An extended autoencoder model for reaction coordinate discovery in rare event molecular dynamics datasets

Frassek, M.; Arjun, A.; Bolhuis, P.G.

DOI
10.1063/5.0058639

Publication date
2021

Document Version
Final published version

Published in
Journal of Chemical Physics

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):
An extended autoencoder model for reaction coordinate discovery in rare event molecular dynamics datasets

Cite as: J. Chem. Phys. 155, 064103 (2021); https://doi.org/10.1063/5.0058639
Submitted: 01 June 2021 • Accepted: 15 July 2021 • Published Online: 09 August 2021

M. Frassek, A. Arjun and P. G. Bolhuis

COLLECTIONS

Paper published as part of the special topic on JCP Editors’ Choice 2021
This paper was selected as an Editor’s Pick

ARTICLES YOU MAY BE INTERESTED IN

Toward the design of chemical reactions: Machine learning barriers of competing mechanisms in reactant space
The Journal of Chemical Physics 155, 064105 (2021); https://doi.org/10.1063/5.0059742

Quantum-electrodynamical time-dependent density functional theory within Gaussian atomic basis
The Journal of Chemical Physics 155, 064107 (2021); https://doi.org/10.1063/5.0057542

Machine learning implicit solvation for molecular dynamics
The Journal of Chemical Physics 155, 084101 (2021); https://doi.org/10.1063/5.0059915
An extended autoencoder model for reaction coordinate discovery in rare event molecular dynamics datasets

Cite as: J. Chem. Phys. 155, 064103 (2021); doi: 10.1063/5.0058639
Submitted: 1 June 2021 • Accepted: 15 July 2021 •
Published Online: 9 August 2021

M. Frassek, A. Arjun, and P. G. Bolhuis

AFFILIATIONS
van’t Hoff Institute for Molecular Sciences, University of Amsterdam, P.O. Box 94157, 1090 GD Amsterdam, The Netherlands

a) Author to whom correspondence should be addressed: p.g.bolhuis@uva.nl

ABSTRACT

The reaction coordinate (RC) is the principal collective variable or feature that determines the progress along an activated or reactive process. In a molecular simulation using enhanced sampling, a good description of the RC is crucial for generating sufficient statistics. Moreover, the RC provides invaluable atomistic insight into the process under study. The optimal RC is the committor, which represents the likelihood of a system to evolve toward a given state based on the coordinates of all its particles. As the interpretability of such a high dimensional function is low, a more practical approach is to describe the RC by some low-dimensional molecular collective variables or order parameters. While several methods can perform this dimensionality reduction, they usually require a preselection of these low-dimension collective variables (CVs). Here, we propose to automate this dimensionality reduction using an extended autoencoder, which maps the input (many CVs) onto a lower-dimensional latent space, which is subsequently used for the reconstruction of the input as well as the prediction of the committor function. As a consequence, the latent space is optimized for both reconstruction and committor prediction and is likely to yield the best non-linear low-dimensional representation of the committor. We test our extended autoencoder model on simple but nontrivial toy systems, as well as extensive molecular simulation data of methane hydrate nucleation. The extended autoencoder model can effectively extract the underlying mechanism of a reaction, make reliable predictions about the committor of a given configuration, and potentially even generate new paths representative for a reaction.

Published under an exclusive license by AIP Publishing. https://doi.org/10.1063/5.0058639

I. INTRODUCTION

Molecular dynamics (MD) simulation provides a powerful tool to gain mechanistic, structural, and dynamical information of activated processes in complex molecular systems, such as chemical reaction, biomolecular isomerization, and phase transitions. However, even using classical force fields, MD is limited to relatively small systems (∼10^{5} atoms) and short timescales (ms). While brute force MD as such is not capable of assessing rare events occurring on longer timescales, these timescales are often connected to high (free) energy barriers, which can be accessed employing enhanced sampling methods, e.g., umbrella sampling, metadynamics, local elevation, and many others. These approaches all require the definition of a (set of) collective variable(s) [CV(s)] describing the process and capability of driving the system reversibly from the reactive to the product state. In most enhanced simulation studies, a (set of) CV(s) is preselected a priori, consisting of functions of the coordinates of a (subset of) all particles. However, as the space of possibilities for these CV functions is infinitely large, the chosen ones are often non-optimal. Moreover, when these CVs are not able to capture the pertinent information contained in the system’s particle positions during the activated process, the resulting free energy landscapes and the structural and dynamical information can be wrong. At first sight, a solution seems to include as many CVs as possible, but most sampling methods suffer from the curse of dimensionality and scale very badly with the number of CVs.

As an alternative to conformational enhanced sampling, trajectory based methods, such as nudged elastic band, milestone, transition path sampling (TPS), transition interface sampling...
The Journal of Chemical Physics

II. METHODS

A. The extended autoencoder model

Similar to a normal AE, the extended autoencoder (EAE) model (Fig. 1) has a bottleneck and is forced to learn an efficient encoding and decoding of the incoming data to minimize its loss function. The AE is called “extended” as the bottleneck is also connected to a committor predictor NN. Consequently, the encoder needs to find an efficient encoding for information relevant for both the reconstruction and the committor prediction.

All three parts of the EAE (encoder, reconstruction decoder, and committor decoder) are NNs. Each of the NNs consists of an input layer, several densely connected hidden layers, and an output layer. For the encoder, the size of the input layer is equal to the number of dimensions used, while the output layer size is determined by the number of bottleneck nodes. The reconstruction decoder uses an input layer of the size of the bottleneck. Its output layer size corresponds to the number of input layers of the encoder. The input layer needs to pass through the bottleneck, the model is forced to learn a mapping from the high-dimensional input space to a low-dimensional latent space and back. The layers leading up to the bottleneck, denoted the encoder, map the input on this latent space, while the layers following the bottleneck, the decoder, map from the latent space back to the original space. The encoder prioritizes the input with the highest information density and identifies complex relationships between the inputs. The latent space, therefore, contains the most relevant information. As the decoder provides an efficient mapping from latent to configurational space, it can also be used generatively. Provided with any point on the latent space, the decoder will reconstruct a full dimensional output: novel instances of the dataset that adhere to the rules learned by the autoencoder. For molecular systems, these generated points can be used as starting points for simulations. Therefore, AEs can facilitate the exploration of new regions of the configurational space.

