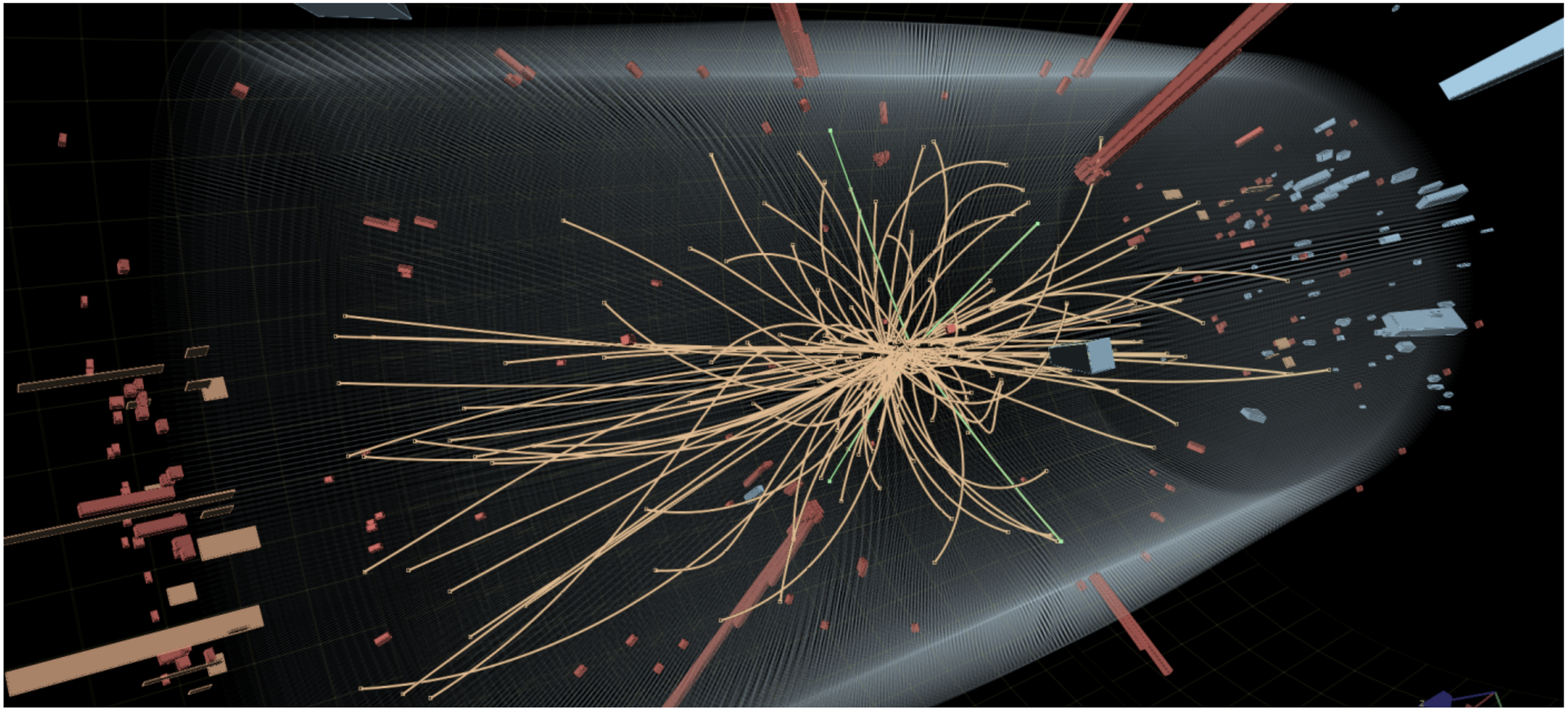
Deep Learning Applications for collider physics Lecture 5

Maurizio Pierini

Plan for these lectures

	Day1	Day2	Day3	Day4	Day5
1st hour	Introduction	ConvNN	LHC & fast Inference	Autoencoders	Graphs
2nd hour	Dense NNNs	GANs	RNNs	Anomaly Detection	Graphs
Tutorial	Dense NNNs	ConvNN	RNNs	Autoencoders	Graphs

European Research Council

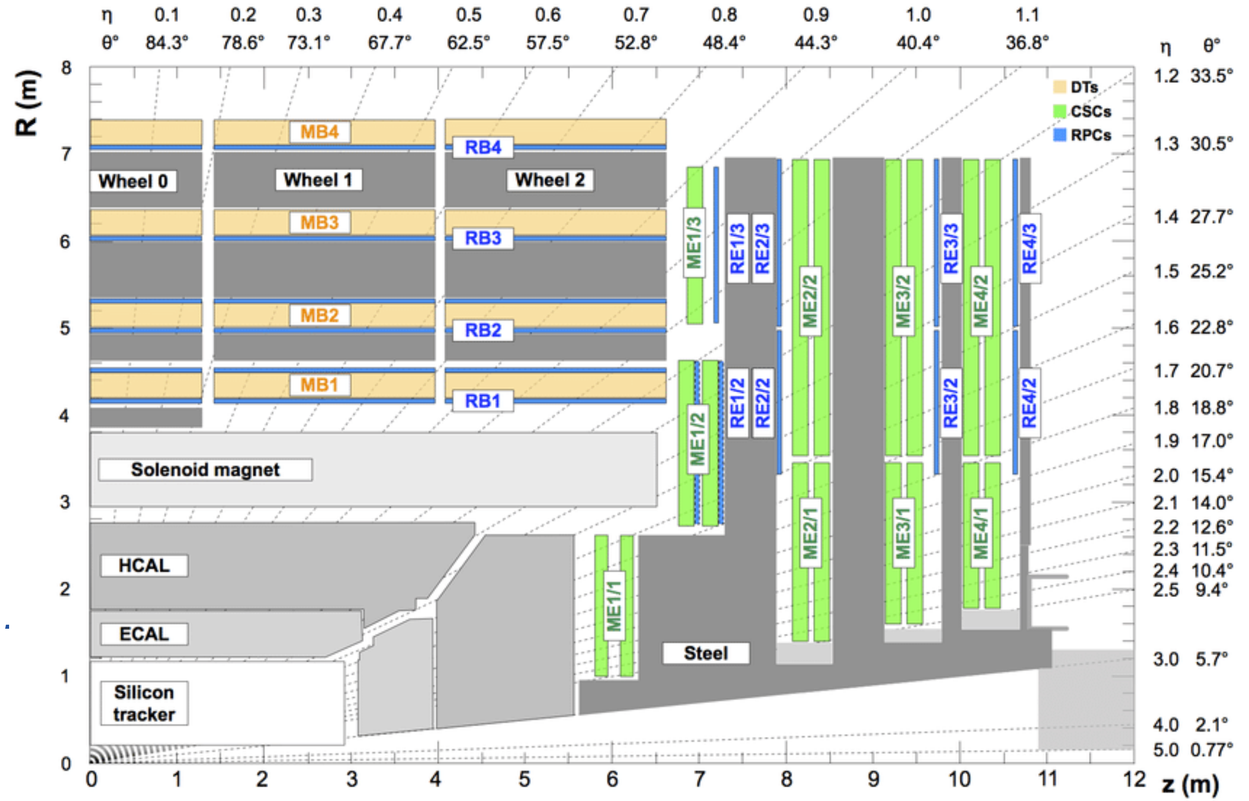


A (particle)physics friendly architecture

European Research Council

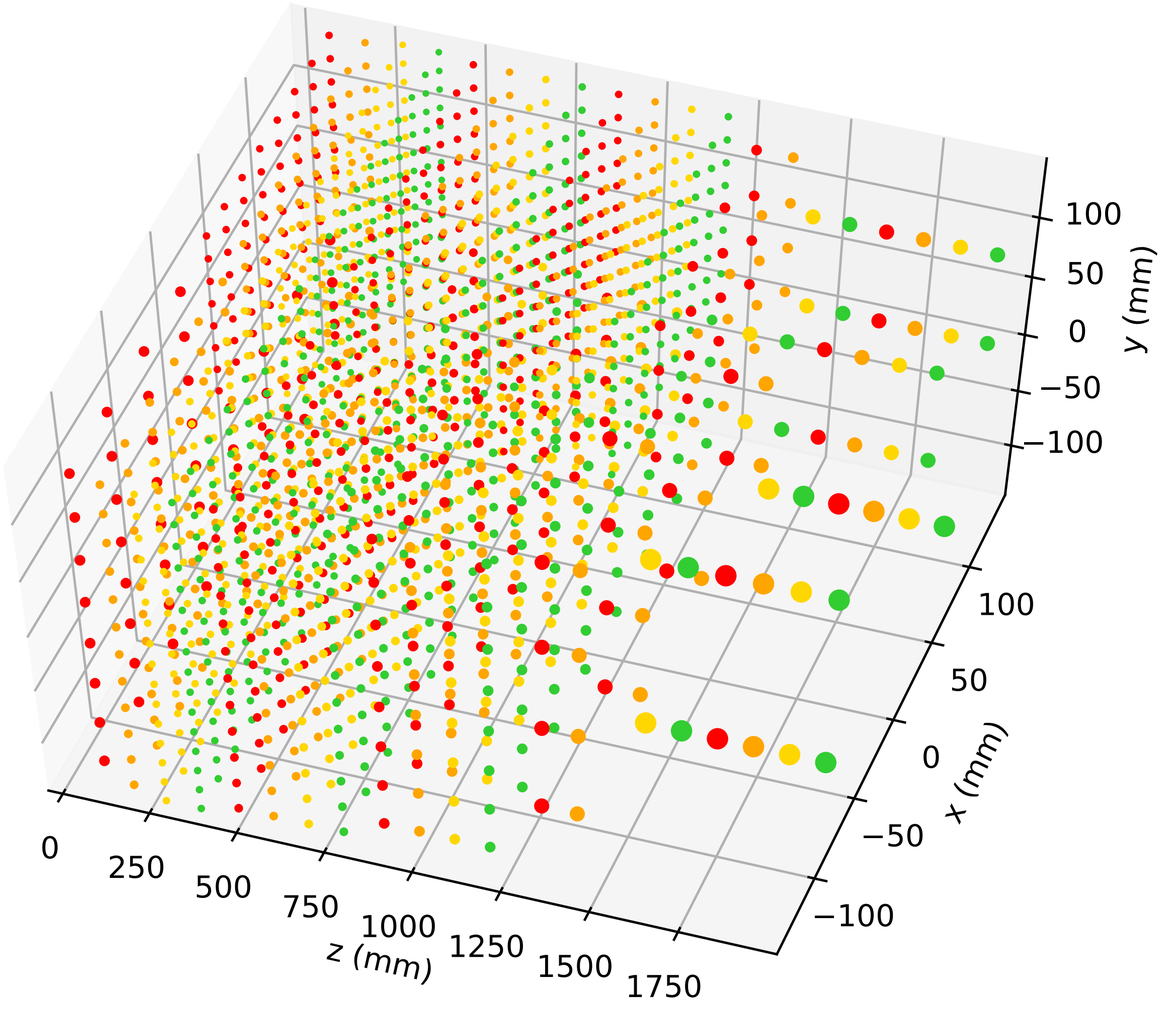
CNN and Collider Detectors

- *CNNs assume that our detectors are regular arrays of sensors*
- *Our detectors are not*
  - *different components with different technologies*
  - *some particle visible only to some part of the detector*
- *CNNs don't really fit the sparse of the collision data*
- *Instead, we think graph networks can work better*



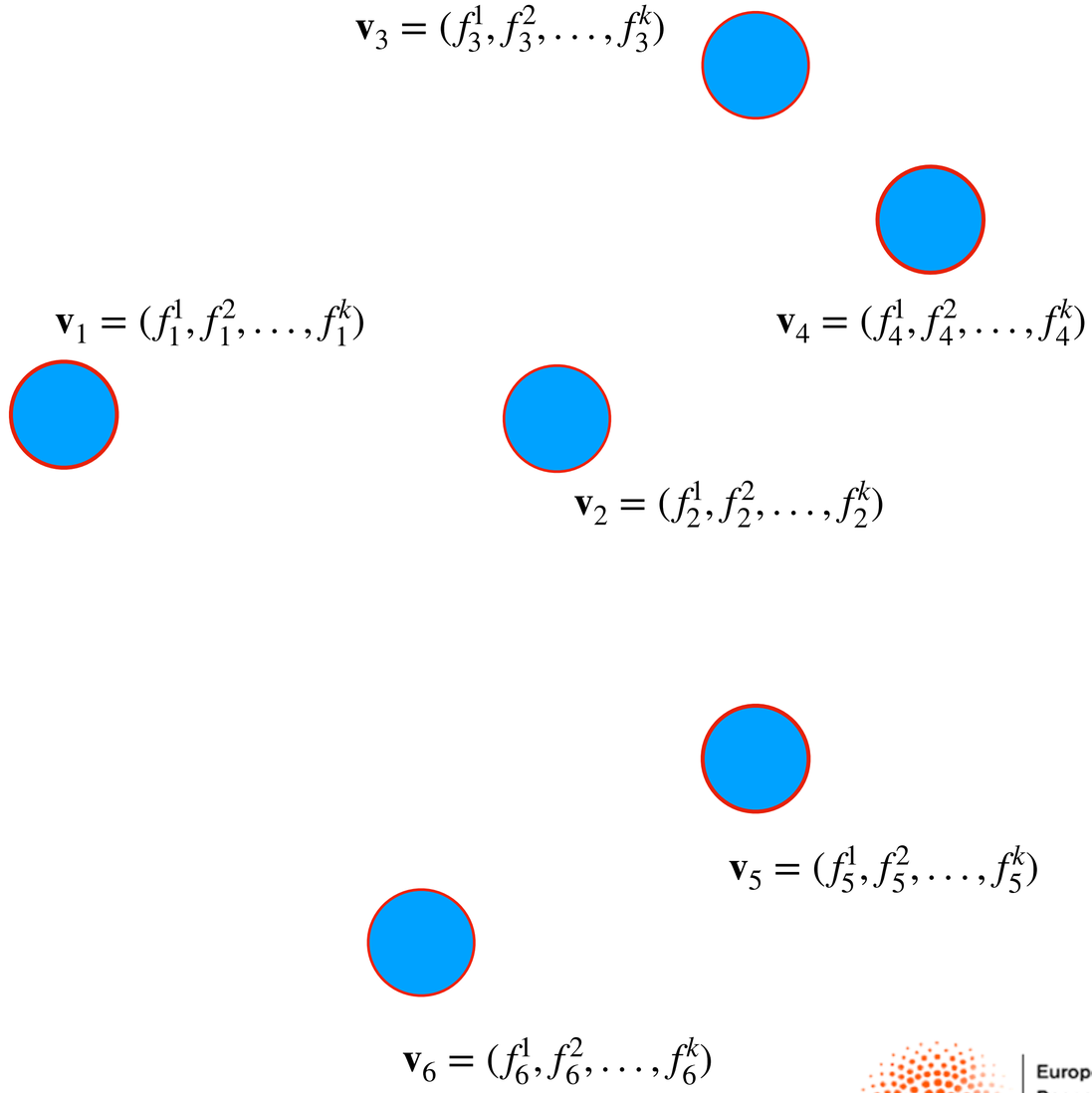
Pros & Cons of RNN

- *Pros: With RNN, one doesn't need to assume anything about the underlying geometry while still reaching good performance*
- *Cons: a few…*
  - *Need to specify an ordering*
  - *Works ok on particles (to be reconstructed) but not easy to generalise top detector hits (the real RAW data)*
  - *Sequential operations in inference -> might be problematic for low-latency applications*



