

Roadmap

David Devoogdt

1 Goal

This roadmap outlines some recent advancements in sampling methods which use as little knowledge as possible to sample a transition between 2 phases. This is necessary because it's not always possible to find a good reaction coordinates. Once a path is known, other methods can be used to find free energy profiles etc.

In section 2 (subjectively) interesting groups of methods are listed. In section 3 the methods that will be implemented in this repository are selected and discussed. As a model, a toy perovskite system with a 5 atoms unit cell is used. This has 2 stable phases.

2 Groups of methods

2.1 Optimal selection of predefined reaction coordinates

There exist quite some methods to make an optimal selection of a number of predefined collective variables. While this is very useful, it still requires some insight in the reaction mechanism before sampling begins. Therefore, these methods are not very useful without prior knowledge. These methods iteratively select better RC and generate new MD data with the selected RC.

Linear combinations The most limited form includes making a linear combination of a pool of preselected order parameters.

time-lagged independent component analysis (tICA) [1]. Selects linear combination based on dominant eigenvalue of transition matrix. The functions in tICA themselves can be non-linear (kernel tICA) and faster version exist (e.g. hTICA)

SGOOP [1] optimises based on a different principle (minimising spectral gap). SGOOP-d improves upon the idea by using commute distance from [2].

The problem is that this still requires one valid reaction coordinate in the pool for the whole reaction. This is still too difficult for complex reactions

Reinforcement learning based adaptive sampling (REAP) [3] clusters explored structures. The reaction coordinate is changed and initial structures are selected to explore the most unsampled regions. This can use different coordinates dependent on the place in the reaction.

2.2 Manifold learning

Manifold learning assumes that the given high dimensional data is in essence sampled from a low dimensional space. The aim is to transform the data back to a low dimensional manifold without distorting the distances between points. Dimension reduction algorithms tend to fall into two categories; those that seek to preserve the pairwise distance structure amongst all the data samples and those that favor the preservation of local distances over global distance. [4]

Successful examples of these techniques include IsoMap [5] and DMAPS [6] Current state of the art techniques are t-SNE [7] and UMAP [4]. The methods from manifold learning are mostly quite mathematical in nature, but broadly applicable.

These methods are generally not directly linked to QM simulations. In essence the points of an MD trajectory form a low dimensional manifold. The idea here is to select alternatingly better collective variables (learnt by the techniques above) and produce new MD trajectories based on these improved CV's

2.2.1 DMAPS

Diffusion maps (DMAPS) [6] construct an affinity matrix K (which measures in some sense how similar 2 configurations are, e.g. with radial basis function kernel). The d largest eigenvectors of normalised K servers as low dimensional coordinates.

Some methods build on this by running short MD simulations, performing a manifold learning technique and spawn new MD simulation outside currently sampled portion of phase space. (see e.g. diffusion-map directed MD ([8]) and intrinsic map dynamics iMapd [9]). This doesn't need any prior knowledge of the system.

Diffusion nets (DNETS) [10] uses an encoder and decoder to learn the DMAPS mapping between rounds. This makes the problem better behaved and allows to use the autoencoder as CV. More about autoencoders can be found in next section.

2.2.2 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) [7] centers gaussians around every original point x_i and t-distributions around every point in embedded space y_i . The KL difference is minimised such that $p(i|j) = q(i|j)$ with p the neighbour probability in x space and q in y space. Embedding parameters are learned. [11] uses WT-MetaD to explore phase space, and uses a NN map x to y .

Active enhanced sampling [12] is similar in spirit: it also minimises KL div between 2 representations.

2.2.3 UMAP

The theory behind uniform manifold approximation and projection (UMAP) [4] is based on manifold theory and topological data analysis. The paper explains it in language of category theory, and hence this it is certainly too much effort to try to understand it. The computational version is still quite difficult. Luckily there is a reference implementation on github.

In [13], this method is gauged against PCA, tICA and t-SNE for a biomolecule. The generated CV's are competitive with t-SNE but have smaller time complexity.

2.3 Information bottleneck

Many algorithms use an autoencoder as dimensionality reduction. Mostly, the sampled data is put through a neural network with a bottleneck (layer with few (latent) variables).

2.3.1 VAMPnets

In VAMPnets [14], info and time lagged info is put through 2 network lobes which learn latent variables χ_0 χ_1 related by matrix K (function of χ_x). This constructs a MSM.

2.3.2 autoencoders

The neural network is trained to recover its input as output (autoencoder). The learned latent variables are used in a new round of enhanced sampling. This continues until the CV's do not change any more [15].

As a variation, [16] trains 2 genetic algorithms: 1 selects CV's from pool of possible CV's (encoder), and a second NN acts as decoder

The autoencoders are much less prone to noise if gaussian noise is added to the latent variables (Variational auto encoders) [17–19]. Predicting time lagged output also improves reliability, as only slow modes are learned [17, 18, 20]

The latest methods use bayesian learning to construct the network. These methods are all able to construct good coarse grained markov state models. State Predictive Information Bottleneck (SPIB) [21] learns state label instead of configuration and constructs MSM's. It is able to predict committor probability. SPIB uses mixture of gaussian as prior. [17] uses autoencoders with good results. Gaussian mixture VAE [19]: Uses gaussian mixture (sum of gaussian with different center) as prior distribution.

