Simulation study of relaxation in glassy polymers via machine learned Markov state models

Genevieve Ke

University of British Columbia, Department of Physics and Astronomy

E-mail: tt1224@student.ubc.ca

Abstract. Amorphous materials like glassy polymers in condensed matter are showing complicated structural dynamics on long timescales. It is hard for analyzing using conventional methods. By using machine learning and deep learning techniques, I constructed a Variational principle for Markov processes(VAMP)/ Graph Dynamical Networks (GDyNets) based Markov state model (MSM) for a model glass formers at T = 1. It coarse-grains the dynamics with complicated local environment with high dimensions into a low-dimensional space. The transition timescale between states defined in the work, state 0 and state 1, is much longer than the conventional *alpha*-relaxation timescale which reveals the glass transition. The learned map of states to the atom particles exhibits different probabilities of being at 1 state. For this homogeneous system, it shows the heterogeneity feature. After evaluating by CK equation, the MSMs were proved to effectively construct the feature map function and thus get the relaxation timescale. These results could give more information by calculations like free energy and shows local packing fluctuations.

1. Introduction

In the condensed matter, it is essential to understand the dynamics at atomic scale for a better design of materials to get the best performance and tackle global environment and energy challenges[1]. Molecular dynamics simulation is widely used to explore the dynamical processes in materials[2]. In amorphous materials, molecular dynamics simulation can generate large amount of data and thus it is challenging to extract useful and relevant statistics for atoms[2]. Amorphous materials is a kind of non-equilibrium material whose arrangement is like liquid[3]. Glassy polymer is one of the most important classes of amorphous materials and is often used in aerospace composites, contract lens, and many other things in our daily life[4]. One of the most prominent features of glassy polymers is the relaxation timescale which can influence mechanical properties of polymers[5]. The relaxation timescale means the time of completely changing the state map. Each atom will be in different state with different properties. To show the importance of relaxation timescale in glassy polymers, one study shows the coarse grained dynamics of excess Voronoi volume. Voronoi volume means the empty space around each atom and different states decide different sizes of volume and result in their packing or unpacking mechanism[6]. However, the relaxation timescale is much longer than the main structural relaxation time, which is known as α-relaxation time associating with the glass transition[7]. Thus, the long time dynamics of a model glass transition from 1 state down to another regime would be very challenging using standard analysis of dynamics. In this work, a two-state Markov state model was constructed to reveal relaxation processes. The conversion timescale is denoted as τ between two defined states, state 0 and state 1.

In order to learn the dynamics in such complex glassy polymer, it is essential to have a method that can share the knowledge learned for a local environment to the global material system to have less data needed[2]. The deep learning architecture, Graph Dynamical Networks (GDyNets), combining Markov state model and graph convolutional neural networks (GCN) has been developed to do such work allowing for a sharing knowledge learned from the local across to the whole system[2]. It is often called Koopman model in complex dynamical systems analysis or fluid dynamics. Variational principle for Markov processes(VAMP) formulation of Markov state models, can extract dynamical information from atoms' trajectories much shorter than the timescale of interest and summarize their dynamics into a low dimensional and linear model[6]. In this work, VAMP/GDyNets based MSMs is used to study the relaxation timescales of a model glass polymer forming from 1 state to the other.

2. Methods

2.1. Molecular Dynamics Simulation in Lammps

For generating molecular dynamics data, firstly I need to do the molecular dynamics simulation. A molecule like protein contains atoms, covalent bonds. Atoms of the molecules are considered as particles while covalent bonds are considered as flexible springs. Thus, the size of each particle can be defined by Van Der Waals radii of that specific atom type and the strength of the spring is defined by the strength of the bonds[**?**]. Electrostatic interactions also play important roles if charges have been assigned to them, but for simplicity I ignored the electrostatic interaction in my work.

