Eur. Phys. J. B (2021) 94:211
https://doi.org/10.1140/epjb/s10051-021-00220-w

The European
Physical Journal B

Topical Review - Computational Methods

Collective variable-based enhanced sampling and machine learning

Ming Chen[a]

Department of Chemistry, Purdue University, West Lafayette, IN 47907, USA

Received 2 April 2021 / Accepted 3 October 2021 / Published online 20 October 2021

Abstract. Collective variable-based enhanced sampling methods have been widely used to study thermodynamic properties of complex systems. Efficiency and accuracy of these enhanced sampling methods are affected by two factors: constructing appropriate collective variables for enhanced sampling and generating accurate free energy surfaces. Recently, many machine learning techniques have been developed to improve the quality of collective variables and the accuracy of free energy surfaces. Although machine learning has achieved great successes in improving enhanced sampling methods, there are still many challenges and open questions. In this perspective, we shall review recent developments on integrating machine learning techniques and collective variable-based enhanced sampling approaches. We also discuss challenges and future research directions including generating kinetic information, exploring high-dimensional free energy surfaces, and efficiently sampling all-atom configurations.

1 Introduction

Molecular dynamics (MD) simulation is an important tool to study thermodynamic properties and kinetic properties of complex systems in chemistry, biology, and materials science [1]. Potential energy functions used in MD simulations usually have tremendous local minima separated by high-energy barriers, while thermal fluctuation is the only driving force of barrier crossing. Therefore, the time to cross high barriers is close to or longer than typical MD simulation timescales and a sufficient sampling of barrier-crossing events can require millisecond-scale MD simulations [2]. Various enhanced sampling methods have been designed to assist barrier crossing and these methods have achieved great successes in understanding properties of various chemical systems. [3–17].

In general, there are two categories of enhanced sampling methods that focus on studying thermodynamic properties. The first one is unbiased enhanced sampling which preserves the Boltzmann distribution. The efficiency of this type of enhanced sampling methods is bounded by the central limit theorem, thus reducing sample correlation is the main theme of methodology development. One famous example of unbiased enhanced sampling is the replica exchange method such as parallel tempering [3] or Hamiltonian replica exchange [4]. In the replica exchange approach, multiple replicas of one MD simulation are running simultaneously with different temperatures or different potential energy functions. Sample correlation is reduced by exchanging configurations among replicas to prevent trapping the simulated system in a stable conformation for a long time. The second type of enhanced sampling methods biases the configuration distribution away from the Boltzmann distribution. There are two motivations to bias the Boltzmann distribution. First, biasing the Boltzmann distribution improves statistics at important high free energy locations, including metastable conformations and transition states. Second, increasing the probability of visiting barrier tops can often reduce sample correlation to enhance sampling efficiencies. Since preserving the Boltzmann distribution is not required, designs of biased enhanced sampling are more flexible. Even non-equilibrium dynamics [5–9] has been adopted as long as the Boltzmann distribution can be recovered with post-analysis. Besides two categories of methods for sampling the Boltzmann distribution, there are many methods focusing on sampling in a path ensemble which are necessary for studying kinetic properties like rate constants [18–25]. We want to emphasize that these methods are important members in the family of enhanced sampling even if they are not the main topic of this perspective.

Most biased enhanced sampling methods focus on several important degrees of freedom. These degrees of freedom are named as collective variables (CVs). CVs form a reduced model for a complex process, like a chemical reaction, a biomolecule conformational change, or a material phase transition. With properly selected CVs, important potential energy barriers are mapped onto a free energy surface (FES) so

[a] e-mail: chen4116@purdue.edu (corresponding author)

that biased enhanced sampling on the FES is able to increase the frequency of barrier crossing. CV-based enhanced sampling methods have been developed for decades and there are enormous successful applications of these methods [26–34]. However, there are many theoretical and practical challenges for these methods.

Recent developments on machine learning techniques are changing the landscape of CV-based enhanced sampling, especially on two aspects: CV design and FES construction. In this perspective paper, we shall briefly review CV-based enhanced sampling in Sect. 2 and we shall introduce basic machine learning concepts in Sect. 3. After that, we shall present methods of training CVs and FESes with machine learning techniques in Sects. 4 and 5. In Sect. 6, we shall discuss challenges and perspectives to develop CV-based enhanced sampling with machine learning.

2 Collective variable-based enhanced sampling

We start our discussion from introducing a system of N_a atoms with Cartesian coordinates $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_a}\}$ where $\mathbf{x}_i$ is the Cartesian coordinates of the i'th atom. The system's Hamiltonian is $H = T + U$ where T is the kinetic energy and $U(\mathbf{x})$ is a potential energy function which describes interactions among atoms. A set of d physically intuited CVs $\mathbf{q}(\mathbf{x}) = \{q_1(\mathbf{x}), \ldots, q_d(\mathbf{x})\} \in \mathbb{R}^d$ is introduced as a coarse-grained representation of the system, where $q_i(\mathbf{x})$ is the i'th collective variable. Fixing $\mathbf{q}(\mathbf{x}) = \mathbf{s}$, a partition function at temperature T is defined as

$$Q(\mathbf{s}) = C \int \mathrm{d}\mathbf{x} e^{-\beta U(\mathbf{x})} \delta(\mathbf{q}(\mathbf{x}) - \mathbf{s}), \tag{1}$$

where C is a constant independent of $\mathbf{s}$, $\beta = 1/kT$, and $\delta(\cdot)$ is the Dirac delta function. In other words, $Q(\mathbf{s})$ is defined as a integration of the Boltzmann factor $e^{-\beta U(\mathbf{x})}$ over a manifold determined by $\mathbf{q}(\mathbf{x}) = \mathbf{s}$. A free energy surface $A(\mathbf{s})$ is then defined from $Q(\mathbf{s})$, i.e.

$$A(\mathbf{s}) = -kT \log Q(\mathbf{s}) + \tilde{C}, \tag{2}$$

where $\tilde{C}$ is also a constant independent of $\mathbf{s}$. Since we are interested in a free energy difference between two conformations, $\tilde{C}$ is ignored in most cases.

Since a FES is a simplified model of a complex system, the FES should be able to represent important macroscopic states of molecules and materials, such as stable conformations of molecules and stable phases of materials. Moreover, it is possible to design CV-based enhanced sampling with appropriate CVs to calculate kinetics properties like rate constants [35]. In many cases, timescales of these important conformational changes or phase transitions are much longer then typical MD simulation timescales due to high free energy barriers. For example, conformations of alanine dipeptide in vacuum can be represented by two Ramachandran dihedral angles Φ and Ψ. Using these two dihedral angles as CVs defines a FES on which three minima represent three stable conformations: C7$_{\mathrm{eq}}$, C5, and C7$_{\mathrm{ax}}$. The height of the barrier separating C7$_{\mathrm{eq}}$ and C7$_{\mathrm{ax}}$ is nearly 8kcal/mol higher than the free energy at the bottom of C7$_{\mathrm{eq}}$ well (see panel (b) in Fig. 1). Crossing such a high barrier requires extremely long timescale MD simulations. Therefore, efficiently exploring a FES usually requires enhanced sampling techniques.

One common enhanced sampling approach is to bias the Boltzmann distribution to enhance statistics in some low probability regions, like metastable conformations and transition states. Biasing the Boltzmann distribution is achieved by modifying the equation of motion. Without loss of generality, we shall use the Brownian dynamics as the MD equation of motion. The Brownian dynamics motion equation is

$$\mu \mathrm{d}\mathbf{x} = -\nabla U(\mathbf{x})\mathrm{d}t + \sqrt{2\mu kT}\mathrm{d}W, \tag{3}$$

where μ is a friction coefficient. Sometimes, an extended Lagrangian scheme is used in CV-based enhanced sampling methods. In an extended Lagrangian scheme, CVs are coupled to fictitious degrees of freedom $\mathbf{s}$ with harmonic potentials [7,8,17,37], i.e.

$$\mu \mathrm{d}\mathbf{x} = \left(-\nabla U(\mathbf{x}) - \sum_i \kappa_i (q_i(\mathbf{x}) - s_i)\nabla q_i(\mathbf{x})\right)\mathrm{d}t + \sqrt{2\mu kT}\mathrm{d}W \tag{4a}$$

$$\mu_i \mathrm{d}s_i = \kappa_i (q_i(\mathbf{x}) - s_i)\mathrm{d}t + \sqrt{2\mu_i kT}\mathrm{d}W_i, \tag{4b}$$

where μ_i is an artificial friction coefficient of a fictitious degree of freedom and κ_i is an artificial coupling constant. With Eqs. (4a) and (4b), Eq. (1) becomes

$$Q(\mathbf{s}) = C' \int \mathrm{d}\mathbf{x} e^{-\beta U(\mathbf{x})} \prod_i e^{-\beta\kappa_i (q_i(\mathbf{x}) - s_i)^2/2}. \tag{5}$$

Eq. (5) agrees with Eq. (1) in the limit of $\kappa_i \to \infty$ for all κ_i. Biased enhanced sampling methods modify either Eq. (3) [5,6,9] or Eqs. (4a)–(4b) [7,8,16,17,37,38].

There are two possible ways to biasing the Boltzmann distribution: changing the potential energy function [5,6,9,17,38–41] and increasing the temperature [7,8]. A biasing potential or biasing force is usually introduced to reshape the potential energy function. The biasing potential can either restrain the simulated molecule around a conformation or the biasing potential can fill up minima on a FES to enhance barrier crossing. One famous example of enhanced sampling with restraint potential is the umbrella sampling. The biasing potential used in the umbrella sampling shares a form of

$$U_{\mathrm{bias}}(\mathbf{x}) = \sum_i \frac{\kappa_i}{2}(q_i(\mathbf{x}) - s_i)^2. \tag{6}$$

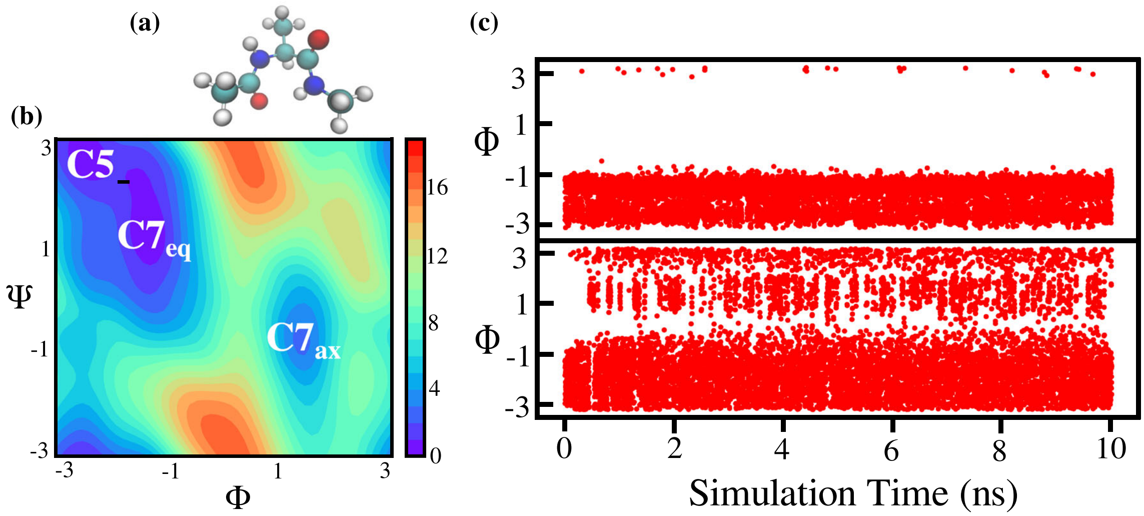

Fig. 1 The FES of an alanine dipeptide **a** in vacuum with two Ramachandran dihedral angles Φ and Ψ as CVs is shown in panel (**b**). The FES in kcal/mol is calculated by a 10ns well-tempered metadynamics simulation with the OPLS-AA force field [36]. Panel **c** shows trajectories of dihedral angle Φ from a MD simulation (upper) and from the well-tempered metadynamics simulation (lower). Without enhanced sampling, the alanine dipeptide molecule is not able to change its conformation to C7$_{\mathrm{ax}}$ within a 10 ns simulation due to high free energy barriers. On the contrary, well-tempered metadynamics significantly enhances barrier crossing between C7$_{\mathrm{eq}}$/C5 and C7$_{\mathrm{ax}}$

