

FPGA for real-time tracking at LUXE experiment

LUXE Simulation

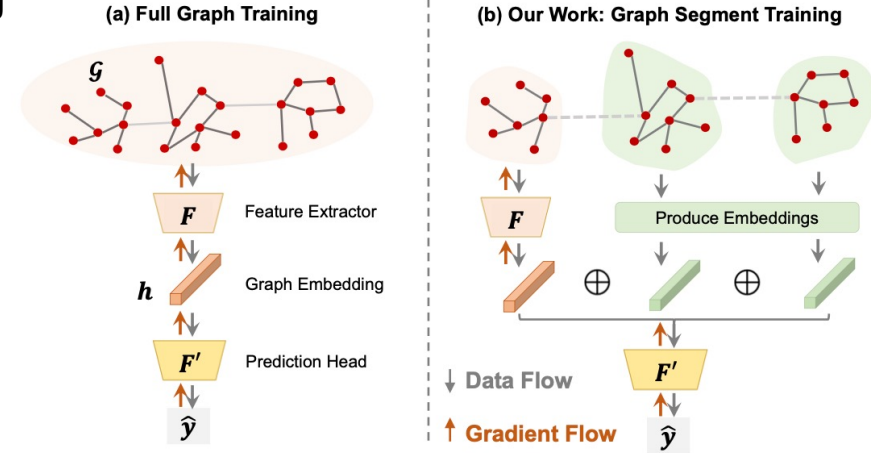
Ashraf Mohamed
SFT meeting 15/02/2024

Sampling graph

Large graph concept

arXiv:2305.12322v2

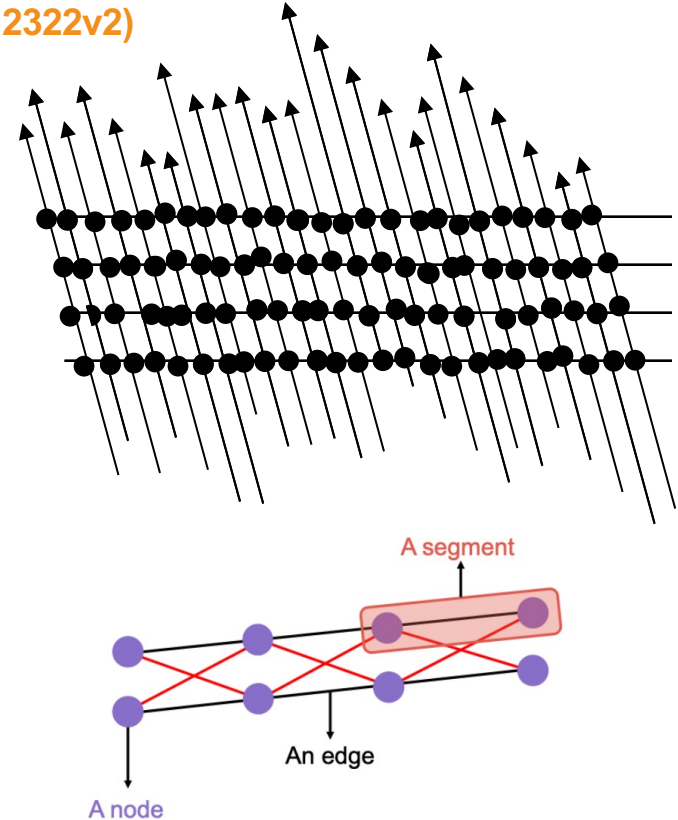
- Full Graph Training: Classically, models are trained using the entire graph \rightarrow all nodes and edges are used to compute gradients
- Graph Segment Training: partition each large graph into smaller segments and select a random subset of segments to update the model
- How large our graph is:
on average: 70k particles per BX produces ~ 250 k hits
every layer receives ~ 70 k hits
- While we build the graph, we hide the truth information
we rely only on geometrical cuts to reject fakes
- Graph Production Statistics:
Using 656 events, 46218570 particles in total
True edges: 143009166 Fake edges: 7780995637



Bootstrapping large graph

Sample particles not hits from google to LUXE (arXiv:2305.12322v2)

