Hypergraph learning for particle reconstruction at collider experiments

Francesco Armando Di Bello
U. and INFN Genova and Pisa. 28/08/25

UNIVERSITÀ DI PISA

INFN GENOVA

Introduction

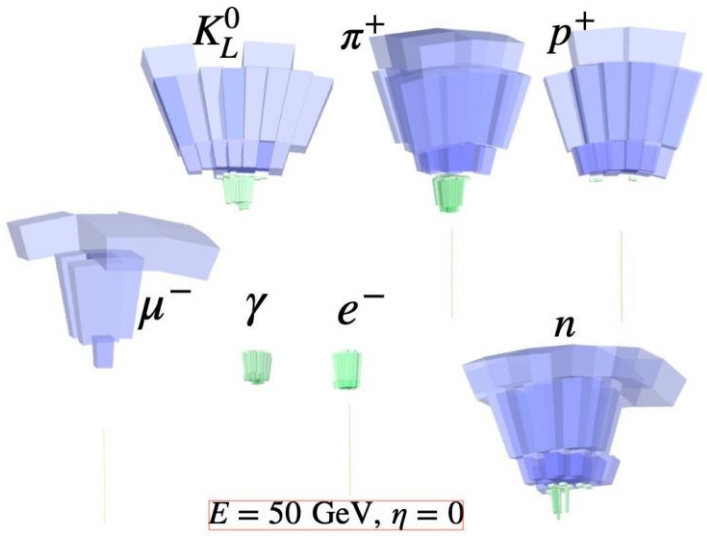
- Particle physics experiments are based on hierarchical, factorized pipelines whose goal is to allow unbiased parameters inference.

$$p(x \mid \theta) = \int dR'\, dR\, dH\, dz\; \delta\big(x - g(R')\big)\, \delta\big(R(H) - R\big)\, p_{\text{sim}}(H \mid z)\, p_{\text{gen}}(z \mid \theta)$$

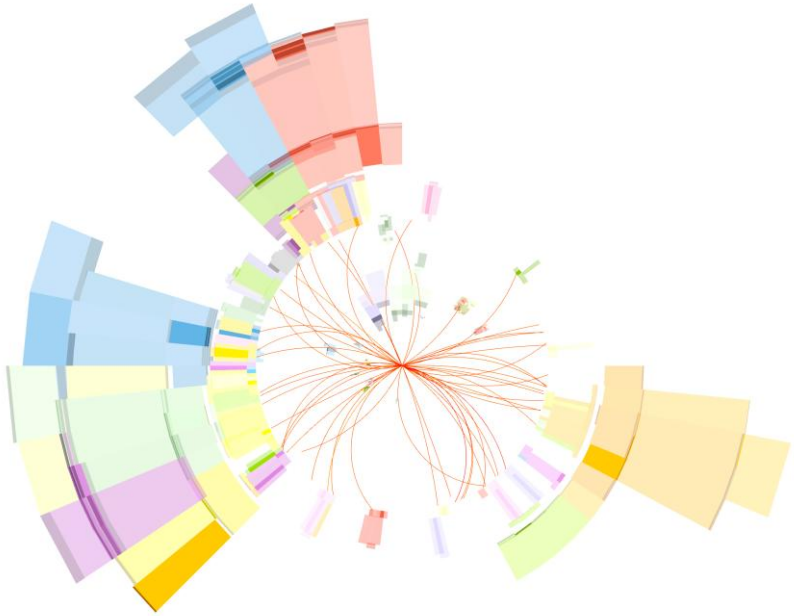
- Particle-flow is a part of this chain and a solution to the inverse problem of particle reconstruction R(H).

It can be phrased as follows:
- given the set of detector outputs, reconstuct the set of particles (and their properties) that generated them –

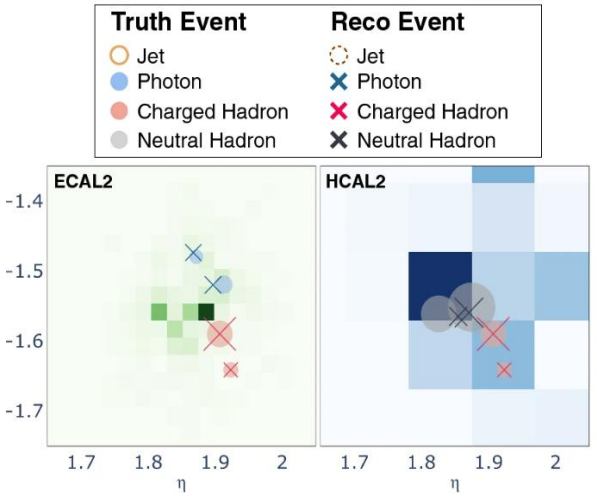
Particles signatures



Detector output



Solving the inverse problem



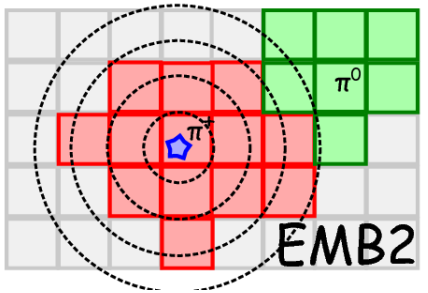
Particle flow generalities

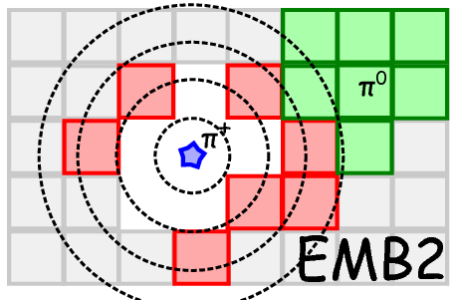
$$\frac{\sigma_E}{E} \approx \frac{a}{\sqrt{E}} \qquad \frac{\sigma_p}{p} \approx a\,p$$

- Particle flow algorithms can optimally account for detector redudancies (e.g. calorimeter vs tracker)

- Typical implementation of Pflow algorithms are based on linearity: parametrize single particle shower – will be calling this PPflow

