5 Supplementary Materials

5.1 Transition State Predictions

A state along the reaction path that corresponds to a maximum of potential energy is called a transition state. A transition state is often of particular interest because it dictates the activation energy of the reaction and corresponds to bonds breaking and new bonds forming [1]. Transition states are helpful in figuring out reaction rates in chemical systems and in determining whether reactions are physically possible. In Figure 4 we display the predicted structures for the highest energy transition state of each reaction. We observe the NN-BAX structures qualitatively match the classical NEB structures.

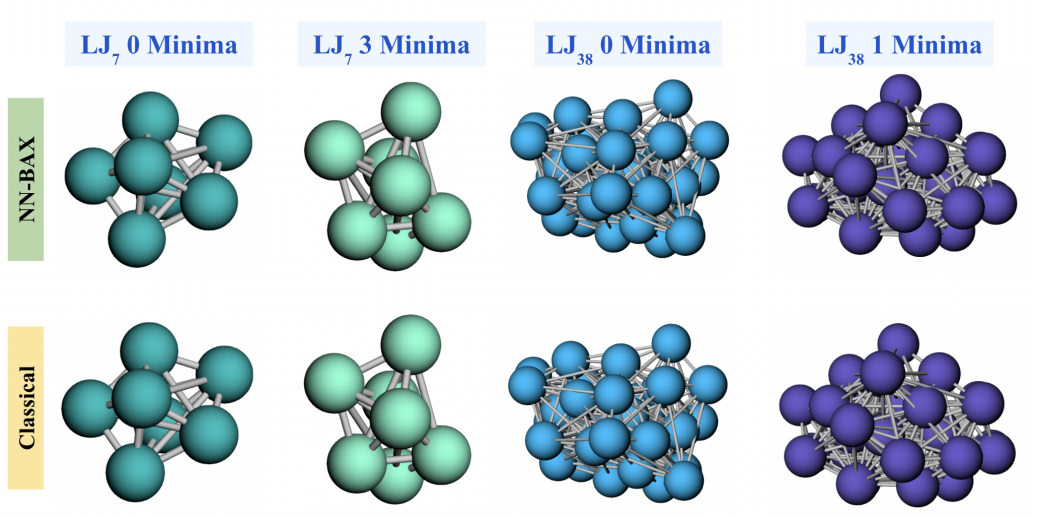

Figure 4: Transition state predictions. The NN-BAX predictions and the classical ground truth atomic arrangements are displayed.

5.2 NN-BAX Convergence

In this paper we introduce a new convergence hyperparameter, MAE_i, which corresponds to the error in the prediction of model $i-1$ on datapoint i. In Figure 5 MAE_i is plotted for each BAX iteration, for the four paths we analyze. We observe it to be a relatively stable parameter, plateauing as we increase the BAX iteration. In order for NN-BAX to converge, $\mathrm{MAE}_i < m$ and $f^i_{max,BAX} < t$ must simultaneously be satisfied, where m and t are the respective convergence thresholds. For all results presented here, we require $\mathrm{MAE}_i < 0.1$. For force convergence, we set $f_{\mathrm{max,BAX}} = n\, f_{\mathrm{max,classical}}$, where $n > 1$ allows the model predictions slightly more error than the true potential. n is a hyperparameter, so various values may work for different paths. In practice, we found $n = 2$ to work well for our paths. For further confidence in convergence we introduce a patience metric p, where the convergence criteria must be achieved p times for NN-BAX to converge. In this paper we use $p = 2$ for both LJ_7 paths and the LJ_{38} path with one minimum, and $p = 3$ for the LJ_{38} path with zero minima.

Finally we comment on convergence time and BAX overhead. Specifically the LJ_{38} 0 minima path took NN-BAX 9.3 hours to run, for 30 BAX iterations, with 200 NEB steps, and with each model trained for 50 epochs. All code was run on a single NVIDIA Tesla A100 GPU. Since calls to the LJ potential are practically instant, this is a good approximation for solely the computational overhead cost of NN-BAX. Thus for a setting in which DFT is being used, for NN-BAX to achieve a wall-clock speedup relative to Classical NEB, the DFT simulations would have to be longer than 25.4 seconds. Finally we note that the NEB calculations with the model were not parallelized, but could be, which would further improve the speedup.

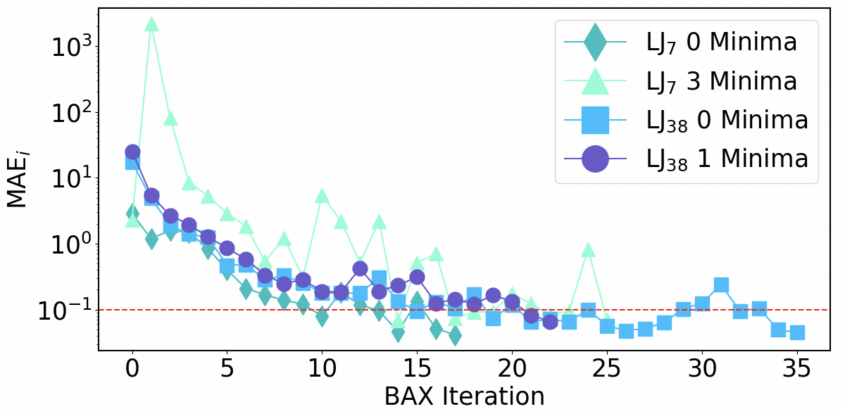

Figure 5: MAE_i convergence metric versus BAX iteration. The red line denotes the convergence threshold of 0.1

5.3 Training

In each BAX iteration, the network was trained for 50 epochs with a batch size of 2 using the AdamW optimizer [44] (weight decay 0.001, gradient clipping at 100) and an initial learning rate of 2×10^{-4}. A cosine learning rate schedule was employed via `LambdaLR`, with a warmup phase lasting 1 epoch (warmup factor 0.2) and a minimum learning rate set to 1% of the initial value. Exponential moving average (EMA) updates with a decay of 0.999 were applied throughout training. We use the mean absolute error (MAE) loss, giving forces four times the weight of energy since forces are critical in NEB precision and convergence. All models are trained on a single NVIDIA Tesla A100 GPU at the SLAC Shared Science Data Facility.