Korean J. Chem. Eng., 38(2), 406-410 (2021)
DOI: 10.1007/s11814-020-0704-1

pISSN: 0256-1115
eISSN: 1975-7220

Deep learning-based initial guess for minimum energy path calculations

Hyunsoo Park, Sangwon Lee, and Jihan Kim†

Department of Chemical and Biomolecular Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291, Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

(*Received 6 September 2020 • Revised 27 October 2020 • Accepted 2 November 2020*)

Abstract−An autoencoder that automatically generates an initial guess for the minimum energy pathway (MEP) calculations has been designed. Specifically, our autoencoder takes in the trajectories of molecular dynamics simulations as its input and facilitates the generation of feasible molecular coordinates. Two molecules (acetonitrile and alanine dipeptide) were tested using the nudged elastic band calculations and the results provided improvements over linear interpolation and image dependent pair potential methods in terms of the number of SCF iterations, demonstrating the utility of using an autoencoder type of an approach for MEP calculations.

Keywords: Deep Learning, Autoencoder, MEP, NEB, Reaction Path

INTRODUCTION

A computational approach to find the minimum energy pathway (MEP) plays an important role in elucidating various chemical reactions [1,2]. The MEP provides information regarding reaction rates, mechanism and transition states, while the highest energy point along the MEP provides an estimate of the transition state, which allows calculation of the transition rate within the harmonic transition state theory (TST) [3]. The chain-of-state methods [4,5] compute the optimal MEP between reactants and products along the potential energy surface (PES). Among different methods [6,7], the widely used nudged elastic band (NEB) method [8-10] computes the energy path with each image in the chain-of-states optimized by spring force to maintain equal spacing along the reaction coordinates. For the NEB, initial guess of MEP is needed as input and the accuracy of the algorithm depends heavily on the accuracy of the initial guess [11,12].

Among many different methods for the initial guess of NEB, linear interpolation (LI) method is the most widely used, in which the initial guess of LI is made by interpolating between the Cartesian coordinates of the initial and final configurations. However, the LI method does not consider interaction potential energy between the atoms of the interpolated images, which often leads to energetically unstable configurations. To overcome this issue, the linear square transit (LST) method [13,14] was developed to minimize the difference between the linear interpolation of the Cartesian coordinates and the interpolated inter-atomic distances by means of a weighted least-squares procedure. Although the initial guess of the LST method can prevent the atomic overlap problem from the LI method, there might still be a discontinuous path due to the synchronous minimization. Finally, the image-dependent pair potential (IDPP) method [15] refines the LST pathways as its objective function is minimized using the NEB optimizers. That is, the spring force used in the process of NEB optimization ensures the IDPP pathways are more continuous. However, the method does not consider the chemical information of the molecules such as pairwise potentials, which can potentially lead to distortions in the interpolated images.

To remedy some of these issues, we propose a deep learning based method that can automatically generate the initial guess for the MEP. It is conceivable that if we can obtain a large data set of the trajectories from classical molecular dynamics (MD) simulations of the molecules in question, a neural network can learn from this dataset to generate appropriate initial guess for the MEP. It turns out that previous works have utilized the trajectory of MD as an initial guess of MEP to search for the TS [16-18]. These MD trajectories include various snapshots created by statistical thermodynamics and, as such, they can be used to make the initial guess that leads to more energetically feasible pathways. However, the previous MD-based methods can use the trajectory as the input of NEB only when the trajectory contains the desired configurations (or final configurations) [19]. On the other hand, because our method generates the initial guess of MEP when the intial and final molecular configurations are given, the previous MD-based methods are not applicable for this case.

Our approach was inspired by various deep-learning methods that were used to generate interpolation of two different images (e.g., interpolating two different people's facial images) [20-22]. An autoencoder [23-25] is a type of artificial neural network (ANN) applied for dimensionality reduction, data denosing, and manifold learning in an unsupervised manner. The autoencoder stores its inputs in the encoding vector of the latent space and is trained to learn condensed representation to reconstruct the inputs. The advantage of the latent space is that its representation has the advantage of being interpolated according to specific features, because the latent space contains a dense representation of the inputs. In this work, the two facial images are analogously mapped to the initial and final molecular configurations in the initial guess of MEP (see