[10,41,42]. There are many enhanced sampling methods focus on designing a biasing potential to assist barrier crossing [5,6,9,16,39,40,43]. For example, metadynamics constructs a biasing potential by depositing Gaussian functions along a simulation trajectory $\{\mathbf{x}(t)\}$ [5,6], i.e.

$$U_{\mathrm{bias}}(\mathbf{x},t) = \sum_{t_i \le t} A(\mathbf{x}(t_i)) \times \exp\left\{-\frac{1}{2}(\mathbf{q}(\mathbf{x}) - \mathbf{q}(\mathbf{x}(t_i)))^{\top} \Sigma^{-1} (\mathbf{q}(\mathbf{x}) - \mathbf{q}(\mathbf{x}(t_i)))\right\}, \tag{7}$$

where Σ is a diagonal matrix determining the width of Gaussian functions and $A(\mathbf{x})$ is a prefactor. $A(\mathbf{x})$ is a constant in the metadynamics [5] while $A(\mathbf{x})$ declines with increasing $U_{\mathrm{bias}}(\mathbf{x},t)$ in the well-tempered metadynamics (WTM) [6]. There are many different modifications of metadynamics which can be found in an excellent review [44]. Since nuclear forces control atomic motions in MD simulations, directly applying a biasing forces instead of a biasing potential leads to the adaptive biasing force (ABF) method [9]. In ABF, mean forces $\mathbf{F}(\mathbf{q})$ felt by CVs are recorded and biasing forces $\mathbf{F}_{\mathrm{bias}} = -\mathbf{F}(\mathbf{q})$ are applied to drive the system out of a stable/metastable conformation [9]. There are also methods applying potentials to restrain a system as well as to filling up minima on the FES. A method, named as "well-sliced metadynamics" that unifies both two types of biasing potentials has also been proposed [38]. In this method, a restraint potential has been applied to enforce the simulated system explore interesting conformations, while a biasing potential of metadynamics-type has also been used to enhance barrier crossing. In "funnel metadynamics" which is useful to study the drug binding problem, a funnel-like restraint potential has been designed to confine a drug molecule's diffusion if the drug molecule is too far away form the protein surface, and a biasing potential of metadynamics-type can enhance sampling of the binding process [45].

As mentioned above, another approach to bias the Boltzmann distribution is to increase the simulation temperature. A naive way of increasing the whole system's temperature is not practical. However, increasing the simulation temperature for CVs is feasible, and it is the core idea of the adiabatic free energy dynamics (AFED) [46], the driven adiabatic free energy dynamics (d-AFED) [8], and the temperature accelerate molecular dynamics (TAMD) [7]. In order to avoid strong non-equilibrium effects, large masses are assigned to CVs to create an artificial time-scale separation between CVs and other degrees of freedom. It is also possible to bias the Boltzmann distribution by applying both a biasing potential and a high temperature to CVs, like the unified free energy dynamics [16]. We want to emphasize that we only mention a few CV-based enhanced sampling methods in order to motivate further discussions. For more completed and detailed introductions to CV-based enhanced sampling, please refer to following outstanding reviews [47,48].

Although trajectory-based enhanced sampling is not the main topic, we want to briefly introduce the ideas of trajectory-based enhanced sampling with three illustrative methods. Trajectories, like configurations, can form an ensemble known as the path ensemble, assigning path probabilities to trajectories in the ensemble [18]. Sampling paths in the path ensemble leads to accurate estimations of kinetic properties, e.g. transition probabilities, correlation functions, and rate constants [18–

25]. In transition path sampling [18,19], a initial path is first proposed to connect two conformations A and B. The path is described by a series of "beads" with phase space coordinates $\{(\mathbf{x}_1, \mathbf{p}_1), \ldots, (\mathbf{x}_N, \mathbf{p}_N)\}$ where $(\mathbf{x}_1, \mathbf{p}_1)$ stay in conformation A and $(\mathbf{x}_N, \mathbf{p}_N)$ stay in conformation B. A Monte Carlo approach named as "shooting move" [49] has been proposed to update the path, analogy to updating coordinates in a Monte Carlo sampling. In transition interface sampling [20], N interfaces are aligned between conformations A and B. The rate constant from A to B can be evaluated by fluxes of trajectories through interfaces and transition probabilities of crossing a interface. Similarly, the milestoning method [25,50] divides the configuration space to cells. Short MD simulations are performed in each cell to evaluate the mean first passage time for the system to leave one cell. These mean first passage times are then used to evaluate other kinetic properties. While the main object of enhanced sampling focusing on thermodynamic properties is the invariant probability $p(\mathbf{x})$, the main object of trajectory-based enhanced sampling is the transition probability $p(\mathbf{x}', t'|\mathbf{x}, t)$ since the transition probability determines path probabilities. Trajectory-based enhanced sampling methods like milestoning are able to recover the invariant probability for free, as long as the transition probability is evaluated in these methods. Trajectory-based sampling methods are also strongly related to CVs. (1) Information provided by trajectory-based sampling methods can be used to identify appropriate CVs. One famous example is the committor analysis from transition path sampling [51]. (2) Predefined CVs are needed for some trajectory-based sampling methods. For example, cells used in the milestoning method are usually defined with CVs [52,53].

Although CV-based enhanced sampling methods have been widely used in chemistry, biochemistry and materials science, there are two fundamental issues limiting applications of these methods. First, it is an open question on how to construct optimal CVs. Second, accurately generating a multidimensional FES is still challenging. Fortunately, recent developments on machine learning techniques bring a revolution to CV-based enhanced sampling methods. In the next section we shall introduce some basic concepts in machine learning and we will discuss how to develop and apply machine learning techniques to lift obstacles in enhanced sampling methods in Sects. 4 and 5.

3 Introduction to machine learning

In physical sciences, new theories are usually established based on inferences. Starting from fundamental assumptions or experimental results, step-by-step inferences lead to conclusions, rules, and theories. These conclusions, rules, and theories are further applied to new problems to explain observed physical phenomena and to predict unknown experimental results. Besides inference-based approaches, another way to explain existing data and to predict new results focus on studying data directly with theories and algorithms from statistics, optimization, computer sciences, and so on. The second approach is known as "machine learning". "Data" is the most important component of machine learning. In general, there are three different types of data: (1) a static set of data with labels, e.g. a set of data $\{(x, f(x))\}$ where $f(x)$ serves as the label of x; (2) a static set of data without labels, e.g. a set of data $\{x\}$; (3) data depending on the on-the-fly execution of a program or an experiment. Three different types of data leads to three different types of machine learning methods: supervised learning, unsupervised learning, and reinforcement learning. In this section, we shall adopt the linear regression method which is one of the most elementary supervised learning method to illustrate how to train a model, how to validate a trained model, and how to predict with a trained model. We want to emphasize that we are only presenting the outline of the linear regression theory. For details and further discussions, please refer to [54].

In a data set $\mathcal{X} = \{(x, y)\}$ with N samples ("training data set"), y is related to x with $y_i = f(x_i) + \varepsilon_i$ where $f(\cdot)$ is an unknown function and ε_i is a random noise. The linear regression model assumes that the unknown function can be represented as a linear combination of M basis functions $\phi_1(x), \ldots, \phi_M(x)$, i.e.

$$y = \sum_{i=1}^{M} w_i \phi_i(x) = \mathbf{w}^\top \boldsymbol{\phi}(x), \tag{8}$$

where $\mathbf{w}$ is a vector of expansion coefficients. Equation (8) is named as a "model" with adjustable ("trainable") parameters $\mathbf{w}$. To recover $f(x)$, $\mathbf{w}$ has to be optimized such that $\mathbf{w}^\top \boldsymbol{\phi}(x_i)$ matches y_i reasonably well for every data point. This idea leads to the least squares objective function or the least squares loss function.

$$L(\mathbf{w}, \mathcal{X}) = \sum_{i=1}^{N} (\mathbf{w}^\top \boldsymbol{\phi}(x_i) - y_i)^2 \ . \tag{9}$$

"Training" the linear regression model means minimizing L. Once L is minimized, the optimal $\mathbf{w}$, named as $\mathbf{w}^*$, is given by

$$\mathbf{w}^* = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{y}, \tag{10}$$

where $\boldsymbol{\Phi}_{ij} = \phi_j(x_i)$. With a new input x^*, the trained linear regression model predicts an output $y^* = \mathbf{w}^{*\top} \boldsymbol{\phi}(x^*)$. An obvious question with this linear regression model is how to choose an optimal M. If M is too small, the model is not flexible enough to approximate $f(\cdot)$ accurately, resulting in systematic errors in predictions. On the other hand, if M is too large, the model is overfitted and noisy predictions are generated. In the scenario of overfitting, the training error given by $L(\mathbf{w}^*, \mathcal{X})$ is very small. However, a test error $L(\mathbf{w}^*, \tilde{\mathcal{X}})$ can be very large. $\tilde{\mathcal{X}} = \{(\tilde{x}, \tilde{y})\}$ is another data set ("test data set") independent of $\mathcal{X}$. It is because that the magnitudes of some w_i become artificially large in order to learn the noise ε. To solve this problem, it is often necessary to restrain the magnitude of $\mathbf{w}$ by adding a regularization term to L, i.e.

$$L(\mathbf{w}, \mathcal{X}) = \sum_{i=1}^{N} (\mathbf{w}^\top \boldsymbol{\phi}(x_i) - y_i)^2 + \lambda \mathbf{w}^\top \mathbf{w}. \tag{11}$$

where λ is called a "hyperparameter". λ determines the ratio between the systematic error introduced by the regularization term and the random error due to overfitting. λ is not a trainable parameter and cross-validation [55,56] is required to find out an optimal value of λ.

The linear regression method can also be interpreted with probability theory. Assuming that the noise ε is distributed as a normal distribution with zero mean and variance σ, a probability $p(y|x, \mathbf{w}, \sigma)$ can be defined as $p(y|x, \mathbf{w}, \sigma) = \mathcal{N}(y|\mathbf{w}^\top \boldsymbol{\phi}(x), \sigma)$. If all data points in the training data set are independent, a "likelihood" probability is defined as

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \sigma) = \prod_{i} \mathcal{N}(y_i|\mathbf{w}^\top \boldsymbol{\phi}(x_i), \sigma), \tag{12}$$

which tells us the probability to observe $\mathbf{y} = (y_1, \dots, y_N)^\top$, given a set of parameters $\mathbf{w}$ and inputs $\mathbf{x} = (x_1, \dots, x_N)^\top$. A large likelihood probability means that we have a better chance to observe $\mathbf{y}$, therefore, $\mathbf{w}$ is optimized if we maximize $p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \sigma)$. Such an approach is named as "maximum likelihood". Logarithm of the likelihood probability leads to the least squares loss function given by Eq. (9).

Both the least squares approach and the maximum likelihood approach result in a single optimal $\mathbf{w}^*$. On the contrary, the Bayesian approach returns a model that tells us the probability of $\mathbf{w}$. The Bayesian approach of linear regression starts from a famous relation:

$$p(\mathbf{w}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y})}, \tag{13}$$

where $p(\mathbf{w})$ as a "prior" distribution describes our prior knowledge of $p(\mathbf{w})$ and $p(\mathbf{w}|\mathbf{y})$ is called a "posterior" distribution. Usually, a prior distribution is an elementary probability distribution, e.g. $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda^{-1}\mathbf{I})$. Using this prior distribution together with the likelihood in Eq. (12), the logarithm of the posterior distribution is

$$\log p(\mathbf{w}|\mathbf{y}) = -\frac{1}{2\sigma} \sum_{i=1}^{N} (\mathbf{w}^\top \boldsymbol{\phi}(x_i) - y_i)^2 - \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}, \tag{14}$$

where we discard the normalization constant. Therefore, maximizing the posterior which tells us the most possible choice of $\mathbf{w}$ is exactly the same as minimizing the loss function Eq. (11). The similarity between Eqs. (11) and (14) suggests that a prior distribution behaves like a regularization term: our prior knowledge prevents $\mathbf{w}$ from taking crazy values if there is not enough training data. However, the power of the Bayesian approach is far beyond adding a regularization term. Given a new input x^*, the Bayesian approach can predict the distribution of y^* which is given by

$$p(y^*|x^*, \mathbf{y}) = \int p(y^*|x^*, \mathbf{w}) p(\mathbf{w}|\mathbf{y}) \mathrm{d}\mathbf{w}. \tag{15}$$

Equation (15) provides a full statistics of a prediction including mean and confident interval.