To our knowledge, until now, no model exists that can map configurations onto and back from a latent space, simultaneously making predictions about the committor of a given configuration, and thus parameterize the committor data while at the same time performing the dimensionality reduction. However, these two tasks are highly compatible, as both of them rely on filtering for the most relevant information from the input. Combining an AE and a predictor NN to share some of their architecture therefore seems a logical next step. Here, we aim to construct such an extended autoencoder (EAE) model as a proof of concept.

We test the EAE model on simple toy models and then apply it to a realistic dataset for methane hydrate nucleation. Moreover, we investigate the obtained latent spaces in the EAE to see what insight can be gained from the mapping between configurational space and latent space and vice versa. Finally, we will look into the feasibility of the decoder for the generation of transition paths. Demonstrating the inner mechanism of the autoencoder in this way, we remove some of its unnecessary mystery.

The remainder of this paper is structured as follows: In Sec. II, we introduce the model, datasets, and data processing procedures. We then discuss the resulting model prediction and reconstruction. We end with concluding remarks.
of the committor decoder is equal in size to the bottleneck as well, while its output layer is of size one.

Conceptually, the encoder maps information from the configurational space onto a lower-dimensional latent space. The dimensionality of this latent space corresponds to the number of bottleneck nodes. The latent space can be accessed by returning the output of the encoder for a given input. Conversely, the reconstruction decoder can map from the latent space back onto the configurational space. Any point with a matching dimensionality can be converted into a corresponding full-dimensional point by the decoder. This allows a generative usage of the reconstruction decoder, even for points on the latent space that were not in the original dataset.

Both the reconstruction decoder and the committor decoder are regression models trained via supervised learning using a loss function. While the output of the encoder is passed to both of these models, they have no direct influence on each other.

Due to the modular build of the models, six different (sub-)models can be accessed: the complete extended autoencoder (encoder, reconstruction decoder, committor decoder), the basic models of the encoder, reconstruction decoder, and committor decoder. Each of these models can be prompted for predictions based on a given input.

The default model parameters used in this work are listed in Table I. These settings were chosen after rigorous testing on the available data as they yielded the best predictions (supplementary material, Appendix C). We note that the linear encoder gives results that in principle could have been achieved using principle component analysis (PCA). However, since we compare linear with non-linear encoders in this work (see, e.g., Sec. III C), we choose to keep the EAE architecture the same. Note also that we used different loss functions for the decoder and the committor. For the decoder, the absolute error was used, while for the committor decoder, we employed the negative binary log likelihood (see Appendix C of the supplementary material).

### B. Datasets

Three different datasets were used to develop and test the EAE model. Two toy datasets based on the double-well and more complex Z-potential were generated as simple test cases before moving to the more complex methane hydrate nucleation dataset produced in Refs. 30 and 31.

#### 1. Double-well potential

For a first validation of the EAE model, we generated a dataset using a simple 2D double-well potential,

\[
V(x_1, x_2) = x_1^6 + x_2^6 - 0.7(e^{-7.5((x_1-0.5)^2+x_2^2)} + e^{-7.5((x_1+0.5)^2+x_2^2)}).
\]

(1)

Two metastable states A and B were defined as circles of radius 0.2 around the minima. The potential and states are depicted in Fig. 2(a).

An MD trajectory of \(10^7\) time steps was generated on the double-well potential utilizing the Langevin BAOAB integrator from the OpenPathSampling\textsuperscript{32} toy engine with time step \(dt = 0.02\), temperature \(T = 0.1\), and friction \(\gamma = 2.5\). Eight additional dimensions were generated from random noise uniformly distributed between \(-1\) and \(1\), acting as decoys that the EAE should filter out.

Paths were cut upon entry or exit of either of the states, and all snapshots lying within the states were discarded. What remained

---

**TABLE I. Default EAE parameters.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Encoder</th>
<th>Reconstruction decoder</th>
<th>Committor decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of input nodes</td>
<td>Number of dims</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Number of output nodes</td>
<td>1</td>
<td>Number of dims</td>
<td>1</td>
</tr>
<tr>
<td>Layer type</td>
<td>Dense</td>
<td>Dense</td>
<td>Dense</td>
</tr>
<tr>
<td>Number of hidden layers</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Number of hidden layer nodes</td>
<td>4 × number of dims</td>
<td>4 × number of dims</td>
<td>4 × number of dims</td>
</tr>
<tr>
<td>Act. f. hidden layers</td>
<td>Linear</td>
<td>tanh</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>Act. f. output layer</td>
<td>Linear</td>
<td>Linear</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>Loss function</td>
<td>None</td>
<td>Absolute error</td>
<td>Binary log likelihood</td>
</tr>
</tbody>
</table>
was a set of disjointed paths. All paths in this set were assigned to one of four classes: AA, AB, BA, or BB paths. The union of these four classes constituted the total path ensemble, in which each path has an identical weight. In total, 28,745 AA, 520 AB, 520 BA, and 30,114 BB paths were generated.

Figure 2 shows the configurational density map as well as the approximated \( p_B \)'s of the double-well toy dataset.

2. Z-potential

As a more complex test case, we generate a dataset based on the Z-potential\(^1\) given by

\[
V(x_1, x_2) = \frac{x_1^4 + x_2^4}{20480} - 3e^{-0.01(x_1 + 5)^2 - 0.2(x_1 + 5)^2} - 3e^{-0.01(x_1 - 5)^2 - 0.2(x_1 - 5)^2} + \frac{5e^{-0.2(x_1 - 5)^2}}{1 + e^{x_1 - 3}} + \frac{5e^{-0.2(x_1 + 5)^2}}{1 + e^{x_1 + 3}} + 3e^{-0.01(x_1 + x_2)}
\]

(2)

States A and B were defined as ellipsoids around the minima,\(^3\)

\[
(x_1, x_2)(x_1 - x_{1a})^2/16 + (x_2 - x_{2a})^2 < 0.25,
\]

where \((x_{1a}, x_{2a})\) are positioned at \((-7.2, -5.1)\) for state A and at \((7.2, 5.1)\) for state B. The barrier between these two states has a height of \(\sim 4.5k_B T\).

