Conditional generation of atomic structures using Recurrent Probabilistic Axis Projections (RePAP)

Bjarke Arnskjær Hastrup[1] and Michael Wamberg[1]. Supervision by Arghya Bhowmik[2] and Peter Bjørn Jørgensen[2]

1 DTU Compute, 2 DTU Energy - Technical University of Denmark

Introduction

Deep generative models have a great potential in molecular science, as they can reduce the usually very high computational costs associated with traditional quantum mechanical calculations [1] [2]. In this project we will focus on the chemical element silicon. What makes silicon particularly interesting to study is its wide use as a semiconductor and the fact that silicon anodes are generally considered to be the next development within lithium-ion battery technologies.

In this work we propose a Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) using Recurrent Probabilistic Axis Projections (RePAP) for learning the structural patterns in physically realistic atomic configurations. To reduce the computational complexity normally encountered when working with a voxel representation of an atomic system, RePAP predicts the location of an atom one axis at a time. This method obviously entails a blatant disregard of Euclidean geometry but turns out to offer drastic advantages in terms of dimensionality and models can be trained and tested at a much faster rate.

The RePAP LSTM is trained on a molecular dynamics trajectory of 40.000 snapshots/frames. After training, the network is then used to quickly generate (in a sequential manner) an entire ensemble of hitherto unseen amorphous silicon structures from the stationary distribution, see Figure 1.

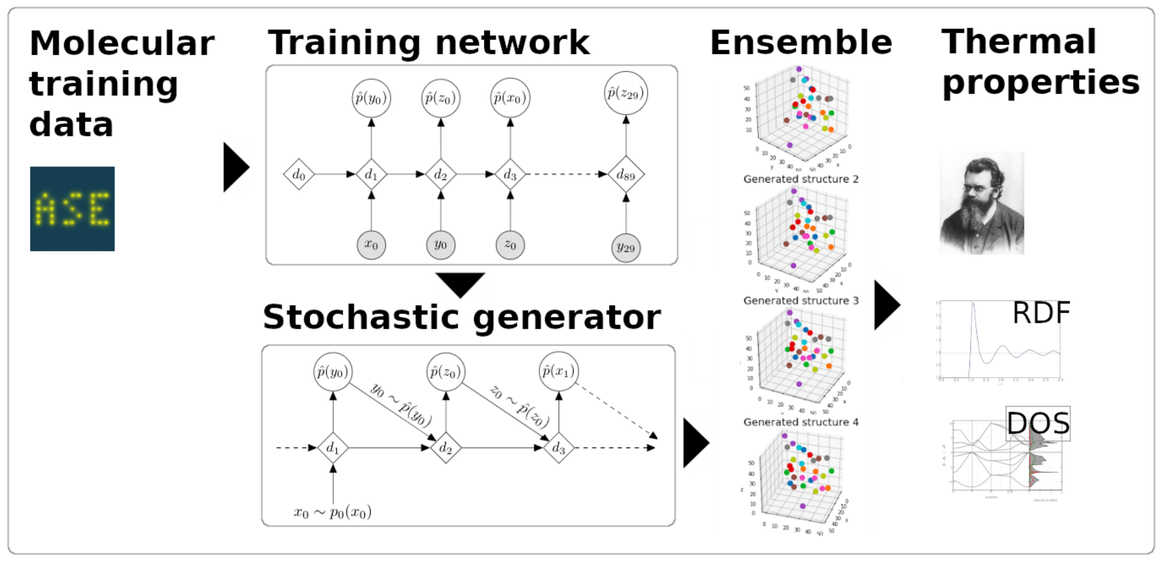

Figure 1: Abstract representation of workflow.