Up to now, we have focused on the elementary linear regression model to introduce various machine learning concepts. Besides linear regression, we also want to briefly introduce the artificial neural network model [57–59] that is commonly used in recent years [60]. An artificial neural network shares a layered structure. We define $\mathbf{x}^{(i)}$ as the i'th layer's inputs. The i'th layer's outputs, $\mathbf{y}^{(i)}$, are given by

$$\mathbf{y}_j^{(i)} = f\left(\sum_k \mathbf{W}_{jk}^{(i)} \mathbf{x}_k^{(i)} + \mathbf{b}_j^{(i)}\right), \tag{16}$$

where $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ are trainable parameters called weights and biases. An activation function $f(\cdot)$ is a non-linear function to introduce non-linearity in the model, typical choices of $f(\cdot)$ include sigmoid function, hyperbolic tangent function, and rectified linear unit (ReLU) [61]. $\mathbf{y}^{(i)}$ are then fed into the $(i+1)$'th layer as inputs. A simple neural network contains only three layers: an input layer, a hidden layer, and an output layer. Nowadays, deep neural network can have 10^2–10^3 layers [62]. Besides the most elementary neural network form introduced above, there are many other important architectures of artificial neural networks including the convolutional neural network [63], the recurrent neural network [59], the residual neural network [62], the graph neural network [64], and so on. Training a neural network is usually achieved by the backward propagation algorithm [59,65,66].

4 From physics-intuited CVs to machine learning-based CVs

Designing CVs is a key step when applying a CV-based enhanced sampling method. Traditionally, physically intuited CVs are used in enhanced sampling simulations. A physically intuited CV either corresponds to an experimentally measurable property or it is designed by understanding underlying physics of the studied process. For example, end-to-end distance is related to mechanical pulling experiments of biomolecules [67]. and radius-of-gyration is designed to describe the "compactness" of a molecule [68]. However, applying sampling methods to study complex biomolecules and materials with limited number (≤ 3) of physically intuited CVs is very challenging. Sampling a complex system efficiently may require 10–100 or more physically intuited CVs [27,69,70]. It is because that many conformational transitions could happen when simulating a complex system, while one physically intuited CV is appropriate to describe only a few conformational changes. A limited number of physically intuited CVs are not enough to represents all of these conformational transitions, and these CVs are not enough to map all important transition barriers explicitly on the FES [71]. Since CV-based enhanced sampling methods are only capable of enhancing barrier crossing for barriers on the FES, barriers not on the FES can significantly reduce the efficiency of a CV-based enhanced sampling algorithm [71] or even reduce the accuracy of sampling results [72]. In [71], an alanine dipeptide molecule was simulated with metadynamics using the radius-of-gyration and the number-of-hydrogen bond as CVs. Barriers connecting $\mathrm{C7_{eq}}$ and $\mathrm{C7_{ax}}$ (see Fig. 1, panel b) were not explicitly mapped on the FES, resulting in a long simulation time for the alanine dipeptide molecule to leave the $\mathrm{C7_{ax}}$ conformation. In [72], linear-response theory was applied to evaluate the systematic error of a FES generated by the TAMD method. The error which is caused by the non-equilibrium factor in TAMD is related to correlation functions of forces felt by CVs and barriers not explicitly mapped on the FES slow down the decay of correlation functions, resulting in a large systematic error of the FES.

The motivation to use physics-intuited CVs is that the corresponding FES is physically meaningful. Nevertheless, it is possible to simulate a complex system with sophisticated CVs even if the physical meaning of the FES is insignificant [73,74]. In this case, generating a FES is not the main subject of the simulation. Instead, sample weights are evaluated by unbiasing algorithms [75–83], and unbiased samples are projected onto physics-intuited CVs in post-analysis [71]. Without requiring physics intuited CVs, a lot of methods have been developed to construct CVs directly from mining simulation data (configurations) [71,73,84–96], which greatly enriches the CV library. Designing CVs which aims to find a low-dimensional representation from high-dimensional configurations exactly matches the task of dimensionality reduction algorithms. Dimensionality reduction methods have been widely studied in the machine learning community. The motivation to develop dimensionality reduction methods is an assumption that data points stays around a low-dimensional manifold even if the data points are embedded in a high-dimensional space. The panel (a) of Fig. 2 presents a set of two-dimensional data points. The points concentrate around a curve (one-dimensional manifold) instead of being randomly scattered across the two-dimensional space. Therefore, it is possible to find out a one-dimensional representation of the data points without prior knowledge of the curve, which is the goal of dimensionality reduction methods. Examples of dimensionality reduction algorithms include principle component analysis (PCA) [97], isomap [98], locally linear embedding (LLE) [99], diffusion map [100], t-distributed stochastic neighbour embedding [101] and many others [102]. There are various studies applying different dimensionality reduction algorithms to construct CVs [92–94,103,104]. However, most dimensionality reduction algorithms only use information of the data probability distribution, while configurations from a MD simulation also represent simulation kinetics. It is often believed that kinetics provides the key information for CV design. For example, the isosurface of a good CV should be an approximation of the committor isosurface which is determined by kinetic information [105,106]. Therefore, it is natural to utilize kinetic information to construct a machine learning-based CV. In practice, other criteria like preserving structure similarity are also used in training machine learning-based CVs. In the next section we shall present several example to train CVs from MD simulations.

Principle component analysis (PCA) We start our discussions from the principle component analysis, which is an elementary dimensionality reduction algorithm. PCA attempts to decompose the sample covariance matrix and to find out directions with large variants. As shown in the panel (a) of Fig. 2, v_1 is the eigenvector of data covariance matrix with the larger eigenvalue. Therefore, v_1 could represent a “flexible mode” while v_2, the eigenvector with the smaller eigenvalue, may represent a vibrational mode of less interest. Therefore, v_1 is a more suitable CV compared to v_2. PCA is a very simple approach to train CVs, and its drawbacks are obvious. The CV v_1 is a linear combinations of coordinates x_1 and x_2. However, panel (a) of Fig. 2 clearly shows that the actual low-dimensional manifold is non-linear. Since linear PCA is not able to accurately model a non-linear manifold, a non-linear dimensionality reduction method is required for constructing CVs.

Sketch map The second direction to develop machine learning-based CVs is based on preserving the structural similarity. In most cases, Cartesian coordinates $\mathbf{x}$ of the studied molecule, like a biomolecule, are clustered around stable conformations. In order to preserve clustering information, sketch map trains CVs based on non-linear distance matching with the following objective function:

$$L = \sum_{ij}(F(\|\mathbf{x}_i - \mathbf{x}_j\|) - f(\|\mathbf{q}_i - \mathbf{q}_j\|), \qquad (17)$$

where both $F(\cdot)$ and $f(\cdot)$ are switch functions [73,74]. $F(\|\mathbf{x}_i - \mathbf{x}_j\|) \approx 1$ if $\|\mathbf{x}_i - \mathbf{x}_j\|$ is smaller then a typical cluster size and $F(\|\mathbf{x}_i - \mathbf{x}_j\|)$ becomes 0 if $\mathbf{x}_i$ and $\mathbf{x}_j$ are far apart. $f(\cdot)$ is similar with $F(\cdot)$. L is small only if a cluster of configurations stay within one cluster in the low-dimensional CV-space. Sketch map has been used to identify stable conformations from enhanced sampling simulations [73,74,107,108].

t-Distributed stochastic neighbor embedding (t-SNE) Besides measuring similarity between two samples with distance, probability distributions are also used to describe sample similarity. This idea leads to a method named as “stochastic neighbor embedding (SNE)” [109]. In SNE, a conditional probability is first defined,

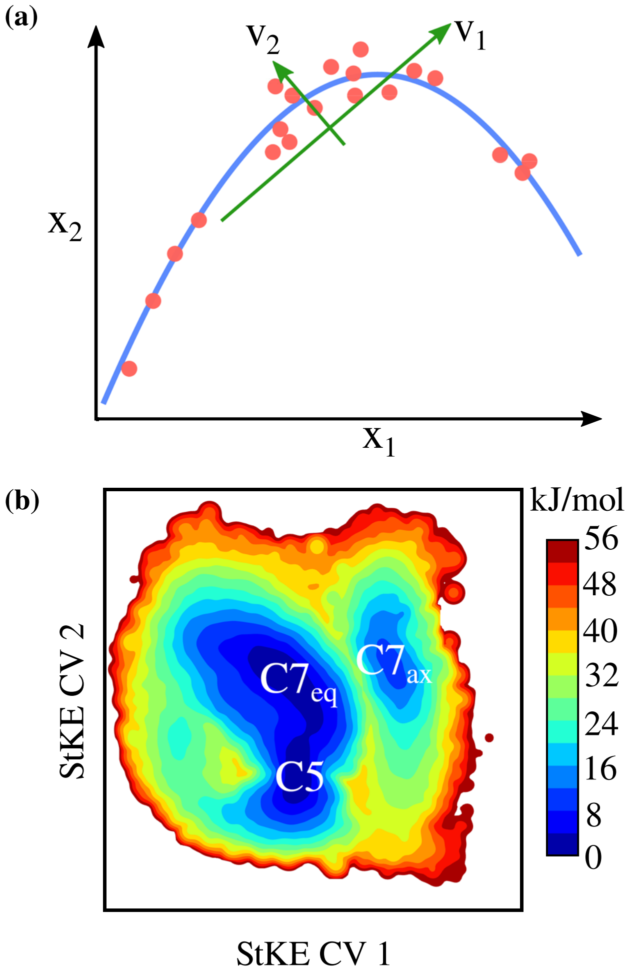

Fig. 2 Panel **a** illustrates the dimensionality reduction problem. Red two-dimensional points are scattered around a blue curve which is unknown to a dimensionality reduction algorithm and the goal of the dimensionality reduction algorithm is to recover the blue curve. Green arrows are corresponding to eigenvectors of the sample covariance matrix. The longer arrow, $\mathbf{v}_1$, is the eigenvector with the larger eigenvalue, while the shorter arrow, $\mathbf{v}_2$, is the eigenvector with the smaller eigenvalue. Panel **b** shows the FES corresponding to two StKE CVs [71]. Configurations are sampled by a $\sim$ 7 ns active enhanced sampling simulation with the OPLS-AA force field [36]

i.e.

$$p^h_{j|i} = \frac{K^h(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k\neq i} K^h(\mathbf{x}_i, \mathbf{x}_k)}, i \neq j$$
$$p^h_{i|i} = 0, \qquad (18)$$

where $K^h(\cdot,\cdot)$ is a symmetric kernel function. A joint probability is then defined as $p^h_{ij} = \frac{p^h_{i|j}+p^h_{j|i}}{2N}$ where N is the number of samples. The design of the joint distribution ensures every sample contributes significantly to the loss function. Similar joint probabilities p^l_{ij} are defined with respect to a low-dimensional representation $\mathbf{q}$, i.e. $p^l_{ij} = \frac{p^l_{i|j}+p^l_{j|i}}{2N}$, where $p^l_{j|i} = \frac{K^l(\mathbf{q}_i,\mathbf{q}_j)}{\sum_{k\neq i} K^l(\mathbf{q}_i,\mathbf{q}_k)}$ for $i \neq j$ and $p^l_{i|i} = 0$. The Kullback–Leibler (K–L) divergence between p^h_{ij} and p^l_{ij} is minimized to preserve data similarity, i.e. the objective function of SNE is

$$L = \sum_{ij} p^h_{ij} \frac{p^h_{ij}}{p^l_{ij}} \equiv D_{\mathrm{KL}}(p^h \| p^l), \qquad (19)$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the K–L divergence. In SNE, both K^h and K^l are Gaussian kernels. However, Gaussian kernels suffer from the “curse of dimensionality”. The distance between two high-dimensional data points is usually much longer then the distance between their low-dimensional projections. Therefore, a fat tail kernel has been used to faithfully preserve the distance between two moderately separated high-dimensional samples. The idea of using mismatched kernel tails to compensate mismatched dimensionality leads to t-SNE in which K^l is a Student's t-distribution [101]. t-SNE has been applied to train a low dimensional representation of configurations to analysis MD trajectories [104].

