

# A Perspective on Deep Learning for Molecular Modeling and Simulations

Jun Zhang<sup>1,2</sup>, Yao-Kun Lei<sup>3</sup>, Zhen Zhang<sup>4</sup>, Junhan Chang<sup>3</sup>, Maodong Li<sup>1</sup>, Xu Han<sup>3</sup>, Lijiang Yang<sup>3</sup>, Yi Isaac Yang<sup>1,†</sup> and Yi Qin Gao<sup>1,3,5,6,‡</sup>

<sup>1</sup> *Institute of Systems Biology, Shenzhen Bay Laboratory, 518055 Shenzhen, China*

<sup>2</sup> *Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany*

<sup>3</sup> *Beijing National Laboratory for Molecular Sciences, College of Chemistry and Molecular Engineering, Peking University, 100871 Beijing, China.*

<sup>4</sup> *Department of Physics, Tangshan Normal University, 063000 Tangshan, China.*

<sup>5</sup> *Beijing Advanced Innovation Center for Genomics, Peking University, 100871 Beijing, China.*

<sup>6</sup> *Biomedical Pioneering Innovation Center, Peking University, 100871 Beijing, China.*

Corresponding authors: † (Y.I.Y) yangyi@szbl.ac.cn or ‡ (Y.Q.G) gaoyq@pku.edu.cn

## Abstract

Deep learning is transforming many areas in science, and it has great potential in modeling molecular systems. However, unlike the mature deployment of deep learning in computer vision and natural language processing, its development in molecular modeling and simulations is still at an early stage, largely because the inductive biases of molecules are completely different from those of images or texts. Footed on these differences, we first reviewed the limitations of traditional deep learning models from the perspective of molecular physics, and wrapped up some relevant technical advancement at the interface between molecular modeling and deep learning. We do not focus merely on the ever more complex neural network models, instead, we emphasize the theories and ideas behind modern deep learning. We hope that transacting these ideas into molecular modeling will create new opportunities. For this purpose, we summarized several representative applications, ranging from supervised to unsupervised and reinforcement learning, and discussed their connections with the emerging trends in deep learning. Finally, we outlook promising directions which may help address the existing issues in the current framework of deep molecular modeling.

**Keywords:** Coarse graining, Enhanced sampling, Molecular dynamics, Machine learning, Artificial neural networks

## Abbreviations

AEV: atomic environment vector; AIMD: ab initial molecular dynamics; ANN: artificial neural network; BPNN: Behler-Parrinello neural network; CG: coarse-grained/graining; CNN: convolutional neural network; CV: collective variable; DPMD: deep potential molecular dynamics; EBM: energy-based models; ERM: empirical risk minimization; FEL: free energy landscape; FES: free energy surface; FG: fine-grained; GAN: generative adversarial network; GCNN: graph convolutional neural network; GNN: graph neural network; GP: gradient penalty; IDM: information distilling of metastability; MC: Monte Carlo; MaxEnt: maximum entropy; MI: mutual information; MD: molecular dynamics; MLP: multi-layer perceptron; MPNN: message-passing neural network; MRF: Markov random field; PES: potential energy surface; PMF: potential of mean force; RNN: recurrent neural network; TALOS: targeted adversarial learning optimized sampling; (RE-)VIFE: (reinforced) variational inference of free energy.

## Introduction

During the past decade, burgeoning development of artificial intelligence, particularly, connectionist models or artificial neural networks (ANNs), starts to transform many aspects in science. For instance, there is a burst of attempts to exploit ANNs to address challenges in the field of molecular modeling and simulations, like the potential energy fitting,<sup>1-2</sup> coarse graining,<sup>3-4</sup> enhanced sampling<sup>5-6</sup> and kinetic modeling,<sup>7-8</sup> *etc.* Technically speaking, an ANN is a parametric model consisting of a network of simple neuron-like processing units that collectively perform complex computations.<sup>9</sup> ANNs are often organized into layers, including an input layer that presents the data, hidden layers that transform the data into intermediate representations, and an output layer that produces a response (e.g., a label or an action). For example, the multi-layer perceptron (MLP) is one classic realization of ANNs. In recent years, connectionist models are undergoing a resurgence in the form of *deep learning* by stacking (usually much) more than one hidden layer in ANNs and giving rise to models in relatively “deep” architectures.<sup>10-11</sup> Most of the state-of-the-art deep ANNs are trained upon large amount of data using the backpropagation algorithm<sup>12</sup> to gradually adjust their connection strengths, and have achieved impressive and even superhuman performance on many challenging tasks like computer vision and natural language processing.<sup>10-11</sup>

A remarkable feature of ANNs is that they are universal approximators, i.e., they can compute any functions. This is known as the *universal approximation theorem*<sup>13</sup> which bedrocks the connectionist models. Since a wide range of processes or tasks in daily life and science can be considered as a form of function computation, this characteristic of ANNs is the reason why they are so versatile. Mathematically speaking, ANNs are essentially a way to represent functions in a certain basis, similar to polynomials and harmonic functions. What differentiates ANNs from polynomials in machine learning is their ability to hierarchically represent and approximate basis functions on their own. Therefore, (deep) ANNs exhibit a better trade-off between flexibility (or expressivity) and tractability, and are more scalable than polynomials when dealing with high-dimensional data. From this respect, deep learning can be particularly useful in molecular modeling, for example, representing the potential energy surface (PES) of Hamiltonian systems; or improving solutions to problems bounded by certain variational principles. But before successfully exploiting deep learning to solve the problems in molecular modeling and simulations, we are facing with two major challenges.

**Challenge 1.** The model architecture suitable for molecular systems is very different from traditional deep models. Although the *universal approximation theorem* confirms the existence of a certain ANN to be a universal approximator, it does not show what architecture that ANN should take in order to be truly universal. In theory, even a MLP with one single hidden layer could be universally expressive; while practitioners of ANNs found that sophisticated architectures with more hidden layers are helpful to both the training efficiency and the test performance. In terms of mathematics, designing the network architecture is equivalent to specifying the hypothesis space for the models, and the hypothesis space should naturally be different for different types of input data. In other words, instead of regarding the ANN as a sheer “black box”, a rational design of the architecture according to the input data is crucial to the performance. In machine learning, capturing the inductive biases of the data is the rule of thumb for designing the network architecture. For instance, the convolutional neural network (CNN)<sup>14</sup> achieves great success in computer vision because it captures the fact that the identity of a visual object should be (at least) invariant w.r.t. its location in an image; while the recurrent neural network (RNN)<sup>15</sup> succeeds in natural language processing provided that it is able to learn the context-dependence of language semantics. The invention of attention mechanism<sup>16-17</sup> and residual (or skipped) connections<sup>18</sup> are also good examples of architecture design inspired by certain inductive biases. However, molecular systems

involve some inductive biases very different from images or texts as will be discussed in Section I. Therefore, it is not straightforward to directly apply traditional ANN model architectures (like MLP, CNN or RNN) to handle molecular systems, and the lack of proper model architecture constitutes one major challenge over deploying deep learning for molecular modeling and simulations.

**Challenge 2.** The learning objectives involved in molecular modeling are very different from traditional deep learning tasks. Taking supervised learning as example, most of the applications of deep learning are focused on classification tasks, leading to a fruitful research field called deep discriminative learning.<sup>10</sup> In contrast, how to solve regression problems which are much more commonly encountered in molecular modeling (where the smoothness of the target functions is of particularly importance), still remains challenging and less addressed. Moreover, many long-standing problems in molecular simulations (e.g., importance sampling and dimensionality reduction) cannot be simply solved in a supervised manner. Instead, deep unsupervised learning (particularly, generative learning)<sup>19</sup> may bring about new solutions to such problems. Besides, in molecular modeling and simulations, one often has to deal with dynamic datasets because the samples are generated or updated on the fly in the meantime of model optimization. This is quite different from the common settings for supervised learning where the dataset is static and prepared *a priori*. Instead, one should resort to an adaptive learning approach to tackle the floating data as in reinforcement learning or active learning. In summary, applications of deep learning in molecular modeling and simulations are beyond the scope of supervised learning, and we are required to carefully formulate the learning objective and adopt the correct learning strategy.

In this review, we will describe some recent proceedings focused on the two challenges introduced above. The rest of the paper is organized as follows: In Section I, we will first summarize some state-of-the-art network architectures that can be used to model molecular systems. From Section II to Section IV, various applications of deep learning in molecular simulations under different learning strategies will be presented. We will also inspect the limitations of the existing methods and learning frameworks, as well as outlook several promising and important directions worthy of further exploration and investigation.

## Section I. Architectures of deep models for molecular modeling and simulations

### 1. Inductive biases of molecular systems

In machine learning, inductive biases are some assumptions about the data, reflecting the internal structure, or invariance, or some other intrinsic properties of the data based on empirical knowledge. Incorporation of inductive biases allows a learning algorithm to restrict the search scope for possible solutions or to prioritize one solution over another, independent of the observed data. From the perspective of deep learning, capturing the inductive biases within the data is the core of successful deep models. Examples include the success of CNNs over images and RNNs over texts etc. Therefore, in order to develop reasonable models for molecular systems, we should first be aware of the inductive biases underlying the molecular systems, as summarized below.

(1) **Reference invariance:** Properties (e.g., the energy) of molecular systems should be invariant to the trans-rotational transform of the system’s coordinates. In other words, molecular properties are determined by the relative but not the absolute positions of the particles in the absence of external field(s).

(2) **Degeneracy and symmetry:** Molecular properties should be invariant w.r.t. the permutation or exchange of identical particles (e.g., atoms, molecules or residues). Besides, many molecular properties are governed by some rules about the geometric symmetry.

(3) **Smoothness:** Changes in molecular properties should be continuous and smooth w.r.t. the changes in positions of the constituting particles.

Therefore, a deep neural network can be adopted to describe molecular systems only if it is able to preserve the above inductive biases. Apparently, traditional deep models developed in machine learning community do not meet these requirements, so novel models of special-purpose architectures are largely needed. Fortunately, some deep models have already been developed for this purpose, and we will term them as the “*deep molecular models*” hereafter. Existing deep molecular models can be roughly distinguished into two types according to their architectures: the descriptor-based models or the graph neural network (GNN)<sup>20</sup> based models. We surveyed some representative deep molecular models in Table 1. On top of the model architecture, choosing proper activation functions and regularization techniques is also crucial for the performance of deep molecular models. We will elaborate on these topics in the following.

### 2. Descriptor-based models

As its name suggests, the descriptor-based model builds up on “descriptors”: vectors that can characterize the molecular systems. Once the descriptors are chosen, they can be directly used as input to a simple feed-forward ANN which approximates a target molecular property. The resulting ANNs are thus eligible for modeling molecular systems, provided that the descriptors are trans-rotational invariant and preserve the symmetry of the system. Let  $\mathbf{x}$  stand for a molecular system consisting of  $N$  particles  $\{\mathbf{x}_i\}_{i=1,\dots,N}$  (the same symbols will be consistently re-used in this paper), the descriptor-based models first transform the  $i$ -th particle coordinate  $\mathbf{x}_i$  into a set of descriptors  $\mathbf{q}(\mathbf{x}_i)$ , and exploit simple ANNs like MLPs, denoted by  $f_\theta$ , to predict the molecular property  $A(\mathbf{x})$  based on the descriptors as summarized in Eqs. (1-2). Figure 1A shows a schematic computational graph of typical descriptor-based models.

$$\mathbf{x}_i \mapsto \mathbf{q}(\mathbf{x}_i) \tag{1}$$

$$A(\mathbf{x}) = \sum_{i=1}^N f_{\theta}(\mathbf{q}(\mathbf{x}_i)) \quad (2)$$

In machine learning, choosing “descriptors” is also known as feature engineering. In order to predict the desired molecular property, in principle, one should find a “complete set” of descriptors determining that property, because neglecting even one of the essential components would cause prediction failure. Clearly, finding a “complete set” is generally too demanding. Nevertheless, in practice, one may still manually pre-define a large set of descriptors which may contain redundant information and yield a very high-dimensional input vector, then handle this high-dimensional descriptor via ANNs. As expected, the performance of such models will heavily rely on the choice of the descriptors.

In terms of machine learning, descriptors are one kind of “*structured input*”. Like other machine learning methods dealing with structured input, the performance is usually not so much sensitive to the model architecture as to the feature engineering or pre-processing. Due to this fact, research and development of descriptor-based models mainly focused on feature engineering, that is, selecting descriptors  $\mathbf{q}(\mathbf{x})$ . The pioneering work along this line can be dated back to the Behler-Parrinello neural network (BPNN).<sup>21</sup> In BPNN, the environment of each atom is transformed by specific symmetric functions and encoded into descriptors, called atomic environment vectors (AEVs). Based on these AEV descriptors, a set of MLPs output the atomic energies, the summation of which gives the overall energy. BPNN shares parameters, that is, uses a same MLP for the same type of atom, but MLPs are separately parametrized for different types of atoms. Noteworthy, the descriptors introduced by BPNN and its following variants ANI and TensorMol,<sup>22-23</sup> not only include two-body terms, but also account for three-body (or angular) features. Following this line, one can even develop four-body descriptors if necessary (e.g., chirality descriptors). On top of such atom-centered symmetry descriptors, many improved or alternative ways of formulating the descriptors were proposed, to name a few, bi-spectrum components,<sup>24</sup> Coulomb matrix,<sup>25</sup> smooth overlap of atomic positions,<sup>26</sup> etc. In a recently developed descriptor-based model, deep potential molecular dynamics (DPMD),<sup>2</sup> treats each atom in a local coordinate frame, of which the rotation and translation degrees of freedom are removed. The local coordinate of each atom can be regarded as one type of AEV thus can be adopted as descriptors as in BPNN.

