

Predicting Sea Ice Movement

Graphice

Petra Ivascu & Raul Incze

The background of the slide is a grid of 20 graph network visualizations. Each visualization shows a set of nodes (circles) connected by directed edges (arrows). The nodes are arranged in a somewhat circular pattern, with edges connecting them in a complex, web-like structure. The edges are colored in shades of blue and green. The graphs are arranged in a 4x5 grid. The first column contains four 'Ground truth' graphs with solution lengths of 5, 6, 7, and 8. The subsequent columns contain 'Model-predicted' graphs for steps 01, 04, 07, and 10 of a 10-step process. The graphs show the progression of a solution over time, with the model's predictions becoming increasingly accurate as the steps progress.

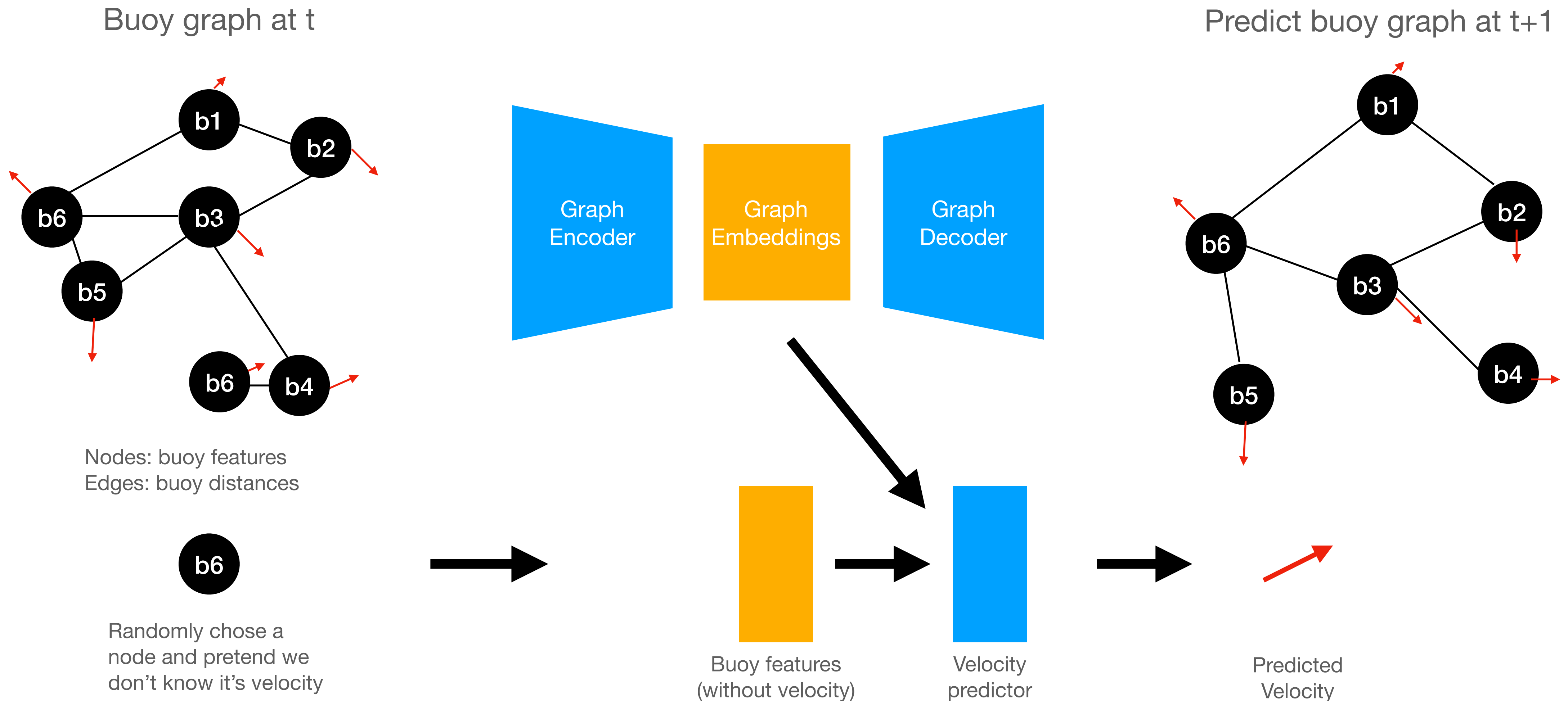
~~Graph Networks~~ for modelling buoy movement

~~GNNs~~

Just some simple dense layers...

