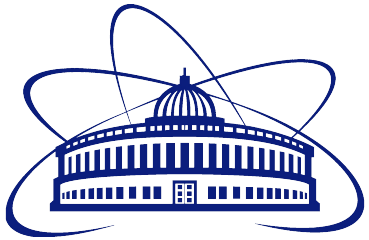
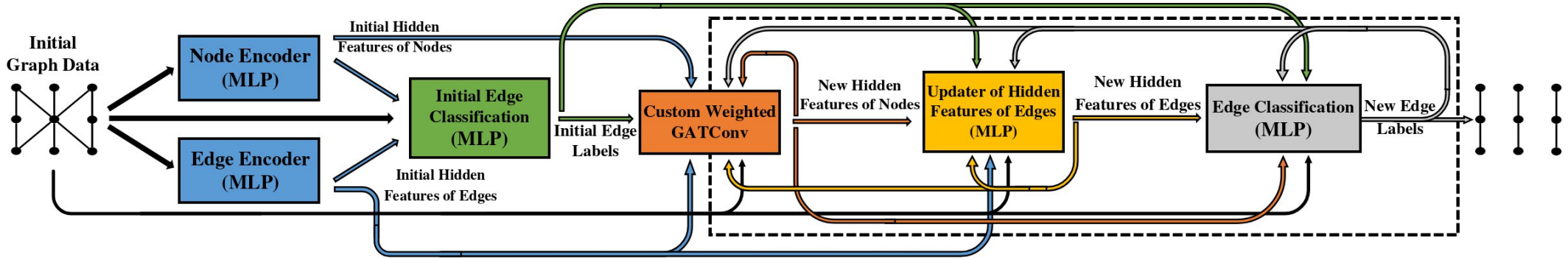
GNN Model for Particle Track Reconstruction Adopted for MPDRoot Framework