One nicety of these descriptor-based models, especially when compared with GNN-based models which will be discussed later, is that the computational complexity is generally linearly scaled w.r.t. the number of particles ( $\mathcal{O}(N)$  where  $N$  is the number of particles) once the descriptors are well formulated. Nonetheless, from the introduction above, it is not hard to follow that descriptor-based models cannot scale well with the “complexity” of the molecular system. Specifically, when the heterogeneity of the system (e.g., the type of constituting particles) increases, the number of necessary descriptors will grow rapidly. As a result, applications of descriptor-based models were mostly limited to relatively simple systems.<sup>27</sup> Worse still, since the suitable descriptors  $\mathbf{q}(\mathbf{x})$  will inevitably vary for different systems, the descriptor-based models have limited transferability between different molecular systems, thus drawbacks their applications for more general purposes like force field development. Despite of these shortcomings, development of the descriptor-based models has furthered and enriched our understanding of how to represent molecular systems.<sup>26, 28</sup> The various descriptors proposed in this field have found wide applications beyond deep molecular models. For example, some descriptors can be used as good collective variables (CVs) or order parameters for the analysis or enhanced sampling of material systems.<sup>29-30</sup> Moreover, finding good descriptors relevant to a particular property on its own can be defined as a machine learning problem, where the manual feature engineering can be replaced by automatic representation learning, which will be an important direction for further studies.

### 3. GNN-based models

In contrast to descriptor-based models, some recently proposed deep molecular models directly take the coordinates and types of the constituting particles as the input. In other words, this kind of models is able to deal with the “*unstructured input*” according to Eq. (3),

$$A(\mathbf{x}) = f_{\theta}(\{z_i, \mathbf{x}_i\}) \quad (3)$$

In Eq. (3),  $z_i$  and  $\mathbf{x}_i$  correspond to the type and coordinates of the  $i$ -th particle (e.g., if the constituting particle is atom,  $z_i$  can be the atomic number or nuclear charge) and the neural network  $f_{\theta}$  takes in the entire set of the particles as the input. The symbol  $\{\cdot\}$  indicates that  $f_{\theta}$  is invariant to the re-ordering of the particles. Such kind of models treats the molecular system as a dense graph, where the constituting particles correspond to the nodes (or vertices) in a graph, and two-body interactions can be straightforwardly represented by weighted edges between nodes. Representing molecules as graphs has the advantage that it naturally captures the inductive biases for molecular systems. Similar graph-based models have been developed in machine learning community to tackle with interacting particles or agents.<sup>31</sup> Following this line, deep molecular models can directly borrow ideas and techniques from GNNs.<sup>20</sup> Among the various types of GNNs, graph convolutional neural networks (GCNNs)<sup>32-33</sup> and message-passing neural networks (MPNNs)<sup>34</sup> are of particular usefulness in modeling molecular systems. Typically, in GNN-based molecular models (Fig. 1B), each particle  $i$  is represented by a node vector  $\mathbf{n}_i$ ,

$$\mathbf{n}_i = f_{\theta_{\text{embed}}}(z_i) \quad (4)$$

During learning, each node  $i$  will first gather messages  $\mathbf{m}_i$  from its neighbors (denoted by  $\mathcal{N}(i)$ ), for example, through Eq. (5),

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} f_{\theta_{\text{message}}}(\{\mathbf{n}_j, |\mathbf{x}_j - \mathbf{x}_i|\}; z_i) \quad (5)$$

Then each node vector is updated based on these messages, for instance, through Eq. (6),

$$\mathbf{n}_i = \mathbf{n}_i + f_{\theta_{\text{update}}}(\mathbf{m}_i; z_i) \quad (6)$$

Equations (5-6) constitute one iteration of message-passing and node update. After repeating a fixed number of such iterations, one gets the final embedding for each node  $\mathbf{n}_i$  and predicts the molecular property via Eq. (7),

$$A(\mathbf{x}) = \sum_i f_{\theta_{\text{read}}}(\mathbf{n}_i) \quad (7)$$

In summary, the overall model Eq. (3) is now decomposed into four modules:  $\theta_{\text{embed}}$  for node embedding,  $\theta_{\text{message}}$  for message passing,  $\theta_{\text{update}}$  for node update, and  $\theta_{\text{read}}$  for final readout. Guided by the general principle formulated in Eqs. (4-7), different GNN-based molecular models differ with each other in their choices of the model architecture for the four specific modules. For example, SchNet<sup>35</sup> along with its earlier ancestor deep tensor neural networks (DTNN)<sup>36</sup> is a continuous analogue to the discrete GCNNs, and performs continuous convolution over the 3D Euclidean space. To achieve this goal, SchNet first employs ANNs to generate convolutional filters based on inter-atomic distances, then adopts these filters to perform convolutional operations over the particles in the space. On the other hand, PhysNet<sup>37</sup> represents another class of GNN-based models which replace convolutional operations with attention mechanism.<sup>16-17</sup> Since attention-based models have achieved the

state of art in many deep learning tasks including computer vision and natural language processing,<sup>17, 38</sup> it is also likely that graph attention operations<sup>39</sup> would further improve the performance of deep molecular models.

Note that GNN-based models consist exclusively of particle-wise operations, thus the overall model can be transferable regardless of the number and type of the particles in the system. Besides, since the GNN-based model takes in unstructured input, it does not rely much on the expertise knowledge and pre-processing of the system. It can be readily applied to complex molecular systems. Due to these merits, GNN-based models are likely suitable for general-purpose applications like force-field development. However, there are some caveats in deploying the existing GNN-based models: (1) Many GNN-based models suffer issues like non-smooth cutoff when dealing with condensed-phase systems (particularly, with periodic boundary conditions) and long-range interactions; (2) Existing GNN-based models exclusively build upon pair-wise distances, thus may lack the expressivity for intrinsic higher-order interactions or properties (like chirality) compared to descriptor-based models; (3) GNN-based models are generally intensively over-parameterized, and the computational complexity is generally  $\mathcal{O}(N^2)$ , consequently, both the training and runtime of these models could entail prohibitive computational costs.

#### 4. Model smoothness

No matter a descriptor-based or GNN-based model is exploited, the gradients produced by the model are not necessarily smooth enough, which may cause instabilities for numerical integration in simulations if the gradients are collected as forces in Newtonian mechanics. To attack the issue of irregular gradients while maximally harnessing the expressivity, one usually needs to impose certain regularization in order to achieve a “smooth” model. In the research of deep ANNs, Lipschitz continuity is a commonly adopted measure quantifying the smoothness of the model. A  $K$ -Lipschitz function is a continuous function which has gradients whose norm is no larger than a positive constant  $K$  everywhere. Given this definition, one is motivated to add a gradient-penalty term in order to restrict the model to be  $K$ -Lipschitz.<sup>40</sup> More recent studies revealed that the non-smoothness of ANNs can result from the fact that the weight matrices of ANNs become ill-conditioned during training.<sup>41-42</sup> Therefore, if one is able to properly “normalize” the weight matrices during training and prevent the ill-conditioning, the resulting model will be smooth. Spectral normalization<sup>42</sup> is such a technique working according to this principle, which normalizes the leading singular value of the weight matrices, and it has been shown that spectral normalization can significantly stabilize the training of generative adversarial networks (GANs)<sup>43</sup> and energy-based models (EBMs).<sup>44</sup> Besides, additional regularization techniques which could facilitate the training of the deep ANNs are often simultaneously implemented. Particularly, for over-parametrized molecular models, it is important to guarantee that the models are less prone to overfitting. For this purpose, one can directly borrow the general techniques developed in deep learning community, for instance, L2-regularization<sup>45</sup> and multi-task learning,<sup>46</sup> etc.

Another related issue is that ANNs usually work robustly only for inputs and outputs which are neither very small nor very large. However, in molecular modeling, the value of the output or label (e.g., energy or inter-particle distance) is often unbounded. To address this issue, one common approach is to rescale the output using suitable constants.<sup>47</sup> Under some settings, although the value of the output is unbounded, we may be only concerned with values within a certain range. If this is the case, one can first decide on a value range of interest, then discretize this range with a sufficient number of bins. A simplex over these discretized bins can then be predicted by the ANNs (e.g., via softmax).<sup>34, 48</sup>



In addition to the gradient regularizer, choice of activation functions is also critical to the performance and smoothness of the deep molecular models. Instead of the saturating functions like sigmoid or hyperbolic tangent, quasi-linear activation functions with stronger Lipschitz-continuity are preferred for molecular modeling. Such activation functions include but not limit to (shifted) softplus,<sup>49-50</sup> exponential linear units (ELU),<sup>51</sup> Swish,<sup>52</sup> etc, and there is growing evidence that these activation functions could help achieve desired performance on molecular property predictions.<sup>35, 37</sup>

## Section II. Deep supervised learning for molecular modeling and simulations

The aim of deep supervised learning is to find a model  $f_\theta(\mathbf{x})$ , which inputs  $\mathbf{x}$  and predicts the corresponding label  $A(\mathbf{x})$ . The core of supervised learning lies in the acquisition of the labels. In machine learning, classification tasks and regression problems all belong to the realm of supervised learning. Supervised learning is perhaps the most widely exploited learning scheme for deep molecular modeling. For instance, there were several attempts to extract order parameters via supervised learning.<sup>53-54</sup> Besides, many studies tried to fit PES or free energy surfaces (FES) with regressive deep learning, and we will present several examples in the following.

### 1. Fitting potential energy surfaces

If the atomic positions  $\mathbf{x}$  are chosen as the input, and the value of the corresponding potential energy  $U(\mathbf{x})$  is provided as the label, one can formulate a regression problem where a model  $f_\theta(\mathbf{x})$  is trained to minimize the loss function ( "loss function" in this paper is interchangeable with the "cost function" in machine learning literature) in Eq. (8),

$$L(\theta) = \langle |f_\theta(\mathbf{x}) - U(\mathbf{x})|^2 \rangle_{\mathbf{x}} \quad (8)$$

where  $\langle A(\mathbf{x}) \rangle_{\mathbf{x}}$  stands for the expectation value of  $A(\mathbf{x})$  over a sample set of  $\mathbf{x}$ . In this way, one is trying to fit a PES using a deep molecular model  $f_\theta(\mathbf{x})$ . At the first sight, fitting an energy function  $U(\mathbf{x})$ , the analytical form of which is known, by another unknown function  $f_\theta(\mathbf{x})$  makes little sense. However, if the evaluation of  $f_\theta(\mathbf{x})$  is significantly cheaper than  $U(\mathbf{x})$ , replacing  $U(\mathbf{x})$  by  $f_\theta(\mathbf{x})$  will help reduce a great deal of computational effort. This is often the case in quantum chemistry and *ab initio* molecular dynamics (AIMD),<sup>55</sup> where the evaluation of even a single-point energy entails heavy computation. Traditionally, one has to trade the accuracy for computational speed in order to apply AIMD or quantum-level calculations in complex molecular systems. While by virtual of deep molecular models, one can now perform simulations or single-point energy calculations on a surrogate deep PES, thus striking a good balance between accuracy and computational complexity.

Although fitting a PES appears to be a simple deep learning task, the model architectures and the training objectives have undergone dramatic evolution along history. One of the earliest attempts along this line is BPNN.<sup>21</sup> As introduced earlier, BPNN encodes the molecular configuration into atom-wise descriptors. Based on these AEV descriptors, a set of MLPs output the atomic energy for each atom, the summation of which gives the overall potential energy. BPNN is trained mainly through energy matching, i.e., Eq. (8). Due to the relative demanding requirements of suitable AEV descriptors, applications of BPNN and its following variants were mostly limited to simple systems, for example, small organic molecules<sup>22</sup> or homogeneous material systems.<sup>27</sup>

As another descriptor-based model, DPMD<sup>2</sup> adopts a different feature-engineering strategy so that it is capable of handling larger and even more complex systems. Besides, DPMD included and emphasized more on a force-matching term in the training objective in addition to the energy-matching term. Apparently, the force-matching term can be viewed as one kind of multi-task learning, thus helps regularize the gradients or smoothness of the model. We remark here that the force-matching brings additional benefit from the perspective of data augmentation. As introduced, the foundation of a successful supervised learning is the acquisition of labels, so data augmentation is a very important technique for supervised learning. Given a fixed dataset of  $\{\mathbf{x}, U(\mathbf{x}), \nabla_{\mathbf{x}}U(\mathbf{x})\}$ , predicting the force at  $\mathbf{x}$  is roughly equivalent to predicting the energy at an adjacent point, that is,  $U(\mathbf{x} + d\mathbf{x})$ . Therefore, inclusion of the force-matching term implicitly augments the training data. Other techniques for explicit data augmentation, such as active learning and adaptive sampling, were also introduced by related studies.<sup>37</sup>

On the other end, GNN-based models have seen rapid development during the last couple of years. For example, SchNet<sup>35</sup>, DTNN,<sup>36</sup> PhysNet<sup>37</sup> and DimeNet<sup>56</sup> raised the bar of the performance of deep molecular models on many benchmark tasks. Besides, the authors of PhysNet found it necessary to incorporate known physics laws in order to achieve correct asymptotic behavior in the long range. Specifically, they additionally predicted the atomic partial charges as part of the overall training objective. From the viewpoint of physics, predicting atomic charges is equivalent to giving the electrons a separate description, and also allows the incorporation of the Coulomb’s law to capture the correct long-range interactions.<sup>37</sup> Similar ideas were also employed by TensorMol<sup>23</sup> which separately predicted van der Waals interactions which help correct the short-range asymptotic behavior. We comment here that such change of the training objective makes sense in that asymptotic behavior can be regarded as a special type of inductive biases, and designing loss functions that explicitly account for the asymptotic behavior will help lead to better models.