Furthermore, eight further dimensions were added through harmonic oscillators, centered around the origin, so that the total potential becomes

\[
V(x_1, x_2, \ldots, x_6) = \sum_i \langle x_i \rangle^2 + V_{\text{Z}}(x_1, x_2).
\]

(3)

An MD trajectory of \(3.2 \times 10^7\) time steps was generated on the Z-PES utilizing the Langevin BAOAB integrator at \(dt = 0.02\), temperature \(T = 1.0\), and \(\gamma = 1.0\). As for the double-well dataset, the resulting long path was split into the sets of short AA, AB, BA, or BB paths. In total, 207,563 AA, 582 AB, 582 BA, and 210,139 BB paths were generated, all with identical weight. Figure 3 depicts the configurational density, as well as the approximated \( p_B \)'s of the Z-potential dataset.

3. Nucleation data

The third dataset consisted of methane hydrate nucleation trajectories, obtained using path sampling.\(^2\) Methane hydrate is a gas hydrate, consisting of methane and water molecules. This ice-like solid forms preferentially at high pressure and low temperature.\(^2,3\) Generally, in gas hydrates, the water molecules form cage-like structures in which the gas molecules are captured. These cages can have several geometries and arrange into different larger structures. The three predominant cage types in gas hydrates are \(5\)\(^1\) (12 pentagonal faces), \(5\)\(^2\)6\(^2\), and \(5\)\(^3\)6\(^3\), (12 pentagonal and two or four hexagonal faces, respectively). Structure I (sI) is composed of \(5\)\(^1\) and \(5\)\(^2\)6\(^2\) cages in a 1:3 ratio. Structure II (sII) on the other hand is made up of \(5\)\(^1\) and \(5\)\(^3\)6\(^3\) cages in a 2:1 ratio.\(^3\) Naturally, methane hydrate occurs mainly in sI but can also occur in sII as well as in an amorphous state.\(^2,3\)

Previous computational investigations into methane hydrate nucleation have identified a temperature-dependent shift in the reaction mechanism. At strong undercooling (270 K), the water/methane solution nucleates into the amorphous solid state, while at higher temperatures (285 K), the crystalline sI prevails.\(^3\) The RC of the amorphous nucleation was shown to mainly depend on the size of the nucleus. The RC of the crystalline nucleation, however, depended on both nucleus size and structural features.\(^3\) Here, we test these findings using the EAE model.

The mutual coordinated guest (MCG) order parameter can distinguish the solid from the liquid\(^3\) (see also Appendix B of the supplementary material). The liquid state A was defined as MCG \(\leq 18\), while the solid state B was defined as MCG \(\geq 120\). A barrier of \(\sim 57 k_B T\) separated these two states.\(^2\)

Both TIS and TPS were used to produce the dataset. AA and AB paths were generated via TIS, while additional AB paths were generated via TPS\(^3,3\) (see Appendix B of the supplementary material for more details). The paths from TIS were reweighed,

\[
\begin{align*}
\text{FIG. 2.} & \quad \text{Left: Two-dimensional energy surface of the double-well potential. Contour lines every 0.5 k_BT. Red: State A. Blue: State B. Middle and Right: Visualization of all points along the x_1 and x_2 dimensions occupied by instances of the double-well potential dataset. (Middle) Configurational density. (Right) Approximated p_B's.}
\end{align*}
\]
leading to the reweighed path ensemble (RPE). TPS paths were added to this ensemble as specified in the supplementary material. The final dataset consisted of paths with their respective weights and label AA or AB. No BB or BA paths were present in the dataset. In total, the dataset consisted of 13 787 AA paths and 222 AB paths amounting to a total of 7 038 943 snapshots.

While in principle one could use all coordinates of the ∼3000 atoms as the input in the EAE, in practice this is unfeasible. A common solution is to preprocess the data to compute important features or order parameters (OPs). As in Refs. 30 and 31, 22 OPs were selected to describe the system, listed in Table II.

### C. Data processing

All three datasets were treated equally. Each path was decomposed into a list of snapshots. The snapshots inherited the label and weight from their parent paths.

**TABLE II.** Order parameters of the methane hydrate nucleation dataset.

<table>
<thead>
<tr>
<th>OP</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCG</td>
<td>Number of nuclear methanes</td>
</tr>
<tr>
<td>$N_{1,2}$</td>
<td>Number of nuclear methanes with $\leq 2$ nuclear methanes within 9 Å</td>
</tr>
<tr>
<td>$N_{1,3}$</td>
<td>Number of nuclear methanes with $\leq 3$ nuclear methanes within 9 Å</td>
</tr>
<tr>
<td>$N_{1,4}$</td>
<td>Number of nuclear methanes with $\leq 4$ nuclear methanes within 9 Å</td>
</tr>
<tr>
<td>$N_c,2$</td>
<td>Number of core methanes: $MCG - N_{1,2}$</td>
</tr>
<tr>
<td>$N_c,3$</td>
<td>Number of core methanes: $MCG - N_{1,3}$</td>
</tr>
<tr>
<td>$N_c,4$</td>
<td>Number of core methanes: $MCG - N_{1,4}$</td>
</tr>
<tr>
<td>$N_w,2$</td>
<td>Number of waters with $\geq 2$ nuclear methanes within 6 Å</td>
</tr>
<tr>
<td>$N_w,3$</td>
<td>Number of waters with $\geq 3$ nuclear methanes within 6 Å</td>
</tr>
<tr>
<td>$N_w,4$</td>
<td>Number of waters with $\geq 4$ nuclear methanes within 6 Å</td>
</tr>
<tr>
<td>$N_{w,2}-3$</td>
<td>Number of surface waters: $N_{w,2} - N_{w,3}$</td>
</tr>
<tr>
<td>$N_{w,3}-4$</td>
<td>Number of surface waters: $N_{w,3} - N_{w,4}$</td>
</tr>
<tr>
<td>$5^{12}6^2$</td>
<td>Number of cages with 12 pentagons and 2 hexagons</td>
</tr>
<tr>
<td>$5^{12}6^3$</td>
<td>Number of cages with 12 pentagons and 3 hexagons</td>
</tr>
<tr>
<td>$5^{12}6^4$</td>
<td>Number of cages with 12 pentagons and 4 hexagons</td>
</tr>
<tr>
<td>$4^55^{10}6^2$</td>
<td>Number of cages with 1 square, 10 pentagons, and 2 hexagons</td>
</tr>
<tr>
<td>$4^55^{10}6^3$</td>
<td>Number of cages with 1 square, 10 pentagons, and 3 hexagons</td>
</tr>
<tr>
<td>$4^55^{10}6^4$</td>
<td>Number of cages with 1 square, 10 pentagons, and 4 hexagons</td>
</tr>
<tr>
<td>CR</td>
<td>Ratio of $5^{12}6^2$ and $5^{12}$ cages</td>
</tr>
<tr>
<td>$R_g$</td>
<td>Radius of gyration of the nucleus</td>
</tr>
<tr>
<td>$F4$</td>
<td>Mean cos of HO⋯OH torsion angles for waters $\leq 3.5$ Å apart</td>
</tr>
</tbody>
</table>
All snapshots and their corresponding labels and weights were shuffled to obtain a randomly ordered dataset. This dataset was then split into training, validation, and test set, according to a 6:3:1 split ratio.