Graph Networks

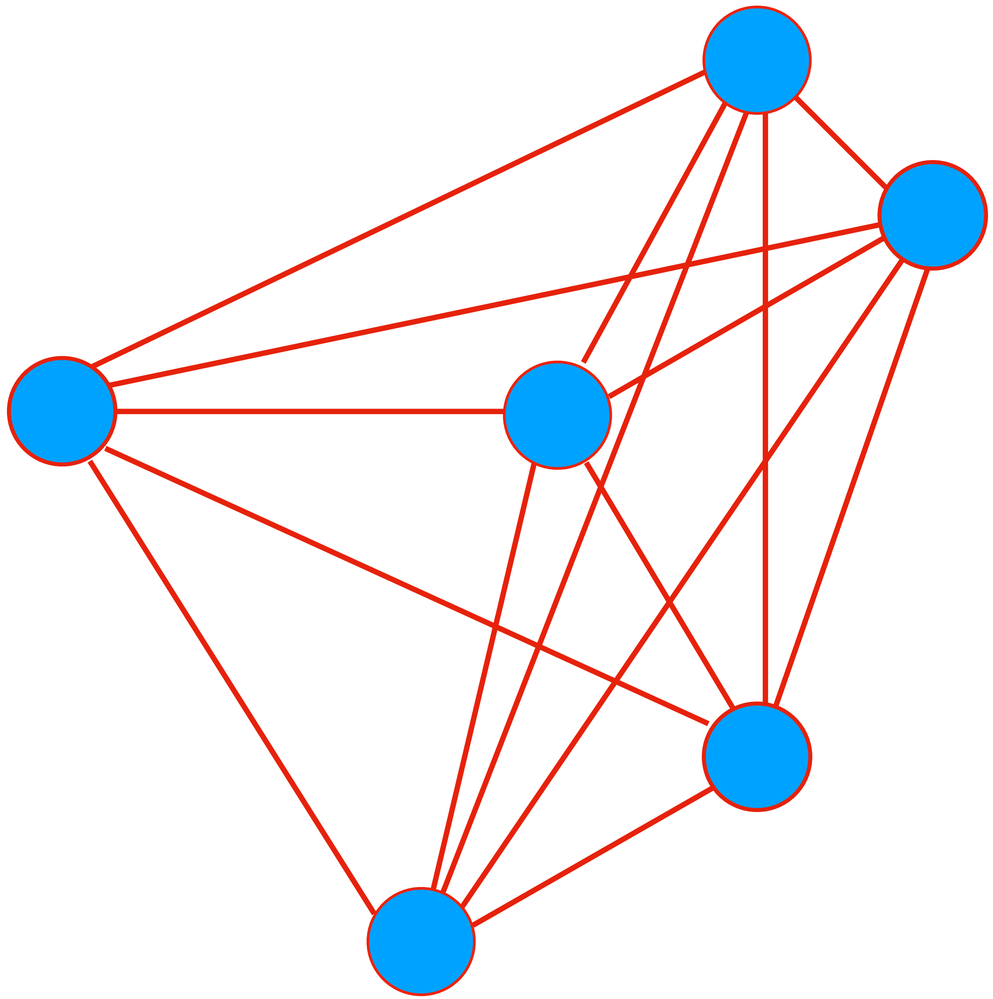
- *Graphs Nets are architectures based on an abstract representation of a given dataset*
  - ***Each example in a dataset is represented as a set of vertices***
  - ***Each vertex is embedded in the graph as a vector of features***
  - *Vertices are connected through links*
  - *Messages are passed through links and aggregated on the vertices*
  - *A new representation of each node is created, based on the information gathered across the graph*



https://arxiv.org/pdf/1704.01212.pdf

Graph Networks

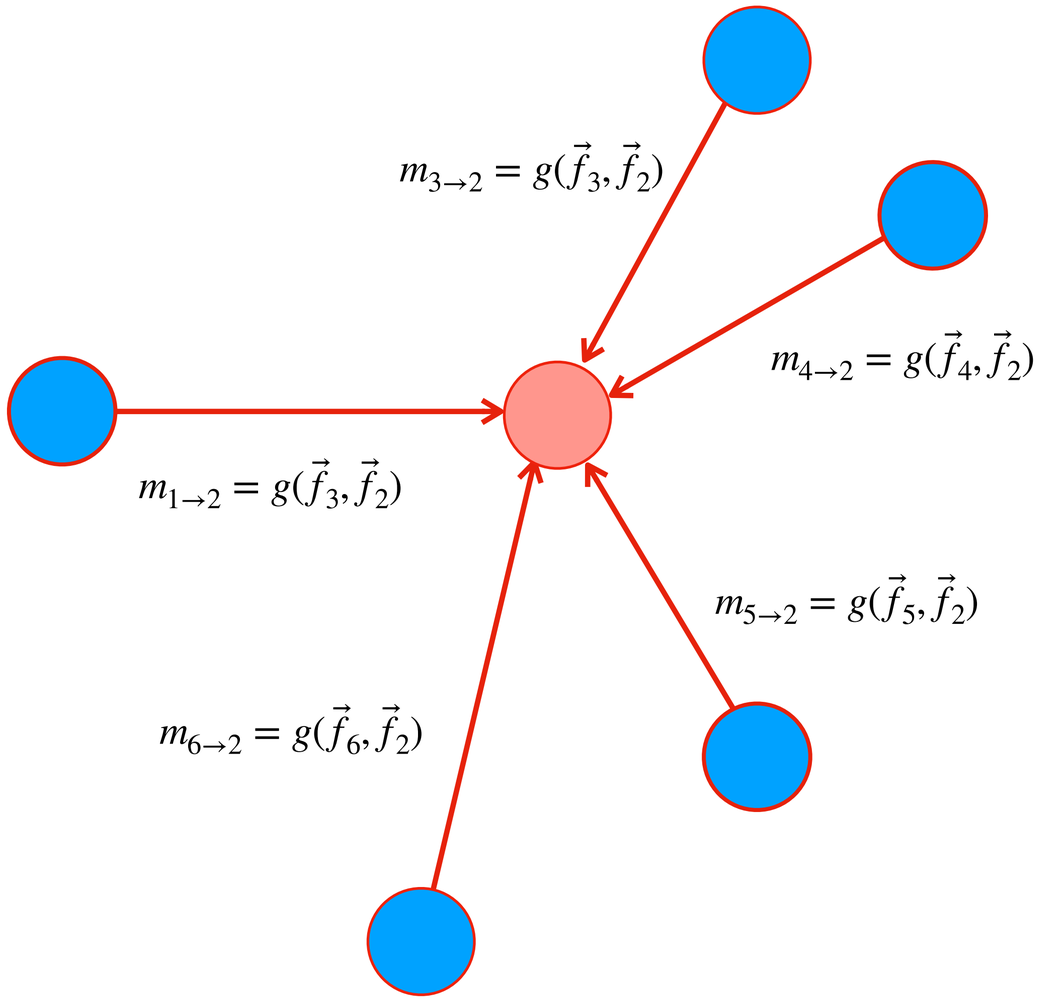
- *Graphs Nets are architectures based on an abstract representation of a given dataset*
  - *Each example in a dataset is represented as a set of vertices*
  - *Each vertex is embedded in the graph as a vector of features*
  - ***Vertices are connected through links***
  - *Messages are passed through links and aggregated on the vertices*
  - *A new representation of each node is created, based on the information gathered across the graph*



https://arxiv.org/pdf/1704.01212.pdf

Graph Networks

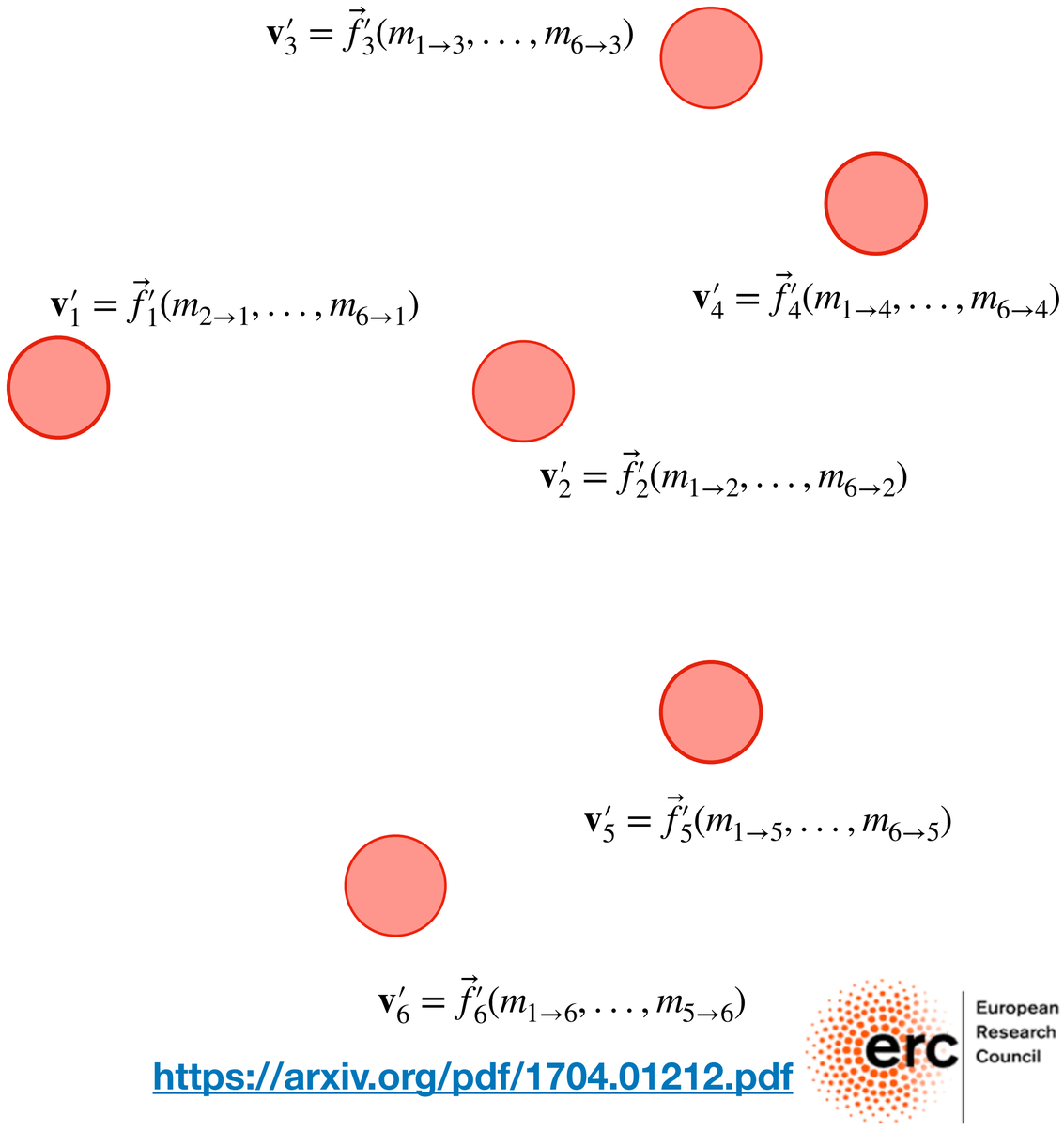
- *Graphs Nets are architectures based on an abstract representation of a given dataset*
  - *Each example in a dataset is represented as a set of vertices*
  - *Each vertex is embedded in the graph as a vector of features*
  - *Vertices are connected through links*
  - ***Messages are passed through links and aggregated on the vertices***
  - *A new representation of each node is created, based on the information gathered across the graph*



https://arxiv.org/pdf/1704.01212.pdf

Graph Networks

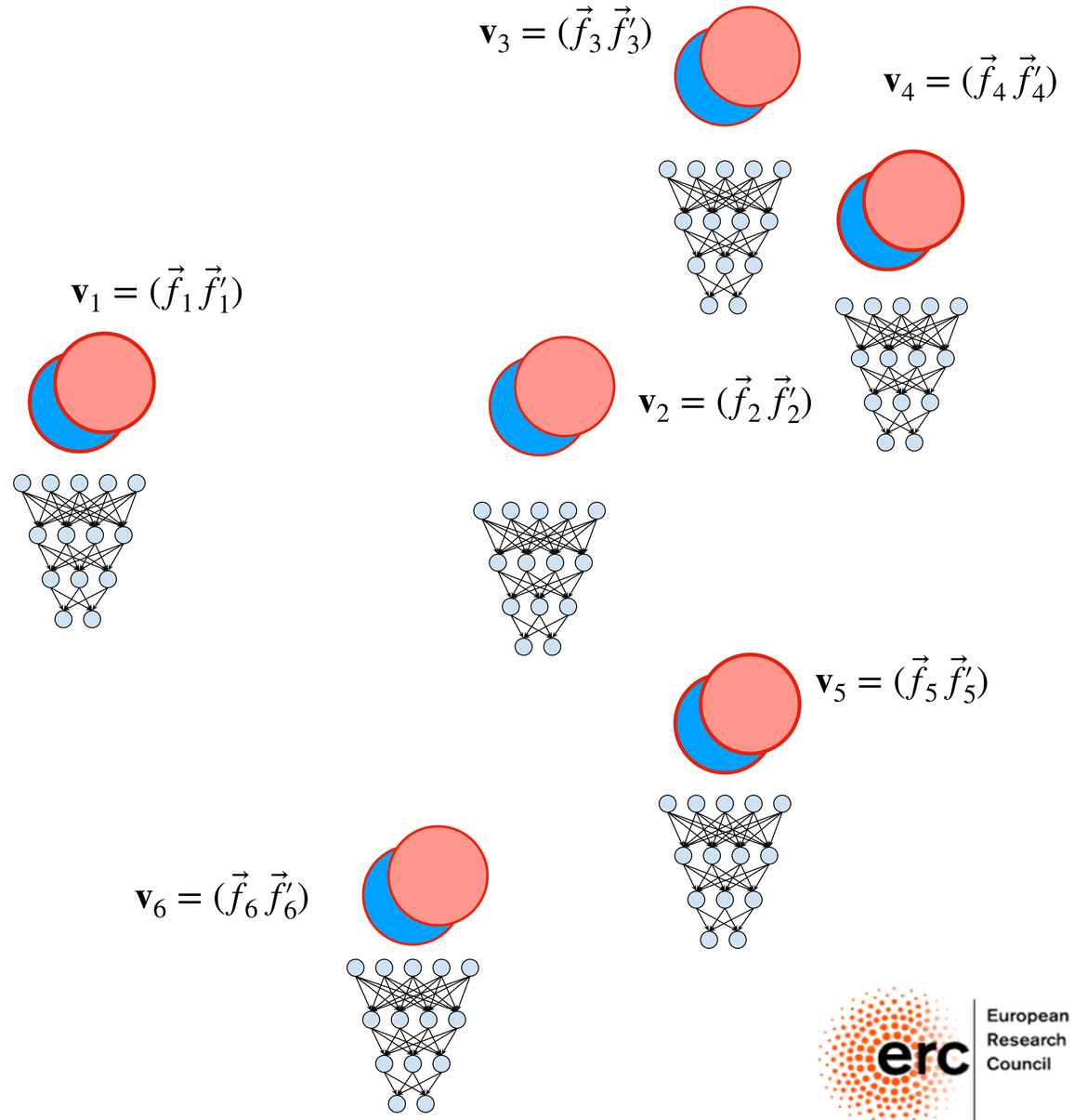
- *Graphs Nets are architectures based on an abstract representation of a given dataset*
  - *Each example in a dataset is represented as a set of vertices*
  - *Each vertex is embedded in the graph as a vector of features*
  - *Vertices are connected through links*
  - *Messages are passed through links and aggregated on the vertices*
  - ***A new representation of each node is created, based on the information gathered across the graph***



