Physics Informed Neural Networks for Molecular Dynamics Applications

Taufeq Mohammed Razakh[1], Beibei Wang[1,2], Shane Jackson[1,2], Ken-ichi Nomura[1,2], Aiichiro Nakano[1,2], Rajiv Kalia[1,2], Priya Vashishta[1,2]
[1] University of Southern California, Los Angeles, CA
[2] Collaboratory for Advanced Computing and Simulation

USC Viterbi School of Engineering

TL;DR: A neural network approach to solve the differential equations governing molecular dynamics(MD) systems where the dynamics are governed by Hamilton's equations

Overview

- Neural Network(NN) learns by minimizing the least action, function considers the equation of motions, initial and boundary conditions and provides phase-space trajectories that are in excellent agreement with the trajectories obtained by the numerical solution approach
- An Object-Oriented software package allowing users flexible interaction with MD engine and NN utilities written in C++
- Code available at `https://github.com/USCCACS/PND`

Related Work

- Highly scalable software packages such as RXMD simplifies the need for time-accelerated algorithms for atom trajectories by the use with the application of neural networks as an alternative to numerical solver for ODEs.
- Physics-informed neural networks (PINNs) have been successful in applying automatic-differentiation to solve many DEs including heat equation [1], Burger equation[2], Navier Stoke's equation [3], Schrodinger equation [4], Hamilton's equations of motions [5] and general applications [6] [7] [8]