Molecular dynamics simulation consists of the numerical solution of the two classical equations of motion, which for one simple system of an atom can be written as:

$$m\ddot{r} = f$$

$$f = -\frac{\partial U}{\partial x}$$

We need to be able to calculate the forces f acting on the atoms, and these are usually derived from the potential energy U. Thus, we focus on this function U first. For the potential energy, it consists of two main interactions, non-bonded interactions and bonded interactions between atoms. The Lennard-Jones potential is the non-bonded interactions which describes the potential energy between two non-bonding atoms based on their distance of separation. The potential equation looks like:

$$V = \frac{Q_1 Q_2}{2\pi\epsilon r}$$

In this equation, V is the inter-molecular potential between the two atoms, and ϵ is the well depth measuring of how strongly the attraction between two particles. Besides the non-bonded potentials, we must also consider the bonded interactions. The molecular model will be similar to two balls connected by a spring, so the bonded potentials can be expressed like:

$$V = \frac{1}{2}\sum k(r - r_{eq})^2$$

In this equation, a harmonic form with specified equilibrium separation is assumed, so the bond stretching can be described by Hooke's Law.

In summary, the mechanics of molecular dynamics simulation is shown as in fig 1. Firstly, each atom is considered as a particle with mass m, and the initial position is provided, which acts as input information. We can get the forces acting on each atom by the potential energy.

The potential energy is equal to non-bonded potentials and bonded potentials. Then, we can calculate the acceleration of the atom at time t by Newton's Second Law. Velocity at next time t' can be calculated by knowing the acceleration. Thus, the new position at t' can be calculated by knowing the velocity. The time difference in the two time t' and t will be the time step of the process. We can do this cycle continuously till the end of the running time. Finally, each cycle will generate data.

Finally, in the simulation box, boundary conditions are required to control the flow of the atoms in then system. Because there is a called edge effect, which is due to movement of system and atom interactions during simulation, periodic boundary conditions need to be used. As shown in fig 2, the central box represents the simulation system and surrounding boxes are replica. In each system, whenever an atom leaves a simulation cell, it is replaced by the one which enters from the opposite cell and keeps the number of atoms in one simulation box conserved.

In my work, I did the molecular dynamics simulation in a Molecular Dynamics Simulator software called Lammps. The simulation was focused on an extremely well characterized model glass former, namely N=32,000 particles of a Lennard-Jones bead-spring polymer in a periodic simulation box at temperature T = 1, using Lennard-Jones units. This is part of the information in the input script for the MD simulation.

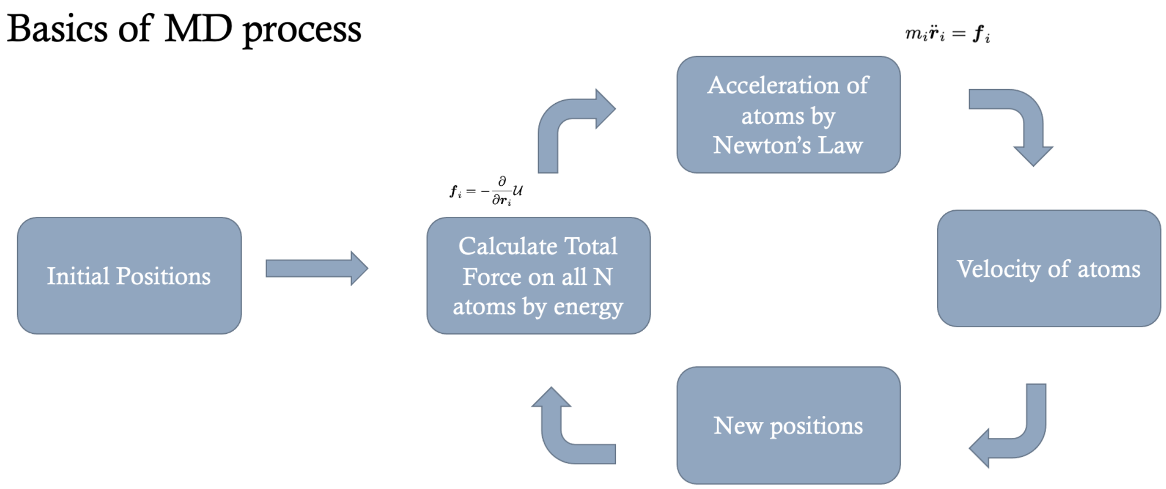
Basics of MD process
$m_i\ddot{\boldsymbol{r}}_i = \boldsymbol{f}_i$
Acceleration of atoms by Newton's Law
$\boldsymbol{f}_i = -\frac{\partial}{\partial \boldsymbol{r}_i}\mathcal{U}$
Initial Positions
Calculate Total Force on all N atoms by energy
Velocity of atoms
New positions