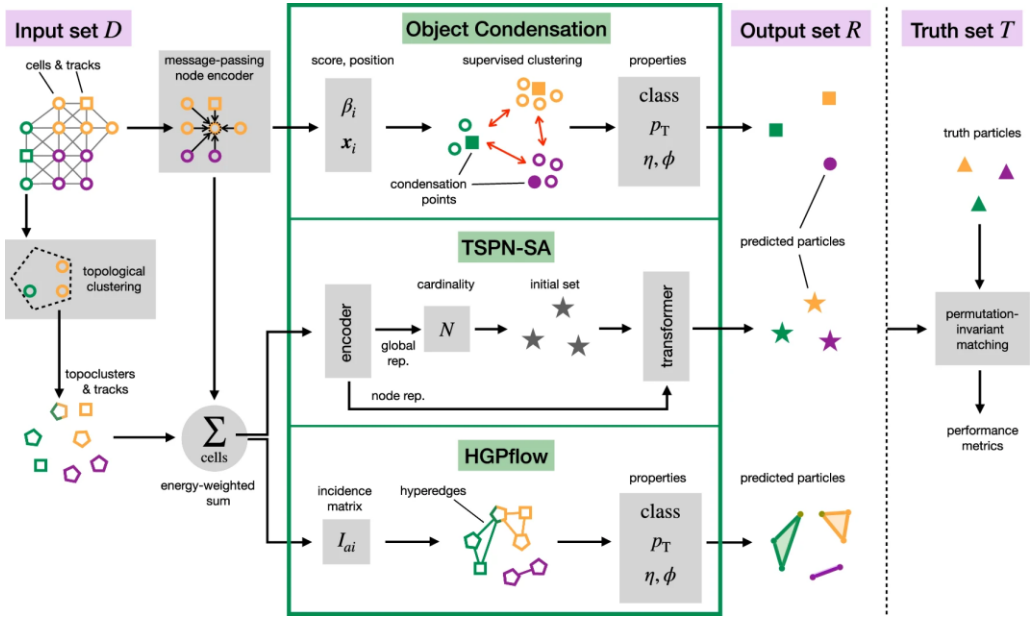
Eur. Phys. J. C 77 (2017) 466



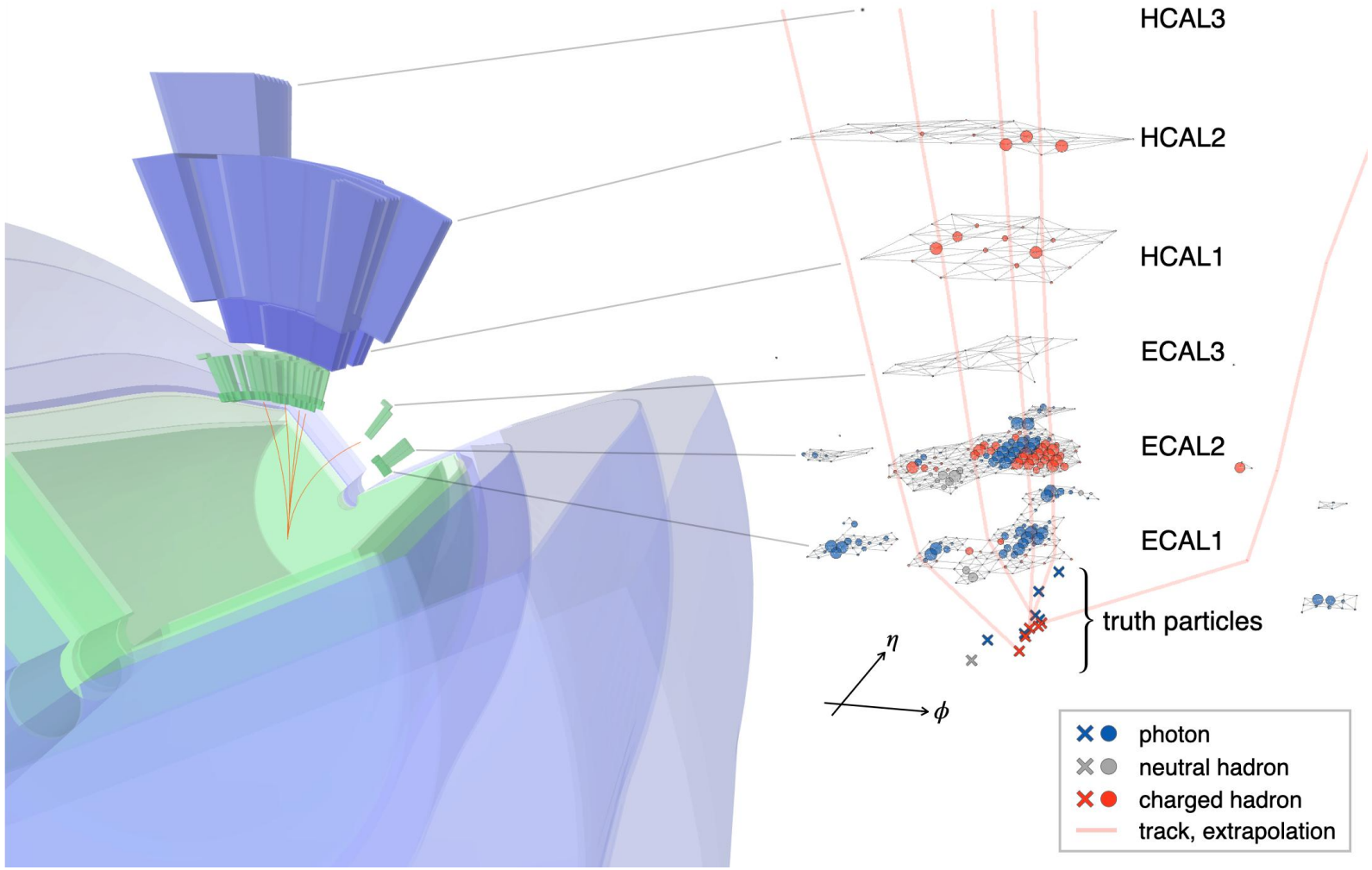


Can Machine Learning help Particle Flow reconstruction?

Yes. A non exhaustive list: PRD111.09(2025) Comm. Physics 124 (2024) EPJC596 (2023) CMS(2025)Note



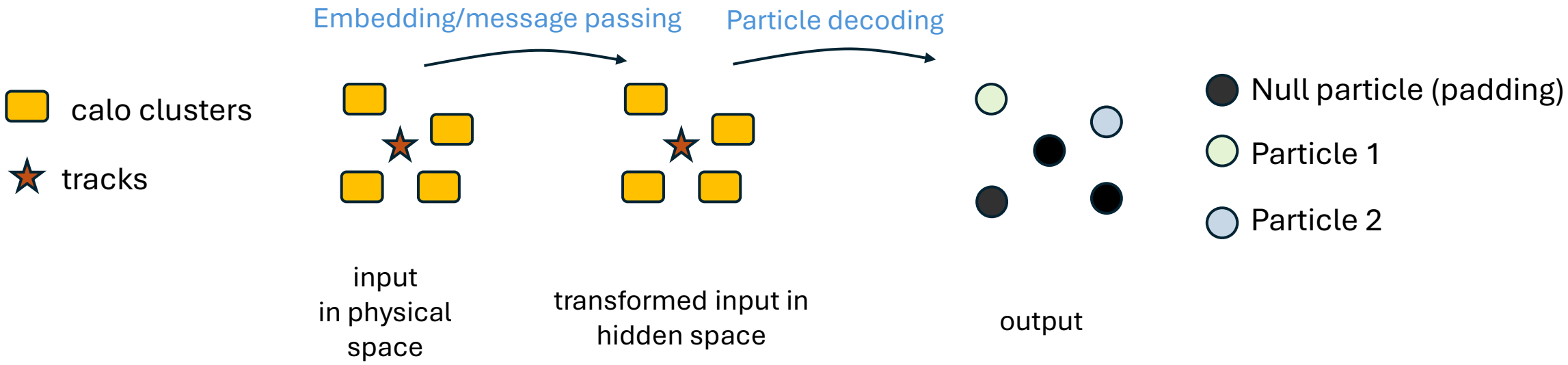
Constructing the inputs



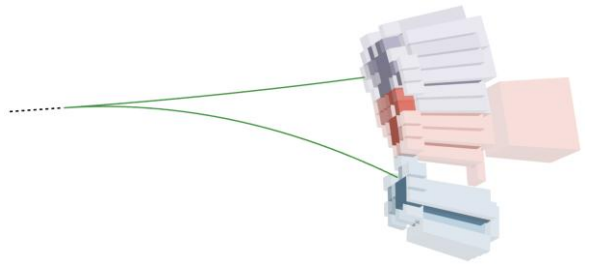
Starting point is to build a graph from the detector outputs. This comes with some arbitrariness

Machine learning for particle reconstruction

- One of the first, if not the first proposal was Machine Learning for Particle Flow (MLPF): MLPF EPJC(2021)
- Take as an example a simplified scenarios, two particles, 4 calo-cluster, one track
- Need to define an ordering rule to associate which target particle belongs to a given input object

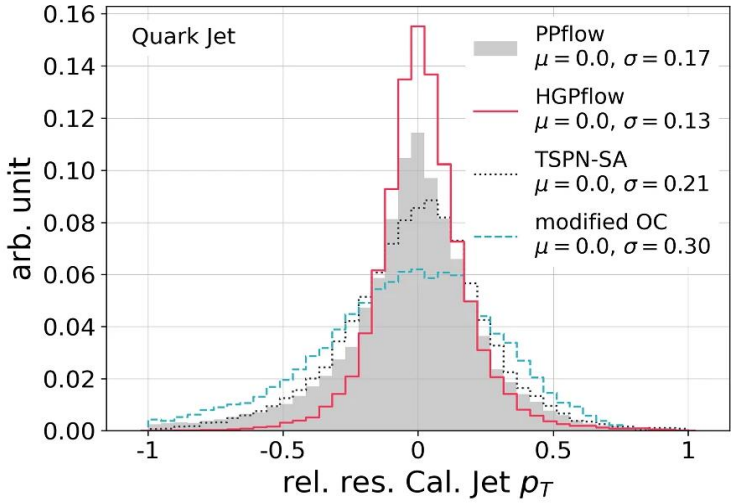


N.B: the definition of target particle is highly non trivial, requires some semi-fiducial definition

