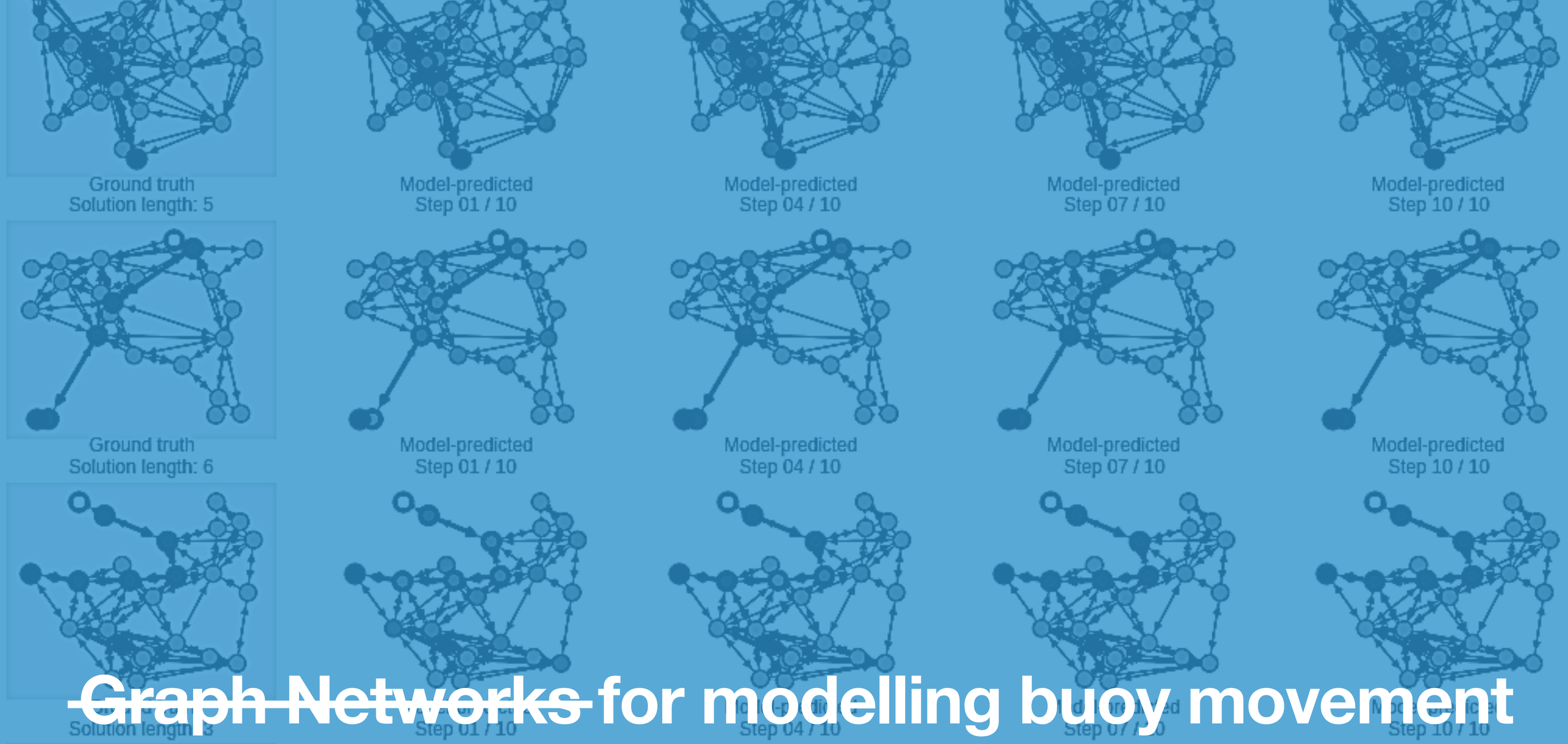
Predicting Sea Ice Movement

Graphice

Petra Ivascu & Raul Incze



~~Graph Networks~~ for modelling buoy movement

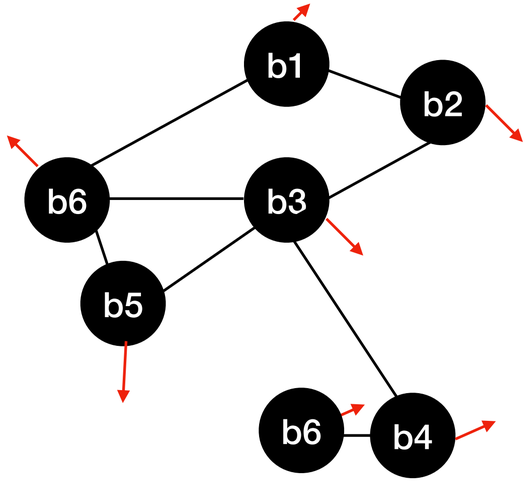
~~GNNs~~

Just some simple dense layers…

Initial idea: Graph Neural Networks

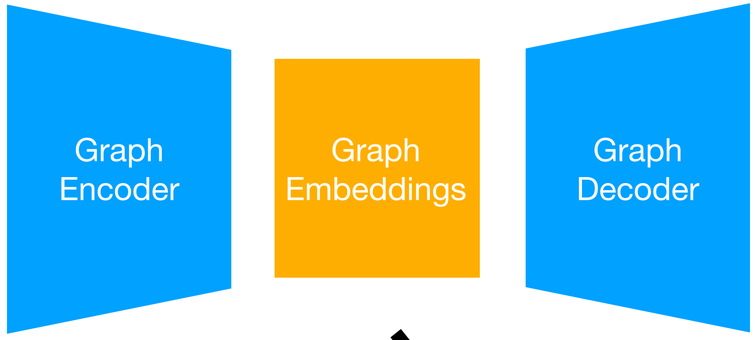
The glacier is a material and buoys are simulated particle

