DL Clus, Jul 31

Haiwang Yu

Brookhaven National Laboratory

Status

1st round:

- resolved computing issue: solving, graph building
- ~900 samples, bag of script repo, "Hello, World!" GNN trained, bad perfomrance

2nd round:

- better structured codebase
- detailed check on input data
    - AI-Gen utilities to help
    - fixed mis-matching geom
    - init handscan to get a sense on the task

Brookhaven National Laboratory

Links

Code:

- old: https://github.com/HaiwangYu/sbnd-dl-clustering
- new: https://github.com/HaiwangYu/dlclus

Sample data:
/exp/sbnd/app/users/yuhw/dl-clustering/sample/20250618-geom-v0200
/exp/sbnd/app/users/yuhw/dl-clustering/sample/20250618

Handscan note + BEE:
https://docs.google.com/spreadsheets/d/13AsnURrziBjoR4_FYEOkYJjOMc-RNyrFIUZGN-kEftk/edit?gid=0#gid=0

Brookhaven National Laboratory

A "Hello, World!" GNN

```
GNNModel
│
├── Input Layer
│   └── Point Features (x) and Edge Connections (edge_index)
│
├── EdgeConv Layer 1
│   ├── Message Passing: Concatenates features of connected nodes
│   ├── Neural Network:
│   │   ├── Linear(input_dim*2 → hidden_dim)
│   │   ├── ReLU()
│   │   └── Linear(hidden_dim → hidden_dim)
│   └── Aggregation: Mean of neighbors' messages
│
├── ReLU Activation
│
├── EdgeConv Layer 2
│   ├── Message Passing: Concatenates features of connected nodes
│   ├── Neural Network:
│   │   ├── Linear(hidden_dim*2 → hidden_dim)
│   │   ├── ReLU()
│   │   └── Linear(hidden_dim → hidden_dim)
│   └── Aggregation: Mean of neighbors' messages
│
├── ReLU Activation
│
└── Output Layer
    └── Linear(hidden_dim → 2) [Binary classification: Non-neutrino/Neutrino]
```

How the Model Works

Input feature: x,y,z,q,blob_index

1. **Input**: Takes point cloud data (x) and their connectivity (edge_index)
   - x shape: [num_nodes, input_dim]
   - edge_index shape: [2, num_edges]
2. **EdgeConv Layers**: These are the key components of your GNN
   - Each EdgeConv layer:
     - Combines features from connected nodes (message passing)
     - Processes these features through a small neural network
     - Aggregates messages from neighbors using mean
3. **Learning Process**:
   - The model learns node representations that incorporate information from neighboring nodes
   - This captures local geometric structures in your point cloud
   - With two EdgeConv layers, each node's representation includes information from its 2-hop neighborhood
4. **Output**: For each node, predicts whether it belongs to a neutrino interaction (1) or not (0)