Yauheni Talochka
MLIT, JINR

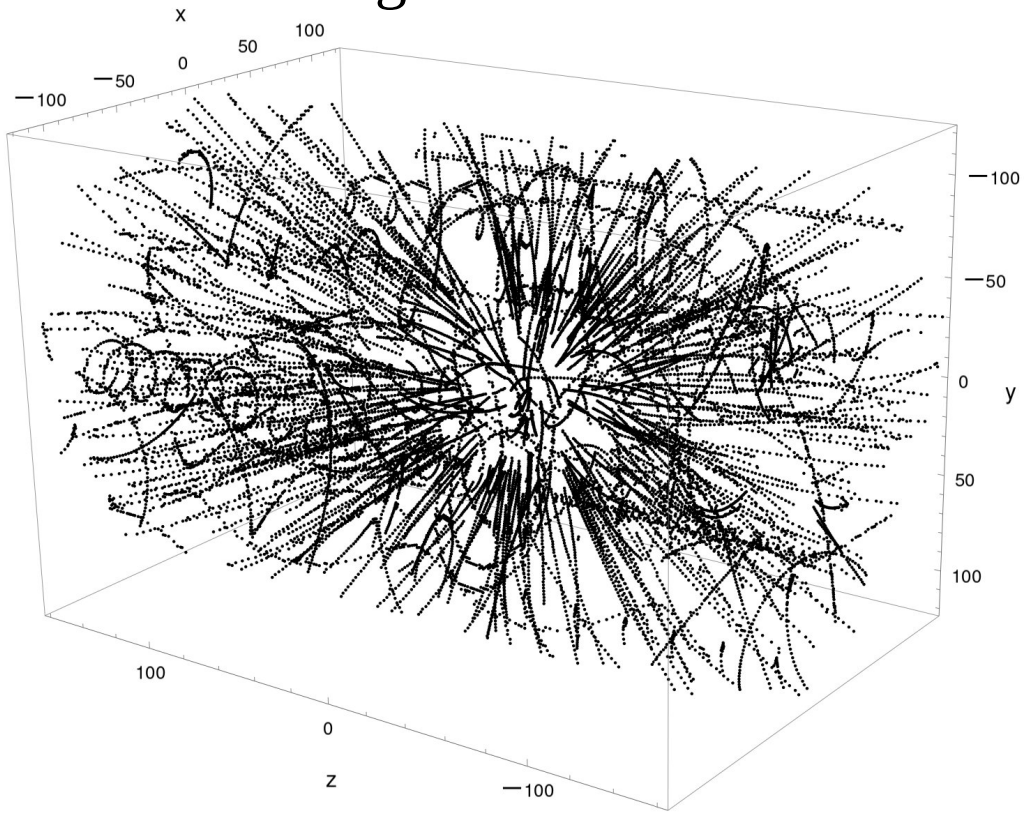
July 17, 2025



GNN model & Dataset



single MPD event

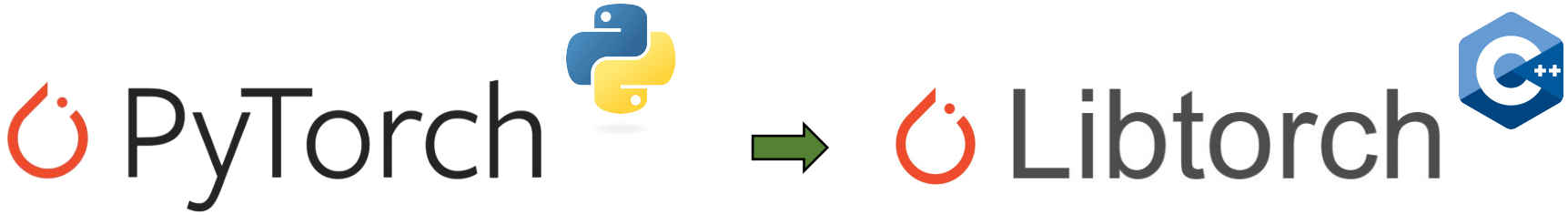


The node features are $\{r, \varphi, z\}$,
the edge features are $\{\Delta\eta, \Delta\varphi, \Delta\varrho, \Delta z\}$.

Here, r, φ and z are the cylindrical coordinates of a hit, η is the pseudorapidity, $\Delta\varrho^2 = \Delta\varphi^2 + \Delta\eta^2$.

Hits related to particles with $p_t < 150$ MeV are considered as noise, otherwise as useful information.