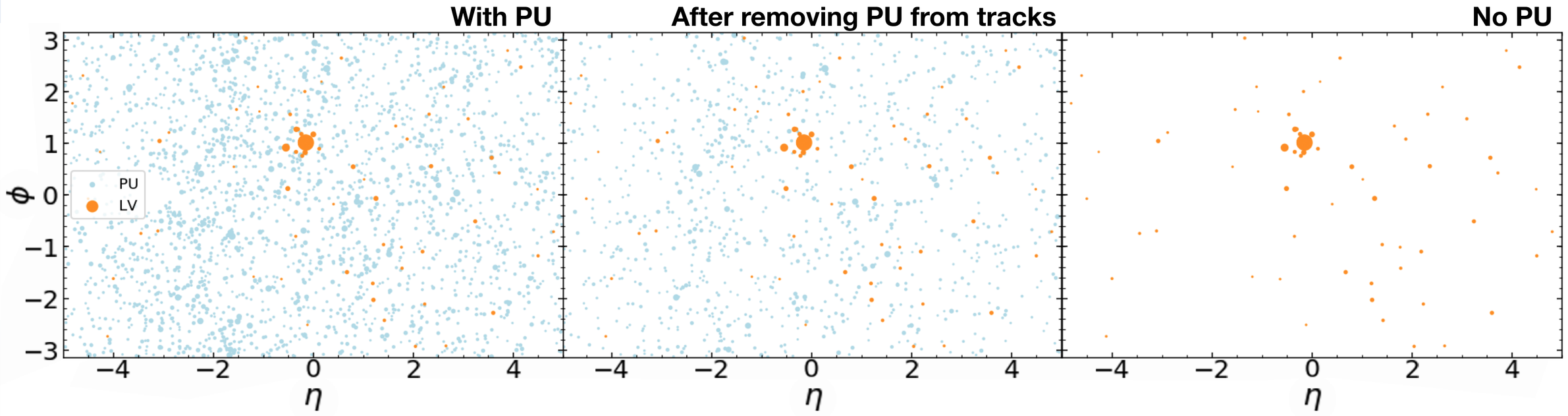
https://arxiv.org/pdf/1704.01212.pdf

The inference step

- *The inference step usually happens on each vertex*
- *But, depending on the problem, it might happen across the graph*
- *Usually, this is done with a DNN taking*
  - *the initial features fi*
  - *the learned representation fi'*
  - *[optional] some ground-truth label (for classifiers)*



Applications: pileup removal

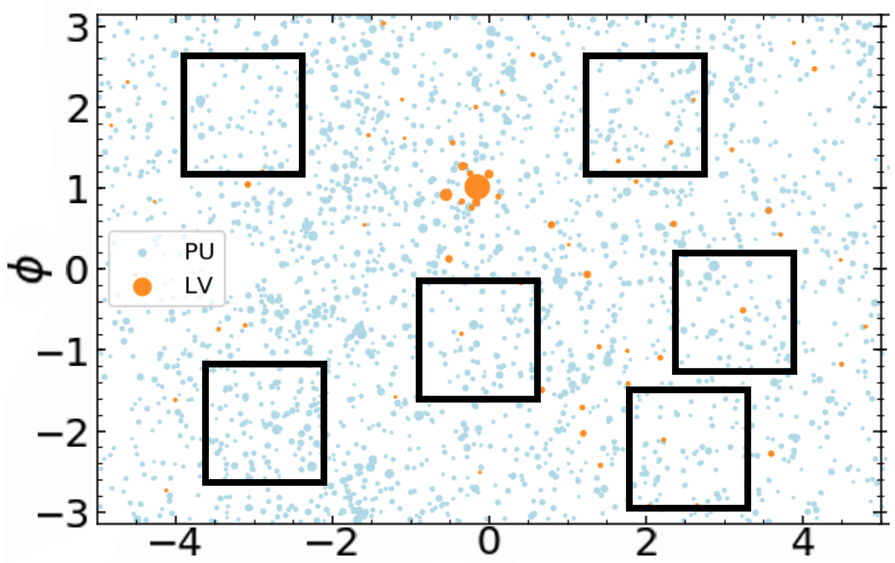


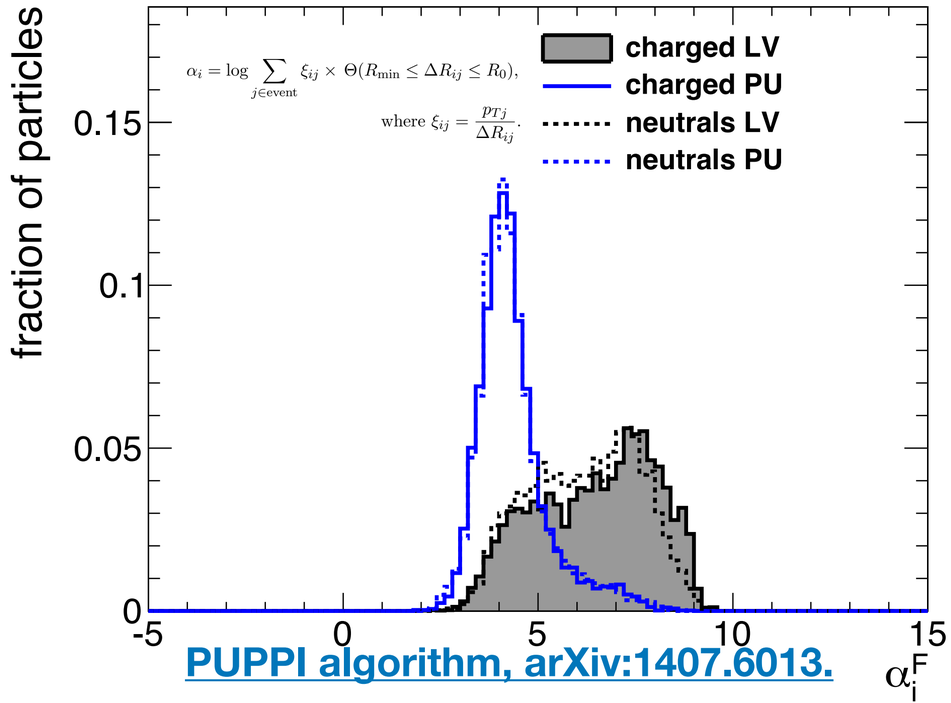
- *Pileup particles tend to spread across the full detector*
- *Pileup introduces noise to any reconstruction algorithm, so we want to remove it*
- *Easy for charged particles, since we can be tracked back to the vertex*
- *What about the neutrals?*

European Research Council

Applications: pileup removal

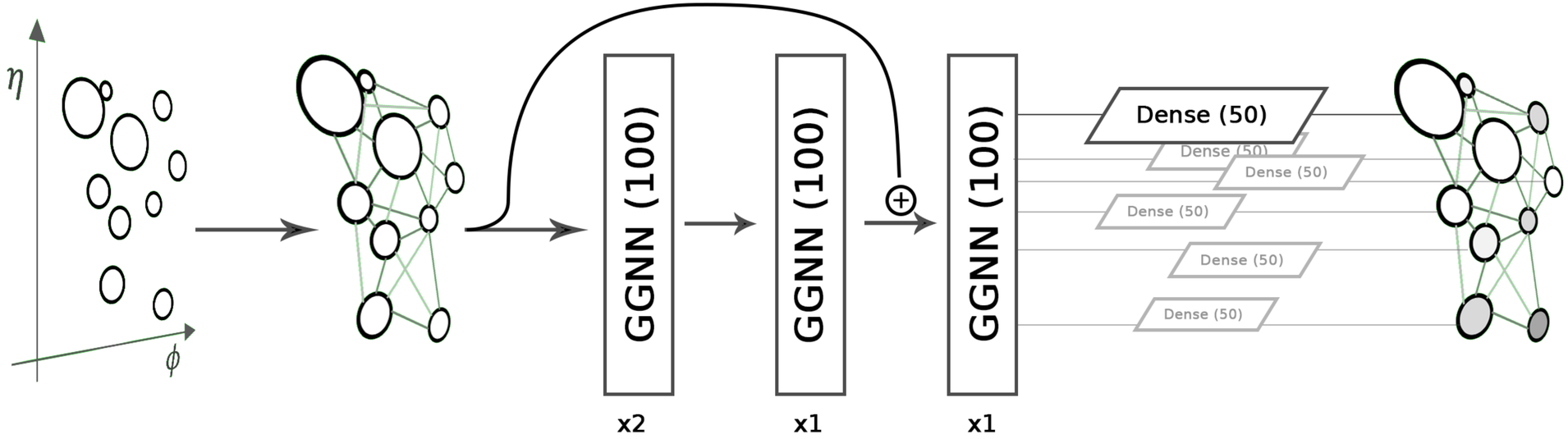
- *Interesting event will produce localised clusters of particles*
- *This difference is normally exploited to remove pileup*
  - *by computing the **average energy density** and subtracting it*
  - *by computing high-level features to select on*





PUPPI algorithm, arXiv:1407.6013.

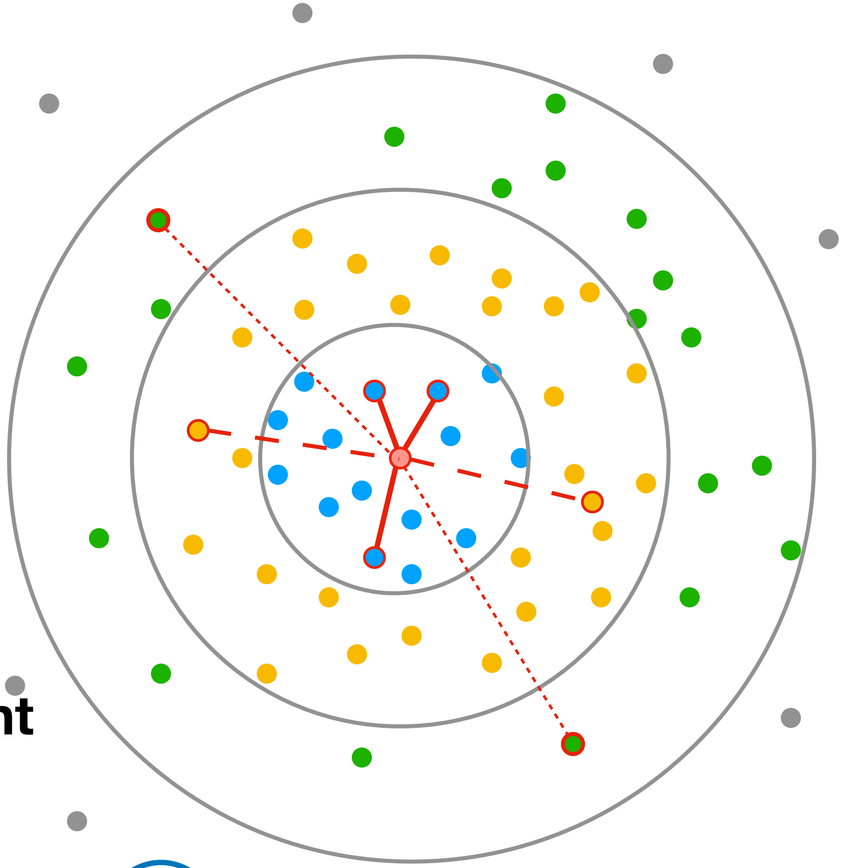
Graph Networks for PU removal



- *Use Gated Graph Neural Network (GGNN), a special kind of message-passing architecture*
  - *Start from a set of particles, each represented as a set of features h*
  - *Build the graph*
  - *Use the GRU sequential processing to "pass" the message and evolve it, depending on the next inputs*

Building the graph

- *Start with one particle (the red one)*
  - *Connect it to the closest ones (R<R0) with one kind of link*
  - *Connect it to the next-to-closest ones with a different kind of link*
  - *…*
- *Each link comes with a message*



gathered information

Messages

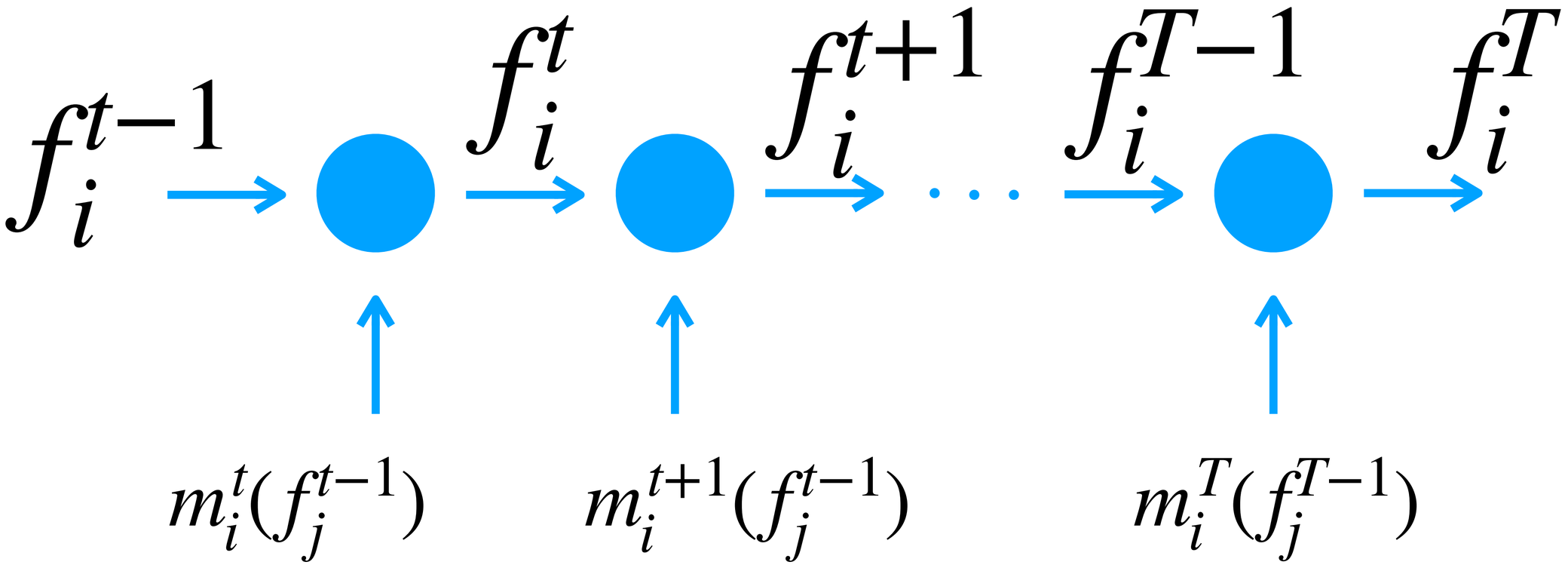
Learnable matrix (different for different kinds of links