†To whom correspondence should be addressed.
E-mail: jihankim@kaist.ac.kr

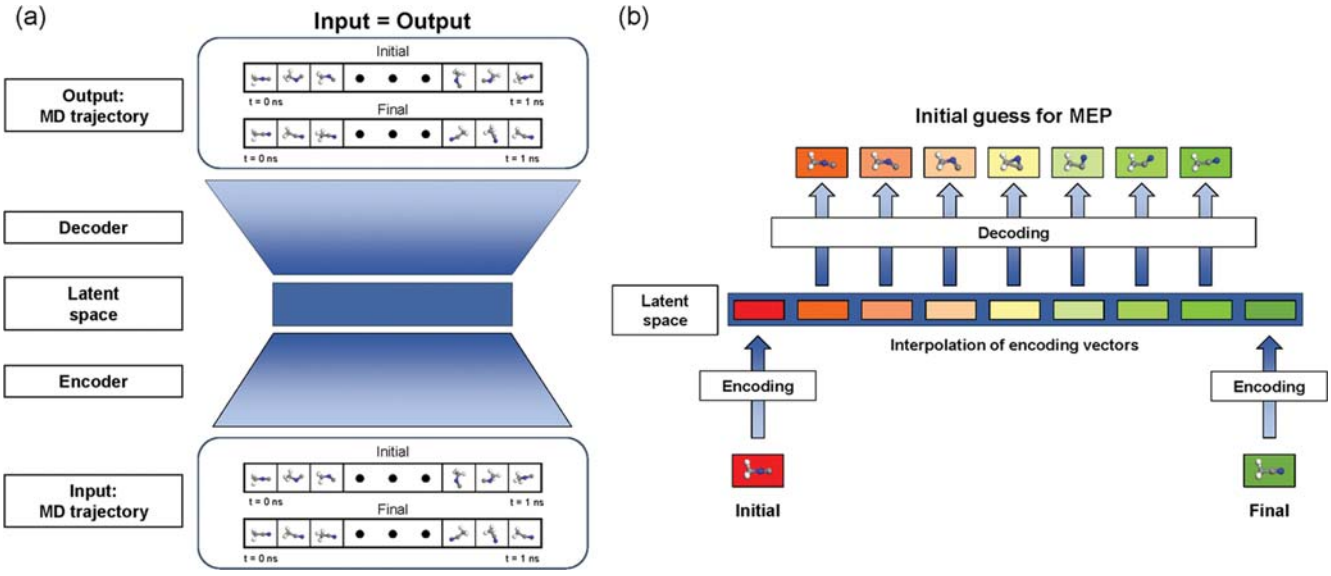

Fig. 1. Overall schematic of the AE used in this work. (a) The AE that learns to reconstruct the MD trajectories of initial and final molecular configurations. (b) A procedure providing MEPs for initial and final molecular configurations by interpolating within latent space of the trained AE.

Fig. 1, for example, on initial and final guesses of a molecule). As such, one can in practice obtain a large dataset of MD simulation trajectories and train the neural network to provide interpolated images of the molecular configurations that would be served as the initial guess.

METHODS

Fig. 1 illustrates the overall schematic of our autoencoder (AE) that reconstructs the trajectories of the MD simulations and provides the initial guess for the MEP. Fig. 1(a) shows that the MD trajectories of the intial and final configurations are used as training set inputs of the AE. As such, the latent space of the AE contains dense information regarding the energetically feasible (based on classical molecular simulations in MD) molecular configurations. After optimizing the neural network, one can use AE to linearly interpolate the encoding vectors of the initial and final molecular configurations and then subsequently decode them to generate the interpolated path of the initial and final molecular configuraitons, which can be used as initial guess for the MEP (Fig. 1(b)).

The AE was trained by taking the Cartesian coordinates of the molecule's snapshots from the MD trajectories starting from the optimized initial and final molecular configurations. After completion of training, the encoder of the AE generated two encoding vectors that correspond to the initial and final configurations in order to generate the interpolated path. These two encoding vectors were interpolated with a ratio, $\alpha=(M+1)^{-1}$, where M is the user-desired number of images for the NEB calculation. Finally, the interpolated encoding vectors were decoded into the molecular structure, and as such can be used as inputs of the NEB calculations (see Fig. 1(b)).

To generate the training data for the AE, MD simulations were conducted with the LAMMPS software package [26,27] and the universal force field (UFF) [28] was used for all MD simulations. The 100,000 snapshots of MD trajectories of the initial and final molecular configurations were collected every 10 fs for a 1 ns total simulation time, respectively. All MD simulations were conducted at a high temperature of 1,000 K to ensure the vast exploration of possible trajectories and configurations. In many of the NEB cases, there is a significant barrier that separates the phase space of the initial and final molecular configurations and, as such, high temperature can provide the needed energy to explore these intermediate configurations that can be used to train the neural network. We also introduced the resampling of the snapshots of MD trajectory, which ensured that the inputs of autoencoder are not biased in a specific range of potential energies (see Fig. S2).

Within the AE, the input $\mathbf{x} \in \mathbb{R}^{d_x}$ is transformed into the $\mathbf{z} \in \mathbb{R}^{d_z}$ in the latent space by an encoding function, $f(\mathbf{x})=\mathbf{z}$. Then, z is transformed into the output $\hat{\mathbf{x}} \in \mathbb{R}^{d_x}$ by the decoding function, $g(\mathbf{z})=\hat{\mathbf{x}}$. To ensure the proper functioning of the AE as well as to provide optimized guesses for the NEB, three loss functions were introduced. The first loss function is a reconstruction loss function used to minimize which the difference between the input and the output vectors, which can be written as

$$L_{RC}=\sqrt{\frac{1}{3N}\sum_{i=1}^{N}\|\mathbf{x}-\hat{\mathbf{x}}\|^2} \quad (1)$$

where L_{RC} is the root mean square error (RMSE) calculated by the differences between the Catesian coordinate of $\mathbf{x}$ and $\hat{\mathbf{x}}$ for N atoms. Because $\mathbf{x}$ only contains the information of the Cartesian coordinate of the MD trajectories, the AE is trained by focusing more on the actual Cartesian coordinate as opposed to the atomic distance of all snapshots of the MD trajectory. Therefore, when interpolating between the encoding vectors of the initial and final molecular configurations and reconstructing using decoder, it will generate the interpolated path, which is simply obtained via linear

interpolating the Cartesian coordinate similar to the conventional LI method.