Initial idea: Graph Neural Networks

The glacier is a material and buoys are simulated particle

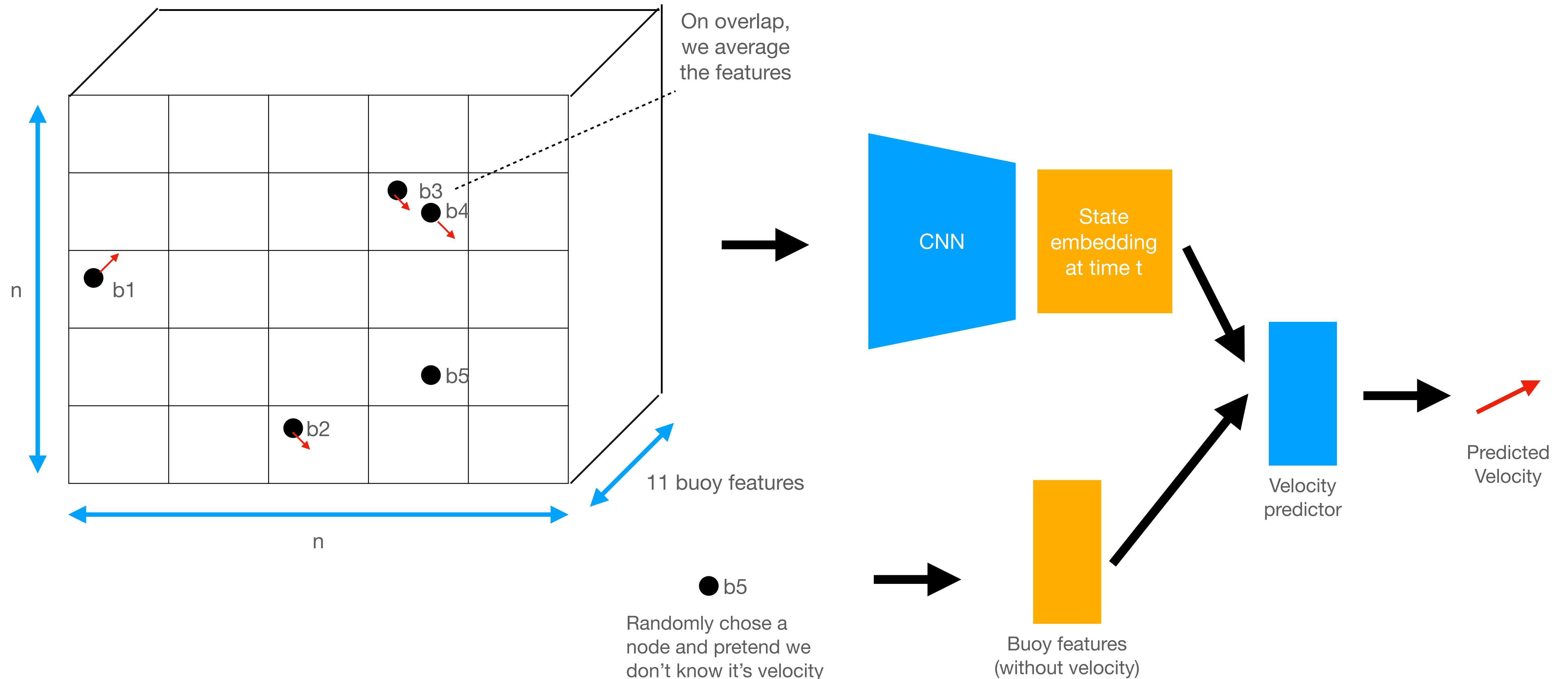


For the graph representation to work, we'd need a constant set of nodes across multiple time steps, which doesn't really happen :(