$$m_j = \frac{1}{N}\sum_{i}^{N} m_{v,v_j} = \sum_{i} A \times f_j$$

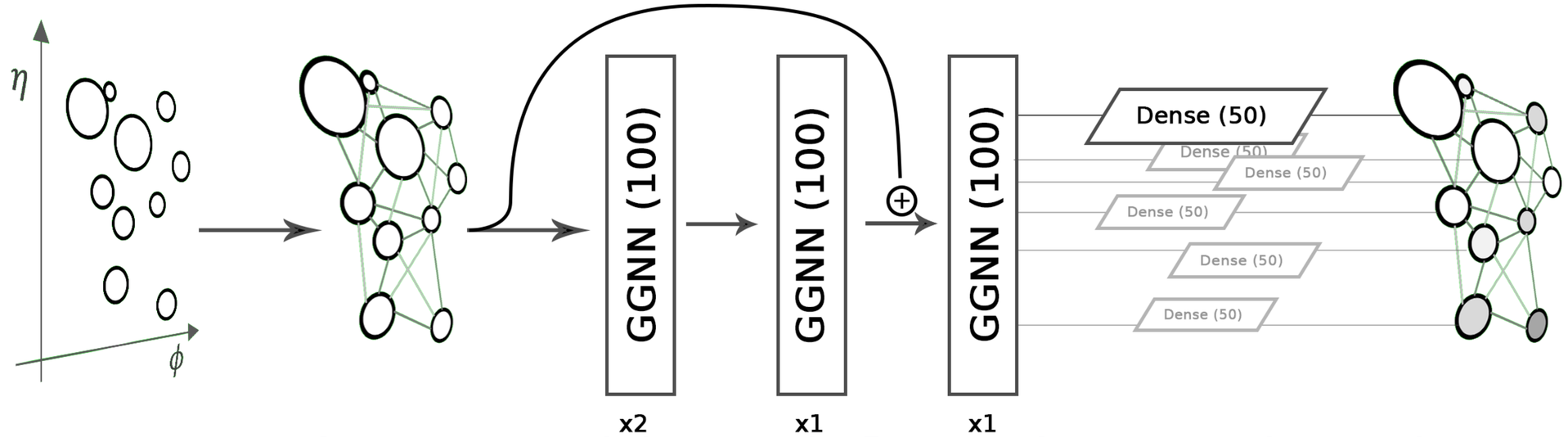
features of the j-th vertex

Multiple message passing

- *The procedure is repeated T times (back and forth) using a GRU with n steps*



Done 3 times (for 3 GRU layers)

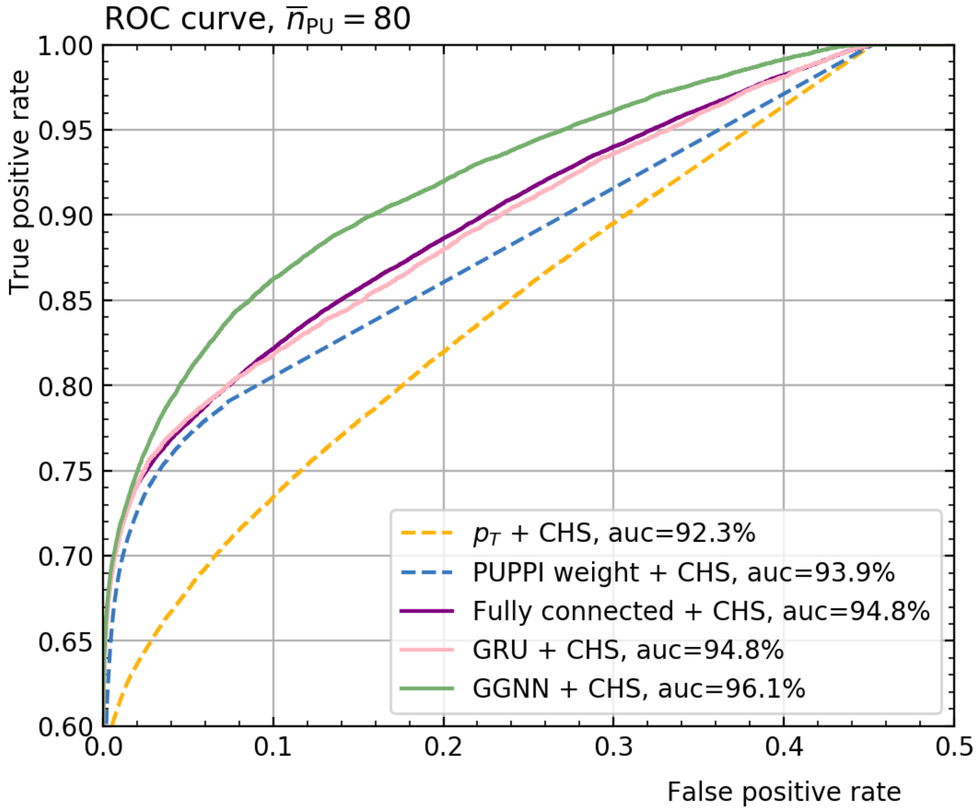


- *The representation created by each layer is passed to the next*
- *A ResNet-like skip connection is implemented (input -> last layer)*
- *The per-particle outcome set of features (function of the features of the connected particles) is used to train a dense classifier: PU vs interesting particle?*

PUPPIML: Graph Nets for PU subtraction

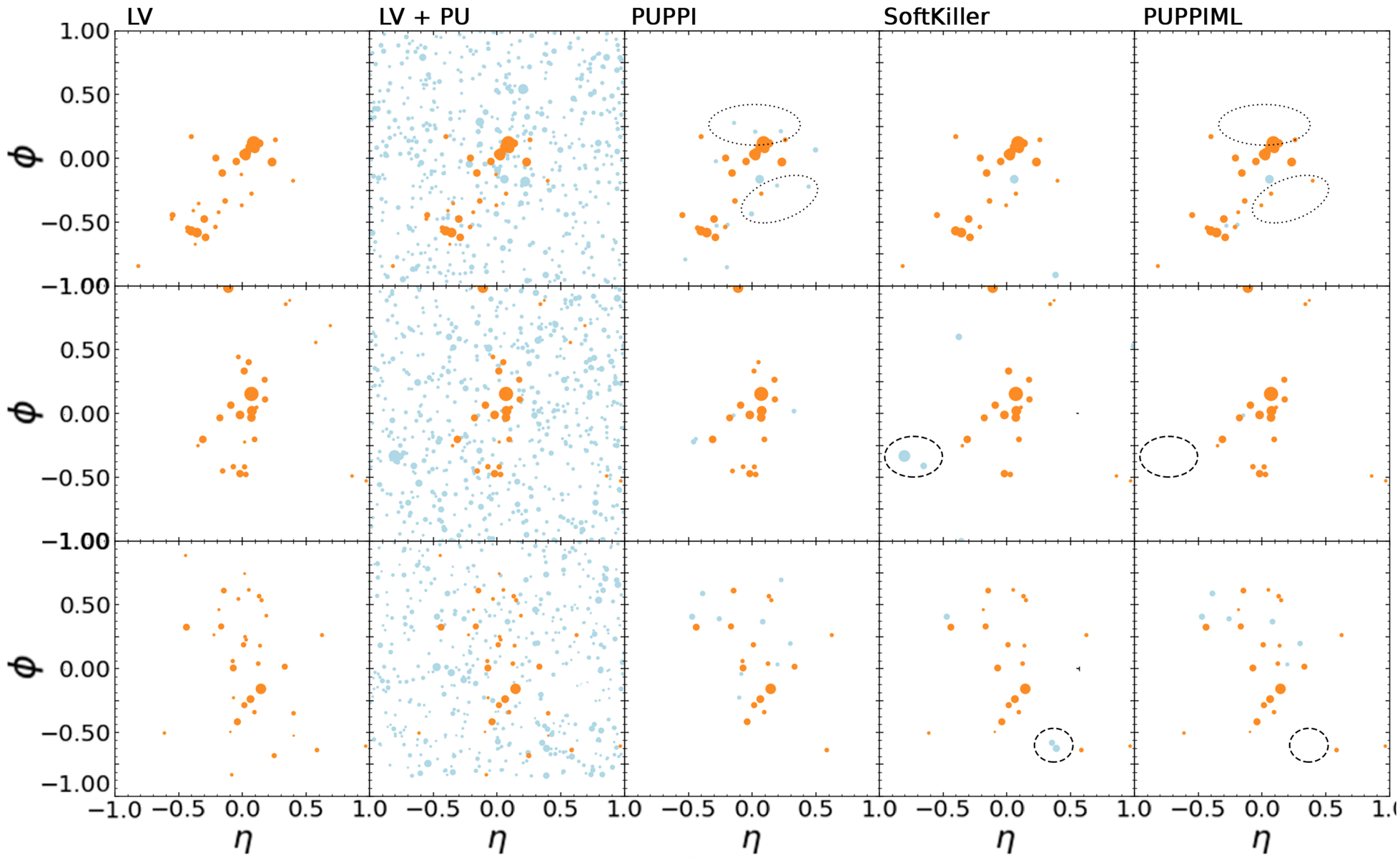
- *Improve state-of-the-art algorithms substantially*
- *Little dependence of algorithm tuning on pileup conditions*
- *Small/No performance loss with average number of PU collisions*
- *Outperforms alternative particle-based architectures (DNNN, simple GRU)*

$\overline{n}_{\mathrm{PU}}$	20 (CHS)	80 (CHS)	140 (CHS)	80 (No CHS)
p_T	92.3%	92.3%	92.5%	64.9%
PUPPI weight	94.1%	93.9%	94.4%	65.1%
Fully-connected	95.0%	94.8%	94.8%	68.5%
GRU	94.8%	94.8%	94.7%	68.8%
GGNN	**96.1%**	**96.1%**	**96.0%**	**70.1%**



https://arxiv.org/pdf/1810.07988.pdf

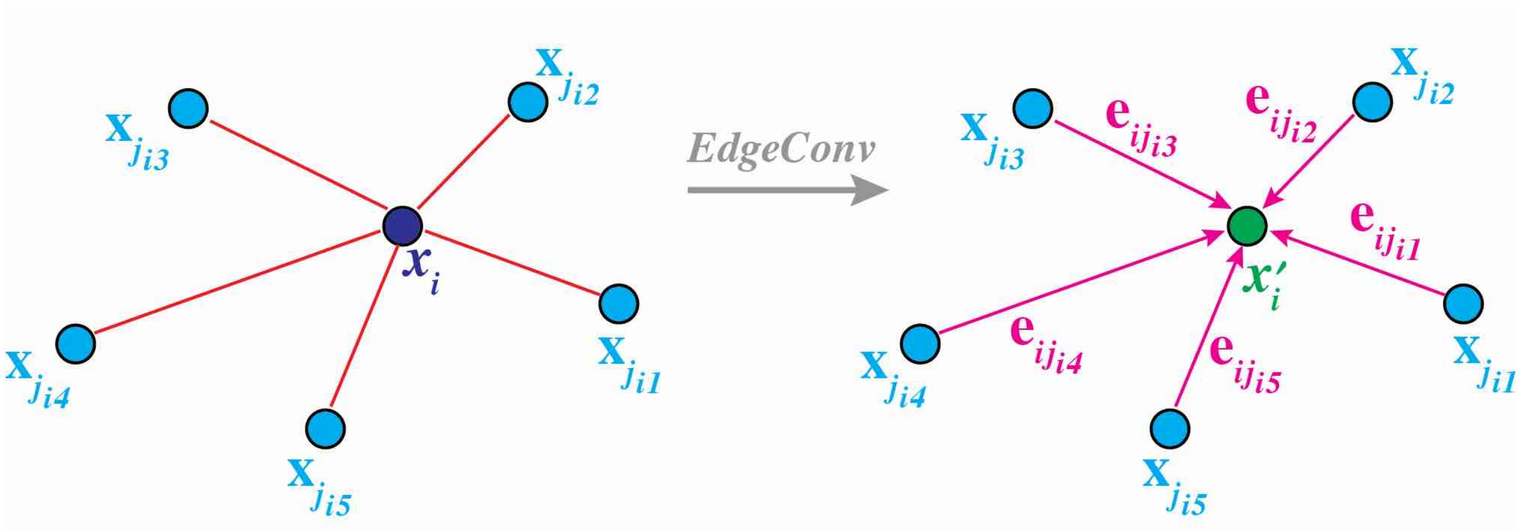
PUPPIML: Graph Nets for PU subtraction

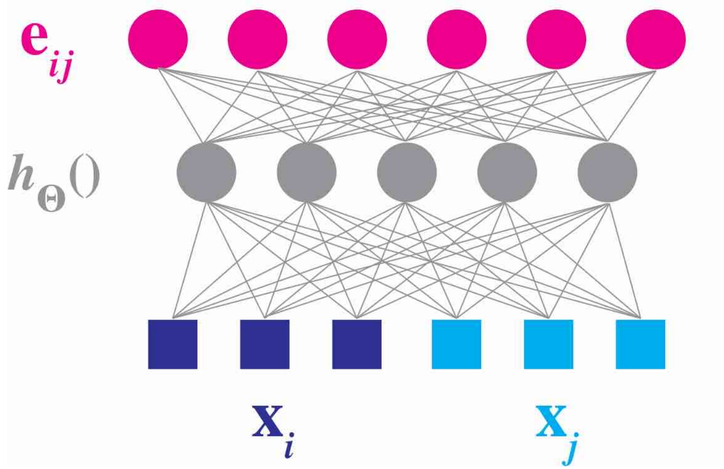


European Research Council

EdgeConv

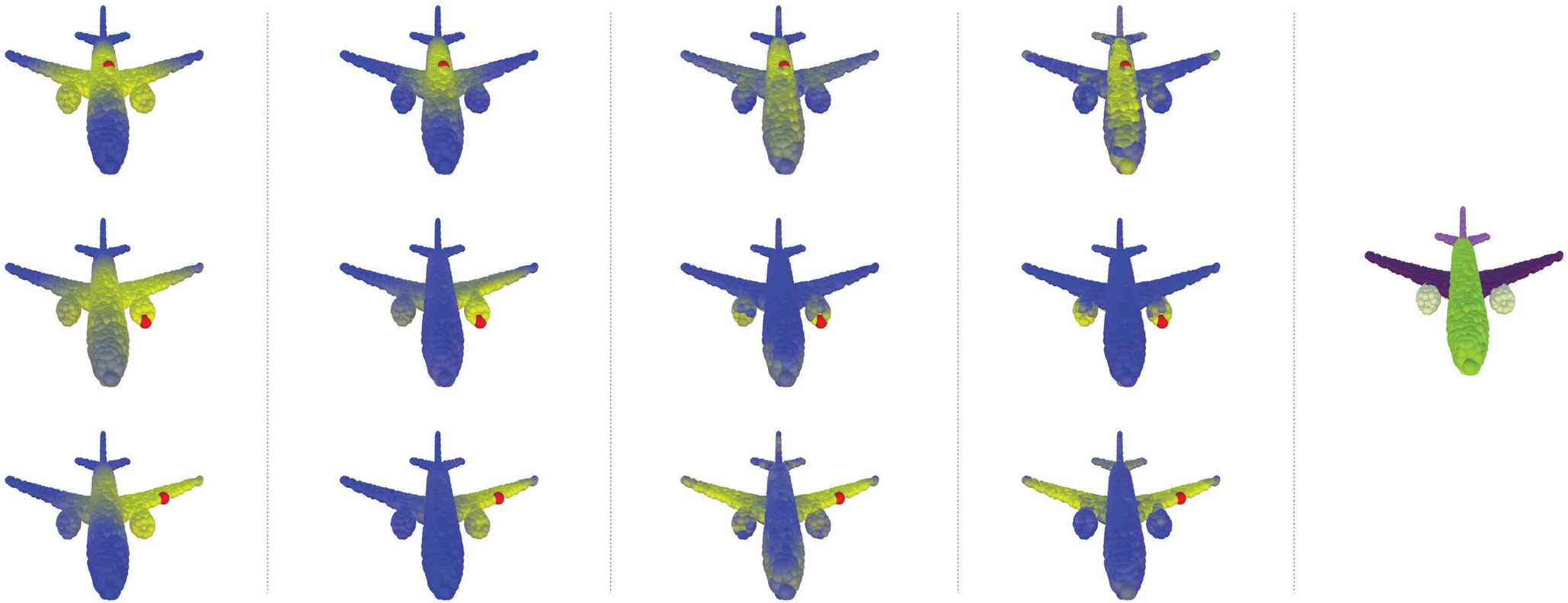
- *The EdgeConv is a generalization of the ConvNN to an irregular set of vertices (the equivalent of the image pixels)*
- *It starts from a graph connecting each point to its k nearest neighbours*
- *A trainable DNN is used to learn the optimal message function*





