

Thoughts on dl-clus

Review of current dl-clus, nodes

hit — the converter stores far more than the model consumes. Per plane node it writes `pos` and 15 feature columns; the load-time `Transform` appends a plane index (16 cols), then keeps only `feat_keep = in_features - pos_dim = 4 - 2 = 2` columns and prepends `pos`:

Stored tensor	Columns (order as written by <code>pywcml</code>)	Consumed in prod?
<code>hit.pos</code> [N,2]	charge-weighted (x_{drift} , wire-pitch) position of the stripe	yes — prepended, becomes cols 0–1 of <code>hit.x</code>
<code>hit.x</code> (stored) [N,15]+plane	0–1: <code>Q_tot</code> (summed stripe charge), <code>dQ_err</code> (mean charge error)	yes — kept, become cols 2–3 of <code>hit.x</code>
	2–4: <code>n_hits</code> , <code>pitch_min</code> , <code>pitch_max</code>	no — trimmed by Transform
	5–8: vertex feats <code>vtx_dist</code> , <code>vtx_dx</code> , <code>vtx_dy</code> , <code>vtx_dz</code> (mean over linked blobs)	no — trimmed
	9–14: sidecar <code>d_wall</code> , <code>d_top</code> , <code>linearity</code> , <code>sphericity</code> , <code>ty</code> , <code>tz</code> ; 15: plane index	no — trimmed

Final encoder input: `hit.x` [N,4] = [x_{drift} , wire_pitch, `Q_tot`, `dQ_err`] → InputNorm → Linear(4→256). Note `in_features` counts *pos included* (`feat_keep = in_features - 2`).

sp — spacepoints have no stored `x`; their embedding is *built* by the encoder:

Stored tensor	Content	Consumed in prod?
<code>sp.pos</code> [M,3]	blob centroid (x, y, z)	not by the encoder — used at runtime for the edge head's geometric features ($\Delta x, \Delta y, \Delta z, \ \Delta r\ $) and by eval/Bee
<code>sp.features</code> [M,6]	0: <code>charge</code> · 1: <code>reco_cluster_id</code> · 2–5: <code>vtx_dist</code> , <code>vtx_dx</code> , <code>vtx_dy</code> , <code>vtx_dz</code>	cols 0–1 only (<code>use_vtx_features=False</code>); LayerNorm → concat with hit-aggregated embedding → Linear+GELU → <code>sp.x</code> [M,64]

The vertex columns (2–5) are treated as truth-adjacent ("cheating") features and are excluded in prod; the converter also zeroes them on ghost blobs. Caveat: the encoder comment labels col 1 "hit_count", but the converter actually writes `reco_cluster_id` there — the code comment is stale.

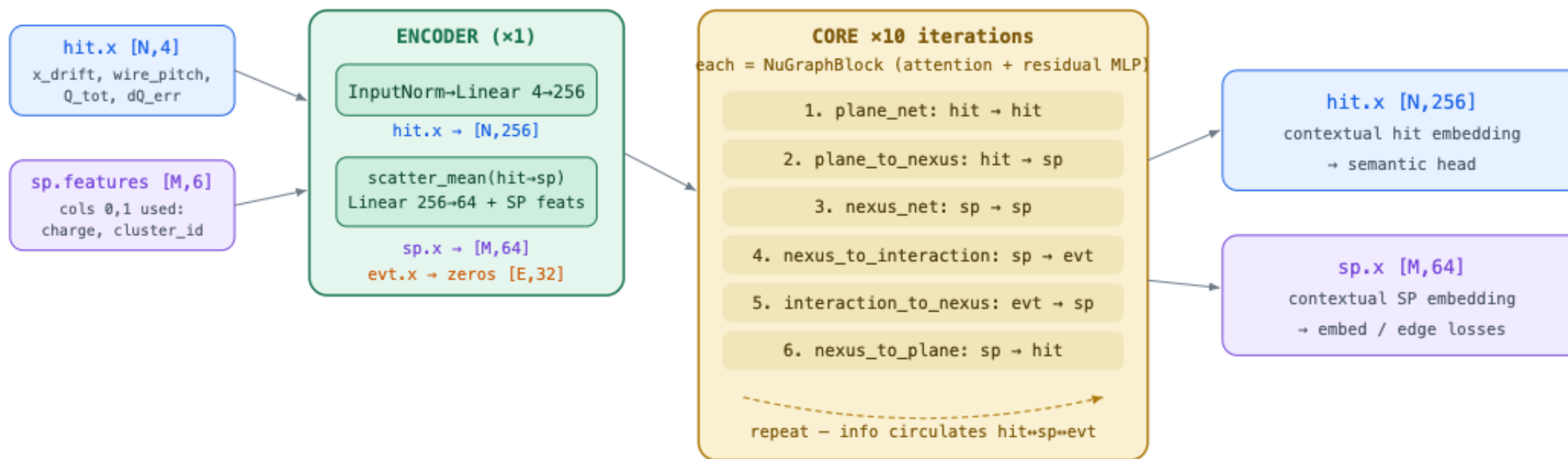
evt — **no input features**. `Transform` creates `evt.x` as an empty [1,0] tensor and the encoder overwrites it with zeros [1,32]; its embedding is populated purely by message passing from spacepoints.