Autoencoder An autoencoder model contains an encoder and a decoder. The encoder is a function f_e that maps high-dimensional inputs $\mathbf{x}$ to low-dimensional latent variables $\mathbf{q}$ while the decoder f_d decodes $\mathbf{q}$ back to high-dimensional outputs $\mathbf{x}'$ [110]. Nowadays, both f_e and f_d are usually represented by neural networks. The encoder is a natural choice of CVs: $\mathbf{x}$ represent Cartesian coordinates or high-dimensional features of a configuration while $\mathbf{q}$ are low-dimensional CVs. There is no unique way to design an encoder and we will introduce several examples.

In the first example, an autoencoder has been trained by optimizing the following loss function:

$$L = \sum_n \|\mathbf{x}_n - f_d(f_e(\mathbf{x}_n))\|^2 + \text{Regularization}, \qquad (20)$$

where the regularization term is usually a summation of the L^2 norm of parameters in f_d and the L^2 norm of parameters in f_e. Equation (20) minimizes the difference between an encoder's inputs and the corresponding outputs of the decoder [91]. In practice, $\mathbf{x}$ could be atomic Cartesian coordinates or internal coordinates of the studied molecule. If $\mathbf{x}$ are internal coordinates which are invariant under rigid rotation, Eq. (20) can be applied directly. However, further modifications of Eq. (20) are needed to train rotation-invariant CVs if Cartesian coordinates are used [91] as the encoder's inputs. The trained encoder can be used to indicate regions in $\mathbf{x}$ space that are already been sampled by MD simulations. Umbrella sampling simulations are then employed with restraint potentials located at boundaries of the sampled regions. Configurations from these simulations further expand the sampled regions. The iterations continue until the sampling process converges [91]. In the second example, the objective function to train an autoencoder is a linear combination of coordinate-matching objective function (Eq. 20) and sketch-map objective function (Eq. 17) [111]. This work further demonstrates that the trained decoder is capable of generating all-atom structures from latent variables. In the third example, an autoencoder model has been developed by integrating kinetic information, e.g. the input of encoder is $\mathbf{x}$ at time t and the output of decoder is $\mathbf{x}'$ at time $t+\tau$ [112].

A recent development of autoencoder theory is based on the Bayesian theory. This autoencoder method, named as "variational autoencoder" is a generative model to generate $\mathbf{x}$ following a probability distribution $p(\mathbf{x})$ [110]. In the variational autoencoder model, an "Evidence Lower Bound" (ELBO) [54] loss function is optimized, i.e.

$$L=\sum_i D_{\mathrm{KL}}(\tilde{p}_\phi(\mathbf{q}_i|\mathbf{x}_i)\|p(\mathbf{q}_i))-\mathbb{E}_{\tilde{p}_\phi(\mathbf{q}_i|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x}_i|\mathbf{q}_i)\right], \tag{21}$$

where $\tilde{p}_\phi(\mathbf{q}|\mathbf{x})$ approximates $p(\mathbf{q}|\mathbf{x})$ and $p(\mathbf{q})$ is a prior distribution of $\mathbf{q}$ which is usually a standard normal distribution. The first term of Eq. (21) is a regularization term to make sure $\tilde{p}_\phi(\mathbf{q}|\mathbf{x})$ staying close to $p(\mathbf{q})$. Minimizing the second term of Eq. (21) attempts to match the decoder outputs with the encoder inputs, i.e. the decoder has the largest probability to output $\mathbf{x}$ if this $\mathbf{x}$ are the encoder's inputs. In practice, $\tilde{p}_\phi(\mathbf{q}|\mathbf{x})$ is a normal distribution whose mean and standard deviation are outputs of the encoder neural network. $p_\theta(\mathbf{x}|\mathbf{q})$ is another normal distribution with mean generated by the decoder neural network and standard deviation as hyperparameters. The decoder can be easily adopted as a configuration generator [95], while applying the encoder as CVs is not straightforward. Variational autoencoder suggests that $\mathbf{q}$ is a random variable instead of a deterministic function of $\mathbf{x}$. In order to use $\mathbf{q}$ in a enhanced sampling method like metadynamics, an extra fitting step has been designed to train a deterministic function $\mathbf{q}_d(\mathbf{x})$ such that the K–L divergence between the probability of $\mathbf{q}_d$ and the marginal probability distribution of $\mathbf{q}$ is minimized [86].

Classification neural network Recently, artificial neural networks have achieved great successes in classification problems. In a classification problem, e.g., recognizing objects in an image, the image is fed into a neural network and the neural network returns probabilities of assigning different classes to a pattern on the image. Similarly, local order parameters are also "classifiers" that indicate whether a local structure of a solid state material belongs to a fcc structure, a bcc structure, a disordered structure or other structures. A classification deep neural network which outputs probabilities of assigning crystal structure classes performs as a local order parameter $\mathbf{q}$ [96]. A global order parameter is then defined as $\mathbf{Q}=\frac{1}{N}\sum_i \mathbf{q}_i$, where i loops over N atoms. $\mathbf{Q}$ has been applied successfully to study solid state phase transitions [96].

Time-lagged independent component analysis (TICA) As discussed above, preserving kinetic information is an important guideline of CV design. One successful approach to train CVs with kinetic information is "Time-lagged Independent Component Analysis" (TICA). In TICA, a time correlation matrix $\mathbf{C}$ is built with a set of predefined high-dimensional trail CVs, i.e. $\mathbf{C}_{ij}(\tau)=\langle q_i^h(\mathbf{x}(t))q_j^h(\mathbf{x}(t+\tau))\rangle$ where τ is a lag time [113,114]. The motivation of TICA is obvious: optimal CVs should represent slow modes which are characterized by slow-decay correlation functions. By solving the generalized eigenvalue problem $\mathbf{C}(\tau)\tilde{q}=\lambda\mathbf{C}(0)\tilde{q}$, an optimal CV is the eigenvector with the largest eigenvalue. The lag time τ should be selected to distinguish fast and slow modes. TICA is an important tool to construct a Markov state model [88,115–117] and TICA CVs trained from MD simulations are able to accelerate metadynamics simulations [89]. With proper approaches to unbias correlation functions from biased enhanced sampling, it is also possible to use biased enhanced sampling trajectories to train TICA CVs [118].

Past-future information bottleneck (PIB) Studying correlation functions is not the only way to utilize kinetic information, such information can also be recovered by directly investigating the relationship between $\mathbf{x}(t)$ and $\mathbf{x}(x+\Delta t)$. Following this idea, CVs are constructed by the principle of past-future information bottleneck [87,119]. In past-future information bottleneck, CVs $\mathbf{q}(\mathbf{x})$ are trained by maximizing an objective function

$$L=I(\mathbf{q}(\mathbf{x}(t)),\mathbf{x}(t+\Delta t))-\gamma I(\mathbf{x}(t),\mathbf{q}(\mathbf{x}(t))), \tag{22}$$

where $I(\mathbf{u},\mathbf{v})$ is mutual information between two random variables $\mathbf{u}$ and $\mathbf{v}$. Maximizing $I(\mathbf{q}(\mathbf{x}(t)),\mathbf{x}(t+\Delta t))$ means $\mathbf{q}(\mathbf{x}(t))$ contains as much information as possible to predict the future states $\mathbf{x}(t+\Delta t)$ while minimizing $I(\mathbf{x}(t),\mathbf{q}(\mathbf{x}(t)))$ keeps the form of $\mathbf{q}$ as simple as possible [87].

Diffusion map It is also possible to approximate kinetic information, like transition probabilities, with the Boltzmann distribution. $\mathcal{L}$, the infinitesimal generator of Eq. (3), describes dynamical behaviors of a system, i.e. the time-dependent probability density $\rho(\mathbf{x},t)$ becomes

$$\rho(\mathbf{x},t)=\rho(\mathbf{x})+\sum_i c_i e^{-\lambda_i t}\psi_i(\mathbf{x})\rho(\mathbf{x}), \tag{23}$$

where $\lambda_i>0$ and ψ_i is the i'th eigenvalue and the i'th eigenfunction of $\mathcal{L}$ and $\rho(\mathbf{x})$ is the Boltzmann distribution. c_i are determined by the initial probability distribution. Equation (23) suggests that $\psi(\mathbf{x})$ with small λ are "slow" degrees of freedom. Thus eigenfunctions $\psi(\mathbf{x})$ with small λ become natural choices of CVs [94,120]. However, computational costs of solving the eigenfunction problem of $\mathcal{L}$ are prohibitively high for realistic systems. Fortunately, diffusion map provides an alternative approach to approximately solve the eigenproblem of $\mathcal{L}$ [100,121].

Assuming a set of sample $\{\mathbf{x}_i\}$ generated from Eq. (3), a kernel matrix is defined as $K_{ij}=G(\mathbf{x}_i,\mathbf{x}_j)$ where $G(\mathbf{x},\mathbf{y})$ is a Gaussian kernel function with broadening

σ. The kernel matrix is then scaled by the Boltzmann distribution, i.e. $D_{ij} = K_{ij}/(\sqrt{\rho(\mathbf{x}_i)}\sqrt{\rho(\mathbf{x}_j)})$. Finally, D_{ij} is normalized to form a transition matrix: $M_{ij} = D_{ij}/(\sum_k D_{ik})$. In the limit of infinite samples and $\sigma \to 0$, the right eigenvectors of M weakly converges to the eigenfunction of $\mathcal{L}$ [100,121]. Diffusion map has been used to analysis MD simulation data [94] and diffusion-map-based enhanced sampling methods have also been developed [120].

Spectral gap optimization of order parameters (SGOOP) The second example of training kinetics-based CVs with thermodynamic properties is spectral gap optimization of order parameters (SGOOP) [84]. In SGOOP, grids are first built in the low-dimensional CV-space. The time-dependent probability of CVs on the n'th grid point, $p_n(t)$, is propagated with

$$\frac{\partial p_n}{\partial t} = \sum_m K_{mn} p_n, \tag{24}$$

where m and n loop over all grids in the CV-space. A transition probability Ω is estimated by $\Omega = \exp\{\mathbf{K}\delta t\} \approx \mathbf{I} + \mathbf{K}\delta t$ where δt is a small time interval. Ω is generated by the maximum caliber approach, i.e. maximizing entropy of microscopic paths with physical constraints [122]. The entropy is defined as

$$S = -\sum_{mn} p_n \Omega_{mn} \log \Omega_{mn}. \tag{25}$$

A path-dependent physical observable A discretized on grids is denoted as A_{mn}. Optimal Ω_{mn} can be obtained by maximizing S defined in Eq. (25) with a constraint that the path-ensemble averaging of A_{mn} equals a certain value. It has been shown that solving the maximization problem leads to $\Omega_{mn} = \sqrt{\frac{p_n}{p_m}} e^{-\lambda A_{mn}}$, where λ is the Lagrange multiplier [123]. In the simplest case, the average times of transitions with $A_{mn} = 1$ is the only constraint physical quantity, leading to $\Omega_{mn} = \sqrt{\frac{p_n}{p_m}} e^{-\lambda}$ which has been used in SGOOP. Ω as well as its eigenvalues ε varies with different CVs. Maximizing the spectral gap $|\varepsilon_\alpha - \varepsilon_{\alpha+1}|$ results in optimal CVs where α represents the number of barriers on the FES [84].

Stochastic kinetic embedding (StKE) Although diffusion map provides an approach to project existing samples, using $\psi_i(\mathbf{x})$ as CVs in enhanced sampling simulations requires explicit function form of $\psi_i(\mathbf{x})$ so that $\nabla\psi_i(\mathbf{x})$ can be evaluated analytically. We will introduce an alternative method, named as stochastic kinetic embedding (StKE) [71], which is based on diffusion map to construct differentiable CVs.