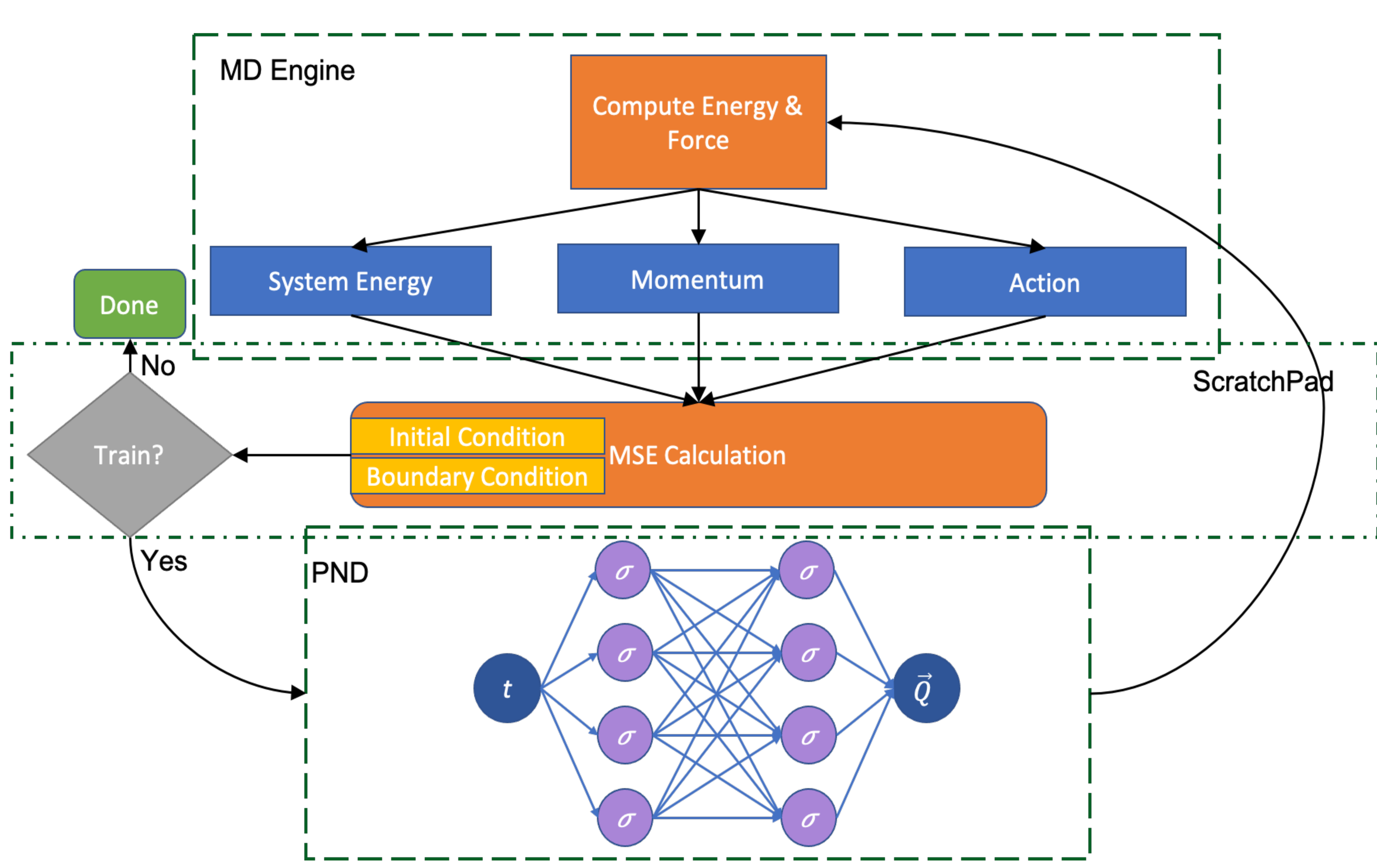
Problem Setting

- Atoms in a system are at co-ordinates $r^N = (r_1, r_2, ..r_N)$ with potential energy $\mathcal{U}(r^N)$ the atomic moment $p^N = (p_1, p_2, ..p_N)$ can be written in terms of kinetic energy as
$$\mathcal{K}(p^N) = \sum_{i=1}^{N} |p|_i^2 / 2m_i$$
- The energy or the Hamiltonian $\mathcal{H} = \mathcal{K} + \mathcal{U}$
- The equations of motion are $\dot{r}_i = p_i/m_i$ & $\dot{p}_i = f_i$
- Canonical/ Hamilton's equations $\dot{r}_i = \frac{\partial \mathcal{H}}{\partial p_i}$ & $\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i}$
- For the purpose of expressing the Hamilton's equations in symplectic notation we assume the variable, z represent the collection of space and momenta co-ordinates
$z = (r_1, r_2, ..r_N, p_1, p_2, ..p_N)^T$
- $\mathrm{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
- $\dot{z} = \mathrm{J}.\Delta_z \mathcal{H}(z)$

Physics Informed Neural Networks

- The maximum discretization unit in the time integral is dictated by physical properties of the target system - pressure, temperature, phonon frequency. **A bottleneck arises due to the sequential dependency in the time integration to solve the equations of atomic motion.**
- Too large time renders the numerical solver unstable; drifts in conserved properties
- One way to overcome the limitation in discretization is to solve partial differential equations using neural network with a point in time *t* as an input to the network
- General form of solution when using NN:
$\hat{z}$(t) = z(0) + N(t)
Where, N(t) is feed forward fully connected neural network and Z(0) is initial state at t=0
- Generalizing MSE/Loss function:
$$L = \frac{1}{K}\sum_{n=1}^{K}(\hat{z}(\mathrm{n}) - \mathrm{J}.\Delta_z\mathcal{H}(z))^2$$
- Defining Onsager Machlup (OM) Action term as
$$S_{OM} = \int_0^T \left[\sum_{i=0}^{N} \left\| m_i \ddot{\mathrm{r}}_i(t) + \frac{\partial \mathcal{U}(R(t))}{\partial \mathrm{r}_i} \right\|^2\right] dt$$
Where, m_i is mass of i^{th} particle; $\ddot{\mathrm{r}}_i(t)$ *is second-order time derivative (i.e., acceleration) of i^{th} particle;* $\mathrm{R}(t)$ *is Coordinates of all the particles in the system at time t and $\mathcal{U}(r^N)$ is the potential energy of the system*
- $$Loss = \lambda_1 S_{OM} + \lambda_2 (Q(0) - Q_0)^2 + \lambda_3 (Q(T) - Q_T)^2 + \lambda_4 \sum_{t=0}^{T} \left[E\left(Q(t), \dot{Q}(t)\right) - E_0\right]^2$$

$E\left(Q(t), \dot{Q}(t)\right)$ is the total energy with ε and σ as parameters of the LJ potential