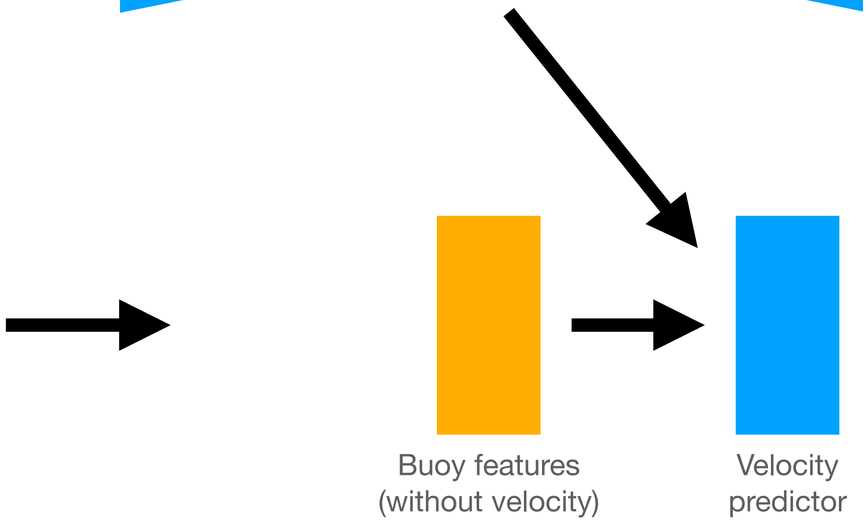
Buoy graph at t

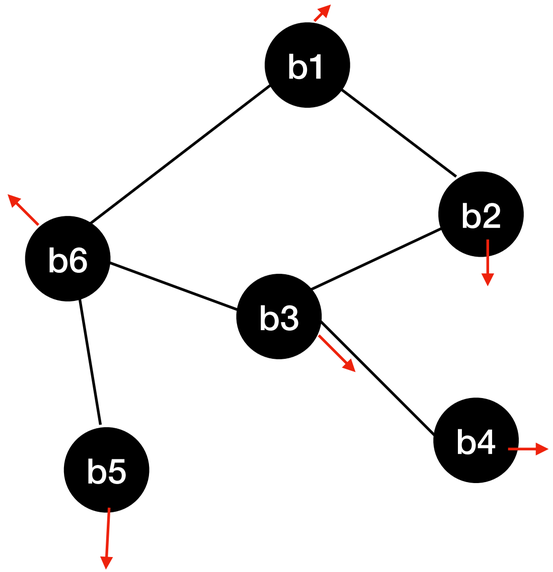


Nodes: buoy features
Edges: buoy distances

Randomly chose a node and pretend we don't know it's velocity







Predict buoy graph at t+1

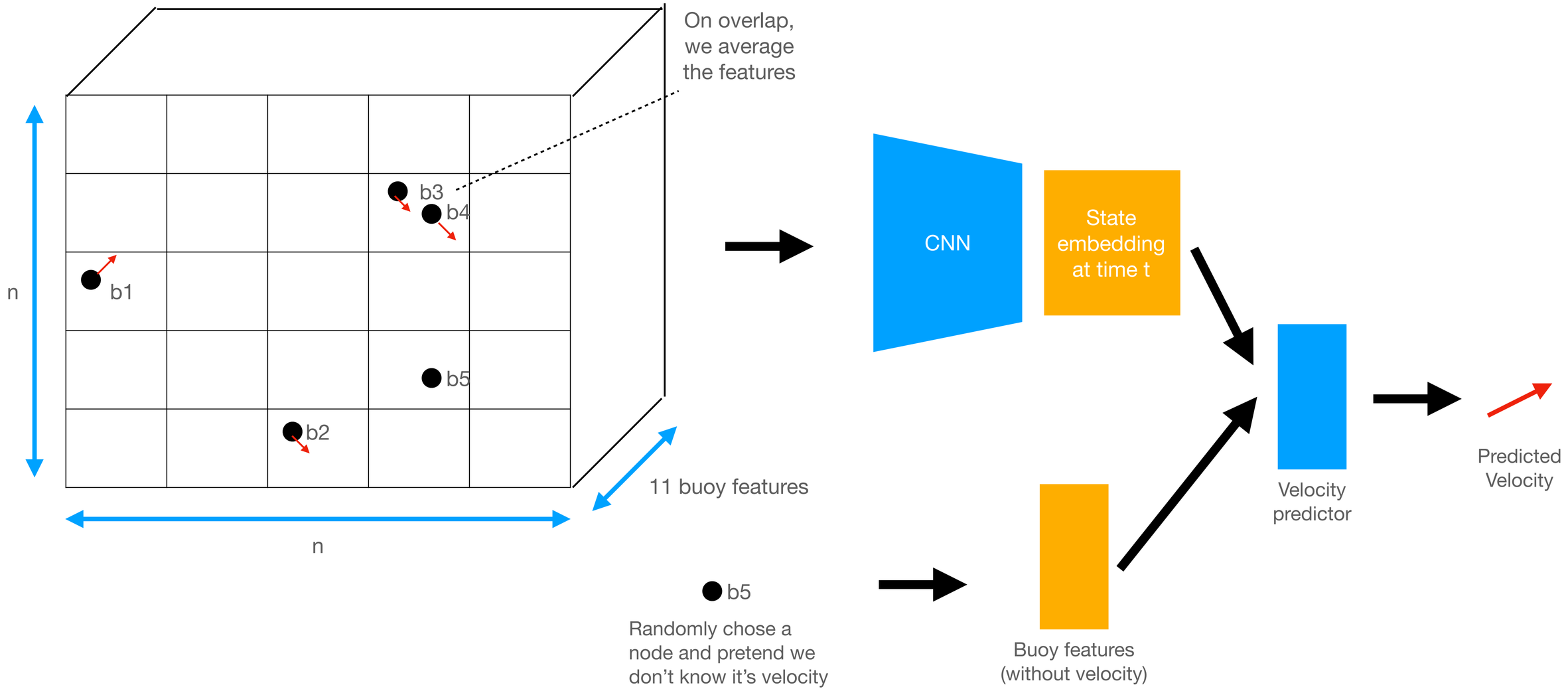
Predicted Velocity

For the graph representation to work, we'd need a constant set of nodes across multiple time steps, which doesn't really happen :(