https://arxiv.org/abs/1801.07829

EdgeConv

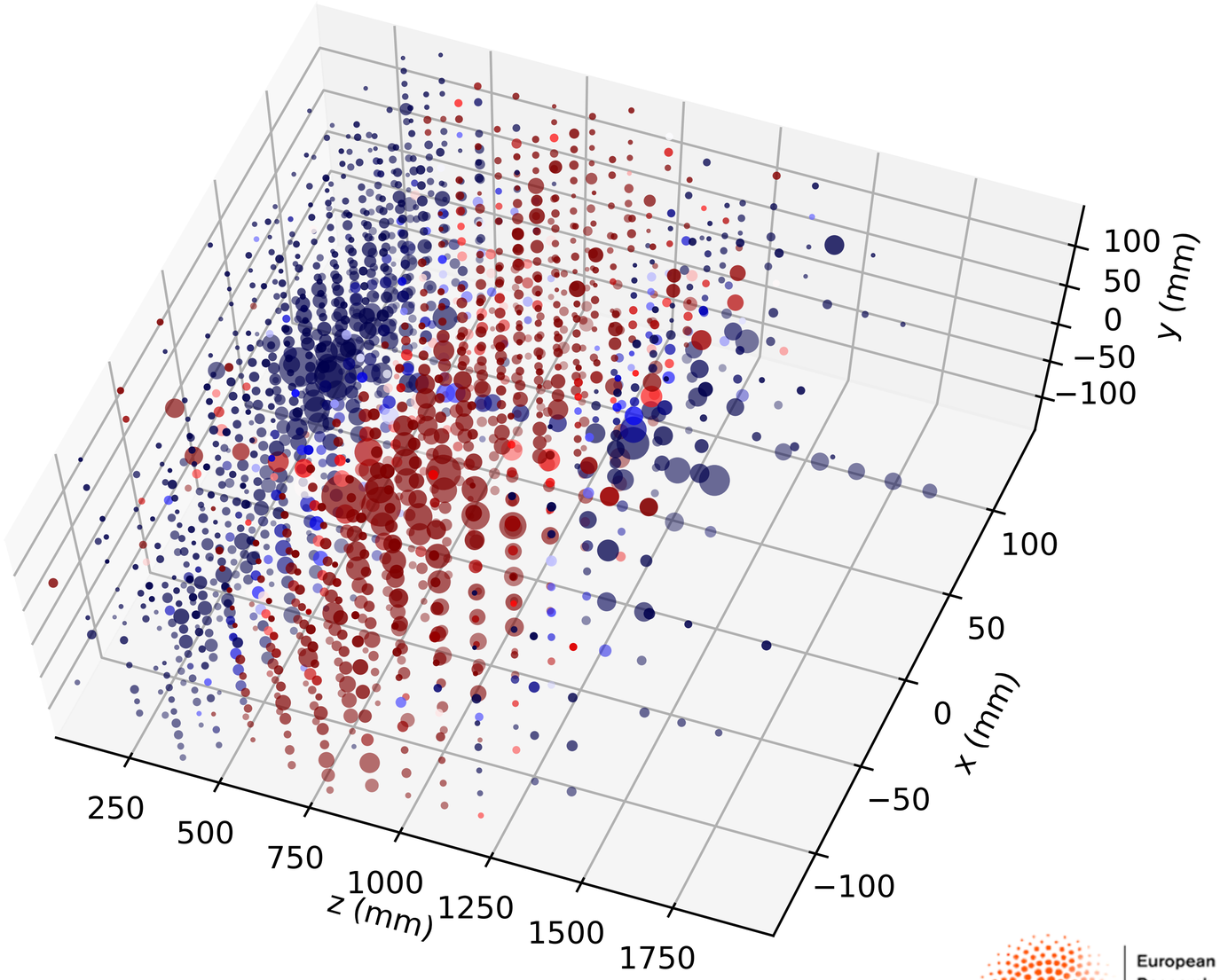


https://arxiv.org/abs/1801.07829

European Research Council

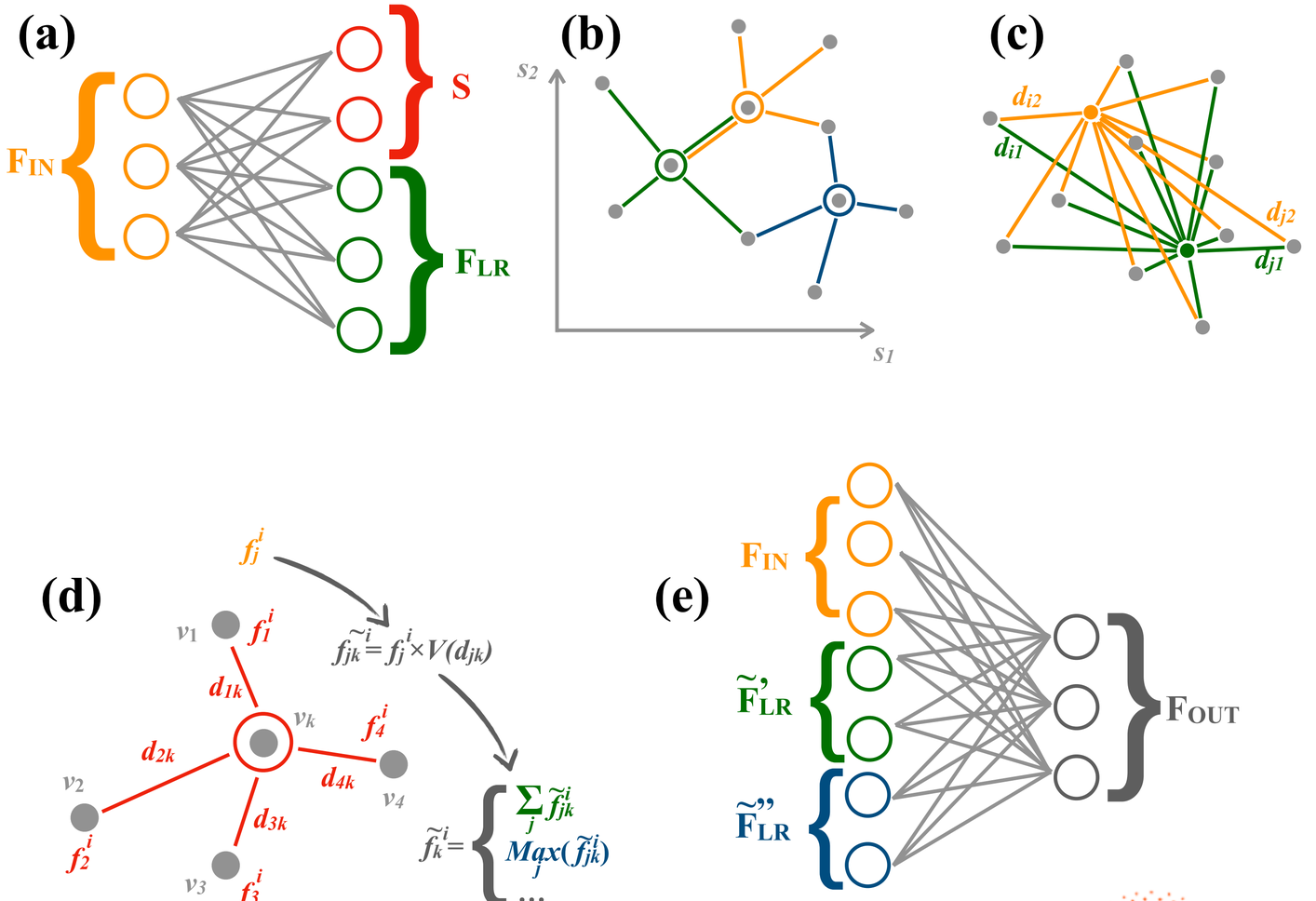
EdgeConv for Particle Physics

- *This architecture fits well our raw data: a collection of hits*
  - *each hit represented as coordinates + energy*
  - *EdgeConv used to learn representation*
- *Pros: no assumption on the underlying geometry*
- *Cons: large memory consumption (large number of connections)*



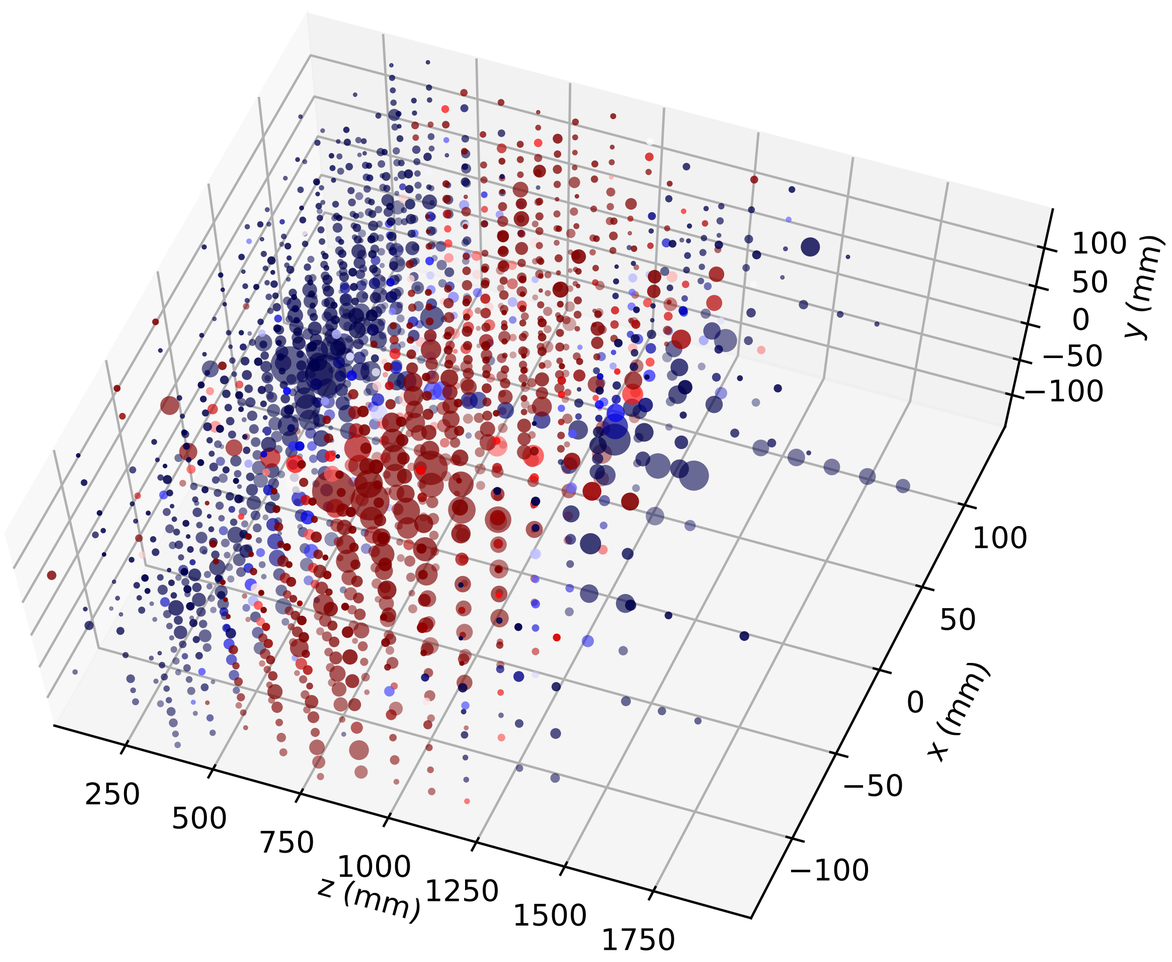
Reducing memory consumption

- *Introduce step to learn an optimized spatial distribution*
  - *as coordinates of a new space (b)*
  - *as distances from a fixed number of aggregators (c)*
- *Use customised architectures to keep resources under control*
  - *Gravnet: weight connection by potential of euclidean distance*
  - *Garnet: keep number of connections small through small number of aggregators*