As mentioned, the LI method can unfortunately create high energetic molecular configurations due to the atomic distances not being properly taken into account. This problem can be overcome by taking into account the pairwise inter-atomic distances, which is put into the second loss function for distance matrix as follows:

$$L_{distance} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|\mathbf{d}_{ij} - \hat{\mathbf{d}}_{ij}\right| \quad (2)$$

where $\mathbf{d}_{ij}$ and $\hat{\mathbf{d}}_{ij}$ are distance matrices with size N×N for $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. $L_{distance}$ is a mean absolute error of distance matrices of $\mathbf{x}$ and $\hat{\mathbf{x}}$. The AE is trained to learn the proper pairwise interatomic distances of the feasible molecular configurations from the snapshots of the MD trajectories.

Although $L_{distance}$ creates a refined interpolated path by considering the pairwise inter-atomic distances, we have observed that atomic overlaps occur in certain cases. Because the atomic overlaps exert significant impact on the performance of the NEB calculations, an additional loss function was used to explicitly take into account these overlaps, which is written as

$$L_{overlap} = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\min(\tilde{\mathbf{d}}_{ij}, C) - C\}^2} \quad (3)$$

where $L_{overlap}$ imposes a heavy penalty on generating molecules with very small inter-atomic distances. To implement this, two inputs are selected randomly from a minibatch. Then, encoding vectors, $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{d_z}$ of the selected inputs are created by the encoder of the AE. And the vectors are interpolated with random ratio α, such that random interpolated encoding vectors, $\tilde{\mathbf{z}} = \alpha\mathbf{z}_1 + (1-\alpha)\mathbf{z}_2$ of size 3N are generated where $\alpha \in [0, 1]$. Then, by using $\tilde{\mathbf{z}}$ as the input of the decoder of the AE, randomly interpolated molecular configurations are generated and the distance matrices $\tilde{\mathbf{d}}_{ij}$ of random interpolated molecular configurations are computed. All elements of calculated distance matrices are compared with distance cutoff, C (around 0.8 to 1.0 Å), and loss is only computed if the values are smaller than the cutoff. Then, the RMSE is calculated between $\min(\tilde{\mathbf{d}}_{ij}, C)$ and C for all entries of the distance matrix.

Molecular configurations of the initial and final molecular configurations were optimized geometrically using the density functional theory (DFT) method. The DFT calculations for NEB were performed with the Vienna ab initio simulation package (VASP) [29-31] with the projector augmented wave (PAW) [32] potentials. The NEB method was implemented in the VASP transition state theory (VTST) code in VASP package. The Perdue-Burke-Ernzerhof (PBE) [33] was utilized for exchange-correlation functional. For all DFT simulations, unit cell of 30×30×30 Å was used with Γ-centered 2×2×2 k-mesh. FIRE [34] optimizer was used for NEB optimizer with 15 images. NEB was conducted with a maximum force criterion of 0.5 eV $Å^{-1}$ initially, and climbing-image nudged elastic band (CINEB) [35] was conducted with a tighter criterion of 0.1 eV $Å^{-1}$.

RESULTS AND DISCUSSION

To test our AE method, two test case molecules were selected: acetonitrile and alanine dipeptide. The initial guesses of LI, IDPP, and AE methods were used to compare the performance via the number of SCF iterations for NEB and CINEB calculations. For LI and IDPP methods, the initial guess of NEB was made using the ASE package [36].

The first example is the rearrangement of acetonitrile as shown in Fig. 2(a). Table 1 indicates that the LI method required the largest number of SCF iterations (i.e., 457) for convergence. Because the angle of the C-H-O backbone in initial and final molecular configurations is 180°, the interpolation path of LI method leads to slow convergence when calculating NEB due to atomic overlap. On the other hand, although the interpolated path of IDPP method avoids

Table 1. The total number of SCF iterations in the NEB and CINEB calculations required to obtain convergence using the DFT calculation

	The total number of SCF interation in the NEB and CINEB calculations		
	LI	IDPP	AE
Acetonitrile	457	252	76
Alanine dipeptide	-	130	98

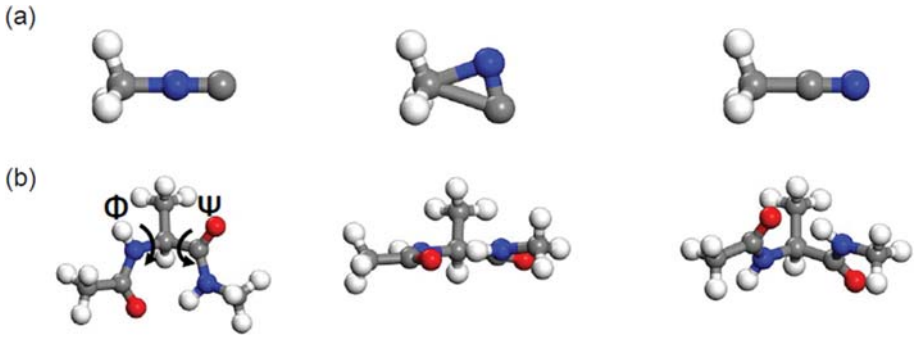

Fig. 2. Two applications of AE method: (a) The rearrangement of acetonitrile and (b) Isomerization of alanine dipeptide. The first and third columns contain snapshots of the two local minima of MEP, whereas the second column is the optimized transition state configuration computed using the initial guess of AE as input for NEB. LI, IDPP and AE have almost same energy and molecular configuration for the transition states calculated by NEB.