Back to the drawing board…

Take 2: Convolutional Networks for Embedding States

Snap buoys on a grid, consider their features channels and feed them through a CNN

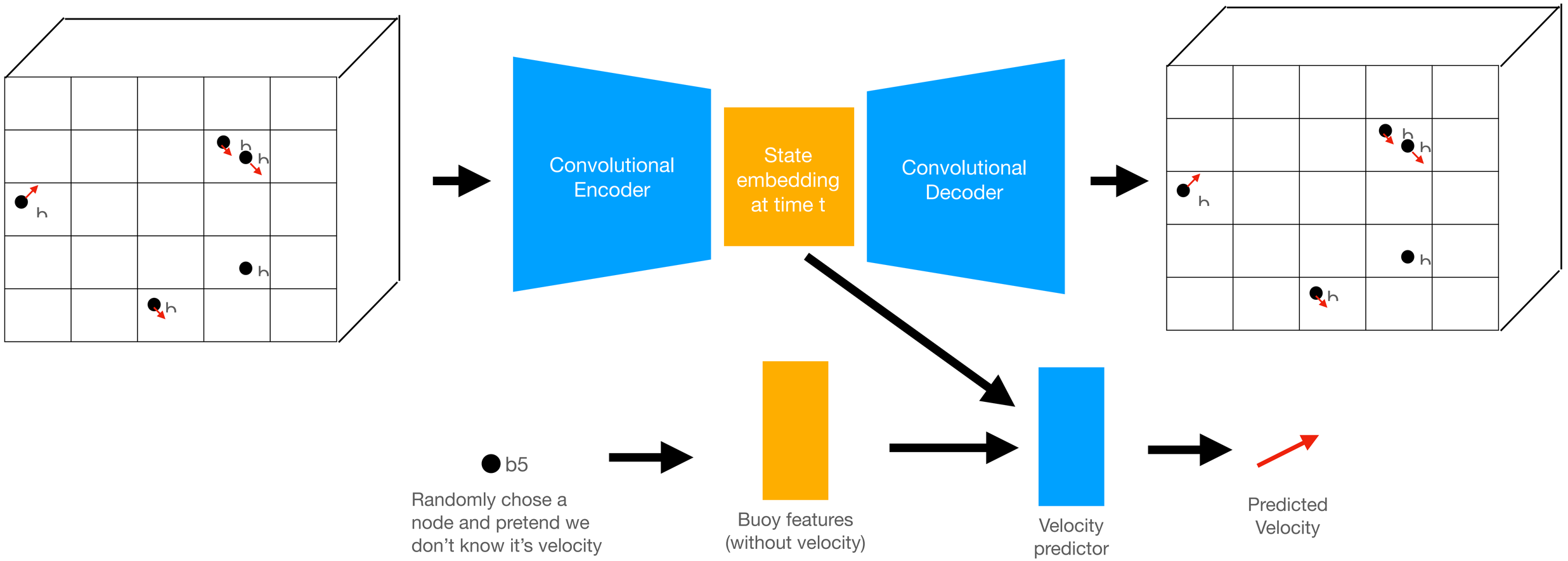


Performance is not great, let’s make sure the embeddings learned are meaningful!

Back to the drawing board…

Take 2.5: Convolutional Autoencoders

Let's try to reconstruct the initial state from the embeddings!