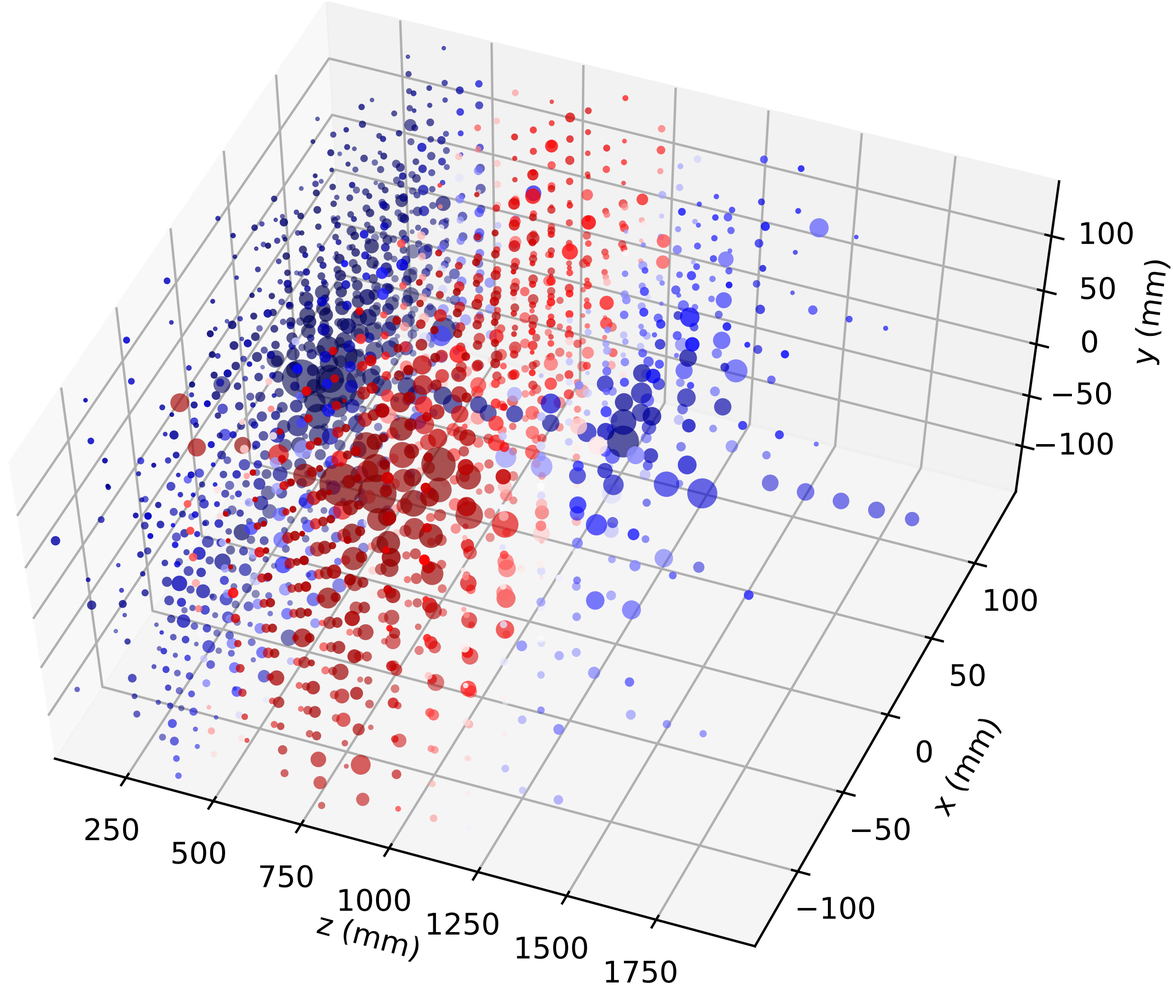


https://arxiv.org/abs/1902.07987

Separating overlapping showers



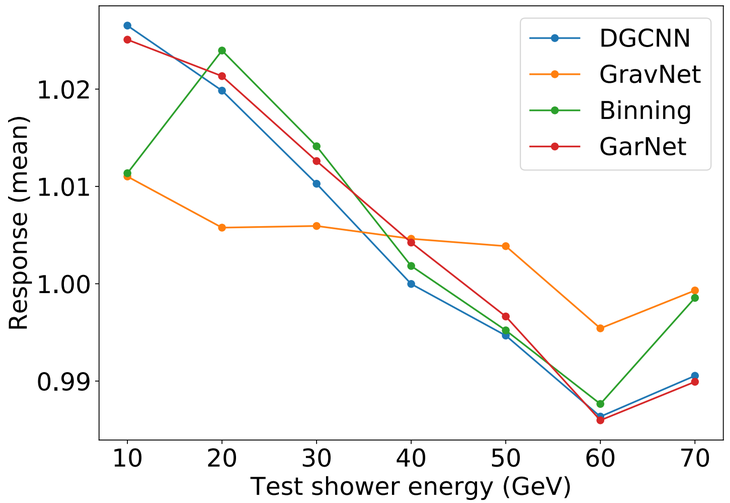
(a) Truth



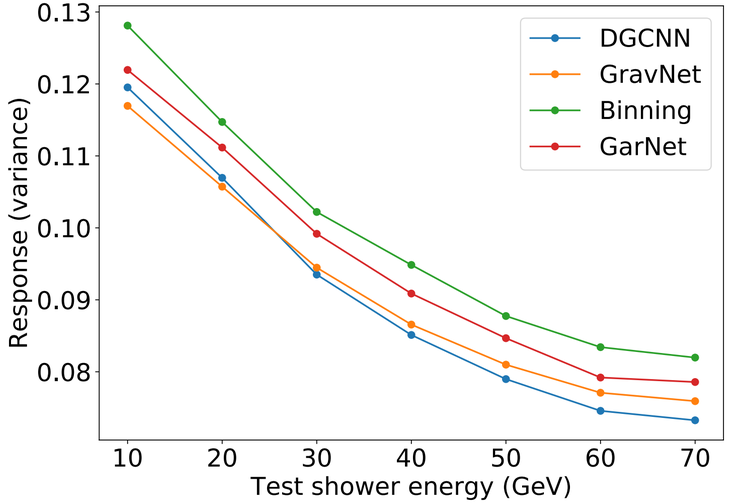
(b) Reconstructed

GraphNets for Calorimetry

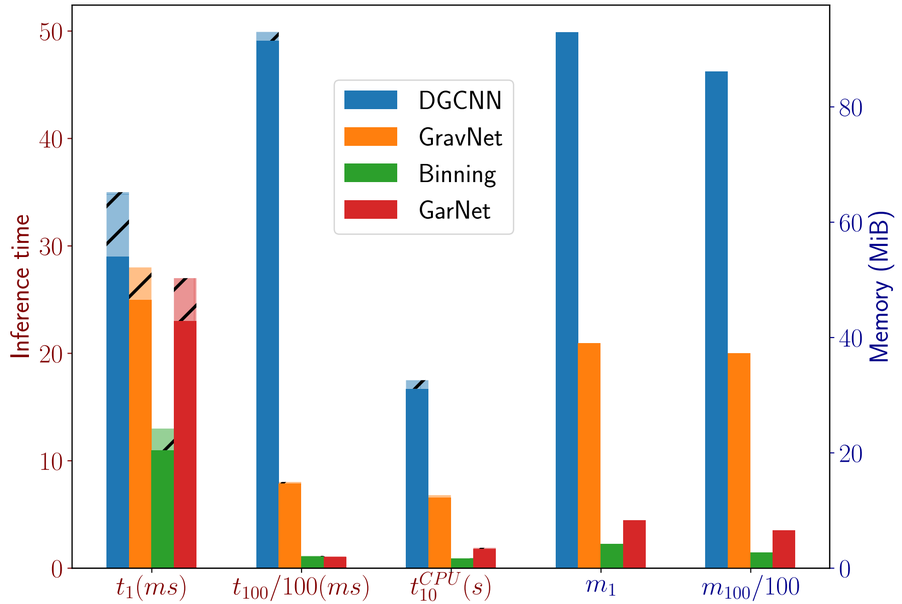
- *Good performance achieved, comparable to more traditional approaches*
- *Using a potential (V(d)) to weight up the near neighbours allows to keep memory footprint under control (with respect to other graph approaches)*



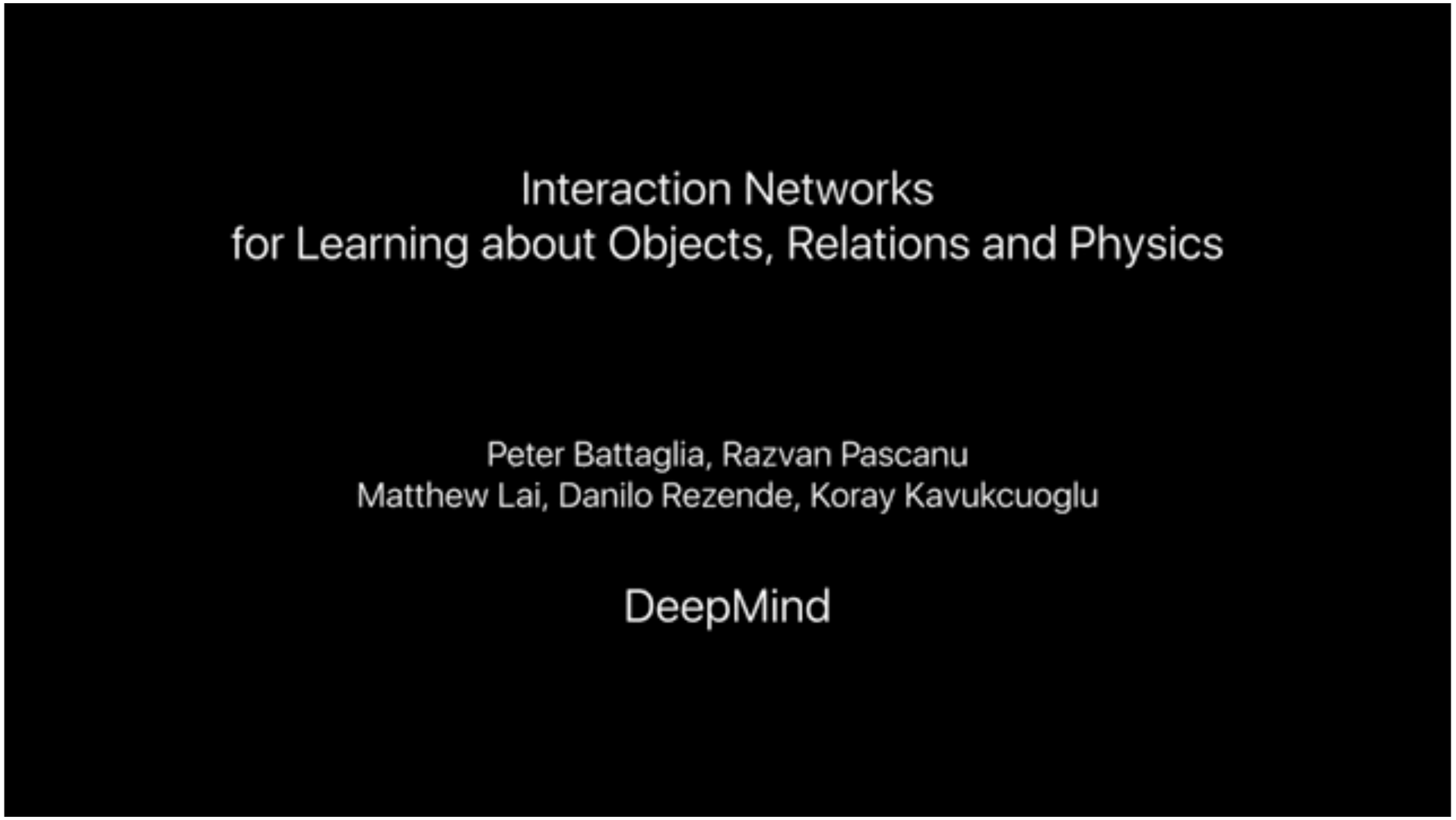
(c) Mean



(d) Variance



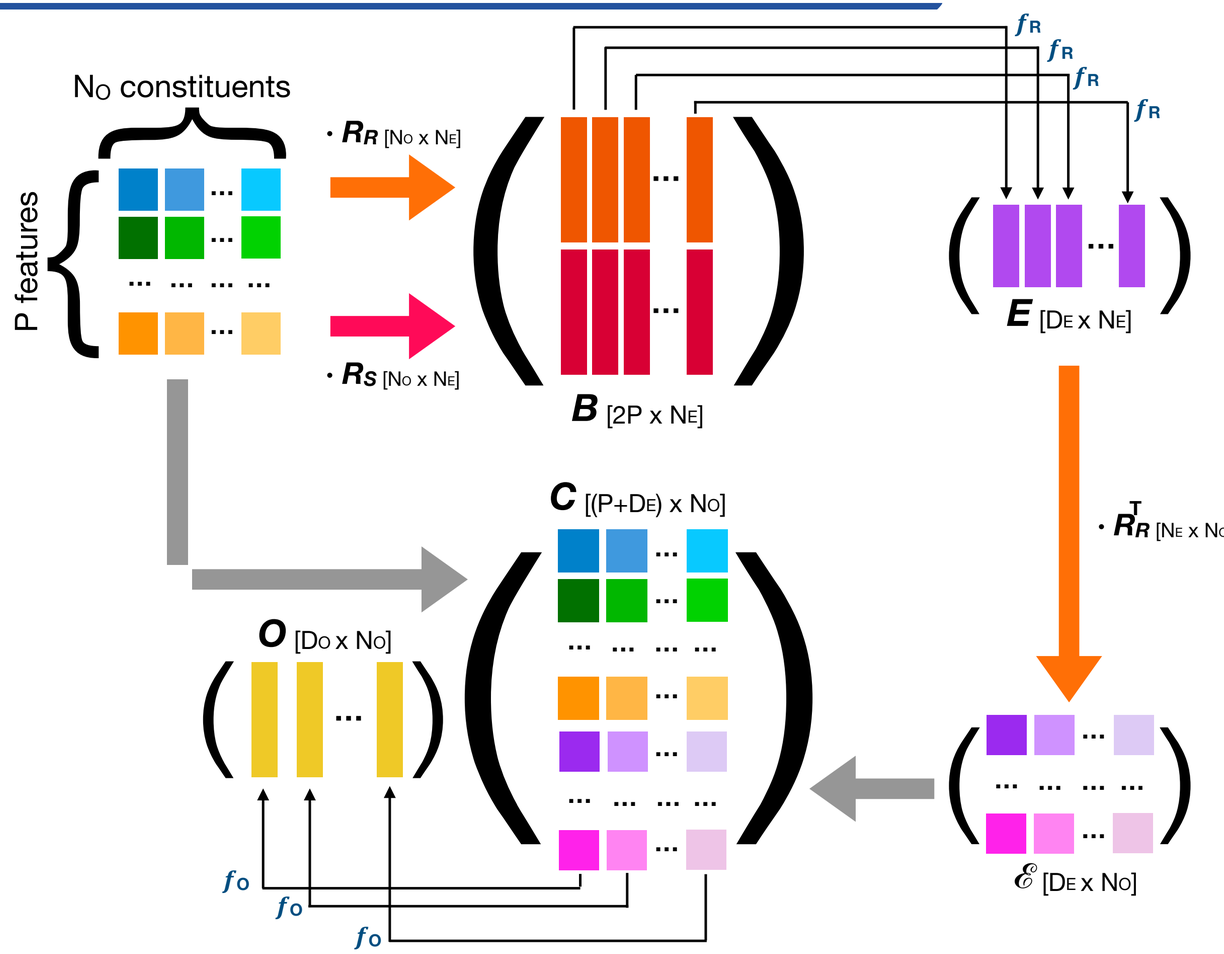
Interaction Networks



https://arxiv.org/abs/1612.00222

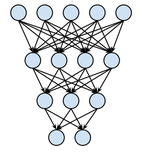
Interaction Networks

- *INs process a list of No x P inputs in pairs, through Receiving and Sending matrices*



NO: # of constituents
P: # of features
NE = NO(NO-1): # of edges
DE: size of internal representations
DO: size of post-interaction internal representation

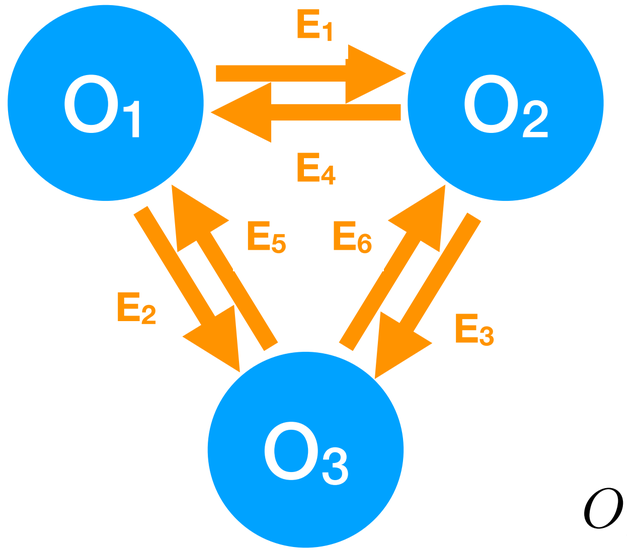
ϕC, fO, fR parameterized as neural networks



erc European Research Council

Interaction Networks

- *INs process a list of No x P inputs in pairs, through Receiving and Sending matrices*
- *The effect of the interaction is learned by fR and combined with the input to learn (through fo) a post-interaction representation*