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

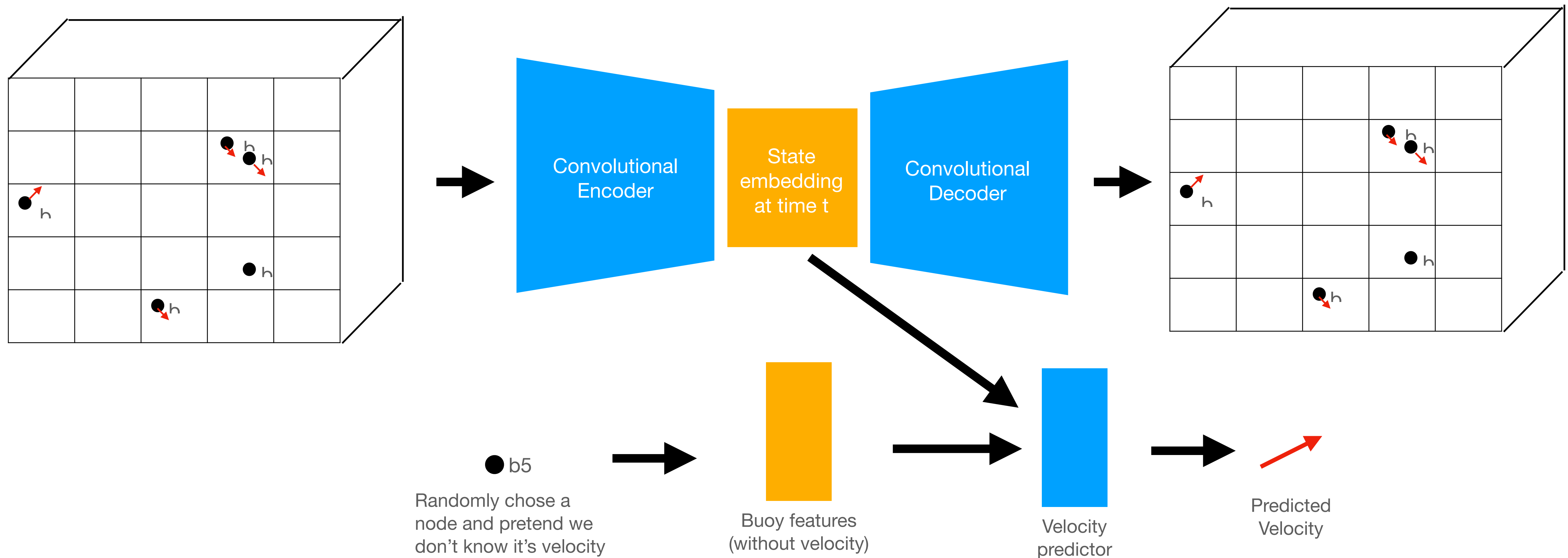


Performance is not great, let's
make sure the embeddings
learned are meaningful!

Back to the drawing board...