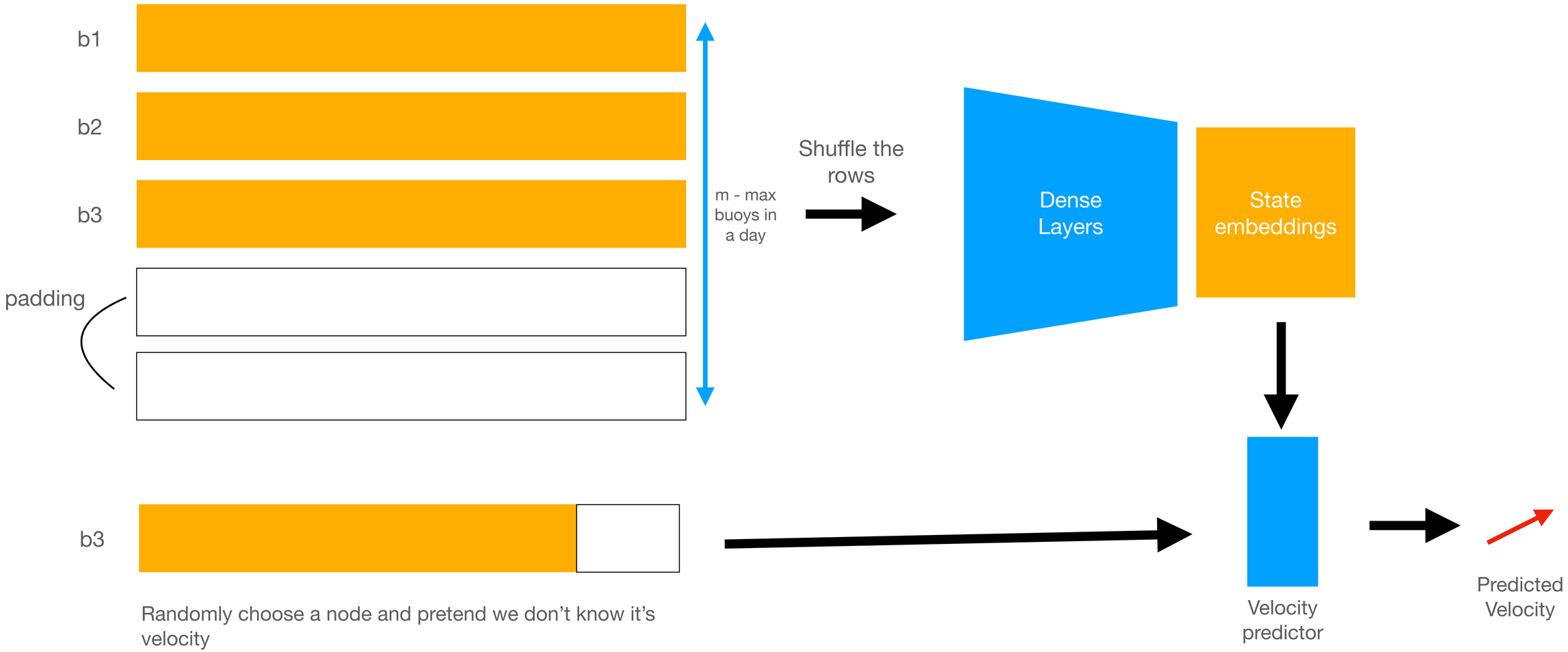


More than half of the days recorded have buoys that would overlap in the grid space…