Reflecting the history from BPNN to PhysNet, in addition to designing ever more complex molecular models, we should re-think how to *correctly* exploit them. Of course we can regard the deep ANN as a sheer black-box function approximator, and use it to directly fit the overall energy function (as in BPNN). Indeed, the training objectives introduced in the above examples are all empirical risks defined by the users. For supervised learning, the choice of the empirical risk is often vital to the performance of the final model. A recent study<sup>57</sup> indicates that if merely based on the empirical risks like Eq. (8), the partition of the overall energy into atomic contributions could be arbitrary and non-interpretable as evidence of overfitting for models with sufficient capacity. Therefore, before deciding on an empirical risk as the loss, one should reflect whether there is any alternative learning objective that can lead to better sample efficiency and model transferability.

## 2. Fitting potential of mean force

To investigate molecular systems, one is often not interested in the potential energy  $U(\mathbf{x})$ , but the free energy or potential of mean force (PMF),  $F(\mathbf{q})$ , of some reduced descriptors  $\mathbf{q}$  like coarse-grained (CG) variables<sup>58</sup>:

$$p(\mathbf{q}) = \frac{\int e^{-\beta U(\mathbf{x})} \delta(\mathbf{q} - \mathbf{q}(\mathbf{x})) d\mathbf{x}}{\int e^{-\beta U(\mathbf{x})} d\mathbf{x}} = \frac{e^{-\beta F(\mathbf{q})}}{Z} \quad (9)$$

$$F(\mathbf{q}) = -\frac{1}{\beta} [\log p(\mathbf{q}) + \log Z] \quad (10)$$

where  $Z = \int e^{-\beta U(\mathbf{x})} d\mathbf{x}$  is the partition function,  $\beta$  is the inverse temperature and  $\delta$  the Dirac delta function. If  $\mathbf{q}$  corresponds to the slowly changing variables governing the process of interest,  $F(\mathbf{q})$  becomes a CG description of the original thermodynamic system, and simulations performed under  $F(\mathbf{q})$  are generally much faster than those run on the fine-grained (FG) potential  $U(\mathbf{x})$ . Therefore, Equations (9-10) are also known as the principle of thermodynamic consistency for coarse graining.<sup>59</sup>

However, it is often challenging to derive a reliable analytical form for  $F(\mathbf{q})$  given access to samples drawn from  $U(\mathbf{x})$ , particularly when  $\text{Dim}(\mathbf{q})$  is large ( $\text{Dim}(\cdot)$  is short for the dimensionality). This problem is particularly pronounced in the realm of coarse graining. Considered that ANNs may offer extra flexibility and expressivity to this end, several recent studies proposed supervised learning approaches to fit  $F(\mathbf{q})$  by ANNs.<sup>60-62</sup> For instance, Schneider *et al.* gridded the possibly high-dimensional CG variable space, and adopted enhanced sampling techniques to generate samples over the  $M$  grids. Based on

these samples denoted by  $\{\mathbf{q}^{(i)}\}_{i=1,\dots,M}$ , the corresponding free energy,  $F(\mathbf{q}^{(i)})$ , and the mean force,  $-\nabla_{\mathbf{q}^{(i)}}F(\mathbf{q}^{(i)})$ , at these gridded points can be estimated, and their values were used to train an ANN  $F_\theta(\mathbf{q})$  in order to fit the FES. The training objective is to minimize the empirical risk in Eq. (11),

$$L(\theta) = \langle |F_\theta(\mathbf{q}^{(i)}) - F(\mathbf{q}^{(i)})|^2 \rangle_i \quad (11)$$

Wang *et al.* further extended this idea into particle-based coarse graining models which are widely adopted for bio-molecular systems.<sup>4</sup> Since the FES of a particle-based model should satisfy the inductive biases of molecular models, Wang *et al.* trained an CG potential  $F_\theta(\mathbf{s})$  using one kind of deep molecular model (called ‘‘CGnet’’) to represent the FES of the CG particles, and trained the CGnet using the mean-force information,

$$L(\theta) = \langle \left\| \nabla_{\mathbf{q}^{(i)}} F_\theta(\mathbf{q}^{(i)}) - \nabla_{\mathbf{q}^{(i)}} F(\mathbf{q}^{(i)}) \right\|_F^2 \rangle_i \quad (12)$$

The authors of CGNet found that imposing physics-based restraints into the overall model is essential for a reasonable performance. Inclusion of physics-based restraints in  $F_\theta$  is straightforward thanks to the additive compositionality of the energy function. Noteworthy, both Eq. (11) and Eq. (12) coined a regression problem of  $F(\mathbf{q})$  via empirical risk minimization (ERM). If  $\mathbf{q}^{(i)}$  is drawn from the equilibrium distribution, Eq. (12) is also known as score-matching and equivalent to minimizing the Fisher’s divergence.<sup>63</sup> However, since the training data is manually curated and generally off-equilibrium, ERM is very sensitive to the sample coverage and the formula of the loss function. For example, replacing the L2-norm or the Frobenius norm in Eqs. (11-12) with other norms, or adding weights to the norm calculation would dramatically influence the model performance, thus the choice of proper dataset and loss function can be very tricky. Besides, fitting  $F(\mathbf{q})$  in a regression fashion has several potential drawbacks: It necessitates gridding the space of  $\mathbf{q}$ , which would be computationally prohibitive for large  $\text{Dim}(\mathbf{q})$ ; Beside, it is rather data-inefficient because calculating  $F(\mathbf{q})$  at one point requires a large amount of samples from  $U(\mathbf{x})$  at the neighborhood of  $\mathbf{q}$ , and requires calculation of the mean forces which is usually not done in practice. We will show in the following section that, instead of ERM, distribution learning and density estimation methods are more mathematically proper ways to deal with these problems (i.e., coarse graining or inferring FES), and they are all central topics of unsupervised learning.

### Section III. Deep unsupervised learning for molecular modeling and simulations

Although most successful applications of deep molecular models are centered at supervised learning, many challenging problems in molecular modeling and simulations are of unsupervised nature. For instance, dimensionality reduction and clustering are two commonly encountered unsupervised learning problems in molecular simulations. Besides, density estimation is another important unsupervised learning problem which is widely involved in molecular modeling scenarios like free energy calculation and coarse graining. To tackle these problems via deep learning, one needs to translate the physical or mathematical nature of the problems into optimizable (or possibly, variational) objectives. In the following we will present several examples in which deep unsupervised learning was implemented to eliminate possible subjective biases and offer more flexible solutions to these problems.

#### 1. Dimensionality reduction and clustering

Many molecular dynamic processes in chemistry and biology, e.g. chemical reaction, protein folding and ligand binding etc., can be described by a clustered free energy landscape (FEL)<sup>64</sup> consisting of many local free energy minima which correspond to the metastable states. This picture of a clustered FEL (or the metastability) is the cornerstone of many kinetic models dealing with diffusive and complex dynamics, and a simplified and informative visualization of complex FEL amenable for downstream tasks such as clustering is often required by these kinetic models. Zhang *et al.* recently proposed a parametric learning approach, called Information Distilling of Metastability (IDM), to perform clustering in the meantime of reducing the dimensionality for molecular systems.<sup>8</sup> IDM is end-to-end differentiable thus scalable to ultra-large dataset. Remarkably, IDM neither requires a cherry-picked distance metric nor the ground-true number of clusters defined *a priori*, and it can be used to unroll and zoom-in the hierarchical FEL with respect to different timescales.

IDM is based on the idea of information co-distillation originated from machine learning community.<sup>65-66</sup> Given two observations  $\mathbf{x}$  and  $\mathbf{x}'$  belonging to the same metastable state, one can find a function  $\chi$  capturing what is in common between them by maximizing the mutual information (MI),  $I(\chi(\mathbf{x}), \chi(\mathbf{x}'))$ . In order to avoid the trivial solution of  $\chi$  being the identity function, Zhang *et al.* confines  $\chi$  to the family of classification functions (or indicator functions) with finite categories. Consequently, the entropy of  $\chi$  is upper bounded, hence the information is bottlenecked and distilled. Clustering can thus be achieved by maximizing the following closed-form mutual information:

$$I_{\beta}(\chi(\mathbf{x}), \chi(\mathbf{x}')) = \sum_{c,c'=1}^K \langle \chi_c(\mathbf{x}), \chi_{c'}(\mathbf{x}') \rangle \ln \frac{\langle \chi_c(\mathbf{x}), \chi_{c'}(\mathbf{x}') \rangle}{\langle \chi_c(\mathbf{x}) \rangle^{\beta} \langle \chi_{c'}(\mathbf{x}') \rangle^{\beta}} \quad (13)$$

where a controlling hyper-parameter  $\beta$  is introduced without loss of generality.  $\chi_c(\mathbf{x})$  denotes the  $c$ -th entry of  $\chi$  corresponding to the probability of  $\mathbf{x}$  belonging to cluster  $c$ . Intuitively, Eq. (13) means that the clustering mapping of  $\mathbf{x}$ , namely  $\chi(\mathbf{x})$ , should be maximally informative with its temporal neighbors. In practice, this is obtained when  $\chi(\mathbf{x}) \approx \chi(\mathbf{x}')$  for arbitrary  $\mathbf{x}$ .

Since Eq. (13) is a variational objective, one is motivated to adopt ANNs as the clustering function  $\chi$  because they usually offer sufficient variational flexibility over high dimensional data. In order to “hit two birds with one stone”, that is, reduce the dimensionality while clustering, IDM employs the Matching Networks,<sup>67</sup> a special deep neural network architecture consisting of two twin networks and mapping two different vector spaces onto a same vector space. One network projects the molecular configuration  $\mathbf{x}$  to a low dimensional *matching space* which serves as the reduced representation of FEL. The other twin-

network maps constant vector  $\mathbf{c}$ , which acts as the preimage of a cluster centroid, to the same matching space (Fig. 2A). The final soft clustering labels  $\chi(\mathbf{x})$  are then defined by the similarity between the projected images of  $\mathbf{x}$  and  $\mathbf{c}$  on the matching space. Noteworthy, IDM does not require a reference to the exact number of ground-true clusters. Instead, IDM robustly clusters the data as long as the maximal allowed number of clusters,  $K$  in Eq. (13), is large enough. Besides, IDM yields soft cluster labels  $\chi(\mathbf{x})$ , which are preferred in many scenarios to hard ones. Finally, trained upon  $N$  decreasing temporal neighborhood sizes ( $\tau_1 \gg \tau_2 \gg \dots \gg \tau_N$ ), IDM allows one to zoom-in the FEL with increasing time resolutions, which is equivalent to performing a top-down divisive clustering, hence IDM is able to extract the hierarchy of the FEL according to timescales rather than geometry-based metrics.

As an example, IDM was divisively performed on Ala2 for illustration (Figs. 2B and 2C).<sup>8</sup> Three timescales were chosen for training: (1)  $\tau = 200$  ps, (2)  $\tau = 20$  ps, and (3)  $\tau = 2$  ps. In all the three cases, the maximal allowed cluster number  $K$  was set to be 16. On the longest timescale ( $\tau = 200$  ps), IDM partitions all conformations into two metastable states (Fig. 2B, panel 1), corresponding to the *cis/trans*-isomers of the torsional angle  $\phi$ . This is in good agreement with kinetic modeling result that isomerization of  $\phi$  is much slower than  $\psi$ . As expected, the reduced representation obtained by IDM at this timescale only preserves two distinguishable metastable states (Fig. 2C, panel 1). When progressing to a smaller timescale of 20 ps, IDM further divides the conformations into four clusters, as reported in previous sections (panel 2 in Fig. 2B), and the reconstructed FEL now consists of 4 distinguishable metastable states (panel 2 in Fig. 2C). At an even smaller timescale,  $\tau = 2$  ps, IDM categorized all the conformers of Ala2 into 5 metastable states (Fig. 2B, panel 3) and learns a reduced representation accordingly (Fig. 2B, panel 3). This example shows that IDM enables one to zoom-in the FEL with increasing time resolutions and track the hierarchy of metastable states accordingly.

## 2. Density estimation

In the previous section, we discussed the pros and cons for constructing a FES  $F(\mathbf{q})$  given access to samples drawn from  $U(\mathbf{x})$  via supervised empirical risk minimization. An alternative (perhaps also more reasonable) view of inferring  $p(\mathbf{s})$  is provided by statistical modeling and unsupervised learning, in which density estimation of high-dimensional data has been a long-standing goal.<sup>19, 68</sup> Conventionally, if  $\mathbf{q}$  is low-dimensional (say,  $\text{Dim}(\mathbf{q}) \leq 3$ ), non-parametric methods like kernel density estimation (KDE)<sup>69</sup> can be adopted to infer  $F(\mathbf{q})$ , but they become quickly infeasible as  $\text{Dim}(\mathbf{q})$  increases. Recently, a burst of work sparks new ideas to exploit deep learning to this end, giving rise to an active research field of generative learning and models including variational auto-encoders,<sup>70</sup> GANs,<sup>43</sup> auto-regressive,<sup>71</sup> and normalizing-flow models.<sup>72</sup> Nevertheless, these methods are mostly footed on certain simple prior distributions, hence they are likely to suffer from issues like mode-dropping and often assign probability mass to areas unwarranted by the data.<sup>73</sup> Worse still, these generative models generally do not comply with the inductive biases of molecular systems. In contrast, generative learning with EBMs has an even longer history,<sup>74-76</sup> which in principle can fit arbitrarily complex distributions due to the flexibility and plasticity of the energy landscapes.<sup>44, 77</sup> Based on EBMs, Zhang *et al.* recently proposed a variational approach, called variational inference of free-energy (VIFE),<sup>3</sup> to approximate the FES ( $F(\mathbf{q})$ ) by parametric models and without supervision.