Assuming that a system can be described by a large number of physically intuited CVs $\mathbf{q}^h$ (high-dimensional CVs) with a FES $A^h(\mathbf{s}^h)$ at $\mathbf{q}^h(\mathbf{x}) = \mathbf{s}^h$, StKE approximates dynamics of CVs as a Brownian dynamics, i.e.

$$\tilde{\mu}\mathrm{d}\mathbf{s}^h = -\nabla A^h(\mathbf{s}^h)\mathrm{d}t + \sqrt{2\tilde{\mu}kT}\mathrm{d}W \tag{26}$$

where $\tilde{\mu}$ is an effective friction coefficient. In principle, a generalized Langevin equation should be used to accurately model CV dynamics [124,125]. However, several studies have suggested that Brownian dynamics is a reasonable approximation of CV dynamics in metadynamics simulations [126,127]. StKE aims to learn a low-dimensional representation $\mathbf{s}^l(\mathbf{s}^h;\theta)$, where θ are trainable parameters. Similar to diffusion map, we can build a Markov chain with transition matrix M^h_{ij} from samples $\{\mathbf{s}^h_i\}$. Similarly, we can construct a transition matrix $M^l_{ij} \equiv M^l_{ij}(\{\mathbf{s}^l(\mathbf{s}^h,\theta)\})$ with respect to the low-dimensional representation $\mathbf{s}^l(\mathbf{s}^h;\theta)$. If $\mathbf{s}^l$ is an optimized low-dimensional representation of $\mathbf{s}^h$, M^l_{ij} should be close to M^h_{ij}. Therefore, the K–L divergence has been used to train $\mathbf{s}^l(\mathbf{s}^h;\theta)$, i.e. the loss function becomes

$$L = \sum_{ij} M^h_{ij} \log \frac{M^h_{ij}}{M^l_{ij}}. \tag{27}$$

In practice $\mathbf{s}^l(\mathbf{s}^h;\theta)$ is modeled by a deep neural network and θ are trained by optimizing Eq. (27). Panel (b) of Fig. 2 presents the FES of alanine dipeptide in vacuum with StKE CVs. It is clear that StKE CVs are able to map stable conformations and transition paths explicitly on the FES.

5 Learning FES from enhanced sampling simulations

One of the most important goal of an enhanced sampling simulation is to generate a FES associated with selected CVs. There are different approaches to calculate FES from simulation trajectories according to different enhanced sampling methods. For example, weights can be assigned to configurations from umbrella sampling via the weighted histogram analysis method (WHAM) [79], the multistate Bennett acceptance ratio (MBAR) [80], and the family of transition-based reweighting analysis methods (TRAM) [81–83]. Metadynamics is able to generate a FES either by inverting a biasing potential or by unbiasing samples [5,6,75–77,83]. In d-AFED/TAMD, a FES is calculated with the probability of $\mathbf{s}$ or by fitting mean forces. [7,8,16,128]. In ABF, a FES is fitted by matching mean forces if $d > 1$ [9]. In general, a FES is usually evaluated with CV probability and/or mean forces.

Histogram or kernel density estimation is a common approach to evaluate a probability distribution. With kernel density estimation, the probability at point $\mathbf{s}$ is given by $\rho(\mathbf{s}) = \frac{1}{N}\sum_i \omega_i K(\mathbf{s},\mathbf{s}_i)$, where ω_i is the weight of the sample $\mathbf{s}_i$ and $K(\mathbf{s},\mathbf{s}_i)$ is a symmetric, positive-definite kernel function. A kernel can either have a fixed bandwidth (broadening) or a flexible bandwidth [129–131]. Besides kernel density estimation, there are other machine learning approaches to learn a probability distribution. For example, a mixture model like a mixture of Gaussians can also model a probability distribution. The Gaussian mixture model is usually used as a clustering algorithm with a model probability distribution (likelihood)

$$\rho_{\mathrm{GM}}(\mathbf{s}|\{p\},\{\boldsymbol{\mu}\},\{\boldsymbol{\Sigma}\}) = \sum_{i=1}^{N_C} p_i \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{28}$$

where p_i is a marginal distribution of $\mathbf{s}$ corresponding to the i'th cluster [132]. $\mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a normal distribution function with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$ associated with the i'th cluster. Training a Gaussian mixture model with maximum likelihood results in optimized $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and p_i, as well as probabilities of identifying each sample in all clusters. The log likelihood function of a Gaussian Mixture model is $\sum_i \log \rho(\mathbf{s}_i)$ and maximizing the log likelihood function is equivalent to minimizing the K–L divergence $D_{\mathrm{KL}}(\rho_{\mathrm{samp}}||\rho_{\mathrm{GM}})$, where ρ_{samp} is the sample probability distribution. Therefore, optimizing ρ_{GM} not only classifies samples but also establishes an optimized probabilistic model of ρ_{samp}. In practice, the number of Gaussians used in the Gaussian mixture model can be determined by Bayesian inference criterion which is a criteria for model selection [133]. Gaussian Mixture models have been used in enhanced sampling algorithms to approximate CV distributions [70,133]. Besides Gaussian Mixture models, some generative models are also capable of predicting the probability of a sample besides generating new samples. For example, propagating along one direction of a normalizing flow model [134] generates a random sample while propagating along the other direction of the normalizing flow model returns the probability of an input sample. Detailed discussions about normalizing flow models can be found in the next section.

Another approach to generate a FES is to "integrate" mean forces. If $d = 1$ and Eq. (3) is used, mean force $F(s)$ at $q(\mathbf{x}) = s$ can be evaluated by

$$\begin{aligned} -\frac{dA}{dq}\Big|_{q(\mathbf{x})=s} &\equiv F(s) \\ &= \mathbb{E}\left(-\frac{\partial U}{\partial q} - \frac{\partial \log|J|}{\partial q}\Big| q(\mathbf{x}) = s\right), \end{aligned} \tag{29}$$

where J is the Jacobian matrix of generalized coordinates [11]. Theoretically, applying Eq. (29) requires full knowledge on the Jacobian matrix, which is impracticable for CVs with complex function forms. To simplify Eq. (29), alternative formulas have been proposed in ABF and blue moon sampling [135–137]. The formula to calculate mean force is greatly simplified with the extended Lagrangian framework, i.e.

$$F_{\mathrm{el}}(s) = \mathbb{E}\left(\kappa(q(\mathbf{x}) - s)|s\right) \quad . \tag{30}$$

We want to emphasize that extending Eq. (30) to $d > 1$ is trivial, which is suitable for multi-CV simulations. However, a systematic error exists in F_{el} with finite κ. Decreasing the systematic error requires increasing κ, while increasing κ can significantly enlarge the variance of $\kappa(q(\mathbf{x}) - s)$, leading to a large statistical error. In practice, κ should be tuned to balance the systematic error and the statistical error. Recently, a new mean force estimator, named as "corrected z-averaged restraint (CZAR) estimator", has been developed, which can significantly reduce the systematic error [17].

With the CV probability and/or mean forces, it is possible to construct a FES with various machine learning models. The simplest model is linear regression which expands the FES with a linear combination of basis functions, i.e.,

$$A(\mathbf{s}) = \sum_i C_i \Phi_i(\mathbf{s}), \tag{31}$$

where $\{C_i\}$ are expansion coefficients and $\{\Phi_i(\mathbf{s})\}$ are basis functions. This approach has been applied to metadynamics in the extended Lagrangian framework [128], unified free energy dynamics [16], ABF [9,17] and other methods [41,138]. Linear regression has been widely used due to its quadratic objective function with a unique global minimum. However, this method scales poorly with the number of CVs (dimensionality) d. For example, constructing a four-dimensional FES is already non-trivial [16]. Therefore, applying the linear regression method to learn a high-dimensional FES is not feasible.

Other advanced machine learning methods have been used to train a FES. We will describe several models in this section to introduce basic ideas of training a FES.

Artificial neural network Artificial neural networks have been used in various studies to train FESes due to their capabilities to accurately approximate arbitrary smooth functions with reasonable computational costs [139]. In one study, a neural network has been trained by matching the neural network's gradient with mean forces (force estimator) [140,141]. Fitting a neural network with free energies evaluated at different points in the CV-space (energy estimator) has also been proposed [142]. Although a neural-network-based FES can be trained with either a force estimator or an energy estimator, combing two estimators can significantly improve the accuracy of the trained FES [143]. Besides training a neural-network-based FES as a post-analysis of enhanced sampling simulations, neural networks also serve as biasing potentials in various enhanced sampling methods. For example, an artificial neural network has been used to construct a biasing potential in the variational enhanced sampling method where the converged biasing potential can further be used to evaluate the FES [144]. Within the ABF framework, a neural network has been trained to provide smooth estimations of mean forces [145]. In a reinforcement-learning-based enhanced sampling approach, a neural-network-represented FES can indicate regions in the CV space with insufficient samples to guide MD simulations to explore these regions [141].

Kernel methods Kernel methods such as kernel ridge regression (KRR) and support vector regression have also been tested as possible models to train a FES [142]. Kernel regression is closely related to linear regression by recasting the regression problem with dual formulation [54]. Therefore, kernel regression deals with kernel functions instead of finite basis functions, which allows to implicitly use a large number or even infinite number of basis functions. We will use the KRR as an illustrative example. In KRR, a FES is approximated with

$$A(\mathbf{s}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{s}, \mathbf{s}_i), \tag{32}$$

where i loops over N samples and $K(\cdot,\cdot)$ is a kernel function. Trainable parameters $\boldsymbol{\alpha}$ are determined by $\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{A}$ where $\mathbf{A} = (A(\mathbf{s}_1), \ldots, A(\mathbf{s}_N))^T$ and $\mathbf{K}_{ij} = K(\mathbf{s}_i, \mathbf{s}_j)$. λ, as a hyperparameter, controls the regularization strength. Intuitively, KRR attempts to "interpolate" the free energy at a new location $\mathbf{s}$ by measuring the similarity between $\mathbf{s}$ and each sample $\mathbf{s}_i$ with the kernel function and by assigning an appropriate weight α_i to $K(\mathbf{s}, \mathbf{s}_i)$.

It is also possible to establish a kernel method via the Bayesian theory. One famous example is the Gaussian process regression (GPR) method [54]. In GPR, a Gaussian process is used as a function prior. Gaussian process is a stochastic process $y(t)$ such that the joint distribution of $\{y(t_1), \ldots, y(t_n)\}$ is a multivariate normal distribution with any $t_1, \ldots, t_n$, i.e.

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \lambda\mathbf{I}), \tag{33}$$

where $\mathbf{K}_{ij} = K(t_i, t_j)$ with $K(\cdot,\cdot)$ as a kernel function and $\mathbf{I}$ is an identity matrix. λ represents the precision of data noise. Without loss of generality, we assume that there is a data set of free energies $\mathbf{A} = (A_1, \ldots, A_n)^\top$ at $\mathbf{s} = (s_1, \ldots, s_n)^\top$. If we want to estimate the free energy A^* at a new location s^*, the joint probability $p(A^*, \mathbf{A})$ follows Eq. (33) and the conditional probability $p(s^*|\mathbf{s})$ becomes another normal distribution with mean μ^* and covariance σ^* given by

$$\mu^* = \mathbf{k}^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{A} \tag{34a}$$

$$\sigma^* = K(x^*, x^*) + \lambda - \mathbf{k}^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}, \tag{34b}$$

where $\mathbf{k}_i = K(s^*, s_i)$. The averaged prediction (Eq. 34a) is the same as a KRR prediction. Besides generating an averaged prediction, GPR also provides the confidence of a prediction. The GPR method has been used to train a FES [138] with free energies as well as mean forces [138].

6 Challenges and perspective

6.1 Kinetics from biased enhanced sampling simulation

Usually, exact kinetic information like transition probabilities or correlation functions are calculated from unbiased MD simulations with methods like Markov state models [146,147] or trajectory-based enhanced sampling methods like milestoning [25,50]. Evaluating exact kinetics from biased enhanced sampling simulations is attractive as it opens a door to introduce highly efficient biased enhanced sampling approaches to build kinetic models. Moreover, exact kinetics is needed for some CV construction methods like TICA [113,114]. Since most biased enhanced sampling methods are only designed to calculate thermodynamic properties, efforts are needed to develop efficient and accurate algorithms to unbias dynamical information. In WTM, unbiased correlation functions can be evaluated by a change of variable on time [76,118]. Time is "compressed" in metadynamics since a biasing potential accelerates the simulation. A Properly designed change of variable on time can recover the unbiased timescale. This approach has been applied to constructing TICA CVs from metadynamics simulations [118]. However, the formula of the change of variable is asymptotic and it is only valid in the long time limit [76,118]. Another way to obtain correct kinetic information with metadynamics is to unbias the dynamics of metadynamics with transition state theory [35]. In this approach, Gaussians are deposited less frequently to avoid biasing the transition state ensemble during a simulation and unbiasing is achieved by reweighing the partition function of reactant conformations. The accuracy of this method depends on the frequency of Gaussian deposition, which requires careful tuning and testing [148]. A more general approach to obtain accurate kinetic information is to unbias metadynamics path probabilities with the Girsanov theorem [149,150]. The Girsanov theorem evaluates a change of path probability associated with one diffusion process, while the path itself is generated by another diffusion process. Although this approach does not require any modifications of the metadynamics algorithm, it limits the dynamical equation to be Brownian dynamics [149] or Langevin equation [150], which rules out deterministic thermostats. Also, the unbiasing formula depends on the integrator used in a simulation [150], thus updating the formula is needed if a different integrator is used. Recently, evaluation of path probability with a path integral formula has been integrated to metadynamics with a polymer model [151]. This model requires simulating multiple replicas of a system and it needs further benchmarks on simulating more challenging systems like biomolecules or materials. In summary, methods introduced in this paragraph suggest that it is highly non-trivial to evaluate kinetics from biased enhanced sampling simulations and it requires further developments on both theory and numerical algorithms.