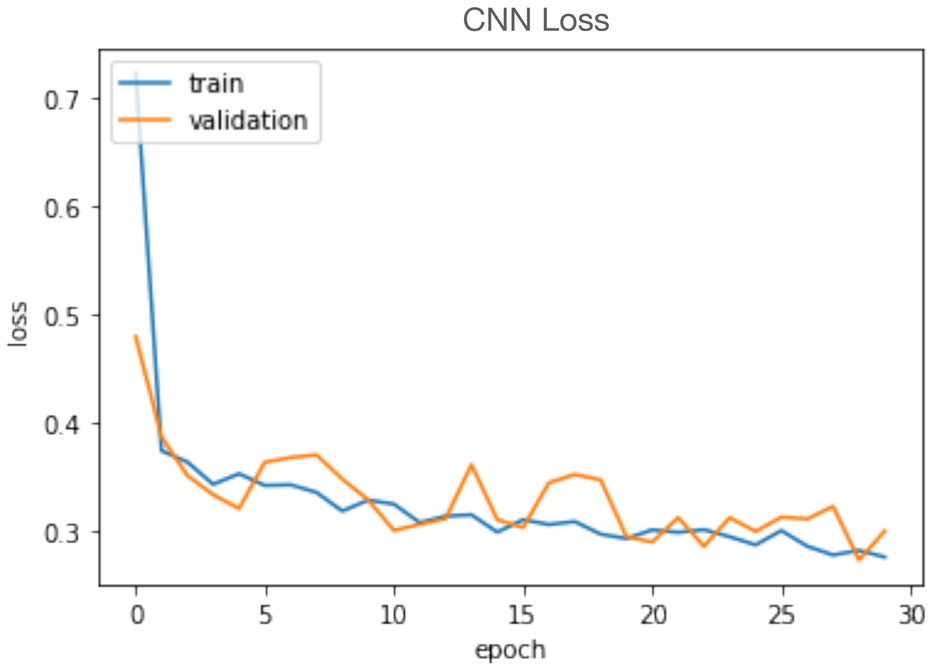
Maybe there is another way to feed the data into a neural network? Back to the drawing board again…