As the datasets had limited usability in their crude form, further processing steps followed. Each of these steps was executed by a different function. Due to the modularity and unified interfaces of these functions, it was possible to assemble them into a variety of different pipelines. The pipeline used during this work and its components are described below. A schema of the pipeline is shown in Fig. 4.

1. Outlier removal

As a first step, outliers were removed to avoid too many empty bins in the subsequent binning procedure. For this, the values at the 2nd and 98th percentile of each input dimension were chosen as the lower and upper bound, respectively. The complete dataset was revisited, and values falling below the lower bound or above the higher bound were respectively increased or decreased to match it. As a consequence, the values of each dimension were capped by its lower and upper bound.

2. Normalization

Having input variables that span different ranges can impede the performance of a NN. Normalization was therefore applied to bring all input dimensions to a comparable order of magnitude. The data of each dimension $i$ were normalized according to $x_{\text{norm}} = \frac{x_i - \mu_i}{\sigma_i}$, where $x_i$ is the original value of the current data point, $\mu_i$ is the mean value, and $\sigma_i$ is the standard deviation along dimension $i$. After the normalization, the data for each dimension were centered around 0 with a standard deviation of 1.

3. Reduction of input dimensions

To save memory and increase performance speed in the case of the methane hydrate nucleation, where only a subset of dimensions present in the data was of direct interest, the data columns of the methane hydrate nucleation, where only a subset of dimensions present in the data was of direct interest, the data columns were eliminated. A schema of the pipeline is shown in Fig. 4.

4. Snapshot binning

As the values of the input dimensions were lying on a continuous spectrum, most positions on the hyper-cube were only visited by a maximum of one path. To get a reliable estimate of an average $p_B$ value, however, a higher number of paths passing through a point was needed. Therefore, the points were binned, using a resolution $r$. The value along each dimension of each point was then mapped onto one of the $r$ discrete values, by shifting, rescaling, and rounding using

$$x_{\text{bin}} = \left\lfloor \frac{x_i - \min_i}{\max_i - \min_i} r + 0.5 \right\rfloor,$$

where $i$ represents the current dimension, $x_{\text{bin}}$ represents the rescaled value of dimension $i$ of the current point, $x_i$ represents the original value of dimension $i$ of the current point, $\min$, represents the minimal value of dimension $i$, $\max$, represents the maximal value of dimension $i$, and $r$ represents the resolution.

5. $p_B$-approximation

The $p_B$ value for each of the respective bins was approximated as the sum of weighted labels divided by the sum of weights,

$$p_B = \frac{\sum_{i=1}^{n} w_i l_i}{\sum_{i=1}^{n} w_i},$$

where $n$ is the number of snapshots within the bin, $l_i$ is the label of the $i$th snapshot within the bin, and $w_i$ is the weight of the $i$th snapshot within the bin. The use of hash tables allowed a memory-efficient approximation of $p_B$'s even for high-dimensional data. We stress that while the resulting dataset is still noisy, for bins with $O(10^3)$ entries, the error is expected to be just a few percent. In principle, changing $r$ can improve this error, at the cost of lower resolution.

6. Packing of Tensorflow datasets

Finally, the snapshots and the approximated $p_B$ values were packed together into Tensorflow dataset objects, distributed into batches of 64 instances each, and passed to the model for training and validation.

D. Input importance measure

Often non-linear machine learning models are hard to interpret and seen as black boxes. Therefore, several strategies have been developed to extract insights about the relevance and relationships of the input. One such approach is to measure the contribution of each input dimension to the overall quality of the model. Methods such as holdback input randomisation, input permutation, input perturbation, and input averaging have been put forward. These methods share that they act on an already trained model and modify the dataset along one dimension to estimate its importance. While the fact that no retraining of the model is necessary can be time-saving, these methods tend to overestimate the importance of dimensions that are highly correlated with others. As the methane hydrate nucleation dataset used in this work possessed many strongly correlated dimensions, these methods were deemed unsuitable. However, as all relevant information passes through the bottleneck, the problem of finding the input importance of the complete model can be reduced to finding the input importance of the encoder.
The linear encoder allows a simple importance measure by assessing the linear contribution of each input to the encoder's output. We implemented an importance measure for linear encoders based on this principle. The method takes the linear contributions to the encoder and visualizes them for a simple assessment of input importance.

E. Code

In this work, custom code was written using IPython, Matplotlib, Numpy, Openpathsampling, Plotly, Scikit-learn, and Tensorflow. The code is available via GitHub.

