Integrating Machine Learning with Physics-Based Modeling

Weinan E[1,2], Jiequn Han[1], and Linfeng Zhang[2]

[1]Department of Mathematics, Princeton University
[2]Program in Applied and Computational Mathematics, Princeton University

Abstract

Machine learning is poised as a very powerful tool that can drastically improve our ability to carry out scientific research. However, many issues need to be addressed before this becomes a reality. This article focuses on one particular issue of broad interest: How can we integrate machine learning with physics-based modeling to develop new interpretable and truly reliable physical models? After introducing the general guidelines, we discuss the two most important issues for developing machine learning-based physical models: Imposing physical constraints and obtaining optimal datasets. We also provide a simple and intuitive explanation for the fundamental reasons behind the success of modern machine learning, as well as an introduction to the concurrent machine learning framework needed for integrating machine learning with physics-based modeling. Molecular dynamics and moment closure of kinetic equations are used as examples to illustrate the main issues discussed. We end with a general discussion on where this integration will lead us to, and where the new frontier will be after machine learning is successfully integrated into scientific modeling.

1 Fundamental laws and practical methodologies

Physics is centered on two main themes: the search for fundamental laws and the solution of practical problems. The former has resulted in Newton's laws, Maxwell equations, the theory of relativity and quantum mechanics. The latter has been the foundation of modern technology, ranging from automobiles, airplanes, computers to cell phones. In 1929, after quantum mechanics was just discovered, Paul Dirac made the following claim [1]:

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. "

What happened since then has very much confirmed Dirac's claim. It has been universally agreed that to understand problems in chemistry, biology, material science, and engineering, one rarely needs to look further than quantum mechanics for first principles. But solving practical problems using quantum mechanics principles, for example,

the Schrödinger equation, is a highly non-trivial matter, due, among other things, to the many-body nature of the problem. To overcome these mathematical difficulties, researchers have proceeded along the following lines:

1. Looking for simplified models. For example, Euler's equations are often enough for studying gas dynamics. There is no need to worry about the detailed electronic structure entailed in the Schrödinger equation.

2. Finding approximate solutions using numerical algorithms and computers.

3. Multi-scale modeling: In some cases one can model the behavior of a system at the macroscopic scale by using only a micro-scale model.

We will briefly discuss each.

1.1 Seeking simplified models

Seeking simplified models that either capture the essence of a problem or describe some phenomenon to a satisfactory accuracy has been a constant theme in physics. Ideally, we would like our simplified models to have the following properties:

1. express fundamental physical principles (e.g. conservation laws),

2. obey physical constraints (e.g. symmetries, frame-indifference),

3. be as universally accurate as possible: Ideally one can conduct a small set of experiments in idealized situations and obtain models that can be used under much more general conditions,

4. be physically meaningful (interpretable).

Euler's equations for gas dynamics is a very successful example of simplified physical models. It is much simpler than the first principles of quantum mechanics, and it is a very accurate model for dense gases. For ideal gases, the only parameter required is the gas constant. For complex gases, one needs the entire equation of state, which is a function of only two variables. Other success stories include the Navier-Stokes equations for viscous fluids, linear elasticity equations for small deformations of solids, and the Landau theory of phase transition.

Unfortunately, not every effort in coming up with simplified models has been as successful. A good example is the effort of developing extended Euler equations for rarified gas. Since the work of Grad [2], there have been numerous efforts on developing Euler-like models for the dynamics of gases at larger Knudsen numbers. So far this effort has not produced any widely accepted models yet.

To put things into perspective, we briefly discuss the methodologies that people use to obtain simplified models.
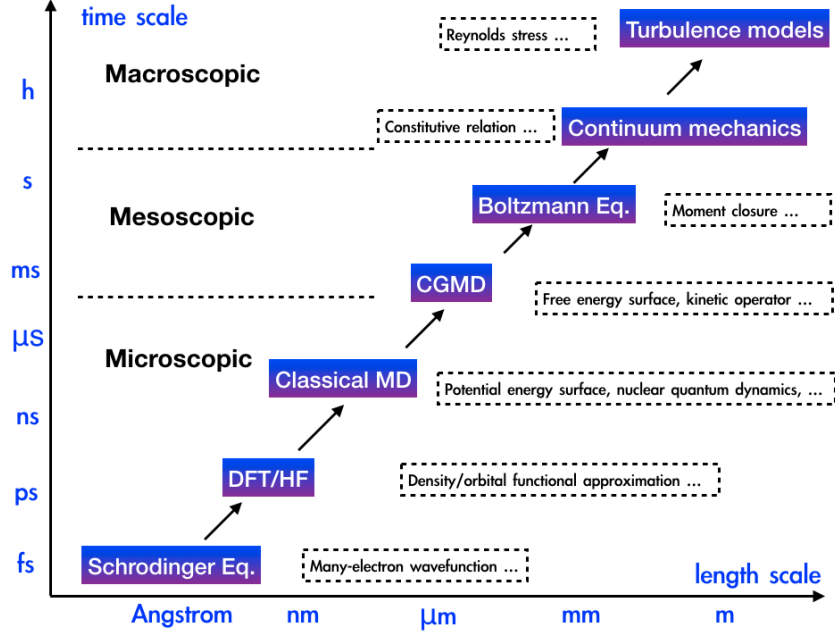
time scale

Turbulence models

Reynolds stress ...

Macroscopic

h

Constitutive relation ...   Continuum mechanics

s

Mesoscopic

Boltzmann Eq.   Moment closure ...

ms

CGMD   Free energy surface, kinetic operator ...

μs

Microscopic

Classical MD   Potential energy surface, nuclear quantum dynamics, ...

ns

DFT/HF   Density/orbital functional approximation ...

ps

fs   Schrodinger Eq.   Many-electron wavefunction ...   length scale

Angstrom   nm   μm   mm   m

Figure 1: Representative physical models at different scale and their most important modeling ingredients.

1. Generalized hydrodynamics [3]. The idea is to use symmetries, conservation laws, and the second law of thermodynamics to extract as much definitive information about the dynamics as possible, and model the rest using linear constitutive relations. A successful example is the Ericksen-Leslie equations for nematic liquid crystals, which works reasonably well in the absence of line defects [4].

2. The weakly nonlinear theory and gradient expansion trick. This is an idea championed by Landau. It has been successfully used in developing models for phase transition, superconductivity, hydrodynamic instability, and a host of other interesting physical problems.

3. Asymptotic analysis. This is a mathematical tool that can be used to systematically extract simplified models by exploiting the existence of some small parameters. For example, Euler's equation can be derived this way from the Boltzmann equation in the limit of small Knudsen number.

4. The Mori-Zwanzig formalism. This is a general strategy for eliminating unwanted degrees of freedom in a physical model. The price to be paid is that the resulting model becomes nonlocal, for example, with memory effects. For this reason, the application of the Mori-Zwanzig formalism has been limited to situations where the nonlocal term is linear. An example is the generalized Langevin equations [5].

5. Principal component-based model reduction. This is a broad class of techniques used in engineering for developing simplified models. A typical example is to project the original model onto a few principal components obtained using data produced from

the original model. Well-known applications of this technique include reduced models that aim to capture the dynamics of large scale coherent structures in turbulent flows [6].