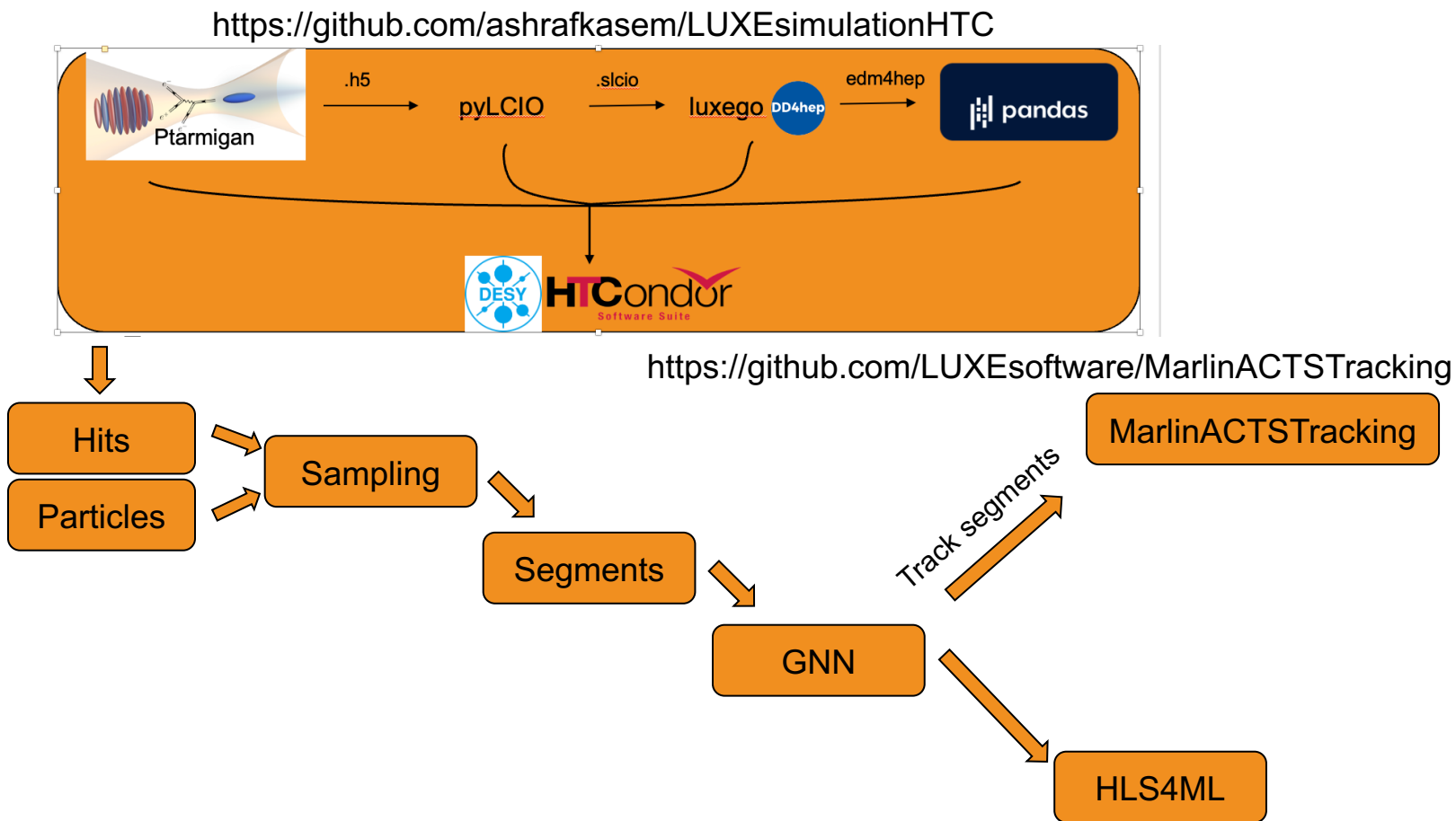
- In the aforementioned paper the graph is for social network
- In our case the graphs is of geometric nature, i.e. the segments need to stack together to form a complete track
- Sampling from the hits will lead to many incomplete tracks i.e fakes will be high
- Sampling from track segments is computationally expensive → need to build the entire graph segments and then sample from them
- Sampling from the particles list is remarkably faster and doesn't lead to discontinuous tracks
- Sampling size is dependent on the number of desired epochs --> each epoch while making the batch, we pick different sample of the same events



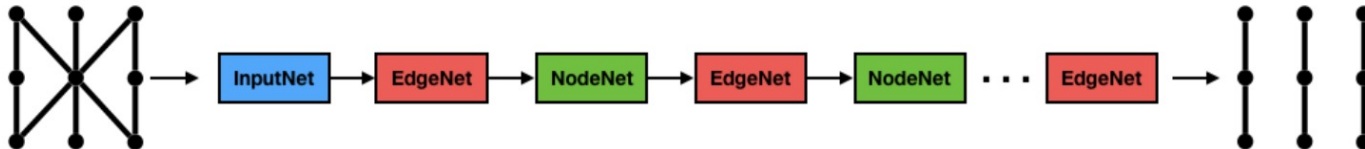
Graph Production Statistics.....

Using 8260 events, 29094320 particles in total

Total: True edges: 90026301 Fake edges: 244828904



<https://github.com/ashrafkasem/LUXETrackML>



Input transformation layer

A module which computes weights for edges of the graph. For each edge, it selects the associated nodes' features and applies some fully-connected network layers with a final sigmoid activation.

A module which computes new node features on the graph. For each node, it aggregates the neighbour node features (separately on the input and output side), and combines them with the node's previous features in a fully-connected network to compute the new features

A message-passing graph neural network model which performs binary classification of nodes.

gpu : A100-SXM4-80GB

Validation size : 10%

Training size : 90%

Lr : 0.001

batch_size : 1/10/20

n_iters : 4

n_epoch : 100

hid_dim : 128

optimizer : SGD

loss_func : BinaryCrossentropy

Inference

Model trained with bootstrapped sample size of 5% inference on [5%,10%,20%,30%,40%,100%]

Training performed using 5% sample size						
Metric	5%	10%	20%	30%	40%	100%
Accuracy	92.4	91.4	89.5	87.8	86.3	80.0

- Trainings on other sample sizes are in partially done and inference is work in progress
- Expected to get the best performance when train on 40% sample size
 - Hopefully, we get good performance when evaluate on the 100% of the event size

Thanks