Take 2.5: Convolutional Autoencoders

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

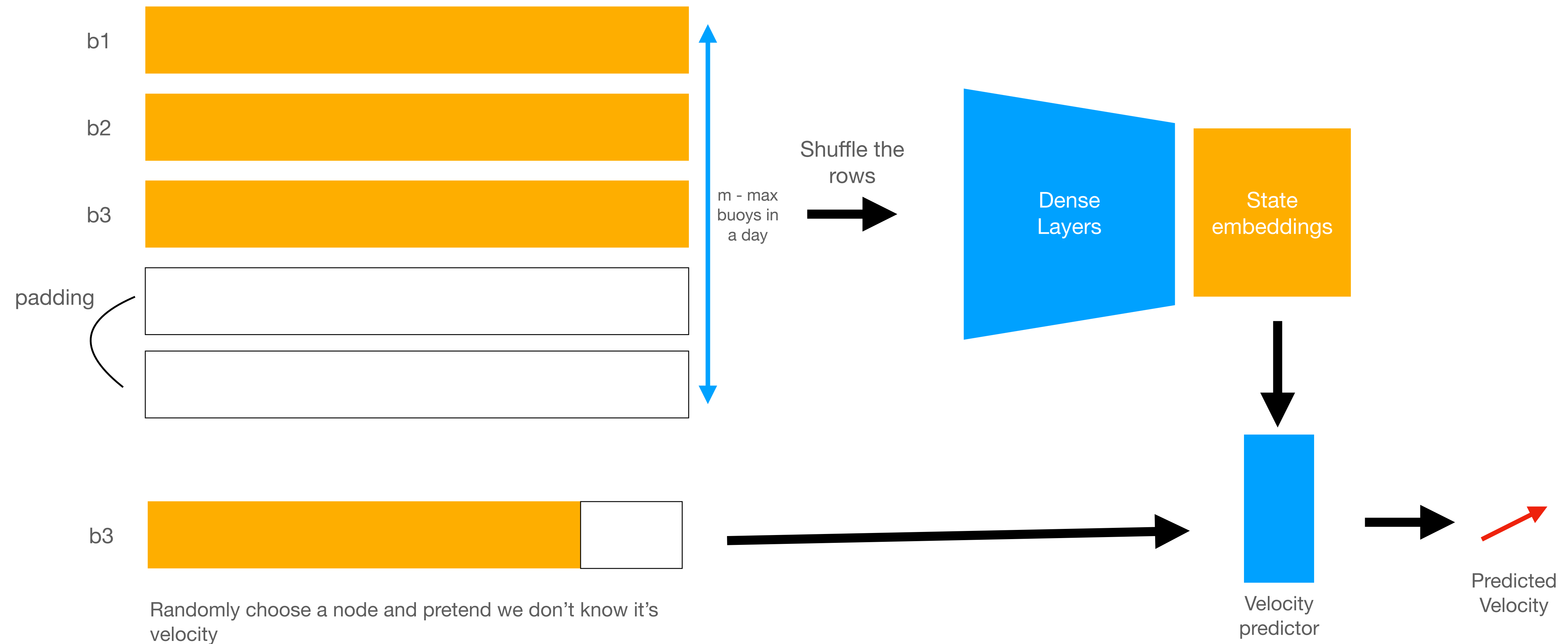


More than half of the days
recorded have buoys that would
overlap in the grid space...