Brookhaven National Laboratory

EAF, training

- Loading the venv takes long time ~5min

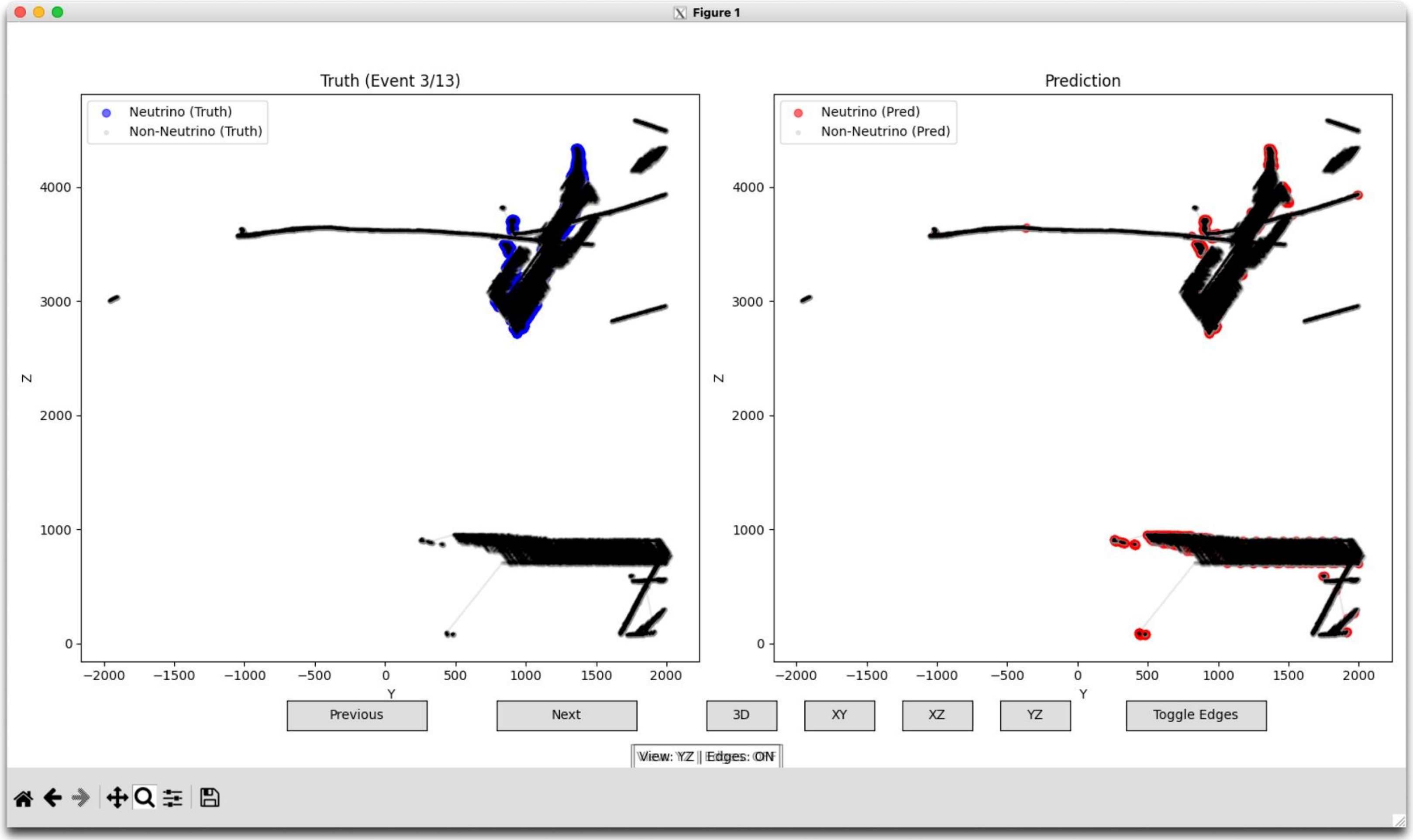
```
Found 917 data files to load
100%|
Successfully loaded 917 data files
Preparing datasets...
Train size: 458, Validation size: 458, Test size: 1
/exp/sbnd/app/users/yuhw/dl-clustering/venv_eaf/lib/python3.10/si
  warnings.warn(out)
Initializing model...
Starting training...
all_preds.shape: 458
all_labels.shape: 458
Epoch: 001, Loss: 26724.3620, Val Acc: 0.8636, Val F1: 0.0788
Saved model to ../models/best_model.pt
all_preds.shape: 458
all_labels.shape: 458
Epoch: 002, Loss: 14010.6604, Val Acc: 0.9458, Val F1: 0.0177
all_preds.shape: 458
all_labels.shape: 458
Epoch: 003, Loss: 12787.0734, Val Acc: 0.9520, Val F1: 0.0014
all_preds.shape: 458
all_labels.shape: 458
Epoch: 004, Loss: 12756.2652, Val Acc: 0.9513, Val F1: 0.0037
all_preds.shape: 458
all_labels.shape: 458
Epoch: 005, Loss: 11933.9169, Val Acc: 0.9511, Val F1: 0.0083
all_preds.shape: 458
all_labels.shape: 458
Epoch: 006, Loss: 11939.5326, Val Acc: 0.9513, Val F1: 0.0069
all_preds.shape: 458
all_labels.shape: 458
Epoch: 007, Loss: 12087.0750, Val Acc: 0.9499, Val F1: 0.0160
all_preds.shape: 458
all_labels.shape: 458
Epoch: 008, Loss: 11776.2444, Val Acc: 0.9518, Val F1: 0.0033
all_preds.shape: 458
all_labels.shape: 458
Epoch: 009, Loss: 12218.3793, Val Acc: 0.9518, Val F1: 0.0027
all_preds.shape: 458
all_labels.shape: 458
Epoch: 010, Loss: 12087.1230, Val Acc: 0.9490, Val F1: 0.0225
all_preds.shape: 458
all_labels.shape: 458

Test set evaluation:
Accuracy: 0.8636
Precision: 0.0582
Recall: 0.1218
F1 Score: 0.0788

real    11m25.183s
user    913m6.237s
sys     1m17.286s
```