Figure 1. The basic flow of Molecular Dynamics process

2.2. Markov State Model

We can use Markov models to study long time dynamics by reducing the dimensions of the dynamics. The relationship between the state at time t and the state at time t + τ is related to the transition matrix K, as shown in the following equations:

$$P(t+\tau) = \mathbf{K}P(t)$$

All information about the dynamics and long time behaviour of the system is now encoded in this matrix. In the transition matrix K, the elements are probabilities of observing the system in one state at t + τ, given it was in another state at time t. The solution to the system of the equation would be:

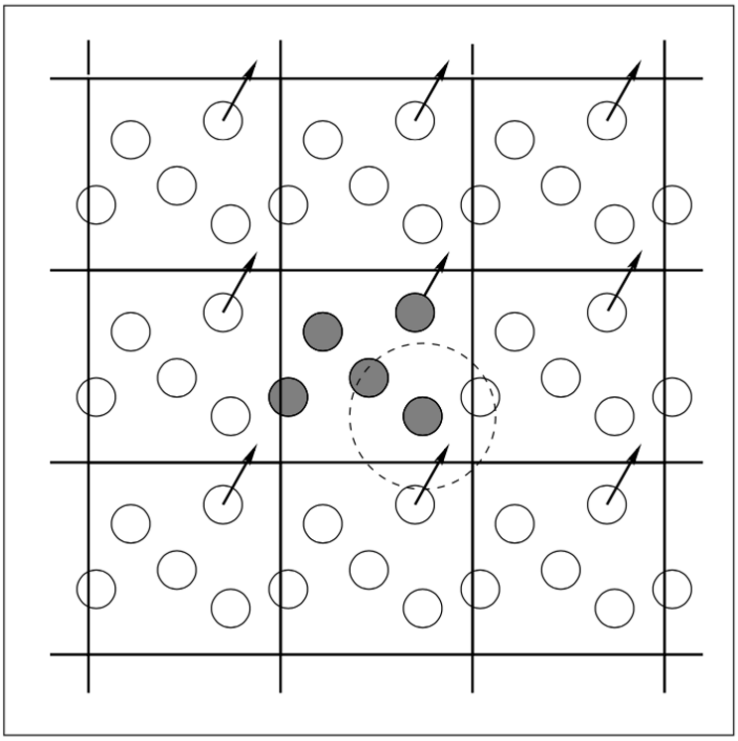

Figure 2. The periodic boundary condition demonstration

$$P(t) = e^{\mathbf{K}t}C$$
$$P(t+\tau) = e^{\mathbf{K}\tau}P(t)$$

In the analysis of atomic scale dynamics, I described the dynamics of the atoms and their surrounding atoms as a discrete process in MD simulations, as shown in the following equation:

$$x_{t+\tau} = F(x_t)$$

where x_t and $x_{t+\tau}$ represent the local configuration of the atom and their surrounding atoms at time steps t and t + τ, respectively. This above equation shows that the dynamics of x is Markovian, so $x_{t+\tau}$ will only depend on x_t, not any configurations before it. We usually know the Markov state model in condensed matter as the Koopman theory, which states that there exists a function χ. The local configuration of target atom x can be mapped into a lower dimensional feature space by using the feature map function, χ, such that the dynamics function F of atom, which is non-linear, can be simplified by a linear transition matrix K. The equations of mapping are following:

$$\chi(x_{t+\tau}) = K^T\chi(x_t)$$

It will be powerful in classifying and dimensionality reduction.

2.3. Learning Feature Map Function with Graph Dynamical Networks

Now, the most important thing is to learn the feature map function χ. GCN (graph convolutional neural networks) was used to learn the feature map function. GCN provides a general framework to encode the invariant structure of materials [2]. The method, convolutional neural network, that we generally know, enables us to propagate the information of neighbors to each input. As illustrating in figure 3 and figure 4, a given size of filter encodes information of a cell and its neighbors and pick important features in data. The filter overlays onto the image and values on the filter cells linearly combine with the pixel values which gives the output. For example, the filter in figure 3 searches for vertical lines in the image. In atomic simulations, however, the particles are not in regularly spaced grids. Thus we need graphs that involve nodes, representing particles, and edges representing distances between neighbouring atoms.