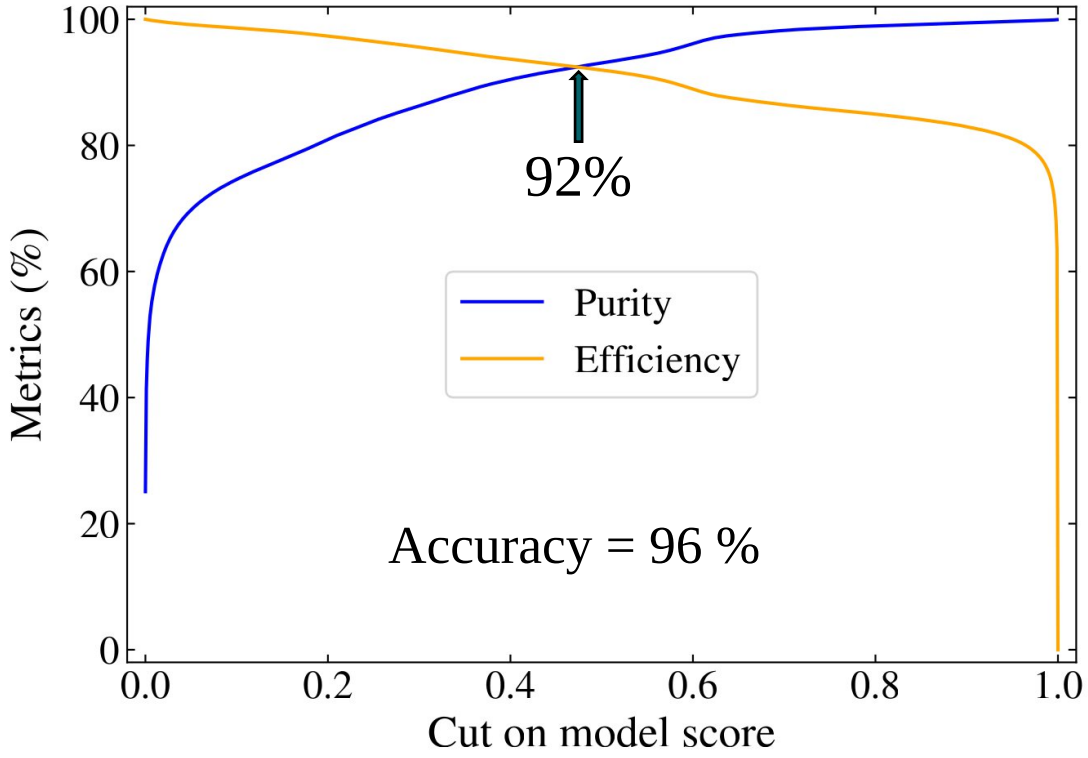
From Python to C++



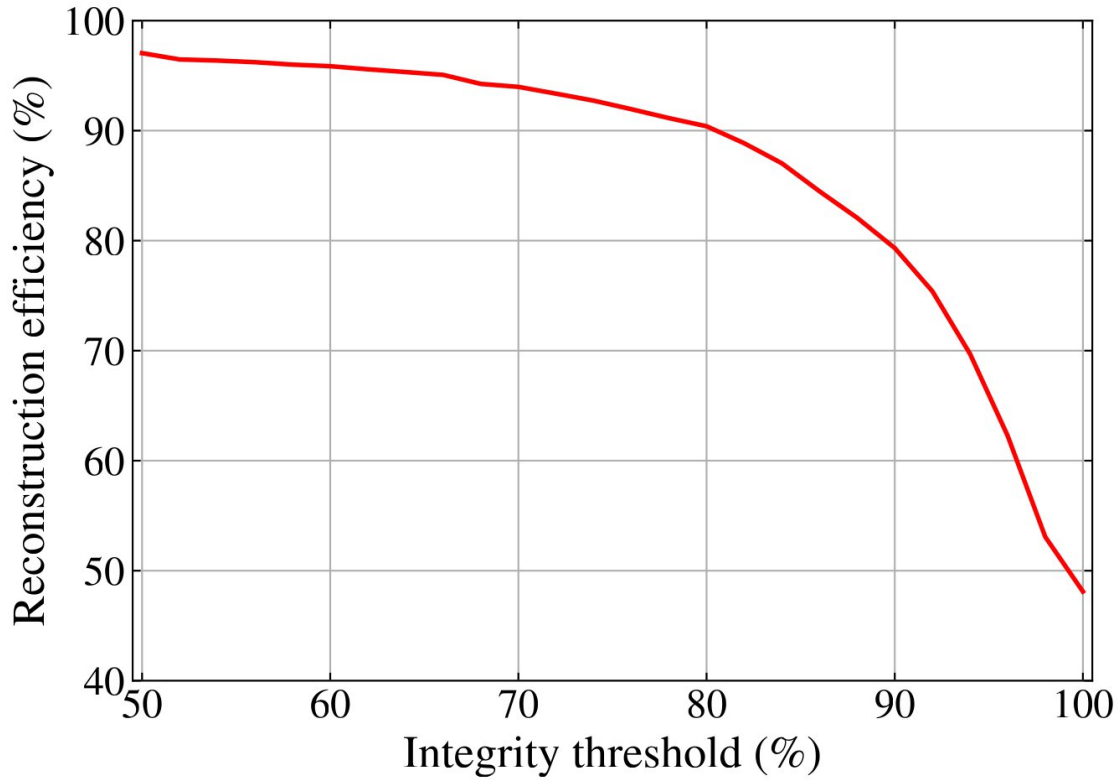
- LibTorch 2.7.1
- CUDA 12.8
- gcc 14.3.1
- yaml-cpp 0.7.0