Specifically, let us denote the approximate free energy function and the associated probability distribution as  $F_\theta(\mathbf{q})$  and  $p_\theta(\mathbf{q})$  respectively (where  $\theta$  are optimizable parameters), and define the Kullback-Leibler divergence,  $D_{\text{KL}}(p||p_\theta)$ , between  $p$

and  $p_\theta$ .  $D_{\text{KL}}(p||p_\theta)$  is a strict divergence in that  $D_{\text{KL}}(p||p_\theta) \geq 0$  where the equality holds if and only if  $p = p_\theta$ , hence can be used as a variational objective.<sup>78-80</sup> Particularly, the gradient of  $D_{\text{KL}}(p||p_\theta)$  w.r.t.  $\theta$  takes the following form in Eq. (14),

$$\nabla_\theta D_{\text{KL}}(p||p_\theta) = \langle \beta \nabla_\theta F_\theta(\mathbf{q}) \rangle_{p(\mathbf{q})} - \langle \beta \nabla_\theta F_\theta(\mathbf{q}) \rangle_{p_\theta(\mathbf{q})} \quad (14)$$

If one has access to, say, equilibrium MD samples drawn from  $U(\mathbf{x})$ , the distribution of which is denoted as  $p_{\text{FG}}(\mathbf{q})$ , one can use  $p_{\text{FG}}$  to approximate  $p$  and optimize  $F_\theta$  w.r.t. a surrogate objective  $D_{\text{KL}}(p_{\text{FG}}||p_\theta)$ . Since the analytical forms of  $F_\theta(\mathbf{q})$  and  $\nabla_{\mathbf{q}} F_\theta(\mathbf{q})$  are both available, one can perform CG simulations on  $F_\theta(\mathbf{q})$  to estimate  $\langle \nabla_\theta F_\theta(\mathbf{q}) \rangle_{p_\theta(\mathbf{q})}$ . Due to the simplicity of  $F_\theta(\mathbf{q})$ , the CG simulations are usually computationally economic and converge relatively fast, hence brute-force simulations via Monte Carlo (MC) sampling or Langevin dynamics generally suffice.

As an example, VIFE was benchmarked on a 2-dimensional 3-well toy model, the PES of which is shown in Fig. 3A.<sup>3</sup> Langevin dynamics simulation was first performed on  $U(x, y)$  and yielded 50,000 samples which were used as  $p_{\text{FG}}$ . The CG variable  $\mathbf{q}$  was identical to  $(x, y)$ , and  $F_\theta$  was built in order to reproduce  $U(x, y)$  rather than coarse graining. An ANN representing  $F_\theta$  was trained according to Eq. (14). It can be seen that  $D_{\text{KL}}$  between the two distributions quickly diminished during training, and the optimization of  $F_\theta$  converged within 50 training epochs (each epoch is composed of 100 mini-batch optimization steps). Compared to the original PES, the optimized  $F_\theta$  preserves all the three local minima correctly, showing no signs of mode-dropping (Fig. 3B). This is an advantage of EBMs over many other generative methods because metastable states usually play functionally important roles for bio-molecules, and dropping any of them during may lead to unreasonable models. Besides, the overall contour and landscape of  $F_\theta$  also resemble  $U(x, y)$ , especially on regions with higher densities. Now that the analytical form of  $F_\theta$  was obtained based merely on samples from  $U(x, y)$  (rather than knowing the mathematical form of  $U(x, y)$ ), one can locate the free energy minima and cluster samples. The simulation samples were minimized over  $F_\theta$ , and all the samples finally fell into three distinct local minima as is shown by different colored symbols in Fig. 3C, and the noisily distributed samples were indeed assigned to different metastable states quite reasonably.

In addition to the above example, density estimation and generative learning via EBMs can also be adopted for prediction tasks. For example, Ingraham *et al.* proposed a generative model to predict folded protein structures.<sup>81</sup> They first constructed a Markov Random Field (MRF) via deep ANNs using coevolution information such as homologous sequences to quantify the effective energy of a specific protein conformation. Meanwhile, they also parametrized a Langevin dynamics with a trainable diffusion field, in combination with the drifting field governed by the MRF. Running this Langevin dynamics thus allows sampling of protein conformations given a particular amino acid sequence. Because the overdamped Langevin dynamics simulator (both the drifting field and diffusion field) is end-to-end differentiable, the authors termed it as a ‘‘Differentiable Simulator’’, and they trained it by maximizing the likelihood of the generated conformations w.r.t. the known folded structures. Similarly, AlphaFold<sup>48</sup> also predicts a PMF for the protein conformations based on deep learning over coevolution information. AlphaFold takes advantage of the compositionality of EBMs, and decomposed the PMF into additive and interpretable components including, e.g., one term accounting for the pair-wise distances between residues and one for the backbone torsional angles. In the Differentiable Simulator and AlphaFold, how to perform sampling within EBMs is worthy of particular discussion. In the Differentiable Simulator, the authors trained a sampling engine in addition to the energy function. In contrast, instead of performing equilibrium sampling (as by Langevin dynamics), AlphaFold predicts the protein structures by direct gradient descent over the energy function thus avoiding the explicit sampling procedure, but such generating procedure no longer allows maximum likelihood optimization as in the Differentiable Simulator. Indeed, many recent studies were trying to

develop gradient-descent-like inference method which also supports maximum likelihood optimization of EBMs for prediction tasks.<sup>82-83</sup> Nonetheless, sampling is still a critical obstacle for most practitioners of EBMs. Once the sampling issue is better solved, energy-based generative learning would find more applications in other challenging molecular modeling tasks such as inverse coarse graining. In the next section, we will discuss how reinforcement learning could offer advanced solutions to the challenging sampling problem.



## Section IV. Deep reinforcement learning for molecular modeling and simulations

Under settings of a conventional reinforcement learning (RL) task, the agent is trained to find the optimal strategy,  $\pi(a, s)$  which dictates the probability of choosing an action  $a$  at the state  $s$ , in order to maximize a specific (task-related) reward function  $R(s)$ .<sup>84</sup> Usually the RL problem consists of a sequence of decision making, leading to a trajectory of state-action pairs along with the intermediate and final accumulated rewards. Due to its definition, RL is arguably the most general framework for machine learning. It has drawn increasing attention from deep learning community and seen the most burgeoning development in recent years. Although RL seems not directly related to molecular modeling and simulations in light of its definition, the philosophy and mathematical framework of RL are deeply connected to statistical physics.

One of the mathematical cornerstones of RL is *expectation maximization*,<sup>85-86</sup> which arises from the fact that one can only manage to maximize the *expected* reward due to the stochastic and noisy environment. In order to calculate the expectation, sampling according to the policy is thus inevitable. Consequently, the sampling issue suffered by molecular simulations is also pronounced in RL. Moreover, since the policy is ever-changing during training, the dataset is always dynamic in RL. Therefore, how to handle partial sampling and dynamic dataset remains as central research topics in RL, which may also help address similar issues encountered in molecular modeling and simulations.

Besides, the concept of “reward” is quite ambiguous. Actually the reward function  $R(s)$  can be defined arbitrarily. Therefore, if one can define a proper reward specific to the concerned problem, one can naturally deploy methods and techniques from RL to solve their own problems. For instance, some adaptive sampling strategies were newly proposed in the spirit of RL.<sup>61, 87</sup> Although the authors did not rigorously formulate their methodologies in terms of RL, they all signified that their methods take a flavor of RL. In fact, these adaptive sampling methods all belong to a category called curiosity-based RL,<sup>88</sup> which is an active research field in machine learning community.

Perhaps more interesting connections between RL and EBMs can be revealed by the recently developed deep RL algorithms for continuous state and action spaces.<sup>89-90</sup> Since the policy is defined over a continuous space of actions, it is natural to parametrize the policy in the form of energy functions, so that the probability of picking a certain action is governed by the Boltzmann distribution. In turn, we can also regard the Boltzmann distribution induced by the energy function in EBMs as a special “policy”, under which the sampler (an imaginary agent) is traveling through the phase space. In the following we will present several examples which incorporate these interesting ideas and techniques from RL in molecular modeling and simulations.

### 1. Actor-Critic enhanced sampling

Many dynamic events of interest in molecular systems, e.g. phase transitions, protein folding and ligand binding, are embedded in a complex free energy landscape where long-lived metastable states are separated by kinetic bottlenecks (i.e. free energy barriers) which impede the transitions.<sup>91</sup> Therefore, timescales of these rare events are well out of the reach of brute-force simulations, and enhanced sampling methods are largely needed to this end. Zhang *et al.* recently re-formulated the enhanced sampling problem as a distribution learning problem, and proposed a *minimax* game as in GANs to solve it. The proposed approach is called targeted adversarial learning optimized sampling (TALOS).<sup>6</sup>

TALOS simultaneously trains a bias potential,  $V_\theta(\mathbf{x})$ , which is applied to a simulation engine like MD (which plays a role as the “generator” in GANs), and a discriminator  $D_w(\mathbf{q}(\mathbf{x}))$  that attempts to differentiate samples generated by the biased

sampler from those drawn from the desired target distribution  $p(\mathbf{q})$  (where  $\mathbf{q}$  is a descriptor as a function of  $\mathbf{x}$ ). Note that  $\theta$  and  $w$  denote the trainable parameters of the deep molecular models. The biased sampler receives a reward when it fools the discriminator, whereas the discriminator is rewarded when it successfully classifies a sample as from the “real” target distribution or the “fake” biased simulation. The training ends when the *Nash equilibrium* of the game is reached. Inspired by Wasserstein-GAN<sup>40</sup>, the entire training process of TALOS is equivalent to solving the following *minimax* problem:

$$\min_{V_\theta} \max_{D_w \sim \mathbb{L}_1} [\langle D_w(\mathbf{q}) \rangle_{p(\mathbf{q})} - \langle D_w(\mathbf{q}) \rangle_{p_\theta(\mathbf{q})}] \quad (15)$$

where  $\max_{D_w \sim \mathbb{L}_1} [\langle D_w(\mathbf{q}) \rangle_{p(\mathbf{q})} - \langle D_w(\mathbf{q}) \rangle_{p_\theta(\mathbf{q})}]$  is the Wasserstein-1 distance<sup>92</sup> between the target distribution  $p$  and the biased distribution  $p_\theta \propto \exp(-\beta(U(\mathbf{x}) + V_\theta(\mathbf{x})))$ .  $D_w(\mathbf{q})$  belongs to the family of 1-Lipschitz ( $\mathbb{L}_1$ ) functions mapping the descriptor space  $\mathbf{q}$  to the real space. However, the analytical form of the optimal  $D_w$  is unknown, so a neural network conforming to a Lipschitz restraint, parametrized by  $w$ , is employed to approximate it following the gradient:

$$\nabla_w L(w) = \langle \nabla_w D_w(\mathbf{q}) \rangle_{p_\theta} - \langle \nabla_w D_w(\mathbf{q}) \rangle_p \quad (16)$$

Once the discriminator  $D_w$  is fixed, the bias potential  $V_\theta(\mathbf{x})$  is optimized following the gradient:

$$\nabla_\theta L(\theta) = \langle (D_w(\mathbf{q}) - \langle D_w(\mathbf{q}) \rangle_{p_\theta}) \beta \nabla_\theta V_\theta(\mathbf{x}) \rangle_{p_\theta} \quad (17)$$

where  $\beta$  is the inverse temperature factor. Equations (16-17) bear intriguing similarity with the Actor-Critic algorithm,<sup>93</sup> hence rendering TALOS an interpretation in terms of RL: Equation (16) is parallel to the training of a Critic which is used to approximate the value function of being in a state  $\mathbf{q}(\mathbf{x})$ ; whereas Eq. (17) is analogous to the famous REINFORCE algorithm (also known as the policy gradient or Expectation Maximization)<sup>85</sup> which maximizes the expected value  $\langle D_w(\mathbf{q}) \rangle_{p_\theta}$  for the Actor who visits state  $\mathbf{q}(\mathbf{x})$  with a probability proportional to  $\exp(-\beta(U(\mathbf{x}) + V_\theta(\mathbf{x})))$ . With this insight, one can develop variants of TALOS by defining other task-specific value functions and manipulate the PES accordingly.

Noteworthy, TALOS does not require the bias potential and the target distribution to operate in the same space, providing additional flexibility in designing and applications of the method. As an example, TALOS was applied to an all-atom simulation of the  $S_N2$  reaction  $\text{Cl}^- + \text{CH}_3\text{Cl} \leftrightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$  (Fig. 4A).<sup>6</sup> The difference between the two C-Cl bond distances  $q^* = d_1 - d_2$  serves as a 1-dimensional descriptor (Figs. 4A and 4B) to define the target distribution (Fig. 4C), whereas a 2-dimensional bias potential as a function of  $(d_1, d_2)$  is constructed. During training, the bias potential was updated in an interpretable manner similar to metadynamics: The critic appears to be able to “feel” the difference between the target and current distributions so that a positive bias potential was added to penalize where was oversampled and a negative one to encourage where was undersampled (Figs. 4D and 4E). As a consequence, the system was driven to the predefined target distribution giving rise to faster chemical transitions (Fig. 4D). Intriguingly, although the target descriptor is 1-dimensional, the bias potential optimized by TALOS is complementary to the 2-dimensional FES (Fig. 4E). Given that the bias potential and target distribution can be flexibly chosen, TALOS is applicable to a variety of situations where inference or modification to the Hamiltonian is needed in order to achieve a specific distribution in addition to enhanced sampling.