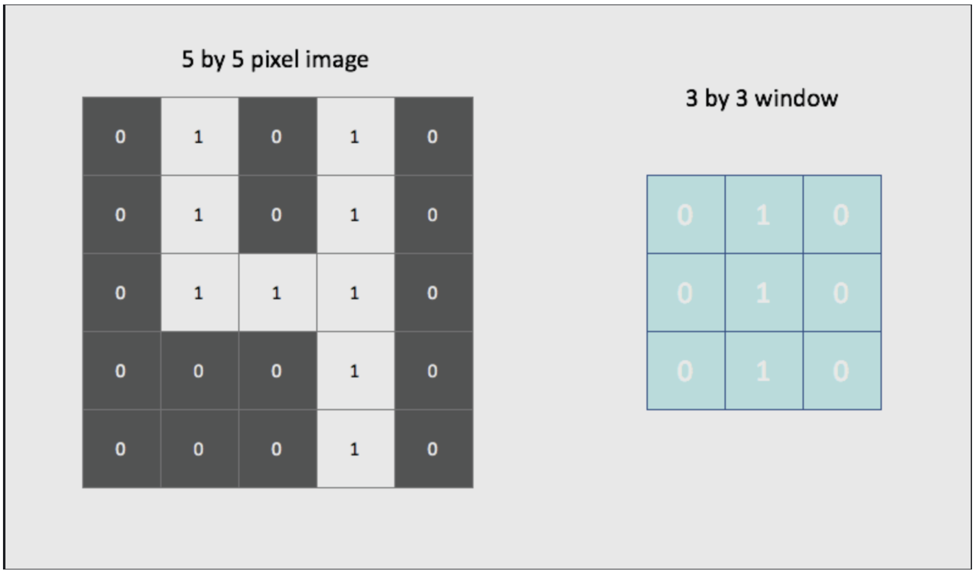

Figure 3. The window searches for vertical lines in the image

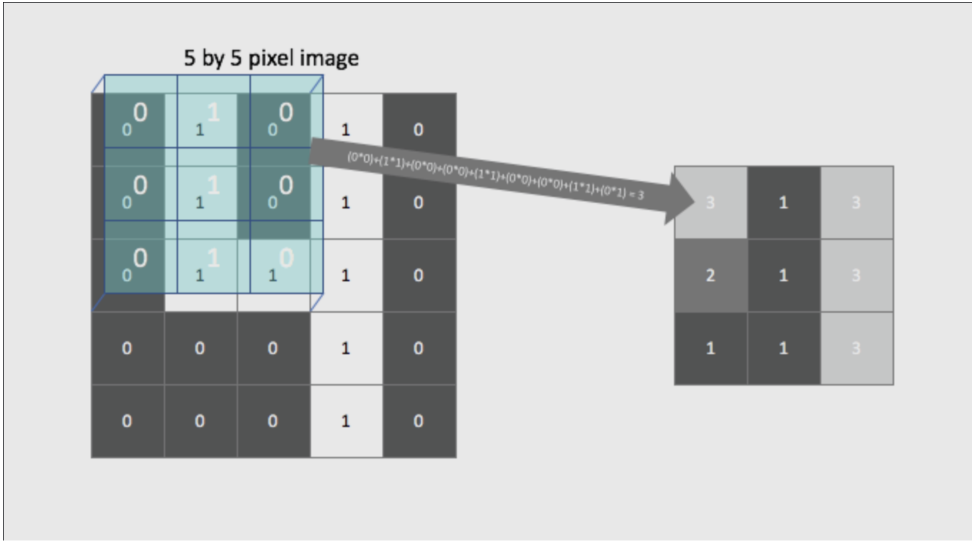

Figure 4. The output (shown on the right) by using the CNN (convolutional neural network)

For each time step in the MD trajectory, a graph is constructed based on its current configuration with node v_i, representing an atom and edge u_ij representing the distance or bond connecting the nearby atoms. Then, each graph that already constructed will be the input to the same graph convolutional neural network (GCN) to learn an embedding for each atom. It also incorporates the information of its surrounding environments. After K times operations, information of the Kth neighbors will be embedded to each atom, including the local environment, so that a vector which quantifies local environment of particles will be the output. In the end, we are able to build a shared two-layer connected neural network and an output layer. It has been shown that the best approximation of the feature map function is obtained when VAMP score, which is a score that we can compute the loss of the system, is maximized. Basically VAMP score can minimize the error in Markov state model[2].