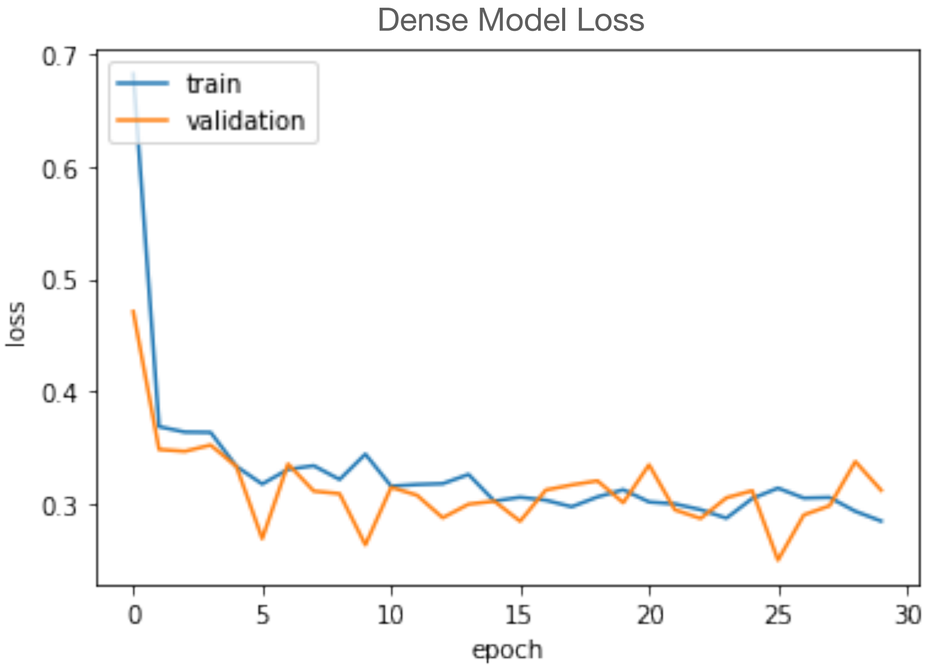
Third Take: A simpler approach

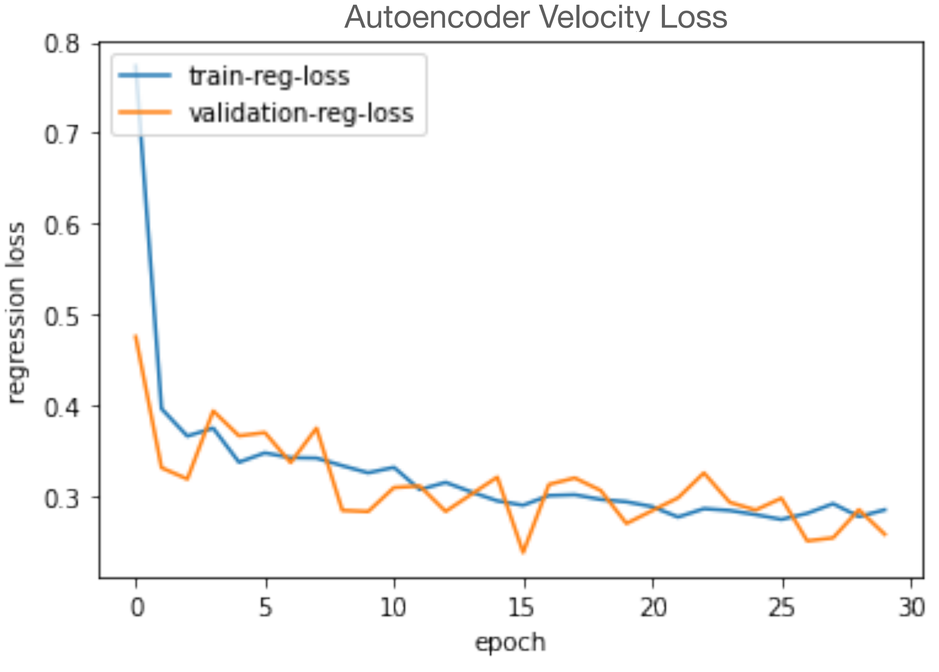
Simple dense, permutation-invariant dense layer to encode the state.

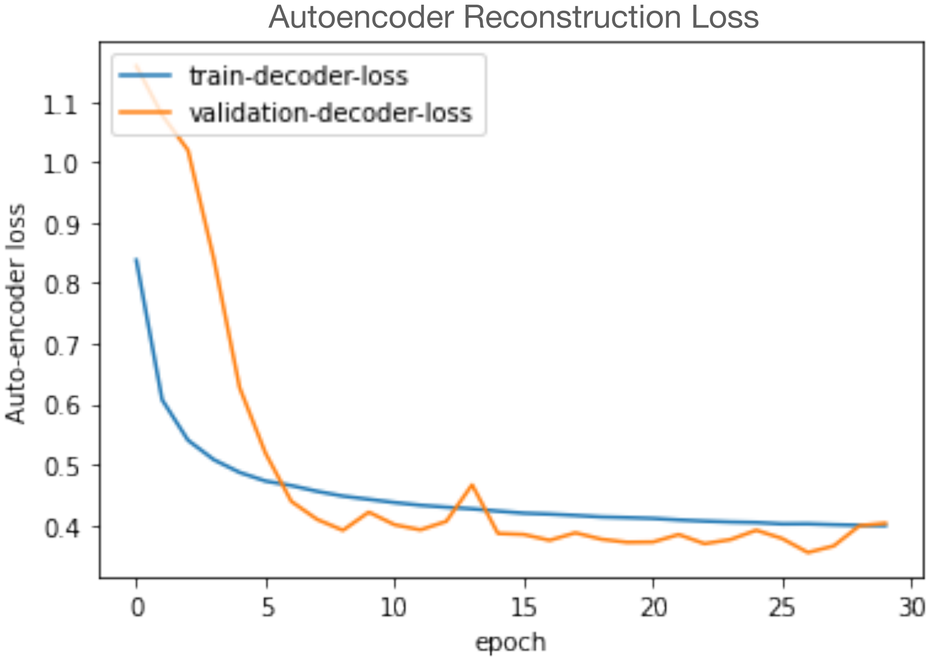


Results


CNN Loss
train
validation
loss
epoch


Dense Model Loss
train
validation
loss
epoch


Autoencoder Velocity Loss
train-reg-loss
validation-reg-loss
regression loss
epoch


Autoencoder Reconstruction Loss
train-decoder-loss
validation-decoder-loss
Auto-encoder loss
epoch

Ideas we didn’t have time for