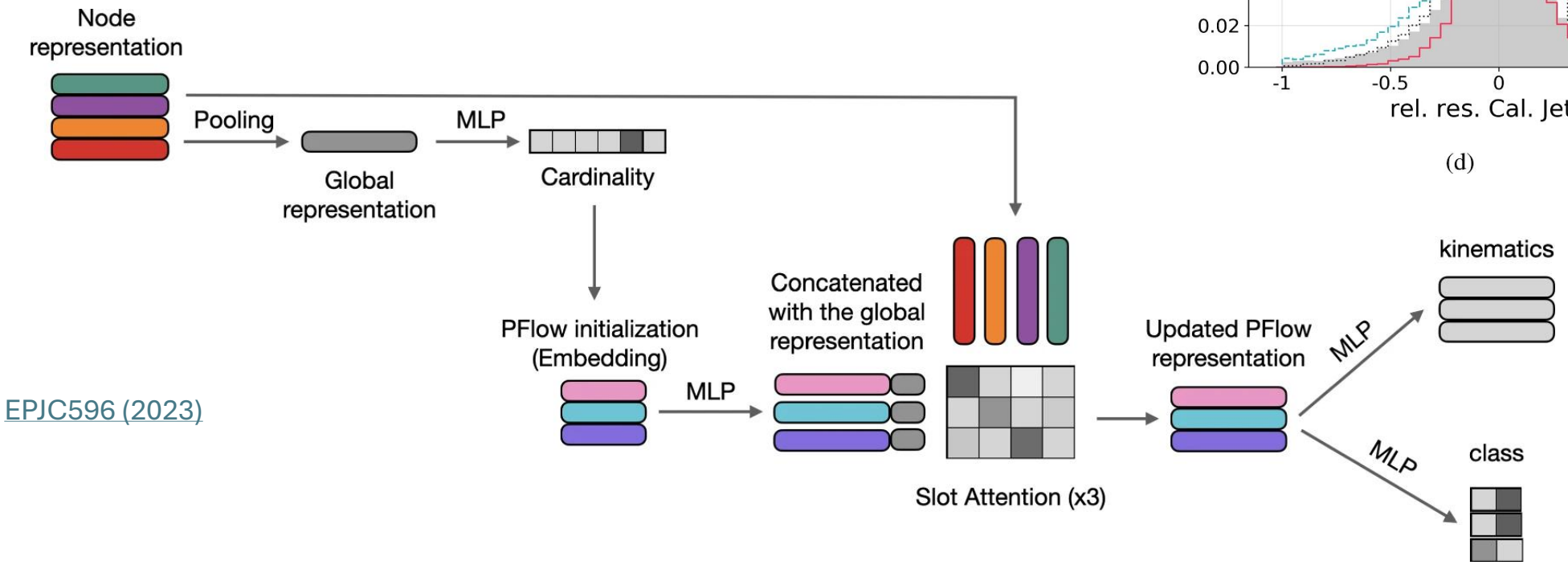


Machine learning for particle reconstruction

- Can we include inductive bias from the simulation in an explicit form?
- Starting point was to build a cross-attention between particles and input objects



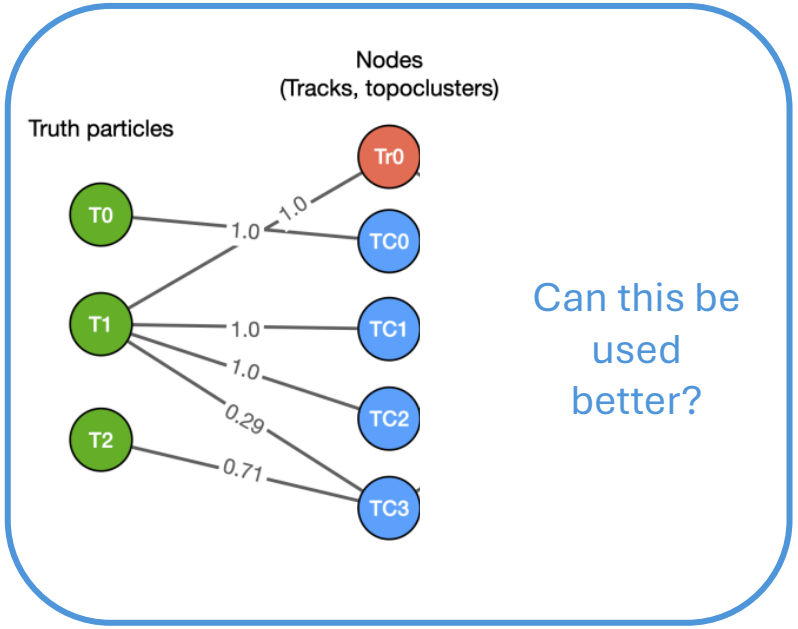
(d)

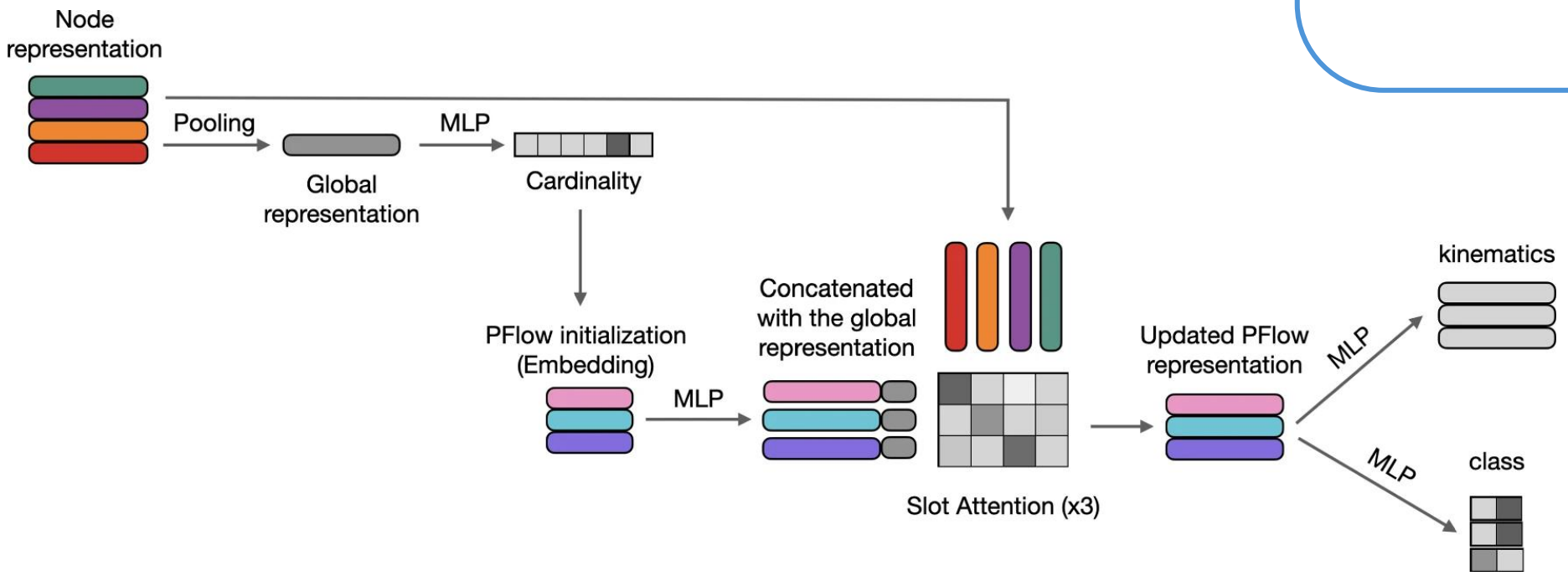


EPJC596 (2023)

Machine learning for particle reconstruction

- Can we include inductive bias from the simulation in an explicit form?
- Starting point was to build a cross-attention between particles and input objects



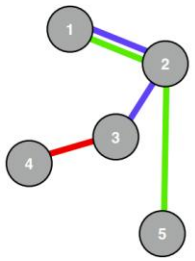


- Provided reasonble results but still non satisfactory. Can we build a model that levearges auxiliary information from the simulation and improves both results and interpretability?

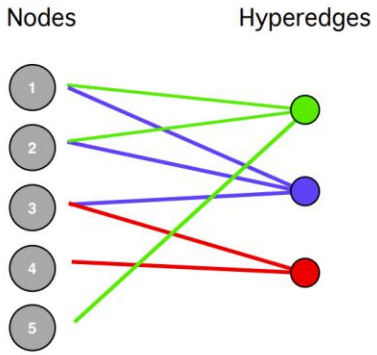
Hypergraps for particle reconstruction

- One possibility is to attack the problem as a search for hyperedges in a graph formed by the detector outputs

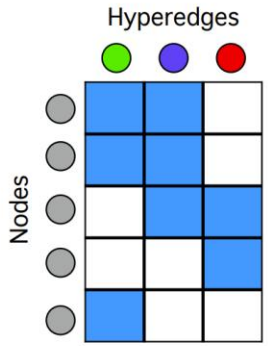
$$I_{ia} = \frac{E_{ia}}{\sum_{\text{particles } b} E_{ib}} = \frac{E_{ia}}{E_i}$$



Hypergraph

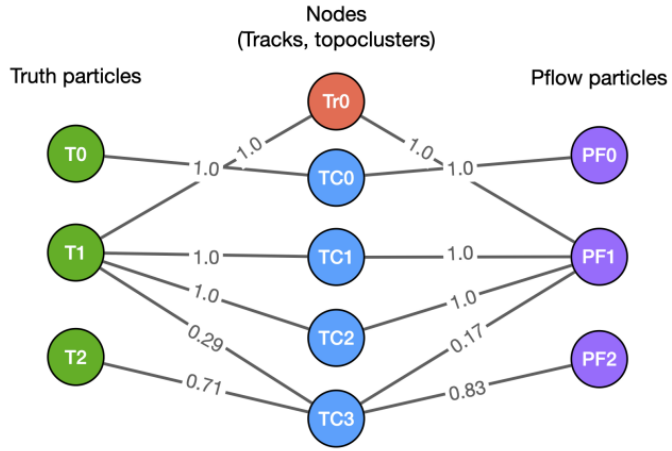


Bipartite graph



Incidence matrix

Learnable!



