Object Condensation Tracking

Gage DeZoort\textsuperscript{1} @gageDeZoort

Kilian Lieret\textsuperscript{1,2} @klieret

\textsuperscript{1}Princeton University, \textsuperscript{2}IRIS-HEP

With contributions from
Javier Duarte, Peter Elmer,
Isobel Ojalvo, Savannah Thais
Tracking

Point cloud
(coordinates of hits in detector)

Different colors = different particles

noise
Tracking as an edge classification task

**Edge construction**
(geometric constraints, module map, latent space)

**Postprocessing**
(e.g., “graph walking”)

**Edge Classifier (EC)**
Graph neural network

Thicker line = higher assigned probability

**Fitting**

Object Condensation Tracking | Gage deZoort, Kilian Lieret
Tracking with Object Condensation: Vision

Learnt latent space
Hits already clustered by particle; Clusters can be collected trivially

Repulsion & attraction of points in latent space

Condensation point
Represents the track, can learn track parameters like pT (WIP)
Object condensation in action

2D latent space; random selection of particles colored
Early simplified study (much fewer hits than in real life)

Click here if video doesn’t play
Object condensation: Our current pipeline

**STAGE 1: EC**
Graph construction based on geometric cuts

**EC GNN**
Loss fct = focal loss for pt > 0.9 hits

**STAGE 2: OC**

**OC GNN**
Learnt latent space

**STAGE 3: Collect clusters**

**DBSCAN**

- All three stages have their own hyperparameters
- Can be trained/optimized separately (fixing the previous stage)
Object condensation: Training losses

GNN predicts condensation likelihoods (CL) for every hit. Hit with max CL for particle* is condensation point (CP)

*during inference: for cluster

Attractive loss function
- rewards hits close to their CP
- quadratic potential
  - Attraction stronger if CP’s CL is high

Repulsive loss function
- penalizes hits close to other CP
- hinge loss: no more repulsion after certain distance
  - repulsion stronger for strong CP CLs

Background loss function
- noise hits should have low CL

Loss functions implemented from Kieseler 2020 (2002.03605)
• Full event is sectorized in 32 sectors (see 2103.16701); 5 random sectors per batch

• **Graph construction**
  - Currently: **Geometric cuts** only (see 2103.16701)
  - Soon: Comparison to module map
  - Mid term: transitioning to a point cloud network

• Main building block: **Interaction Networks** (1612.00222)

• **Edge classification** (EC) performance is vital:
  - Using FocalLoss (https://arxiv.org/abs/1708.02002) for class imbalance
  - Ignoring false negatives for edges connecting $p_T < 0.9$ GeV hits
  - EC threshold is around maximum attainable MCC (and this is used to rank different ECs)

• Track condensation network starts from edge classification latent space

• **Condensation space** dimension is $\sim 10$
Metrics

**Perfect**
Cluster contains only hits from one particle and no hits outside of cluster

Clusters with < 3 hits or non-reconstructable majority particle are discarded

Perfect efficiency = 1/5
Perfect fakes = 5/5

**LHC**
Cluster contains ≥ 75% hits from one particle

#clusters with ≥ 3 hits & majority particle reconstructable

LHC efficiency = 2/5
LHC fakes = 4/6

**Double Majority**
Cluster contains ≥ 50% hits from one particle and this particle has < 50% of its hits outside

#reconstructable particles

DM efficiency = 2/5
DM fakes = 4/5

We also evaluate these metrics at pT thresholds: pT cut is applied to majority particle of cluster or particle (this is not a truth cut on the data, but simply a efficiency vs pT study)

Reconstructable: ≥ 3 hits
**Most recent result**

Regarding 🍏 to 🍎 comparisons for HL-LHC benchmarking:

- Evaluated on trackML 2.0 dataset (generated with the ACTS geometry)
- **Pixel layers only: This might be a harder problem than using the full detector** (very dense regions)!
- Full trackML detector results very soon

<table>
<thead>
<tr>
<th></th>
<th>Perfect (&quot;cluster = particle&quot;)</th>
<th>LHC (&quot;homogeneous clusters&quot;)</th>
<th>Double Majority (&quot;1:1 match cluster &lt;&gt; particle&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_T \geq 1.5 \text{ GeV}$</td>
<td>70%</td>
<td>97% fake: 2.9%</td>
<td>91% fake: 3.4%</td>
</tr>
<tr>
<td>$p_T \geq 0.9 \text{ GeV}$</td>
<td>72%</td>
<td>97% fake: 2.7%</td>
<td>92% fake: 3.4%</td>
</tr>
</tbody>
</table>

No truth cut on $p_T$ or other simplification

EC FocalLoss currently set to ignore low $p_T$ false negatives \(\rightarrow\) lower performance for low $p_T$
Summary

• Proof of concept for object condensation applied to the HL-LHC tracking challenge without truth cuts
• **Promising performance on trackML pixel layer:** > 90% of particles with $p_T > 0.9$ GeV are uniquely (double majority) matched to a cluster
• Currently working on applying to full detector geometry
• Much to be explored: Point cloud networks and more
• **Fully open-source framework:** Let’s make prototyping new architectures for tracking accessible to everyone

**gnn_tracking** (Public)
Charged particle tracking with graph neural networks
- Python ⭐️ 13 🔍 MIT 🔧 7 🔴 60 (8 issues need help) 🔴 2 Updated yesterday

**hyperparameter_optimization** (Public)
Hyperparameter optimization submission & helper scripts
- Python ⭐️ 3 🔍 MIT 🔧 0 🔴 9 🔴 0 Updated 4 days ago

**tutorials** (Public)
Tutorials for onboarding of the GNN Tracking project
- Jupyter Notebook ⭐️ 3 🔍 MIT 🔧 0 🔴 1 🔴 0 Updated last week

[GitHub link: github.com/gnn-tracking]
Shoutouts: More object condensation

Lea Reuter
Object condensation tracking at Belle II

Daniel Murnane/Paolo Calafiura
Object Condensation with “Influencer” approach
Backup
Point cloud sectorization
Architecture

Input node feats  Input edge feats

Edge Classification
Edge/node encoder (MLP) +
5 layers of INs w/ residual connections

node latent space  edge latent space

MLP

edge weights

orphan node pruning  threshold mask

Track condensor
3 layers of INs w/ residual connections

Condensation space

Condensation Likelihoods