^p - y_i))$$



<https://www.cs.cornell.edu/courses/cs4780/2015fa/web/lecturenotes/lecturenote10.html>

TAGGER CALIBRATION IN DATA

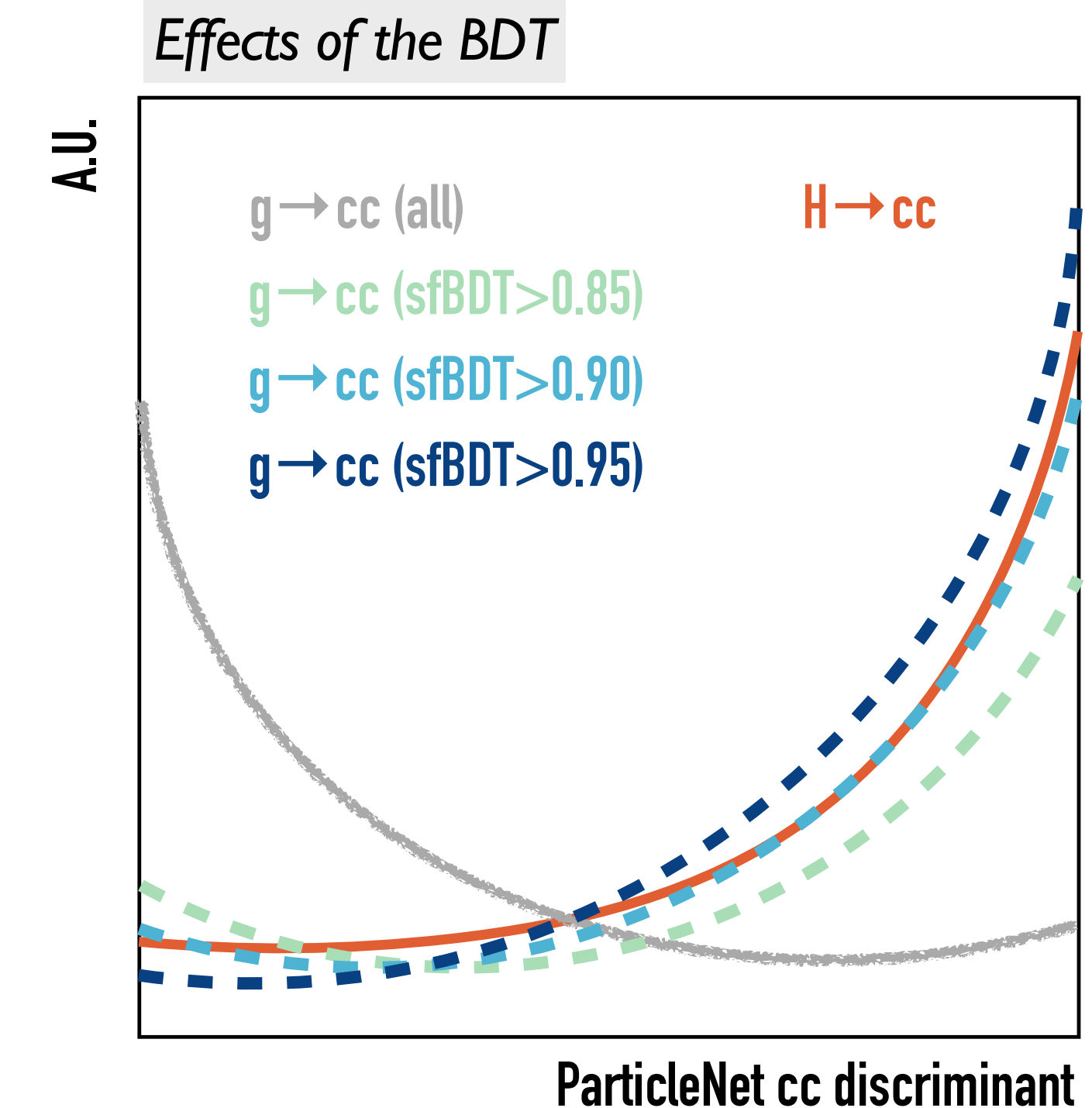
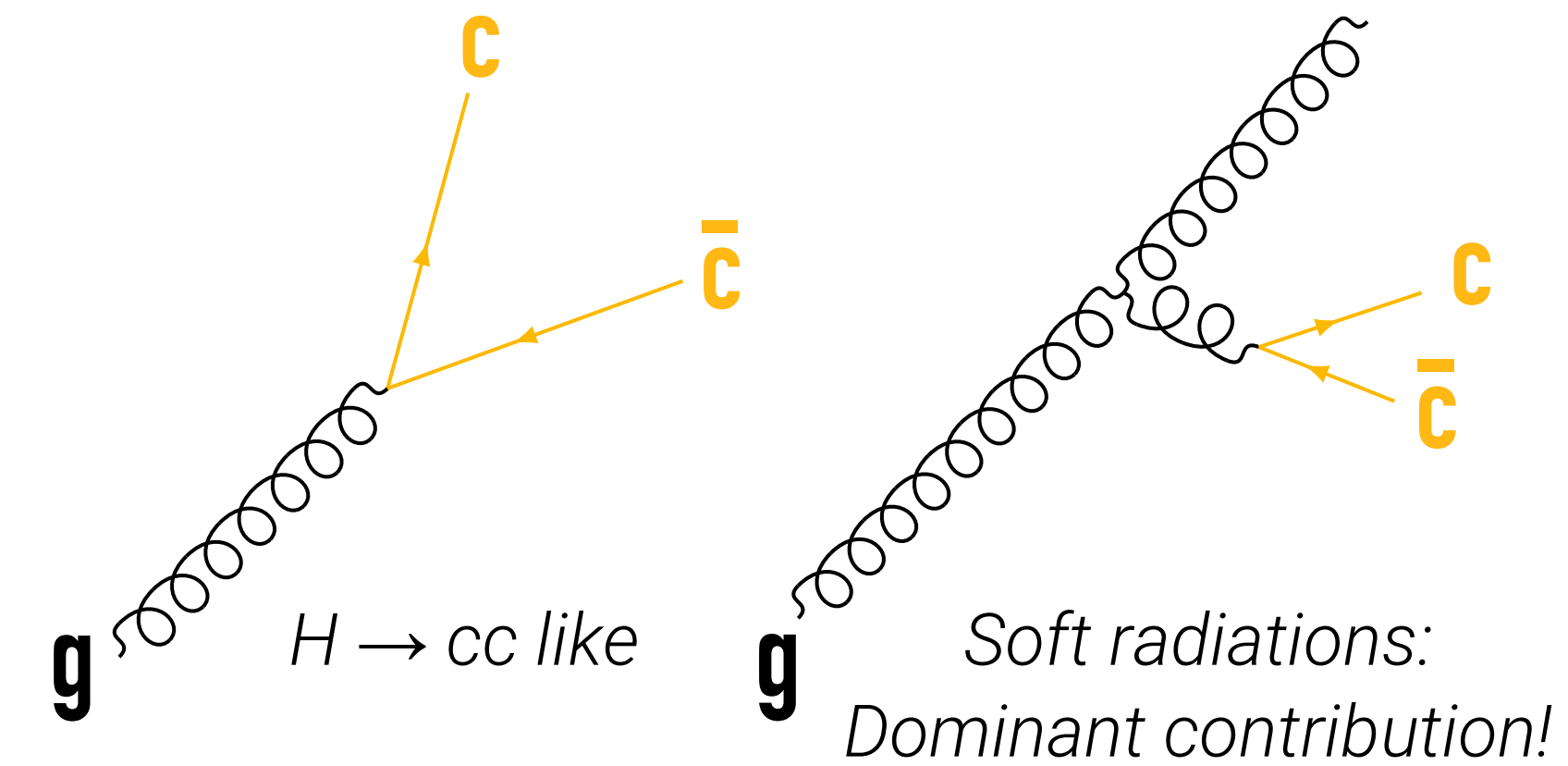
- Crucial to calibrate these taggers in real data for them to be used in analyses
- Top/W tagging efficiency