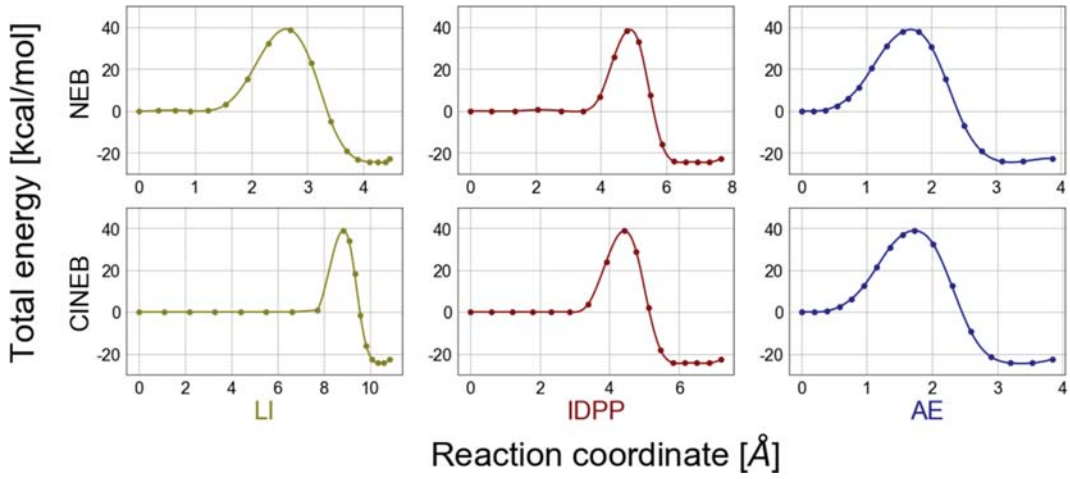

Fig. 3. (top) The converged NEB energy profile of acetonitrile using LI, IDPP, and AE method with a maximum force criterion of 0.5 eV Å^{-1} (bottom). The converged CINEB energy profile computed from the NEB results (top), respectively, with a maximum force criterion of 0.1 eV Å^{-1}.

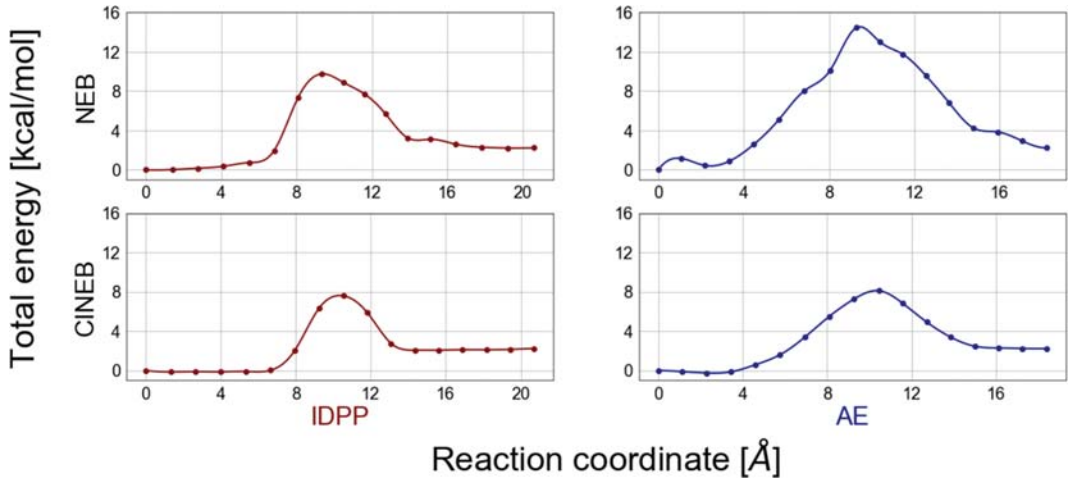

Fig. 4. (top) The converged NEB energy profile of alanine dipeptide using LI, IDPP, and AE method with a maximum force criterion of 0.5 eV Å^{-1} (bottom). The converged CINEB energy profile computed from the NEB results (top), respectively with a maximum force criterion of 0.1 eV Å^{-1}.

the atomic overlap (see Fig. S4), so larger number of optimization steps are required compared to the AE method (252 versus 76) as shown in Table 1. Fig. 3 indicates the results of the MEP computed by the NEB and CINEB for the arrangement of acetonitrile. It turns out that the change in the potential energy profile from neighboring points near the transition state region is larger in IDPP compared to that of the AE and the result of the AE shows a more smooth and gradually changing energy profile with respect to the reaction coordinate. This energy profile contributes to its relatively fast convergence of NEB calculations. Because the AE is trained using the trajectories of the MD simulations, it tends to provide more smooth transition points between neighboring image points. This tendency is also maintained in the CINEB results.