round 2

Check input edges

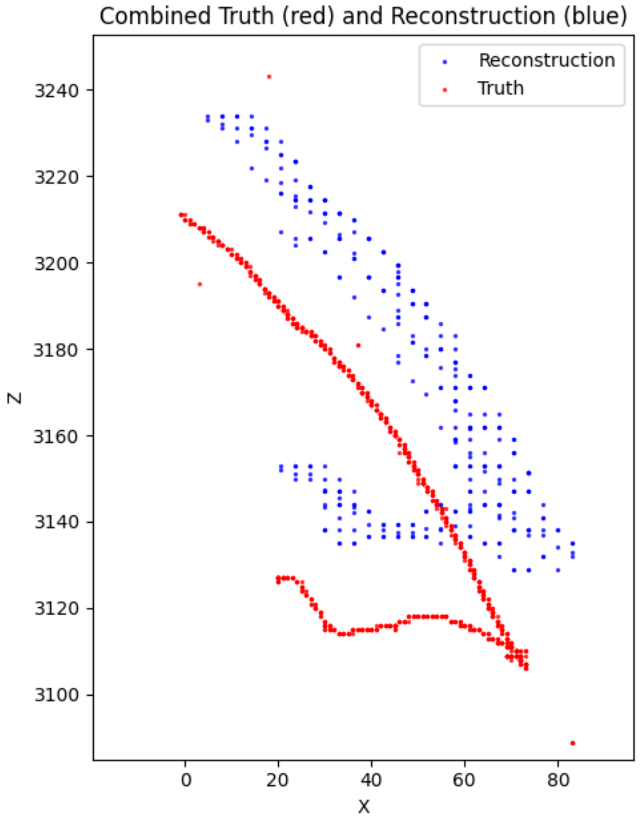


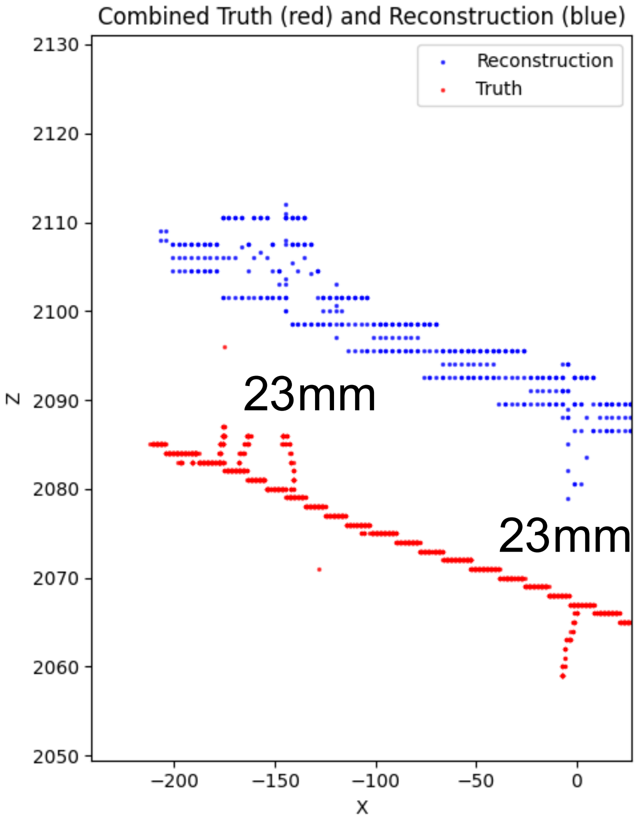
Brookhaven National Laboratory

z_offset

/exp/sbnd/app/users/yuhw/dl-clustering/sample/20250618/77451011_0/rec-apa1-9.npz

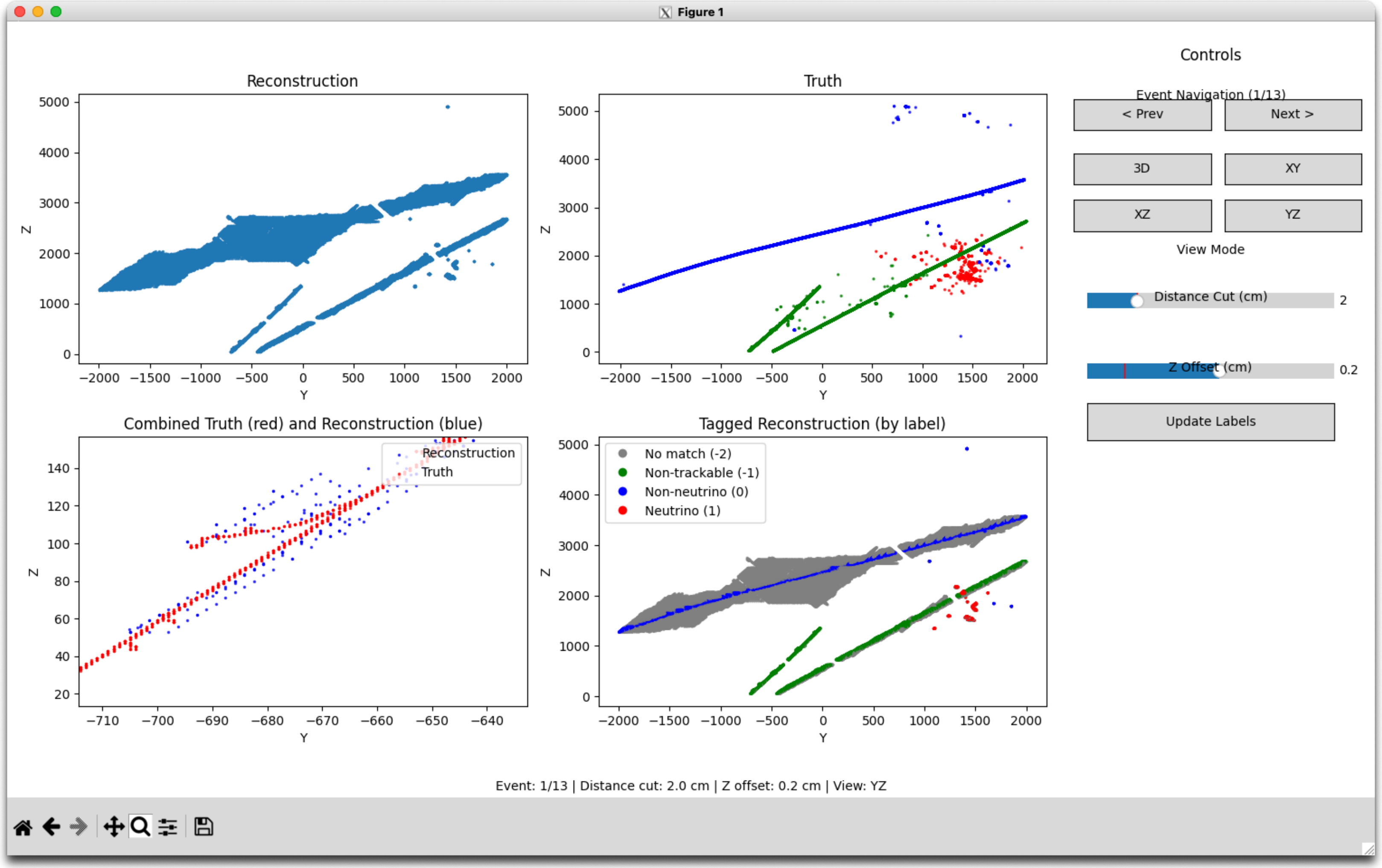
3151-3126 = 25



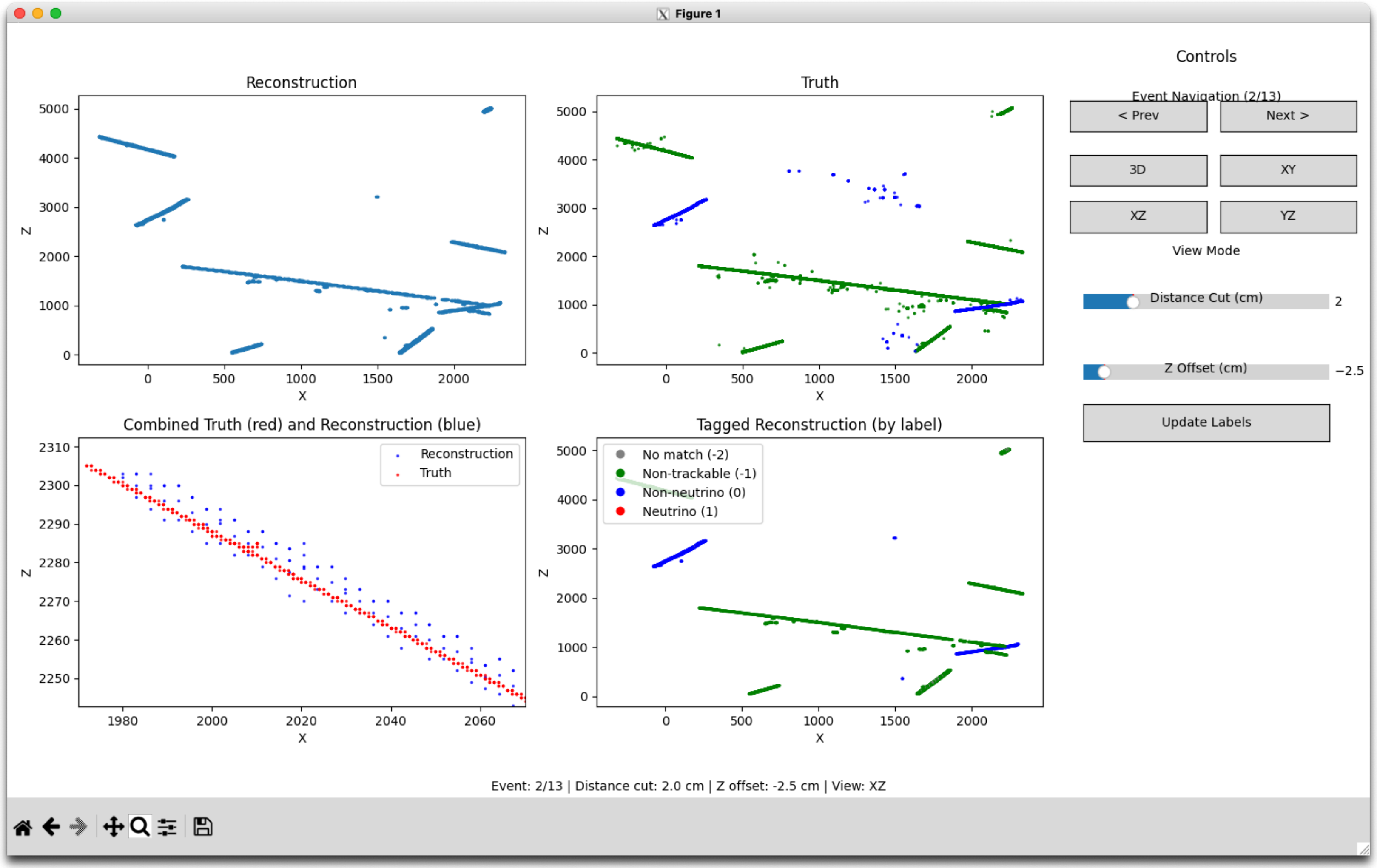


z_offset, APA0: 0

APA0



z_offset, APA1: -2.5cm



sbnd-wires-geometry-v0202.json.bz2

```cpp
std::vector<WireCell::Binning> collate_byplane(const ITrace::vector& traces, const IAnodePlane::pointer anode,
                                               ITrace::vector byplane[])
{
    std::vector<int> uvwt[4];
    for (auto trace : traces) {
        const int chid = trace->channel();
        std::cout << "[yuhw] chid: " << chid << std::endl;
        auto wpid = anode->resolve(chid);
        const int iplane = wpid.index();
        if (iplane < 0 || iplane >= 3) {
            THROW(RuntimeError() << errmsg{"Illegal wpid"});
        }
        uvwt[iplane].push_back(chid);
        byplane[iplane].push_back(trace);
        uvwt[3].push_back(trace->tbin());
        uvwt[3].push_back(trace->tbin() + trace->charge().siz
    }
```