Besides unbiasing kinetics with physical principles, it is also interesting to explore machine learning related theories and methods to learn kinetic properties from biased enhanced sampling. There is a likelihood function to estimate transition matrix $\mathbf{P}$ from a counting matrix $\mathbf{C}$ in Markov state model with unbiased MD simulations, i.e.

$$p(\mathbf{C}|\mathbf{P}) = \prod_i \prod_j P_{ij}^{C_{ij}}, \tag{35}$$

where P_{ij} is the transition probability from the i'th state to the j'th state and C_{ij} is the number of transitions from the i'th state to j'th state [152]. This formula can be extended to equilibrium multiensemble simulations with different temperatures or different biasing potentials [153]. Although this approach has been applied to umbrella sampling, it is not clear whether this approach can be extended to quasi-equilibrium enhanced sampling like metadynamics, d-AFED/TAMD or ABF. Therefore, developing machine learning techniques to learn a kinetics model from quasi-equilibrium enhanced sampling simulations is another open problem.

Up to now we have discussed developments and challenges to obtain unbiased kinetics information from biased enhanced sampling simulations. Due to difficulties of unbiasing kinetics, methods have been proposed to train CVs with approximate kinetics information. Although various approximations of kinetics have been used in different CV-training methods, [71,84,87,94], the trained CVs work quite well in all of these methods. Actually, it is not clear whether a rigorous kinetics model is necessary for training CVs, and the relationship between qualities of CVs and approximations in kinetic models is not clear. It is possible that the capability of trained CVs to represent minimum free energy paths is an important factor to the effectiveness of trained CVs [84]. However, systematic studies are needed to answer this question.

6.2 Exploring a high-dimensional FES

Section 4 introduces various methods to construct CVs as a low-dimensional representation of a system. One key question is how to estimate the dimensionality of the low-dimensional representation, i.e. what is an optimal number of CVs. It has been believed that configurations in $\mathbb{R}^{3N}$ sampled from MD simulations stay closely to a d dimensional manifold where d, named as intrinsic dimensionality (ID), is much smaller than $3N$ [154–156]. Recently an elegant algorithm has been developed to estimate the intrinsic dimensionality d [156]. The idea is to test a ratio $\alpha = r_2^{(i)}/r_1^{(i)}$ where $r_1^{(i)}$ is the radius of i'th sample's nearest neighbour and $r_2^{(i)}$ is the radius of the i'th sample's second nearest neighbour. There exists a relationship between α, the cumulative distribution $F_c(\alpha)$ of α, and d, i.e.

$$\frac{\log(1 - F_c(\alpha))}{\log(\alpha)} = d\,. \tag{36}$$

In practice the empirical cumulative distribution is used to replace $F_c(\alpha)$ and d is recovered with linear regression. This method has been applied to several biomolecule systems. For example, the ID of RNA trinucleotide AAA with 98 atoms is $\sim$ 10 [156]. The ID of a mini protein, FiP35 WW domain, is around 14 [131]. A study on Villin headpiece folding free energy landscape suggests the ID for this system is around 12 [157]. However, there is no guarantee that ID is always around 10 for more complex systems. A recent study on SARS-CoV-2 main protease which is a homodimer with 306 residues in each monomer suggests that the ID is about 27 for each monomer [158]. Therefore, applying enhanced sampling techniques to more challenging problems like protein complexes requires large number of CVs even if these CVs are optimal.

Difficulties to explore a high-dimensional FES may show up in the sampling step or in the FES construction step. For example, d-AFED/TAMD simulations are feasible with many CVs [16,27]. Calculating a free energy difference between two points in the CV space is also trivial. However, unbiasing samples from d-AFED/TAMD simulation is extremely hard with a large number of CVs since it requires full knowledge of the FES [78]. A common implementation of Metadynamics has difficulties to construct biasing potential with ≥4 CVs as the biasing potential is usually stored and evaluated on grids [159]. Similar difficulties also exists with the ABF method [160]. Progress has been achieved to push biased enhanced sampling method working with a high-dimensional FES. For example, machine learning-based biasing potentials have been developed with the Gaussian Mixture model [70], the artificial neural network model [144], and the kernel density estimation method [161]. Similarly, training a high-dimensional FES from d-AFED/ TAMD with various machine learning models has also been tested [142], as introduced in Sect. 5.

Either training an accurate high-dimensional FES or learning an accurate high-dimensional probability is difficult. First, free energies and/or FES derivatives are needed prior to FES training in a supervised learning approach. However, obtaining accurate free energies and derivatives are non-trivial with large d. Samples tends to be "sparse" in a high-dimensional space. For example, assuming a quadratic FES $A(\mathbf{s}) = 1/2\|\mathbf{s}\|^2$ where $\mathbf{s} \in \mathbb{R}^d$, the Boltzmann distribution is simply a standard normal distribution. The distance between two independent samples $\mathbf{s}_1$ and $\mathbf{s}_2$ increases with d on average, i.e. $\mathbb{E}(\|\mathbf{s}_1 - \mathbf{s}_2\|^2) = 2d$. The sparsity suggests that a large bin size or a large kernel bandwidth is required. However, larger bin size or larger kernel bandwidth leads to larger systematic errors in calculating free energies and FES derivatives.

Second, samples on a high-dimensional FES are typically distributed with a "spider web" structure [73]. Clusters of samples are located at local minima on the FES together with free energy paths to connect different minima. However, biased enhanced sampling methods, including metadynamics, d-AFED/TAMD, and ABF, attempt to uniformly or near-uniformly sample the CV-space, which is very challenging on a high-dimensional FES as the volume of the CV-space increases exponentially with the dimensionality d. Therefore, efficiently sampling on a high-dimensional FES requires focusing on exploring minima and transition paths on the FES instead of uniformly converging the FES [69]. For example, from our experiences the d-AFED/TAMD method usually discourages to use extremely high CV temperature with a large number of CVs in order to avoid spending too much time on exploring high-free-energy regions. The trade-off between enhancing barrier crossing and avoiding exploring high-free-energy regions requires fine tuning on sampling methods.

Finally, the "spider web" type of configuration distribution suggests that data for training CVs or biasing potential only cover a small percentage of the CV-space. An enhanced sampling simulation may lead the system to explore regions on the FES that are not supported by data. Applying the machine learning-based biasing potential or machine learning-based CVs in these regions means extrapolating the model which could perform poorly. Therefore, one difficulty of exploring a high-dimensional FES with machine learning methods is to appropriately deal with the extrapolation problem. For example, a biasing potential in the form of $kT\log(\rho(\mathbf{s})+\varepsilon)$ with a shifting constant ε can be used to avoid errors in the trained $\rho(\mathbf{s})$ from damaging MD simulations, especially in the places where $\rho(\mathbf{s})$ is small [71,133]. In another study, Ensemble learning has been applied to avoid applying biasing potential to unexplored conformations [141]. In this approach a biasing potential is switched off if the variance of trained biasing potential is too large. The same extrapolation problem has been discussed in applying StKE CVs [71]. A physically intuited CV was combined with StKE CVs during the enhanced sampling simulation to maintain high sampling efficiency when sampling new conformations. However, systematic studies are still needed to improve the extrapolation capability or the generalizability of the machine learning-based enhanced sampling approaches on a high-dimensional FES.

Besides developing enhanced sampling methods to overcome challenges of exploring a high-dimensional FES, developing theories and methods to unify enhanced sampling and building a coarse-grained (CG) model is another interesting direction. Training a high-dimensional FES is closely related to constructing a coarse-grained (CG) model with some bottom-up approaches [162–165]. In this scenario CG degrees of freedom become CVs while the interaction potential of the CG model is a FES. Since a CG model requires transferability, CG degrees of freedom are usually center of atom groups. Some CG potential fitting method, like force matching [162–164], is very similar to training a FES via mean forces. Recently, machine learning-based CG potentials have been developed with significantly improvement on accuracies [166–169]. However, limited studies have been proposed to unify enhanced sampling and CG model building [169]. There are two possibilities of integrating enhanced sampling and CG model. (1) Use enhanced sampling to build a transferable CG model. In order to transfer the CG model to other systems, a CG degree of freedom should be a local CV that is a function of a group of neighborhood atoms. Moreover, a few-body interaction potential is needed for the CG potential energy function. In a recent study, machine learning-based CG potential energy functions with few-body (two-body to five-body) interactions have been proposed [170]. Samples from CG potential and samples from all-atom simulations have been projected onto TICA CVs to generate a CG FES and an all-atom FES. With the five-body potential, the CG FES agreed with the all-atom FES reasonably well. This study suggests that it is possible to train a CG model with few-body interactions. (2) Use all-atom MD simulations to train a system-specific CG model and apply this model to enhance all-atom MD simulations of the same system. In this case, transferability may not be required. For example, applying the CG potential energy function as a biasing potential [169] can use a CG potential energy function with full-body interactions. However, transferable CG model is still needed if the CG model is designed to predict conformations that have not been sampled.

6.3 Machine learning-based all-atom sampler for CV-based enhanced sampling

The idea of CV-based enhanced sampling is analogous to reinforcement learning [141,169]: a high CV temperature or a biasing potential acts like a policy to guide the exploration of the configurational space while MD serves as an "explorer" to discover new conformations or to recurrently visiting conformations to improve statistics. However, efficiency of a MD simulation is limited. A typical time step of a MD simulation for molecules is about 0.5–2 fs. A nearly independent configuration can be sampled every ∼1 ps, which requires ∼ 1000 times of force evaluations. Massive force evaluations, especially non-bonded force calculations, are the most time-consuming steps in a MD simulation thus these calculations significantly reduce the sampling efficiency. One way to enhance the efficiency of MD simulations is to increase the time step. For example, the time step to evaluate non-bonded interactions can increase up to ∼100fs in a simulation with the isokinetic ensemble [171,172]. However, further improvements on all-atom sampling efficiencies are still needed.

Although there are very limited studies on sampling all-atom configurations with machine learning methods, these developments are changing the world of all-atom sampling [95,111,173–176]. Here we will briefly discuss some methods as interesting directions. We shall start the discussion from the autoencoder method which has been introduced in Sect. 4. While the encoder part is used to construct CVs, the decoder part is able to generate atomic structures for given CV values. We want to emphasize that fixed values of CVs correspond to an constrained ensemble of $\mathbf{x}$. Therefore, the generated structures should be randomly sampled from the ensemble. In one work, the trained variational autoencoder model can generate random structures of alanine dipeptide [95]. However, nonphysical features such as collapsed atoms can be found on generated structures. In another work the decoder was trained by matching a decoder output with the corresponding encoder input [111]. In this case a structure from decoder is more like an "interpolation" of simulated structures with similar CV values. Nevertheless, the generated structures are valuable as initial structures for further refinement.