Initial proxy for particle properties

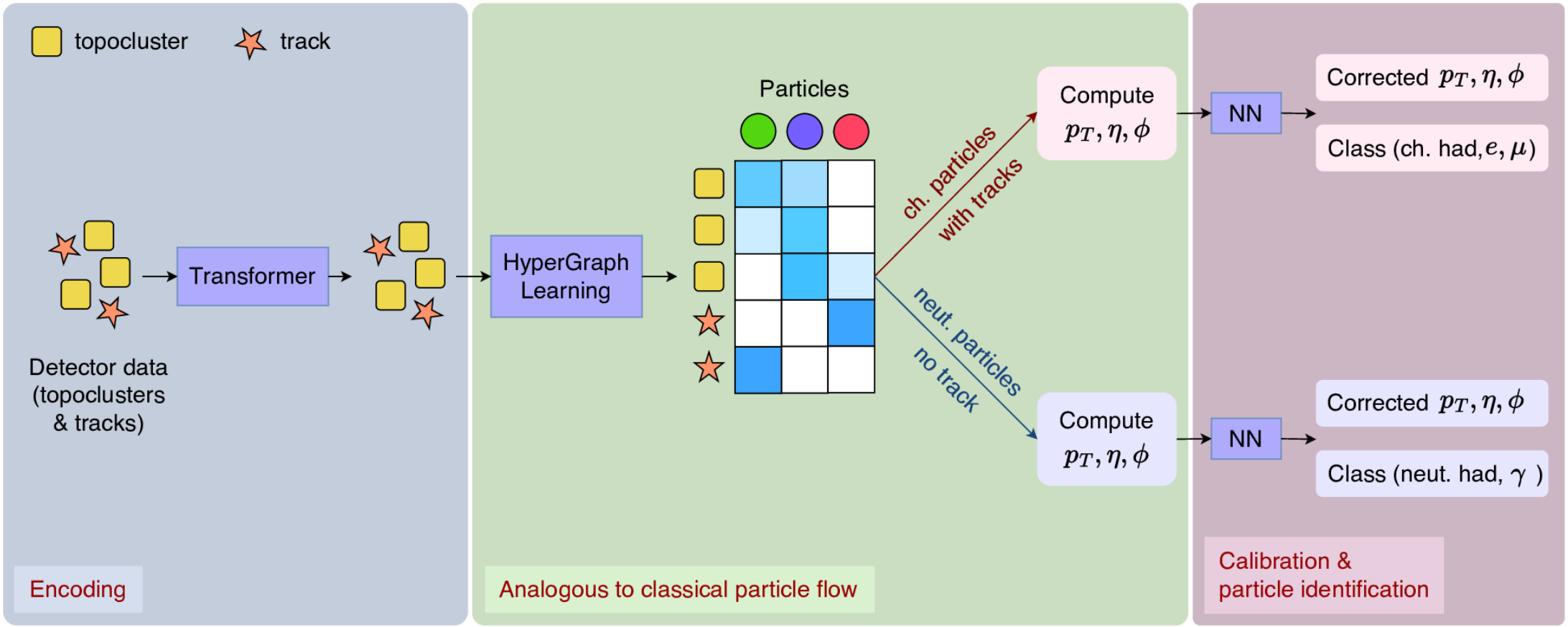
$$\hat{E}_a = \sum_i I_{ai} \cdot E_i$$

$$\{\hat{\eta}_a, \hat{\phi}_a\} = \frac{1}{\hat{E}_a} \sum_i I_{ai} \cdot E_i \cdot \{\eta_i, \phi_i\}$$

Images from: HGPFlow

HGPflow

- All in all this is what it looks like. The full model has around 2M training parameters
- Additionally, there is a maximum number of particles, and a score is associated to each of them to determine whether it is a null candidate

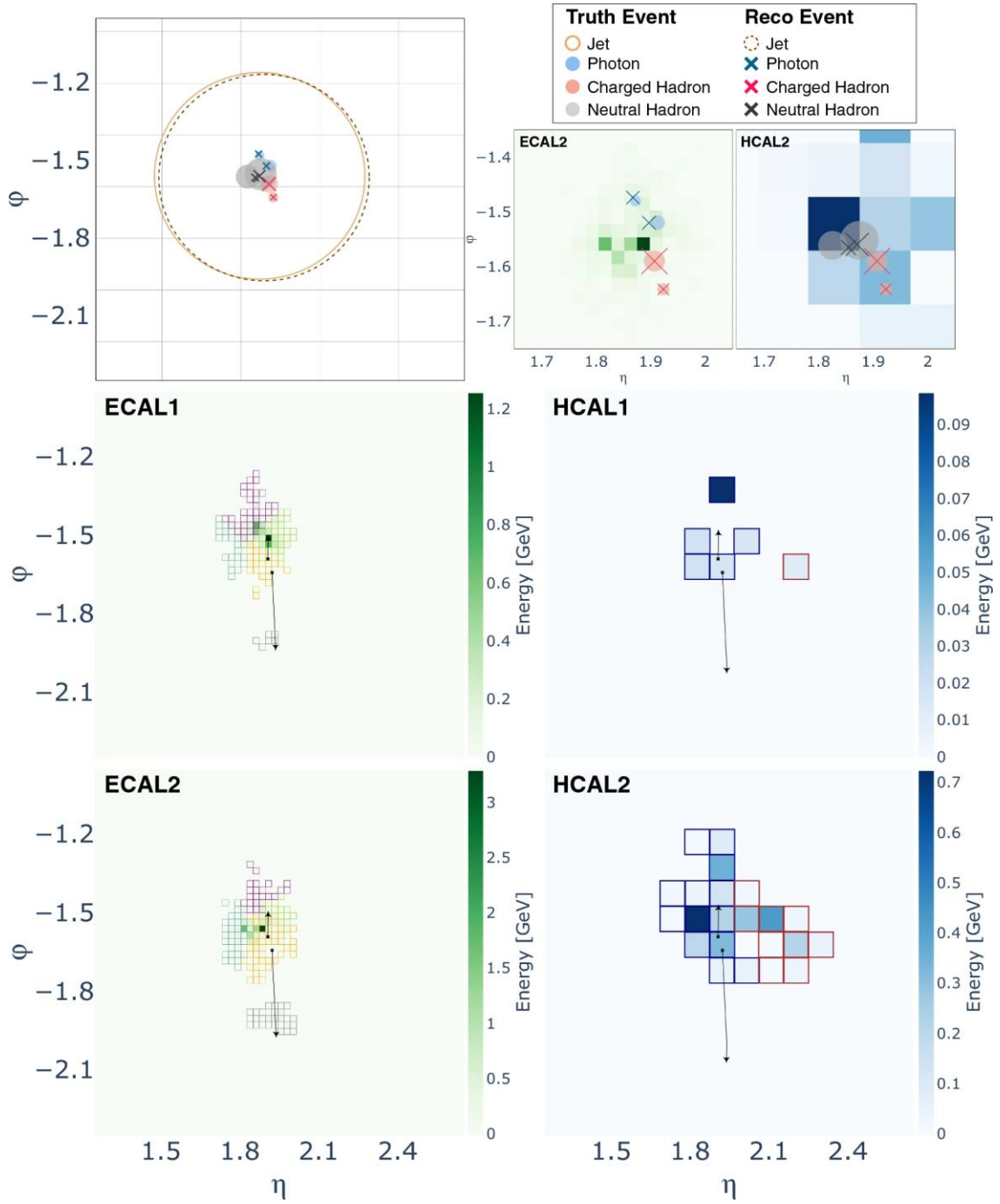


NB: there is a another set of inductive bias not yet explicitly used. e.g. - photons in jets