[yuhw] chid: 5632

```
An exception was thrown while processing module WireCellToolkit/sig2img run: 1 subRun: 1 event: 5
../root/src/MagnifySink.cxx(127): Throw in function collate_byplane
Dynamic exception type: boost::wrapexcept<WireCell::RuntimeError>
[tag_stacktrace*] =  0# collate_byplane(std::vector<std::shared_ptr<WireCell::ITrace const>, std::all
, std::vector<std::shared_ptr<WireCell::ITrace const>, std::allocator<std::shared_ptr<WireCell::ITrace cons
 1# WireCell::Root::MagnifySink::operator()(std::shared_ptr<WireCell::IFrame const> const&, std::shar
 2# WireCell::IFunctionNode<WireCell::IFrame, WireCell::IFrame>::operator()(boost::any const&, boost:
 3# WireCell::Pgraph::Function::operator()() in /exp/sbnd/app/users/yuhw/opt/lib/libWireCellPgraph.so
 4# WireCell::Pgraph::Graph::execute() in /exp/sbnd/app/users/yuhw/opt/lib/libWireCellPgraph.so
 5# WireCell::Pgraph::Pgrapher::execute() in /exp/sbnd/app/users/yuhw/opt/lib/libWireCellPgraph.so
 6# WireCell::Main::operator()() in /exp/sbnd/app/users/yuhw/opt/lib/libWireCellApps.so
 7# wcls::WCLS::process(art::Event&) in /cvmfs/larsoft.opensciencegrid.org/products/larwirecell/v10_0
 8# art::detail::Producer::doEvent(art::EventPrincipal&, art::ModuleContext const&, std::atomic<unsig
cegrid.org/products/art/v3_14_04/slf7.x86_64.e26.prof/lib/libart_Framework_Core.so
 9# art::Worker::runWorker(art::EventPrincipal&, art::ModuleContext const&) in /cvmfs/larsoft.opensci
10# 0x00007F568E0407E3 in /cvmfs/larsoft.opensciencegrid.org/products/art/v3_14_04/slf7.x86_64.e26.pr
k<hep::concurrency::SerialTaskQueue::QueuedTask>::execute(tbb::detai
ncy.so
larsoft.opensciencegrid.org/products/tbb/v2021_9_0/Linux64bit+3.10-2
larsoft.opensciencegrid.org/products/tbb/v2021_9_0/Linux64bit+3.10-2
larsoft.opensciencegrid.org/products/tbb/v2021_9_0/Linux64bit+3.10-2
wait(tbb::detail::d1::task&, tbb::detail::d1::task_group_context&, t
4bit+3.10-2.17-e26/lib/libtbb_debug.so.12
cess<(art::Level)4>() in /cvmfs/larsoft.opensciencegrid.org/products
cess<(art::Level)3>() in /cvmfs/larsoft.opensciencegrid.org/products
cess<(art::Level)2>() in /cvmfs/larsoft.opensciencegrid.org/products
cess<(art::Level)1>() in /cvmfs/larsoft.opensciencegrid.org/products
cess<(art::Level)0>() in /cvmfs/larsoft.opensciencegrid.org/products
pletion() in /cvmfs/larsoft.opensciencegrid.org/products/art/v3_14_0
st::program_options::options_description&, std::vector<std::unique_p
ete<art::OptionsHandler> > > >&&) in /cvmfs/larsoft.opensciencegrid.
23# artapp(int, char**, bool) in /cvmfs/larsoft.opensciencegrid.org/products/art/v3_14_04/slf7.x86_64
24# main in lar
25# __libc_start_main in /lib64/libc.so.6
26# 0x00000000004012D0 in lar

[WireCell::tag_errmsg*] = Illegal wpid
```

```
local chsel_pipes =
  g.pnode({
    type: 'ChannelSelector',      You, 8 months ago via  PR #362 • copied from Ewerton, can be run.
    //name: 'chsel%d' % n,
    name: 'chsel%d' % anode.data.ident,
    data: {
      channels: std.range(5638 * anode.data.ident, 5638 * (anode.data.ident + 1) - 1), // v0200: 5632, v0202: 5638
      //tags: ['orig%d' % n], // traces tag //commented? Ewerton 2023-09-xx
      tags: ['gauss%d' % anode.data.ident, 'wiener%d' % anode.data.ident], // changed Ewerton 2023-09-27
    },
  }, nin=1, nout=1, uses=[anode]),
```

Brookhaven National Laboratory

wirecell-util plot-wires sbnd-wires-geometry-v0200.json.bz2 sbnd-wires-geometry-v0200.pdf
wirecell-util plot-wires sbnd-wires-geometry-v0202.json.bz2 sbnd-wires-geometry-v0202.pdf

v0200: 5632 chan/apa

```
anodeID:0 faceID:0 planeID:0
        wcount:0 wind:0 wident:0 chan:0
        wcount:1 wind:1983 wident:1983 chan:1983
anodeID:0 faceID:0 planeID:1
        wcount:0 wind:1984 wident:0 chan:1984
        wcount:1 wind:3967 wident:1983 chan:3967
anodeID:0 faceID:0 planeID:2
        wcount:0 wind:3968 wident:0 chan:3968
        wcount:1 wind:5631 wident:1663 chan:5631
anodeID:1 faceID:1 planeID:0
        wcount:0 wind:7615 wident:1983 chan:7615
        wcount:1 wind:5632 wident:0 chan:5632
anodeID:1 faceID:1 planeID:1
        wcount:0 wind:9599 wident:1983 chan:9599
        wcount:1 wind:7616 wident:0 chan:7616
anodeID:1 faceID:1 planeID:2
        wcount:0 wind:11263 wident:1663 chan:11263
        wcount:1 wind:9600 wident:0 chan:9600
```

v0202: 5638 chan/apa

```
anodeID:0 faceID:0 planeID:0
        wcount:0 wind:0 wident:0 chan:0
        wcount:1 wind:1983 wident:1983 chan:1983
anodeID:0 faceID:0 planeID:1
        wcount:0 wind:1984 wident:0 chan:1984
        wcount:1 wind:3967 wident:1983 chan:3967
anodeID:0 faceID:0 planeID:2
        wcount:0 wind:3968 wident:0 chan:3968
        wcount:1 wind:5637 wident:1669 chan:5637
anodeID:1 faceID:1 planeID:0
        wcount:0 wind:7621 wident:1983 chan:7621
        wcount:1 wind:5638 wident:0 chan:5638
anodeID:1 faceID:1 planeID:1
        wcount:0 wind:9605 wident:1983 chan:9605
        wcount:1 wind:7622 wident:0 chan:7622
anodeID:1 faceID:1 planeID:2
        wcount:0 wind:11275 wident:1669 chan:11275
        wcount:1 wind:9606 wident:0 chan:9606
```