Besides generating atomic structures from CVs, sampling atomic structures directly is much more challenging. A groundbreaking method, named as "Boltzmann generator", has been proposed by using normalizing flow [173]. Normalizing flow [134] learns a change of variable function $f(\cdot)$ represented by a neural network, i.e. $\mathbf{x} = f(\mathbf{z})$ for $\mathbf{x} \in \mathbb{R}^{3N}$ and $\mathbf{z} \in \mathbb{R}^{3N}$. $f(\cdot)$ has to be a bijection function with a Jacobian matrix $\partial\mathbf{x}/\partial\mathbf{z}$. If $\mathbf{z}$ is distributed as the probability $\rho_{\mathbf{z}}(\mathbf{z})$, the induced probability distribution of $\mathbf{x}$, $\rho_{\mathbf{x}}(\mathbf{x})$, is given by

$$\rho_{\mathbf{x}}(\mathbf{x}) = \rho_{\mathbf{z}}(\mathbf{z}) \left| \det\left(\frac{\partial\mathbf{z}}{\partial\mathbf{x}}\right)\right|, \tag{37}$$

where $\mathbf{x} = f(\mathbf{z})$. With careful design, a normalizing flow neural network is reversible and its Jacobian matrix is a product of triangular matrices with a trivial determinant. $\rho_{\mathbf{z}}(\mathbf{z})$ is usually a standard multivariate normal distribution. Normalizing flow is a very powerful generative model since it can generate independent samples of $\mathbf{x}$ just by sampling a normal distributed random variable $\mathbf{z}$ followed by transforming $\mathbf{z}$ to $\mathbf{x}$. However, there are many unsolved problems of this model. For example, applying this model with explicit solvent is still a challenging problem. Also, sampling with normalizing flow still requires reweighting, which suggests that improvement of accuracy is needed [177,178].

There are some general open questions for machine learning-based all-atom samplers. The first one is how to efficiently combine these samplers with enhanced sampling algorithms. The answer for autoencoder is relatively straightforward since the concept of CV is already in the model. Also, the configuration probability conditioned on given CV values is approximately unbiased in many enhanced sampling methods, which makes it easier to train an autoencoder model. The second question is about the accuracy of extrapolating the sampler to new conformations that are not in the training dataset, which requires further studies.

7 Conclusion

In this perspective we have reviewed recent developments on combining machine learning methods with CV-based enhanced sampling, especially in CV construction and FES training. Although introducing machine leaning algorithms to sampling has achieved great successes, many challenges still exist, which include generating accurate kinetic information from biased enhanced sampling simulations, exploring a high-dimensional FES, and developing machine learning-based all-atom samplers for CV-based enhanced sampling. Integrating machine learning techniques with enhanced sampling is often beyond applying generic machine learning algorithms directly. Developing physical theories and enhanced sampling methods to collaborate with machine learning techniques is also needed. Finally, building machine learning models tuned for molecules and materials is also important for unifying machine learning with CV-based enhanced sampling methods [179].

Acknowledgements The author acknowledges financial support from the Center for Computational Study of Excited State Phenomena in Energy Materials (C2SEPEM) at the Lawrence Berkeley National Laboratory, which is funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division under Contract No. DE-AC02-05CH11231 as part of the Computational Materials Sciences Program.

Author contributions

MC wrote and revised the whole paper.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: Data is available upon request from the authors.]

References

1. G. Ciccotti, M. Ferrario, C. Schuette, *Molecular Dynamics Simulation* (MDPI AG, 2018)
2. D.E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R.O. Dror, M.P. Eastwood, J.A. Bank, J.M. Jumper, J.K. Salmon, Y. Shan et al., Science **330**, 341 (2010)
3. Y. Sugita, Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999)
4. J.D. Faraldo-Gómez, B. Roux, J. Comput. Chem. **28**, 1634 (2007)
5. A. Laio, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **99**, 12562 (2002)
6. A. Barducci, G. Bussi, M. Parrinello, Phys. Rev. Lett. **100**, 020603 (2008)
7. L. Maragliano, E. Vanden-Eijnden, Chem. Phys. Lett. **426**, 168 (2006)
8. J.B. Abrams, M.E. Tuckerman, J. Phys. Chem. B **112**, 15742 (2008)
9. E. Darve, D. Rodríguez-Gómez, A. Pohorille, J. Chem. Phys. **128**, 144120 (2008)
10. G. Torrie, J. Valleau, J. Comput. Phys. **23**, 187 (1977)
11. E. Carter, G. Ciccotti, J.T. Hynes, R. Kapral, Chem. Phys. Lett. **156**, 472 (1989)
12. M. Sprik, G. Ciccotti, J. Chem. Phys. **109**, 7737 (1998)

13. R.W. Zwanzig, J. Chem. Phys. **22**, 1420 (1954)
14. J.G. Kirkwood, J. Chem. Phys. **3**, 300 (1935)
15. H. Oberhofer, C. Dellago, P.L. Geissler, J. Phys. Chem. B **109**, 6902 (2005)
16. M. Chen, M.A. Cuendet, M.E. Tuckerman, J. Chem. Phys. **137**, 024102 (2012)
17. A. Lesage, T. Lelièvre, G. Stoltz, J. Hénin, J. Phys. Chem. B **121**, 3676 (2017)
18. C. Dellago, P.G. Bolhuis, F.S. Csajka, D. Chandler, J. Chem. Phys. **108**, 1964 (1998)
19. C. Dellago, P.G. Bolhuis, D. Chandler, J. Chem. Phys. **110**, 6617 (1999)
20. T.S. van Erp, D. Moroni, P.G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003)
21. D. Moroni, P.G. Bolhuis, T.S. van Erp, J. Chem. Phys. **120**, 4055 (2004)
22. J. Rogal, P.G. Bolhuis, J. Chem. Phys. **129**, 224107 (2008)
23. R.J. Allen, P.B. Warren, P.R. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005)
24. G. Huber, S. Kim, Biophys. J. **70**, 97 (1996)
25. A.K. Faradjian, R. Elber, J. Chem. Phys. **120**, 10880 (2004)
26. A. Barducci, M. Bonomi, M. Parrinello, WIREs Comput. Mol. Sci. **1**, 826 (2011)
27. C.F. Abrams, E. Vanden-Eijnden, Proc. Natl. Acad. Sci. U.S.A. **107**, 4961 (2010)
28. T.Q. Yu, M.E. Tuckerman, Phys. Rev. Lett. **107**, 015701 (2011)
29. T.Q. Yu, P.Y. Chen, M. Chen, A. Samanta, E. Vanden-Eijnden, M. Tuckerman, J. Chem. Phys. **140**, 214109 (2014)
30. A. Samanta, M.E. Tuckerman, T.Q. Yu, W. E, Science **346**, 729 (2014)
31. D. Bonhenry, F. Dehez, M. Tarek, Phys. Chem. Chem. Phys. **20**, 9101 (2018)
32. C. Chipot, J. Héénin, J. Chem. Phys. **123**, 244906 (2005)
33. A. Bidon-Chanal, E.M. Krammer, D. Blot, E. Pebay-Peyroula, C. Chipot, S. Ravaud, F. Dehez, J. Phys. Chem. Lett. **4**, 3787 (2013)
34. C.H. Tse, J. Comer, S.K. Sang Chu, Y. Wang, C. Chipot, J. Chem. Theory Comput. **15**, 2913 (2019)
35. P. Tiwary, M. Parrinello, Phys. Rev. Lett. **111**, 230602 (2013)
36. G.A. Kaminski, R.A. Friesner, J. Tirado-Rives, W.L. Jorgensen, J. Phys. Chem. B **105**, 6474 (2001)
37. M. Iannuzzi, A. Laio, M. Parrinello, Phys. Rev. Lett. **90**, 238302 (2003)
38. S. Awasthi, V. Kapil, N.N. Nair, J. Comput. Chem. **37**, 1413 (2016)
39. T. Huber, A.E. Torda, W.F. van Gunsteren, J Computer-Aided Mol Des **8**, 695 (1994)
40. H. Grubmüller, Phys. Rev. E **52**, 2893 (1995)
41. L. Maragliano, E. Vanden-Eijnden, J. Chem. Phys. **128**, 184110 (2008)
42. J. Kästner, W. Thiel, J. Chem. Phys. **123**, 144104 (2005)
43. O. Valsson, M. Parrinello, Phys. Rev. Lett. **113**, 090601 (2014)
44. O. Valsson, P. Tiwary, M. Parrinello, Annu. Rev. Phys. Chem. **67**, 159 (2016)
45. V. Limongelli, M. Bonomi, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **110**, 6358 (2013)
46. L. Rosso, P. Mináry, Z. Zhu, M.E. Tuckerman, J. Chem. Phys. **116**, 4389 (2002)
47. Y.I. Yang, Q. Shao, J. Zhang, L. Yang, Y.Q. Gao, J. Chem. Phys. **151**, 070902 (2019)
48. P. Tiwary, A. van de Walle, *A Review of Enhanced Sampling Approaches for Accelerated Molecular Dynamics* (Springer International Publishing, Cham, 2016), pp. 195–221, ISBN 978-3-319-33480-6
49. C. Dellago, P.G. Bolhuis, D. Chandler, J. Chem. Phys. **108**, 9236 (1998)
50. R. Elber, Q. Rev, Biophys. **50**, e8 (2017)
51. P.G. Bolhuis, C. Dellago, D. Chandler, Proc. Natl. Acad. Sci. U.S.A. **97**, 5877 (2000)
52. K. Kuczera, G.S. Jas, R. Elber, J. Phys. Chem. A **113**, 7461 (2009)
53. T.Q. Yu, M. Lapelosa, E. Vanden-Eijnden, C.F. Abrams, J. Am. Chem. Soc. **137**, 3041 (2015)
54. C. Bishop, *Pattern Recognition and Machine Learning* (Springer-Verlag, New York, 2006)
55. D.M. Allen, Technometrics **16**, 125 (1974)
56. M. Stone, J.R. Stat, Soc. Series B Stat. Methodol. **36**, 111 (1974)
57. W.S. McCulloch, W. Pitts, Bull. Math. Biophys. **5**, 115 (1943)
58. F. Rosenblatt, Psychol. Rev. pp. 386–408 (1958)
59. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Nature **323**, 533 (1986)
60. I.J. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, USA, 2016)
61. X. Glorot, A. Bordes, Y. Bengio, *Deep Sparse Rectifier Neural Networks*, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by G. Gordon, D. Dunson, M. Dudík (PMLR, Fort Lauderdale, FL, USA, 2011), Vol. 15 of Proceedings of Machine Learning Research, pp. 315–323
62. K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778
63. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Neural Comput. **1**, 541 (1989)
64. F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, IEEE Trans. Neural Netw. **20**, 61 (2009)
65. H.J. Kelley, ARS Journal **30**, 947 (1960)
66. S. Dreyfus, J. Math. Anal. Appl. **5**, 30 (1962)
67. G. Hummer, A. Szabo, Proc. Natl. Acad. Sci. U.S.A. **98**, 3658 (2001)
68. G. Bussi, F.L. Gervasio, A. Laio, M. Parrinello, J. Am. Chem. Soc. **128**, 13435 (2006)
69. M. Chen, T.Q. Yu, M.E. Tuckerman, Proc. Natl. Acad. Sci. U.S.A. **112**, 3235 (2015)
70. G.A. Tribello, M. Ceriotti, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **107**, 17509 (2010)
71. J. Zhang, M. Chen, Phys. Rev. Lett. **121**, 010601 (2018)
72. M. Chen, Ph.D. thesis, New York University (2016)
73. M. Ceriotti, G.A. Tribello, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **108**, 13023 (2011)