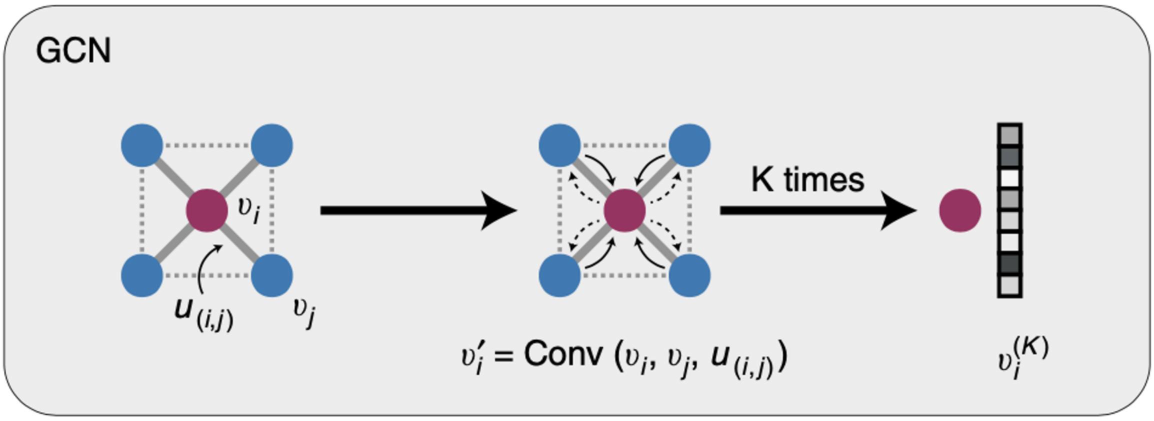
GCN
υ_i
$u_{(i,j)}$
υ_j
K times
$\upsilon_i' = \mathrm{Conv}(\upsilon_i, \upsilon_j, u_{(i,j)})$
$\upsilon_i^{(K)}$

Figure 5. Illustration of the architecture of the graph convolutional neural networks.

2.4. Model Validation

After learned the feature map function, we need to find out if the function is valid, which means if the dynamic model learned is Markovian. This can be evaluated with a Chapman-Kolmogorov (CK) equation:

$$K(n\tau) = K^n(\tau), n = 1, 2, ...$$

If this equation holds, then the dynamic model learned is Markovian, so it can accurately predict the long-time dynamics of the system. This means if the transition matrix at time τ to the power of n, where n is the integer from 1 to infinity, is equal to the transition matrix at time $n\tau$, then the model passes.

3. Results and Discussion

Now let's look at what I got for the results. In figure 6, the output graph shows the average population of each state in all the data frames. 54.97% of the average atoms from all data frames were in the state 0 while 45.03% of the population were in the state 1. In the 3D visualization of simulation box in Ovito, a software tool for scientific data visualization and analysis for molecular dynamics simulation model. As clearly shown in figure 7, each ball represents an atom and the color represents the probability of being in state 0 at the time step that taking this snapshot. The color which is close to red and black means lower probability of that atom being in state 0 whereas higher probability of being in state 1 at that time step. The color which is close to yellow and white means higher probability of that atom being in state 0 whereas lower probability of being in state 1 at that time step. Distinct clusters of particles emerge in the same state in this homogeneous model, which reflecting a form of heterogeneity. Also, to have more snapshots at different time steps, an animation of how the states of atoms evolve along with the time can be made.