Model specification

RePAP methodology

- The probabilistic axis projection framework consists of representing each atom (in each snapshot of the molecular training trajectory) as a discretized Gaussian density cloud that is projected onto each of the three axes x, y, and z.
- With 30 atoms in total, this corresponds to 90 axis projections (i.e. 90 atom coordinates). The 30 atoms in a snapshot are ordered w.r.t. mean distance to origin over the entire trajectory, from smallest to largest distance. The 90 atom coordinates are then ordered as $x_0, y_0, z_0, x_1, y_1, z_1, ..., x_{29}, y_{29}, z_{29}$. This is the sequential order fed to the recurrent network.
- With a 50-dimensional grid along each axis, the density of the i'th atom coordinate at the k'th element of the grid vector is

$$M_{i,k} = \frac{1}{C}\exp\left(-\frac{(grid_k - r_i)^2}{2\sigma}\right), \tag{1}$$

with σ being the built in uncertainty in the location of the nucleus, and C being a normalization constant.

Prediction task and loss function

- Since we are doing next atom coordinate predictions, the target sequence is simply the input sequence shifted by one atom coordinate, see Figure 2.
- The network outputs a categorical distribution, $\hat{p}(r_i)$, that is compared to the true (discretized) Gaussian density (equation (1)), grid cell by grid cell in a MSE fashion:

$$\mathcal{L} = \sum_{i=1}^{89}\sum_{k=1}^{50}(\hat{p}_k(r_i) - M_{i,k})^2 \tag{2}$$

Input representation

- Input data to the LSTM network consists of one-hot-encoding of the inputted atom coordinate (vector of length 50), concatenated with one-hot-encoding of the atom number (length 89), thus cancelling the order invariance (or "time invariance") of the network.

LSTM network

- The RNN consists of deterministic nodes $d_i = LSTM(d_{i-1}, x_i)$, $i = 1, ..., 89$ representing both the hidden state and the cell/memory state of the LSTM.
- Two layers are used (not shown of figure), each with size 256. Dropout is applied (p=0.5).
- ReLU activation function after top hidden layer, followed by a final feedforward layer (dimension 256⟶50) with a sigmoid transformation.

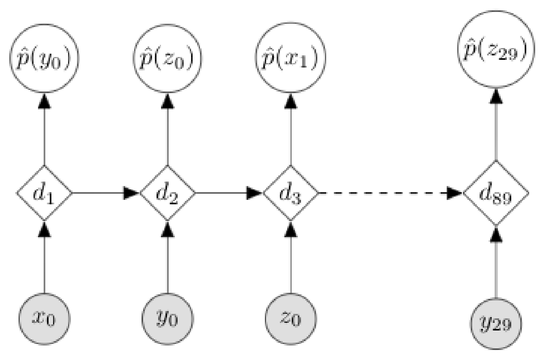

Figure 2: The graphical representation of the recurrent neural network used for training.

Density predictions from trained model

- Categorical distribution has flexibility to output any distribution but outputs nice discretized Gaussians, see Figure 3. (Targets and predictions on a test snapshot shouldn't really be directly evaluated and deviance is absolutely expected.)

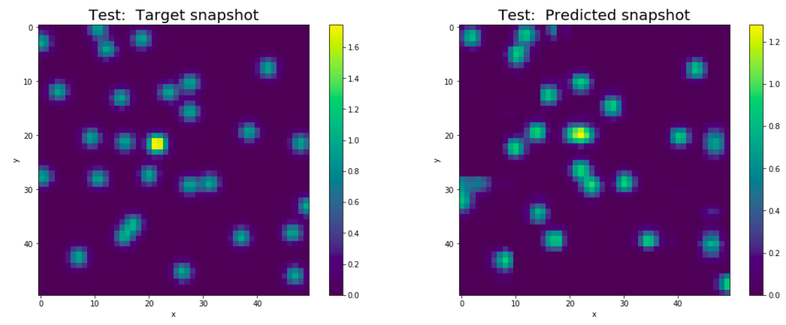

Figure 3: 2d density comparison on a test set.

