La Rivista del Nuovo Cimento (2025) 48:1–94
https://doi.org/10.1007/s40766-025-00065-4

REVIEW PAPER

Foundations of molecular dynamics simulations: how and what

Giovanni Ciccotti[1,2,3] · Sergio Decherchi[4] · Simone Meloni[5]

Received: 2 May 2024 / Accepted: 15 December 2024 / Published online: 28 February 2025

Abstract
In this review, we discuss computational methods to study condensed matter systems and processes occurring in this phase. We begin by laying down the theoretical framework of statistical mechanics starting from the fundamental laws governing nuclei and electrons. Among others, we present the connection between thermodynamics and statistical mechanics using a pure statistical language, which makes it easier to extend the microscopic interpretation of thermodynamic potentials to other relevant quantities, such as the Landau free energy (also known as the potential of the mean force). Computational methods for estimating the relevant quantities of equilibrium and non-equilibrium statistical mechanics systems, as well as reactive events, are discussed. An extended Appendix is added, where we present artificial intelligence methods recently introduced. These methods can enhance the power of atomistic simulations, allowing to achieve at the same time accuracy and efficiency in the calculation of the quantities of interest.

Keywords Statistical mechanics · Thermodynamics · Atomistic simulations · Molecular dynamics

✉ Giovanni Ciccotti
giovanni.ciccotti@roma1.infn.it

✉ Sergio Decherchi
sergio.decherchi@iit.it

✉ Simone Meloni
simone.meloni@unife.it

1 Istituto per le Applicazioni del Calcolo "Mauro Picone"-IAC, Consiglio Nazionale delle Ricerche, Via dei Taurini 19, 00185 Rome, Italy

2 Department of Physics, University of Rome "La Sapienza", Piazzale Aldo Moro 2, 00185 Rome, Italy

3 School of Physics, University College Dublin, Belfield, Dublin 4, Dublin, Ireland

4 Data Science and Computation Facility, Fondazione Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genoa, Italy

5 Department of Chemical, Pharmaceutical and Agricultural Sciences-DOCPAS, University of Ferrara, Via Luigi Borsari 46, 44121 Ferrara, Italy

1 Introduction

The general framework to describe and predict the behavior of macroscopic matter, statistical mechanics, was laid down in the 19th century by Boltzmann and Gibbs. The first ingredient of this theoretical framework is the atomistic hypothesis: matter is made of *particles* obeying Newton's equations. Complications came about when, in the 20th century, it was found that matter *obeys* quantum mechanics. This, however, does not disrupt our theoretical framework, as discussed in the following. Boltzmann postulated that since the characteristic time of the motion of atoms is much shorter than the macroscopic time of our observations, macroscopic phenomena result from the time average of microscopic events along the trajectory of these particles. Boltzmann figured out that in equilibrium, the microscopic states of the system are distributed according to a suitable probability density (ensemble). This idea was further developed and widened by Gibbs, who observed that the concept of average along a trajectory was unnecessary, that the role of this trajectory is sampling microscopic states (drawn from an ensemble), which are observed with a given frequency along it. Thus, the Boltzmann's *time average* is turned into an *ensemble average*. This two-time framework, fast atoms and slow observations, is insufficient to represent all phenomena occurring in nature. For example, chemical reactions are (at least) a three-time phenomenon:

(i) the time of fast individual atomic motions (e.g., vibrations),
(ii) the characteristic time of the reaction at hand, with the ensuing bond rearrangement,
(iii) the equilibration time (in principle infinite, in practice long but finite) the system takes to go from the initial condition, e.g., when the system contains only the reactant, to the final equilibrium state, with the proper chemical composition of the system, containing the proper amount of reactant and product species.

Thus, the theoretical description of reactive processes within the framework initiated by Boltzmann and Gibbs requires the introduction of a macroscopic *intermediate* time, a time which is long on the atomistic scale and short on the equilibrium scale. A *local* time average around this time allows one to determine the value of the macroscopic observables at that time.