The second example is the isomerization of the alanine dipeptide, which is the basic element of the polypeptide backbone in Fig. 2(b) and is a popular system, as many studies for conformational isomerization of the alanine dipeptide have been reported [37-39]. Two dihedral angles Φ, Ψ, marked in the first snapshot in Fig. 2(b), make representative conformational isomers $C7_{eq}$ (the first snapshot in Fig. 2(b)) and $C7_{ax}$ (the third snapshot in Fig. 2(b)). The dihedral angles Φ, Ψ of $C7_{eq}$ are -77° and 87° and those of $C7_{ax}$ are 61° and -70°. From Table 1, we see that LI method failed to converge for the alanine dipeptide due to the atomic overlap of interpolated images in the initial guess. Fig. 4 shows the results of MEP calculated by the NEB and CINEB for the isomerization of the alanine dipeptide. Although the calculated energy barrier from the NEB for AE (14.5 kcal mol^{-1}) is higher than that of IDPP (9.8 kcal mol^{-1}), the energy profile of from the AE method shows smooth and gradual changes, similar to that of acetonitrile. The converged energy profile from CINEB in Fig. 4 also shows a more energetically continuous path for the AE method, which is a result of the improved initial guess. All python codes and results of NEB can be downloaded from https://github.com/schoonbroot/Autoencoder_Initialguess.

CONCLUSION

We successfully trained an autoencoder to learn the information regarding the "natural motion" of molecules using the trajectories of the classical MD simulations via unsupervised learning. As such, this information can be utilized to create a reasonable initial guess for the MEP calculations. To facilitate convergence and for efficiency purposes, three loss functions were introduced for reconstruction, distance matrix and atomic overlap. Our method was tested against other LI and IDPP methods and showed improved performance for the case study of acetonitrile and alanine dipeptide. It remains to be seen how much of this method can extend to more complicated systems and this will serve as future work.

ACKNOWLEDGEMENT

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-MA1702-07.

SUPPORTING INFORMATION

Additional information as noted in the text. This information is available via the Internet at http://www.springer.com/chemistry/journal/11814.

REFERENCES

1. K. Fukui, *Acc. Chem. Res.*, **14**, 363 (1981).
2. H. B. Schlegel, *J. Comput. Chem.*, **24**, 1514 (2003).
3. K. J. Laidler and M. C. King, *J. Phys. Chem.*, **87**, 2657 (1983).
4. L. R. Pratt, *J. Chem. Phys.*, **85**, 5045 (1986).
5. R. Elber and M. Karplus, *Chem. Phys. Lett.*, **139**, 375 (1987).
6. E. Weinan, W. Ren and E. Vanden-Eijnden, *Phys. Rev. B - Condens. Matter Mater. Phys.*, **66**, 523011 (2002).
7. E. Weinan, W. Ren and E. Vanden-Eijnden, *J. Chem. Phys.*, **126**, 164103 (2007).
8. H. Jonsson, G. Mills and K. W. Jacobsen, in *Classical and quantum dynamics in condensed phase simulations: Proceedings of the international school of physics*, B. J. Berne, G. Ciccotti and D. F. Coker, World Scientific, Singapore (1998).
9. D. Sheppard, P. Xiao, W. Chemelewski, D. D. Johnson and G. Henkelman, *J. Chem. Phys.*, **136**, 074103 (2012).
10. G. Henkelman and H. Jónsson, *J. Chem. Phys.*, **113**, 9978 (2000).
11. D. Sheppard, R. Trrell, and G. Henkelman, *J. Chem. Phys.*, **128**, 1 (2008).
12. H. C. Herbol, J. Stevenson and P. Clancy, *J. Chem. Theory Comput.*, **13**, 3250 (2017).
13. L. R. Raber, *Chem. Eng. News*, **75**, 39 (1997).
14. N. Govind, M. Petersen, G. Fitzgerald, D. King-Smith and J. Andzelm, *Comput. Mater. Sci.*, **28**, 250 (2003).
15. S. Smidstrup, A. Pedersen, K. Stokbro and H. Jonsson, *J. Chem. Phys.*, **140**, 214106 (2014).
16. E. Martínez-Núñez, *J. Comput. Chem.*, **36**, 222 (2015).
17. L. P. Wang, A. Titov, R. McGibbon, F. Liu, V. S. Pande and T. J. Martínez, *Nat. Chem.*, **6**, 1044 (2014).
18. L. P. Wang, R. T. McGibbon, V. S. Pande and T. J. Martinez, *J. Chem. Theory Comput.*, **12**, 638 (2016).
19. A. L. Dewyer, A. J. Argüelles and P. M. Zimmerman, *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **8**, 1 (2018).
20. X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever and P. Abbeel, *Adv. Neural Inf. Process. Syst.*, 2180 (2016).
21. P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala and K. Weinberger, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* 2017-Janua, 6090 (2017).
22. D. Berthelot, I. Goodfellow, C. Raffel and A. Roy, *7th Int. Conf. Learn. Represent. ICLR 2019* (2019).
23. M. A. Kramer, *AIChE J.*, **37**, 233 (1991).
24. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P. A. Manzagol, *J. Mach. Learn. Res.*, **11**, 3371 (2010).
25. D. P. Kingma and M. Welling, *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, **1** (2014).
26. S. Plimpton, *J. Comput. Phys.*, **117**, 1 (1997).
27. S. Plimpton and A. P. Thomson, *MRS Bulletin*, **37**, 513 (2012).
28. A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard and W. M. Skiff, *J. Am. Chem. Soc.*, **114**, 10024 (1992).
29. G. Kresse and J. Hafner, *Phys. Rev. B*, **48**, 13115 (1993).
30. G. Kresse, J. Furthmüller and J. Hafner, *Phys. Rev. B*, **50**, 13181 (1994).
31. G. Kresse and J. Furthmuller, *Phys. Rev. B*, **54**, 11169 (1996).
32. P. E. Blöchl, *Phys. Rev. B*, **50**, 17953 (1994).
33. J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, **77**, 3865 (1996).
34. E. Bitzek, P. Koskinen, F. Gähler, M. Moseler and P. Gumbsch, *Phys. Rev. Lett.*, **97**, 1 (2006).
35. G. Henkelman, B. P. Uberuaga and H. Jónsson, *J. Chem. Phys.*, **113**, 9901 (2000).
36. A. Larsen, J. J. Mortensen, J. Blomqvist, I. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer and C. Hargus, *J. Phys. Condens. Matter*, **29**, 273002 (2017).
37. W. Ren, E. Vanden-Eijnden, P. Maragakis and E. Weinan, *J. Chem. Phys.*, **123**, 134109 (2005).
38. A. Allouche, *J. Comput. Chem.*, **32**, 174 (2012).
39. P. G. Bolhuis, C. Dellago and D. Chandler, *Proc. Natl. Acad. Sci. U.S.A.*, **97**, 5877 (2000).