III. RESULTS AND DISCUSSION

A. Double-well potential

As a proof of concept, the model was tested on the 2D double-well potential dataset, in which the eight out ten dimensions acted as decoys. In Fig. 5, the ground truth (i.e., original dataset) and the respective predictions by the model are displayed in pairwise heat maps for five (out of ten) dimensions of the dataset ($x_1, x_2, ..., x_5$). Additionally, Fig. 6 shows a close-up view of the most relevant $x_1/x_2$ plot for both ground truth and prediction.

In the first column of Fig. 5, the ground truth shows a clear split of the approximated $p_B$ values along the $x_1$ axis. Low $p_B$ values are associated with negative values for $x_1$ and high $p_B$ values with positive ones. A vertical dividing surface is located at $x_1 = 0$ for all subfigures in this column (red dots). The predictions by the model closely resemble the ground truth in all heat maps of the first column. The model also captures the split between the low and high committor along the $x_1$ axis. For the second, third, and fourth column, the heat maps of the ground truth appear to be dominated by random noise, with a large fraction of values lying close to an approximated $p_B$ of 0.5 (red dots). This behavior is captured less clearly by the model. While its predictions are also dominated by values close to 0.5, there are also regions with higher or lower predicted $p_B$ values. However, no clear trend appears in the heat maps belonging to these columns. In contrast, the $x_1/x_2$ plots in Fig. 6 show that the model predicts the committor with high accuracy. The position of the dividing surface and the colors on the two sides of the barrier are closely matched between the ground truth and prediction. Thus, the model has clearly learned the underlying dynamics of the reaction and can reproduce them.

Apart from the model’s predictions, its reconstruction was also assessed. Figure 7 depicts the reconstructed values for each of the input dimensions. A good reconstruction of the input values is signified by a diagonal line, indicating that low input values yield low output values and high input values yield high output values. Such a diagonal line is only present for $x_1$. The reconstruction line for all other input dimensions is horizontal, indicating poor reconstruction. The model focused on the $x_1$ dimension, deeming it the most important one while placing little value on the other dimensions. Again, this captures the reaction mechanism correctly and allows the model to reconstruct snapshots with the most relevant dimension intact.

Passing an exemplary path in the latent space to the reconstruction decoder, it is mapped into the configurational space. Figure 8 depicts in a radar chart (left) how this path develops along all five dimensions with each axis starting from the origin, as well as how the $x_1$ and $x_2$ components develop with respect to the underlying double-well potential (right). The $x_1$ values start low and increase together with the latent space values. All other dimensions stay roughly constant around values of 0 as the path progresses, indicating that they are unaffected by the change in the latent variable. In the overlay with the double-well potential energy surface, the
The first point of the generated path lies left of state A and the next point, however, already lies within state A. From there on, the path moves in a straight line passing over the barrier toward state B, ending to the right of this state. Figure 8 shows that the model reconstructs snapshots with the most relevant dimension intact. The model thereby produced an exemplary path that is representative of the underlying reaction mechanisms associated with the double-well potential. The generated path is not a perfect representation of a transition path, however, as it begins and ends outside of state A and state B, respectively. This is probably due to the fact that the latent path was selected such that it spans the complete latent space. Adjusting the range spanned by the latent path should yield an accurate representative path.

Overall, the EAE model learns the reaction mechanism of the double-well potential dataset and makes predictions that closely match the known dynamics. While the double-well potential data are quite simplistic, this success served as a first proof of concept.

**B. Z-potential**

Next, we test the model on a ten-dimensional dataset generated on the 2D Z-potential, which again contained 8 decoy dimensions.
A visual comparison of ground truth and predictions by the model is depicted in Figs. 9 and 10.

The $x_1/x_3$ plot of the ground truth shows the characteristic $z$-shape that is associated with the Z-potential. The lower region of the plot shows $p_B$ values below 0.5, and the higher region shows values above 0.5. At the barrier, a set of red dots is present, indicating a region with an approximated $p_B$ of around 0.5. The prediction somewhat deviates from the ground truth. While there is also a clear split between the lower and upper regions, the split between these two regions is flattened out compared to the ground truth. Furthermore, only a transition from green to yellow, but no red dividing region is present, indicating that there is a sudden change in the predicted $p_B$ values. For the other heat maps in the first column of Fig. 9, the ground truth shows a clear split along the $x_2$ dimension. The prediction captures the transition from low to high $p_B$ values, but again no red dividing region can be observed. Similarly, for the second column, a clear split along the $x_2$ dimension can be seen in the ground truth, which is also captured by the model’s prediction. The remaining two columns are dominated by random noise. In the ground truth, the majority of points are red, indicating an approximated $p_B$ close to 0.5. In the prediction, this behavior is not captured. Instead, both higher and lower values are being predicted for the heat maps in these columns.

Figures 9 and 10 thus show that the model learned the dependence of the committor on both $x_1$ and $x_2$, while it disregarded the random dimensions $x_3$, $x_4$, and $x_5$. For the $x_1/x_2$ plot, the model’s predicted dividing surface is flatter than the one found in the ground truth (see Fig. 10), indicating that the model struggles to exactly capture this complex, non-linear barrier. Hence, while the model appears to have learned an approximation of the reaction mechanism, it failed to learn the exact mechanism with the resources available. This is likely due to the small size of the bottleneck in the EAE, as the limited amount of information that can pass through it might not be sufficient to fully describe the barrier exactly. However, while increasing the size of the bottleneck leads to a better performance of the model (supplementary material, Appendix C), a larger bottleneck also means that the information is less condensed and the latent space itself becomes less informative. A bottleneck size of one therefore still appears to have the best trade-off between the performance and information density.

To test the reconstruction of inputs, again input and output values of the AE part of the model were compared. Figure 11 depicts this comparison. For the reconstruction of inputs, $x_1$ shows a very clear diagonal line, indicating a good reconstruction. In addition, $x_2$ shows a stepwise pattern, where input values below 0 yield a constant low output, input values above 0 yield a constant high output, and in between a swift transition occurs. While this captures some behavior of the data, the reconstruction of these values is not so good. The other dimensions show mainly horizontal lines, meaning that the input value for a given dimension does not influence the corresponding output value. The model correctly identified these dimensions as irrelevant and did not focus on learning them.