So far, we implicitly excluded that external perturbations are acting on the system. However, the statistical mechanics framework can be extended to this case. Here, the time-dependence of the macroscopic observable arises not only from the initial conditions, possibly out of equilibrium, but from the action of external biases, being them either genuinely time-dependent or constant but acting on the system starting from a given time.

The framework can also be extended to take into account the fact that the *particles* constituting matter obey quantum mechanics. Additionally, a connection can be drawn between statistical mechanics and thermodynamics, achieving the double objective of providing to thermodynamics a theoretical foundation and a systematic procedure to calculate its quantities, which otherwise can only be obtained from experiments , e.g., the value of the heat capacity, or from approximate phenomenological laws, e.g, the van der Waals equation of state. With these two extensions, statistical mechanics is a complete theory providing the laws to describe and predict properties of matter in

equilibrium or under the action of external perturbations. A key problem is that the solution of these laws is of exceptional complexity. Thus, initially, apart the simplest cases, the community could not follow the plan given by the fundamental dynamical laws: apply the theory to physical problems and obtain, using approximations when necessary, all possible consequences. The advent of computers, and their availability to the scientific community in the 1940s, opened novel perspectives: numerically solve the equations of statistical mechanics. These laws typically require calculation of averages of suitable microscopic observables over ensembles (mechanical properties) or calculation of absolute frequencies (partition functions) associated to thermal quantities. Brute force approaches to compute these integrals require a prohibitive effort (see Sect. 3) and the founding fathers of computational physics proposed an alternative approach: (i) one introduces a model of the real system, a computational system made of a large enough number of particles suitably interacting among them, and (ii) evolves these particles according to some suitable law (which has to be established, it is not part of the general statistical mechanics theory) sampling from the given ensemble and computing averages. This strategy, however, is not straightforward. For example, while it is perfectly suitable for computing simple averages, it does not allow, in practice, to compute absolute frequencies, hence it is of no help to compute thermal properties. e.g., thermodynamic potentials. Additionally, while methods for sampling equilibrium ensembles were immediately identified, for non-equilibrium problems, where one does not explicitly know the mathematical form of the time-dependent ensemble, a way to formulate sampling methods has to be found. Even more, while statistical mechanics of the early days focused on sampling states of a system, the power of its descriptive capacity encouraged physicists (as well as mathematicians and chemists) to extend its use to the statistical analysis of *paths* connecting states of the system. These concepts provide the statistical mechanics framework for (bio)chemical, (bio)physical processes and, in the dreams of Kirkwood and Irving, Alder and many others, the theoretical foundation of engineering of this and next centuries. As for sampling paths, the problem required the introductions of novel statistical concepts. For example, while in basic statistics, one asks what is the probability that a given event occurs, the statistics of paths requires to sample probabilities that time-correlated events occur. In other words, the availability of computers allowed to estimate not only relevant quantities of statistical mechanics, but also processes, so enabling to address questions of growing complexity, e.g., reactive paths. This, in turn, fostered theoretical research to formulate problems and equations in a form that can be better solved on computers, also bringing to the development of novel numerical analyses and algorithms. This has been supported by computational capabilities which progressed exponentially (Moore law) over the last $\sim$ 30–40 years.

The objective of this paper is to lay down the theoretical framework of statistical mechanics to describe condensed phase system starting from the fundamental laws governing nuclei and electrons. This requires several steps, which are summarized below in this introduction. We start by presenting and discussing the theoretical framework of statistical mechanics. After, we introduce computational methods to estimate the quantities of interest within this theory. In particular, we focus on methods based on *molecular dynamics* (MD). We are well-aware that complementary approaches exist based on Monte Carlo [1]. However, the generous but limited space available to this

article, our expertise in molecular dynamics, and the goal of writing an article in which the theory and methods are discussed in sufficient detail, and not just listed, convinced us to focus only on methods belonging to MD. For analogous reasons, we do not discuss all MD-based methods available in the literature. In other words, concerning methods, we privileged depth to breath.