An event display

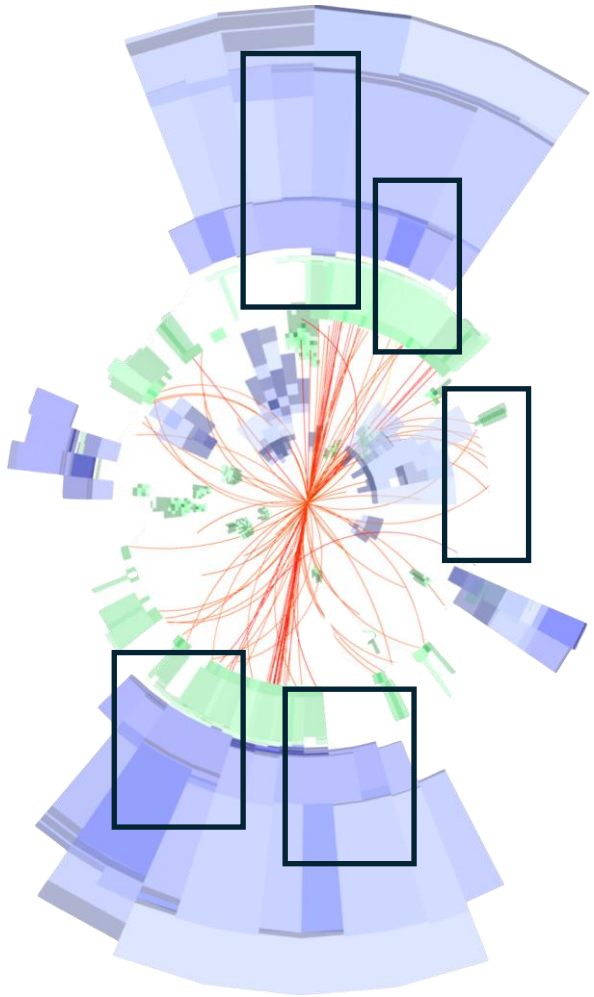
This worked well on single jets

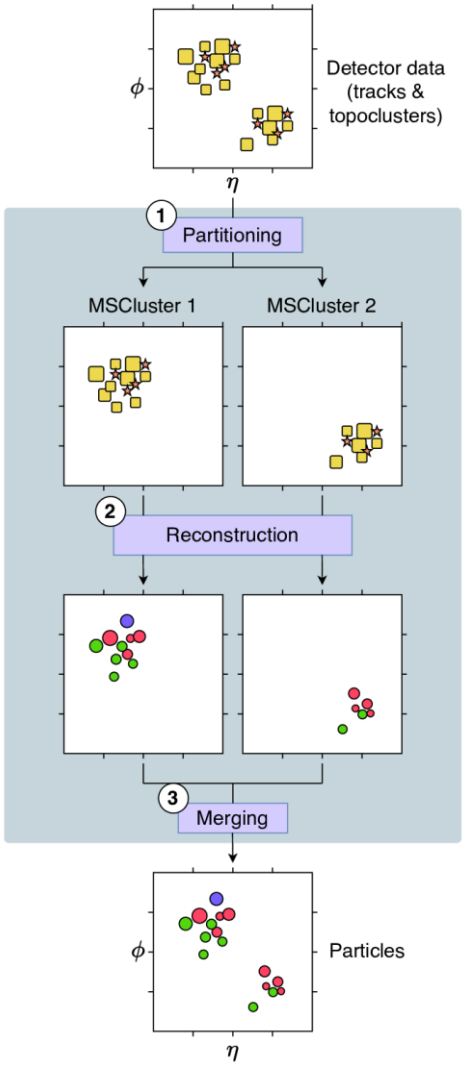
What about full event reconstruction?

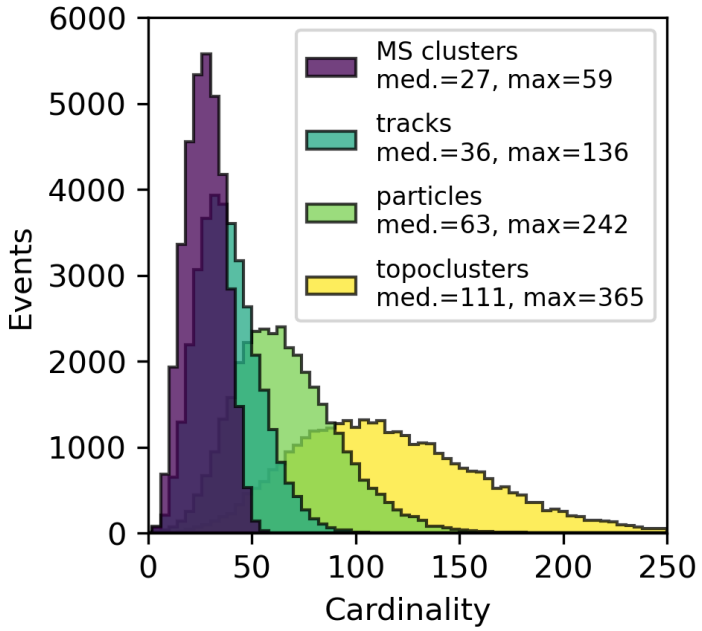


From single jet to full event reconstruction

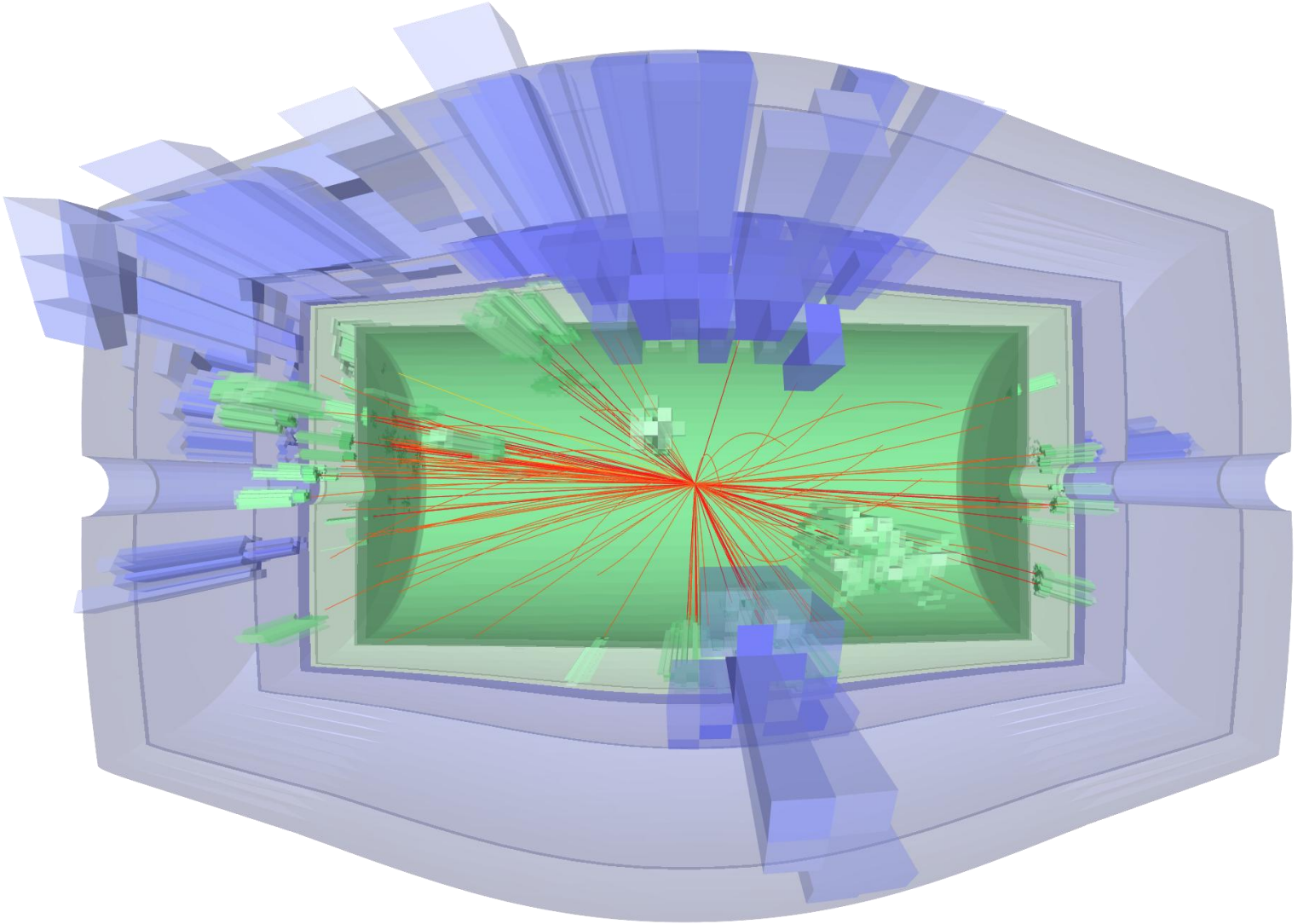
Event partition







The simulation: COCOA

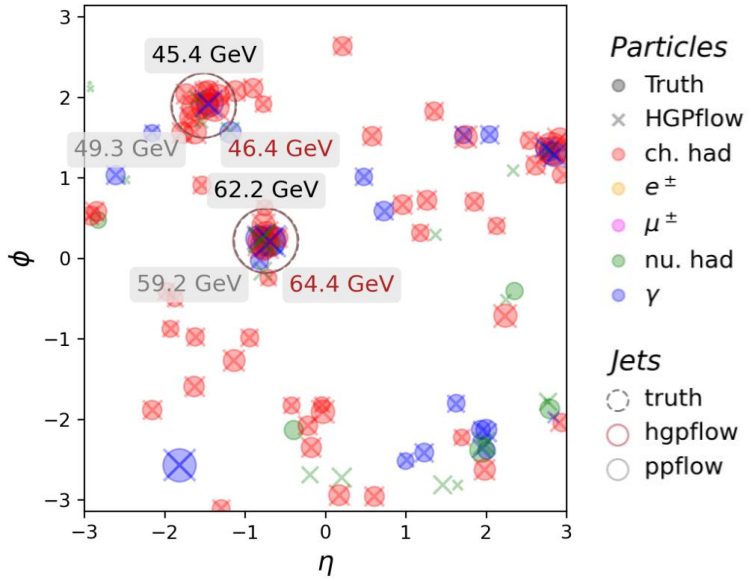


PU not included

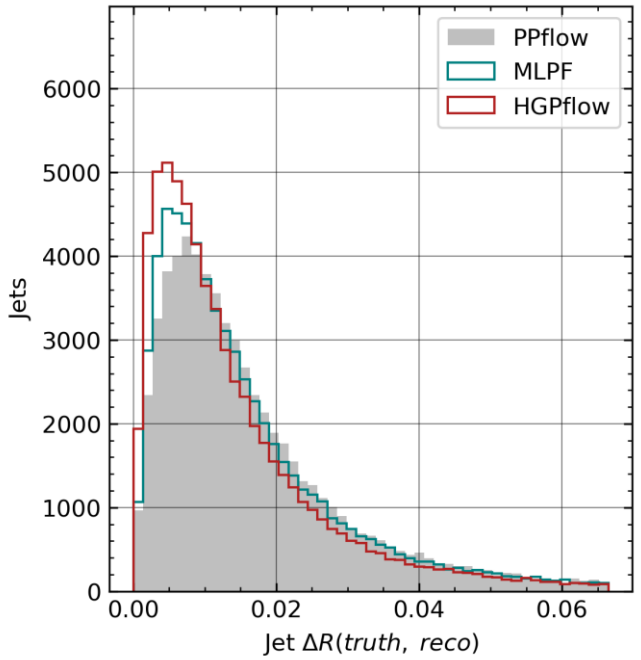
COCOA MLST(2023)