## 2. Reinforced multi-scale molecular modeling

As introduced in the previous section, VIFE can be used to perform density estimation or coarse graining given equilibrium samples from a fine-grained (FG) simulation, denoted by  $p_{\text{FG}}$ . However, in a more common setting where equilibrium FG

samples are not available beforehand, one has to perform sampling over  $U(\mathbf{x})$  from scratch. Since  $\text{Dim}(\mathbf{x})$  is usually very large, estimating ensemble averages over  $p(\mathbf{q})$  in Eq. (14) is often infeasible for brute-force FG (e.g., all-atom or *ab initio*) simulations.

Zhang *et al.* showed that VIFE combined with RL, provides a new solution to the this sampling problem, and called this new method reinforced VIFE (RE-VIFE).<sup>3</sup> Inspired by the off-policy RL,<sup>90</sup> RE-VIFE exploits a two-timescale learning scheme, where a bias potential  $V_\phi(\mathbf{q})$  with  $\phi$  denoting optimizable parameters (equivalent to a policy function in RL) is separately trained in addition to  $F_\theta(\mathbf{q})$  which can now be viewed as a value function in the spirit of Actor-Critic RL.<sup>94</sup> In RE-VIFE, a target distribution  $p_T(\mathbf{q})$  is defined according to the value function  $F_\theta(\mathbf{q})$ , and the target distribution encourages the sampler to explore less-visited regions. To move the sampling distribution towards  $p_T$ , the bias potential  $V_\phi(\mathbf{q})$  can be optimized by minimizing a strict divergence, for instance,  $D_{\text{KL}}(p_T||p_\phi)$ ,

$$\nabla_\phi D_{\text{KL}}(p_T||p_\phi) = \langle \beta \nabla_\phi V_\phi(\mathbf{q}) \rangle_{p_T} - \langle \beta \nabla_\phi V_\phi(\mathbf{q}) \rangle_{p_\phi} \quad (18)$$

where  $p_\phi$  denotes the Boltzmann distribution under  $(U + V_\phi)$ . For example, if one chooses a well-tempered target distribution (Eq. (19)),

$$p_T = p_\theta^\gamma \propto \exp\left(-\frac{\beta}{\gamma} F_\theta\right) \quad (19)$$

where  $\gamma > 1$  is the well-tempering factor,<sup>95</sup> then the optimal  $V_\phi$  will converge to the well-tempered free energy. In terms of imitation learning,<sup>96</sup>  $F_\theta(\mathbf{q})$  plays the role of a leader that coins a moving target based on the current density estimation, while  $V_\phi(\mathbf{s})$  learns to tune the policy in order to catch up (Fig. 5A). A dramatic advantage of such leader-chaser scheme lies in the fact that  $F_\theta$  along with  $p_T$  is constructed based on the samples drawn from simulations under  $V_\phi$ , so  $p_T$  and  $p_\phi$  always share substantial overlap; otherwise  $D_{\text{KL}}$  would fall victim to the notorious vanishing gradient issue.<sup>40</sup>

As an example, consider the problem of building a CG model for a prototypical bio-molecular system, alanine dipeptide (Ala2) in explicit water, where the backbone torsional angles,  $\mathbf{q} = (\phi, \varphi)$ , were chosen as the CG variables (Fig. 5B).<sup>3</sup> Assuming that no FG samples were provided *a priori*, one has to perform sampling over the all-atom model from scratch. However, since isomerization of  $(\phi, \varphi)$  is a rare event due to relatively high free energy barriers, brute-force FG simulations of Ala2 converge too slowly to obtain an accurate estimate of  $\langle \nabla_\theta F_\theta(\mathbf{q}) \rangle_{p_{\text{FG}}}$ . RE-VIFE was thus adopted to enhance the FG sampling over  $\mathbf{q}$ , and two simulations were launched simultaneously: one FG (all-atom) MD simulation under a bias potential  $V_\phi(\mathbf{q})$ , and a CG MC simulation over  $F_\theta(\mathbf{q})$ . Figure 5C showcases a 1-ns FG simulation trajectory for torsions  $\phi$  and  $\varphi$  produced by vanilla MD was presented in contrast to those produced by simulations biased by  $V_\phi$ . It can be found that the brute-force MD can hardly produce equilibrium samples covering all important metastable states in this short simulation length. In contrast, RE-VIFE helps enhance the sampling of FG models in that the isomerization of torsion  $\phi$  becomes reasonably frequent. Such a boosted sampling efficiency can be explained by the optimized bias potential  $V_\phi$  via RE-VIFE (Fig. 5D), which appears complementary to the ground-true FES of  $\mathbf{q} = (\phi, \varphi)$  thus effectively flattens the FES. On the other hand, as samples from  $V_\phi$  are better representatives of  $p_{\text{FG}}$  (excellently covering both the free-energy minima and the transition regions), they can be reliably used to optimize the CG model. The final CG model ( $F_\theta$ ) optimized via RE-VIFE for Ala2 in explicit water is shown in Fig. 4E. Noteworthy,  $F_\theta$  captures all known metastable states of Ala2 w.r.t.  $(\phi, \varphi)$  (i.e., no mode-dropping), and

quantitatively agrees well with the ground truth (Fig. 5E), demonstrating that simulations on multiple scales can be bridged by RE-VIFE and that CG models can be reliably inferred even without access to FG samples *a priori*.

## Outlook

Despite the encouraging progress so far, there are still many obstacles limiting the development and applications of deep molecular models. As a closing to this review, we will enumerate some issues that cause wide concerns and critics over the role of deep learning in molecular modeling and simulations, and outlook possible solutions to address these issues.

### 1. Transferability

Although there are already attempts using deep molecular models to construct deep PES, unfortunately, they have proved limited transferability or “generalizability” in practice. In other words, to establish a deep PES for a new system, one has to build up the model from scratch, including choosing a proper model architecture, creating and curating the training data, optimizing and validating the model etc. Frustratingly, it remains hard to directly re-use models previously trained by others. Such non-transferability is very unfriendly to users, thus severely limits the usage of current deep PES models. Though the model architectures have evolved to be more transferable than ever before, the transferability of a deep PES is still hindered by the empirical risk which is used as the learning objective.

One may argue that the PES can be fitted in the same fashion according to Eq. (8) for different systems, so the learning objective as in Eq. (8) is transferable. But since we have only limited amount of training samples, a model trained according to an improper objective is likely to fall victim to overfitting due to low sample efficiency. Overfitting makes the model performing well on a specific system to be possibly completely wrong for another. As explained earlier, directly fitting of the overall energy may not be a good empirical risk in terms of transferability. An objective suitable for transferable learning should encourage the model to discover common patterns or rules across different systems and make predictions mainly on them.

Indeed, incorporation of some recurring patterns and rules has led to the success of classical force fields.<sup>97</sup> In the force fields, interactions are described by separate but fixed functional forms. Besides, force fields usually include terms responsible for the correct asymptotic behavior. For example, Coulomb interactions are generally present in order to describe the long-range interactions, whereas van der Waals-like terms are included to account for the Pauli exclusion. Therefore, force fields successfully capture the common patterns and rules for various molecular systems, thus exhibiting flexible transferability. Besides, the number of parameters of force fields are usually limited, and such models are not prone to overfitting. Inspired by these observations, one may introduce deep molecular models to extend the scope of known force fields. As an example, PhysNet predicts the atomic partial charges based on nuclei positions during training, which can be regarded as “re-parametrizing” the static atomic partial charges with floating ones. Instead of directly predicting the overall energy, the re-parametrized models can alternatively predict the correct parameters for specific interaction functions (like the charges in Coulomb’s law), thus are possibly more transferable and sample-efficient. We think that re-parametrization and development of force-field-like deep PES models (like PhysNet and TensorMol) might be a promising direction for future research.

### 2. Learn more, with less

The sample efficiency is another critical issue faced by existing deep molecular models, particularly in the realm of supervised learning like fitting deep PES. Usually fitting a deep PES for a condensed-phase molecular system requires a huge amount of training data, that is, samples of the single-point energy from a high-dimensional configuration space. Since

computing the single-point energy is generally expensive, one thus desires to train a reliable deep PES model with as few samples as possible. However, it is non-trivial for current deep molecular models to achieve this goal due to the following reasons:

(1) The deep molecular models are usually heavily over-parametrized. Mathematically, the parameter space of a deep ANN is usually too vast, and the capacity of the resulting model is excessive.<sup>45</sup> As a result, to find an optimal model with good generalization ability for a specific regression task requires a lot of instances.

(2) The current deep molecular models are not readily transferable. As explained above, one cannot straightforwardly make use of existing or pre-trained models due to the non-transferability issue. Every time one hopes to fit a new deep PES, one needs to perform everything from scratch, and cannot exploit others’ previous efforts and results.

In deep learning community, there are two useful approaches to increase the sample efficiency for the models. The first one is pre-training plus fine-tuning, also known as transfer learning.<sup>98</sup> Specifically, one can pre-train the model over several related tasks, thus obtaining a model with near-optimal values for most parameters. Then for a specific new task, a satisfying model can be obtained by merely fine-tuning a small number of parameters (usually corresponding to the last several layers next to the output layer). Mathematically speaking, transfer learning effectively reduces the number of parameters which needs optimization for a specific task, thus increases the sample efficiency of the model meanwhile preserving the overall expressivity. This strategy has been shown extremely successful in computer vision<sup>99</sup> and natural language modeling.<sup>100</sup> Clearly, if one wishes to implement transfer learning for molecular modeling, one has to select transferable models and suitable transferable learning objectives which can be shared by various molecular systems.

The other promising approach is meta-learning, also known as learning to learn.<sup>101-103</sup> Similar to transfer learning, in meta-learning, the model is also first trained upon a number of related tasks. Different from transfer learning which optimizes the model with a normal supervised learning objective, meta-learning generally optimizes the model w.r.t. its ability of generalization.<sup>67, 104-105</sup> In other words, meta-learning aims to train a model so that it can fast adapt to new samples to fulfil a new task within only a few optimization steps. Transfer learning can be viewed as a special type of meta-learning. Compared to transfer learning, meta-learning is particularly useful in one- or few-shot learning, which means that the model can make reasonable predictions in a new task given only one or a few instances. Both transfer learning and meta-learning could be promising in improving the sample efficiency of deep molecular models.

### 3. Handle uncertainty

Many critics over deep molecular models arise from the fact that assessment or validation of the models is very tricky. For example, we usually do not know and cannot tell when the model will perform well and when it will fail. Taking deep PES models as example, even if we get a “perfect” model which zeroes the error for the training data, we are still unsure about its performance in those regions uncovered by the training samples. This difficulty is caused by the deterministic learning mechanism adopted by current deep molecular models. Specifically, given an input  $\mathbf{x}$ , the deterministic learning only predicts the value of the outcome  $f_{\theta}(\mathbf{x})$ , but leaves no information about the uncertainty of the prediction.

The lack of uncertainty assessment also calls trouble for the preparation and curation of the training data. For example, if we knew the prediction uncertainty, we would naturally prepare more training data over the regions with higher uncertainty. This procedure is known as active learning for data augmentation in machine learning,<sup>106-107</sup> because the data preparation is achieved

through the user’s active interactions with the model. Unfortunately, almost no existing deep molecular models provide the uncertainty information.

Currently, some researchers ameliorated this issue by training an ensemble of deep molecular models,<sup>37, 61</sup> so the uncertainty of predictions can be approximated by the divergence between the predictions of these models. However, this is not a theoretically justified approach, and a reasonable estimation of the uncertainty usually entails a large number of independent models. Considering that even a single deep molecular model would be computationally expensive, training an ensemble of models is often prohibitive in practice.

On the other hand, incorporation of uncertainty in statistical models is formally considered by Bayesian learning.<sup>108</sup> Particularly, Bayesian learning has given rise to many fruits for regression tasks which are ubiquitous in molecular modeling. For example, Gaussian Processes (GP) is famous for its ability to handle uncertainty in regression problems, and is widely used for functional regression or Bayesian optimization.<sup>109</sup> However, the vanilla GP is a non-parametric method so it cannot directly scale to high-dimensional input data. Nonetheless, there are increasing attempts to cross-fertilize Bayesian learning with deep learning.<sup>110-111</sup> We thus note here that augmenting the current deterministic learning mechanism with Bayesian learning may be a promising and reasonable direction for further investigation in order to handle the uncertainty of deep molecular models.

#### 4. Incorporate experiments

Most of the existing applications of deep molecular modeling were dealing with data from simulations rather than experiments. However, we note there that deep molecular modeling can find wide applications in connection with experiments. In related studies, incorporating experimental information into molecular modeling and simulations is largely based on maximum entropy (MaxEnt) principle.<sup>112-113</sup> Although having been proved useful in many cases,<sup>114-116</sup> MaxEnt suffers some limitations. For example, MaxEnt may efficiently process low-dimensional restraints but becomes less feasible for inter-dependent high-dimensional ones. Furthermore, many single-molecule experiments provide information about the distribution of certain observables. However, MaxEnt can only incorporate the expectation value as restraints but wastes the information about the higher-order moments.