6. Drastically truncated numerical schemes. The well-known Lorenz system is an example of this kind. It is the result of a drastic truncation of the two-dimensional incompressible Navier-Stokes-Boussinesq equation modeling thermal convection, an idealized model for weather prediction. By keeping only three leading modes for the stream-function and temperature, one arrives at the system

$$\dot{x} = \sigma(y - x), \quad \dot{y} = -xz + rx - y, \quad \dot{z} = xy - bz,$$

where σ is the Prandtl number, $r = Ra/Ra_c$ is the normalized Rayleigh number, and b is some other parameter. A well-known feature of this model is that it exhibits chaotic behavior, and this has been used as an indication for the intrinsic difficulties in weather prediction. Of course, the obvious counter-argument is how well the Lorenz system captures the dynamics of the original model.

1.2 Numerical algorithms

Since analytical solutions are rare even after we simplify the models, one has to resort to numerical algorithms to find accurate approximate solutions. Many numerical algorithms have been developed for solving the partial differential equations (PDEs) that arise from physics, including finite difference, finite element and spectral methods. The availability of these algorithms has completely changed the way we do science, and to an even greater extent, engineering. For example nowadays numerical computation plays a dominant role in the study of fluid and solid mechanics. Similar claims can be made for atmospheric science, combustion, material sciences and a host of other disciplines, though possibly to a lesser extend.

Roughly speaking, one can say that we now have satisfactory algorithms for low dimensional problems (say three dimensions). But things quickly become much more difficult as the dimensionality goes up. A good example is the Boltzmann equation: The dimensionality of the phase space and the nonlocality in the collision kernel makes it quite difficult to solve the Boltzmann equation using the kinds of algorithms mentioned above, even though the dimensionality is small compared to the ones encountered in the many-body Schrödinger equation.

This brings us to the issue that lies at the core of the many difficult problems that we face: The *curse of dimensionality (CoD)*: As the dimensionality grows, the complexity (or computational cost) grows exponentially.

1.3 Multi-scale modeling

One important idea for overcoming the difficulties mentioned above is multi-scale modeling [5], a general philosophy based on modeling the behavior of macro-scale systems using reliable micro-scale models, instead of relying on *ad hoc* macro-scale models. The idea is to make use of the results of the micro-scale model on much smaller spatial-temporal

domains to predict the kind of macro-scale quantities that we are interested in [5]. This general philosophy is valid for a wide range of scientific disciplines. But until now, the success of multi-scale modeling has been less spectacular than what was expected twenty years ago. The following challenges contributed to this.

1. The micro-scale models are often not that reliable. For example when studying crack propagation, we often use molecular dynamics as the micro-scale model. But the accuracy of molecular dynamics models for dynamic processes that involve bond breaking is often questionable.

2. Even though multi-scale modeling can drastically reduce the size of the micro-scale simulation required, it is still beyond our current capability.

3. The key benefit we are exploiting with multi-scale modeling is the separation of the micro- and macro-scales of the problem. But for the most interesting and most challenging problems, this often breaks down.

4. At a technical level, efficient multi-scale modeling requires effective algorithms for extracting the relevant information needed from micro-scale simulations. This is a data analysis issue that has not been adequately addressed.

There are two basic strategies for multi-scale modeling, the sequential multi-scale modeling and the concurrent multi-scale modeling. In sequential multi-scale modeling, the needed components from the micro-scale model, for example, the constitutive relation, are obtained beforehand, and this information is then supplied to some macro-scale model. For this reason, sequential multi-scale modeling is also called "precomputing". In concurrent multi-scale modeling, the coupling between the macro- and micro-scale models is done on the fly as the simulation proceeds. Sequential multi-scale modeling results in new models. Concurrent multi-scale modeling results in new algorithms.

1.4 Difficulties that remain

A lot of progresses have been made using these methodologies in combination with physical insight as well as trial and error parameter fitting. This has enabled us to solve a wide variety of problems, ranging from performing density functional theory (DFT) calculations to predict properties of materials and molecules, to studying the climate using general circulation models. In spite of these advances, many issues remain as far as getting good models is concerned, and many problems remain difficult. Here are some examples:

1. A crucial component of the Kohn-Sham DFT is the exchange-correlation functional. However, systematically developing efficient and accurate exchange-correlation functionals is still a very difficult task.

2. The most important component of a molecular dynamics model is the potential energy surface (PES) that describes the interaction between the nuclei in the system. Accurate and efficient models of PES has always been a difficult problem in molecular dynamics. We will return to this issue later.

3. An attractive idea for modeling the dynamics of macromolecules is to develop coarse-grained molecular dynamics models. However, implementing this idea in practice is still a difficult task.

4. Developing hydrodynamic models for non-Newtonian fluids is such a difficult task that the subject itself has lost some steam.

5. Moment closure models for rarified gases. This was mentioned above and will be discussed in more detail later.

6. For fluids, we have the Navier-Stokes equations. What is the analog for solids? Besides linear elasticity models, there are hardly any other universally accepted continuum models of solids. This comment applies to nonlinear elasticity. Plasticity is even more problematic.

7. Turbulence models. This has been an issue ever since the work of Reynolds and we still do not have a systematic and robust way of addressing this problem.

From the viewpoint of developing models, one major difficulty has always been the "closure problem": When constructing simplified models, we encounter terms that need to be approximated in order to obtain a closed system. Whether accurate closure can be achieved also depends in an essential way on the level at which we impose the closure, i.e. the variables we use to close the system. For example, for DFT models, it is much easier to perform the required closure for orbital-based models than for orbital-free models. For turbulence models, the closure problem is much simplified in the context of large eddy simulation than Reynolds average equations, since a lot more information is kept in the large eddy simulation.

From the viewpoint of numerical algorithms, these problems all share one important feature: There are a lot of intrinsic degrees of freedom. For example turbulent flows are governed by the Navier-Stokes equations, which is a low dimensional problem, but its highly random nature means that we should really be looking for its statistical description, which then becomes a very high dimensional problem with the dimensionality being proportional to the range of active scales. Existing numerical algorithms can not handle these intrinsically high dimensional problems effectively.

In the absence of systematic approaches, one has to resort to *ad hoc* procedures which are not just unpleasant but also unreliable. Turbulence modeling is a very good example of the kind of pain one has to endure in order to address practical problems.

1.5 Machine learning comes to the rescue

Recent advance in machine learning offers us unprecedented power for approximating functions of many variables. This allows us to go back to all the problems that were made difficult by CoD. It also provides an opportunity to reexamine the issues discussed above, with a new added dimension.

There are many different ways in which machine learning can be used to help solving problems that arise in science and engineering. We will focus our discussion on the following issue:

While machine learning can be a very powerful tool, it usually does not work so well when used as a blackbox. One main objective of this article is to discuss the important issues that have to be addressed when integrating machine learning with physics-based modeling.

1. The models should satisfy the requirements listed in Section 1.1.

2. The dataset that one uses to construct the model should be a good representation of all the practical situations that the model is intended for. It is one thing to fit some data. It is quite another thing to construct *reliable* physical models.

3. To reduce the amount of human intervention, the process of constructing the models should be end-to-end.

Later we will examine these issues in more detail after a brief introduction to machine learning.

2 Machine learning

Before getting into the integration of machine learning with physics-based modeling, let us briefly discuss an issue that is in many people's mind: What is the magic behind neural network-based machine learning? While this question is still at the center of very intensive studies in theoretical machine learning, some insight can already be gained from very simple considerations.