An exemplary latent path was converted into a path on the configurational space using the reconstruction decoder. Figure 12 visualizes how the generated path develops along $x_1, \ldots, x_5$ (left), as well as only along the $x_1$ and $x_2$ dimensions (right). When mapping the latent path onto the configurational space, mainly the $x_1$ and $x_2$ values change as the path progresses, while the other dimensions stay mostly unchanged. Initially, the $x_1$ values start low, while the $x_2$ values begin somewhat higher but also in the negative region. As the path progresses, at first the $x_1$ values rapidly increase, while the $x_2$ values fall slightly. Then simultaneously, the $x_1$ values decrease, while the $x_2$ values quickly increase. Finally, the $x_1$ values once more increase rapidly, while the $x_2$ values decrease moderately. At the same time, $x_1$, $x_4$, and $x_5$ experience only minimal changes in values. In the overlay of the $x_1$ and $x_2$ dimensions with the Z-potential energy surface in Fig. 12(b), a Z-shaped path is apparent. The path begins left and below state $A$ and then moves past the state before turning and passing over the barrier. The path then turns again and moves toward state $B$. In the final frame, the generated path leaves state $B$ and ends to its right. The constructed path depicted in Fig. 12 is highly representative of how $AB$ paths on the Z-potential develop. The close reproduction of the Z-shaped motion along the $x_1$ and $x_2$ axis indicates that the
model successfully learned the underlying reaction mechanism of the Z-potential dataset. Similar to the double-well potential data, the generated path begins and ends outside of the states. Moreover, it passes state A, without entering. While a more careful selection of the latent space path could contribute to the construction of a more realistic path, the reproduction of the configurational path from the latent space path clearly reveals the inner workings of the autoencoder.

In summary, while the Z-potential dataset posed a larger challenge, the EAE model was still able to discern relevant variables.

![Diagram](image-url)
from irrelevant ones, learn a good approximation of the underlying reaction mechanism, and make reliable committor predictions, even with a single bottleneck node.

C. Methane hydrate nucleation

After showing the feasibility of the EAE model, it was tested on the methane hydrate nucleation dataset. While the dataset consisted of 22-dimensional data, first only a subset of eight representative dimensions was utilized (MCG, 512, 51262, 51264, CR, Rg, F4, Nw,3). Preliminary results for the full-dimensional dataset can be found in Appendix D of the supplementary material.

As for the nucleation dataset, the type of activation function in the encoder significantly influences the results, and we compare an encoder with non-linear activation functions for all nodes, with a variant that uses only linear activation functions.

1. Committor prediction

Figure 13 compares the ground truth with the predictions made by the EAE using the non-linear and linear encoders. We first focus on the non-linear encoder (top right panel). From the ground truth, it is apparent that certain variables are correlated with the committor. For increasing values of MCG, 512, 51262, and Nw,3, a trend of an increasing committor can be observed. For CR, Rg, F4, and 51264, however, changes in the values of these dimensions have little or no influence on the committor—no or only a weak correlation between these dimensions and the committor exists. Furthermore, plots for F4 and 51264 are sparse, displaying many empty rows or columns, as only a few distinct values exist for these variables. The predictions appear to capture the trends well. While the model does not seem to reproduce the lowest committor values in all plots, it still reliably replicates the trends seen in the ground truth. The correlation of the committor and MCG, 512, 51262, and Nw,3 can be seen clearly in the predictions. Even the shape of the dividing surfaces is reproduced by the model.

Figure 13 (bottom left) shows the prediction of the model with a linear encoder. Similar to the non-linear encoder, the linear encoder model captures the correlations between MCG, 512, 51262, and Nw,3. However, very low committor values are not reproduced well. Furthermore, the model appears to capture the general trend of the dividing surfaces, although not all their details.

Both linear and non-linear encoders capture the underlying reaction mechanism and reproduce the dependence of the committor on MCG, 512, 51262, and Nw,3. Until now, both variants fail to replicate very low committor values found in the ground truth. This is particularly apparent for the linear encoder variant. The committor dividing surfaces are, for the most part, reproduced by both models, although the linear encoder variant performs less well in this respect. Compared to the non-linear encoder variant, the dividing surfaces sometimes lack details and diverge more from the ground truth (e.g., in the 512 − 51262 or Nw,3 − 51262 plots). The non-linear encoder variant of the EAE model, therefore, appears to be more suitable for a reliable learning of the committor landscape, allowing for a more complex encoding of the information that passes through the bottleneck, thereby retaining more of the information and relationships relevant for the committor prediction.

2. Reconstruction

To assess the model’s ability for input reconstruction, inputs and their respective reconstruction for the eight dimensions of interest are plotted in Fig. 14. For the non-linear encoder (top panel), a good reconstruction is found for MCG, 512, 51262, and Nw,3. The scatter plots for these variables display a diagonal trend, although some points deviate from a strict diagonal. For CR and Rg, reconstruction is less reliable. The reconstruction for Rg follows a more step-wise pattern, while the CR reconstruction is highly scattered along the diagonal line. Finally, F4 and 51264 show no diagonal trends at all. The output values for F4 are mostly unchanged by the input values, indicating a poor reconstruction.

The reconstruction based on inputs for the linear encoder is displayed in the bottom panel of Fig. 14. The subplots for MCG and Nw,3 show almost perfect diagonal lines, while a clear diagonal trend can also be made out for 512 and 51262. Rg shows a non-diagonal trend, flattening out for higher input values. CR shows some increase in reconstructed values as the input values grow, but has neither a clear diagonal trend nor a steep slope. Finally, F4 shows no clear dependence of the output values on the input values at all.