The plan of this work is the following: (i) Starting from quantum mechanics of a system made up of nuclei and electrons, we derive a representation of the system made up of classical nuclei and quantum electrons. Then we show that under a large set of conditions of interest for systems on Earth, one can reduce the problem to the knowledge of the electronic ground state (depending on nuclear configuration) or, with increasing approximations, *basic* electron excitations. (ii) This entitles one to develop the equilibrium and time-dependent statistical mechanics of classical nuclei interacting via a potential deriving from electrons in their ground state. (iii) The connection between thermodynamics and statistical mechanics is introduced and discussed. (iv) (standard) Molecular Dynamics is introduced to compute expectation values of quantities of interest for equilibrium statistical mechanics in the so-called microcanonical ensemble. (v) This case is exploited to discuss the key ingredients of Molecular Dynamics, in particular the so-called *classical* molecular dynamics, where interatomic forces are approximated by empirical force models. (vi) modifications of Newton's/Hamilton's equations of motion (EoM) are introduced to allow the sampling of ensembles beyond the microcanonical one. Here a general theory of the statistical mechanics of non-Hamiltonian systems is presented, systems governed by equations of motion that cannot be derived according to Hamilton. (vii) Next we consider the case in which forces are derived from electronic ground state calculations. (viii) We proceed introducing special techniques for computing ordinary and Landau free energies. As alluded above, these quantities in general cannot be straightforwardly computed by Molecular Dynamics. Landau free energy is a key ingredient in the calculation of (ix) rates of processes governed by barriers and (x) techniques for finding the most likely reactive path and other relevant properties for these processes. (xi) We conclude the review considering the case of non-equilibrium systems. In this case, no general technique exists for investigating all non-equilibrium systems and problems. Thus, here we focus on two cases: the calculation of transport coefficients and evolution of a system initially at some constrained equilibrium relaxing to equilibrium or evolving under the action of an external perturbation. (xii) The article is completed by an extensive Appendix discussing artificial intelligence methods to enhance atomistic simulations. We conclude with some comments on future perspectives.

2 Classical statistical mechanics in condensed matter

In this section (i) we will introduce the system at the core of this article, a sample of nuclei and electrons; (ii) we will explain what legitimates us, within suitable conditions, to treat it as a system of point particles sitting at the nuclear positions and interacting *via* an effective potential; (iii) summarize the statistical description of this system; finally; (iv) we will connect the fundamental quantities of statistical mechanics to their thermodynamic counterparts.

2.1 From the quantum to the classical mechanics of condensed matter systems at low energies

Let us first introduce the systems we consider in this article. We focus on condensed phase systems at *low energies*, i.e., in the regime in which relativistic effects can be neglected and nuclei are stable objects that can be treated as point particles, i.e., we can avoid to explicitly take into account the strong and weak forces among nucleons. Additionally, we consider systems that do not interact with photons (absorbing/emitting). Summarizing, we deal with systems made of nuclei and electrons interacting through electrostatic forces, possibly under the action of external fields. The gravitational force among these particles can be neglected as this is much weaker than the other terms. This does not imply that the effect of an external gravitational field can be neglected. Indeed, physical phenomena such as Benárd convective cells are the result of combined thermal gradient and gravity field.

A system such as the one described above evolves according to the (time-dependent) Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},\mathbf{R},t)=\mathcal{H}(\mathbf{r},\mathbf{R},t)\Psi(\mathbf{r},\mathbf{R},t), \tag{1}$$

where $\Psi(\mathbf{r},\mathbf{R},t)$ is the wavefunction of the system made by N_e electrons and N_n nuclei at time t. $\mathbf{r}$ and $\mathbf{R}$ are the $3N_e$ and $3N_n$ dimensional vectors of the electron's and nuclei's positions, respectively. $\mathcal{H}(\mathbf{r},\mathbf{R},t)$ is the Hamiltonian of the system, here assumed to be of general time-dependent form. This Hamiltonian can be split into two parts: an *internal* and an *external* contribution:

$$\mathcal{H}(\mathbf{r},\mathbf{R},t)=\mathcal{H}_{int}(\mathbf{r},\mathbf{R})+\mathcal{V}_{\text{ext}}(\mathbf{r},\mathbf{R},t). \tag{2}$$