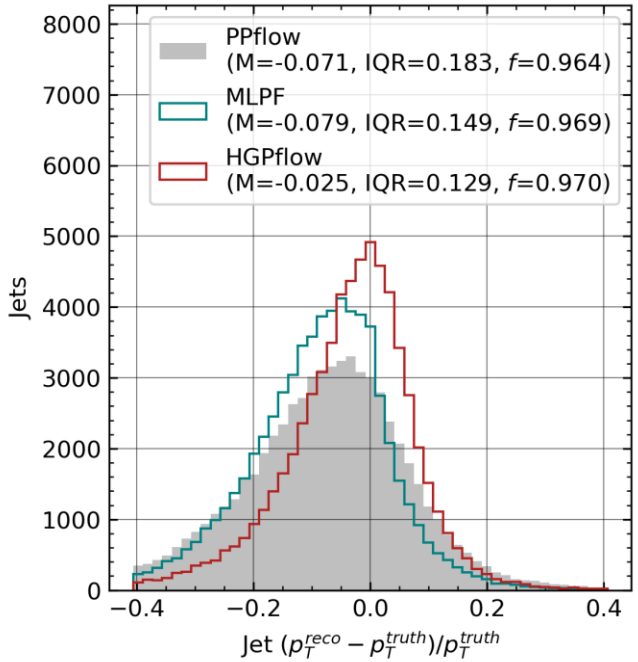
Results

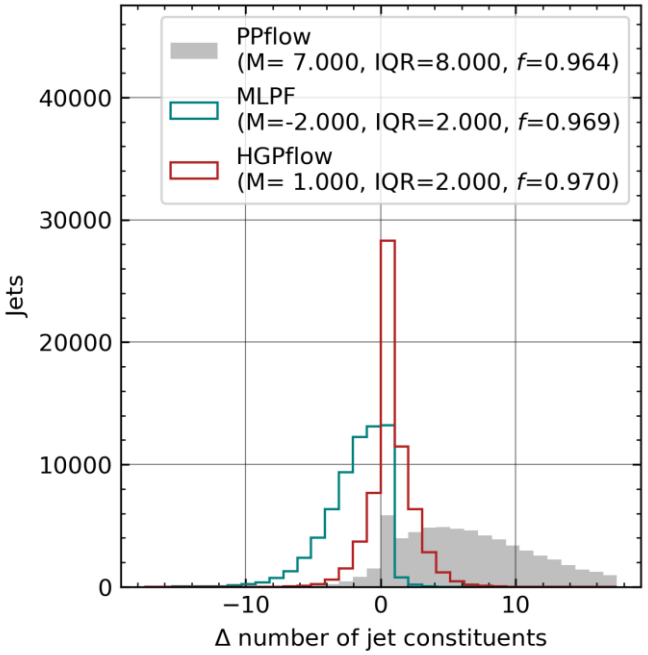
- Trained on QCD di-jet events (250k events)
- Evaluation on di-jet events
- But several topologies have been tested to check extrapolation results…



dijet

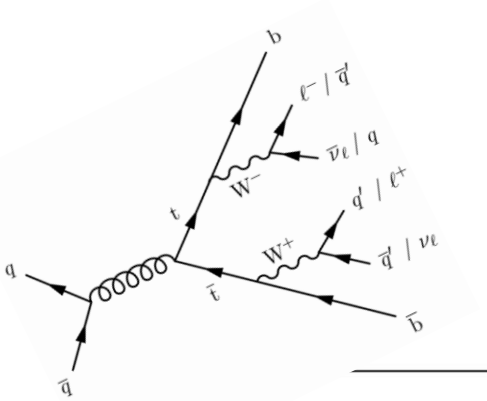






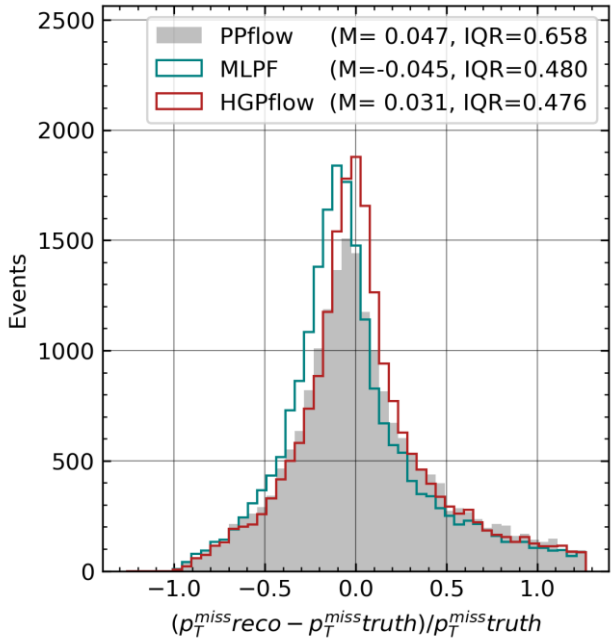
Results

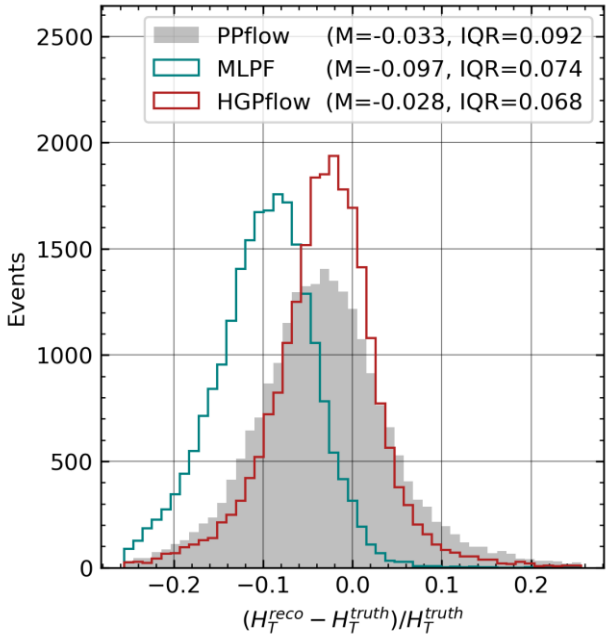
- Evaluation on ttbar events, train still on QCD events
- Missing momentum and scalar sum. Looks promising for HGPflow
- Role of indicutive bias? The more physics is given, the better the generalization
- Similar results on boosted Higgs observed