For both variants, the reconstruction of MCG, 512, 51262, and Nw,3 was good, while Rg, 51262, and CR were reconstructed less well and the reconstruction of F4 was poor. This again indicates the importance of MCG, 512, 51262, and Nw,3 as the models appear to have favored learning these compared to the other dimensions. This aligns well with the correlation between the committor and these dimensions seen for the committor predictions. While both

![FIG. 14. Output of the reconstruction decoder of the trained model, based on given inputs for eight representative dimensions. Top: Non-linear encoder. Bottom: Linear encoder.](image-url)
variants produce good reconstructions for $MCG_{30}$, $MCG_{45}$, and $N_{w,3}$, the linear encoder variant outperforms the non-linear encoder variant as its reconstructions of these four dimensions follow a clearer diagonal trend and show less scattering than the reconstructions made by the non-linear variant. A linear relationship between the input and reconstructed output is more readily retained when the encoding is linear as well, and the non-linear decoder only needs to maintain the linear relationship passed to it by the encoder. With a non-linear encoder, on the other hand, the decoder needs to revert a non-linear relationship back to a linear one, which is potentially harder to achieve.

3. Committor path

To investigate how the committor develops over the course of a path, AA paths from four of the TIS interfaces as well as one AB path generated via TPS were selected and mapped onto the committor space. As each successive TIS interface is closer toward state $B$, these paths should gradually interpolate between AA and AB paths. Figure 15 depicts the selected paths mapped onto the committor space by the model. The results are similar for both variants. All paths begin with a committor value close to 0. The paths belonging to the lower interfaces ($MCG_{30}$ and $MCG_{45}$) do not deviate from this low committor in the slightest as they progress. The path belonging to $MCG_{70}$ shows an almost negligible excursion from $p_B = 0$, as it does not exceed $p_B = 0.01$. In contrast, the path belonging to $MCG_{100}$ shows a clear excursion toward higher $p_B$ values and even reaches a committor around 0.25 before returning to a value close to 0. Finally, the TPS path starts at a low $p_B$ value but quickly reaches a $p_B$ close to 1. At around 60% of the path, the committor drops back to below 0.2 before rising again to 1 and staying there until the end of the path.

Both variants have captured the paths belonging to higher interfaces moved farther toward state $B$ and thereby reached higher committor values, before returning to state $A$, compared to paths belonging to lower interfaces. As expected higher interfaces corresponded to higher maximal committor values for the respective paths with both variants. The linear encoder predicts a higher committor than the non-linear one for the $MCG_{100}$ interface. The final AB path drops to lower committor values before reaching 1, indicating that the path moves through regions where the model associates with lower committor values. A potential explanation is that the model learned to recognize one specific channel, while the chosen path either deviates somewhat from this channel or moves through another channel altogether. This would force the model to extrapolate, potentially yielding poor predictions. Such drops might therefore be a special attribute of only a subset of the AB paths. Overall, the projection onto the committor space seems to function well for both the linear and the non-linear encoder and no strong difference in performance appears between these two variants.

We note that the analysis presented here is closely related to recent work on the committor time evolution.\[50\]

4. Latent space path

A path was defined on the latent space and mapped onto the configurational space using the reconstruction decoder. Figure 16 depicts the development of the different OPs over the course of the path for both variants. For the non-linear encoder [Fig. 16(a)], at the start of the latent path, for lower values of the latent variable, all OPs also exhibit low values. While the latent variable value rises to 0.2, the output of these OPs rises to their maximum and remains there. $5^{12}_{62}$ follows a similar pattern but does not grow as steadily in the first phase of the latent path. Instead, it
shows a limited increase until the latent variable rises to ~0.4, then exhibits a larger increase as the latent variable rises to ~0.2, and again experiences a large increase between 0.0 and 0.2. \( R_g \) at first also grows steadily before making a larger jump to its final value, but this jump already occurs between latent values of ~0.2 and 0.0. \( S^{12}6^2 \), \( F_4 \), and \( CR \) show erratic patterns of increase and decrease as the latent variable changes.

Figure 16(b) shows the reconstruction from a latent path generated via the linear encoder model. In addition, here, \( MCG \), \( S^{12}6^2 \), \( S^{11}6^2 \), and \( N_{w,3} \) correlate with the committor. Once the latent value increases, the reconstructed values for these four OPs increase as well. \( S^{12}6^2 \) initially increases more slowly than the other values and only catches up as the path progresses. \( R_g \) also shows an increase in values as the latent value rises, \( F_4 \), however, stays mostly unaffected by the change in the latent value. Finally, \( S^{11}6^2 \) and \( CR \) at first increase in value but then drop again as the latent value further increases.

Both variants of the model appear to capture a transition from low to high values for \( MCG \), \( S^{12}6^2 \), \( S^{11}6^2 \), and \( N_{w,3} \) as the latent space values increase, while the other dimensions show a less clear pattern. This again suggests the importance of the former four dimensions as the model focused on reconstructing these. \( S^{12}6^2 \) has a special role among these, as it initially grows at a rate comparable to \( MCG \), \( S^{11}6^2 \), and \( N_{w,3} \) but then falls behind the other values before closing the gap again. This behavior is identified by both the non-linear encoder model and the linear encoder model. As the reconstructed path can be thought of as a representative transition path, this might indicate a commonly occurring step in the methane hydrate nucleation. In such a common path, initially, both \( S^{11}6^2 \) and \( S^{12}6^2 \) cages would form at similar rates. Then, the number of \( S^{11}6^2 \) would increase faster, while the number of \( S^{12}6^2 \) cages would increase more slowly, yielding a lower cage ratio. Eventually, the growth of \( S^{12}6^2 \) cages would accelerate again and overtake the growth rate of the \( S^{11}6^2 \) cages. While the two variants produce similar trends in the reconstructed paths, their outputs differ partially. Most importantly, for the non-linear encoder model, several reconstructed dimensions (\( MCG \), \( S^{12}6^2 \), \( S^{11}6^2 \), \( N_{w,3} \), and \( S^{12}6^2 \)) experience a large jump before remaining constant at their final value even as the latent value continues to increase. In comparison, the linear encoder variant does not produce such a gap and its reconstructed values are more evenly spaced out. The steady increase in the linear encoder variant seems more realistic than the sudden jump of the non-linear encoder variant. Furthermore, this greater coverage of values enables a closer tracking of how the path develops, thereby potentially leading to more insight into the reaction mechanism. Therefore, the linear encoder variant appears to be better suited for projecting paths from the latent space onto the configurational space.

