Chapter 1

Machine Learning Analysis

1.1 Methods

The objective of this part is to build a machine learning-based tool to infer the interaction potential starting from the history of positions and velocities of an ensemble of Active Brownian Particles. As explained in section **??**, one could use as input some global attribute of the system, such as the radial distribution function, but, as showed by **bag_interaction_2021**, this approach does not work well in out-of-equilibrium cases where an active velocity is present.

1.1.1 The Graph Neural Network

Here, we tried to replicate the approach used by **ruiz-garcia_discovering_2024**, where a vector of positions and orientations of a set of ABPs is given as input to a Graph Neural Network which tries to predict resulting velocities. Our GNN is structured as explained in section **??**, namely with a node and a message function. Both of them have 4 layers, with 300 hidden nodes; each hidden layer has a ReLU (Rectified Linear Unit) activation function, while the output layer of each Network is a simple linear layer without any activation. As for Figure **??**, the input layer of message function is $2n_f$ dimensional, where n_f is the number of single particle features, namely x, y, θ to have information about positions and orientations of a pair of interacting agents. Messages are aggregated using sum as an aggregation function (`'add'` in PyG jargon) to respect the physics of the problem. Node function's input dimension is $n_f + n_m$ where n_m is the output dimension of the message function; this is done because node function has the purpose of taking the aggregated message from all the senders (particle inside a threshold distance Γ) along with the single features of the receiver and trying to predict receiver's velocity (acceleration in Newtonian dynamics).

Here it is important to note that the history is not relevant: the network just takes one instant at a time and predicts instantaneous velocities starting from positions and orientations, without knowing what happens before or after. Instantaneous velocities are computed dividing the difference of consecutive positions by the integration time interval, then they get checked for big jumps, that happen when periodic boundary condition correction take place.

The problem of message dimension n_m is tackled in section **??**. Here, we stuck to a 2D message dimension, as is reported in the reference paper

[**ruiz-garcia_discovering_2024**].

1.1.2 Simulations

To understand the role of the potential's functional form in prediction results, we tried to train and test the network on two different simulations, one done using a spring potential with $k_s = 0.1\,\mathrm{N\,m^{-1}}$ and $x_0 = 12\,\mu\mathrm{m}$ and the other using a LJ potential with $\sigma = 4\,\mu\mathrm{m}$ and $\epsilon = 0.01\,\mathrm{pJ}$. These potential have the difference in range: without limiting our interaction threshold radius, elastic force not only has effect at long distance but its absolute value increases, while LJ is a short range potential. We expect the network to perform worse on LJ since it has less relevant interactions to learn from.

These two potentials required different integration steps: $10^{-3}\,\mathrm{s}$ for LJ and $5 \times 10^{-2}\,\mathrm{s}$ for spring potential. Both simulations were ran with a velocity of $15\,\mu\mathrm{m\,s^{-1}}$, in a $100\,\mu\mathrm{m} \times 100\,\mu\mathrm{m}$ box with 100 particles of $2\,\mu\mathrm{m}$ in radius.

1.2 Training and Testing

We let the two systems evolve for a total simulated time of 1000 s, then instantaneous velocities were computed and finally a sample of 1000 equally spaced snapshots was taken from each simulation. We divided these snapshots in 750 for training and 250 for testing. The network was trained and tested on each simulation's data separately to have preliminary results. We tried the same procedure explained in [**ruiz-garcia_discovering_2024**], where two particles are considered as linked in the graph if their distance is less than some threshold, and then increasing the threshold in subsequent training loops. In our case, after learning with the first threshold distance, loss does not decrease and test results do not improve, so what follows refers to a threshold distance of $20\,\mu\mathrm{m}$.

After each epoch, the network is used to predict velocities on the test set, which contain data with the same potential used in training, saving the message function outputs. Then, a minimization process is used to find the best linear transformation that maps message components into the known forces. This is done to see how results change with the epochs.

1.3 Results

As showed in Figure 1.1, in both cases loss decreases in the 100 epochs of training, meaning that the network is in fact learning.

Regarding the message-force plot, a working network with the right linear transformation should show points on the $x = y$ line, being the message components in perfect correspondence with a rotation of the force components as **??** shows. In Figure 1.2 we reported the qualitatively best results, since the loss scoring in this case does not reflect a prediction quality.