https://docs.pytorch.org/cppdocs/index.html

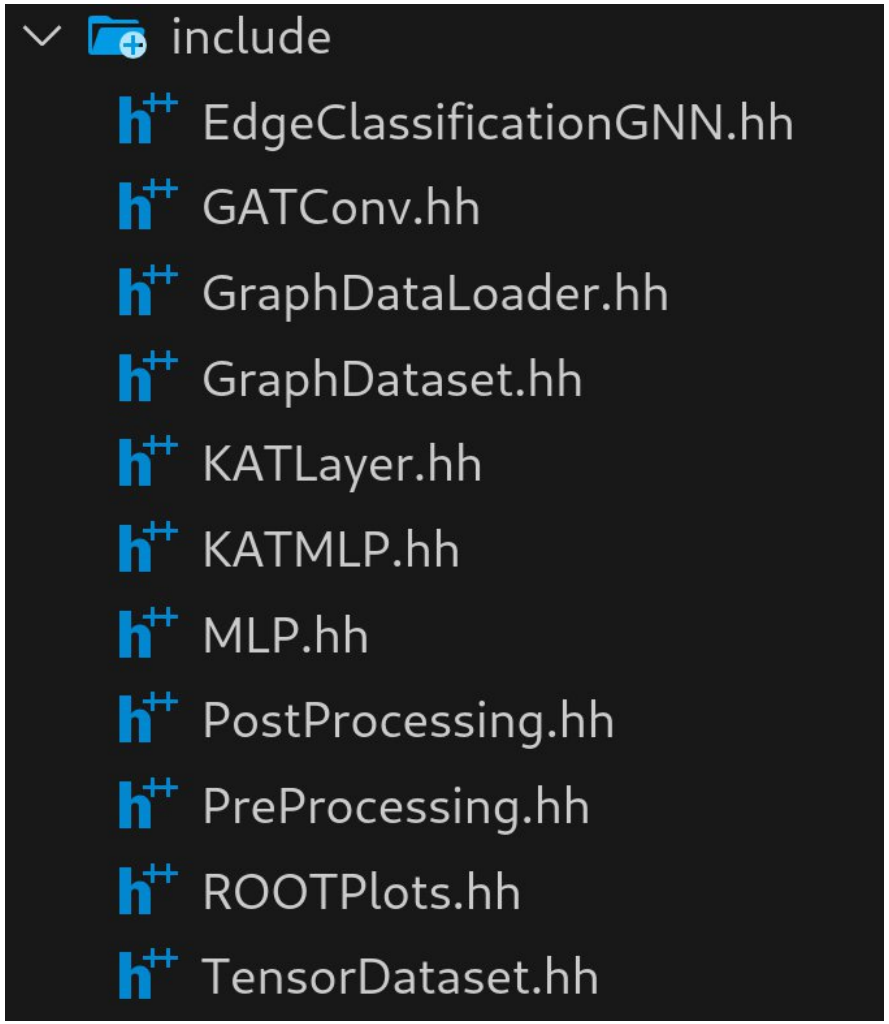
Training results

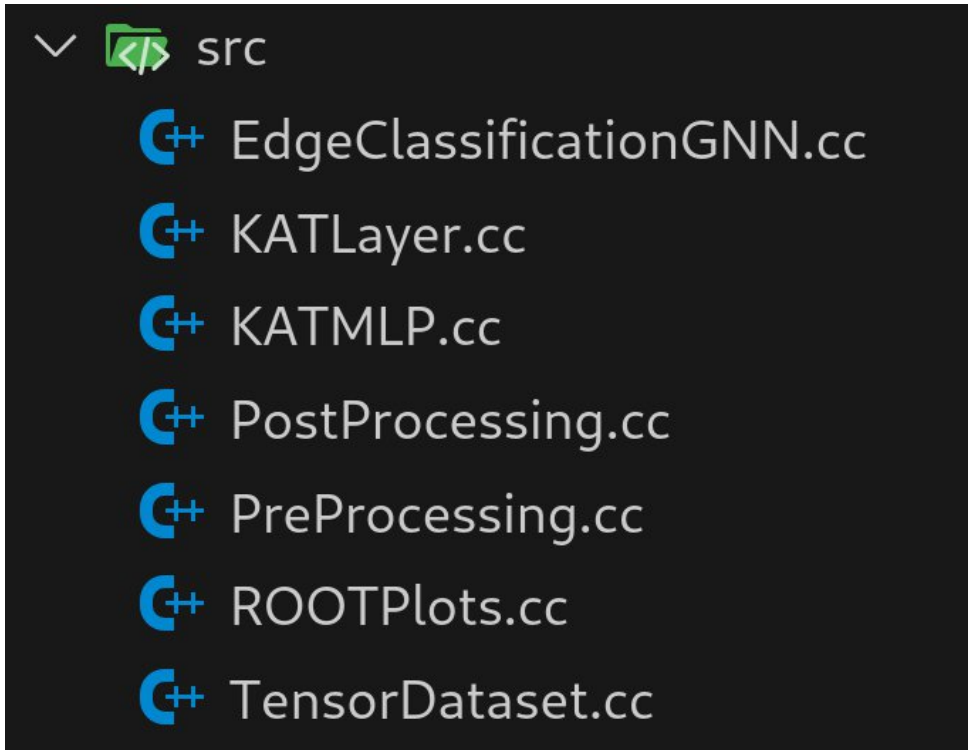


- Nvidia Geforce RTX 4070 Ti (16 GB)
- Batch size = 1
- Epoch number = 100
- Event number = 1000
- GPU RAM usage for training = 14 GB
- Training time = 40 min



libTorchNetworks.so





PreProcessing & GraphSample

```
PreProcessing::PreprocessingParams params = PreProcessing::LoadConfig("../configs/preprocessing_parameters.yaml");
```

preprocessing_parameters.yaml

configs > preprocessing_parameters.yaml

```yaml
input_dir: ../MPD_dataset/MPD_events/
output_dir: ../train_graphs/
dataset_size: 1000

selection:
    dphi_max: 0.06
    z0_max: 400
    chi_max: 1.6
    d_min: 0
    d_max: 12
    pt_min: 0.15
    n_phi_sections: 1
    n_eta_sections: 1
    eta_min: -10
    eta_max: 10
    num_rows: 53
    num_sectors: 24
    rmax: 124.0
    zmax: 169.0
```

```
struct Hit
{
    int hit_id;
    float z;
    float r;
    float phi;
    int row_id;
    int sector_id;
    int track_id;
    float pt;
    int id;
};
```

- processing time: 50 ms/event
- cpu: Intel Core i7-12700H (20 cores)
- gpu: Nvidia Geforce RTX 3060

```
std::vector<PreProcessing::Hit> hits;

std::vector<GraphSample> graphs;

PreProcessing::ProcessEvent(hits, params, graphs, /*train =*/ false);
```

```
struct GraphSample
{
    torch::Tensor edge_index;   // [2, num_edges]
    torch::Tensor node_attr;    // [num_nodes, input_size]
    torch::Tensor node_hit_id;  // [num_nodes, 1]
    torch::Tensor edge_attr;    // [num_edges, edge_attr_size]
    torch::Tensor answer;       // [num_edges, 1]
...
```