We will discuss a basic example, supervised learning: Given ther dataset $S = \{(\mathbf{x}_j, y_j = f^*(\mathbf{x}_j)), j = 1, 2, \cdots, n\}$, learn f^*. Here$\{\mathbf{x}_j, j = 1, 2, \cdots n\}$ is a set of points in the d dimensional Euclidean space. For simplicity we have neglected measurement noises. If the "labels" $\{y_j\}$ take discrete values, the problem is called a classification problem. Otherwise it is called a regression problem. In practice, one divides S into two subsets, a training set used to train the model and a testing set used to test the model.

A well-known example is the classification example of Cifar 10 dataset [7]. Here the task is to classify the images in the dataset into 10 categories. Each image has 32×32 pixels. Thus it can be viewed as a point in $32 \times 32 \times 3 = 3072$ dimensional space. The last factor of 3 comes from the dimensionality in the color space. Therefore this problem can be viewed as approximating a discrete-valued function of 3072 variables.

The essence of the supervised learning problem is the approximation of a target function using a finite dataset. While the approximation of functions is an ancient topic, in classical approximation theory, we typically use polynomials, piece-wise polynomials, wavelets, splines and other linear combinations of fixed basis functions to approximate a given function. These kinds of approximations suffer from CoD:

$$f^* - f_m \sim m^{-\alpha/d}\Gamma(f^*),$$

where m is the number of free parameters in the approximation scheme, α is some fixed constant depending on the approximation scheme, $\Gamma(f^*)$ is an error constant depending on the target function f^*. Take $\alpha = 1$. If we want to reduce the error by a factor of 10, the number of parameters needed goes up by 10^d. This is clearly impractical when the dimensionality d is large.

The one area in which we have lots of experiences and successes in handling high dimensional problems is statistical physics, the computation of high dimensional expectations using Monte Carlo (MC) methods. Indeed computing integrals of functions with millions of variables has become a routine practice in statistical physics that we forget to notice how remarkable this is. Let g and μ be a function and a probability distribution in the d dimensional Euclidean space respectively, and let

$$I(g) = \mathbb{E}_{\mathbf{x} \sim \mu} g(\mathbf{x}), \quad I_m(g) = \frac{1}{m} \sum_j g(\mathbf{x}_j),$$

where $\{\mathbf{x}_j\}$ is a sequence of i.i.d. samples of the probability distribution μ. Then we have

$$\mathbb{E}(I(g) - I_m(g))^2 = \frac{\operatorname{var}(g)}{m}, \quad \operatorname{var}(g) = \int_X g^2(\mathbf{x}) d\mathbf{x} - \left(\int_X g(\mathbf{x}) d\mathbf{x} \right)^2.$$

Note that the error rate $1/\sqrt{n}$ is independent of the dimensionality of the problem. Had we used grid-based algorithms such as the Trapezoidal rule, we would have instead:

$$I(g) - I_m(g) \sim m^{-\alpha/d} \Gamma(g),$$

resulting in the CoD. Note also that $\operatorname{var}(g)$ can be very large in high dimensions. This is why variance reduction is a central theme in MC.

The MC story is the golden standard for dealing with high dimensional problems. A natural question is: Can we turn the function approximation problem, the key issue in supervised learning, into MC integration type of problems? To gain some insight, let us consider the Fourier representation of functions:

$$f^*(\mathbf{x}) = \int_{\mathbb{R}^d} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega}, \mathbf{x})} d\boldsymbol{\omega}. \tag{1}$$

We are used to approximate this expression by some grid-based discrete Fourier transform

$$f^*(\mathbf{x}) \sim \frac{1}{m} \sum_j a(\boldsymbol{\omega}_j) e^{i(\boldsymbol{\omega}_j, \mathbf{x})}, \tag{2}$$

and this approximation suffers from CoD.

If instead that we consider functions represented by

$$f^*(\mathbf{x}) = \int_{\mathbb{R}^d} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega}, \mathbf{x})} \pi(d\boldsymbol{\omega}) = \mathbb{E}_{\boldsymbol{\omega} \sim \pi} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega}, \mathbf{x})}, \tag{3}$$

where π is a probability distribution on the Euclidean space $\mathbb{R}^d$, then the natural approximation becomes

$$f^*(\mathbf{x}) \sim \frac{1}{m} \sum_{j=1}^m a(\boldsymbol{\omega}_j) e^{i(\boldsymbol{\omega}_j, \mathbf{x})}, \tag{4}$$

where $\{\boldsymbol{\omega}_j\}$ are i.i.d. samples of π. For reasons discussed earlier, this approximation does not suffer from CoD. The right hand side of (4) is an example of neural network functions with one hidden layer and activation function σ defined by $\sigma(z) = e^{iz}$. If one wants a one-sentence description of the magic behind neural network models, the reasoning above should be a good candidate.

There is one key difference between the supervised learning problem and MC integration: In MC, the probability distribution μ is given, usually in fairly explicit form. In supervised learning, the probability distribution π is unknown. Instead we are only given information about the target function f^* on a finite dataset. Therefore it is infeasible to perform the approximation in (4). Instead one has to replace it by some optimization algorithms based on fitting the dataset.

3 Concurrent machine learning

In classical supervised learning, the labeled dataset is given beforehand and learning is performed afterwards. This is usually not the case when machine learning is used in connection with physical models. Instead, data generation and training is an interactive process: Data is generated and labeled on the fly as model training proceeds. In analogy with multi-scale modeling, we refer to the former class of problems "sequential machine learning" problems and the latter kind "concurrent machine learning" problems.

Active learning is something in between. In active learning, we are given the unlabeled data, and we decide which ones should be labeled during the training process. In concurrent learning, one also needs to generate the unlabeled data.

To come up with reliable physical models, one has to have reliable data: The dataset should be able to represent all the situations that the model is intended for. On the other hand, since data generation typically involves numerical solutions of the underlying micro-scale model, a process that is often quite expensive, we would like to have as small a dataset as possible. This calls for a procedure that generates the data adaptively in an efficient way.

3.1 The ELT algorithm

The (exploration-labeling-training) ELT algorithm was first formulated in Ref. [8] although similar ideas can be traced back for a long time. Starting with no (macro-scale) model and no data but with a micro-scale model, the ELT algorithm proceeds iteratively with the following steps:

1. **exploration**: explore the configuration space, and decide which configurations need to be labeled. The actual implementation of this step requires an algorithm for exploring the configuration space and a criterion for deciding which configurations need to be labeled.

   Oftentimes the current macro-scale model is used to help with the exploration.

2. **labeling**: compute the micro-scale solutions for the configurations that need to be labeled, and place them in the training dataset.

9

3. **training**: train the needed macro-scale model.

The ELT algorithm requires the following components: a macro-scale explorer, an indicator to evaluate whether a given configuration should be labeled, a micro-scale model used for the labeling, and a machine learning model for the quantities of interest.

One way of deciding whether a given configuration needs to be labeled is to estimate the error of the current machine learning model at the given configuration. When neural network models are used, a convenient error indicator can be constructed as the variance between the predictions given by an ensemble of neural network models (say with the same architecture but different initialization of the optimization algorithm). If the variance is small, then the predictions from different models are close to each other. In this case, it is likely that these predictions are already quite accurate at the given configuration. Therefore this configuration does not need to be labeled. Otherwise, it is likely that the predictions are inaccurate and the given configuration should be labeled.