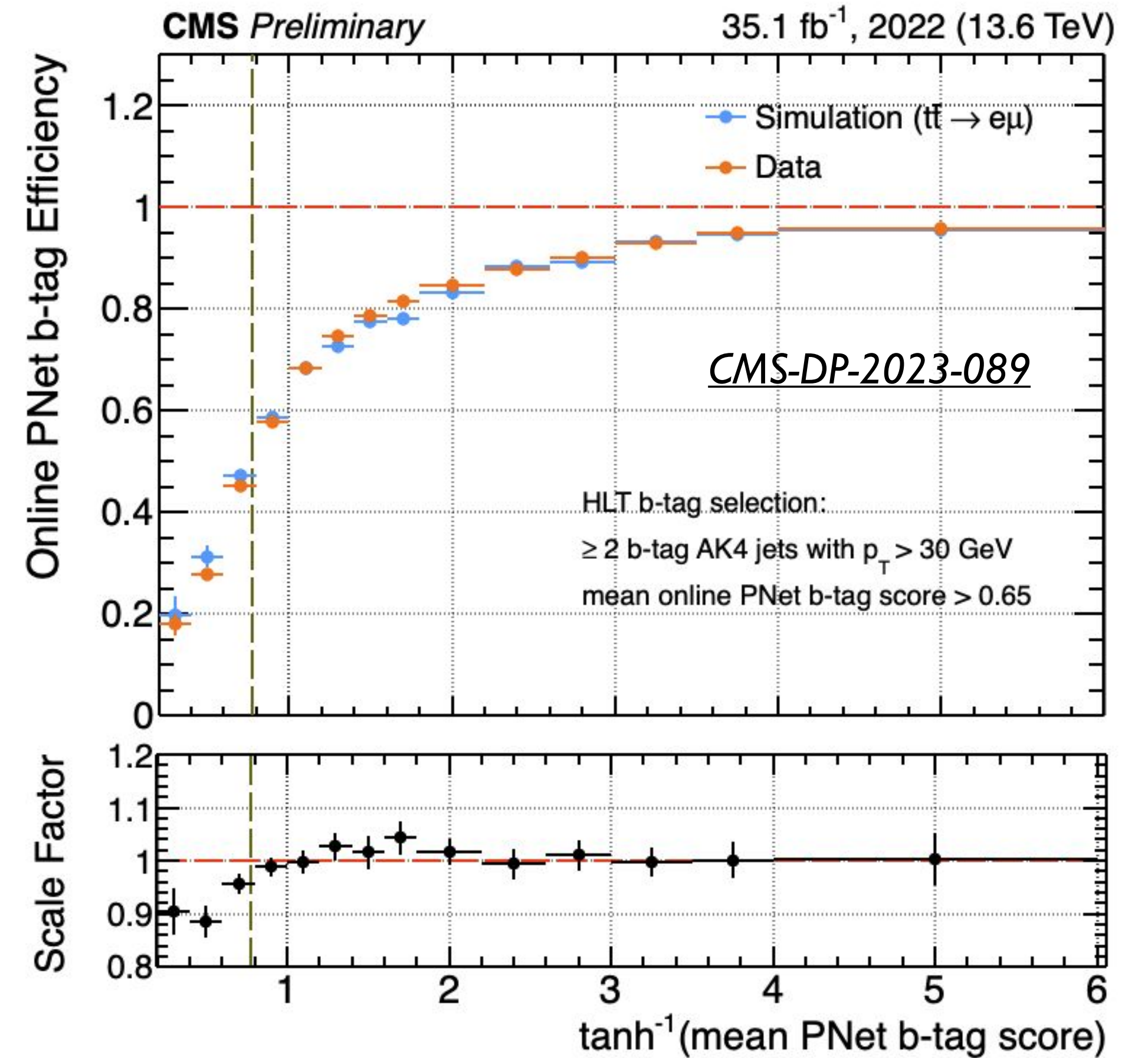
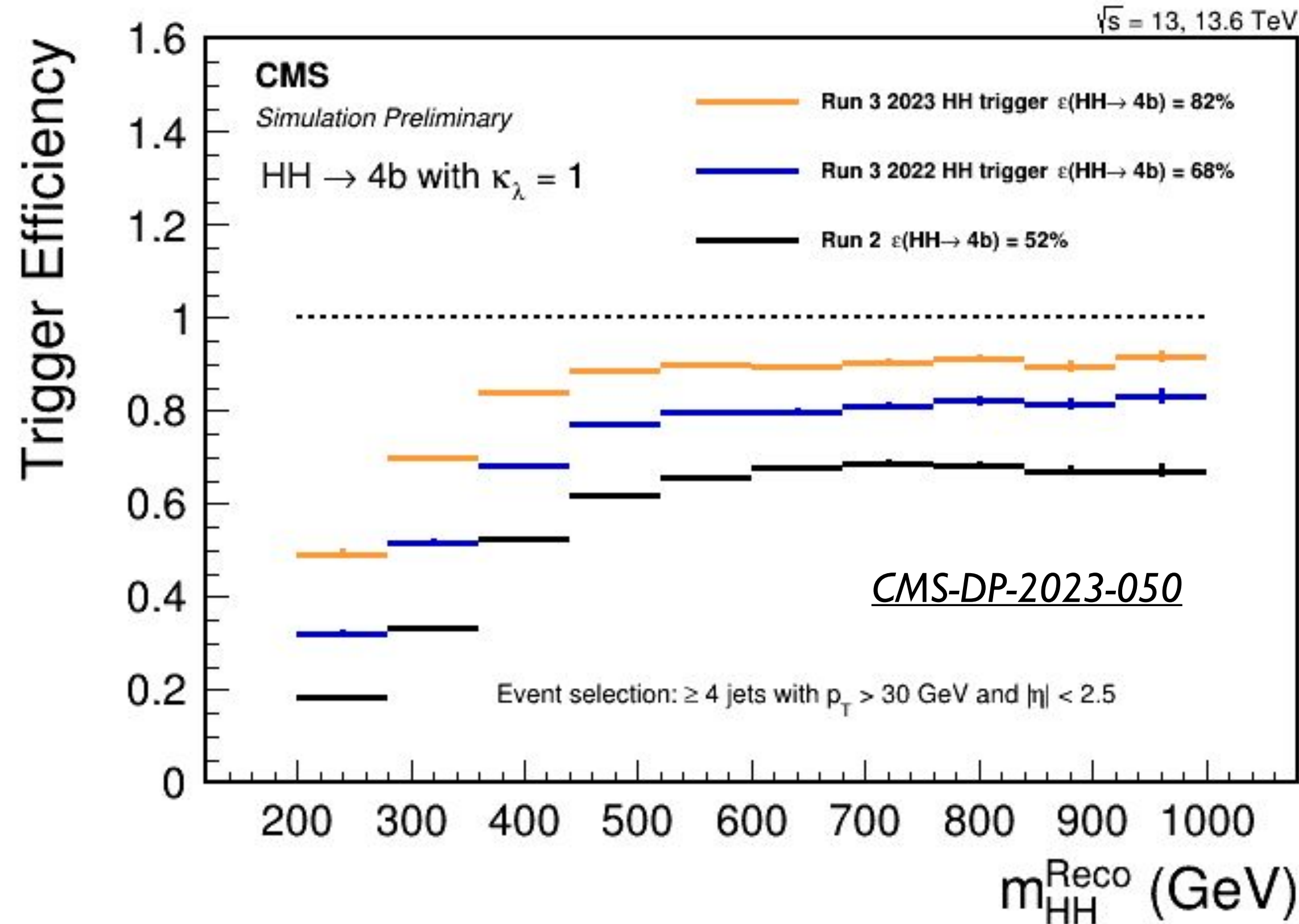
- measured using the single- μ sample enriched in semi-leptonic $t\bar{t}b\bar{a}$ events
- fit jet mass templates in the “pass” and “fail” categories simultaneously to extract efficiency in data
 - simulation-to-data scale factors $SF := \text{eff}(\text{data}) / \text{eff}(\text{MC})$ derived to correct the simulation
- jet mass scale and resolution scale factors can also be extracted
- Mistag rates of background jet typically derived directly from analysis-specific control regions