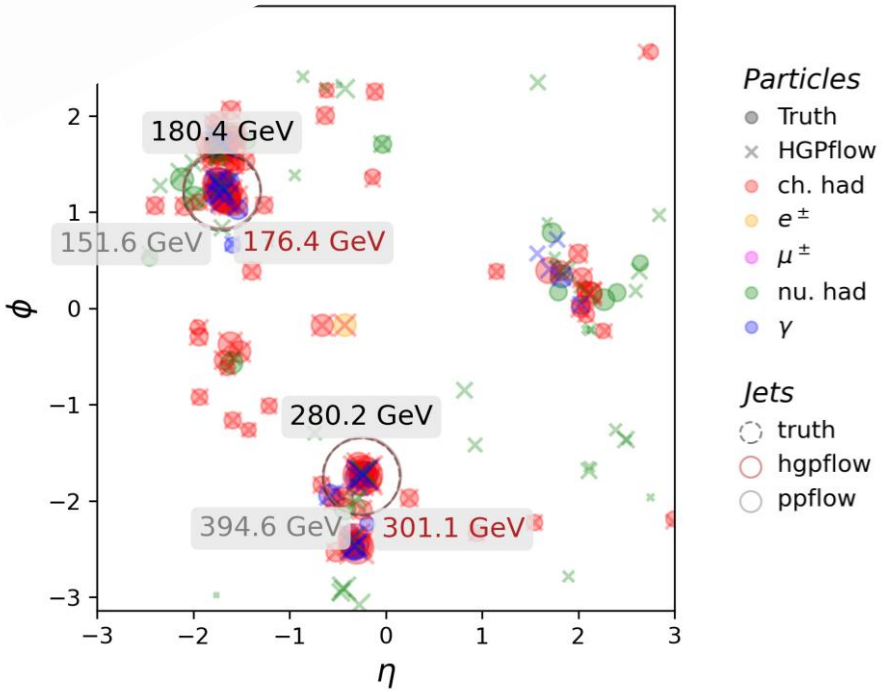


$$p_{\mathrm{T}}^{\mathrm{miss}} = |\sum_i \vec{p}_{\mathrm{T}}{}^i|$$

$$H_{\mathrm{T}} = \sum_i |\vec{p}_{\mathrm{T}}{}^i|$$





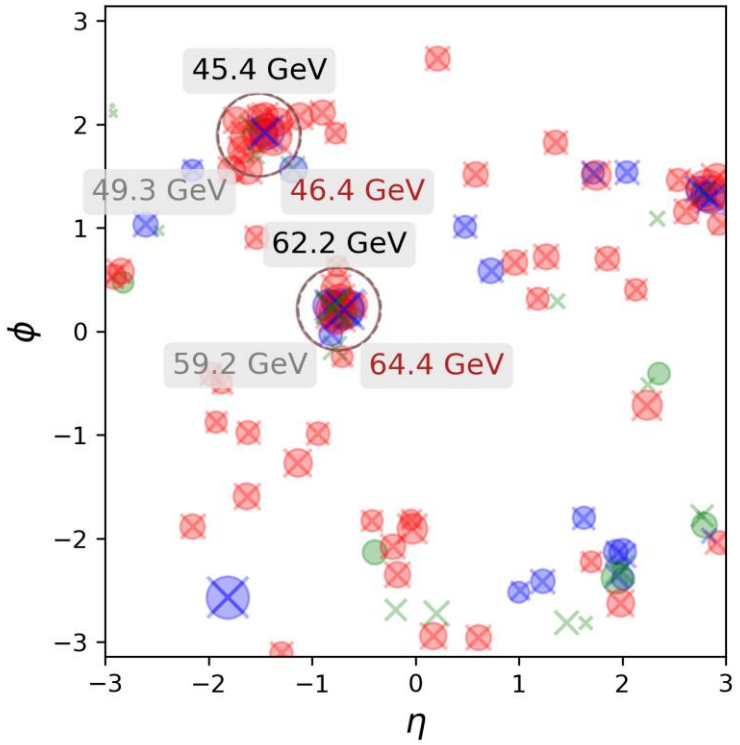


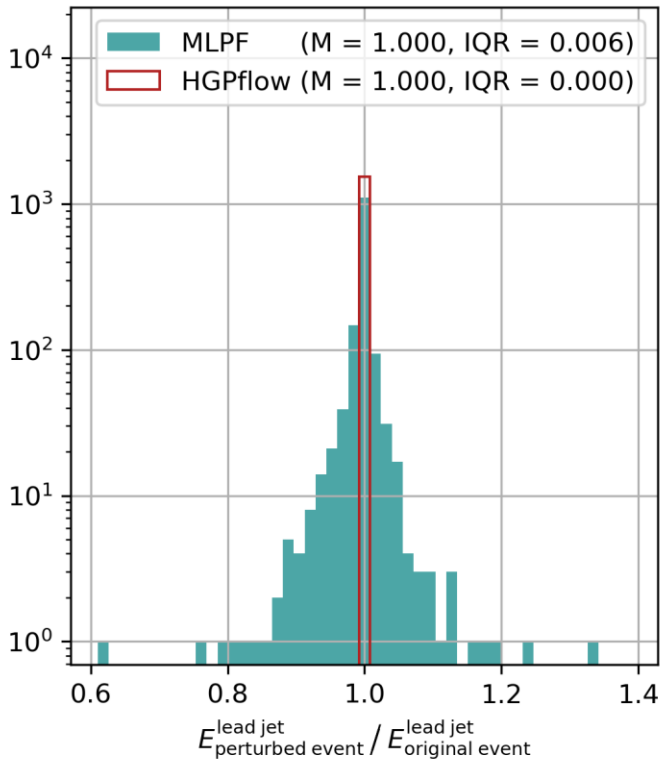
Not seen during training!

Results

- Locality – any particle reconstruction algorithms shall not depend on long-range correlation among physics object in the event -
- Perturbate one jet in a di-jet event, and check if the other is also affected by the change
- Partitioning ensures that by definition. Not totally true for algorithms processing full events

dijet





Conclusion

- Some promising developements. Results for a CLIC detector also show similar conclusion.
- Still a lot to learn and investigate but there is a path forward.
- Generally speaking these algorithms called *Set-to-Set algorithms* and have large applicability in HEP: tracking, multi-vertexing, etc...
- One point of concern is how to handle the modelling with data: i.e. calibrations. Area of active research, some thoughts here: Algren et al (2025)

EPJC(2025)

HGPflow: Extending Hypergraph Particle Flow to Collider Event Reconstruction

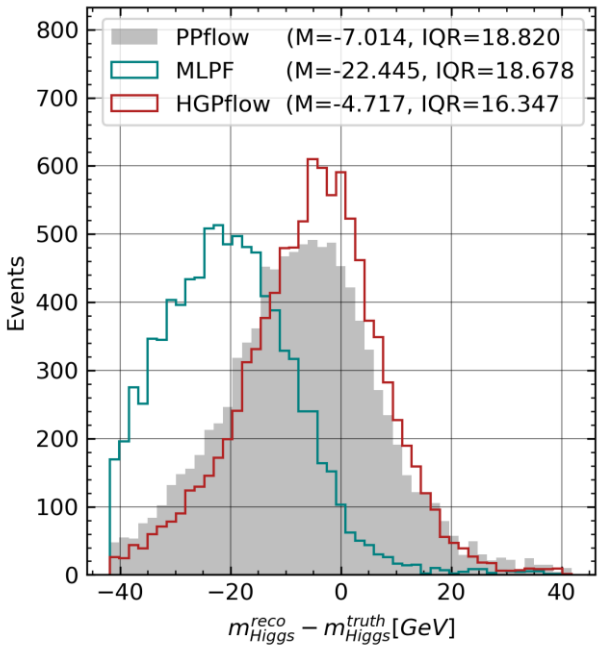
Nilotpal Kakati [1,a], Etienne Dreyer, Anna Ivina [1], Francesco Armando Di Bello [2], Lukas Heinrich [3], Marumi Kado [4], Eilam Gross [1]

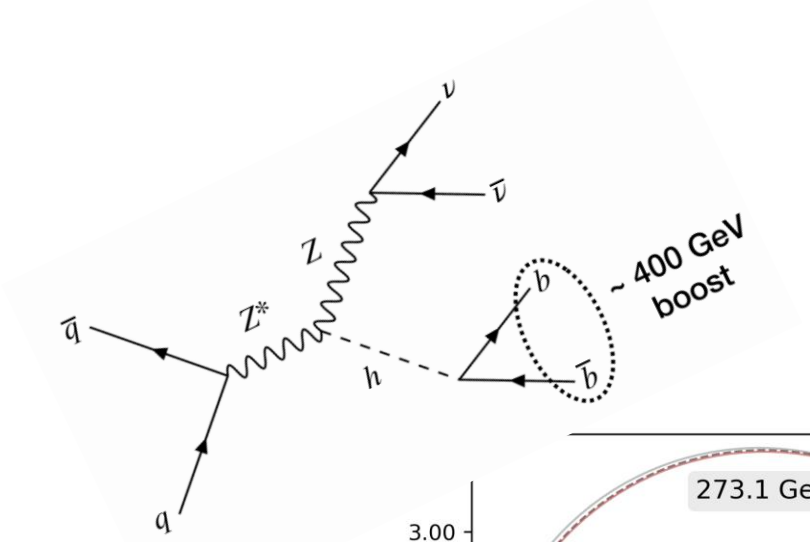
Many thanks to Nilotpal and Etienne, protagonists of this effort

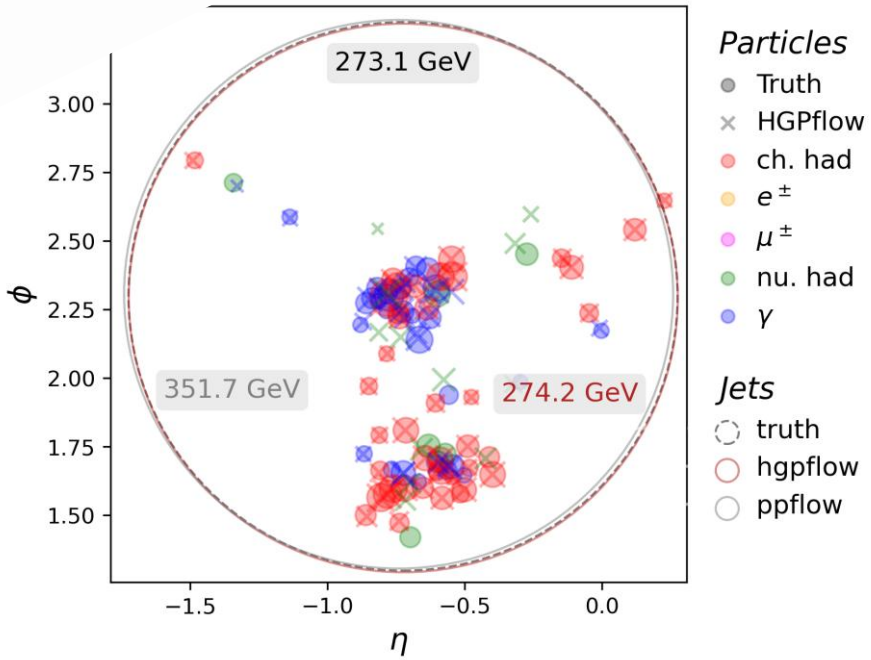
Results

- Looking at highly energetic Higgs boson

not visible during training!