3.2 Variational Monte Carlo

Another example where concurrent learning might be relevant is machine learning-based approach for variational Monte Carlo algorithms. Variational Monte Carlo (VMC) [9, 10], for solving the Schrödinger equation was among the first set of applications of machine learning in computational science [11, 12]. In these and other similar applications, one begins with no data and no solution, the data is generated on the fly as the solution process proceeds. Given a Hamiltonian $\hat{H}$ with the state variable $\mathbf{x}$, the variational principle states that the wave-function Ψ associated with the ground state minimizes within the required symmetry the following energy functional

$$E[\Psi] = \frac{\int \Psi^*(\mathbf{x})\hat{H}\Psi(\mathbf{x})\,d\mathbf{x}}{\int \Psi^*(\mathbf{x})\Psi(\mathbf{x})\,d\mathbf{x}} = \mathbb{E}_{\mathbf{x}\sim|\Psi(x)|^2}\frac{\hat{H}\Psi(\mathbf{x})}{\Psi(\mathbf{x})}. \tag{5}$$

In a machine learning-based approach, one parameterizes the wave-function using some machine learning model, $\Psi \sim \Psi_\theta$ with some parameter θ that needs to determined. In this case both the integrand and the distribution in (5) depend on θ. To find the optimal θ, the learning process consists of two alternating phases, the sampling phase and the optimization phase. In the sampling phase, a sampling algorithm is used to sample the probability distribution generated by the current approximation of Ψ^2. This sample is used to evaluate the loss function in (5). In the optimization phase, one iteration of the optimization algorithm is applied to the loss function. This procedure also resembles the EM (expectation-maximization) algorithm.

4 Molecular modeling

One of the most successful applications of machine learning to scientific modeling is in molecular dynamics. Molecular dynamics (MD) is a way of studying the atomic scale property of materials and molecules by tracking the dynamics of the nuclei in the system using classical Newtonian dynamics. The key issue in MD is how to model the potential

energy surface (PES) that describes the interaction between the nuclei. This is a function that depends on the position of all the nuclei in the system:

$$E = E(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_N),$$

Traditionally, there have been two ways to deal with this problem. The first is to compute the inter-atomic forces on the fly using first principles-based models such as the DFT [13]. This is the so-called *ab initio* molecular dynamics pioneered by Car and Parrinello [14]. This approach gives an accurate description of the system under consideration. However, it is computationally very expensive, limiting the size of the system that one can handle to thousands of atoms. At the other extreme is the approach of using empirical potentials. In a nutshell this approach aims at modeling the PES with empirical formulas. A well-known example is the Lennard-Jones potential, a reasonable model for describing the interaction between inert atoms:

$$V_{ij} = 4\epsilon \left((\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^6 \right), \quad E = \frac{1}{2} \sum_{i \neq j} V_{ij}.$$

Empirical potential-based MD is very efficient, but guessing the right formula that can model the PES accurately enough is understandably a very difficult task, particularly for complicated systems such as high entropy alloys. In practice one has to resort to a combination of physical insight and *ad hoc* approximation. Consequently the gain in efficiency is at the price of reliability. In particular, since these empirical potentials are typically calibrated using data for equilibrium systems, their accuracy at other interesting configurations, such as transition states, is often in doubt.

The pioneering work of Behler and Parrinello introduced a new paradigm for performing *ab initio* molecular dynamics [15]. In this new paradigm, a first principle-based model acts as the vehicle for generating highly accurate data. Machine learning is then used to parametrize this data and output a model of the PES with *ab initio* accuracy. See also the work of Csanyi et al [16].

Behler and Parrinello introduced a neural network architecture that is naturally extensive (see Figure 2). In this architecture, each nucleus is associated with a subnetwork. When the system size is increased, one just has to append the existing network with more subnetworks corresponding to the new atoms added.

As discussed before, to construct truly reliable models of PES, one has to address two problems. The first is getting good data. The second is imposing physical constraints.

Let us discuss the second problem first. The main physical constraints for modeling the PES are the symmetry constraints. Naturally the PES model should be invariant under translational and rotational symmetries. It should also be invariant under the relabeling of atoms of the same chemical species. This is a permutational symmetry condition. Translational symmetry is automatically taken care of by putting the origin of the local coordinate frame for each nucleus at that nucleus. To deal with other symmetry constraints, Behler and Parrinello resorted to constructing "local symmetry functions" using bonds, bond angles, etc., in the same spirit as the constructions in the Stillinger-Weber potential [18]. This construction is a bit *ad hoc* and is vulnerable to the criticism mentioned earlier for empirical potentials.
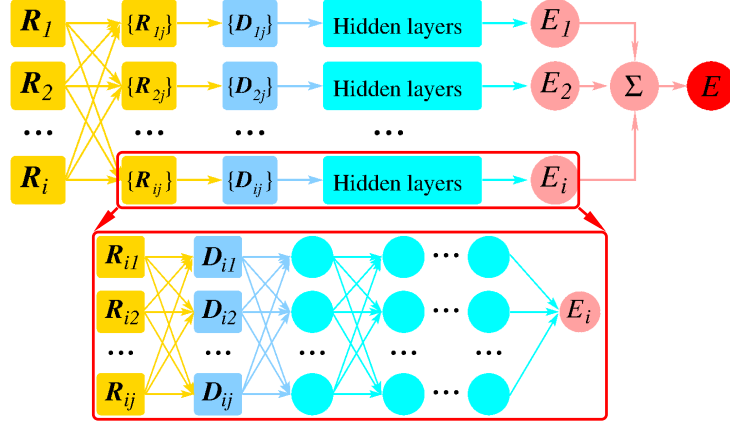
Figure 2: Illustration of the neural network architecture for a system of N atoms. From Ref. [17].

To demonstrate the importance of preserving the symmetry, we show in Figure 3 the comparison of the results with and without imposing symmetry constraints. One can see that without enforcing the symmetry, the test accuracy of the neural network model is rather poor. Even a poor man's version of enforcing symmetry can improve the test accuracy drastically.

A poor man's way of enforcing symmetry is to remove by hand the degrees of freedom associated with these symmetries. This can be accomplished as follows [19, 17]:

- enforcing rotational symmetry by fixing in some way a local frame of reference. For example, for atom i, we can use its nearest neighbor to define the x-axis and use the plane spanned by its two nearest neighbors to define the z-axis, thereby fixing the local frame.

- enforcing permutational symmetry by fixing an ordering of the atoms in the neighborhood. For example, within each species we can sort the atoms according to their distances to the reference atom i.

This simple operation allows us to obtain neural network models with very good accuracy, as shown in Figure 3 (a).

The only problem with this procedure is that it creates small discontinuities when the ordering of the atoms in a neighborhood changes. These small discontinuities, although negligible for sampling a canonical ensemble, show up drastically in a microcanonical molecular dynamics simulation, as can be seen in Figure 3 (b).