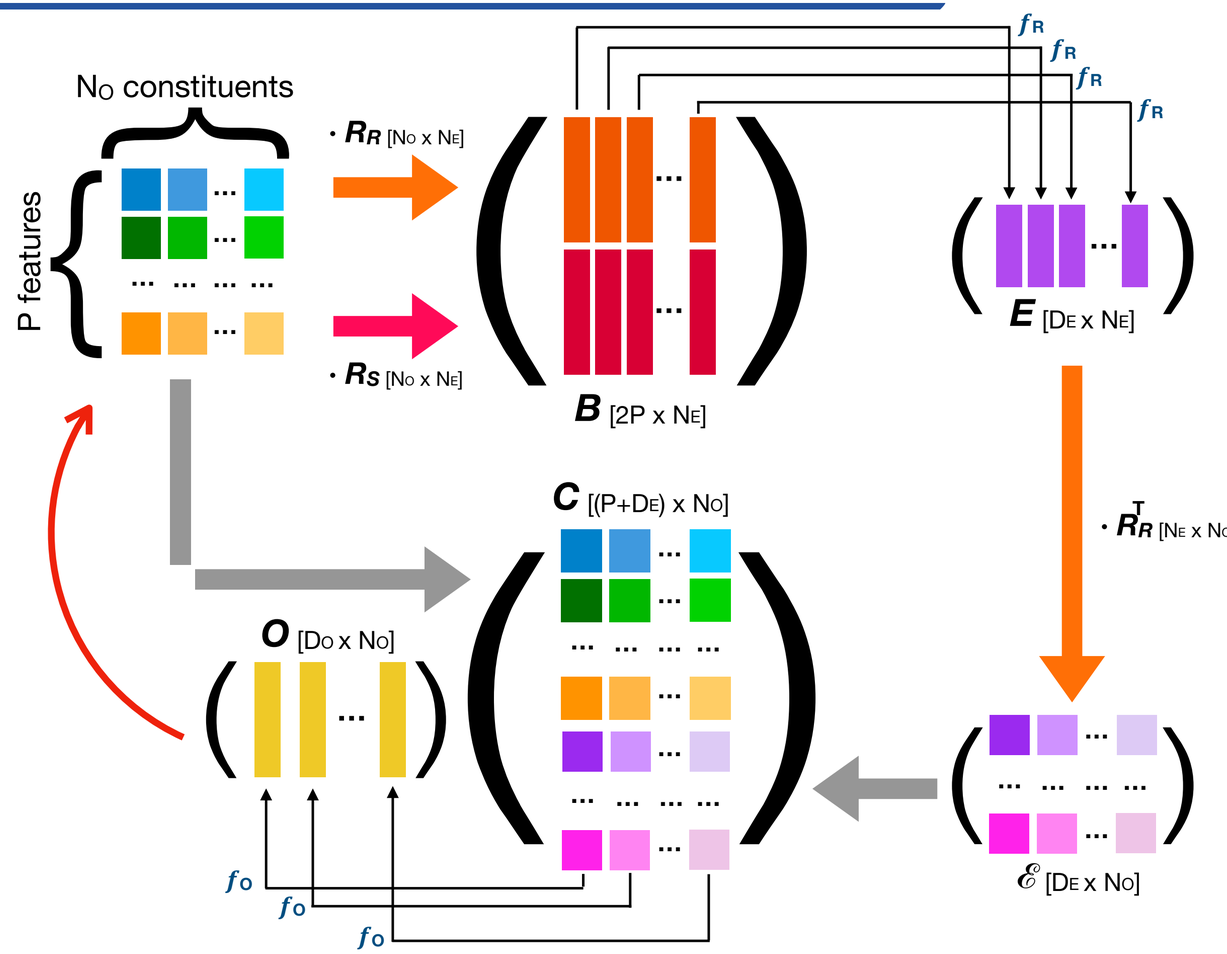


$$R_R = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \end{array} \begin{array}{c} \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$R_S = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \end{array} \begin{array}{c} \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

Interaction Networks

- *INs process a list of No x P inputs in pairs, through Receiving and Sending matrices*
- *The effect of the interaction is learned by fR and combined with the input to learn (through fo) a post-interaction representation*
- *The procedure can then be iterated to produce further steps i the interactions*

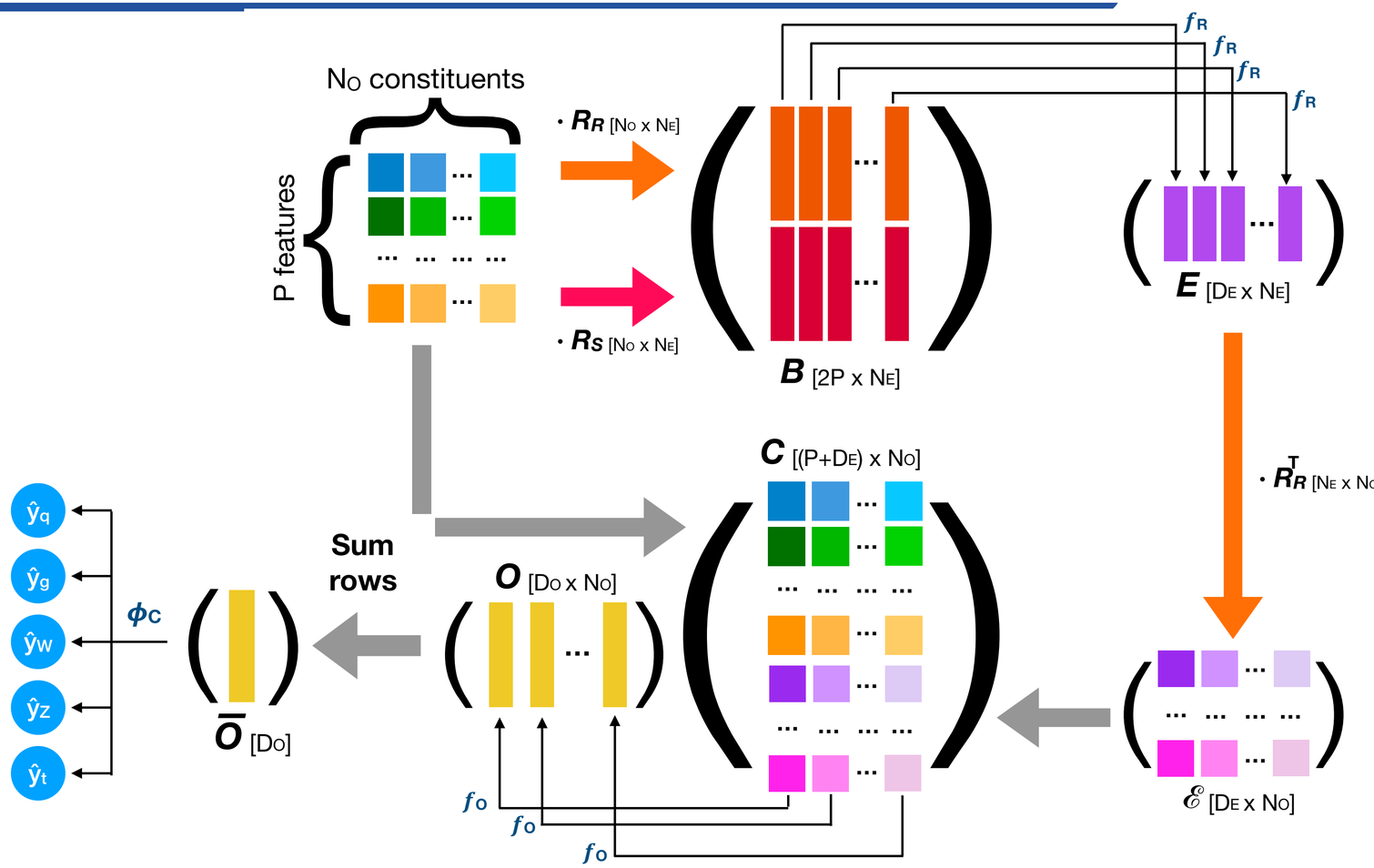


NO: # of constituents
P: # of features
NE = NO(NO-1): # of edges
DE: size of internal representations
DO: size of post-interaction internal representation

ϕC, fO, fR parameterized as neural networks

INs for Jet Identification

- *In our case, it is sufficient to use the post-interaction representation as input to a classifier that returns the jet category*
- *The three networks are simultaneously optimized: the learned representation is chosen to help the classification*

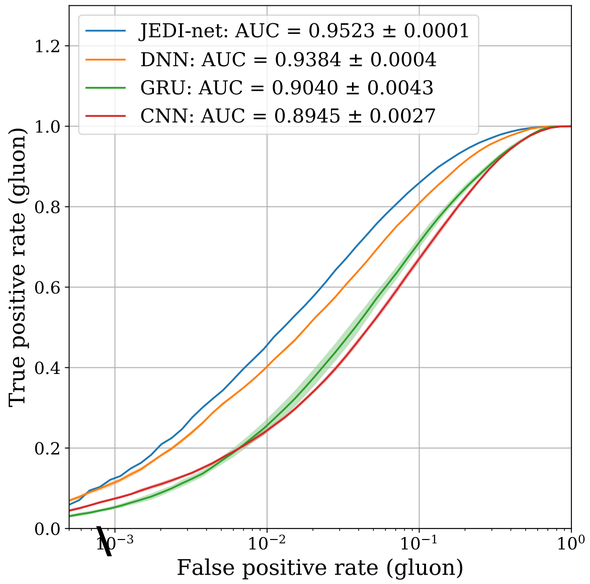


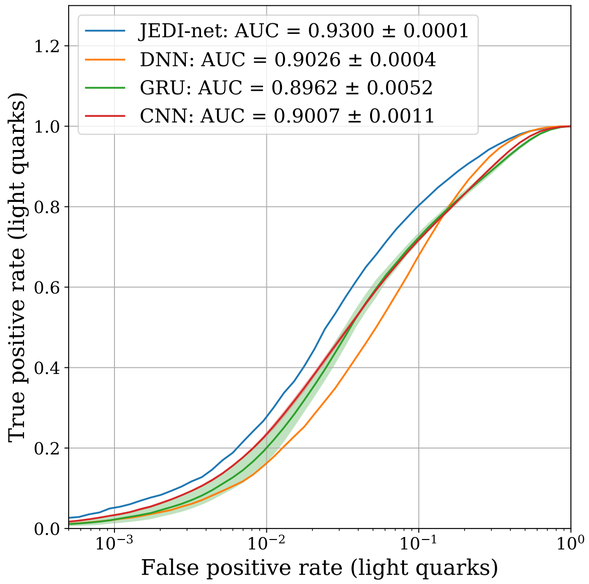
N_O: # of constituents
P: # of features
$N_E = N_O(N_O-1)$: # of edges
D_E: size of internal representations
D_O: size of post-interaction internal representation

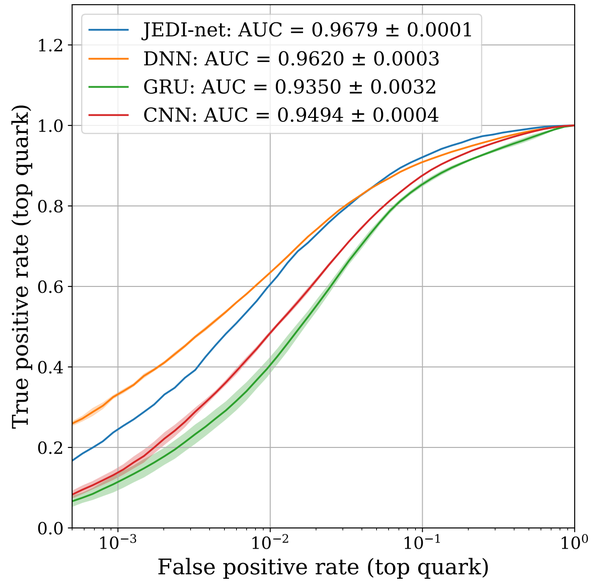
ϕ_C, f_O, f_R parameterized as neural networks