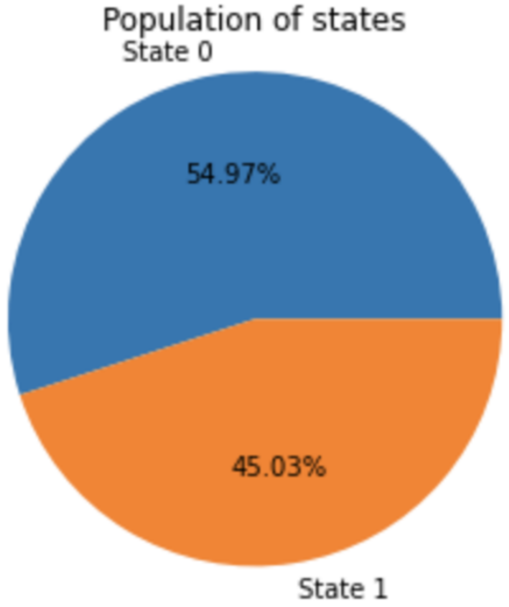

Figure 6. Pie chart of population of states

Next, plotting the relaxation timescales. We plot how the relaxation timescales act as a function of the lag time. Lag time is the time step that set in the MD simulation. Lag in x-axis is an array of lag times at which we compute the relaxation timescales. It could be estimated to something like np.range(1,number of frames in the test data set,5). Once the relaxation timescales stops to change and converges, the model starts to follow the Markovian assumption. We want to find the timescale as a function of time to make sure it converges to a constant value. As shown in figure 8, the black curve shows the line timescale(y-axis) = lag(x-axis). Since the y-axis is on a log scale, it's a curve on a semi-log plot. From figure 8, we can estimate that starting at approximately 0.9, the model starts to follow the Markovian assumption. Now, in order to prove that is valid and accurate, CK test is carried out. The transition matrix with lag time should pass the CK test. As shown in figure 9, blue curve is the values predicted by direct multiplication of the transition matrix (left hand side of the CK equation) and red curve is the estimated values from the trajectory data (right hand side of CK equation). Probability is the y-axis while lag time is the x-axis. There are four graphs in figure 9, which representing from state 0 to state 0, from state 0 to state 1, from state 1 to state 0, and from state 1 to state 1 respectively. As we can see, they match very well so proving it passes the test.

4. Conclusion and Implication

In summary, graph convolutional neural networks presents a general and handy approach for understanding the atomic scale dynamics in materials. Mapping the complicated local environments of atoms to a low-dimensional state space can be much easier than analyzing global dynamics in material systems. The method is able to extract relaxation timescale far beyond the simulation time and also reveals a heterogeneous map. The relaxation timescales learned from the well characterized model glassy polymer former by GCN indicates the potential and ability of applying the method into a wider range and variety of material systems. The graph neural network proves to be a powerful, effective, and successful method to map the complicated local environment of particles into a low-dimensional state space, resulting in a coarse-grained

