Graph Neural Networks in Particle Physics*

RAPHAEL COBE

SPRACE

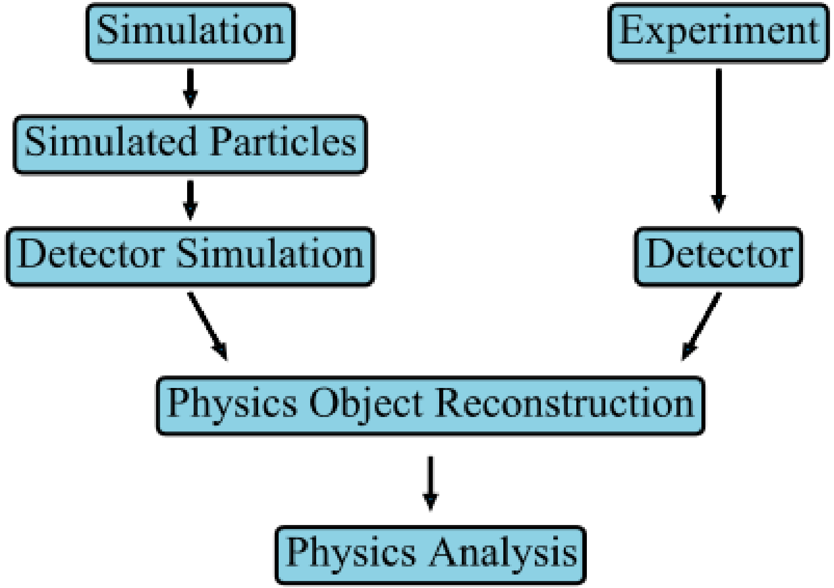
* Based on the Graph Neural Networks in Particle Physics Survey (arXiv:2007.13681)

Particle Physics

HEP Experiments

- Often use machine learning for learning complicated inverse functions;
- Try to infer something about the underlying physics process from the information measured in the detector;

- The Pipeline:



arxiv:2007.13681

Particle Physics

HEP Experiments

- Simulation **creates a ”truth record”** of the physics event which caused a certain detector response;
- ”truth record” is used to **train supervised learning** algorithms;
- **Invert the detector simulation** and infer something about the underlying physics from the observed data;
- Algorithms are then **applied to real data** that were measured by the detector;

Particle Physics

Data Representation

- Detectors with sizes on the order of tens of meters;
- Capture **millions of high-dimensional measurements** each second;
- Multiple sub-detectors: **each using a different technology** to measure the trace of particles;
- Data in particle physics are therefore **heterogeneous;**
- Measurements are **inherently sparse in space**;
- Measurements do not fit homogeneous, grid-like data structures;

Particle Physics

Data Representation

- Some data in particle physics can be interpreted as images
  - Computer vision techniques and CNNs;
- Image representations face some limitations:
  - Irregular geometry of detectors
  - Sparsity of the projections;
  - **Loss of information**.

Particle Physics

Data Representation

- Measurement and reconstructed objects viewed as sequences:
  - Order imposed from theoretical or experimental understanding of the data;
    - Can usually be justified experimentally;
    - Constrains how the data are presented to models;
    - **Not always the case!**
    - There is evidence* that a permutation invariant network outperforms a sequence-based algorithm that uses the exact same input features, for the same classification task;

* inspirehep:1798845

Deep Learning

Convolutional Neural Networks

- Convolutional layers share their underlying kernel function across spatial dimensions of the input signal, while recurrent layers share across the temporal dimension of the input;
  - Most suitable for approximating functions on vectors, grids, and sequences;
  - Problems with data with richer structure;
- Consider learning functions over sets of particles:
  - It is possible to order themby the transverse momentum pT of the particle;
  - The imposed ordering in not unique, and it fails to reflect that particles are fundamentally unordered;

Graph Networks

GNs and physical systems

- Developed and applied for network analysis, especially on internet data;
- GNNs to simulate increasingly complex physical systems;
  - E.g.: particle-based simulation:
    - A set of particle vertices;
    - Interactions are represented by edges and computed via learned functions;
- GNs are highly parallelizable on modern deep learning hardware (GPUs, TPUs, FPGAs);
  - Speed was on par with heavily engineered state-of-the-art fluid simulation engines;

Graph Networks

GN's general architecture*

$$
\begin{aligned}
\mathbf{e}'_k &= \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) & \bar{\mathbf{e}}'_i &= \rho^{e\to v}\left(E'_i\right) & &\triangleright \text{ Edge block}\\
\mathbf{v}'_i &= \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) & \bar{\mathbf{e}}' &= \rho^{e\to u}\left(E'\right) & &\triangleright \text{ Vertex block}\\
\mathbf{u}' &= \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right) & \bar{\mathbf{v}}' &= \rho^{v\to u}\left(V'\right) & &\triangleright \text{ Global block}
\end{aligned}
$$