Edges

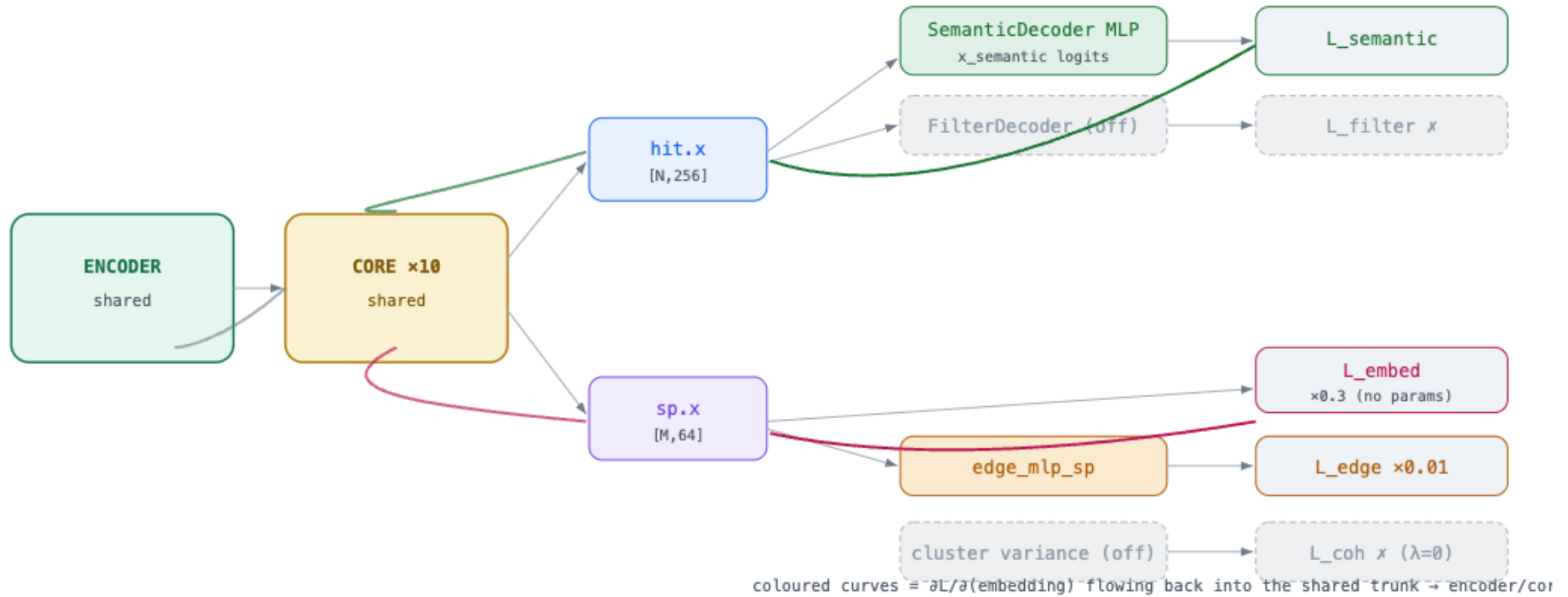
Edge type	Origin	How it's built	Input features
(hit, delaunay-planar, hit)	algorithm , at conversion	2D Delaunay triangulation (<code>scipy</code>) over the stripe positions of each plane (<code>triangulation_edges</code> in <code>pywcml/geometry.py</code>); stored per plane as <code>(u/v/y, plane, u/v/y)</code> , merged into one <code>hit</code> store by <code>Transform</code>	none
(hit, nexus, sp)	algorithm , at conversion	blob-corner projection: each blob's corners are projected onto the plane's pitch axis (<code>project_corners</code>); a stripe within <code>x_tolerance</code> in drift and overlapping the projected pitch range ($\pm$ <code>projection_tolerance</code>) is linked to the blob	none
(sp, nexus, sp)	from file (Wire-Cell)	the NPZ's <code>ppedges</code> (blob-adjacency from WCT clustering) mapped to blob pairs; edges touching ghost blobs pruned; optionally unioned with supervision edges (<code>config.merge_sup_edges_into_mp</code>)	none
(sp, supervision, sp)	from file (truth)	the NPZ's truth <code>edge_index</code> ; stored as <code>sp.edge_label_index</code> and promoted to a proper edge store by <code>dataset.get()</code> . Not used for message passing — only the edge head reads it (filtered to <code>edge_labelable==1</code>)	labels only: <code>edge_y</code> (1 iff endpoints share the true track id), <code>edge_labelable</code> (ghost-free)
(hit, in, evt) · (sp, in, evt)	algorithm , at load	trivial: <code>Transform</code> connects every hit / sp node to the single <code>evt</code> node	none
(hit, cluster-truth, particle-truth)	truth , at load	built by <code>Transform</code> from <code>hit.y_instance</code> ; feeds the NuGraph3 instance decoder, which is not enabled in prod	none