As an alternative approach, deep molecular modeling with EBM treats the experimental observables as evidence so that one can implement the maximum likelihood estimation via Bayesian inference. For instance, incorporating experimental observations in EBM is straightforward by virtue of the compositionality of the energy function,

$$F_{\theta}(\mathbf{x}) = F_{\text{exp}}(\mathbf{q}_{\text{exp}}(\mathbf{x})) + F_{\text{phys}}(\mathbf{x}) \quad (20)$$

where  $F_{\text{exp}}$  corresponds to the experimentally measured quantities  $\mathbf{q}_{\text{exp}}(\mathbf{x})$ , whereas  $F_{\text{phys}}$  accounts for additional physics restraints (e.g., force-field-like terms). Training of such an EBM will yield an energy function  $F_{\theta}(\mathbf{x})$  whose marginal distribution over the experiment-related variables  $\mathbf{q}_{\text{exp}}(\mathbf{x})$  is consistent with observations.<sup>44</sup> Specifically, both TALOS and VIFE represent available algorithms that can potentially achieve this goal. We thus expect deep molecular modeling with EBM to find more applications in conjunction with experiments, especially single-molecule ones.

#### 5. End notes

In addition to deep learning, non-parametric learning has also progressed significantly in the field of molecular modelling.<sup>26, 117-118</sup> Many of the above-mentioned limitations of deep molecular models can be readily circumvented or overcome by these non-parametric models. Therefore, the cross-fertilization between deep learning and non-parametric methods may lead to more reliable and efficient molecular models. Last but not the least, in order to democratize deep learning for molecular modeling and simulations, it is necessary to develop special-purpose hardware and software suitable for fast and user-friendly computation. For example, molecular simulation community will definitely benefit from the auto-differentiation and parallel computation which are the bedrocks of modern large-scale deep learning. Besides, as the machine learners and molecular simulation practitioners usually work in different platforms, a proper linker or interface which could unify the molecular modeling software and deep learning software is highly desired. Indeed, researchers from both scientific and business communities like Google are making efforts to this end.<sup>119-120</sup> With the improvement of the infrastructure, deep learning is expected to bring larger impact and more opportunities to molecular modeling and simulations in the near future.



## Acknowledgements

The authors thank Weinan E, Wenjun Xie, Xing Che and Cheng-Wen Liu for useful discussion. This research was supported by National Natural Science Foundation of China [21927901, 21821004, 21873007 to Y.Q.G.], the National Key Research and Development Program of China [2017YFA0204702 to Y.Q.G.] and Guangdong Basic and Applied Basic Research Foundation [2019A1515110278 to Y.I.Y.]. The author J.Z. thanks the Alexander von Humboldt Foundation for supporting part of this research.

**Notes:** The authors declare no conflict of interest.

## REFERENCES

1. Mueller, T.; Hernandez, A.; Wang, C., Machine learning for interatomic potential models. *journal of chemical physics* **2020**, *152* (5).
2. Zhang, L.; Han, J.; Wang, H.; Car, R.; Weinan, E., Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters* **2018**, *120* (14), 143001.
3. Zhang, J.; Lei, Y.; Yang, Y. I.; Gao, Y., Reinforcement Learning for Multi-Scale Molecular Modeling. *ChemRxiv. Preprint*. **2019**, doi: 10.26434/chemrxiv.9640814.v2.
4. Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; Fabritiis, G. d.; Noé F.; Clementi, C., Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *acs central science* **2019**, *5* (5), 755-767.
5. Bonati, L.; Zhang, Y.-Y.; Parrinello, M., Neural networks based variationally enhanced sampling. *arXiv preprint arXiv:1904.01305* **2019**.
6. Zhang, J.; Yang, Y. I.; Noé F., Targeted Adversarial Learning Optimized Sampling. *journal of physical chemistry letters* **2019**, *10* (19), 5791-5797.
7. Mardt, A.; Pasquali, L.; Wu, H.; Noé F., VAMPnets for deep learning of molecular kinetics. *nature communications* **2018**, *9* (1).
8. Zhang, J.; Lei, Y.-K.; Che, X.; Zhang, Z.; Yang, Y. I.; Gao, Y. Q., Deep Representation Learning for Complex Free-Energy Landscapes. *journal of physical chemistry letters* **2019**, *10* (18), 5571-5576.
9. Rumelhart, D. E.; McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. 1986.
10. LeCun, Y.; Bengio, Y.; Hinton, G., Deep learning. *nature* **2015**, *521* (7553), 436-444.
11. Schmidhuber, J., Deep learning in neural networks. *neural networks* **2015**, *61*, 85-117.
12. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J., Learning representations by back-propagating errors. *nature* **1988**, *323* (6088), 696-699.
13. Hornik, K.; Stinchcombe, M.; White, H., Multilayer feedforward networks are universal approximators. *neural networks* **1989**, *2* (5), 359-366.
14. LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D., Backpropagation applied to handwritten zip code recognition. *neural computation* **1989**, *1* (4), 541-551.
15. Hochreiter, S.; Schmidhuber, J., Long short-term memory. *neural computation* **1997**, *9* (8), 1735-1780.
16. Bahdanau, D.; Cho, K.; Bengio, Y., Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, 2015.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I., Attention is All you Need. In *Neural Information Processing Systems*, 2017; pp 5998-6008.
18. He, K.; Zhang, X.; Ren, S.; Sun, J., Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition*, 2016; pp 770-778.
19. Goodfellow, I.; Bengio, Y.; Courville, A., *Deep learning*. MIT press: 2016.
20. Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V. F.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; Gülçehre, Ç.; Song, H. F.; Ballard, A. J.; Gilmer, J.; Dahl, G. E.; Vaswani, A.; Allen, K. R.; Nash, C.; Langston, V.; Dyer, C.; Heess, N.; Wierstra, D.; Kohli, P.; Botvinick, M.; Vinyals, O.; Li, Y.; Pascanu, R., Relational inductive biases, deep learning, and graph networks. 2018.
21. Behler, J.; Parrinello, M., Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *physical review letters* **2007**, *98* (14).
22. Smith, J. S.; Isayev, O.; Roitberg, A. E., ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *chemical science* **2017**, *8* (4), 3192-3203.
23. Yao, K.; Herr, J. E.; Toth, D. W.; McKintyre, R.; Parkhill, J., The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *chemical science* **2018**, *9* (8), 2261-2269.
24. Bartok, A. P.; Payne, M. C.; Kondor, R.; Csanyi, G., Gaussian Approximation Potentials: The Accuracy of

- Quantum Mechanics, without the Electrons. *Physical Review Letters* **2010**, *104* (13), 136403.
25. Rupp, M.; Tkatchenko, A.; Müller, K.; Von Lilienfeld, O. A., Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical Review Letters* **2012**, *108* (5), 058301.
  26. Bartók, A. P.; Kondor, R.; Csányi, G., On representing chemical environments. *physical review b* **2013**, *87* (18).
  27. Sheng, H. W.; Kramer, M. J.; Cadien, A.; Fujita, T.; Chen, M. W., Highly optimized embedded-atom-method potentials for fourteen fcc metals. *physical review b* **2011**, *83* (13).
  28. Behler, J., Perspective: Machine learning potentials for atomistic simulations. *journal of chemical physics* **2016**, *145* (17).
  29. Piaggi, P. M.; Parrinello, M., Entropy based fingerprint for local crystalline order. *journal of chemical physics* **2017**, *147* (11).
  30. Niu, H.; Piaggi, P. M.; Invernizzi, M.; Parrinello, M., Molecular dynamics simulations of liquid silica crystallization. *proceedings of the national academy of sciences of the united states of america* **2018**, *115* (21), 5348-5352.
  31. Battaglia, P.; Pascanu, R.; Lai, M.; Rezende, D. J.; Kavukcuoglu, K., Interaction networks for learning about objects, relations and physics. In *Neural Information Processing Systems*, 2016; pp 4509-4517.
  32. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y., Spectral Networks and Locally Connected Networks on Graphs. In *International Conference on Learning Representations*, 2014.
  33. Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P., Convolutional networks on graphs for learning molecular fingerprints. In *Neural Information Processing Systems*, 2015; pp 2224-2232.
  34. Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E., Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, 2017; pp 1263-1272.
  35. Schütt, K. T.; Sauceda, H. E.; Kindermans, P. J.; Tkatchenko, A.; Müller, K. R., SchNet – A deep learning architecture for molecules and materials. *journal of chemical physics* **2018**, *148* (24).
  36. Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A., Quantum-chemical insights from deep tensor neural networks. *nature communications* **2017**, *8* (1), 13890-13890.
  37. Unke, O. T.; Meuwly, M., PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *journal of chemical theory and computation* **2019**, *15* (6), 3678-3693.
  38. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A., Self-Attention Generative Adversarial Networks. In *International Conference on Machine Learning*, 2019; pp 7354-7363.
  39. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y., Graph Attention Networks. In *International Conference on Learning Representations*, 2018.
  40. Arjovsky, M.; Chintala, S.; Bottou, L. In *Wasserstein generative adversarial networks*, International conference on machine learning, 2017; pp 214-223.
  41. Jia, K.; Tao, D.; Gao, S.; Xu, X., Improving Training of Deep Neural Networks via Singular Value Bounding. In *Computer Vision and Pattern Recognition*, 2017; pp 3994-4002.
  42. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y., Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*, 2018.
  43. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. In *Generative adversarial nets*, Advances in neural information processing systems, 2014; pp 2672-2680.
  44. Du, Y.; Mordatch, I., Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689* **2019**.
  45. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O., Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.
  46. Ruder, S., An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* **2017**.
  47. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; Freitas, N. d., Learning to learn by gradient descent by gradient descent. In *Neural Information Processing Systems*, 2016; pp 3981-3989.
  48. Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Židek, A.; Nelson, A. W. R.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Kohli, P.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D., Improved protein structure prediction using potentials from deep learning. *nature* **2020**, *577* (7792), 706-710.
  49. Dugas, C.; Bengio, Y.; Bélisle, F.; Nadeau, C.; Garcia, R., Incorporating Second-Order Functional Knowledge for Better Option Pricing. In *Neural Information Processing Systems*, 2000; pp 472-478.
  50. Schütt, K. T.; Kindermans, P. J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K. R., SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Neural Information Processing Systems*, 2017; pp 992-1002.
  51. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S., Fast and Accurate Deep Network Learning by Exponential

- Linear Units (ELUs). In *International Conference on Learning Representations*, 2016.
52. Ramachandran, P.; Zoph, B.; Le, Q. V., Swish: a Self-Gated Activation Function. 2017.
  53. Rogal, J.; Schneider, E.; Tuckerman, M. E., Neural network based path collective variables for enhanced sampling of phase transformations. 2019.
  54. Sultan, M. M.; Pande, V. S., Automated design of collective variables using supervised machine learning. *journal of chemical physics* **2018**, *149* (9).
  55. Tuckerman, M. E., Ab initio molecular dynamics: basic concepts, current trends and novel applications. *Journal of Physics: Condensed Matter* **2002**, *14* (50), R1297.
  56. Klicpera, J.; Groß, J.; Günnemann, S., Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, 2020.
  57. Shao, Y.; Hellström, M.; Mitev, P. D.; Knijff, L.; Zhang, C., PiNN: A Python Library for Building Atomic Neural Networks of Molecules and Materials. *journal of chemical information and modeling* **2020**, *60* (3), 1184-1193.
  58. Voth, G. A., *Coarse-graining of condensed phase and biomolecular systems*. CRC press: 2008.
  59. Noid, W.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C., The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *The Journal of chemical physics* **2008**, *128* (24), 244114.
  60. Schneider, E.; Dai, L.; Topper, R. Q.; Drechsel-Grau, C.; Tuckerman, M. E., Stochastic neural network approach for learning high-dimensional free energy surfaces. *Physical review letters* **2017**, *119* (15), 150601.
  61. Zhang, L.; Wang, H.; E, W., Reinforced dynamics for enhanced sampling in large atomic and molecular systems. *The Journal of chemical physics* **2018**, *148* (12), 124113.
  62. Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C., Machine learning of coarse-grained molecular dynamics force fields. *ACS central science* **2019**.
  63. Hyvärinen, A., Estimation of Non-Normalized Statistical Models by Score Matching. *journal of machine learning research* **2005**, *6*, 695-709.
  64. Amadei, A.; Linssen, A. B.; Berendsen, H. J., Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics* **1993**, *17* (4), 412-425.
  65. Becker, S.; Hinton, G. E., Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* **1992**, *355* (6356), 161.
  66. Ji, X.; Henriques, J. F.; Vedaldi, A., Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653* **2018**.
  67. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. In *Matching networks for one shot learning*, Advances in neural information processing systems, 2016; pp 3630-3638.
  68. Vapnik, V. N., An overview of statistical learning theory. *IEEE transactions on neural networks* **1999**, *10* (5), 988-999.
  69. Silverman, B. W., *Density estimation for statistics and data analysis*. Routledge: 2018.
  70. Kingma, D. P.; Welling, M., Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**.
  71. Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A. In *Conditional image generation with pixelcnn decoders*, Advances in neural information processing systems, 2016; pp 4790-4798.
  72. Dinh, L.; Krueger, D.; Bengio, Y., Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* **2014**.
  73. Abbasnejad, M. E.; Shi, Q.; Hengel, A. v. d.; Liu, L. In *A Generative Adversarial Density Estimator*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019/june.
  74. Salakhutdinov, R.; Mnih, A.; Hinton, G. In *Restricted Boltzmann machines for collaborative filtering*, Proceedings of the 24th international conference on Machine learning, ACM: 2007; pp 791-798.
  75. Salakhutdinov, R.; Hinton, G., Deep Boltzmann Machines. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, David van, D.; Max, W., Eds. PMLR: Proceedings of Machine Learning Research, 2009; Vol. 5, pp 448--455.
  76. Hopfield, J. J., Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* **1982**, *79* (8), 2554-2558.
  77. Wales, D., *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press: 2003.
  78. Shell, M. S., The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *The Journal of chemical physics* **2008**, *129* (14), 144108.
  79. Chaimovich, A.; Shell, M. S., Coarse-graining errors and numerical optimization using a relative entropy framework. *The Journal of chemical physics* **2011**, *134* (9), 094112.
  80. Valsson, O.; Parrinello, M., Variational approach to enhanced sampling and free energy calculations. *Physical review letters* **2014**, *113* (9), 090601.
  81. Ingraham, J. B.; Riesselman, A. J.; Sander, C.; Marks, D., Learning Protein Structure with a Differentiable Simulator. In *International Conference on Learning Representations*, 2019.
  82. Belanger, D.; McCallum, A., Structured prediction energy networks. In *International Conference on Machine Learning*, 2016; pp 983-992.
  83. Gygli, M.; Norouzi, M.; Angelova, A., Deep Value Networks Learn to Evaluate and Iteratively Refine

