Summary of tracking hands-on exercise

Jamie Antonelli

Setup

- exercise built on existing jupyter notebooks
- launched from caltech cluster, no user account creation needed
- authentification via github
- several notebooks with different approaches to same problem

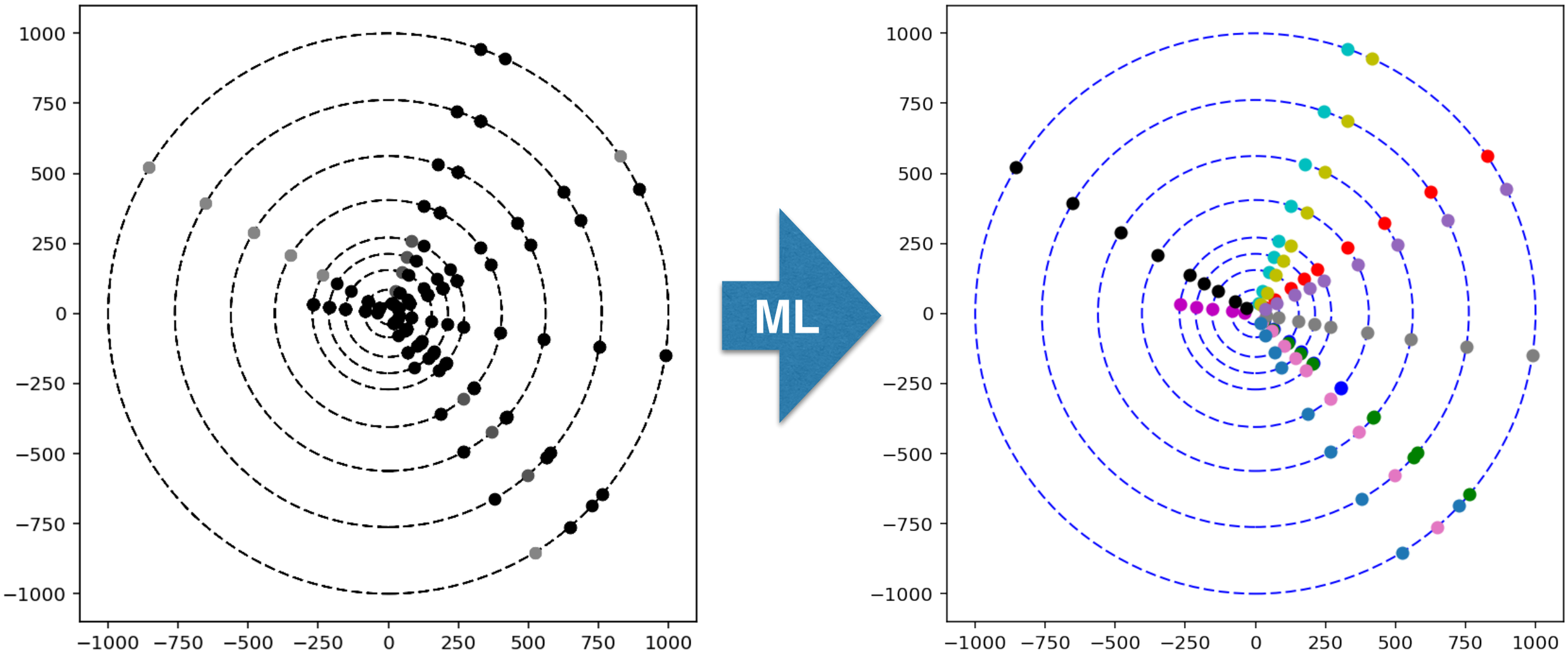
Feedback on structure

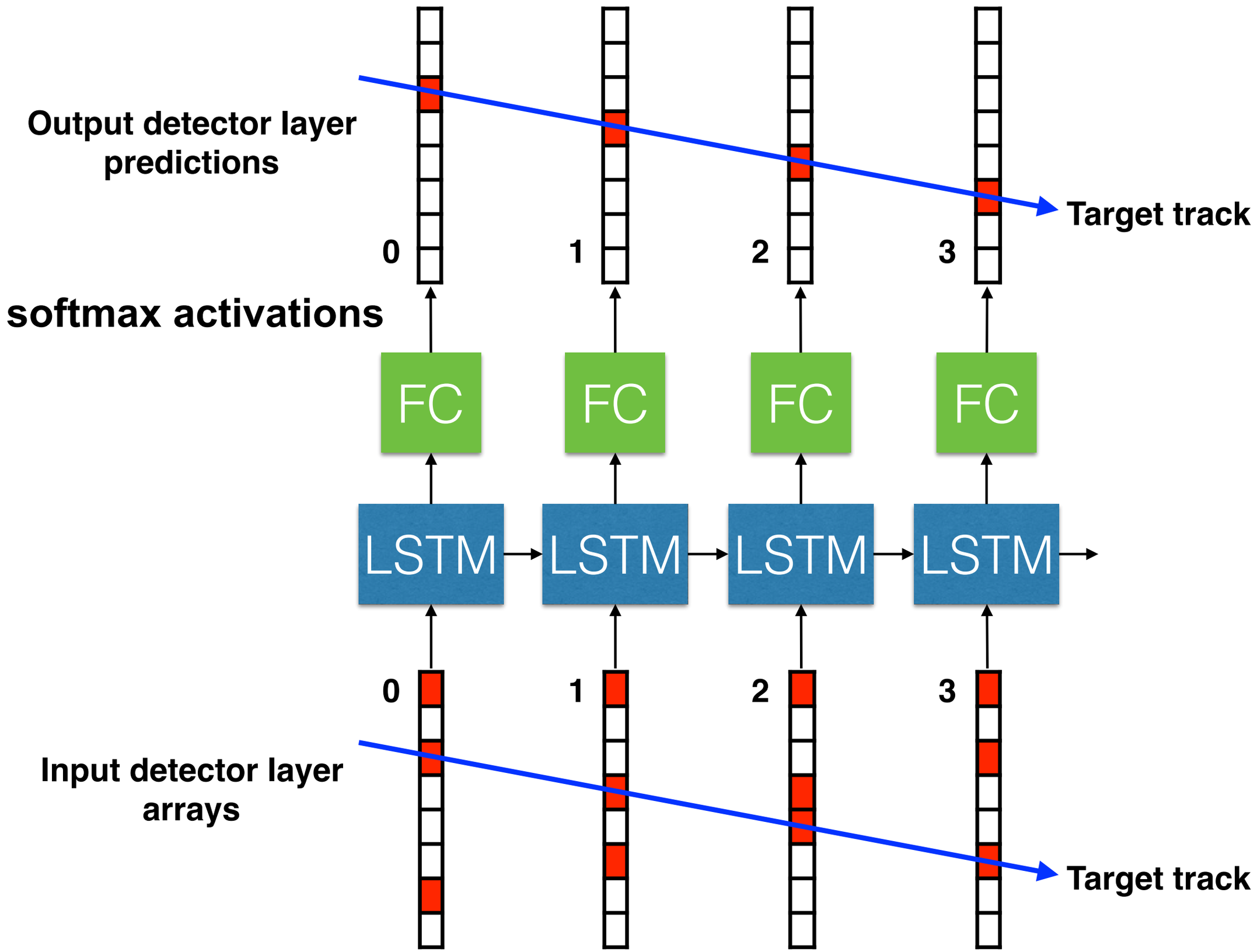
- group had wide range of experience with relevant tools (jupyter, python, keras, tracking, etc)
- some CMS DAS-style asynchronous "pre-exercises" might be useful to add
- insight into NN structures came through conversations with leaders
- more time would be needed to do something substantive beyond running the existing notebooks