To construct a smooth potential energy model that satisfies all the symmetry constraints we have to reconsider how to represent the general form of symmetry-preserving functions. The idea is to precede the fitting network in each subnetwork by an embedding network that produces a sufficient number of symmetry-preserving functions. In this way the end model is automatically symmetry-preserving. The PES generated this way is called (the smooth version of) Deep Potential [20]. MD driven by Deep Potential is called DeePMD. The idea of embedding network is fairly general and can be used in other situations where symmetry is an important issue [21, 22].
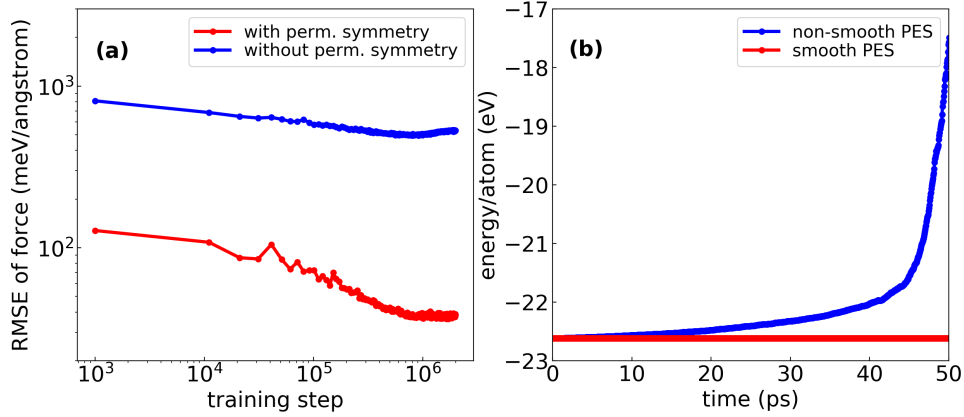
12

Figure 3: (a) Test accuracy without imposing the permutational symmetry (blue) and with a poor man's way of enforcing the permutational constraints (red). (b) Total energy per atom as a function of time from microcanonical DeePMD simulations, using a poor man's way to impose symmetry constraints (blue) and using the smooth embedding network (red).

To construct the general form of symmetry-preserving functions, we draw inspiration from the following two observations.

Translation and Rotation. The matrix $\Omega_{ij} \equiv \boldsymbol{r}_i \cdot \boldsymbol{r}_j$ is an over-complete array of invariants with respect to translation and rotation [23, 24], i.e., it contains the complete information of the point pattern $\boldsymbol{r}$. However, this symmetric matrix switches rows and columns under a permutational operation.

Permutation. We consider a generalization of Theorem 2 of Ref. [25]: A function $f(\boldsymbol{r}_1, ..., \boldsymbol{r}_i, ..., \boldsymbol{r}_N)$ is invariant to the permutation of instances in $\boldsymbol{r}_i$, if and only if it can be decomposed into the form $\rho(\sum_i g(\boldsymbol{r}_i)\boldsymbol{r}_i)$, for suitable transformations g and ρ.

We now go back to the first problem, the generation of good data using the ELT algorithm. The objective is to develop a procedure that can generate Deep Potentials for a given system that can be used in a wide variety of thermodynamic conditions. In the implementation in [26], exploration was done using the following ideas. At the macro-scale level, one samples the (T, p) space of thermodynamic variables in some way. For each value of (T, p), one samples the canonical ensemble, using MD with the Deep Potential available at the current iteration. In addition, the efficiency of the exploration can be improved by initializing the MD using a variety of different initial configurations, say different crystal structures. Labeling was done using the Kohn-Sham DFT with periodic boundary condition. Training was done using Deep Potential. An example is given in Figure 4. In this example, only ~0.005% of the configurations explored was labeled [26]. In particular, the Deep Potential constructed this way satisfies all the requirements listed in Section 1.5.

As a first example of the applications of these ideas, we show in Figure 5 the accuracy of the Deep Potential for a wide variety of systems, from small molecules, to large molecules, to oxides and high entropy alloys. One can see that in all the cases stud-
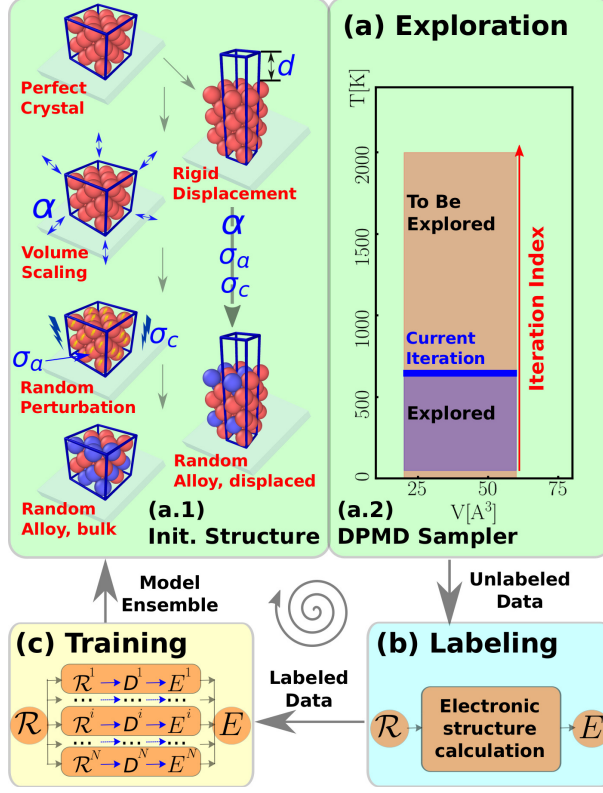
13

Figure 4: Schematic plot of one iteration of the DP-GEN scheme, taking the Al-Mg system as an example. (a) Exploration with DeePMD. (a.1) Preparation of initial structures. We start from stable crystalline structures of pure Al and Mg, compress and dilate the stable structures uniformly to allow for a larger range of number densities, and then randomly perturb the atomic positions and cell vectors of all the initial crystalline structures. Surface-related structures are generated with rigid displacement. Based on configurations of pure metal we also generate random alloy structures. (a.2) Canonical simulation at a given temperature. (b) Labeling with electronic structure calculations. (c) Training with the DP model. From [26].

ied, Deep Potential achieves an accuracy comparable to that of the underlying DFT. As a second example, we show the results of using Deep Potential to study the structural information of liquid water (see Figure 6). One can see that the results for the radial and angular distribution functions are comparable to the results from *ab initio* MD. As a third example, combined with the state-of-art high performance computing platform (Summit), molecular dynamics simulations with *ab initio* accuracy have been pushed to systems of up to 100 million atoms, making it possible to study more complex phenomena that require truly large-scale simulations [27, 28]. An application to nanocrystalline copper was presented in Ref. [28]. Deep Potential and DP-GEN have been implemented in the open-source packages DeePMD-kit [29] and DP-GEN [30], respectively, and have attracted researchers from various disciplines.