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

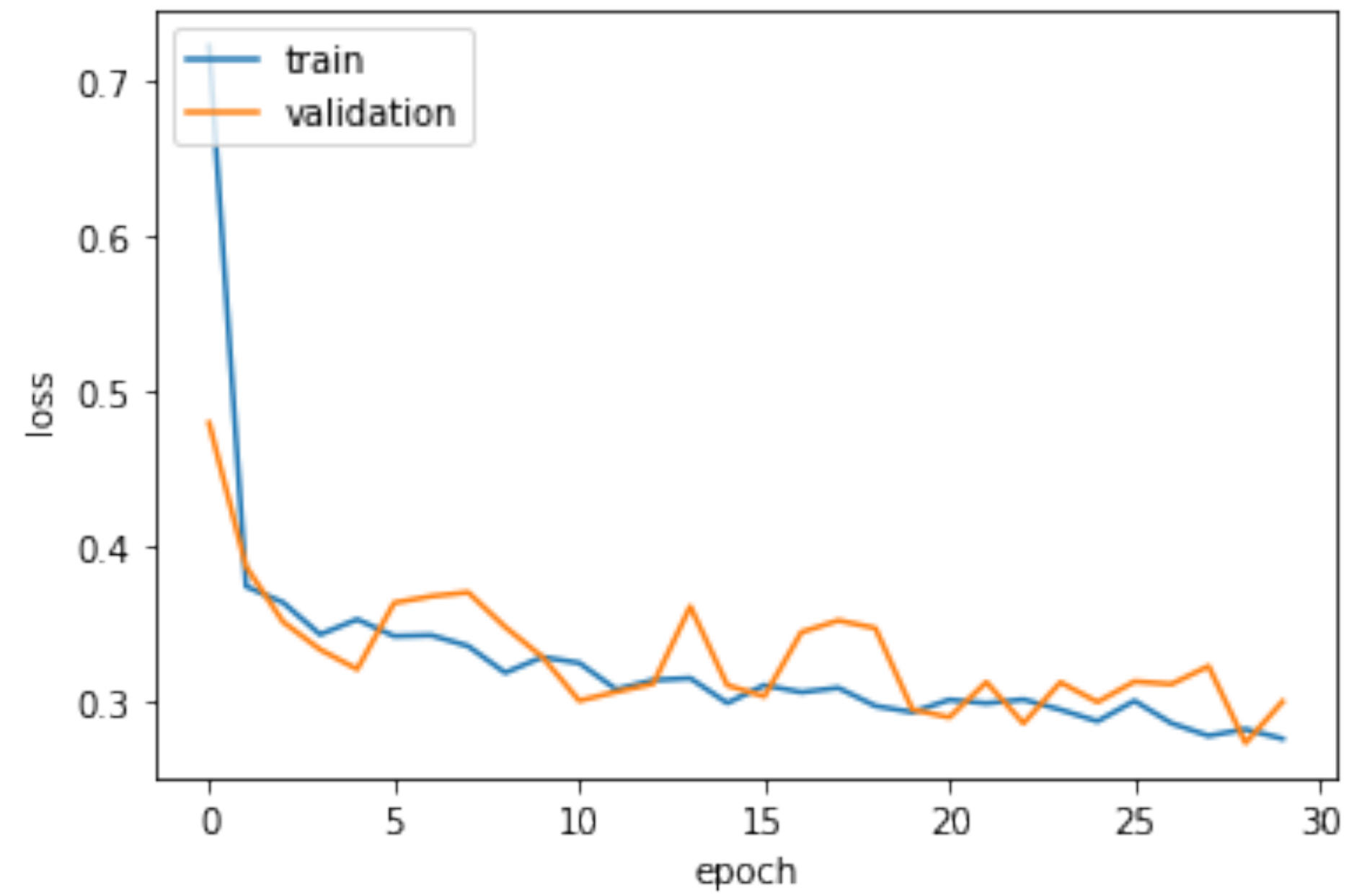
Third Take: A simpler approach