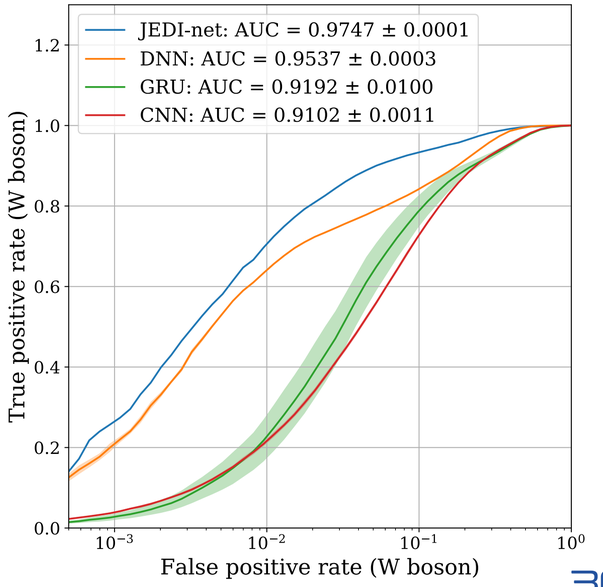
erc European Research Council

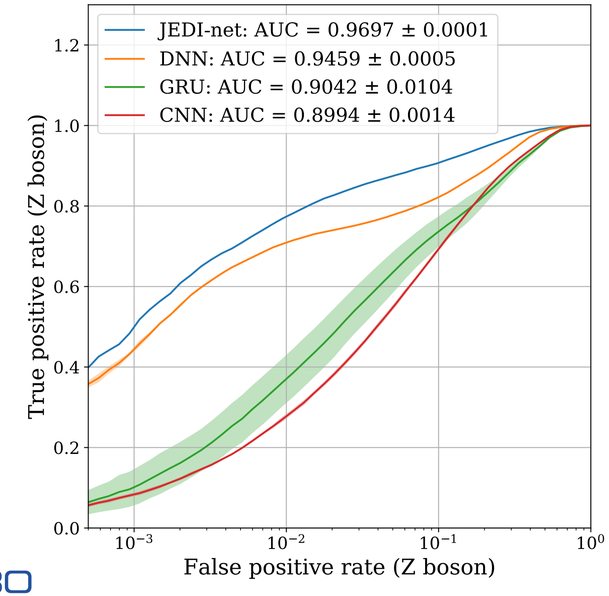
A comparison





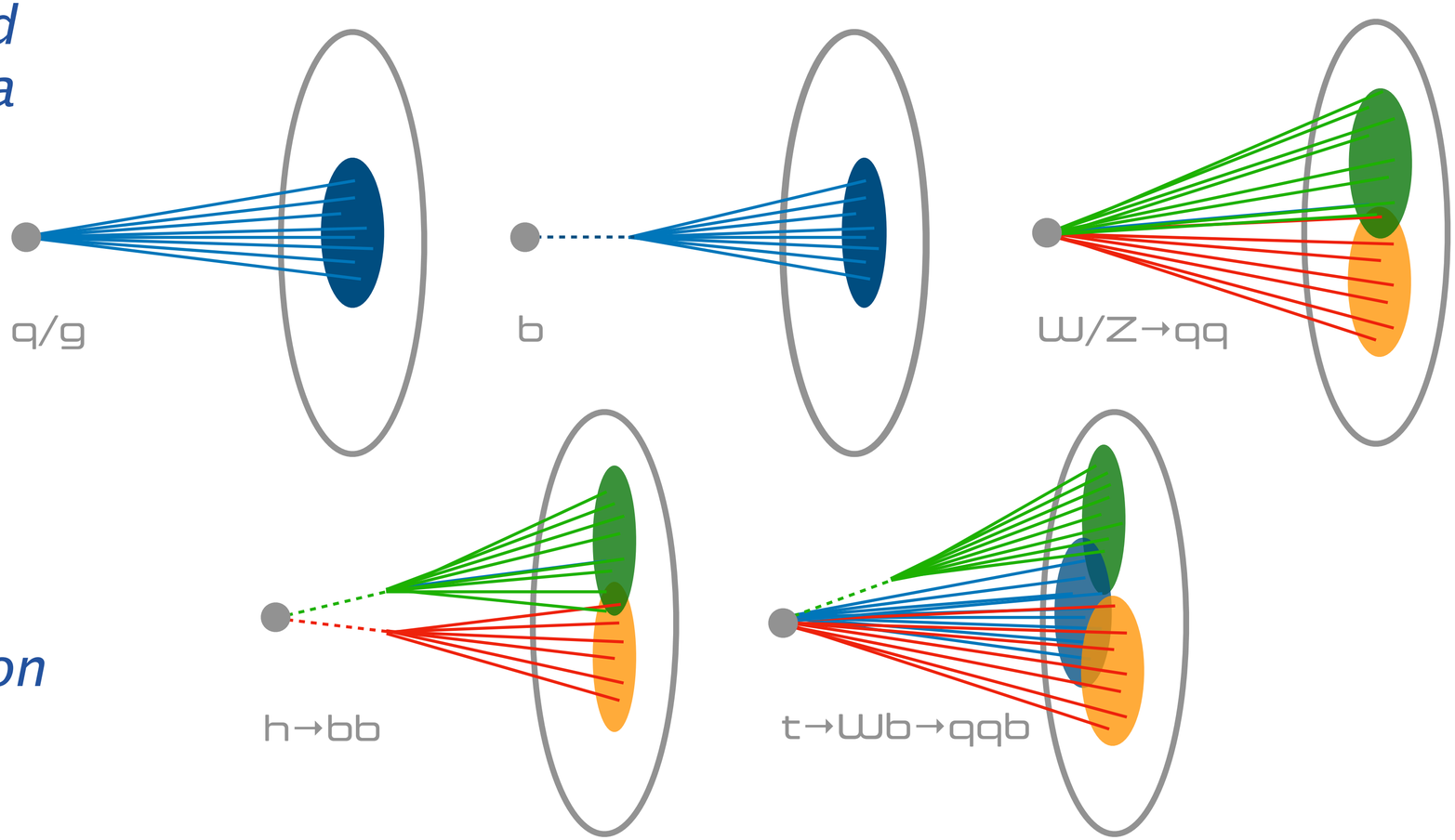






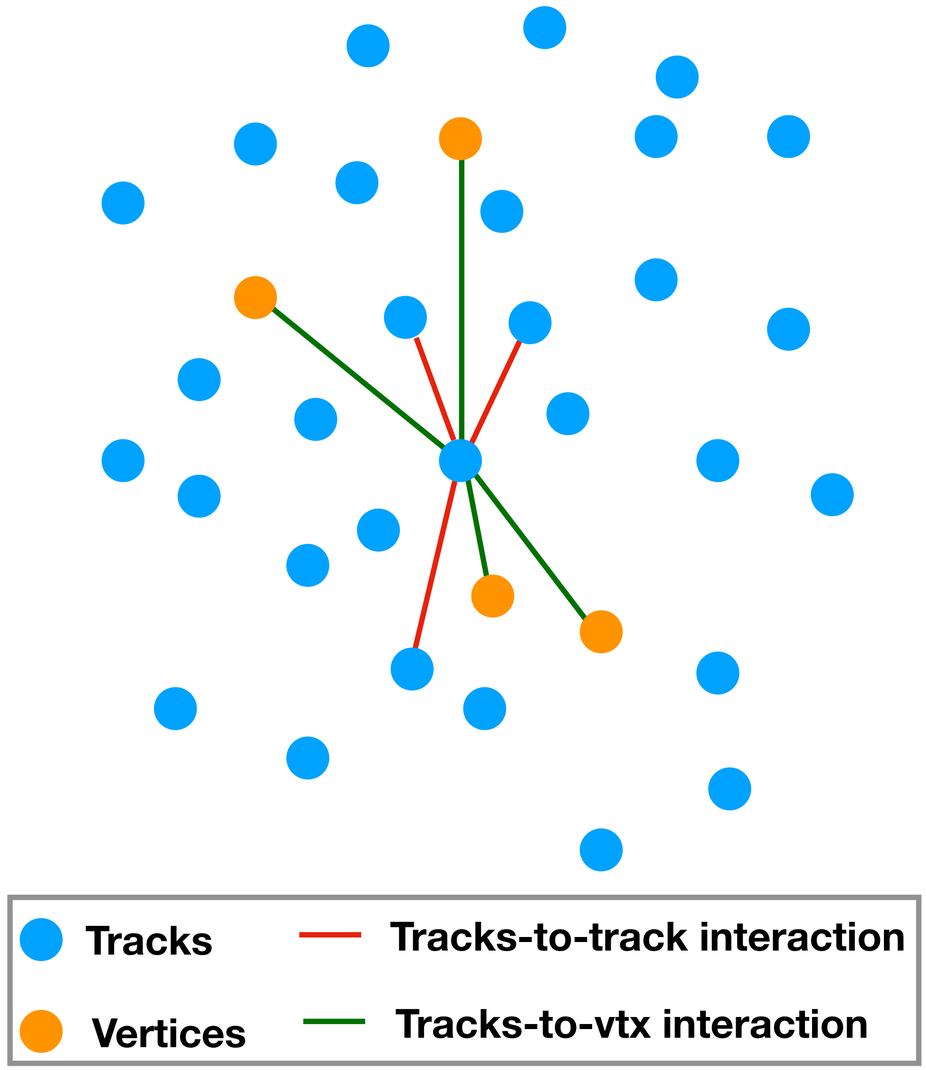
Using it for Higgs bosons

- *Boosted Higgs bosons are similar to Ws and Zs, but have an extra handle*
  - *It decays to b's, which decay after displacement*
  - *the displaced vertices provide extra discrimination handle*
- *The flexibility of the grapnel setup allows to add this easily*

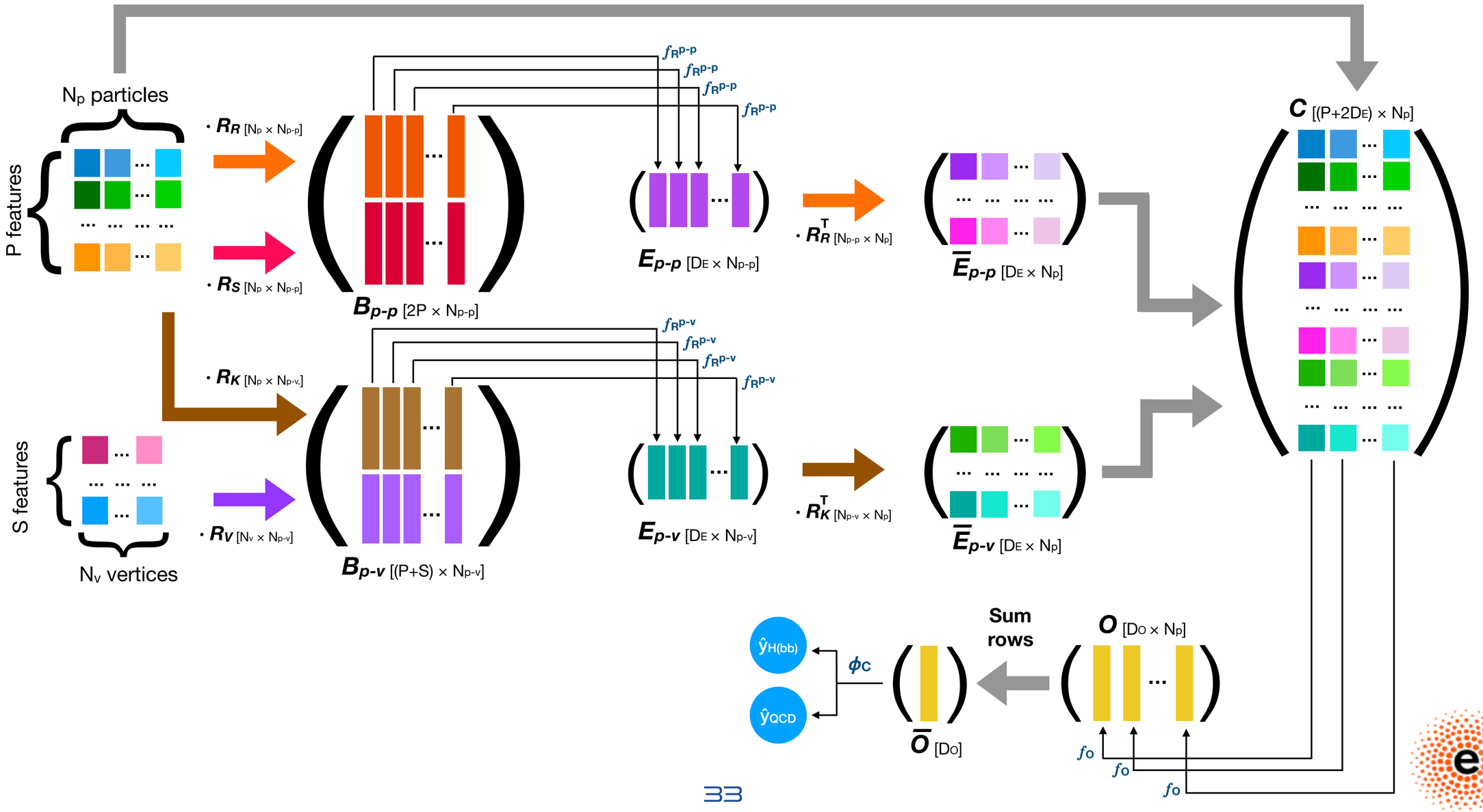


Adding vertices

- *Boosted Higgs bosons are similar to Ws and Zs, but have an extra handle*
  - *It decays to b's, which decay after displacement*
  - *the displaced vertices provide extra discrimination handle*
- *The flexibility of the grapnel setup allows to add this easily*

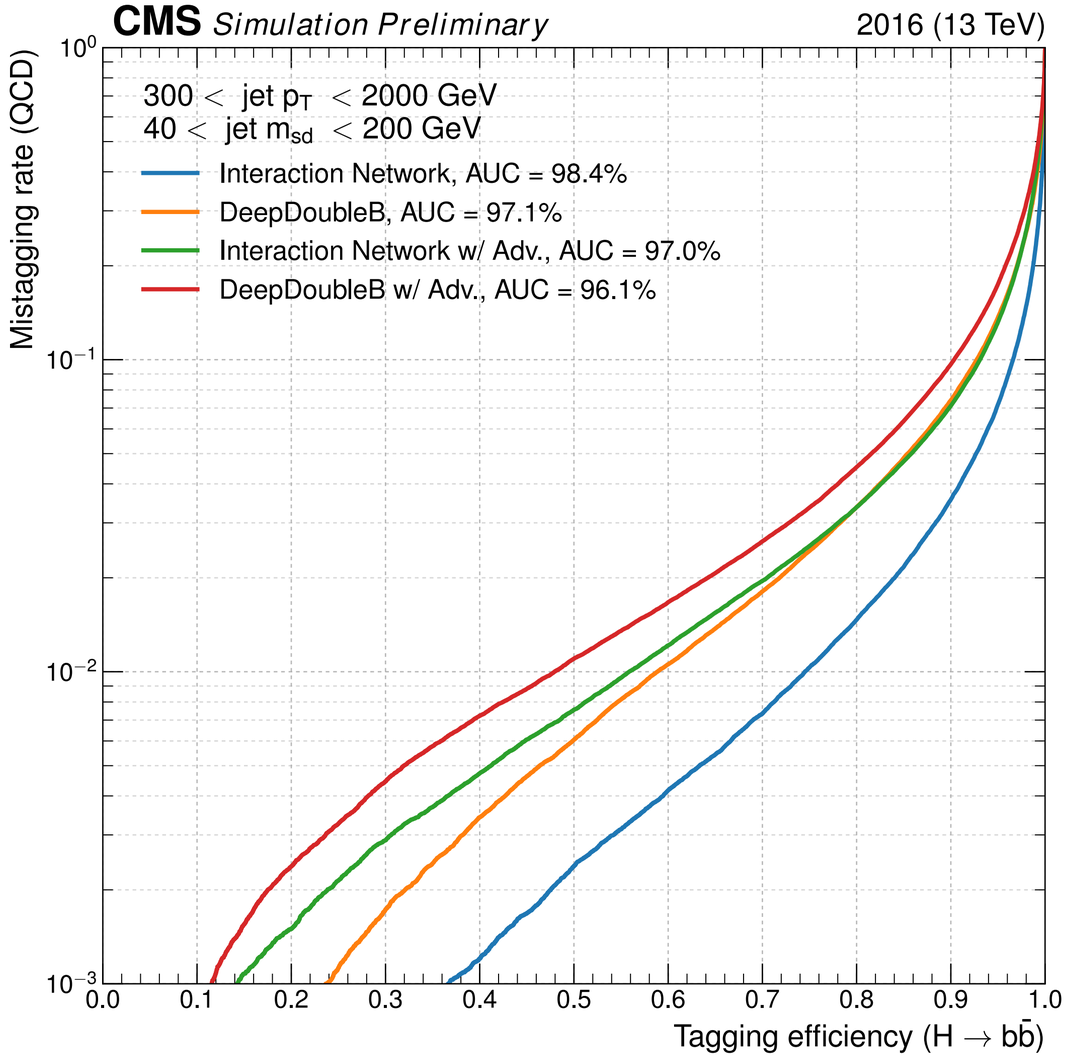


Adding vertices



Using it for Higgs bosons

- *Algorithm trained on CMS simulation from 2016*
- *Performance compared to the algorithm used in CMS*
  - *RNN-based tagger exploiting also neutral particles*
- *INs outperform the RNN algorithm despite using less information*
- *Relying only on tracks make the algorithm even more robust (less dependence on pileup)*
- *Just one example of how GraphNets could become relevant for next runs*



Summary

- *We reviewed Graph Networks*
  - *extend CNN concept beyond the case of physically proximity -> learned representation*
  - *allows to abstract from the detector geometry*
  - *can be used for basic reconstruction and on most abstract objects (particles, vertices, etc)*
- *In particular, we reviewed interaction networks as a candidate with interesting improvements in performance*