## Documentation of Simulator

Comment – anything in this style is a comment regarding how it’s been implemented or suggesting an implementation.

N denotes total number of nodes in the graph. E – nr edges. n\_f denotes nr of node features per node. e\_f denotes number of edge features per edge.

### Training Loop

Text

Description automatically generated

The simulator predicts acceleration of the particles. The target accelerations of particles are also calculated based on their previous positions.

### Predicting Accelerations

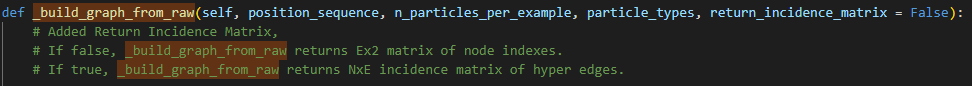
### \_build\_graph\_from\_raw (Henrijs)

The *\_build\_graph\_from\_raw* method extracts N\*n\_f node features, E \* 2 edge list, and E \* e\_f edge features.

A picture containing graphical user interface

Description automatically generated

CHANGE: I have added an extra parameter to \_build\_graph\_from\_raw which when set to true will return the *incidence\_matrix* for a hyper-graph. A basic implementation is working now 😊.



### \_encode\_process\_decode

1. Encodes the graph edge and node features. - Henrijs
2. Runs message passing. - Jakob
3. Extracts accelerations (based on final node embeddings)

Note: step 1 and 2 need changing. Step 3. does not.

Text

Description automatically generated

## Encoder

Text

Description automatically generated

Thought: For Hyper GAT, the attention is learnt, so we don’t really need to process edge features?

## Processor:

1. First edges are updated based on the node embeddings + previous edge embedding
2. Then node representations according to: Logo, company name

   Description automatically generated

This is how it’s currently implemented.

Text

Description automatically generated