Probabilistic generation of new configurations

Joint probability of an entire atomic configuration factorizes

$$p(\mathbf{r}_{0:29}) = p(x_0)p(y_0|x_0)p(z_0|x_0,y_0) \times \prod_{i=1}^{29} p(x_i|x_{<i},y_{<i},z_{<i})p(y_i|x_{\leq i},y_{<i},z_{<i})p(z_i|x_{\leq i},y_{\leq i},z_{<i}). \tag{3}$$

Conditional generation procedure

- Draw x_0 from training distribution of the atom with smallest mean distance to the origin.
- One-hot-encode x_0 and evaluate network to output categorical distribution $\hat{p}(y_0|x_0)$.
- Draw $y_0 \sim \hat{p}(y_0|x_0)$.
- One-hot-encode y_0 and evaluate network to output categorical distribution $\hat{p}(z_0|x_0, y_0)$.
- Draw $z_0 \sim \hat{p}(z_0|x_0, y_0)$ ... and so on and so on.

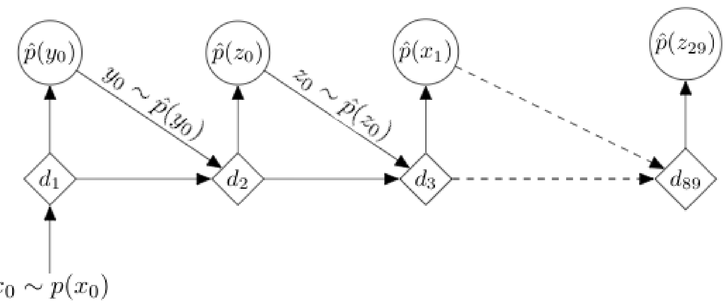

Figure 4: The graphical representation of the probabilistic conditional generator of new structures.

Model performance

As a measure of the quality of the generated samples, we consider the radial distribution function which measures the average number density of atoms at a distance r. The RDF reconstruction error is $Err_{RDF} = \sum_k \frac{1}{k}\left(RDF_k - RDF_k^{true}\right)^2$ and is used to guide model development.

- Our RDF perfectly captures location of peaks in RDF of original construction.
- Some structures have atom pairs with mutual distances $<$ **2** [Å]. These structures will have unrealistically high potential energies due to repulsion.

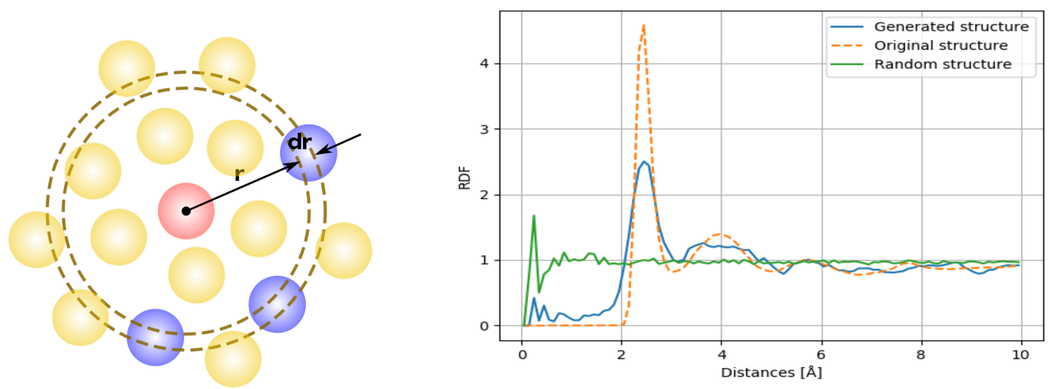

Figure 5: *Left)* A graphical illustration of the radial distribution function. *Right)* Comparison of the radial distribution functions for our generated structures and the original structure, as well as a randomly generated structure.

Summary and outlook

- Increase RDF reconstruction by resampling atom positions when an atom is placed unrealistically close to a previously placed atom and introduce a metric that keeps track of the extend to which atoms are replaced.
- Investigate ability to generalize to trajectories outside training data.
- Increase precision parameters such as coarseness of grid.
- Implement more advanced networks.

References

[1] N. W. A. Gebauer, M. Gastegger, and K. T. Schutt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. 2019.

[2] J. Hoffmann, L. Maestrati, Y. Sawada, J. Tang, J. Sellier, and Y. Bengio. Data-driven approach to encoding and decoding 3-d crystal structures. 09 2019.

{s151602, s146791}@student.dtu.dk