As the animation in [**cranmer_discovering_2020**] shows, as the network learns, the message-force graphs should show a better agreement between what is learned by the GNN and ground-truth. In our case, as it is shown in fig 1.3, in the case of spring potential, the message-force plot has all the points on a vertical line, showing no correspondence between the learned results and ground-truth

forces. This could be caused by a tendency to overfit on the presented data, since loss is decreasing while test result are worse.

Anyway, it is possible to notice a difference between the two potentials: in the case of Lennard Jones, few points are outside the vertical line and they position in an horizontal cloud around it, while for the elastic potential most points are scattered around an horizontal cloud and a density increase can be observed in diagonal lines.

1.4 Conclusions to Chapter 4

Here we presented the preliminary version and results of a machine learning tool that predicts forces between interacting ABPs, in the spirit of the ActiveNet system developed by **ruiz-garcia_discovering_2024**.

As shown in section 1.3, more work is needed in order to make this tool work as expected. In particular, training for longer with more simulation snapshots is certainly needed in order to get the right accuracy. Retraining process with increasing threshold distance must be perfected, in order to get at the same level of the reference paper.

After making the training process work as desired, the next step is to retrain the network with snapshots taken from simulations with different potentials, in order to increase its generalization capability.

Moreover, testing with experimental data will need some more adjustments. In an experimental setting, defects inherited from the fabrication process like stuck particles and particle clumps can make it harder for a machine learning tool to work properly. In the case of experimental data collected in Microscale Robotics Lab, it is hard to measure particles' orientation, since the two hemispheres of silica Janus particles are not clearly distinguishable, though a difference in color is noticeable. Moreover, these particles are free to rotate in three dimensions and nothing prevents them from pointing one hemisphere to the microscope objective, not showing the separation line.

We can state that these results, though certainly not satisfying, can be a starting point to further developments of this tool, getting it to work as expected. With the potentials inferred from experiments, we will be able to model particles

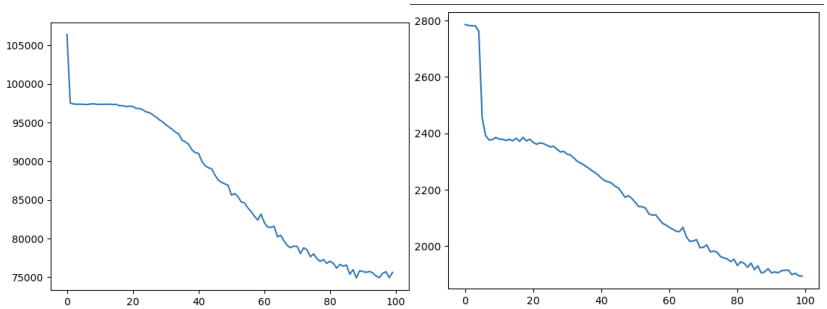
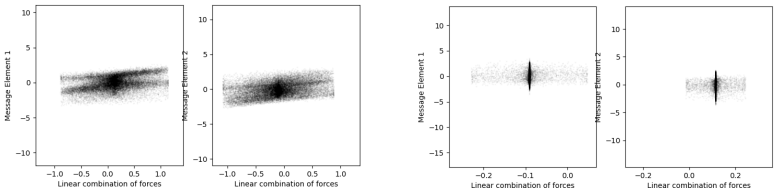
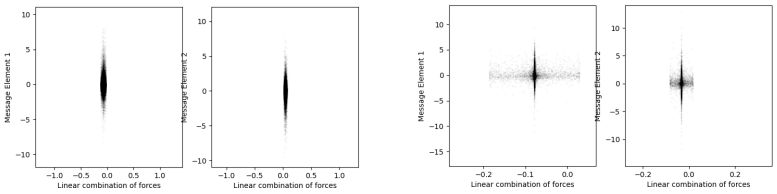


Figure 1.1: **NOT DEFINITIVE IMAGES** Learning curve for elastic potential (left) and Lennard Jones (right). Loss is the absolute difference between predicted and ground truth velocities.



(a) (b)

Figure 1.2: **NOT DEFINITIVE IMAGES** Representative message-force plots of early epochs of training for spring potential (left) and Lennard Jones (right).



(a) (b)

Figure 1.3: **NOT DEFINITIVE IMAGES** Representative message-force plots of last epoch of training for spring potential (left) and Lennard Jones (right).

in a more accurate and complete way, to simulate them and use simulated data back in a simulation driven inference fashion.