Message passing

https://github.com/HaiwangYu/nugraph/blob/feature/nugraph4-beta/docs/nugraph4_architecture.html



Learning head



nc-sideband trad vs. dlclus

https://docs.google.com/document/d/1Z02r_MVp7rSEg-ZnCgtyODSw1_F60Vs5tqliZhSVvw/edit?pli=1&tab=t.3ys1434aezqw

Traditional fails at matching:

<https://www.phy.bnl.gov/twister/bee/set/b3d18304-227d-4c37-8ee9-0a664c744117/event/2/>

<https://www.phy.bnl.gov/twister/bee/set/b3d18304-227d-4c37-8ee9-0a664c744117/event/9/>

Corresponding dl-clus:

<https://www.phy.bnl.gov/twister/bee/set/df1cf6d1-1a82-4434-9424-7f62f7e48ead/event/0/>

<https://www.phy.bnl.gov/twister/bee/set/df1cf6d1-1a82-4434-9424-7f62f7e48ead/event/7/>

Steamline the workflow

- Needed for both inferencing and new production
- New sample from **TensorSetLabeler**
 - Nodes, node labeler – should be more precise
 - Edges:
 - hit-hit – prepare in python?
 - sp-sp – seems to have issue – fixing now
 - hit-sp should be more precise
- From Sne:
 - Avinay's sample: <https://www.phy.bnl.gov/twister/bee/set/7f403353-abfc-43d0-989b-f528a7fbbdd1/event/list/>
 - New sample: <https://www.phy.bnl.gov/twister/bee/set/48016c88-9ca5-4e6f-a105-3e1cd2b4cd0b/event/list/>

Potential improvements and integration in WC

- From Xin:
 - Empirical exploration
 - Theoretical explanation
 - Failure examination

From information point of view, how to get information from unconnected parts?

- Right now, only through the `evt` node, is this good enough? Enlarge its feature? to 1024 or even larger?
- Need to build cross cluster edges? How to?
- Ditch the sp-sp edges and use 3D delaunay?

Including light information?

Plan for new data

Next old new consistency check:

Data:

- ✓ RSE list
- ✓ old h5
- ✓ new h5

Investigation setup on gpvm:

- ☐ AI checks
- ☐ graph vis
- ☐ Inferencing test

- Issue to track the investigation: <https://github.com/HaiwangYu/wire-cell-toolkit-ai-helper/issues/9>
- Computing: Is the sophia very occupied now?
- How to transfer data files between sophia.alcf.anl.gov and the sbndgpvm01.fnal.gov?

backups

run-eval-follow-sne-instruction

Avinay's input:

<https://www.phy.bnl.gov/twister/bee/set/92899ae6-9d85-4538-a6d3-18b223130c8b/event/0/>

2nd run 10 events: <https://www.phy.bnl.gov/twister/bee/set/65f6460e-9c79-4585-9eb9-6b8270ffec3a/event/list/>

TensorSetLabeler v2:

MC: <https://www.phy.bnl.gov/twister/bee/set/a8cfb0ca-fa33-4fba-8576-5024cb84babd/event/0/>

nuecc-lynn: <https://www.phy.bnl.gov/twister/bee/set/aac9f9af-8337-40a6-bfd3-5175097ce520/event/0/>