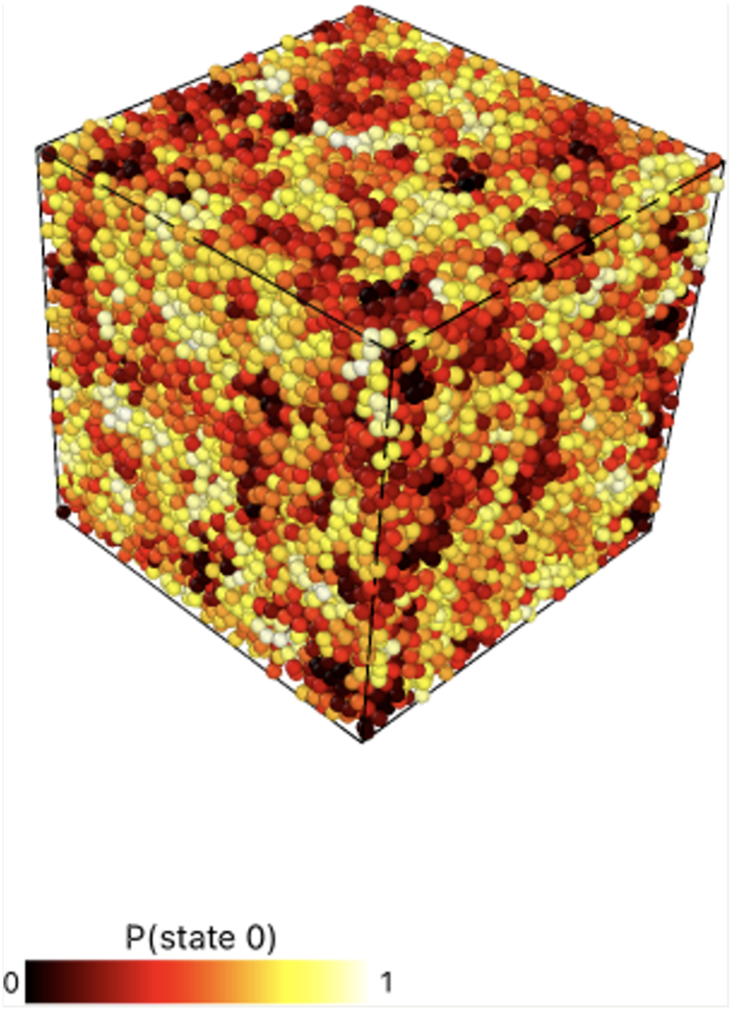

Figure 7. Ovito 3D visualization of probabilities of atoms in state 0

dynamics. Glassy polymer exhibits heterogeneity. Besides what I have done, a lower temperature and a longer trajectory can be simulated and trained to see how the difference would be to the one trained at a higher temperature. From the output of Lammps, I also got the graph of how the total energy of the whole system changes along with the running time when system temperature drops, as shown in figure 10. For having more trained data in a lower temperature, the results trained can be also compared to the lammps output. VAMP/GDyNets can be applied to a variety of problems to understand the complicated dynamics in glassy systems like doing a simulation of a binary mixture polymer to explore local excess Voronoi volumes, as I mentioned in the introduction part.

5. Acknowledgments

Much appreciation to Prof. Joerg Rottler for helping and supervising this project, and Dr. Siavash Soltani Bajestani for helping and providing information in Machine learning. Prof. Joerg Rottler helped me a lot in the several aspects of the MC and software learning.