Supporting Information

Deep learning-based initial guess for minimum energy path calculations

Hyunsoo Park, Sangwon Lee, and Jihan Kim†

Department of Chemical and Biomolecular Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291, Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

(*Received 6 September 2020 • Revised 27 October 2020 • Accepted 2 November 2020*)

1. The Architecture of Autoencoder

The dimension of the inputs is [N, 3], where N represents the total number of atoms for the given molecule, and 3 refers to the number of spatial dimensions. Fig. S1 illustrates an architecture of the autoencoder, which consists of three components: an encoder, a decoder, and an encoding vector in the latent space. The encoder and decoder consist of three dense layers, and the input was flattened prior to being passed onto the encoding layers so that the size of the new input becomes N. The number of unit of encoding layers and decoding layers is dependent on the number of atoms. The number of unit of final encoding layer and the first decoding layer was set to the exponent of 2 (2^m) where m is greater than the number of atoms. For example, encoding and decoding layers consist of 2^5, 2^6, and 2^7 because there are 22 atoms for alanine dipeptide. For training, the batch size was set to 100 and ReLU [1] activation was applied to all dense layers. The autoencoder was implemented using Keras [2] and Tensorflow [3].

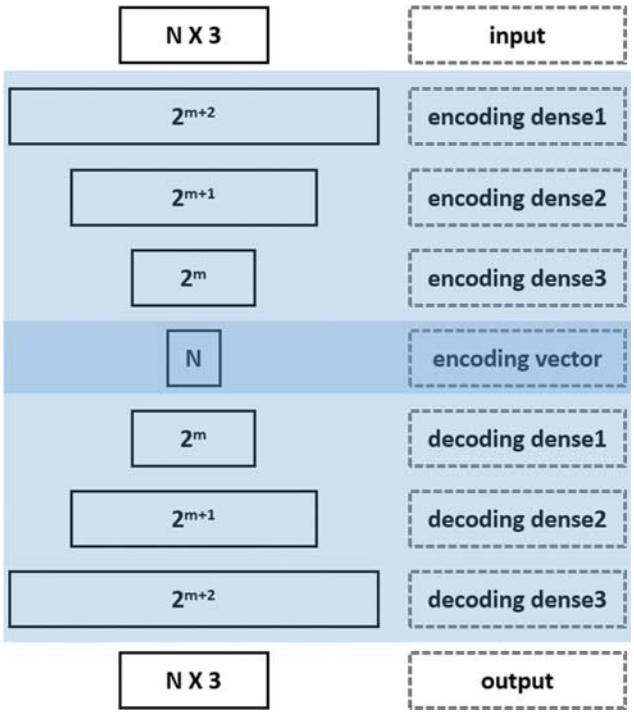

Fig. S1. Scheme for an architecture of the autoencoder used in this work.

2. Resampling of the MD Trajectory

In running the MD simulations and plotting the potential energy distribution of for the alanine dipeptide molecule in Fig. 2(b), one can see that distribution is non-uniform across the range of energies (e.g. peak can be seen at around 80 kcal mol^{-1}) as shown in Fig. S2(a). While this is normal for most molecules, it can become problematic in training the autoencoder. To overcome this drawback, a resampling strategy was imposed where samples with relatively low frequency number of snapshots possess a higher probability of being chosen in the new set. The modified distribution is shown

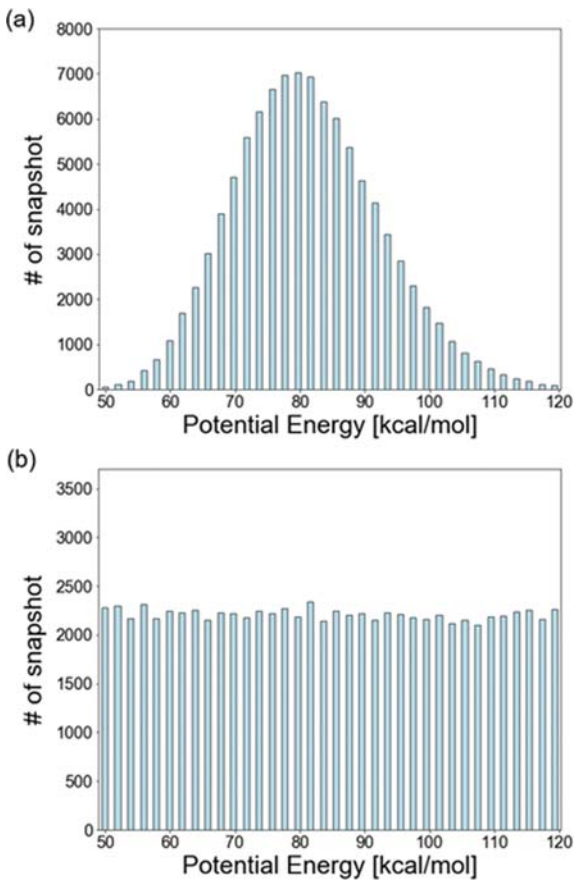

Fig. S2. Histogram probability distributions of MD trajectories with respect to potential energy for the alanine dipeptide. (a) is the probability distribution before resampling and (b) is the probability distribution after resampling.