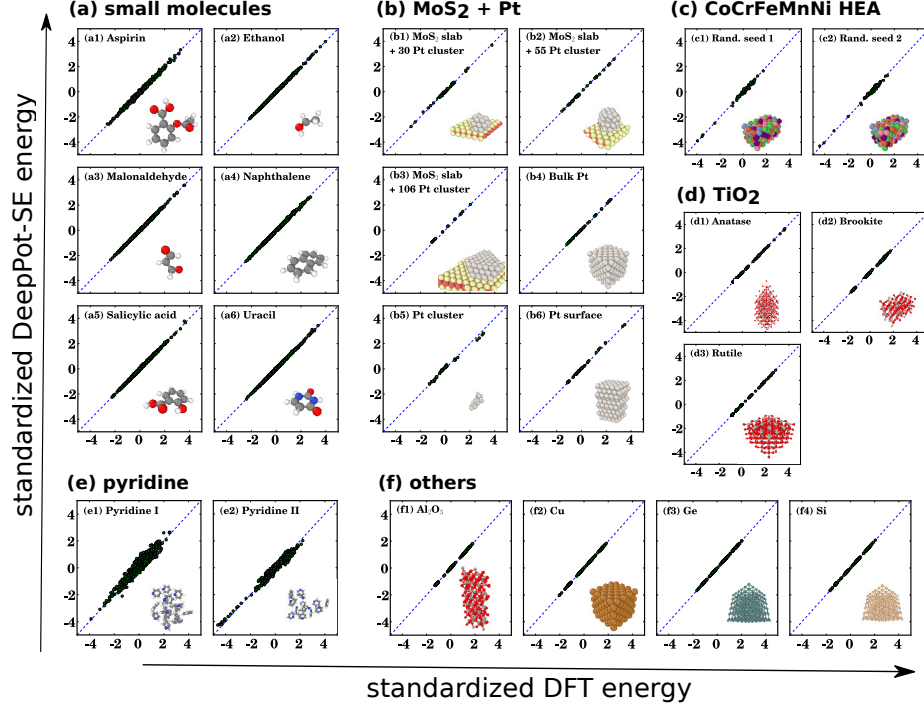For other related work, we refer to [31, 32, 33].

Figure 5: Comparison of the DFT energies and the Deep Potential-predicted energies on the testing snapshots. The energies are standardized for a clear comparison. (a) Small molecules. (b) MoS_2 and Pt. (c) CoCrFeMnNi high-entropy alloy (HEA). (d) TiO_2. (e) Pyridine (C_5H_5N). (f) Other systems: Al_2O_3, Cu, Ge, and Si. From [20].
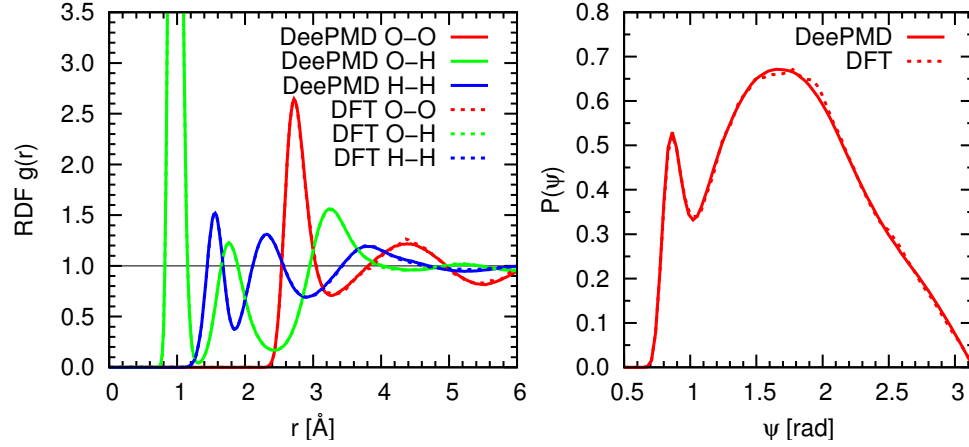

Figure 6: Correlation functions of liquid water from DeePMD and PI-AIMD. Left: radial distribution functions. Right: the O-O-O angular distribution function. From [17].

5 Moment closure for kinetic models of gas dynamics

The dynamics of gases is described very well by the Boltzmann equation for the single-particle phase space density function $f = f(\mathbf{x}, \boldsymbol{v}, t) : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^+$:

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\mathbf{x}} f = \frac{1}{\varepsilon} Q(f). \tag{6}$$

Here we use ε to denote the Knudsen number, Q is the collision operator. When ε is small, the Boltzmann equation can be accurately reduced to the Euler equation for the macroscopic variables, ρ, $\boldsymbol{u}$ and T which represent density, bulk velocity and temperature respectively:

$$\rho = \int_{\mathbb{R}^3} f \, d\boldsymbol{v}, \quad \boldsymbol{u} = \frac{1}{\rho} \int_{\mathbb{R}^3} f \boldsymbol{v} \, d\boldsymbol{v}, \quad T = \frac{1}{3\rho} \int_{\mathbb{R}^3} f |\boldsymbol{v} - \boldsymbol{u}|^2 \, d\boldsymbol{v}. \tag{7}$$

$$\partial_t \boldsymbol{U} + \nabla_{\mathbf{x}} \cdot \boldsymbol{F}(\boldsymbol{U}) = 0, \tag{8}$$

with $\boldsymbol{U} = (\rho, \rho \boldsymbol{u}, E)^T$, $\boldsymbol{F}(\boldsymbol{U}) = (\rho \boldsymbol{u}, \rho \boldsymbol{u} \otimes \boldsymbol{u} + pI, (E + p)\boldsymbol{u})^T$, $p = \rho T$, $E = \frac{1}{2}\rho |\boldsymbol{u}|^2 + \frac{3}{2}\rho T$. Euler's equation can be obtained by projecting Boltzmann's equation onto the first few moments that defined in (7), and making use of the local Maxwellian approximation to close the system:

$$f_M(\boldsymbol{v}) = \frac{\rho}{(2\pi T)^{\frac{3}{2}}} \exp\left(-\frac{|\boldsymbol{v} - \boldsymbol{u}|^2}{2T}\right). \tag{9}$$

The accuracy of the Euler's equation deteriorates when the Knudsen number increases, since the local Maxwellian is no longer a good approximation for the solution of the Boltzmann equation. To improve upon Euler's equation, Grad proposed the idea of using more moments to arrive at extended Euler-like equations [2]. As an example, Grad proposed the well-known 13-moment equation. Unfortunately Grad's 13-moment equation suffers from all kinds of problems, among which is the loss of hyperbolicity in certain regions of the state space. Since then, many different proposals have been made in order to obtain reliable moment closure systems. The usual procedure is as follows.

1. Start with a choice of a finite-dimensional linear subspace of functions of $\boldsymbol{v}$ (usually some set of polynomials, e.g., Hermite polynomials).

2. Expand $f(\mathbf{x}, \boldsymbol{v}, t)$ using these functions as bases and take the coefficients as moments (including macroscopic variables ρ, $\boldsymbol{u}$, T, etc.).

3. Close the system using some simplified assumptions, e.g., truncating moments of higher orders.

For instance, in Grad 13-moment system, the moments are constructed using the basis $\{1, \boldsymbol{v}, (\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u}), |\boldsymbol{v} - \boldsymbol{u}|^2 (\boldsymbol{v} - \boldsymbol{u})\}$. It is fair to say that at the moment, this effort has not really succeeded.

There are two important issues that one has to address in order to obtain reliable moment closure systems. The first is: What is the best set of moments that one should

use? The second is: How does one close the projected system? The second problem is the well-known closure problem.

One possible machine learning-based approach is as follows [34].