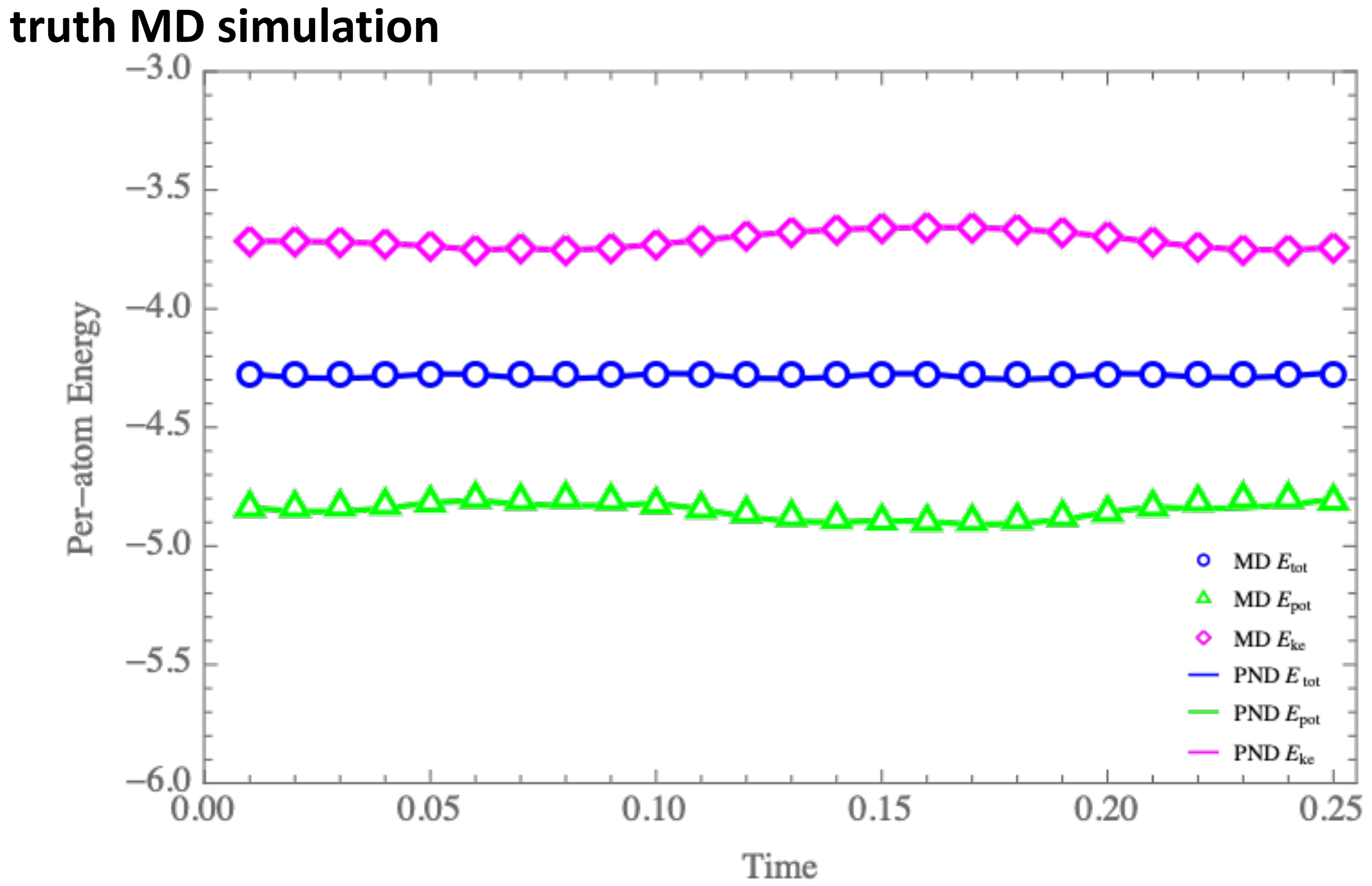


Experimental Results

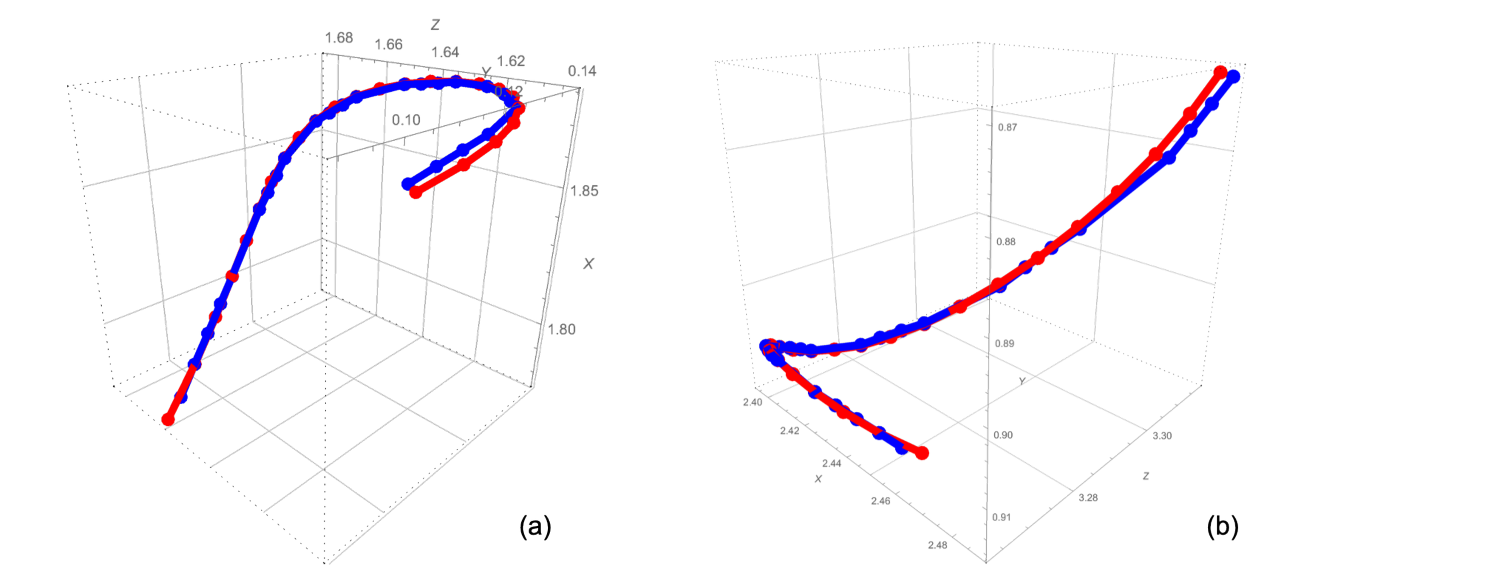
The simulation was carried out for 32 atoms in FCC configuration in time steps of 1 ps

- Gradients computed for 50,000 iterations

The three energies predicted by PINN match well with the ground truth MD simulation



Two selected NN (blue) and MD (red) trajectories show that PINN can reproduce non-trivial paths



References

[1] H. He and J. Pathak, "An unsupervised learning approach to solving heat equations on chip based on Auto Encoder and Image Gradient," *ArXiv,* vol. abs/2007.09684, 2020.
[2] F. Lu, "Data-driven model reduction for stochastic Burgers equations," *ArXiv,* vol. abs/2010.00736, 2020.
[3] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," *Journal of Computational Physics,* vol. 426, p. 109951, 2021-02-01 2021, doi: 10.1016/j.jcp.2020.109951.
[4] A. L. Kapetanovi and D. Poljak, "Numerical Solution of the Schrödinger Equation Using a Neural Network Approach," *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM),* pp. 1-5, 2020.
[5] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, "Hamiltonian Neural Networks for solving differential equations," *ArXiv,* vol. abs/2001.11107, 2020.
[6] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Trans Neural Netw,* vol. 9, no. 5, pp. 987-1000, 1998-01-01 1998, doi: 10.1109/72.712178.
[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," in *Journal of Computational Physics* vol. 378, ed: Elsevier Inc., 2019, pp. 686-707.
[8] D. Zhang, L. Lu, L. Guo, and G. Karniadakis, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," (in English), *Journal of Computational Physics,* Article vol. 397, NOV 15 2019 2019, Art no. ARTN 108850, doi: 10.1016/j.jcp.2019.07.048.