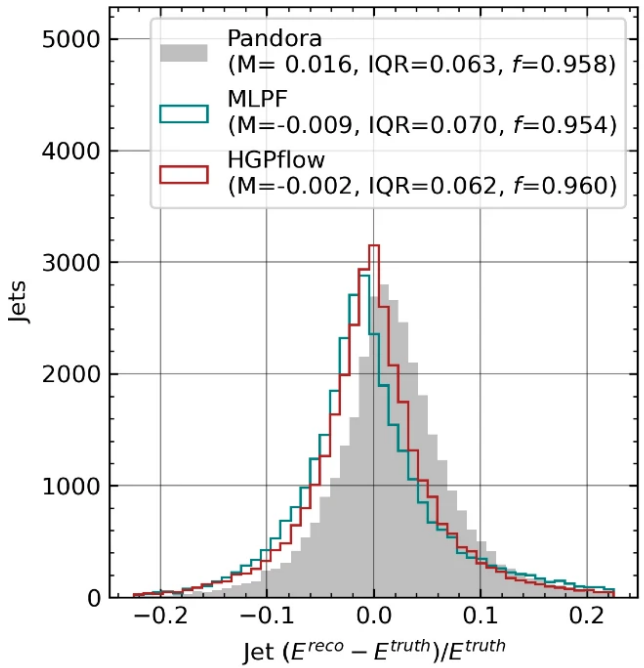
Training stat and architecture

Table 1 Samples used for training and performance testing

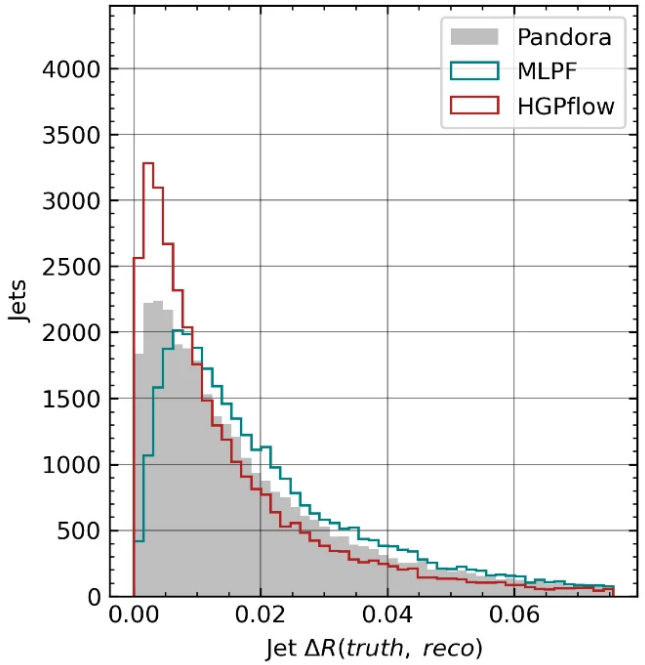
Detector	Process	Statistics		
		Train	Val	Test
COCOA	$p^+p^+ \to q\bar{q}$	250k	10k	35k
	Single π^+	–	–	30k / p_T bin
	$p^+p^+ \to t\bar{t}$	–	–	20k
	$p^+p^+ \to Z(\nu\bar{\nu})H(b\bar{b})$	–	–	10k
CLIC	$e^+e^- \to q\bar{q}$	1M	5k	20k

Optimizer	AdamW [55]
Learning rate (max)	0.001
Batch size –	
Stage 1	256
Stage 2	512
Number of epochs –	
Stage 1	100
Stage 2	30
Maximum number of hyperedges (K)	60
Embedded dimension	128
Number of iterative refinements	12
Number of trainable parameters	2.1M

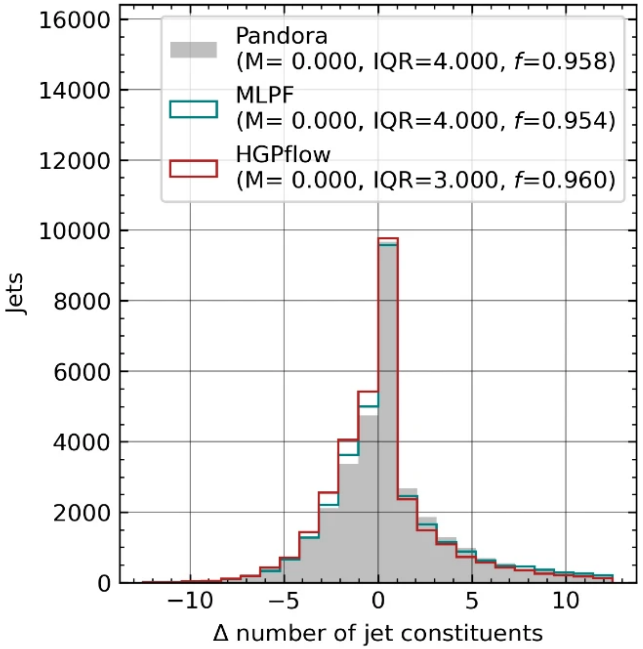
Results on CLIC



(a)

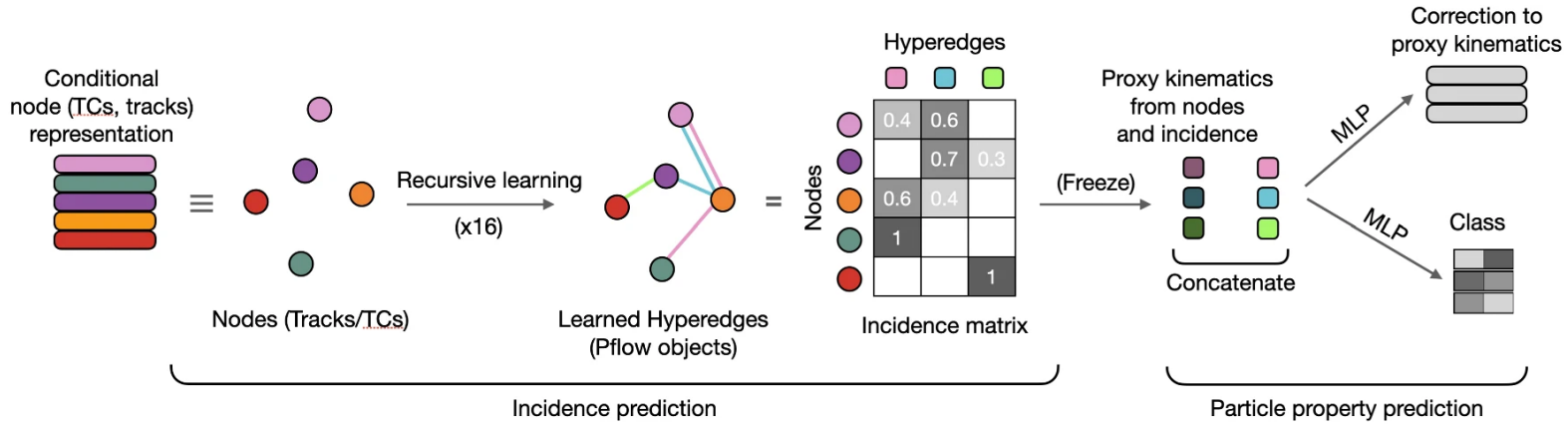


(b)



(c)

Hypergraph learning



$$I_{ia}^{t+1} = \phi_I \left(v_i^t, e_a^t, I_{ia}^t \right)$$

$$V^{t+1} = \phi_V \left(\left\{ v_i^t, \rho_{E \to V}(i,t), v^0 \middle| i = 1 \ldots n \right\} \right)$$

$$E^{t+1} = \phi_E \left(\left\{ e_a^t, \rho_{V \to E}(a,t) \quad \middle| a = 1 \ldots k \right\} \right)$$