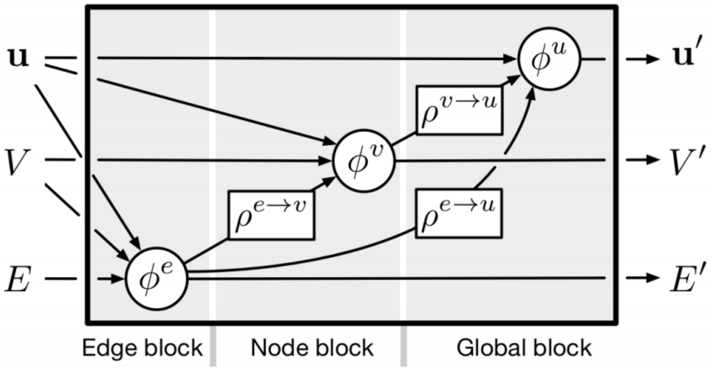
- Update functions: $(\phi^e,\ \phi^v,\ \text{and}\ \phi^u)$
- Aggregation functions: $(\rho^{e\to v},\ \rho^{e\to u},\ \text{and}\ \rho^{v\to u})$
- The edge block computes one output for each edge, and aggregates them by their corresponding receiving node;
- The vertex block computes one output for each node;
- The edge- and node-level outputs are all aggregated in order to compute the global block;

* arxiv:2007.13681

Graph Networks

GN's general architecture*

- In practice, the update functions are often implemented as simple trainable NNs;
- Aggregation functions are tipically implemented as permutation invariant reduction operators, such as elementwise sums, means, or maximums;



* arxiv:2007.13681

Graph Networks

GN's general architecture*

- General framework which can capture a variety of other GNN architectures:
  - Removing or rearranging internal components of the general GN block;
  - Implementing the update and aggregation functions using specific functional forms;
  - Graph Convolutional Network (GCN):

$$\mathbf{e}'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{s_k}) = \mathbf{e}_k \mathbf{v}_{s_k} \ , \qquad \text{where } \mathbf{e}_k = \frac{1}{\sqrt{\text{degree}(r_k)\text{degree}(s_k)}}$$

$$\bar{\mathbf{e}}'_i = \rho^{e\to v}(E'_i) = \sum_{\{k \,|\, r_k = i\}} \mathbf{e}'_k$$

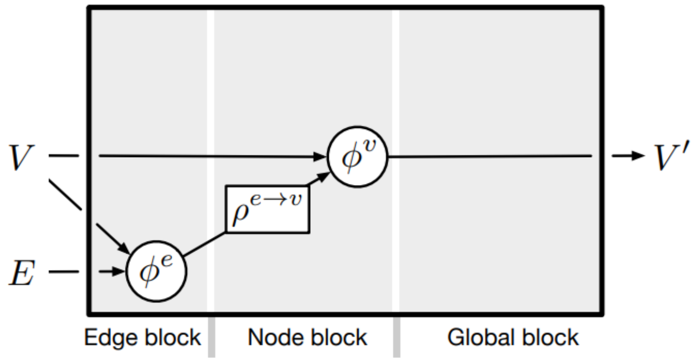
$$\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}'_i) = \sigma(\bar{\mathbf{e}}'_i W)$$

* arxiv:2007.13681

Graph Networks

GN's general architecture*

- ❑ Graph Convolutional Network (GCN):



* arxiv:2007.13681

Applications to Particle Physics

The most important thing!

- Decide how the data could be expressed as a graph:
  - What are the entities and relations that will be represented as nodes and edges?
  - What is the required output, i.e., edge-, node-, or graph-level predictions?
- Choices about the specific GNN architecture:
  - Is a global output network required to produce graph-level outputs?
  - Should pairwise interactions among nodes be computed, or more GCN-like summation and non-linear transformation?
  - How many message-passing steps should be used to propagate information among distant nodes in the graph?

Applications to Particle Physics

Jet Classification

- Jets: sprays of stable particles that are stemming from multiple successive interaction and decays of particles, originating from a single initial object;
- QCD jets are covering an extremely large phase space and constitute an irreducible background;
- Within the particle flow reconstruction, the event is interpreted through a set of particle candidates
  - Interaction network architecture to the purpose of graph categorization*
  - Fully connected graph over the particles of a jet and primary vertices of the event;
  - Extract a graph category after one step of message passing;

* arXiv:1908.05318

Applications to Particle Physics

Jet Classification

- Use *edgeconv* method from to derive a point cloud (unordered set of particles) architecture for jet tagging* :
  - The connectivity of the graph is defined dynamically by computing node neighborhoods over the distance in either the input space, or an intermediate latent space when graph layers are stacked;
  - The performance for the quark/gluon discrimination and top tagging (discriminating hadronic top decay and QCD jet) reported to be better than other previously ML based approaches;

* arXiv:1902.08570

Applications to Particle Physics

Event Classification*

- Use message passing neural network architecture over a fully connected graph composed of the final state particles, and the missing transverse energy
- Messages are computed from the node features and a distance in the azimuth rapidity plane first, then in the node latent space for later iterations.
- Such messages are passed across the graph in two iterations, and each node receives a categorization.
- Reports better results than densely connected models;

* arXiv:1807.09088

Applications to Particle Physics

Particle Flow Reconstruction*

- PF: Algorithm that aims at assigning to a candidate particle all the measurements in each sub-detector;
- GNN method to extract the particles' information from the graph of individual measurements
- The model is set to predict the properties of a smaller number of particles than there are measurements (graph reduction);
- Decides whether a node of the graph has to be considered as representative of a particle in the event;
- Shown to be more efficient and produces fewer fake particles than the standard approach

* arXiv:2002.03605

Applications to Particle Physics

Charged Particle Tracking*

- Edge classification:
  - Each node represents one hit, with edge constructed between pairs of hits with geometrically plausible relations.
  - Multiple updates of the node representation and edge weight over the graph (using the edge weight as attention)
  - Model learns what are the edges truly connecting hits belonging to the same track.
  - transforms the clustering problem into an edge classification

* arXiv:1810.06111