- Structured Outputs. In *International Conference on Machine Learning*, 2017; pp 1341-1351.
84. Sutton, R. S.; Barto, A. G., *Reinforcement learning: An introduction*. MIT press: 2018.
  85. Williams, R. J., Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* **1992**, 8 (3-4), 229-256.
  86. Dayan, P.; Hinton, G. E., Using expectation-maximization for reinforcement learning. *neural computation* **1997**, 9 (2), 271-278.
  87. Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P., Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *journal of chemical physics* **2018**, 149 (7), 72301-72301.
  88. Pathak, D.; Agrawal, P.; Efros, A. A.; Darrell, T., Curiosity-driven Exploration by Self-supervised Prediction. In *International Conference on Machine Learning*, 2017; pp 2778-2787.
  89. Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. 1/1/2016, 2016.
  90. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S., Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, 2018; pp 1856-1865.
  91. Bryngelson, J. D.; Onuchic, J. N.; Socci, N. D.; Wolynes, P. G., Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics* **1995**, 21 (3), 167-195.
  92. Andoni, A.; Indyk, P.; Krauthgamer, R. In *Earth mover distance over high-dimensional spaces*, Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics: 2008; pp 343-352.
  93. Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. In *Asynchronous methods for deep reinforcement learning*, International conference on machine learning, 2016; pp 1928-1937.
  94. Grondman, I.; Busoniu, L.; Lopes, G. A.; Babuska, R., A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2012**, 42 (6), 1291-1307.
  95. Barducci, A.; Bussi, G.; Parrinello, M., Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical review letters* **2008**, 100 (2), 020603.
  96. Oord, A. v. d.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; kavukcuoglu, k.; Driessche, G. v. d.; Lockhart, E.; Cobo, L.; Stimberg, F.; Casagrande, N.; Grewe, D.; Noury, S.; Dieleman, S.; Elsen, E.; Kalchbrenner, N.; Zen, H.; Graves, A.; King, H.; Walters, T.; Belov, D.; Hassabis, D., Parallel WaveNet: Fast High-Fidelity Speech Synthesis. In *International Conference on Machine Learning*, 2018; pp 3915-3923.
  97. Frenkel, D.; Smit, B., *Understanding molecular simulation: from algorithms to applications*. Elsevier: 2001; Vol. 1.
  98. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C., A Survey on Deep Transfer Learning. In *International Conference on Artificial Neural Networks*, 2018; pp 270-279.
  99. Simonyan, K.; Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.
  100. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019; pp 4171-4186.
  101. Thrun, S.; Pratt, L., Learning to learn: introduction and overview. 1998; pp 3-17.
  102. Hochreiter, S.; Younger, A. S.; Conwell, P. R., Learning to Learn Using Gradient Descent. In *International Conference on Artificial Neural Networks*, 2001; pp 87-94.
  103. Lake, B. M.; Salakhutdinov, R.; Tenenbaum, J. B., Human-level concept learning through probabilistic program induction. *science* **2015**, 350 (6266), 1332-1338.
  104. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T., Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016; pp 1842-1850.
  105. Finn, C.; Abbeel, P.; Levine, S., Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017; pp 1126-1135.
  106. Wang, D.; Shang, Y., A new active labeling method for deep learning. In *International Joint Conference on Neural Network*, 2014; pp 112-119.
  107. Gal, Y.; Islam, R.; Ghahramani, Z., Deep Bayesian Active Learning with Image Data. In *International Conference on Machine Learning*, 2017; pp 1183-1192.
  108. Ghahramani, Z., Probabilistic machine learning and artificial intelligence. *nature* **2015**, 521 (7553), 452-459.
  109. Rasmussen, C. E.; Williams, C. K. I., *Gaussian Processes for Machine Learning*. 2005.
  110. Hinton, G.; Neal, R. M., *Bayesian learning for neural networks*. 1995.
  111. Hernandez-Lobato, J. M.; Adams, R., Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *International Conference on Machine Learning*, 2015; pp 1861-1869.
  112. Jaynes, E. T., Maximum Entropy and Bayesian Methods in Applied Statistics: Bayesian Methods: General Background. 1986.
  113. Press, S.; Ghosh, K.; Lee, J.; Dill, K. A., Principles of maximum entropy and maximum caliber in statistical

- physics. *reviews of modern physics* **2013**, 85 (3), 1115-1141.
114. Pitera, J. W.; Chodera, J. D., On the Use of Experimental Observations to Bias Simulated Ensembles. *Journal of Chemical Theory and Computation* **2012**, 8 (10), 3445-3451.
  115. Dannenhofferl, T.; White, A. D.; Voth, G. A., A Direct Method for Incorporating Experimental Data into Multiscale Coarse-Grained Models. *Journal of Chemical Theory and Computation* **2016**, 12 (5), 2144-2153.
  116. Cesari, A.; Reißer, S.; Bussi, G., Using the Maximum Entropy Principle to Combine Simulations and Solution Experiments. *Computation* **2018**, 6 (1), 15.
  117. Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M., Machine learning unifies the modeling of materials and molecules. *science advances* **2017**, 3 (12).
  118. Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R., Machine Learning of Accurate Energy-Conserving Molecular Force Fields. *science advances* **2017**, 3 (5).
  119. Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K., SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *journal of chemical theory and computation* **2019**, 15 (1), 448-455.
  120. Schoenholz, S. S.; Cubuk, E. D., JAX, M.D.: End-to-End Differentiable, Hardware Accelerated, Molecular Dynamics in Pure Python. *bulletin of the american physical society* **2019**.
  121. Fink, T.; Raymond, J.-L., Virtual Exploration of the Chemical Universe up to 11 Atoms of C, N, O, F: Assembly of 26.4 Million Structures (110.9 Million Stereoisomers) and Analysis for New Ring Systems, Stereochemistry, Physicochemical Properties, Compound Classes, and Drug Discovery. *Journal of Chemical Information and Modeling* **2007**, 47 (2), 342-353.
  122. Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A., Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014**, 1 (1), 140022-140022.

## Tables and Figures

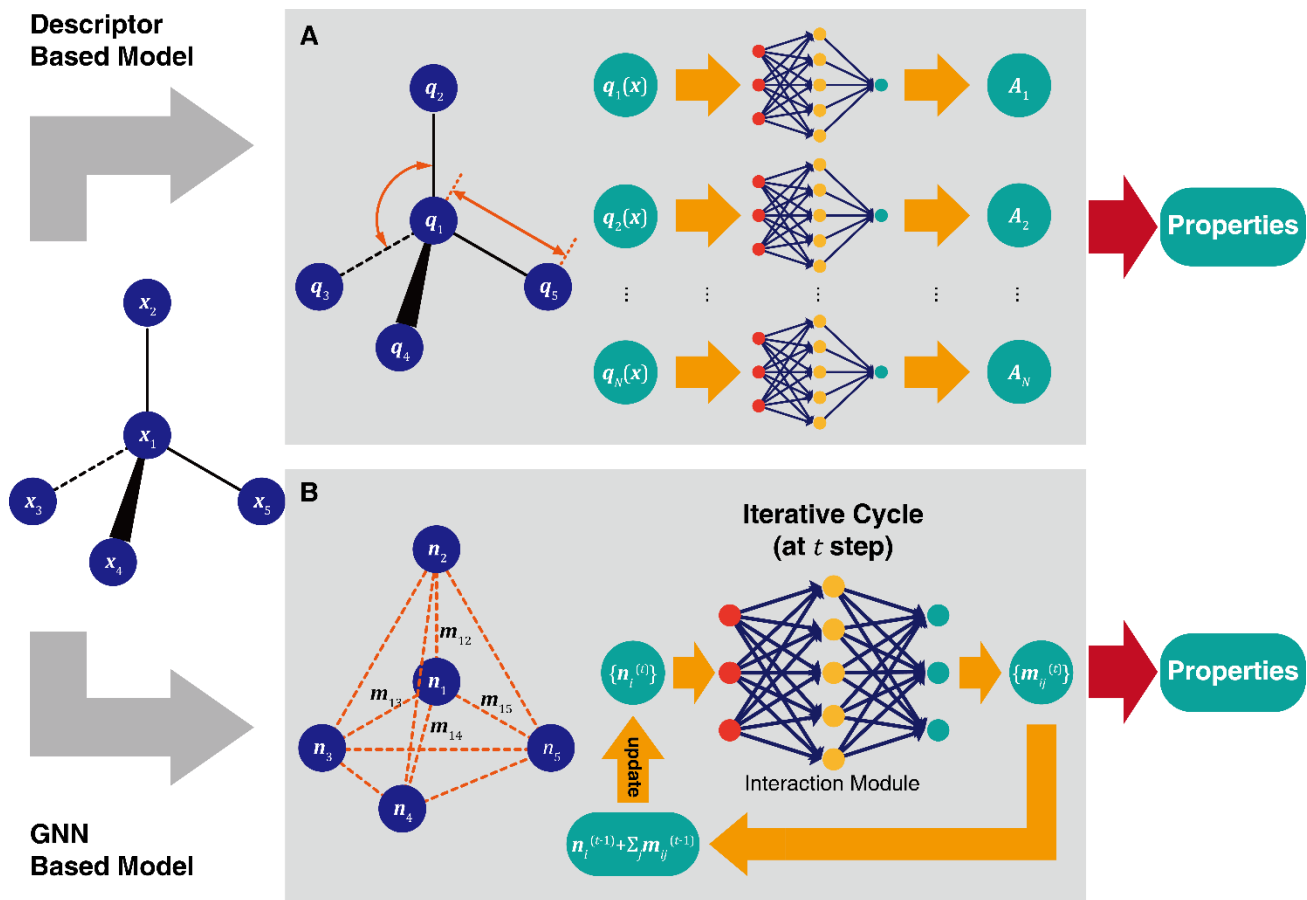
**Table 1.** Survey of deep molecular models

Model	Type <sup>a</sup>	Birth Year <sup>b</sup>	Dataset <sup>c</sup>
BPNN <sup>21</sup>	Descriptor	2007	Silicon <sup>21</sup>
ANI <sup>22</sup>	Descriptor	2017	GDB <sup>121</sup>
MPNN <sup>34</sup>	GNN	2017	QM9 <sup>122</sup>
DTNN <sup>36</sup>	GNN	2017	QM9
TensorMol <sup>23</sup>	Descriptor	2018	Chemspider <sup>23</sup>
SchNet <sup>35</sup>	GNN	2018	MD17 <sup>118</sup> , QM9
DPMD <sup>2</sup>	Descriptor	2018	MD17
PhysNet <sup>23, 37</sup>	GNN	2019	MD17, QM9

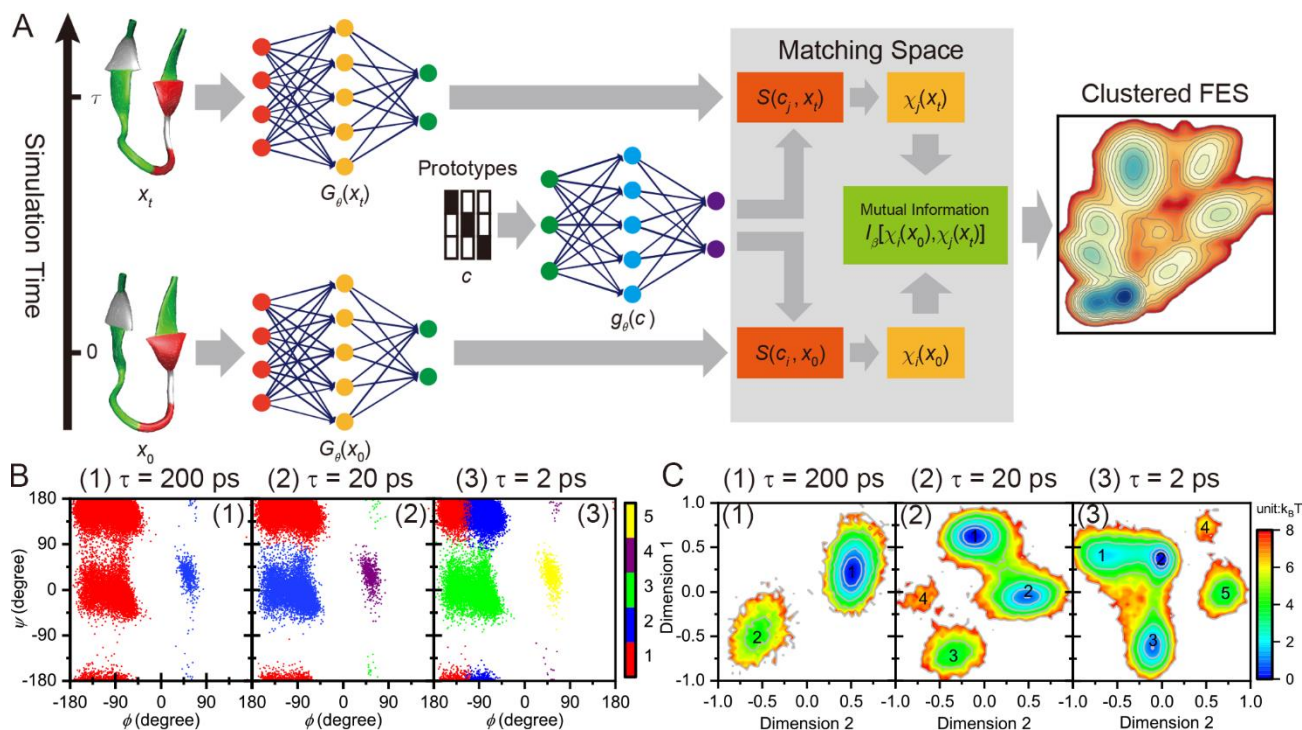
a. Model types are either descriptor-based or GNN-based.

b. The year when the model was first proposed.

c. Dataset used for benchmarking the models.