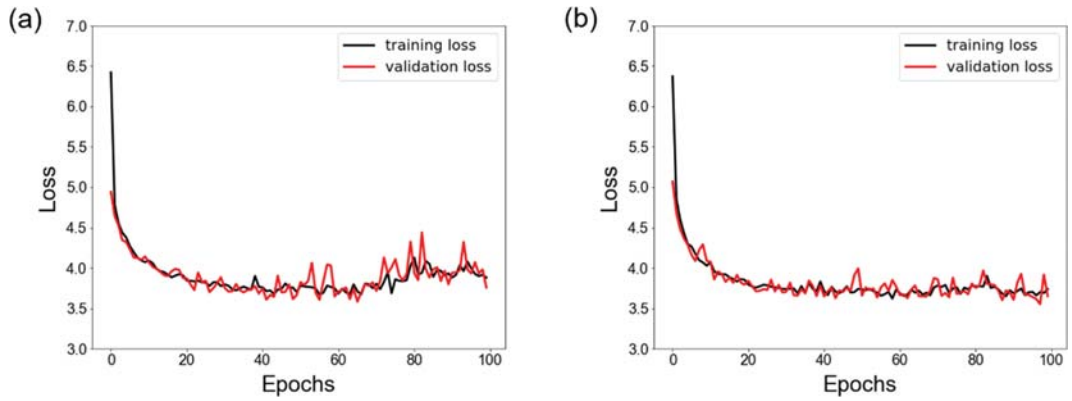

Fig. S3. Graphs of training loss (black line) and validation loss (red line) for 100 epochs: (a) is before resampling and (b) is after resampling.

in Fig. S2(b), where one can see that the distribution is more uniform across all range of energy values. Similarly, the 100,000 snapshots of the final states of alanine dipeptide were also resampled. With the newly sampled 200,000 snapshots, 150,000 snapshots were used as training and 50,000 snapshots as test data sets. Fig. S3 indicates graphs of training loss and validation loss for 100 epochs. While resampling data have the converged loss values for training and validation dataset over epochs in Fig. S3(b), the training and validation loss starts to rise at around epoch 70 without resampling in Fig. S3(a). It indicates resampling helps the autoencoder to be stably trained without being biased.

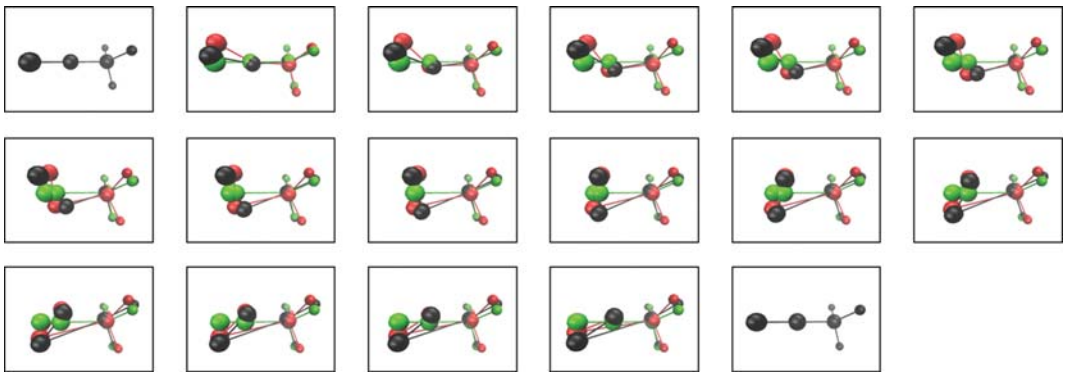

Fig. S4. Initial guess of MEP for rearrangement of acetonitrile: LI (green), IDPP (red) and AE (black). The initial guesses, which used for inputs of NEB, consist of 15 images except two local minima (first and final images).