Work in progress

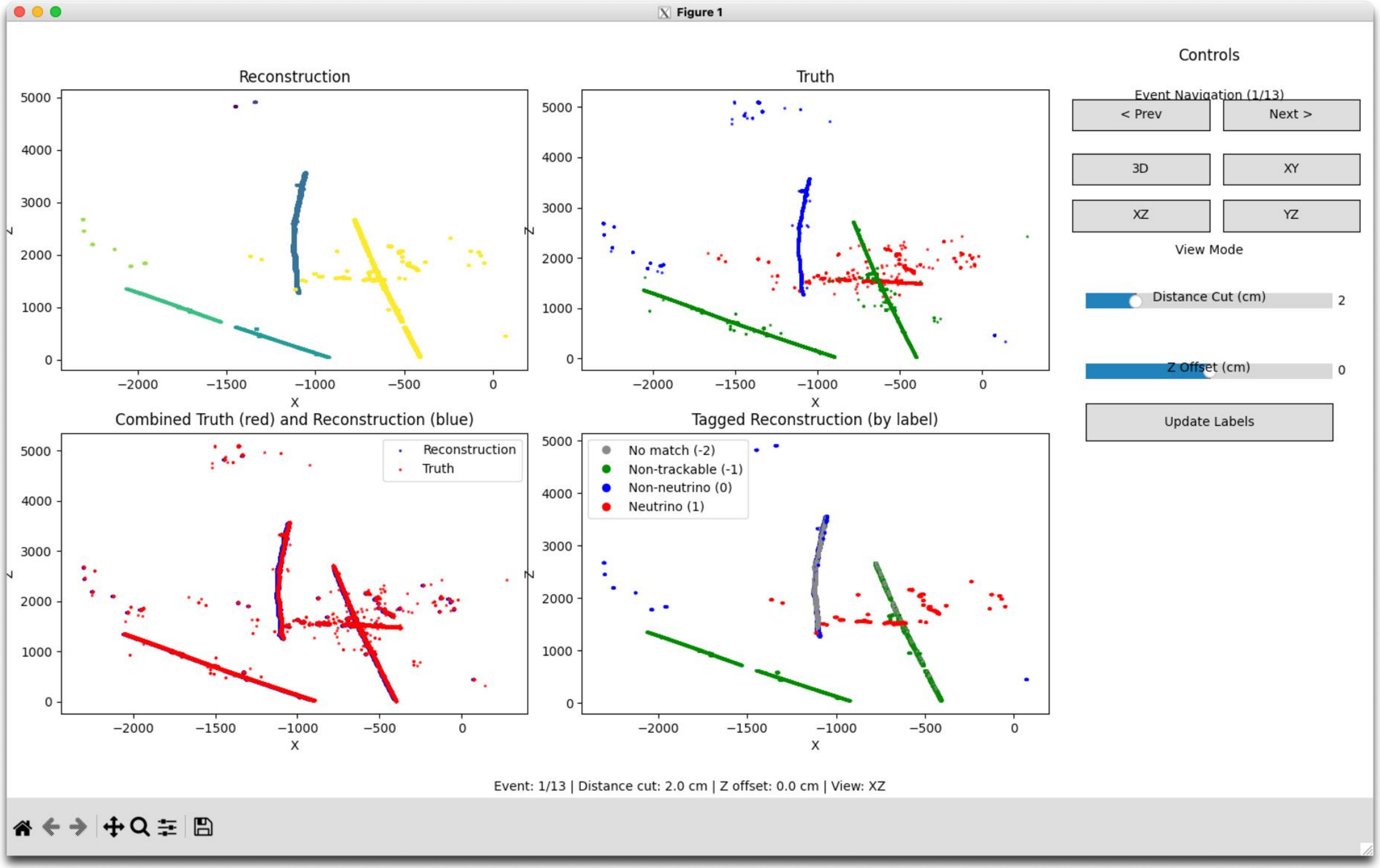
✓ Why the z_offset?