1. Learn the set of generalized moments using some machine learning models such as the auto-encoder. The problem can be formulated as follows: Find an encoder Ψ that maps $f(\cdot, \boldsymbol{v})$ to generalized moments $\boldsymbol{W} \in \mathbb{R}^M$ and a decoder Φ that recovers the original f from $\boldsymbol{U}, \boldsymbol{W}$

$$\boldsymbol{W} = \Psi(f) = \int \boldsymbol{w} f \, \mathrm{d}\boldsymbol{v}, \quad \Phi(\boldsymbol{U}, \boldsymbol{W})(\boldsymbol{v}) = h(\boldsymbol{v}; \boldsymbol{U}, \boldsymbol{W}).$$

The goal is to find an optimal $\boldsymbol{w}$ and h parametrized by neural networks through minimizing

$$\mathbb{E}_{f \sim \mathcal{D}} \| f - \Phi(\Psi(f)) \|^2 + \lambda_\eta (\eta(f) - h_\eta(\boldsymbol{U}, \boldsymbol{W}))^2.$$

where $\eta(f)$ denotes the entropy.

We call $(\boldsymbol{U}, \boldsymbol{W})$ the set of generalized moments.

2. Learn the reduced model for the set of generalized moments.

Projecting the kinetic equation onto the set of generalized moments, one obtains equations of the form:

$$\begin{cases} \partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon) = 0, \\ \partial_t \boldsymbol{W} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{G}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon) = \dfrac{1}{\varepsilon} \boldsymbol{R}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon). \end{cases} \tag{10}$$

This is the general conservative form of the moment system. The term $1/\varepsilon$ is inherited directly from (6). Our task is to learn $\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{R}$ from the original kinetic equation.

To get reliable reduced models, we still have to address the issues of getting good data and respecting physical constraints. To address the first problem, one can use the ELT algorithm with the following implementation [34]:

- exploration: random initial conditions made up of random waves and discontinuities.

- labeling: solving the kinetic equation. This can be done locally in the momentum space since the kinetic models have a finite speed of propagation.

- training: as discussed above.

To address the second problem, one notices that there is an additional symmetry: the Galilean invariance. This is a dynamic constraint respected by the kinetic equation. Specifically, for every $\boldsymbol{u} \in \mathbb{R}^3$, define

$$f'(\mathbf{x}, \boldsymbol{u}, t) = f(\mathbf{x} - t\boldsymbol{u}', \boldsymbol{v} - \boldsymbol{u}', t).$$

If f is a solution of the Boltzmann equation, then so is f'. To enforce similar constraints for the reduced model, we define the Galilean-invariant moments by:

$$\boldsymbol{W}_{\mathrm{Gal}} = \Psi(f) = \int f(\boldsymbol{v}) \boldsymbol{w} \left(\frac{\boldsymbol{v} - \boldsymbol{u}}{\sqrt{T}} \right) \mathrm{d}\boldsymbol{v}. \tag{11}$$

Modeling the dynamics of $\boldsymbol{W}_{\text{Gal}}$ becomes more subtle due to the spatial dependence in $\boldsymbol{u}, T$. For simplicity we will work with a discretized form of the dynamic model in the closure step. Suppose we want to model the dynamics of $\boldsymbol{W}_{\text{Gal},j}$ at the spatial grid point indexed by j. Integrating the Boltzmann equation against the generalized basis at this grid point gives

$$\partial_t \int_{\mathbb{R}^D} f(\boldsymbol{v}) \boldsymbol{w}\left(\frac{\boldsymbol{v}-\boldsymbol{u}_j}{\sqrt{T_j}}\right) \mathrm{d}\boldsymbol{v} + \nabla_{\boldsymbol{x}} \cdot \int_{\mathbb{R}^D} f(\boldsymbol{v}) \boldsymbol{w}\left(\frac{\boldsymbol{v}-\boldsymbol{u}_j}{\sqrt{T_j}}\right) \boldsymbol{v}^{\mathrm{T}} \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^D} \frac{1}{\varepsilon} Q(f) \boldsymbol{w}\left(\frac{\boldsymbol{v}-\boldsymbol{u}_j}{\sqrt{T_j}}\right) \mathrm{d}\boldsymbol{v}.$$
(12)

The closes equations now take the form

$$\partial_t \boldsymbol{W}_{\text{Gal}} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{G}_{\text{Gal}}(\boldsymbol{U}, \boldsymbol{W}_{\text{Gal}}; \boldsymbol{U}_j) = \frac{1}{\varepsilon} \boldsymbol{R}_{\text{Gal}}(\boldsymbol{U}, \boldsymbol{W}_{\text{Gal}}).$$
(13)

One can perform machine learning using this form.

The numerical results given in [34] shows the promise of this approach. Here we only give an example of using this approach to study shock wave structure in high-speed rarefied gas flows, when the gas experiences a fast transition in a tube between two equilibrium states. During the transition the flow deviates substantially from the thermodynamical equilibrium. The classical Navier-Stokes-Fourier (NSF) equations fail when the Mach number is bigger than 2. Figure 7 presents a comparison of the shock wave profiles with the BGK (BhatnagarGrossKrook) collision model [35] when the Mach number is 5.5, and the results of the machine learning-based closure model with 3 additional Hermite moments (HermMLC). One can see that the machine learning-based model achieves good agreements with the original Boltzmann equation.
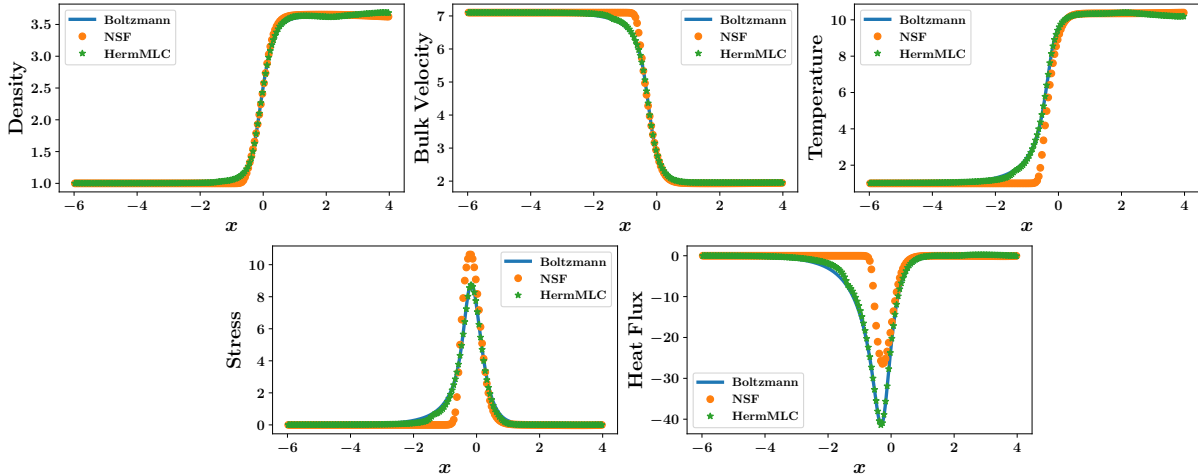

Figure 7: Profiles of mass density, bulk velocity, temperature, normal stress, and heat flux when the Mach number equals 5.5, obtained from the Boltzmann equation, the Navier-Stokes-Fourier equations, and HermMLC.

6 Discussions and concluding remarks

We have focused on developing sequential multi-scale models using current machine learning. These models are very much like the physical models we are used to, except that some functions in the models exist in the form of machine learning models such as neural network functions. In this spirit, these models are no different from Euler's equations for complex gases where the equation of state is stored as tables or subroutines. As we discussed above, following the right protocols will ensure that these machine learning-based models are just as reliable as the original micro-scale models.

In principle the same procedure and same set of protocols are also applicable in a purely data-driven context without a micro-scale model: One just has to replace the labeler by experimental results. In practice, this is still a relatively unexplored territory.