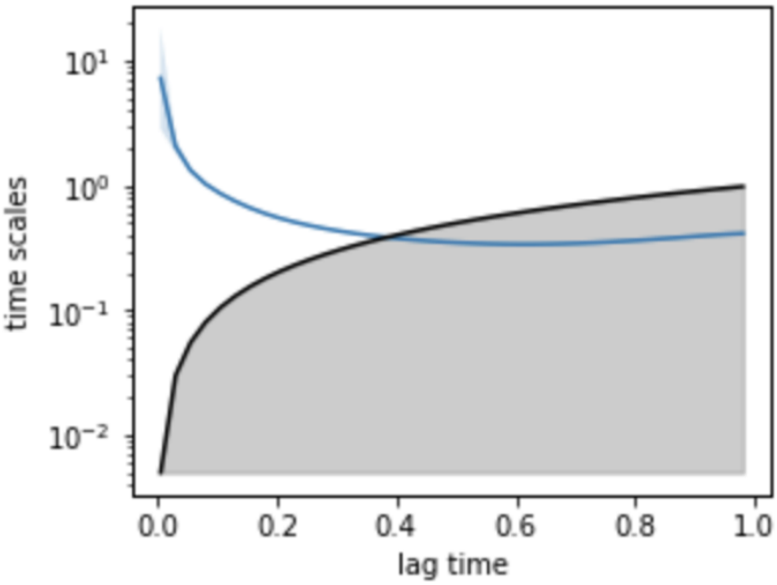

Figure 8. Relaxation timescales change as a function of the lag time

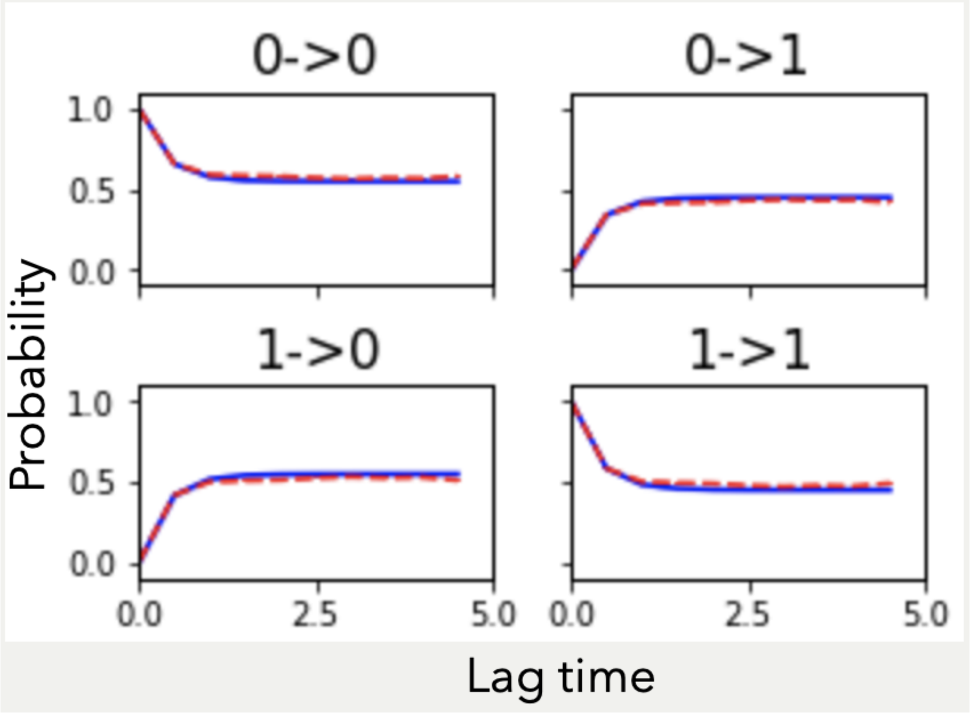

Figure 9. CK Test at $\tau = 0.9$

6. Code reference and availability

Code reference availability: https://github.com/txie-93/gdynet Xie, T., elt, Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials.

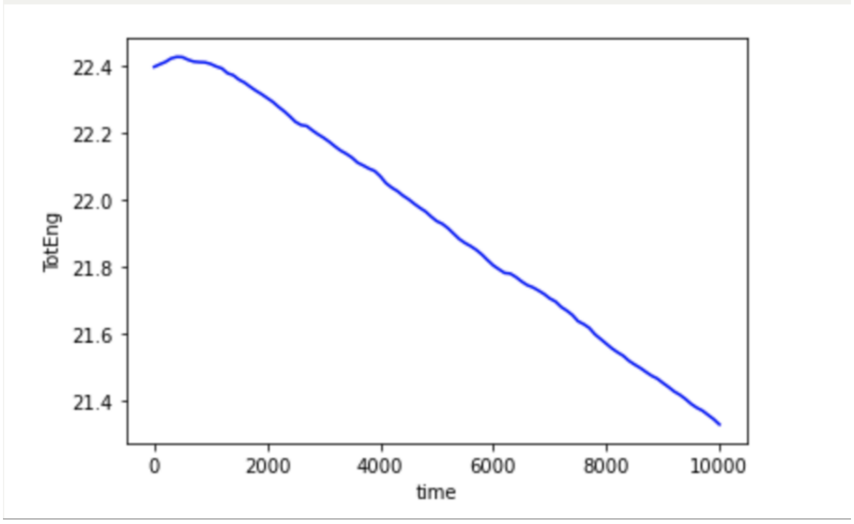

Figure 10. Total energy vs time graph generated in lammps, temperature dropping from T=1 to T=0.5

References

[1] Etacheri V, Marom R, Elazari R, Salitra G, and Aurbach D 2011 *Energy Environ. Sci.* **4** 3243-3262

[2] Xie T, France-Lanord A, Wang Y, Shao-Horn Y, and Grossman J 2019 *NATURE COMMUNICATIONS.***10** 2667

[3] Hu Z, Wang A, and Zhang H 2017 *Modern Inorganic Synthetic Chemistry 2nd* (Shenyang:Elsevier) chapter 22 pp 641-667

[4] Maio E, Lannace S, and Mensitieri G 2014 *Supercritical Fluid Science and Technology* vol 9 (Elsevier) chapter 5 pp 69-177

[5] MIJOVIC J 1994*Polymer Engineering and Science***34**

[6] Soltani S,Sinclair C, and Rottler J 2021*arxiv*

[7] MIJOVIC J 1994*Polymer Engineering and Science***34**