The external, possibly time-dependent, term is the contribution to the Hamiltonian arising from any external field, e.g., the gravitational field. The internal term contains kinetic contributions of electrons and nuclei and the Coulomb interactions among the particles composing the system:

$$\begin{aligned}\mathcal{H}_{int}(\mathbf{r},\mathbf{R})=&\sum_{i=1}^{N_e}\left(-\frac{\hbar^2}{2m_e}\nabla^2_{\mathbf{r}_i}\right)+\sum_{\alpha=1}^{N_n}\left(-\frac{\hbar^2}{2M_\alpha}\nabla^2_{\mathbf{R}_\alpha}\right)\\&+\sum_{i>j=1}^{N_e}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}+\sum_{\alpha>\beta=1}^{N_n}\frac{z_\alpha z_\beta}{|\mathbf{R}_\alpha-\mathbf{R}_\beta|}\\&-\sum_{i=1}^{N_e}\sum_{\alpha=1}^{N_n}\frac{z_\alpha}{|\mathbf{r}_i-\mathbf{R}_\alpha|}.\end{aligned} \tag{3}$$

In Eq. (3), $\nabla_{\mathbf{r}_i}$ and $\nabla_{\mathbf{R}_\alpha}$ denote the gradients with respect to the position of the electron i and nucleus α, respectively. m_e is the mass of electrons, while M_α is the mass of nucleus α. We assume the so-called *atomic units*, where the charge of electrons

and protons is -1 and $+1$, respectively. Finally, z_α is the atomic number of nucleus α, i.e., its charge in atomic units. For reasons that will be clear shortly, we recast the equation above as the sum of two terms:

$$\begin{cases} \mathcal{H}_e(\mathbf{r};\mathbf{R}) = \sum_{i=1}^{N_e}\left(-\frac{\hbar^2}{2m_e}\nabla^2_{\mathbf{r}_i}\right) \\ \qquad\qquad + \sum_{i>j=1}^{N_e}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} - \sum_{i=1}^{N_e}\sum_{\alpha=1}^{N_n}\frac{z_\alpha}{|\mathbf{r}_i-\mathbf{R}_\alpha|} \\ \qquad\qquad + \sum_{\alpha>\beta=1}^{N_n}\frac{z_\alpha z_\beta}{|\mathbf{R}_\alpha-\mathbf{R}_\beta|} \\ \mathcal{K}_N = \sum_{\alpha=1}^{N_n}\left(-\frac{1}{2M_\alpha}\nabla^2_{\mathbf{R}_\alpha}\right). \end{cases} \tag{4}$$

The first term, dubbed *electronic Hamiltonian*, contains the kinetic energy operator of the electrons plus electrons–electrons, electrons–nuclei and nuclei–nuclei potential energy terms. The second term is the kinetic energy operator of the nuclei. Let us consider the eigenfunctions $\{\chi_j(\mathbf{r};\mathbf{R})\}_{j=0,\infty}$ and the corresponding eigenvalues $\{\epsilon_j(\mathbf{R})\}_{j=0,\infty}$ of the electronic Hamiltonian:

$$\mathcal{H}_e(\mathbf{r};\mathbf{R})\chi_j(\mathbf{r};\mathbf{R}) = \epsilon_j(\mathbf{R})\chi_j(\mathbf{r};\mathbf{R}). \tag{5}$$

We remark that $\{\chi_j(\mathbf{r};\mathbf{R})\}_{j=0,\infty}$ and $\{\epsilon_j(\mathbf{R})\}_{j=0,\infty}$ depend (parametrically, in the first case) on $\mathbf{R}$ because the electronic Hamiltonian does. One can expand $\Psi(\mathbf{r},\mathbf{R},t)$ on $\{\chi_j(\mathbf{r};\mathbf{R})\}_{j=0,\infty}$: $\Psi(\mathbf{r},\mathbf{R},t) = \sum_{j=0,\infty}\xi_j(\mathbf{R},t)\chi_j(\mathbf{r};\mathbf{R})$. First, we consider the case where no external (time-dependent) potential ($\mathcal{V}_{\text{ext}}(\mathbf{r},\mathbf{R},t)$) acts on the system. Inserting this representation of $\Psi(\mathbf{r},\mathbf{R},t)$ in Eq. (1) one gets