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

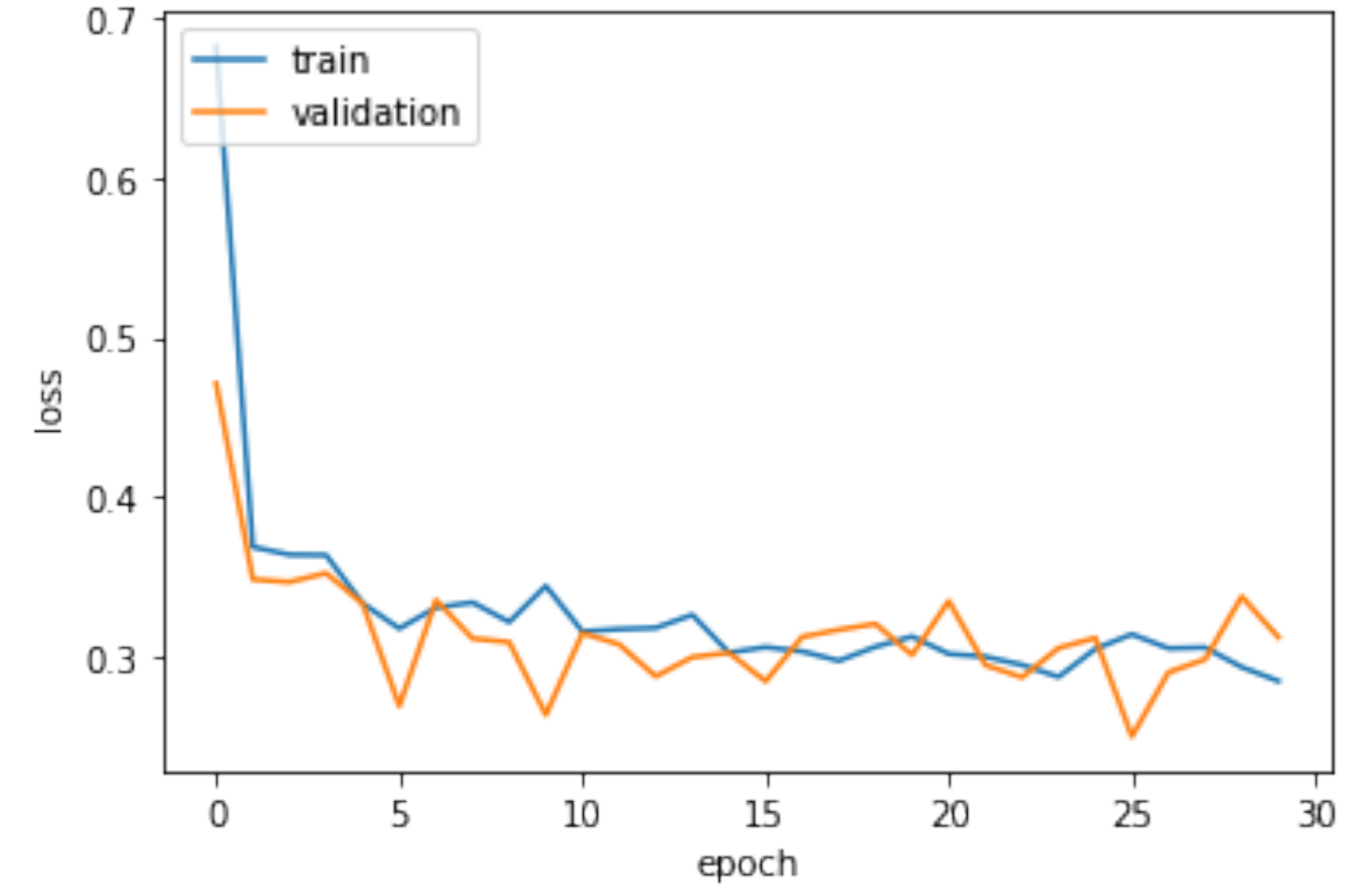


Results

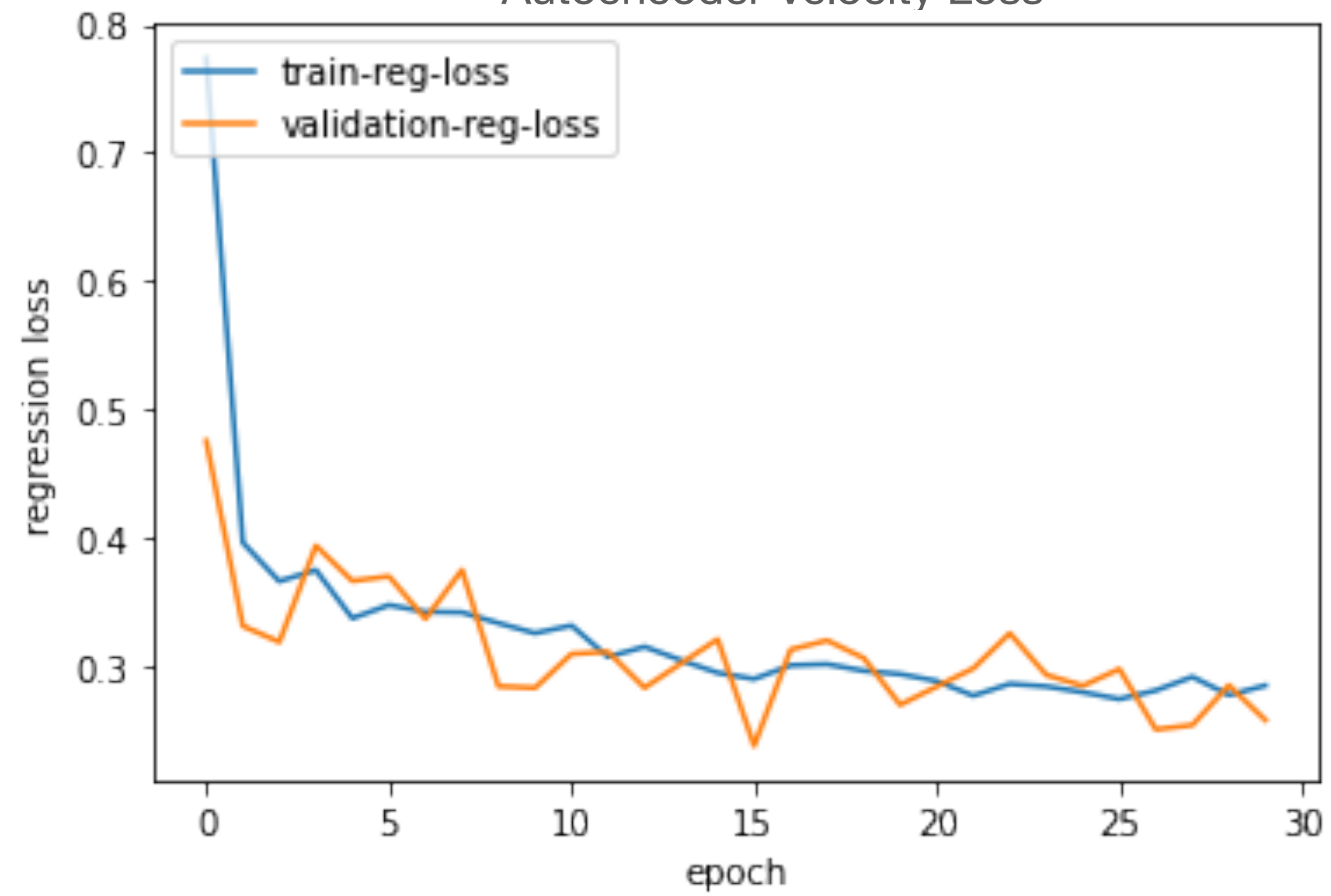
CNN Loss