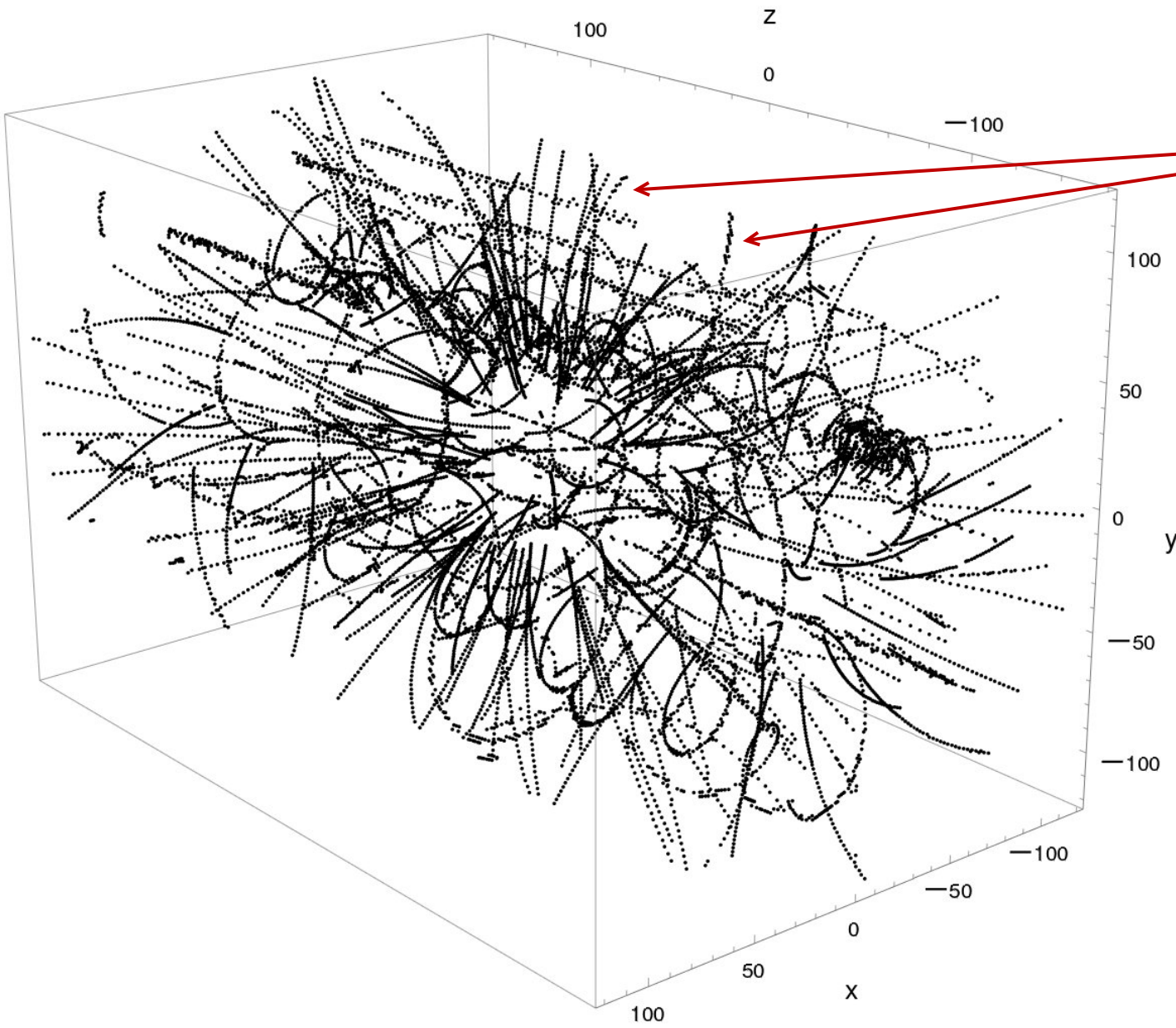
GraphSample Evaluation

```
auto model = EdgeClassificationGNN(node_attr_size, edge_attr_size);
model->to(torch::kCUDA);

try
{
    model->load_model(saved_model_file);
}
catch(const std::exception& ex)
{
    std::cerr << "Error: " << ex.what() << '\n';
    std::exit(1);
}
```

```
model->eval();
torch::NoGradGuard no_grad;

std::vector<std::vector<std::set<int>>> results;

for (auto& graph : test_data)
{
    graph = graph.to(torch::kCUDA);

    auto edge_index = graph.edge_index;
    auto node_attr  = graph.node_attr;
    auto edge_attr  = graph.edge_attr;

    graph.answer = model->forward(edge_index, node_attr, edge_attr);

    graph = graph.to(torch::kCPU);

    results.push_back(PostProcessing::GetTracks(graph, threshold));
}
```

- evaluation time: 30 ms/event
- cpu: Intel Core i7-12700H (single core)
- gpu: Nvidia Geforce RTX 3060

One issue



split tracks

The hit closest to the point averaged over the φ angle is used in each row for each particle to build smooth tracks for training the GNN model.

Thank you for attention!