Calibration of the cc-tagger

- ❑ Need to measure ParticleNet cc-tagging efficiency in data
 - no pure sample of $H \rightarrow cc$ jets (or even $Z \rightarrow cc$) in data
 - using $g \rightarrow cc$ in QCD multi-jet events as a proxy
- ❑ Difficulty: select a phase-space in $g \rightarrow cc$ that resembles $H \rightarrow cc$
 - solution: a **dedicated BDT** developed to distinguish **hard 2-prong splittings** (i.e., high quark contribution to the jet momentum) from **soft cc radiations** (i.e., high gluon contribution to the jet momentum)
 - also allows to adjust the similarity between proxy and signal jets
 - by varying the sfBDT cut – treated as a systematic uncertainty
- ❑ Perform a fit to the secondary vertex mass shapes in the “passing” and “failing” regions simultaneously to extract the scale factors
 - three templates: cc (+ single c), bb (+ single b), light flavor jets
- ❑ Derived cc-tagging scale factors typically 0.9–1.3
 - corresponding uncertainties are 20–30%



CMS B-TAG HLTs



Trigger	Requirement	Rates at 2×10^{34} cm ⁻² s ⁻¹
2023 HH trigger	HT > 280 GeV, 4 jets with $p_T > 30$ GeV, PNet@AK4(mean 2 highest b-tag score) > 0.55	180 Hz
2022 HH trigger	4 jets $p_T > 70, 50, 40, 35$ GeV, PNet@AK4 (mean 2 highest b-tag score) > 0.65	60 Hz
2018 triple b-tag trigger	HT > 340 GeV, 4 jets $p_T > 75, 60, 45, 40$ GeV, 3 b-tags with DeepCSV > 0.24	8 Hz