2.4 Normalising Flows

Another recent parameter free technique is Boltzmann generators [22], rooted in the normalising flows (ML technique). The idea is to make an invertible map between a prior distribution (e.g. gaussian) and the desired Boltzmann distribution $p(x) = e^{-\beta U(x)}$. This can be thought of as scaling the distribution around every point with a suitable jacobian. The mapping function is an invertible neural network. The training of these network happens in 2 different ways:

- 1) sample vectors z from prior distribution and transforms them to real configurations $X = F_{zx}(z)$. Each round, the KL divergence between the sampled distribution $p_X(x)$ and the real distribution $p(x) = e^{-\beta U(x)}$ is minimised by adjusting the weights and biases of the NN map. A round requires around 1000 samples from prior.

- 2) Valid samples are taken from an existing MD trajectory, and are transformed to the prior space $Z = F_{xz}(x)$. Here, the Maximal likelihood is minimised (compared to the prior gaussian distribution)

Some improvements were already made. In [23], some relevant physical symmetries of the physical space X are taken into account (i.e. permutations of atoms, global rotations,...). In [24], the flow is temperature steerable, meaning that the output distribution F_{xz} can be scaled based on the desired temperature. This is possible because $p(x, t') = [e^{-\frac{U(x)}{kt}}]^{\frac{t}{t'}} = p(x, t)^{t/t'}$. By suitably selecting the map, the prior can be sampled and trained at any temp. Metropolis algorithm is needed to select unbiased samples

2.5 General Adversarial Networks

Another technique from ML community is called adversarial learning. This is applied in the following paper Targeted Adversarial Learning Optimized Sampling (TALOS) [25] to advanced sampling methods. This is one of the only papers that uses GAN architecture. It’s still too early at this point to do something with it

2.6 Path-based methods

Transition Path Sampling (TPS) Transition path sampling takes one known path between 2 phases and from there samples a number of reaction paths between the states to determine the statistical properties. Path are accepted based on a metropolis sampling criterium. See e.g. [26].

In the recent paper [27], the committor probability is stored in an NN.

There are some difficulties related to TPS: only one path can be sampled, slow, and initial path needs to be known (and be a faithful path)

Metadynamics of paths [28] combines metaD and path sampling. Seems slow for large systems?

3 Selected methods for Perovskites

This section lists the methods that will be tested for perovskite toy model. The idea is to choose a state-of-the-art model from each different type and compare their performance.

3.1 Boltzmann generators

Boltzmann generators are explained in section 2.4, an excelent implementation can be found here [git](#).

The group of Frank Noé is the only group which develops Boltzmann generators/normalising flows at the moment, and this git repository contains all the variants (equivariant, temperature steering).

While all the components to make a normalising flow where in place, some modifications were necessary to adapt it for perovskites. Some of them are listed below:

Implementation work Some work was needed to couple the generator to CP2K (through ASE). This works now stable and reliable. The training of the Boltzmann generators works as intended. The git repo also stores everything cleanly and save all the used parameters to make everything reproducible. It’s connected to weights and biases for easy comparison of experiments

Flow design The flows original papers uses realNVP flows, which splits the parameter in 2 groups and updates one half with an affine transformation (neural network) conditional on the other half. This is very arbitrarily. The current network makes a block for each atom/cell (feature) to all other features individually. Identical combinations of atoms get the same connection. This is very similar in spirit to message passing networks, such as Schnet. There is also a layer which scales and shifts the inputs to make it more similar to a gaussian prior, by choosing the variance different for e.g. the angles than the cell sizes etc.

Sampling At the moment the initial data is generated with a markov chain monte carlo process. (unrelated: it took some time to read about monte carlo algorithms, optimal sampling parameters, ...). This is not necessary and MD methods are also perfectly fine.

phase transition In the original paper, random points in the prior distribution were connected to find the path in real space. For a crystal, it is less clear which point to connect, as in theory any of the I atoms in phase 1 could go to any of the other atoms in phase 2. At the moment, a simple solution seems to find the shortest path in prior space between all equivalent points (i.e. permutations of I atoms). For bigger unit cells, some work will be needed to keep this manageable

Feeling for parameters As this kind of work is quite new to me (distribution sampling, mcmc/MD) and the example from the papers are very different, it's still quite difficult to develop a feeling for good parameters. Also training neural networks requires some intuition which is sometimes lacking

Results It's still too early to talk about results. In hindsight, my learned models were fitted on bogus initial data (sampler generated too many invalid configurations which were all rejected by the mcmc)

Improvements As noted earlier, the current architecture somewhat resembles schnet. There is definitely more things that can be borrowed from MLP's.

3.2 REAP

[3],git.

This is one of the few methods left in the list that explicitly uses preselected coordinates. This might be handy if the phase diagram needs to be compared against other methods.