As for the toy models, this analysis shows how a latent space path encodes for a much more complex reaction coordinate in the configurational space. This reveals some of the inner workings of the EAE model.

5. Input importance

Using a linear encoder allowed for an input importance measure to identify the relevant OPs. Figure 17 depicts these input importances. The highest importance is given to \( N_{w,3} \) with ~39% of the total importance. \( MCG \) was attributed ~29.5% of the total importance. \( S^{12}6^2 \), \( S^{11}6^4 \), and \( S^{12}6^4 \) were attributed to ~18.5%, ~8%, and ~3% respectively. The remaining dimensions \( R_g \), \( CR \), and \( F_4 \) all contributed less than 1% of the total importance. The standard deviation for all importances was low, with none of them exceeding 2%. Both \( N_{w,3} \) and \( MCG \) dimensions have been identified as relevant by the other aspects of the model in the results discussed above. In both the ground truth and the predictions made by the models, the committor strongly correlated with these two dimensions. They also showed the best reconstruction for both the linear and the non-linear encoder variant of the model. These two OPs are a measure of the cluster size. The size, therefore, seems to play a major role in the reaction mechanism. \( S^{11}6^2 \), \( S^{12}6^2 \), and \( S^{12}6^4 \) were also assigned high importance. Of these three, \( S^{11}6^2 \) and \( S^{12}6^2 \) showed a strong correlation with the committor and were reconstructed well by the model, while \( S^{12}6^4 \) had not been identified as relevant by the other methods before. These three OPs are measures of the structure within the cluster, indicating that the structure also plays a part in the reaction mechanism. \( CR \), \( R_g \), and \( F_4 \) were assigned negligible importance. As for the other OPs, this matches with the previously discussed results since both the committor prediction and input reconstruction seemed to place little value on these dimensions. They either are not directly relevant for the reaction mechanism or might encompass redundant information that is already carried by the other OPs. Overall, the input importance measure reflects similar trends as found in the previous results. Both the size and structure were found to be relevant for the reaction, although the size appears to play a larger role than the structure.

IV. CONCLUSION

In this work, we developed and deployed the extended autoencoder (EAE) model, a combination of an autoencoder with a committor prediction model. After training on path ensemble data, the model can predict the committor for any given configuration of the system, map configurations on a lower-dimensional
latent space, and generate configurations by mapping from the latent space back to the configurational space. The EAE model was tested on the double-well potential and Z-potential datasets and was effective in identifying the relevant order parameters and facilitating a better understanding of the underlying mechanism of the reaction.

For a nucleation dataset generated using both transition path sampling (TPS) and transition interface sampling (TIS), the model reliably reproduced the committor landscape and dividing surfaces observed for the original dataset. Additionally, a clear correlation between the order parameter and committor was found for MCG, $S_{12}$, $S_{12}^6$, and $N_{w3}$. Furthermore, the model reconstructed these four OPs well over their full range. The reconstruction of the other OPs ($S_{12}^6$, $R_v$, $CR$, $F4$) was moderate or poor. The model therefore considered the former OPs relevant while placing less value on the latter ones. When constructing a representative path, again MCG, $S_{12}^6$, $S_{12}^6$, and $N_{w3}$ stood out for following a clear trend as the path progressed, indicating that they played an important role in the transition from state A to state B, while the other OPs showed less clear patterns. The representative path also highlighted that initially very few $S_{12}^6$ cages form, while later $S_{12}^6$ cage formation accelerates and surpasses the number of $S_{12}$ cages. Committor prediction and reconstruction and input importance measure on the linear encoder variant all identified $N_{w3}$, MCG, $S_{12}^6$, $S_{12}^6$, and $S_{12}^6$ as the most relevant OPs. These size and structure parameters are both relevant components of the reaction coordinate, but the size might play a larger role.

Both linear encoder and the non-linear encoder variants performed well. The non-linear encoder variant is better at the committor predictions, while the linear encoder variant performs better during the input reconstruction and the mapping of a representative latent space path onto the configurational space. The input importance measure was only possible for the linear encoder variant.

The model has the capacity to produce representative paths by mapping a latent space path onto the configurational space, thus revealing some of the often-perceived as mysterious inner workings of the autoencoder. To be of more practical use, this functionality still needs to be refined. The model does not ensure (yet) that regions that lie close to each other in the configurational space will also lie near each other on the latent space. To prevent distortions, an additional cost function such as in the SketchMap could be added. This might increase the quality of the reconstructed or freely generated structures produced by the model.

Another avenue for improvement might be to increase the smoothness of the latent space. Smoothness is not ensured by the model when generating a path from the latent space, causing discontinuities. To facilitate a smooth latent space and thereby prevent such artifacts, the AE could be replaced with a variational AE.

When applying the model in a generative way, it might prove particularly useful to select OPs that can be converted back into Cartesian coordinates. Visual inspection of these paths would then yield intuitive understanding of the reaction mechanism. Furthermore, mapping different latent-space paths onto the configurational space might allow exploration of new reactive channels.

As the current importance measure only functions on the linear encoder variant of the model, an importance measure functioning on the non-linear encoder variant should also be implemented. One such alternative scheme is the "improved stepwise" method described by Gevrey et al., where the model is repeatedly retrained while leaving out different dimensions. Implementing this method would allow measuring the input importance irrespective of the activation functions and model architecture used.

While several features can still be added, the EAE model is shown to effectively learn the underlying reaction mechanism of a system and facilitate the extraction of novel insights. While we applied EAE here to methane hydrate nucleation, it can be used on other systems as well. The EAE model’s independence on the set of chosen OPs allows for a variety of different datasets that can be explored with this approach. In future work, the model could be used on other datasets to make use of its full potential.

SUPPLEMENTARY MATERIAL

The supplementary material contains appendices on (A) data visualization, (B) datasets and data processing for hydrate nucleation, (C) parameter settings tested, and (D) additional results.

ACKNOWLEDGMENTS

The authors thank Max Welling for discussions and pointing in the direction of extended autoencoders.

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material and from the corresponding author upon reasonable request.

REFERENCES