lots of useful conversations

a boardful of useful conversations

problem: assign hits to tracks

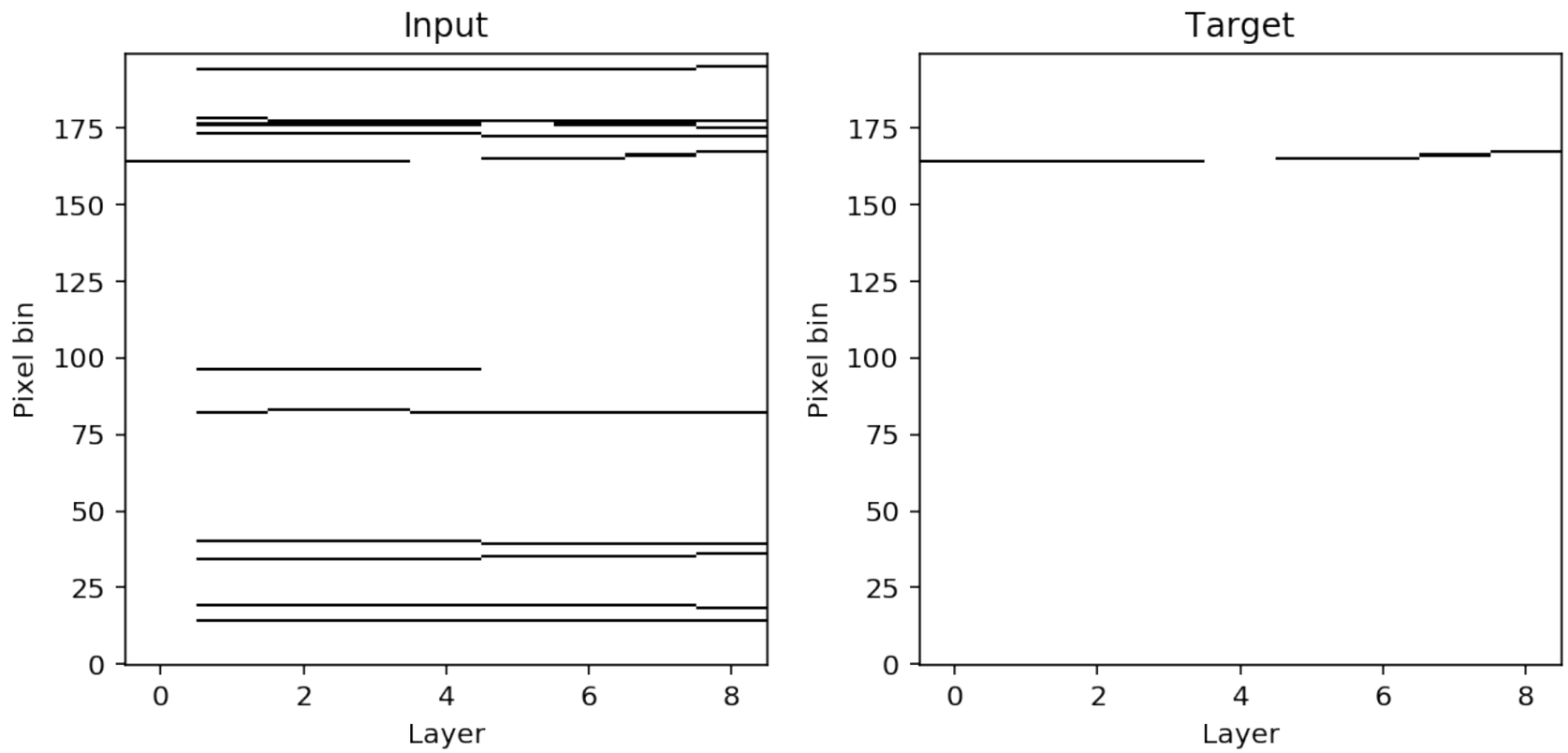


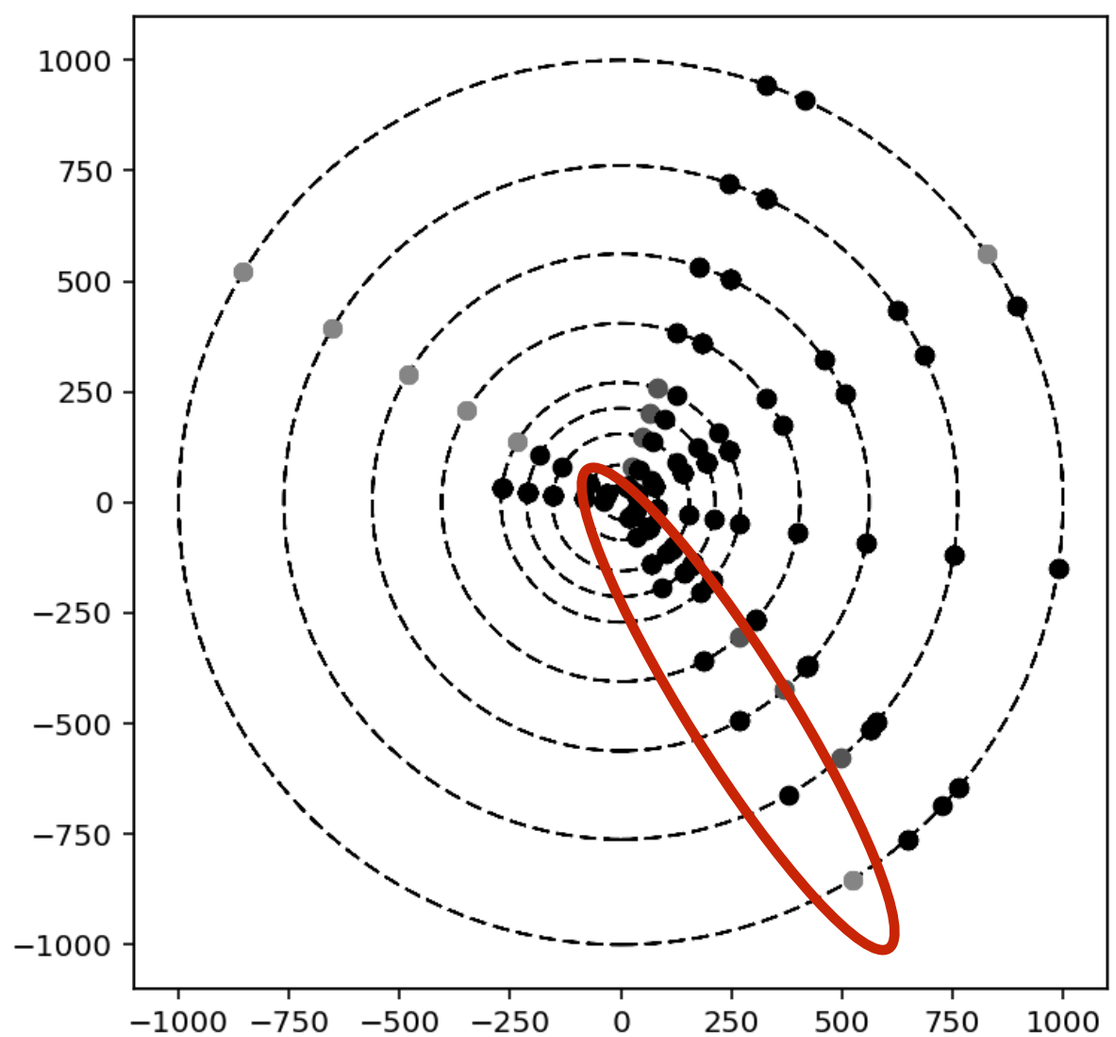

Output detector layer predictions
Target track
0
1
2
3
softmax activations
FC
FC
FC
FC
LSTM
LSTM
LSTM
LSTM
0
1
2
3
Input detector layer arrays
Target track

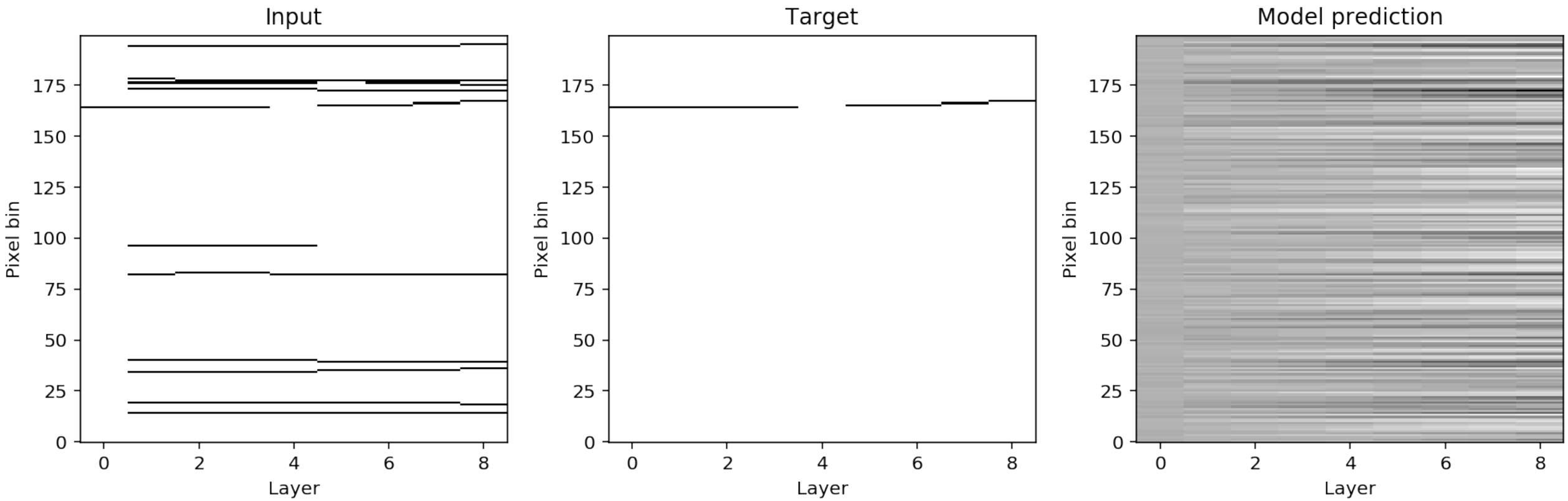
Where the magic happens

```
def build_model(num_hidden, length, dim,
                loss='categorical_crossentropy',
                optimizer='Nadam', metrics=['accuracy']):
    inputs = layers.Input(shape=(length, dim))
    hidden = layers.LSTM(output_dim=num_hidden, return_sequences=True)(inputs)
    outputs = layers.TimeDistributed(layers.Dense(dim, activation='softmax'))(hidden)
    model = models.Model(input=inputs, output=outputs)
    model.compile(loss=loss, optimizer=optimizer, metrics=metrics)
    return model
```

```
num_phi_bin=200
hidden_dim=50
batch_size=128
num_epoch=5
tracker = Clusterer(num_phi_bin=num_phi_bin, hidden_dim=hidden_dim,
                    batch_size=batch_size, num_epoch=num_epoch)
```

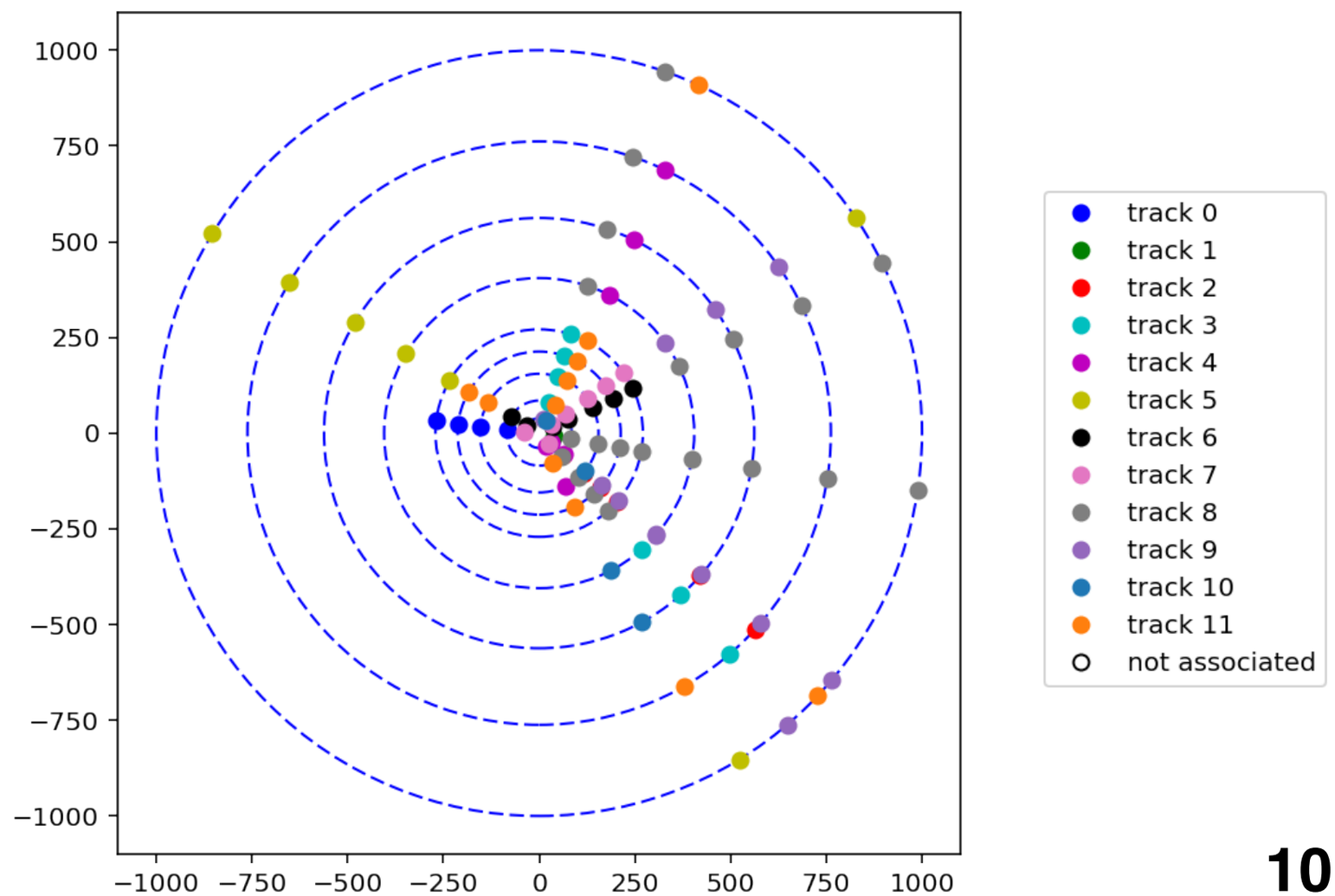

Input
Target
Pixel bin
Layer
0
25
50
75
100
125
150
175
0
2
4
6
8

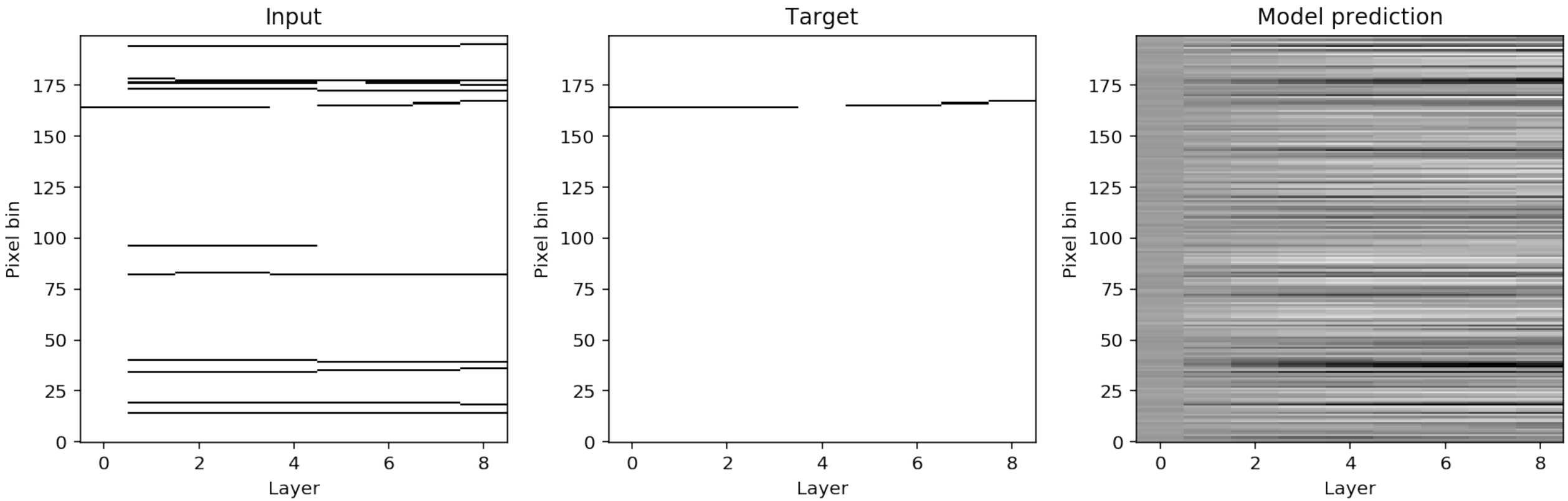

1000
750
500
250
0
−250
−500
−750
−1000
−1000
−750
−500
−250
0
250
500
750
1000



training set:

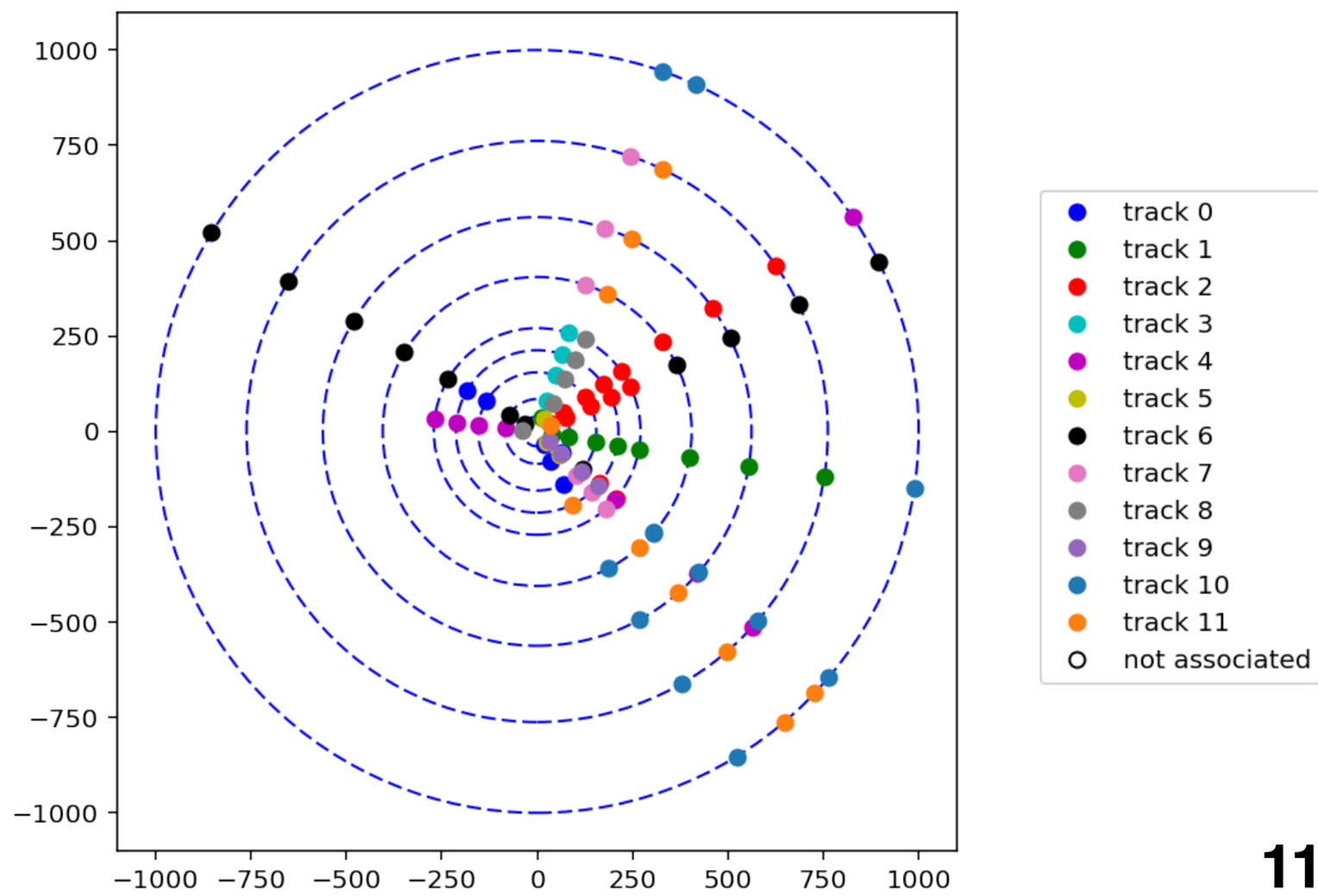
10 events

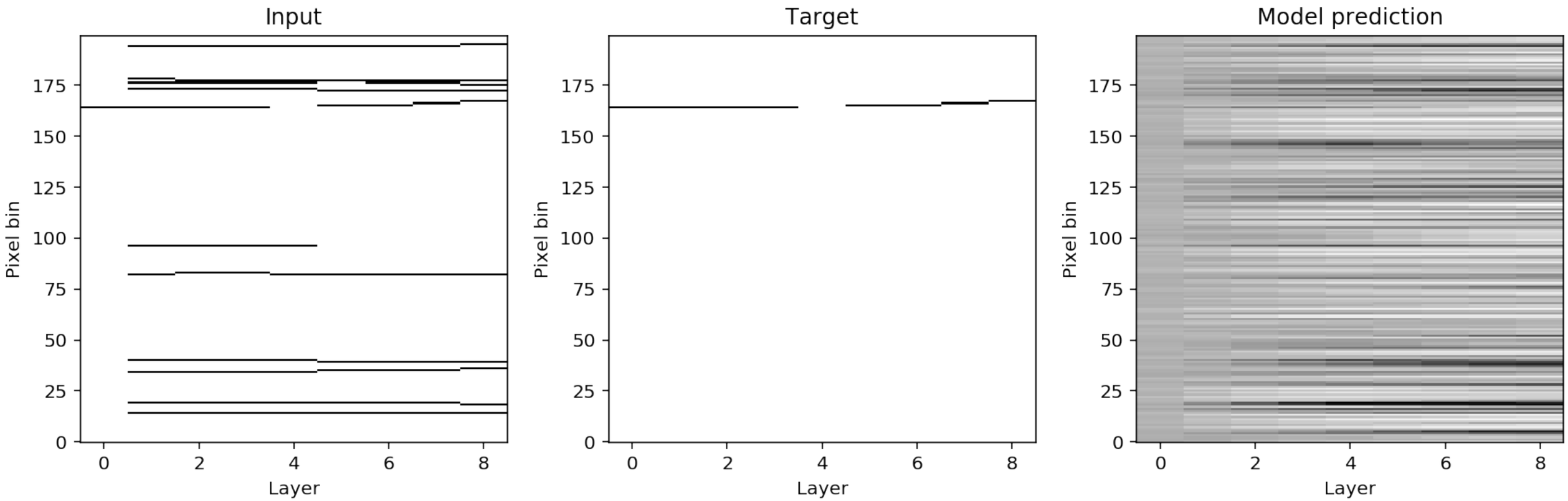




training set:

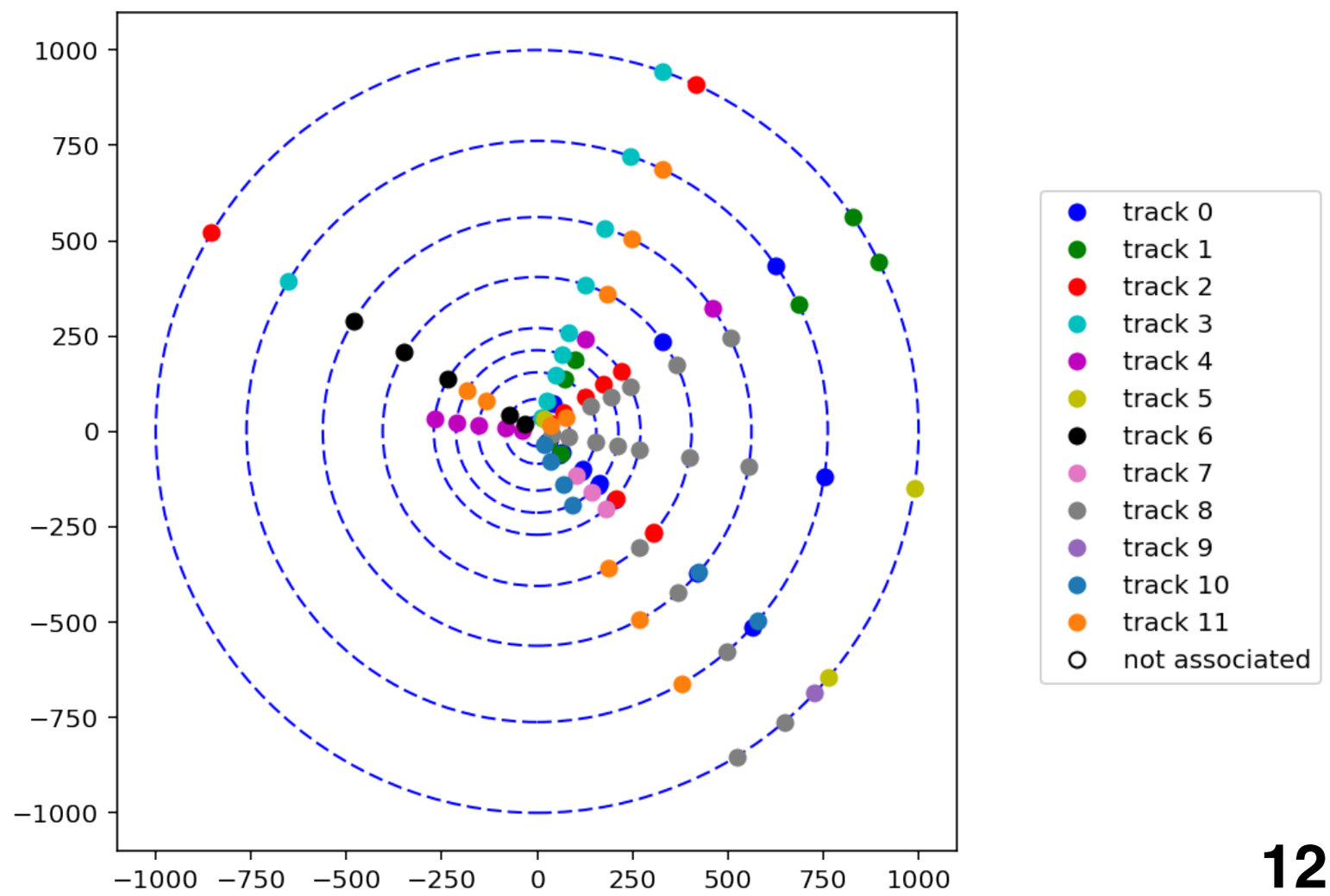
50 events

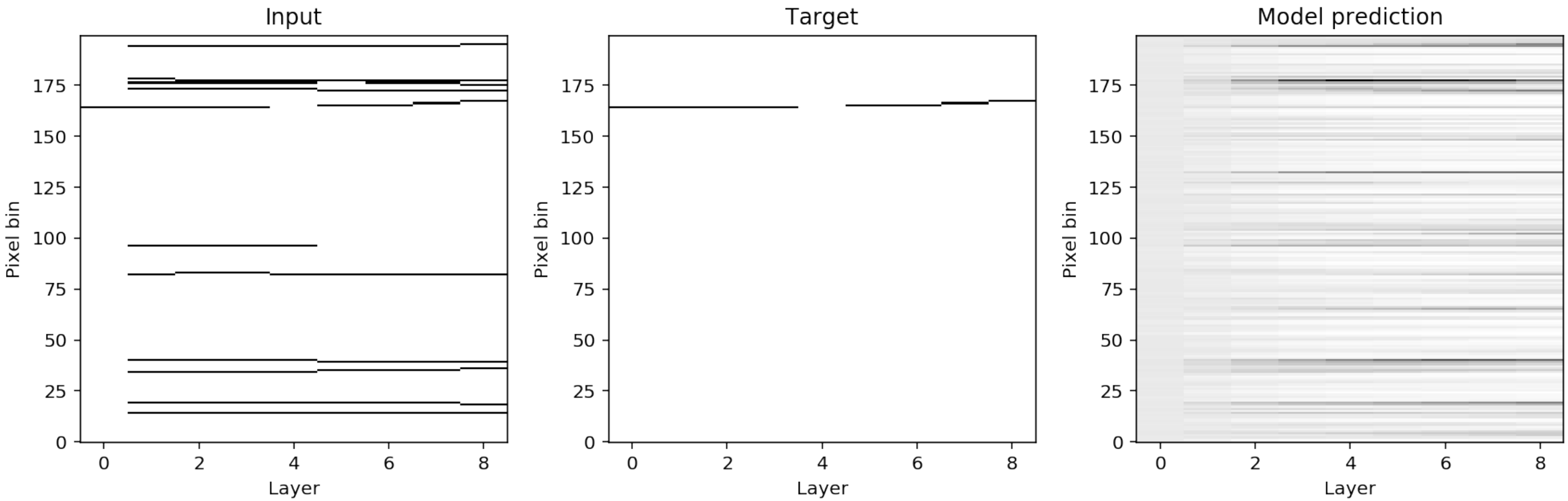




training set:

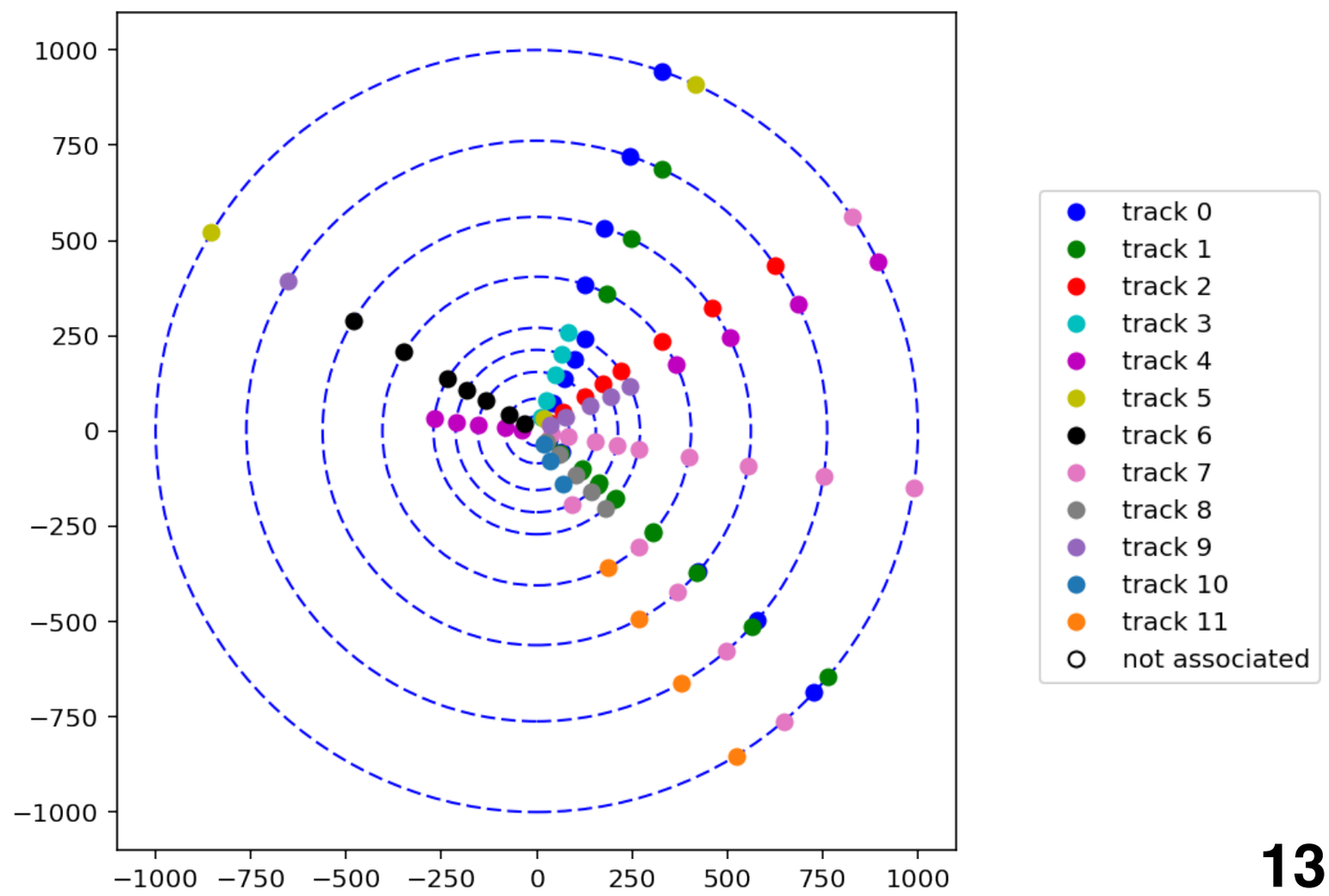
100 events

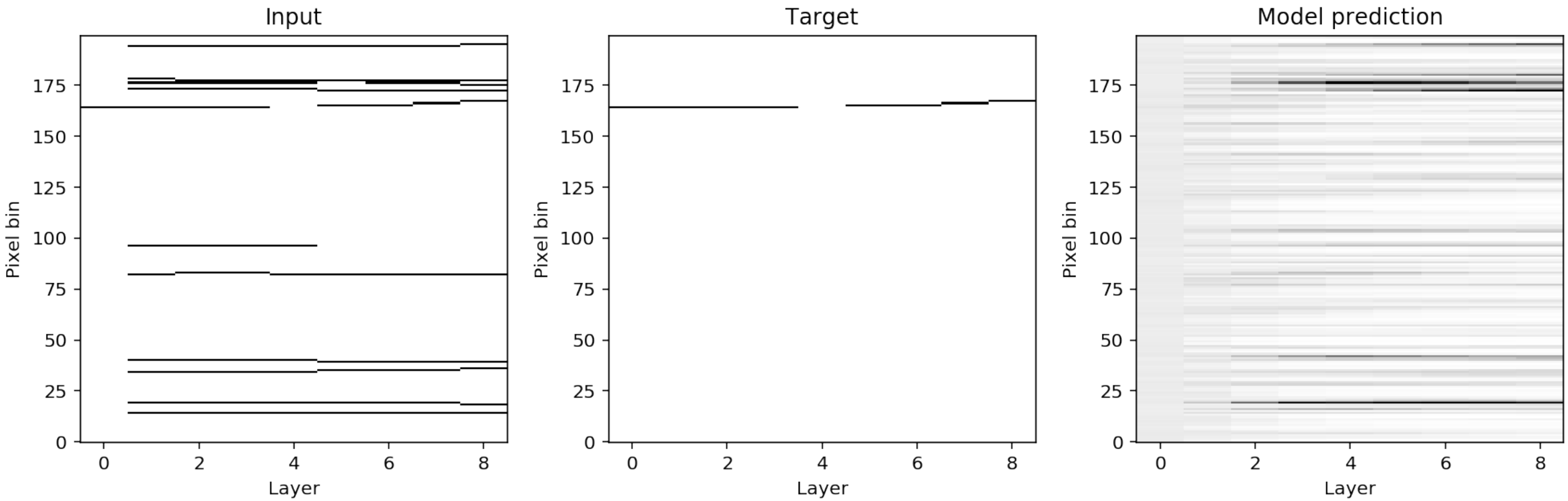




training set:

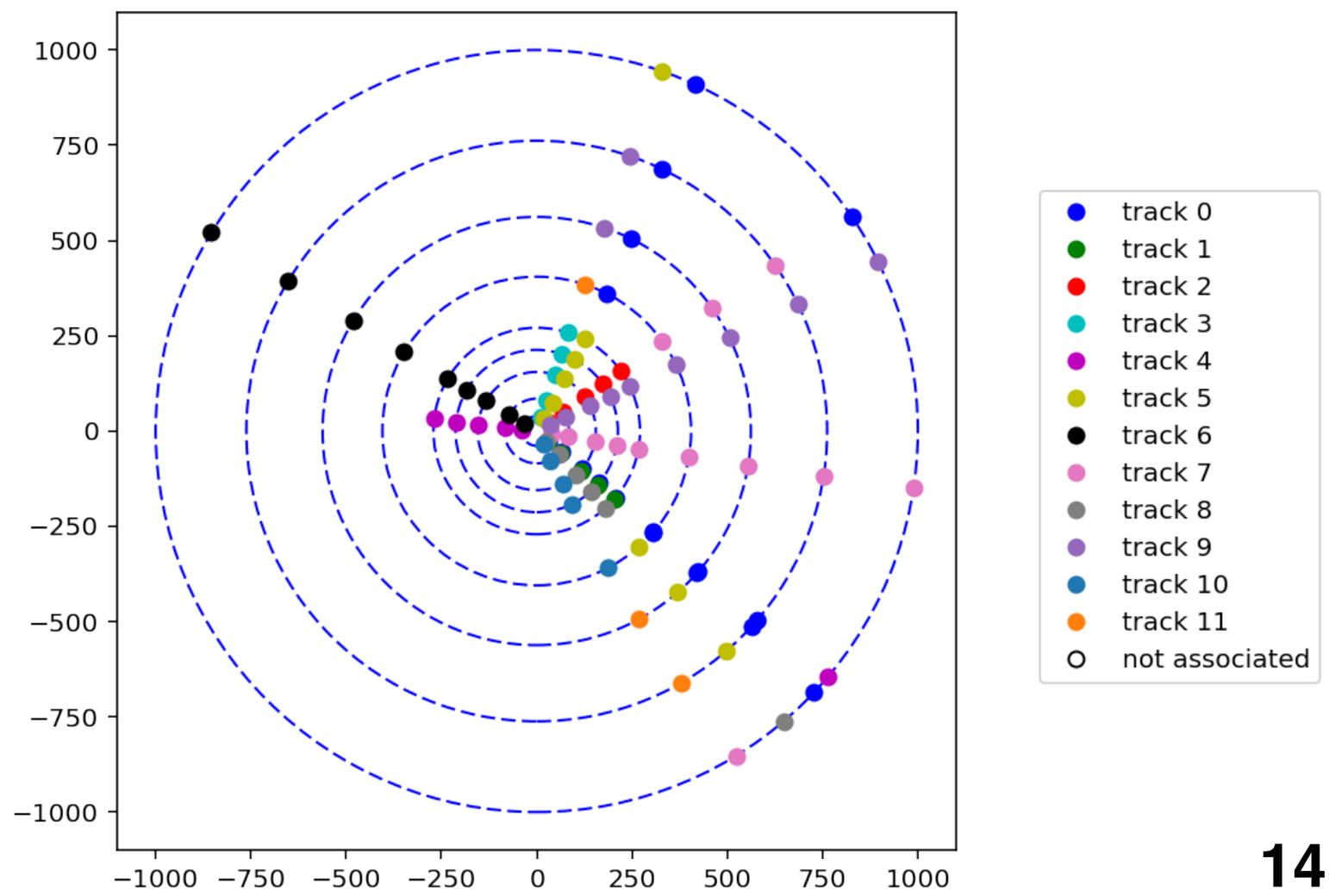
250 events

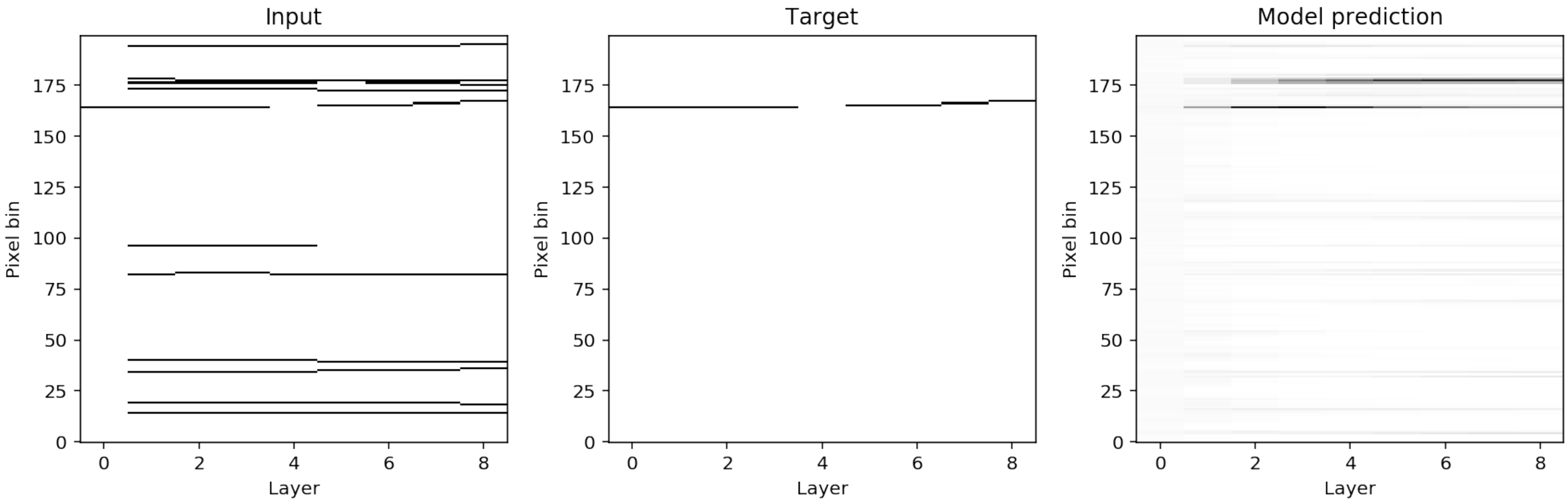




training set:

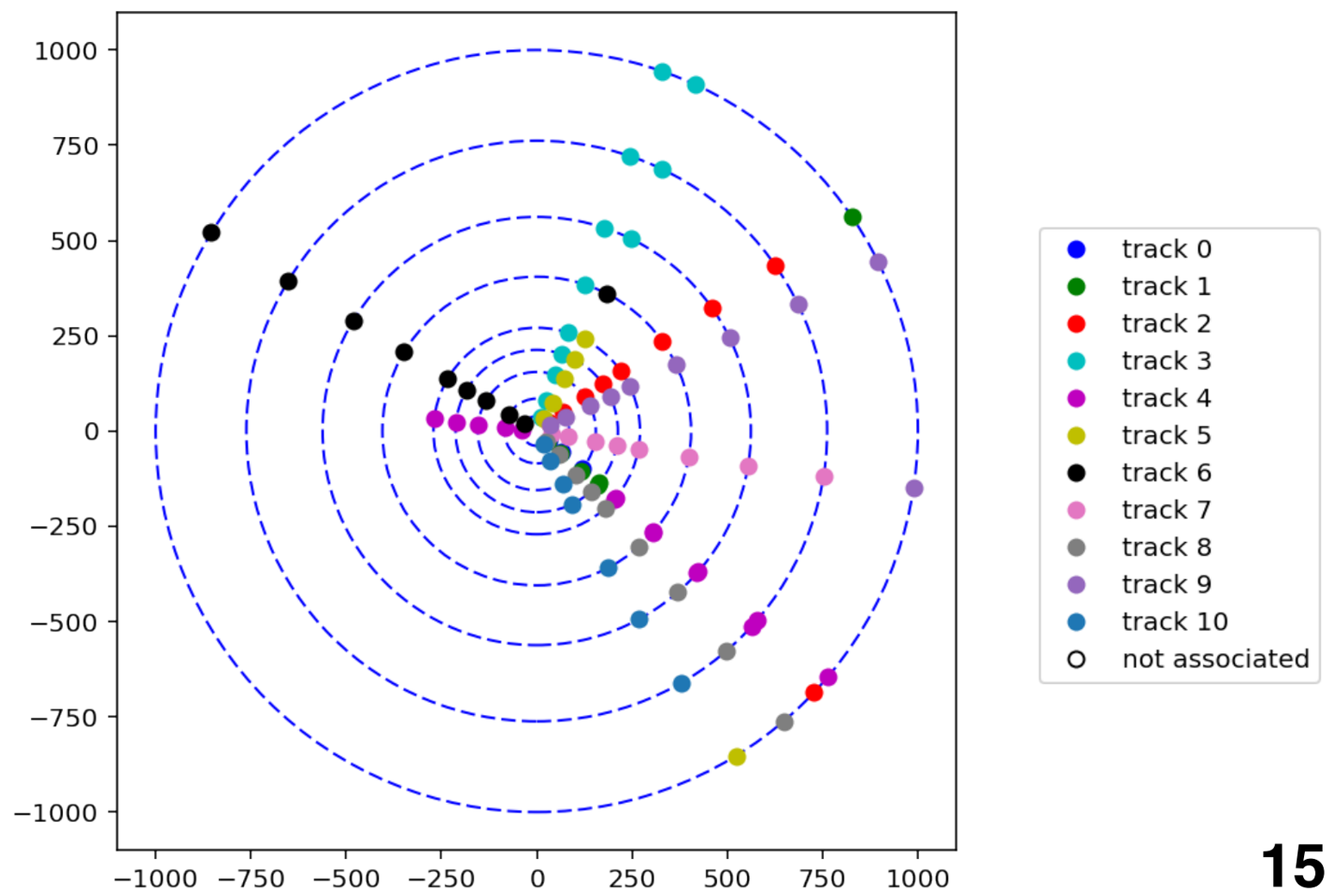
500 events

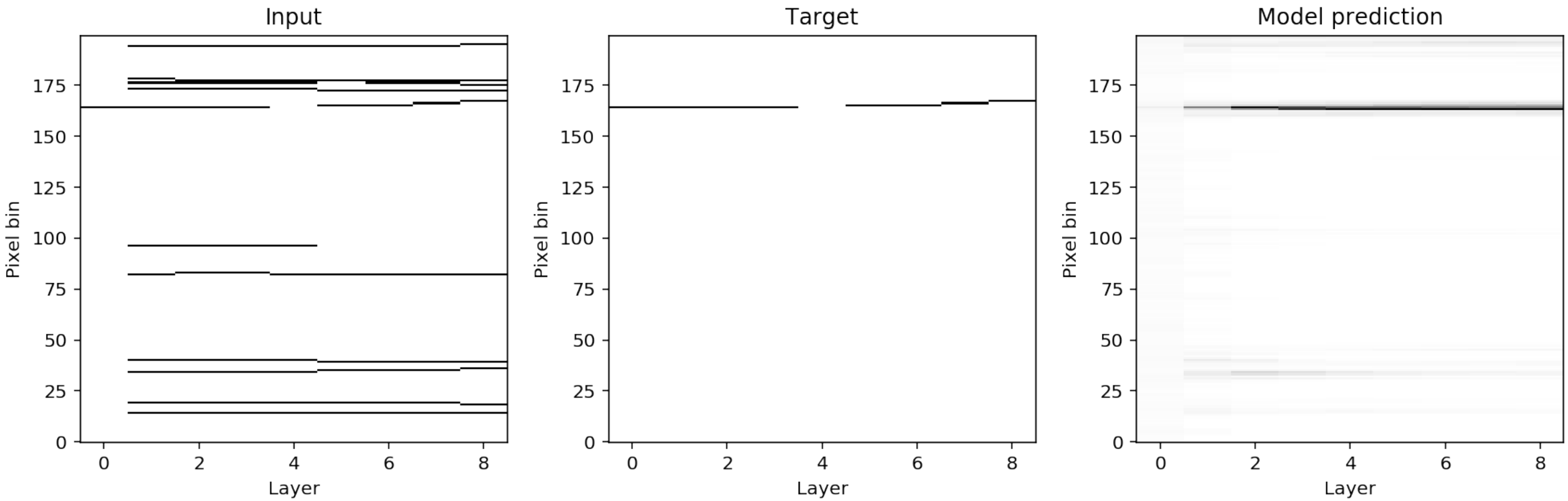




training set:

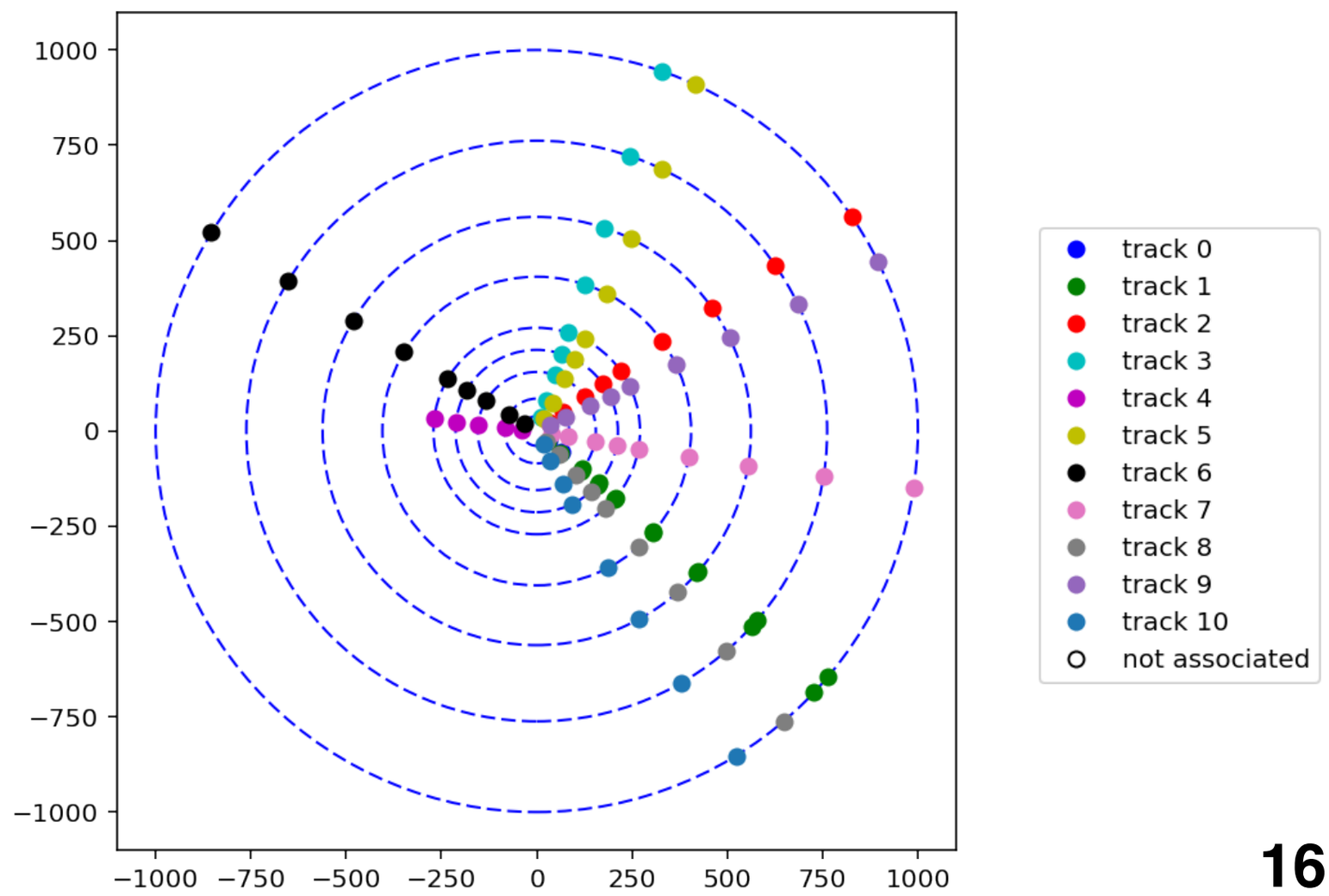
1000 events

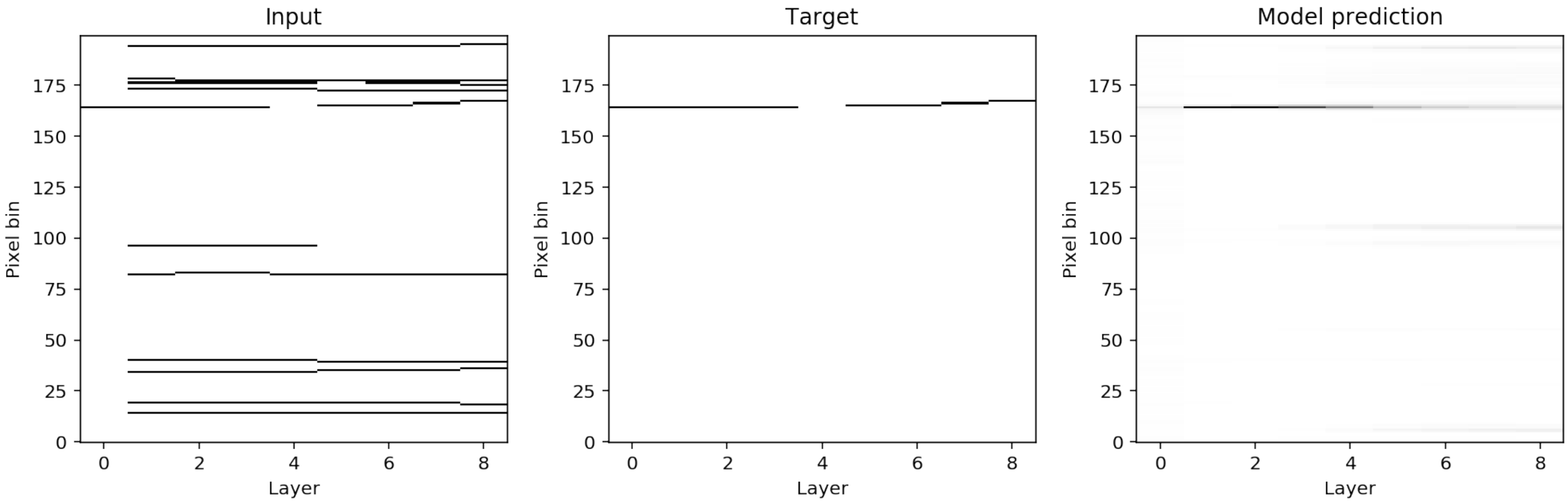




training set:

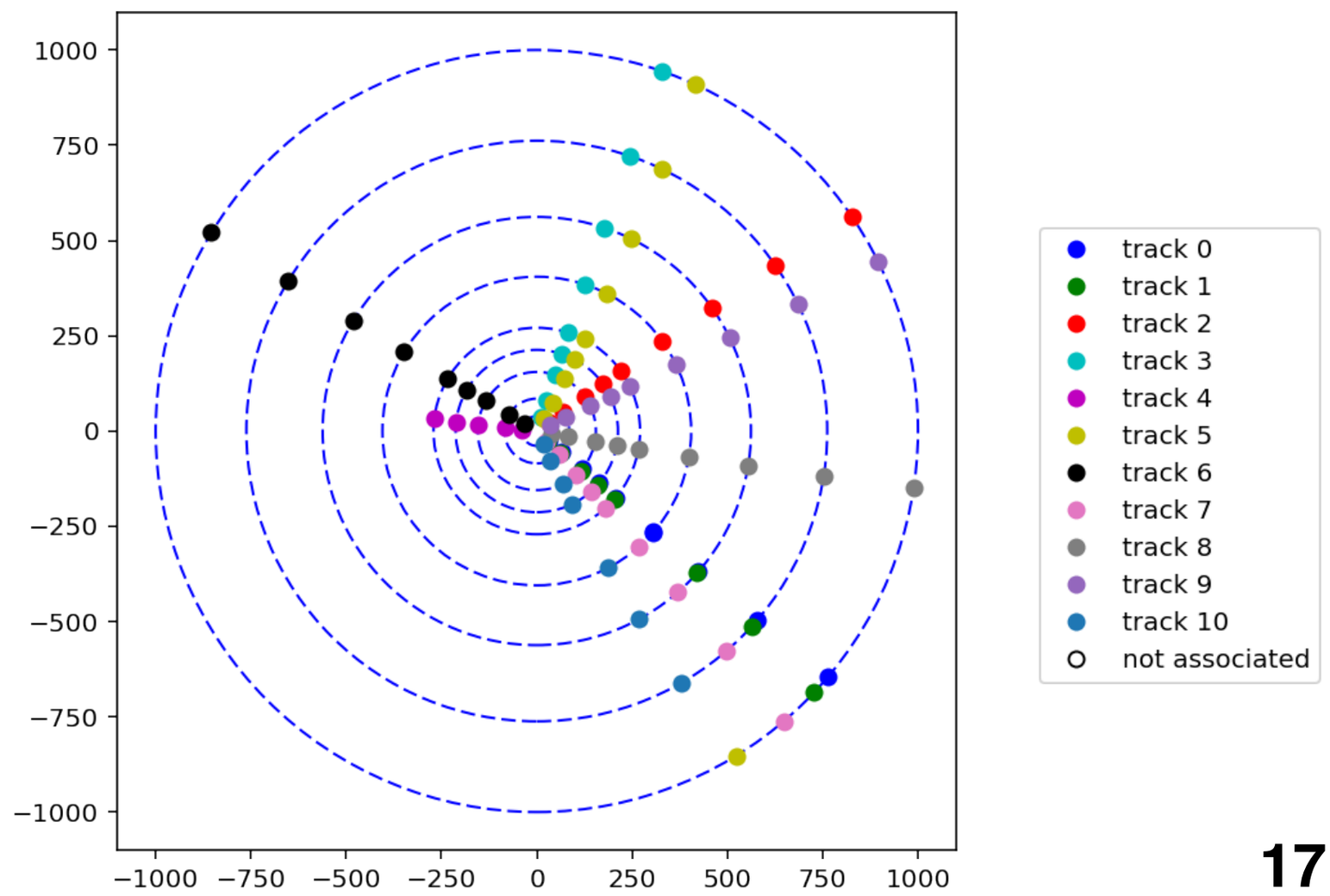
2000 events

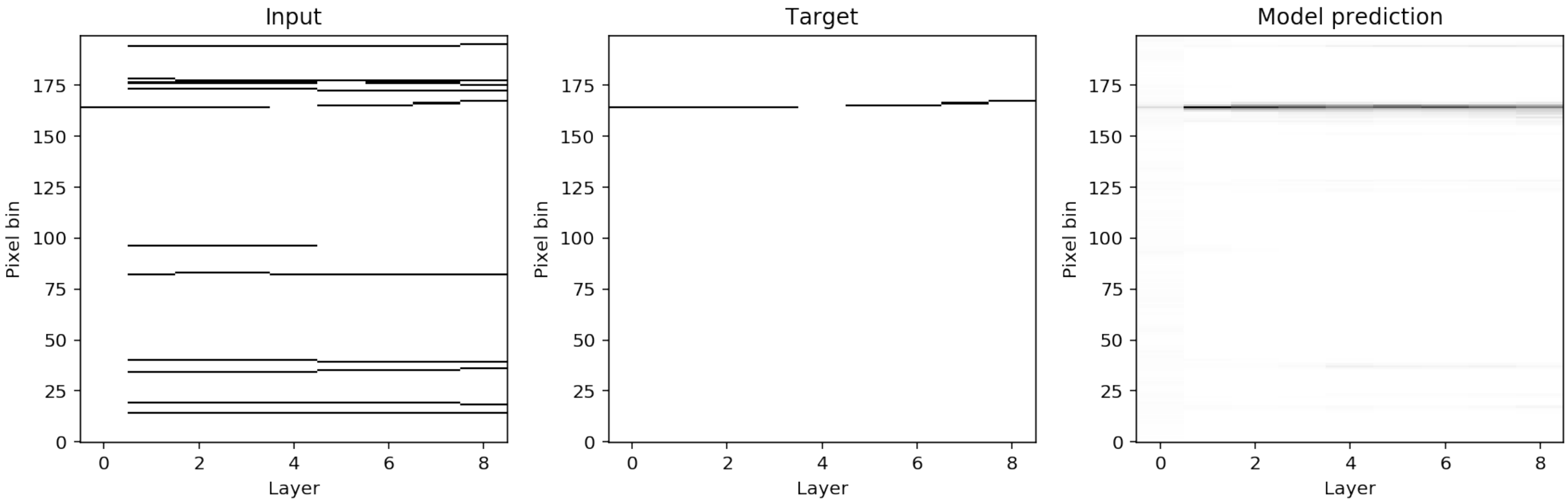




training set:

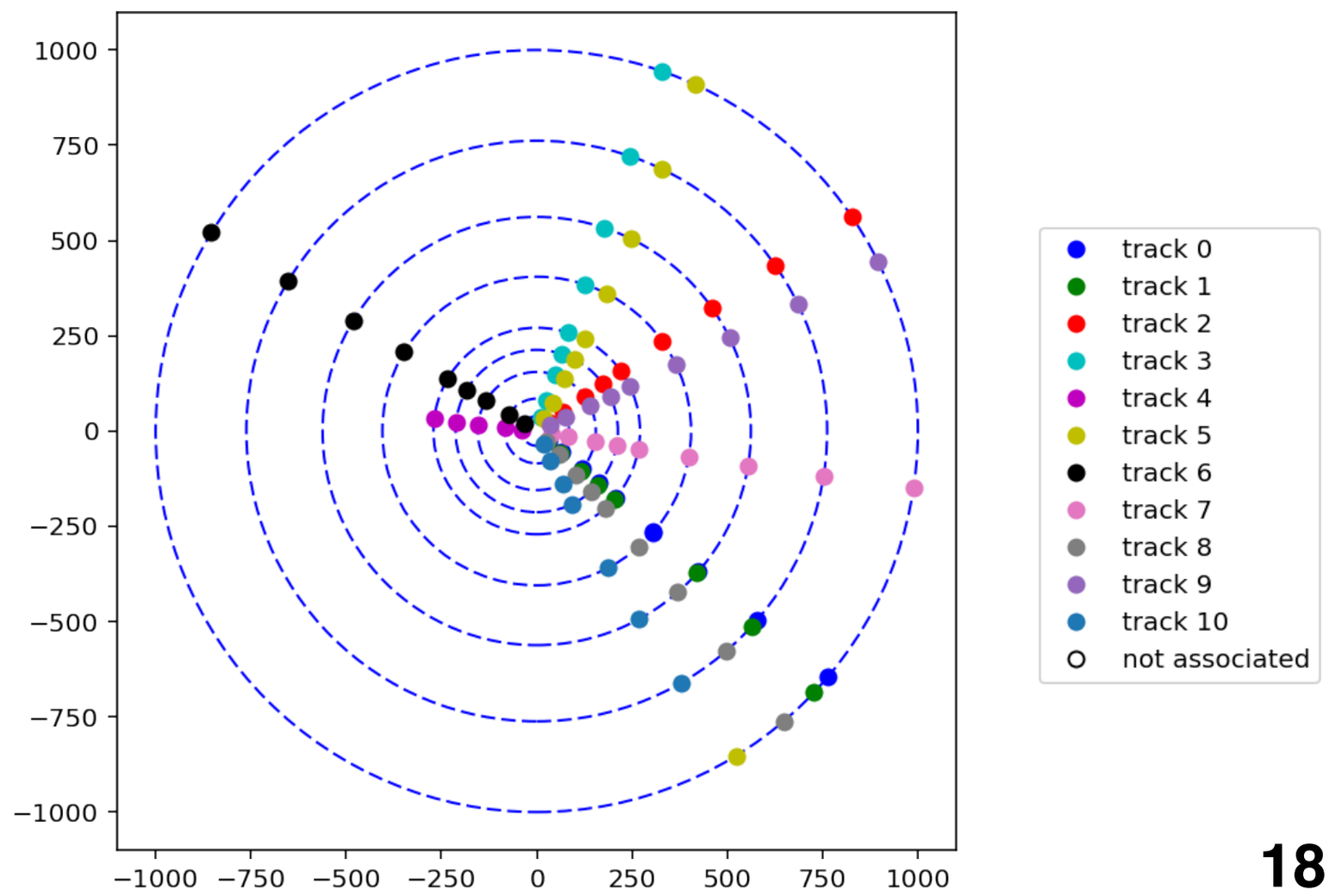
3000 events

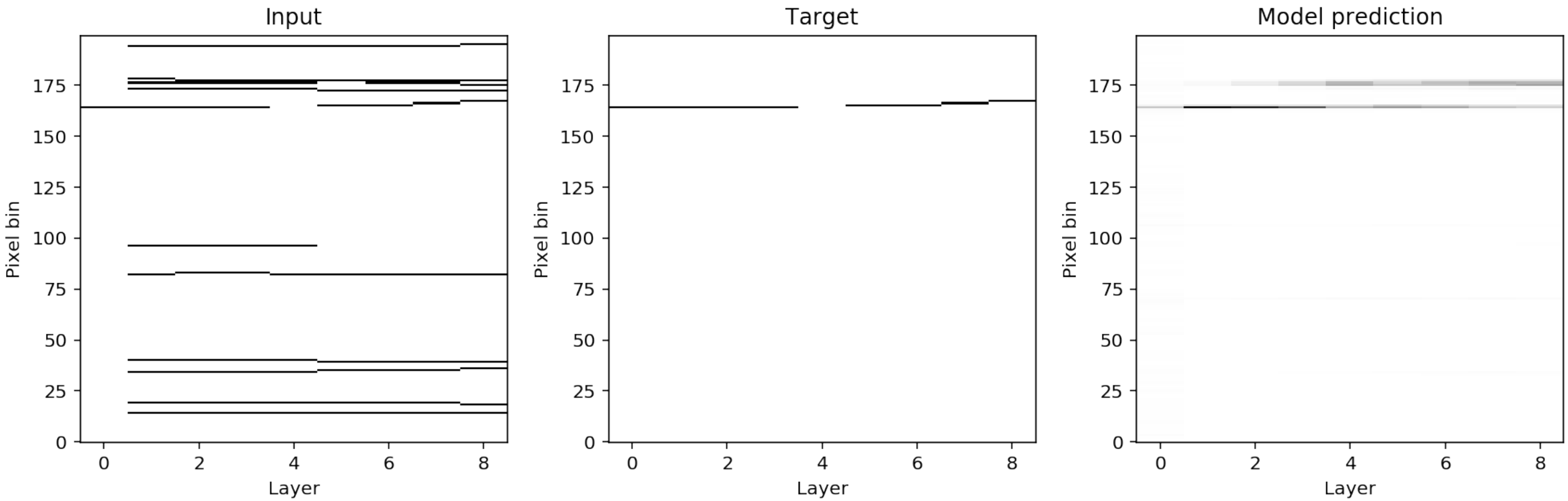




training set:

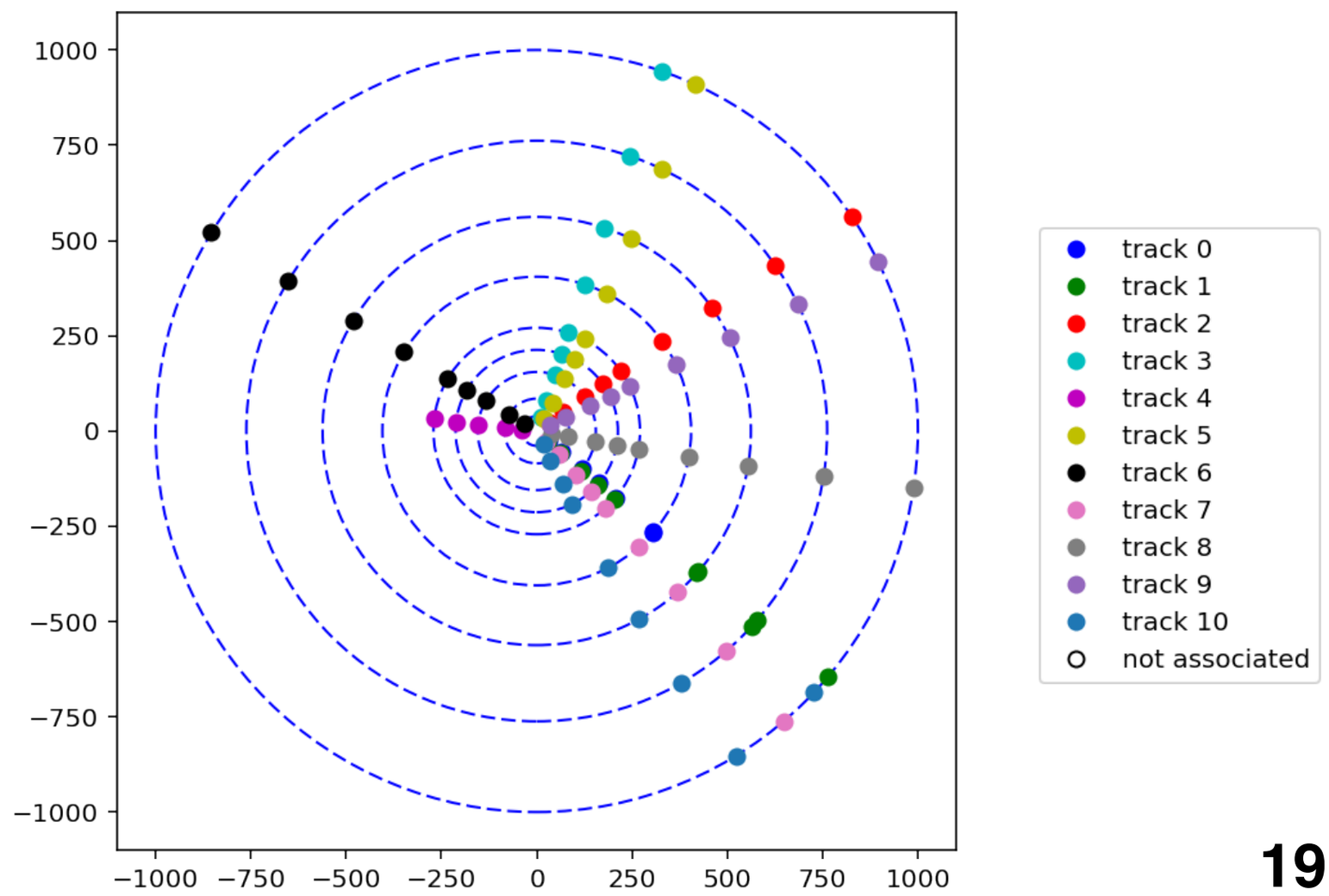
4000 events



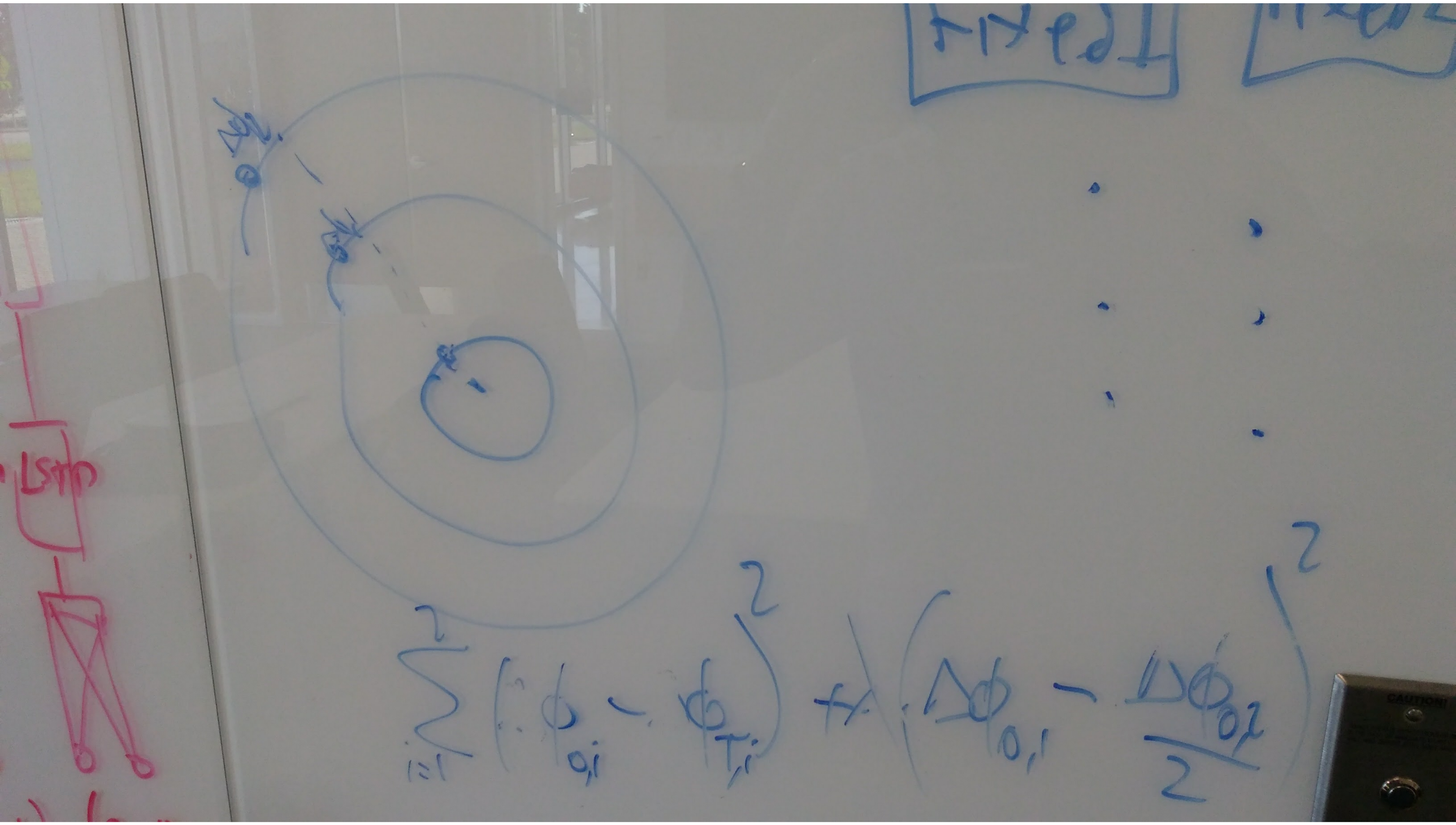


training set:

5000 events

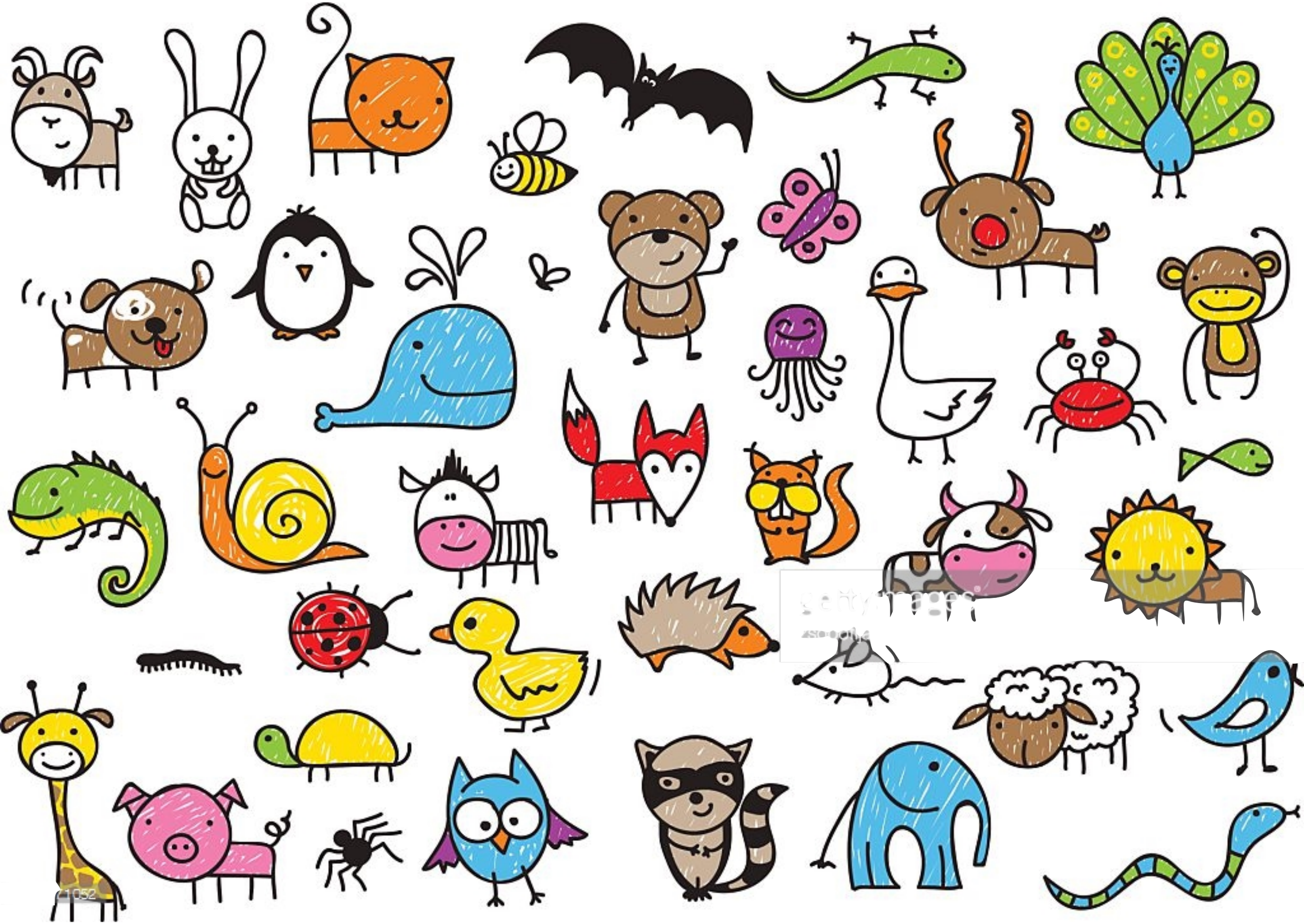


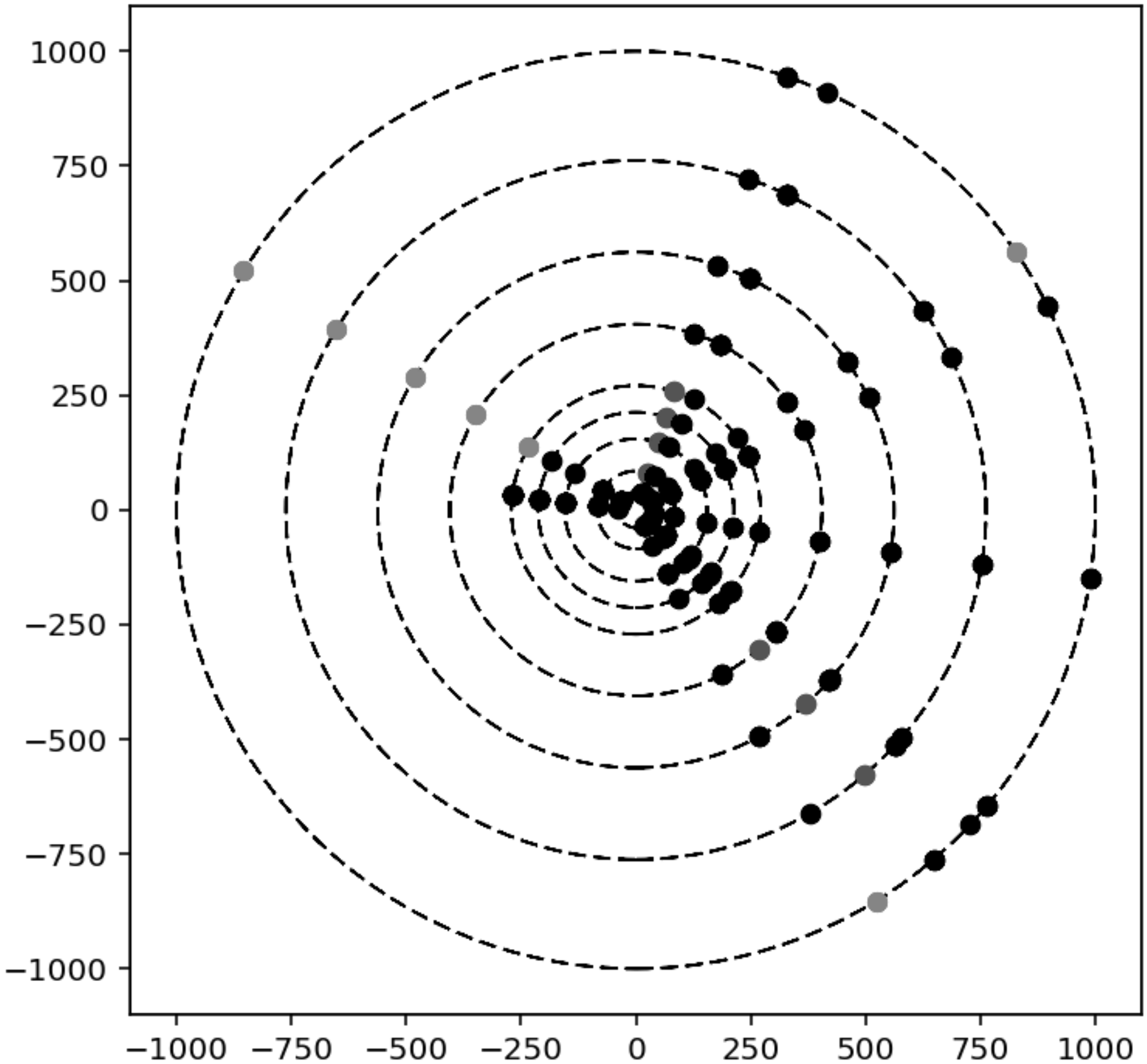
trying to add domain knowledge

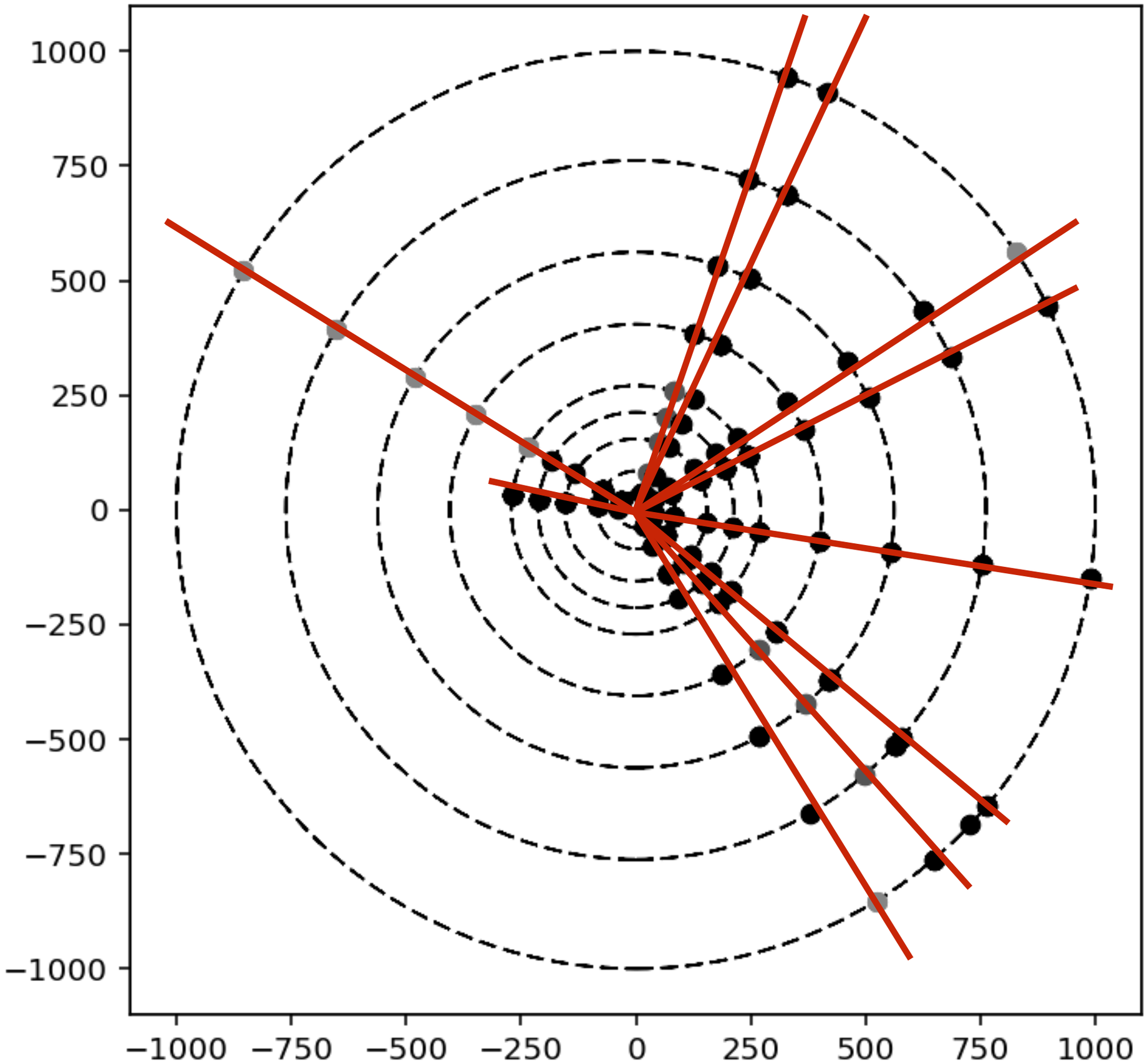


Op-ed: The ML black box

- I was surprised to realize that the algorithm wasn't told anything about tracking
- And how hard it was to ask it what it had learned in the training
- One can define a network in O(10) lines of code
- Understanding and improving the network to address observed failure modes seems like the real challenge




1000
750
500
250
0
−250
−500
−750
−1000
−1000
−750
−500
−250
0
250
500
750
1000


1000
750
500
250
0
−250
−500
−750
−1000
−1000
−750
−500
−250
0
250
500
750
1000

Takeaway

- I came away excited to gain more familiarity with the tools
- And to understand what makes a problem a good candidate for ML-based approaches
- I'm also a bit leery of handing over the reigns to our future robot overlords