74. G.A. Tribello, M. Ceriotti, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **109**, 5196 (2012)
75. M. Bonomi, A. Barducci, M. Parrinello, J. Comput. Chem. **30**, 1615 (2009)
76. P. Tiwary, M. Parrinello, J. Phys. Chem. B **119**, 736 (2015)
77. F. Giberti, B. Cheng, G.A. Tribello, M. Ceriotti, J. Chem. Theory Comput. **16**, 100 (2020)
78. Y. Hu, W. Hong, Y. Shi, H. Liu, J. Chem. Theory Comput. **8**, 3777 (2012)
79. S. Kumar, J.M. Rosenberg, D. Bouzida, R.H. Swendsen, P.A. Kollman, J. Comput. Chem. **13**, 1011 (1992)
80. M.R. Shirts, J.D. Chodera, J. Chem. Phys. **129**, 124105 (2008)
81. H. Wu, F. Noé, Multiscale Model. Simul. **12**, 25 (2014)
82. A.S.J.S. Mey, H. Wu, F. Noé, Phys. Rev. X **4**, 041018 (2014)
83. H. Wu, A.S.J.S. Mey, E. Rosta, F. Noé, J. Chem. Phys. **141**, 214106 (2014)
84. P. Tiwary, B.J. Berne, Proc. Natl. Acad. Sci. U.S.A. **113**, 2839 (2016)
85. D. Mendels, G. Piccini, M. Parrinello, J. Phys. Chem. Lett. **9**, 2776 (2018)
86. J.M.L. Ribeiro, P. Bravo, Y. Wang, P. Tiwary, J. Chem. Phys. **149**, 072301 (2018)
87. Y. Wang, J.M.L. Ribeiro, P. Tiwary, Nat. Commun. **10**, 3573 (2019)
88. F. Nüske, B.G. Keller, G. Pérez-Hernández, A.S.J.S. Mey, F. Noé, J. Chem. Theory Comput. **10**, 1739 (2014)
89. M.M. Sultan, V.S. Pande, J. Chem. Theory Comput. **13**, 2440 (2017)
90. F. Hooft, A. Pérez de Alba Ortíz, B. Ensing, J. Chem. Theory Comput. **0**, null (0)
91. W. Chen, A.R. Tan, A.L. Ferguson, J. Chem. Phys. **149**, 072312 (2018)
92. P. Das, M. Moll, H. Stamati, L.E. Kavraki, C. Clementi, Proc. Natl. Acad. Sci. U.S.A. **103**, 9885 (2006)
93. A.L. Ferguson, A.Z. Panagiotopoulos, P.G. Debenedetti, I.G. Kevrekidis, J. Chem. Phys. **134**, 135103 (2011)
94. M.A. Rohrdanz, W. Zheng, M. Maggioni, C. Clementi, J. Chem. Phys. **134**, 124116 (2011)
95. M. Schöberl, N. Zabaras, P.S. Koutsourelakis, J. Chem. Phys. **150**, 024109 (2019)
96. J. Rogal, E. Schneider, M.E. Tuckerman, Phys. Rev. Lett. **123**, 245701 (2019)
97. K.P. F.R.S., London, Edinburgh Dublin Philos. Mag. J. Sci. **2**, 559 (1901)
98. J.B. Tenenbaum, V.d. Silva, J.C. Langford, Science **290**, 2319 (2000)
99. S.T. Roweis, L.K. Saul, Science **290**, 2323 (2000)
100. R.R. Coifman, S. Lafon, Appl. Comput. Harmon. Anal. **21**, 5 (2006)
101. L. van der Maaten, G. Hinton, J. Mach. Learn. Res. **9**, 2579 (2008)
102. J.A. Lee, M. Verleysen, *Nonlinear Dimensionality Reduction*, 1st edn. (Springer Publishing Company, Incorporated, 2007)
103. V. Spiwok, P. Lipovová, B. Králová, J. Phys. Chem. B **111**, 3073 (2007)
104. H. Zhou, F. Wang, P. Tao, J. Chem. Theory Comput. **14**, 5499 (2018)
105. P.L. Geissler, C. Dellago, D. Chandler, J. Phys. Chem. B **103**, 3706 (1999)
106. P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Annu. Rev. Phys. Chem. **53**, 291 (2002)
107. O. Kukharenko, K. Sawade, J. Steuer, C. Peter, J. Chem. Theory Comput. **12**, 4726 (2016)
108. A. Ardevol, G.A. Tribello, M. Ceriotti, M. Parrinello, J. Chem. Theory Comput. **11**, 1086 (2015)
109. G.E. Hinton, S. Roweis, *Stochastic Neighbor Embedding*, in *Advances in Neural Information Processing Systems*, edited by S. Becker, S. Thrun, K. Obermayer (MIT Press, 2003), Vol. 15
110. D.P. Kingma, M. Welling, arXiv:1312.6114 (2014)
111. T. Lemke, C. Peter, J. Chem. Theory Comput. **15**, 1209 (2019)
112. M.M. Sultan, H.K. Wayment-Steele, V.S. Pande, J. Chem. Theory Comput. **14**, 1887 (2018)
113. L. Molgedey, H.G. Schuster, Phys. Rev. Lett. **72**, 3634 (1994)
114. Y. Naritomi, S. Fuchigami, J. Chem. Phys. **134**, 065101 (2011)
115. C.R. Schwantes, V.S. Pande, J. Chem. Theory Comput. **9**, 2000 (2013)
116. G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, F. Noé, J. Chem. Phys. **139**, 015102 (2013)
117. E. Hruska, V. Balasubramanian, H. Lee, S. Jha, C. Clementi, J. Chem. Theory Comput. **16**, 7915 (2020)
118. J. McCarty, M. Parrinello, J. Chem. Phys. **147**, 204109 (2017)
119. S. Still, Entropy **16**, 968 (2014)
120. J. Preto, C. Clementi, Phys. Chem. Chem. Phys. **16**, 19181 (2014)
121. R.R. Coifman, I.G. Kevrekidis, S. Lafon, M. Maggioni, B. Nadler, Multiscale Model. Simul. **7**, 842 (2008)
122. S. Pressé, K. Ghosh, J. Lee, K.A. Dill, Rev. Mod. Phys. **85**, 1115 (2013)
123. P.D. Dixit, A. Jain, G. Stock, K.A. Dill, J. Chem. Theory Comput. **11**, 5464 (2015)
124. H. Mori, Prog. Theor. Phys. **33**, 423 (1965)
125. R. Zwanzig, Prog. Theor. Phys. **9**, 215 (1973)
126. A. Laio, A. Rodriguez-Fortea, F.L. Gervasio, M. Ceccarelli, M. Parrinello, J. Phys. Chem. B **109**, 6714 (2005)
127. P. Raiteri, A. Laio, F.L. Gervasio, C. Micheletti, M. Parrinello, J. Phys. Chem. B **110**, 3533 (2006)
128. M.A. Cuendet, M.E. Tuckerman, J. Chem. Theory Comput. **10**, 2975 (2014)
129. N.S. Altman, Am. Stat. **46**, 175 (1992)
130. Y. Lin, Y. Jeon, J. Am. Stat. Assoc. **101**, 578 (2006)
131. A. Rodriguez, M. d'Errico, E. Facco, A. Laio, J. Chem. Theory Comput. **14**, 1206 (2018)
132. G.J. McLachlan, K.E. Basford, *Mixture Models* (Inference and applications to clustering (Marcel Dekker, New York, 1988)
133. J. Debnath, M. Parrinello, J. Phys. Chem. Lett. **11**, 5076 (2020)
134. I. Kobyzev, S. Prince, M. Brubaker, IEEE Trans. Pattern Anal. Mach. Intell. pp. 1 (2020)
135. M.J. Ruiz-Montero, D. Frenkel, J.J. Brey, Mol. Phys. **90**, 925 (1997)
136. W.K. den Otter, J. Chem. Phys. **112**, 7283 (2000)

137. G. Ciccotti, R. Kapral, E. Vanden-Eijnden, ChemPhysChem **6**, 1809 (2005)
138. T. Stecher, N. Bernstein, G. Csányi, J. Chem. Theory Comput. **10**, 4079 (2014)
139. K. Hornik, M. Stinchcombe, H. White, Neural Netw. **2**, 359–366 (1989)
140. E. Schneider, L. Dai, R.Q. Topper, C. Drechsel-Grau, M.E. Tuckerman, Phys. Rev. Lett. **119**, 150601 (2017)
141. L. Zhang, H. Wang, W. E, J. Chem. Phys. **148**, 124113 (2018)
142. J.R. Cendagorta, J. Tolpin, E. Schneider, R.Q. Topper, M.E. Tuckerman, J. Phys. Chem. B **124**, 3647 (2020)
143. E. Sevgen, A.Z. Guo, H. Sidky, J.K. Whitmer, J.J. de Pablo, J. Chem. Theory Comput. **16**, 1448 (2020)
144. L. Bonati, Y.Y. Zhang, M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **116**, 17641 (2019)
145. A.Z. Guo, E. Sevgen, H. Sidky, J.K. Whitmer, J.A. Hubbell, J.J. de Pablo, J. Chem. Phys. **148**, 134108 (2018)
146. B.E. Husic, V.S. Pande, J. Am. Chem. Soc. **140**, 2386 (2018)
147. J.D. Chodera, F. Noé, Curr. Opin. Struct. Biol. **25**, 135 (2014)
148. Y. Wang, O. Valsson, P. Tiwary, M. Parrinello, K. Lindorff-Larsen, J. Chem. Phys. **149**, 072309 (2018)
149. L. Donati, B.G. Keller, J. Chem. Phys. **149**, 072335 (2018)
150. S. Kieninger, B.G. Keller, J. Chem. Phys. **154**, 094102 (2021)
151. D. Mandelli, B. Hirshberg, M. Parrinello, Phys. Rev. Lett. **125**, 026001 (2020)
152. J.H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J.D. Chodera, C. Schütte, F. Noé, J. Chem. Phys. **134**, 174105 (2011)
153. H. Wu, F. Paul, C. Wehmeyer, F. Noé, Proc. Natl. Acad. Sci. U.S.A. **113**, E3221 (2016)
154. S. Piana, A. Laio, Phys. Rev. Lett. **101**, 208101 (2008)
155. R. Hegger, A. Altis, P.H. Nguyen, G. Stock, Phys. Rev. Lett. **98**, 028102 (2007)
156. E. Facco, M. d'Errico, A. Rodriguez, A. Laio, Nat. Commun. **7**, 12140 (2017)
157. G. Sormani, A. Rodriguez, A. Laio, J. Chem. Theory Comput. **16**, 80 (2020)
158. M. Carli, G. Sormani, A. Rodriguez, A. Laio, J. Phys. Chem. Lett. **12**, 65 (2021)
159. G.A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, G. Bussi, Comput. Phys. Commun. **185**, 604 (2014)
160. J. Comer, J.C. Gumbart, J. Hénin, T. Lelièvre, A. Pohorille, C. Chipot, J. Phys. Chem. B **119**, 1129 (2015)
161. M. Invernizzi, M. Parrinello, J. Phys. Chem. Lett. **11**, 2731 (2020)
162. S. Izvekov, G.A. Voth, J. Phys. Chem. B **109**, 2469 (2005)
163. Y. Wang, W.G. Noid, P. Liu, G.A. Voth, Phys. Chem. Chem. Phys. **11**, 2002 (2009)
164. W.G. Noid, J.W. Chu, G.S. Ayton, V. Krishna, S. Izvekov, G.A. Voth, A. Das, H.C. Andersen, J. Chem. Phys. **128**, 244114 (2008)
165. M.S. Shell, J. Chem. Phys. **129**, 144108 (2008)
166. J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N.E. Charron, G. de Fabritiis, F. Noé, C. Clementi, A.C.S. Cent, Sci. **5**, 755 (2019)
167. J. Wang, S. Chmiela, K.R. Müller, F. Noé, C. Clementi, J. Chem. Phys. **152**, 194106 (2020)
168. B.E. Husic, N.E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis et al., J. Chem. Phys. **153**, 194101 (2020)
169. J. Zhang, Y.K. Lei, Y.I. Yang, Y.Q. Gao, J. Chem. Phys. **153**, 174115 (2020)
170. J. Wang, N. Charron, B. Husic, S. Olsson, F. Noé, C. Clementi, J. Chem. Phys. **154**, 164113 (2021)
171. P. Minary, M.E. Tuckerman, G.J. Martyna, Phys. Rev. Lett. **93**, 150201 (2004)
172. B. Leimkuhler, D.T. Margul, M.E. Tuckerman, Mol. Phys. **111**, 3579 (2013)
173. F. Noé, S. Olsson, J. Köhler, H. Wu, Science **365** (2019)
174. M. Xu, S. Luo, Y. Bengio, J. Peng, J. Tang, arXiv:2102.10240 (2021)
175. M. Stieffenhofer, M. Wand, T. Bereau, Mach. Learn.: Sci. Technol. **1**, 045014 (2020)
176. S. Hunkler, T. Lemke, C. Peter, O. Kukharenko, J. Chem. Phys. **151**, 154102 (2019)
177. J. Köhler, L. Klein, F. Noe, *Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities*, in *Proceedings of the 37th International Conference on Machine Learning*, edited by H.D. III, A. Singh (PMLR, 2020), Vol. 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370
178. H. Wu, J. Köhler, F. Noe, *Stochastic Normalizing Flows*, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 5933–5944
179. J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, arXiv:1812.08434 (2018)