- How to handle isocronous?
  - prioritize eff.
    - full blob tagging?
    - full trad cluster tagging?
  - bigger issue for long tracks, seems OK for NCpi0

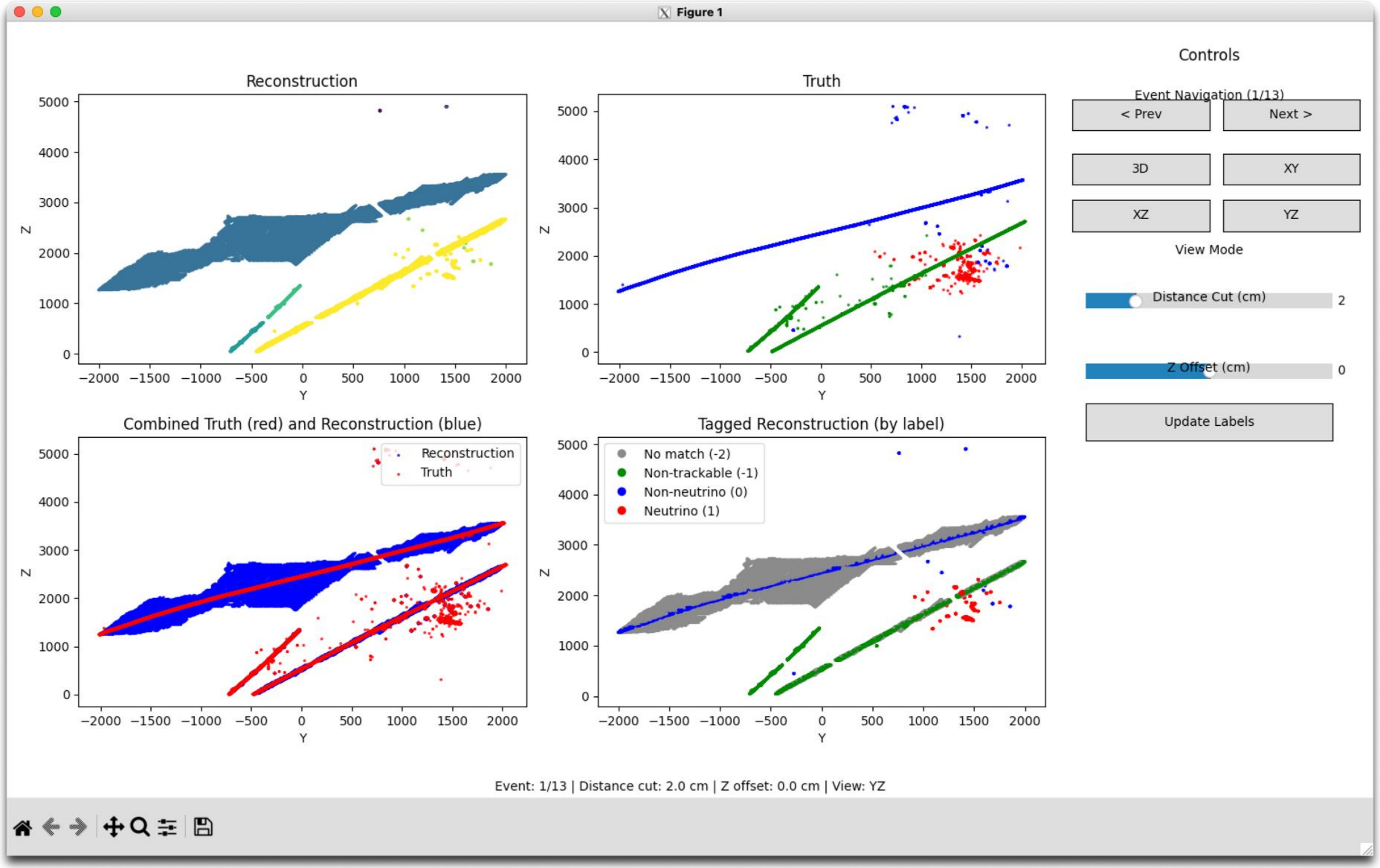
- Looking at current events, is that possible to do NCpi0 clustering?
  - what could be possibly learned?
  - can human do that? – Handscan 13 events, 7 have big nu activities
    - Two main features:
      - busy activity: short track (<10cm) + dots
      - associate dots to nu not long track
    - **Need APA stiching: 4 out of 7 events**

Brookhaven National Laboratory

hand scan example



truth tagging for isocronous



Brookhaven National Laboratory

Next

- Another training with correct z_offset – simple dist. based tagging
- Add blob_idx, clus_idx in the point feature for the training
- Add light barycenter to the feature
- Explore alternative tagging – all blob, cluster, etc
- More samples:
    - more NC
    - generic nu + cosmic
    - BNB + cosmic