Another way to integrate machine learning with physics-based modeling is from the viewpoint of learning dynamical systems. After all, the physical models we are looking for are all examples of dynamical systems. One representative machine learning model for learning dynamical systems is the recurrent neural networks [36]. In this connection, an interesting observation made in [37] is that there is a natural connection between the Mori-Zwanzig formalism and the recurrent neural network model. Recurrent neural network models are machine learning models for time series. It models the output time series using a local model by introducing hidden variables. In the absence of these hidden variables, the relationship between the input and output time series will be nonlocal with memory effects. In the presence of these hidden variables (which are learned as part of the training process), the relationship becomes local. Therefore, recurrent neural network models can be viewed as a way of unwrapping the memory effects by introducing hidden variables. One can also think of it as a way of performing Mori-Zwanzig by compressing the space of unwanted degrees of freedom. This connection and this viewpoint of integrating machine learning with physics-based modeling has yet to be explored.

Going back to the seven problems listed in Section 1.4, we have already addressed problems 2 and 5. For problem 1, some progress has been reported in [38] and [39, 40] for molecules. Similar models/algorithms as DeePMD and Deep Potential have been developed for coarse-grained molecular dynamics [41]. For problem 4, some initial progress has been made in [42]. For problem 7, efforts that use neural network models can be traced back to [43] but there is still a huge gap compared with the standards promoted in this paper. Problem 6 is wide open. Overall, one should expect substantial progress to be made on all these problems in the next few years.

This brings us to a new question: What will be the next frontier, after machine learning is successfully integrated into physics-based models? With the new paradigm firmly in place, the key bottleneck will quickly become the micro-scale models that act as the generators of golden standard data. In the area of molecular modeling, this will be the solution of the original many-body Schrödinger equation. In the area of turbulence modeling, this will be the Navier-Stokes equation. It is possible that machine learning can also play an important role in developing the next-generation algorithms for these problems [11, 44].

References

[1] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society A*, 123(792):714–733, 1929.

[2] Harold Grad. On the kinetic theory of rarefied gases. *Communications on Pure and Applied Mathematics*, 2(4):331–407, 1949.

[3] Sybren Ruurds De Groot and Peter Mazur. *Non-equilibrium thermodynamics*. Courier Corporation, 2013.

[4] Pierre-Gilles De Gennes and Jacques Prost. *The physics of liquid crystals*, volume 83. Oxford university press, 1993.

[5] Weinan E. *Principles of multiscale modeling*. Cambridge University Press, 2011.

[6] Hendrik Tennekes and John Leask Lumley. *A first course in turbulence*. MIT press, 1972.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] Linfeng Zhang, Han Wang, and Weinan E. Reinforced dynamics for enhanced sampling in large atomic and molecular systems. *The Journal of Chemical Physics*, 148(12):124113, 2018.

[9] William Lauchlin McMillan. Ground state of liquid He 4. *Physical Review*, 138(2A):A442, 1965.

[10] David Ceperley, Geoffrey V Chester, and Malvin H Kalos. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, 1977.

[11] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[12] Weinan E and Bing Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[13] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133, 1965.

[14] Roberto Car and Michele Parrinello. Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters*, 55(22):2471, 1985.

[15] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[16] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

[17] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120:143001, Apr 2018.

[18] Frank H Stillinger and Thomas A Weber. Computer simulation of local order in condensed phases of silicon. *Physical Review B*, 31(8):5262, 1985.

[19] Jiequn Han, Linfeng Zhang, Roberto Car, and Weinan E. Deep potential: a general representation of a many-body potential energy surface. *Communications in Computational Physics*, 23(3):629–639, 2018.

[20] Linfeng Zhang, Jiequn Han, Han Wang, Wissam A Saidi, Roberto Car, and Weinan E. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. In *Advances of the Neural Information Processing Systems (NIPS)*, 2018.

[21] Linfeng Zhang, Mohan Chen, Xifan Wu, Han Wang, Weinan E, and Roberto Car. Deep neural network for the dielectric response of insulators. *arXiv preprint arXiv:1906.11434*, 2019.

[22] Grace Sommers, Marcos Calegari, Linfeng Zhang, Han Wang, and Roberto Car. Raman spectrum and polarizability of liquid water from deep neural networks. *Physical Chemistry Chemical Physics*, 2020.

[23] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

[24] Hermann Weyl. *The Classical Groups: Their Invariants and Representations*. Princeton university press, 2016.

[25] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[26] Linfeng Zhang, De-Ye Lin, Han Wang, Roberto Car, and Weinan E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Physical Review Materials*, 3(2):023804, 2019.

[27] Denghui Lu, Han Wang, Mohan Chen, Jiduan Liu, Lin Lin, Roberto Car, Weinan E, Weile Jia, and Linfeng Zhang. 86 pflops deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy. *arXiv preprint arXiv:2004.11658*, 2020.

[28] Weile Jia, Han Wang, Mohan Chen, Denghui Lu, Jiduan Liu, Lin Lin, Roberto Car, Weinan E, and Linfeng Zhang. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. *arXiv preprint arXiv:2005.00223*, 2020.

[29] Han Wang, Linfeng Zhang, Jiequn Han, and Weinan E. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications*, 228:178 – 184, 2018.

[30] Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and Weinan E. Dp-gen: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, page 107206, 2020.

[31] Evgeny V Podryabinkin and Alexander V Shapeev. Active learning of linearly parametrized interatomic potentials. *Computational Materials Science*, 140:171–180, 2017.

[32] Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148(24):241733, 2018.

[33] Noam Bernstein, Gábor Csányi, and Volker L Deringer. De novo exploration and self-guided learning of potential-energy surfaces. *npj Computational Materials*, 5(1):1–9, 2019.

[34] Jiequn Han, Chao Ma, Zheng Ma, and Weinan E. Uniformly accurate machine learning-based hydrodynamic models for kinetic equations. *Proceedings of the National Academy of Sciences*, 116(44):21983–21991, 2019.

[35] Prabhu Lal Bhatnagar, Eugene P Gross, and Max Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511, 1954.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[37] Chao Ma, Jianchun Wang, and Weinan E. Model reduction with memory and the machine learning of dynamical systems. *Communications in Computational Physics*, 25(4):947–962, 2018.

[38] Yixiao Chen, Linfeng Zhang, Han Wang, and Weinan E. Ground state energy functional with hartree-fock efficiency and chemical accuracy. *arXiv preprint arXiv:2005.00169*, 2020.

[39] Lixue Cheng, Matthew Welborn, Anders S Christensen, and Thomas F Miller III. A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules. *The Journal of Chemical Physics*, 150(13):131103, 2019.

[40] Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. Completing density functional theory by machine learning hidden messages from molecules. *npj Computational Materials*, 6(1):1–8, 2020.

[41] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. DeePCG: constructing coarse-grained models via deep neural networks. *The Journal of Chemical Physics*, 149(3):034101, 2018.

[42] Huan Lei, Lei Wu, and Weinan E. Machine learning based non-newtonian fluid model with molecular fidelity. *arXiv preprint arXiv:2003.03672*, 2020.

[43] F Sarghini, G De Felice, and S Santini. Neural networks based subgrid scale modeling in large eddy simulations. *Computers & fluids*, 32(1):97–108, 2003.

[44] Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, 2019.