Implementation TODO

3.3 UMAP

section 2.2.3 git

Umap seems the most promising and scalable method in manifold mapping.

Implementation TODO

3.4 SPIB

[21] git

This uses all recent advances in for autoencoders: bayesian interference learning, time lagging and it produces a reliable MSM

Implementation TODO

3.5 Path based methods

skip for the moment, because accurate initial trajectory is needed.

References

- [1] P. Tiwary and B. J. Berne, “Spectral gap optimization of order parameters for sampling complex molecular systems,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, pp. 2839–2844, 3 2016.
- [2] F. Noé, R. Banisch, and C. Clementi, “Commute maps: Separating slowly mixing molecular configurations for kinetic modeling,” *Journal of Chemical Theory and Computation*, vol. 12, pp. 5620–5630, 11 2016.
- [3] Z. Shamsi, K. J. Cheng, and D. Shukla, “Reinforcement learning based adaptive sampling: Reaping rewards by exploring protein conformational landscapes,” *J. Phys. Chem. B*, vol. 122, p. 8395, 2018.
- [4] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” 2 2018.
- [5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Philos. Trans. R. Soc. London Ser. B*, vol. 67, p. 1777, 1995.
- [6] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 7 2006.
- [7] L. V. D. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [8] J. Preto and C. Clementi, “Fast recovery of free energy landscapes via diffusion-map-directed molecular dynamics,” *Physical Chemistry Chemical Physics*, vol. 16, pp. 19181–19191, 8 2014.
- [9] E. Chiavazzo, R. Covino, R. R. Coifman, C. W. Gear, A. S. Georgiou, G. Hummer, and I. G. Kevrekidis, “Intrinsic map dynamics exploration for uncharted effective free-energy landscapes,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, pp. E5494–E5503, 7 2017.
- [10] G. Mishne, U. Shaham, A. Cloninger, and I. Cohen, “Diffusion nets,” 2015.
- [11] J. Rydzewski and O. Valsson, “Multiscale reweighted stochastic embedding: Deep learning of collective variables for enhanced sampling,” *J. Phys. Chem. A*, vol. 125, p. 6302, 2021.
- [12] J. Zhang and M. Chen, “Unfolding hidden barriers by active enhanced sampling,” *PHYSICAL REVIEW LETTERS*, vol. 121, p. 10601, 2018.
- [13] F. Trozzi, X. Wang, and P. Tao, “Umap as a dimensionality reduction tool for molecular dynamics simulations of biomacromolecules: A comparison study,” *Journal of Physical Chemistry B*, vol. 125, pp. 5022–5034, 5 2021.

- [14] A. Mardt, L. Pasquali, H. Wu, and F. Noé, “Vampnets: Deep learning of molecular kinetics,” *Nature Communications*, vol. 9, 10 2017.
- [15] W. Chen, A. R. Tan, and A. L. Ferguson, “Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design,” *Journal of Chemical Physics*, vol. 149, 8 2018.
- [16] F. Hooft, A. Pérez De Alba Ortíz, and B. Ensing, “Discovering Collective Variables of Molecular Transitions via Genetic Algorithms and Neural Networks,” *Journal of Chemical Theory and Computation*, vol. 17, no. 4, pp. 2294–2306, 2021.
- [17] M. Schöberl, N. Zabaras, and P. S. Koutsourelakis, “Predictive collective variable discovery with deep bayesian models,” *Journal of Chemical Physics*, vol. 150, 1 2019.
- [18] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, “Variational encoding of complex dynamics,” *Physical Review E*, vol. 97, 11 2017.
- [19] Y. B. Varolgünes, T. Bereau, and J. F. Rudzinski, “Interpretable embeddings from molecular simulations using gaussian mixture variational autoencoders,” *Machine Learning: Science and Technology*, 2020.
- [20] C. Wehmeyer and F. Noé, “Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics,” *Journal of Chemical Physics*, vol. 148, 6 2018.
- [21] D. Wang and P. Tiwary, “State predictive information bottleneck,” 2021.
- [22] F. Noé, S. Olsson, J. Köhler, and H. Wu, “Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning,” *Science (New York, N.Y.)*, vol. 365, 9 2019.
- [23] J. Köhler, L. Klein, and F. Noé, “Equivariant flows: sampling configurations for multi-body systems with symmetric energies,”
- [24] M. Dibak, L. Klein, and F. Noé, “Temperature steerable flows and boltzmann generators,”
- [25] J. Zhang, Y. I. Yang, and F. Noé, “Targeted adversarial learning optimized sampling,” *Journal of Physical Chemistry Letters*, vol. 10, pp. 5791–5797, 10 2019.
- [26] C. Dellago, P. G. Bolhuis, S. Csajka, and D. Chandler, “Transition path sampling and the calculation of rate constants,” 1998.
- [27] H. Jung, R. Covino, and G. Hummer, “Artificial intelligence assists discovery of reaction coordinates and mechanisms from molecular dynamics simulations,” 1 2019.
- [28] D. Mandelli, B. Hirshberg, and M. Parrinello, “Metadynamics of paths,” 2020.