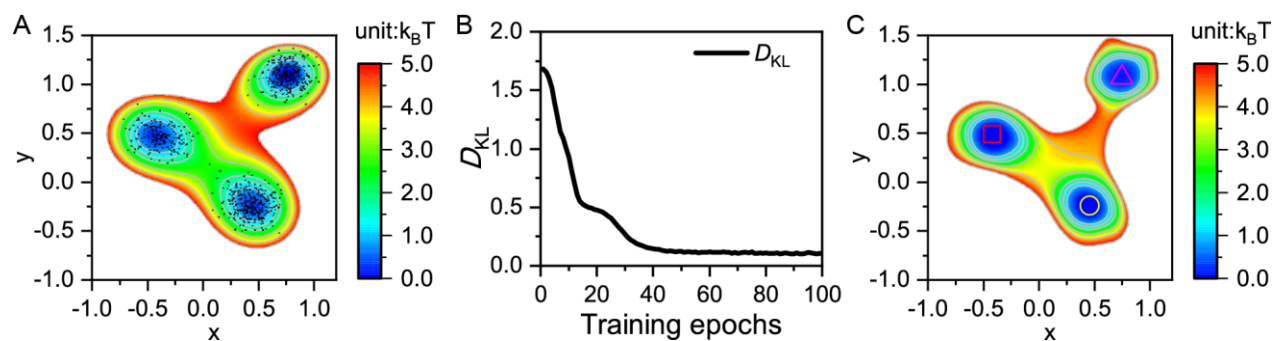


**Figure 1.** Architectures of deep molecular models. (A) Descriptor-based models (Eqs. (1-2)).  $\mathbf{q}(\mathbf{x}_i)$  corresponds to the descriptors to the  $i$ -th particle (atom), which is further fed into a neural network. The output of each atomic neural network is then summed up to predict molecular properties. (B) GNN-based models (Eqs. (3-7)). Each particle (atom) is represented by a node vector ( $\mathbf{n}_i$ ). An interaction module collects messages ( $\mathbf{m}_{ij}$ ) between nodes which are used to update the node vector. After several iterations of message passing, the node vectors are used to predict the molecular properties.

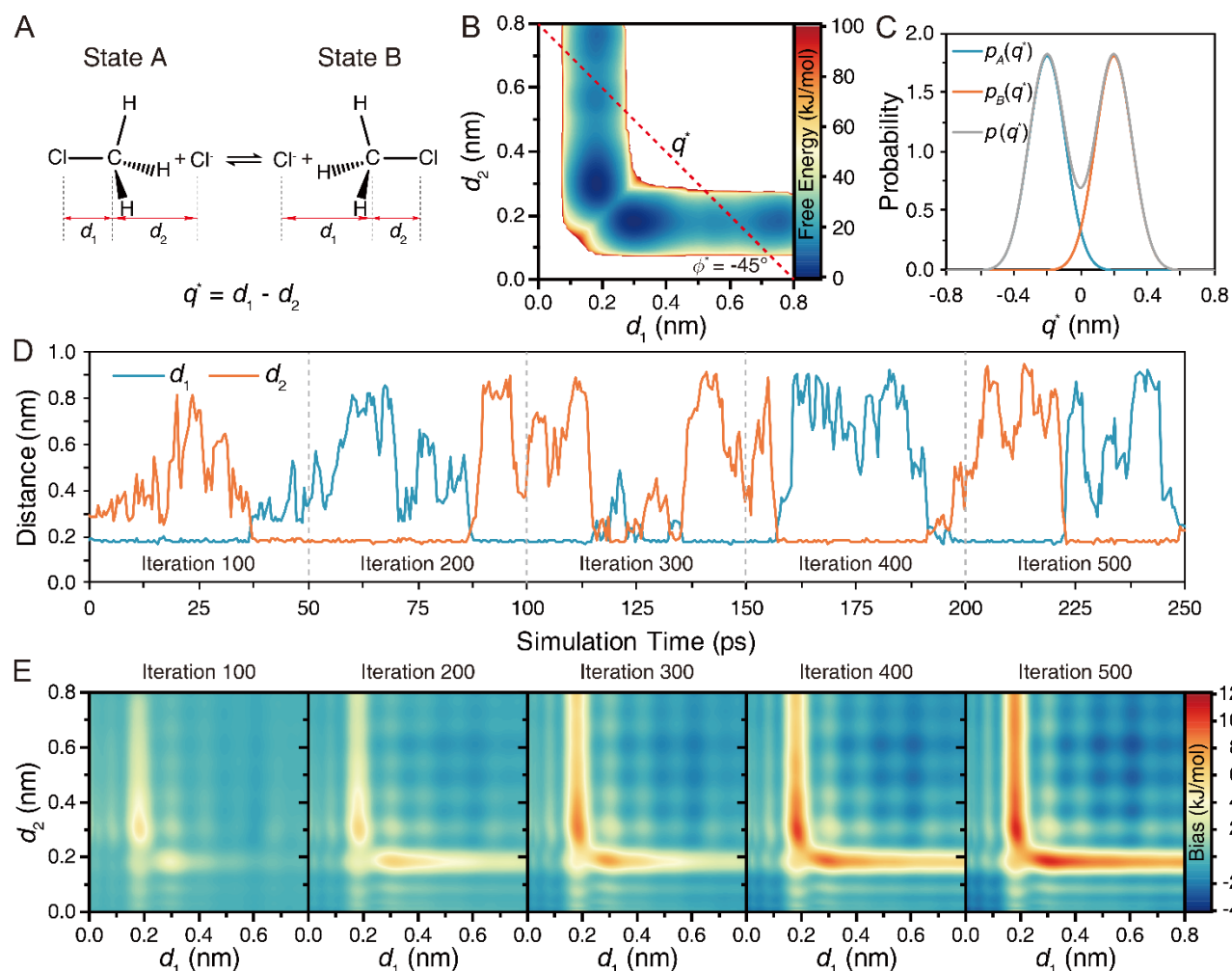


**Figure 2.** Information distilling of metastability (IDM). (A) Illustrative working flow of IDM. High-dimensional input vectors ( $\mathbf{x}$ ) and preimages of cluster centroids ( $\mathbf{c}$ ) are projected into images in the matching space by  $G_\theta$  and  $g_\theta$ , respectively. Similarity score between  $\mathbf{x}$  and  $\mathbf{c}$  calculated in the matching space,  $S(\mathbf{x}, \mathbf{c})$ , leads to output  $\chi(\mathbf{x})$  which is further used for Mutual Information ( $I_\beta$ ) maximization. The matching space serves as the reduced embedding for  $\mathbf{x}$ , and  $\chi(\mathbf{x})$  as the soft clustering labels. (B) IDM clustering results for Ala2 over decreasing timescales. From left to right: (1)  $\tau = 200$  ps, 2 clusters identified; (2)  $\tau = 20$  ps, 4 clusters identified; (3)  $\tau = 2$  ps, 5 clusters identified. Cluster identity is marked by different colors (with the color bar on the right). (C) IDM dimension reduction results for Ala2 over decreasing timescales (in the same order as in panel B). The PMF for each reduced representation is shown, and centers of the metastable states are indexed by numbers. Figures were reproduced with permission from Zhang et al., *J. Phys. Chem. Lett.* **2019**, *10* (18), 5571-5576. Copyright 2019 American Chemical Society.

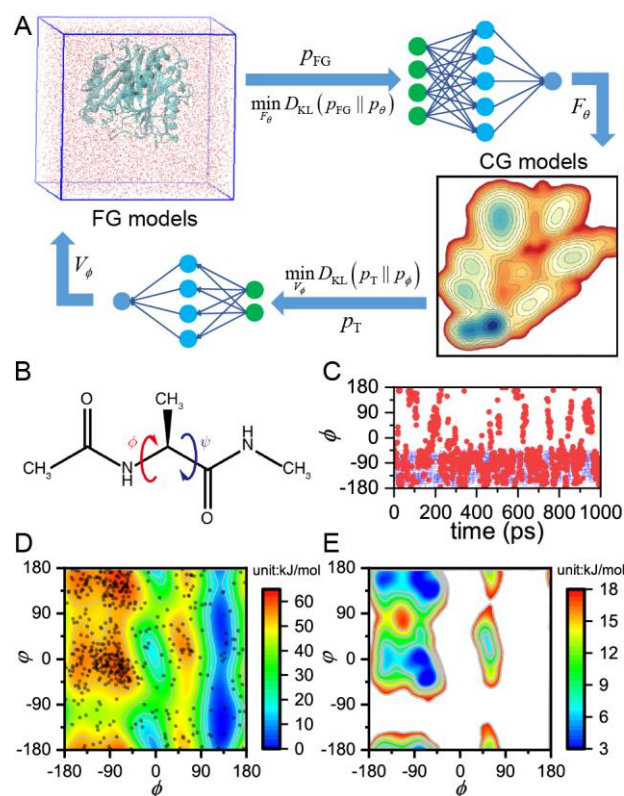




**Figure 3.** Application of variational inference of free energy (VIFE) on a 2D numerical potential. (A) 2D potential energy surface (PES),  $U(x, y)$  of the model. Black dots correspond to representative samples randomly drawn from  $p_{FG}$ . (B) Evolution of the KL-divergence  $D_{KL}(p_{FG} || p_{\theta})$  against training epochs in VIFE. (C) The contour map corresponds to the optimized  $F_{\theta}$  by VIFE, the three symbols (magenta triangle, red square and yellow circle) represent three free-energy minima found in  $F_{\theta}$ . Figures were reproduced with permission from Zhang et al., “Reinforcement Learning for Multi-Scale Molecular Modeling”, e-print ChemRxiv: 9640814 (2019). Copyright 2019 Author(s), licensed under a CC BY-NC-ND 4.0 License.



**Figure 4.** Targeted adversarial learning optimized sampling (TALOS) over a chemical reaction. (A) The reaction under study:  $\text{Cl}^- + \text{CH}_3\text{Cl} \leftrightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$ . Two C-Cl bond distances ( $d_1, d_2$ ) are chosen as CVs, over which the bias potential is constructed; whereas a linear descriptor  $q^* = d_1 - d_2$  is used to define the target distribution. (B) 2D FES over ( $d_1, d_2$ ) space and the illustration of  $q^*$ . (C) The distribution of  $q^*$  for the reactant (blue), product (orange) and the interpolated target distribution (grey). (D) Examples of the biased MD trajectories projected on  $d_1$  (blue) and  $d_2$  (orange) at different TALOS training iterations. (E) The optimized 2D bias potential at different TALOS training iterations. Figures were reproduced with permission from Zhang et al., *J. Phys. Chem. Lett.* **2019**, 10 (19), 5791-5797. Copyright 2019 American Chemical Society.



**Figure 5.** Reinforced variational inference of free energy (RE-VIFE). (A) Illustration of RE-VIFE. Given simulation samples from fine-grained (FG) models ( $p_{\text{FG}}$ ), a coarse-grained (CG) potential  $F_\theta$  can be variationally approximated. In turn, a target distribution  $p_T$  can be defined based on CG simulations running over  $F_\theta$ , according to which a bias potential  $V_\phi$  can be variationally optimized to boost the FG simulation. (B-E) RE-VIFE sampling of Ala2 in explicit water: (B) Structure of Ala2 and two CG variables are the torsions  $\phi$  and  $\varphi$ . (C) 1-ns simulation trajectories projected on the slowly-changing torsion  $\phi$ . Blue squares correspond to vanilla MD, red dots to MD biased by  $V_\phi$ . (D) The contour map of  $V_\phi$  optimized via RE-VIFE. Superimposed black dots are representative samples produced by the enhanced MD simulation under  $V_\phi$ . (E) Contour map of  $F_\theta$  optimized via RE-VIFE. Figures were reproduced with permission from Zhang et al., “Reinforcement Learning for Multi-Scale Molecular Modeling”, e-print ChemRxiv: 9640814 (2019). Copyright 2019 Author(s), licensed under a CC BY-NC-ND 4.0 License.