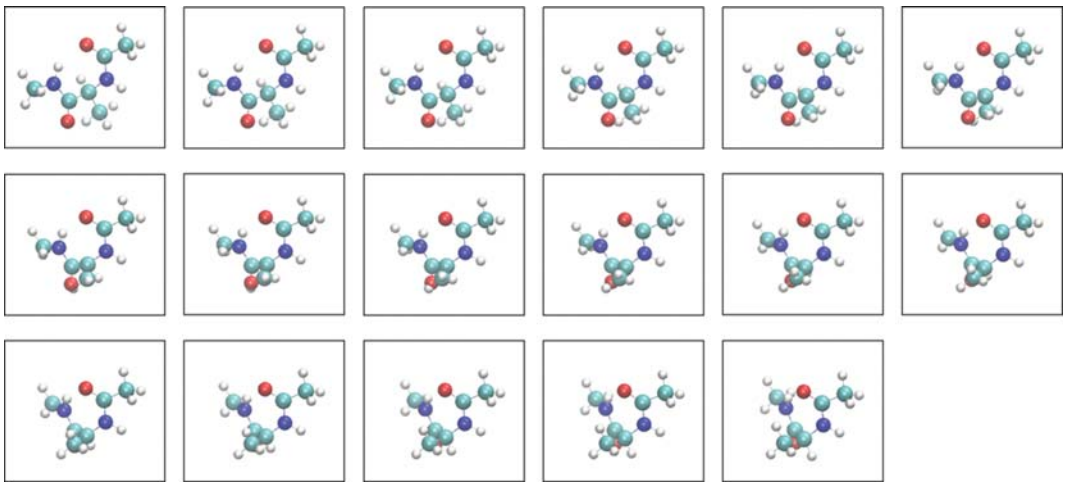

Fig. S5. Initial guess of MEP for rearrangement of alanine dipeptide of LI method. The initial guesses, which used for inputs of NEB, consist of 15 images except two local minima (first and final images).

3. Initial Guess of MEP of LI, IDPP and AE Methods for Two Test Cases: Acetonitrile, Alanine Dipeptide

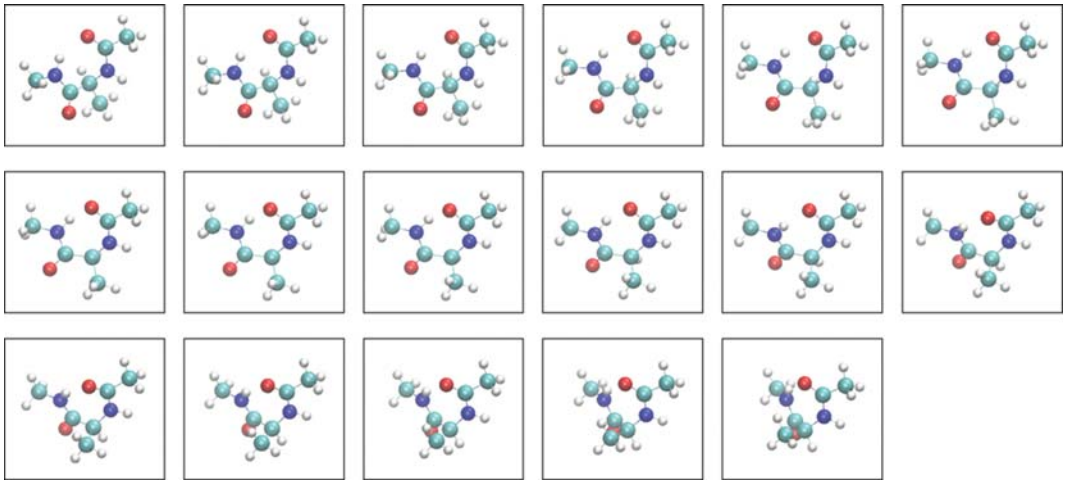

Fig. S6. Initial guess of MEP for rearrangement of alanine dipeptide of IDPP method. The initial guesses, which used for inputs of NEB, consist of 15 images except two local minima (first and final images).

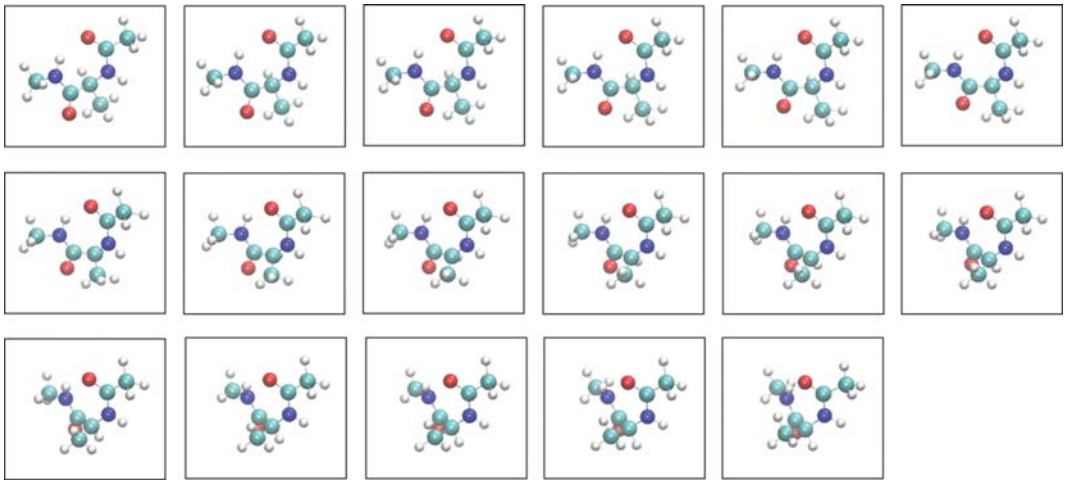

Fig. S7. Initial guess of MEP for rearrangement of alanine dipeptide of AE method. The initial guesses, which used for inputs of NEB, consist of 15 images except two local minima (first and final images).

REFERENCES

1. V. Nair and G. E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning* (2010).
2. F. Chollet, Keras: The python deep learning library, *ascl*, ascl: 1806.022 (2018).
3. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016* (2016).