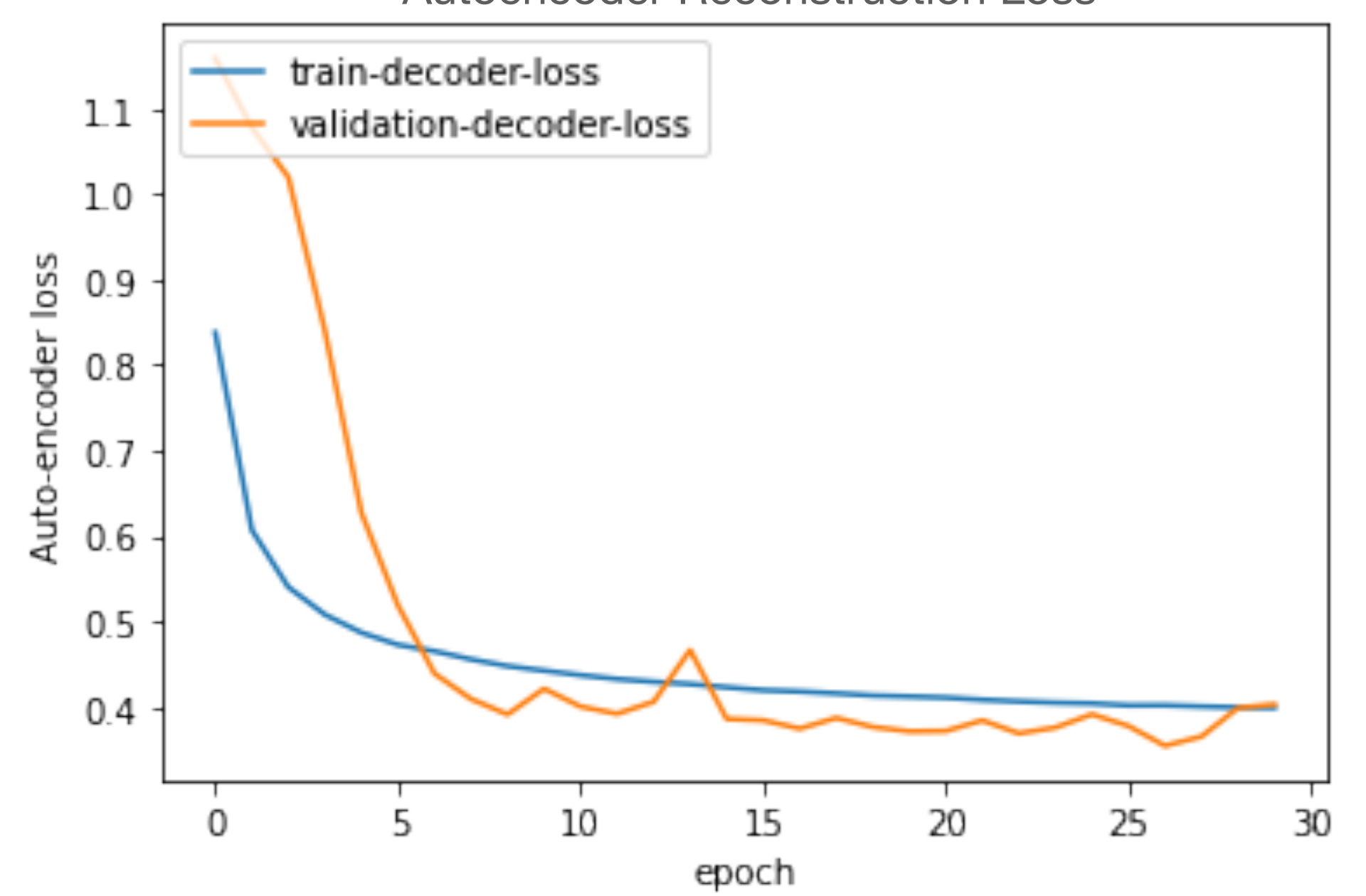
Dense Model Loss



Autoencoder Velocity Loss



Autoencoder Reconstruction Loss



Ideas we didn't have time for