$$\begin{aligned} \sum_{j=0,\infty}\chi_j(\mathbf{r};\mathbf{R})\, i\hbar\frac{\partial\xi_j(\mathbf{R},t)}{\partial t} &= \sum_{j=0,\infty}\left[\mathcal{K}_N(\mathbf{R}) + \mathcal{H}_e(\mathbf{r};\mathbf{R})\right]\chi_j(\mathbf{r};\mathbf{R})\xi_j(\mathbf{R},t) \\ &= \sum_{j=0,\infty}\chi_j(\mathbf{r};\mathbf{R})\left[\mathcal{K}_N(\mathbf{R}) + \epsilon_j(\mathbf{R})\right]\xi_j(\mathbf{R},t) \\ &\quad + \sum_{j=0,\infty}\xi_j(\mathbf{R},t)\mathcal{K}_N(\mathbf{R})\chi_j(\mathbf{r};\mathbf{R}) \\ &\quad + \sum_{j=0,\infty}\sum_{\alpha=1}^{N_n}\left(-\frac{1}{M_\alpha}\right)\nabla_{R_\alpha}\xi_j(\mathbf{R},t)\cdot\nabla_{R_\alpha}\chi_j(\mathbf{r};\mathbf{R}) \end{aligned} \tag{6}$$

where in the second equality, we exploited the fact that $\chi_j(\mathbf{r};\mathbf{R})$ is an eigenfunction of $\mathcal{H}_e(\mathbf{r};\mathbf{R})$. Neglecting the last two terms of the r.h.s., i.e., those related to the derivative of $\chi_j(\mathbf{r};\mathbf{R})$ on the nuclear degrees of freedom, is the so-called *Born–Oppenheimer* (BO) approximation. Within the BO approximation, Eq. (6) turns into a set of independent *effective* Schrödinger equations, one per eigenstate $\chi_j(\mathbf{r};\mathbf{R})$ of the electronic

Hamiltonian, of the form

$$i\hbar\frac{\partial\xi_j(\mathbf{R},t)}{\partial t} = \left[\mathcal{K}_N(\mathbf{R}) + \epsilon_j(\mathbf{R})\right]\xi_j(\mathbf{R},t) \tag{7}$$

where $\epsilon_j(\mathbf{R})$, the j-th eigenvalue of the electronic Hamiltonian, plays the role of an effective potential driving the dynamics of the *effective* eigenfunction $\xi_j(\mathbf{R},t)$. A consequence of the Born–Oppenheimer approximation is that if the system is initially in a state consistent with a single term of the expansion on $\left\{\chi_j(\mathbf{r};\mathbf{R})\right\}_{j=0,\infty}$, it will remain forever in this state. Considering the difference between $\epsilon_j(\mathbf{R})$ of different electronic states, a system in thermal equilibrium at the typical temperatures of most phenomena occurring on the planet Earth is in its ground electronic state. This implies that we can consider only the term $j = 0$ of the set of Eq. (7).

Next, we focus on some general characteristics of $\xi_j(\mathbf{R},t)$, namely the order of magnitude of its (thermal) wavelength, Λ. Consider the de Broglie relation, $\Lambda = \hbar/\sqrt{M_\alpha k_\mathrm{B} T}$, with k_B the Boltzmann constant. Thus, for example, for carbon at room temperature $\Lambda \sim 10^{-12}$ m, two orders of magnitude smaller than typical values of bond lengths, for covalently bonded systems, and interatomic distances, at ordinary pressures, 10^{-10}–10^{-9} m. This entitles one to treat nuclei as classical point particles obeying Newton's equations of motion driven by the potential $\epsilon_j(\mathbf{R})$:

$$M\ddot{\mathbf{R}} = -\nabla_R\epsilon_0(\mathbf{R}). \tag{8}$$

Here, $\epsilon_0(\mathbf{R})$ is the ground state eigenvalue of the electronic Hamiltonian. A practical implementation of Eq. (8) takes advantage of the so-called Hellmann–Feynmann theorem, [2, 3] showing that $\nabla_R\epsilon_0(\mathbf{R}) = \langle\nabla_R\mathcal{H}(\mathbf{r};\mathbf{R})\rangle_0$. Here, $\langle\cdot\rangle_0$ denotes the expectation value on the electronic ground state wavefunction. We remark that an equivalent theorem exists for the density functional theory approach to the quantum many-body problem, [4] which is the quantum mechanical approach most used these days in simulations. This approach requires the calculation of the ground state eigenfunction of the electronic Hamiltonian within some suitable approximation. This is computationally very expensive, unfortunately.

For many systems, it resulted possible to approximate $\epsilon_0(\mathbf{R})$ with a many-body empirical potential function explicitly depending on nuclear positions. This many-body potentials typically depend on parameters, whose value can be obtained from the fitting of experimental properties or quantum mechanics data (see, e.g., Ref. [5]). Often, these many-body potentials can be expressed as sum of additive pair potential terms. A typical example is the Lennard–Jones potential, embodying the short-distance repulsive and the long-range attractive (van deer Waals) interactions between non-covalently bonded atoms. Given N atoms in the configuration $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_N)$:

$$\epsilon_\mathrm{LJ}(\mathbf{R}) = \sum_{\alpha>\beta} 4\varepsilon\left[\left(\frac{\sigma}{R_{\alpha\beta}}\right)^{12} - \left(\frac{\sigma}{R_{\alpha\beta}}\right)^{6}\right] \tag{9}$$

where $R_{\alpha\beta}$ is the distance between point particles α and β representing the corresponding atoms. A detailed description of so-called *force fields* for covalently bonded and

non-bonded systems is left for the following. In the Appendix A, we introduce potentials exploiting recent developments in the field of *artificial intelligence*. Here, we simply remark that these empirical potentials, requiring no calculation of the ground state of the electronic Hamiltonian, are computationally very efficient.

Let us now consider the effect of an external, possibly time-dependent, field acting on the system *via* the potential $\mathcal{V}_{\text{ext}}(\mathbf{r}, \mathbf{R}, t)$. In the limit of weak perturbation, e.g., when the potential is small with respect to the energy difference between the ground and first excited electronic state, the problem can be treated within the first-order perturbation theory (see Ref. [6], page 111 and following). Here, one represents the time-dependent wavefunction in terms of projection on the stationary solutions of the unperturbed system. Non-negligible projections are restricted to those stationary states of energy within $2\pi\hbar/t$, where t is the time from the beginning of the perturbation. Thus, already on the time scale of the nuclear motion, few femtoseconds, only stationary states within few tenths of eV have a non-negligible projection and, as a consequence, semiconducting and insulating systems (including molecular systems) can be safely described by their electronic ground state. Thus, Eq. (7) (restricted to the electronic ground state) can be written as:

$$\begin{aligned} i\hbar\frac{\partial\xi_0(\mathbf{R},t)}{\partial t} &= \left[\mathcal{K}_N(\mathbf{R}) + \epsilon_0(\mathbf{R}) + \int \mathrm{d}\mathbf{r}\ |\chi_0(\mathbf{r};\ \mathbf{R})|^2\, \mathcal{V}_{\text{ext}}(\mathbf{r},\mathbf{R},t)\right]\xi_0(\mathbf{R},t) \\ &= \left[\mathcal{K}_N(\mathbf{R}) + \epsilon_0(\mathbf{R}) + \mathcal{V}^{\text{eff}}_{\text{ext}}(\mathbf{R},t)\right]\xi_0(\mathbf{R},t) \end{aligned} \tag{10}$$

where $\mathcal{V}^{\text{eff}}_{\text{ext}}(\mathbf{R},t) = \int \mathrm{d}\mathbf{r}\ |\chi_0(\mathbf{r})|^2\, \mathcal{V}_{\text{ext}}(\mathbf{r},\mathbf{R},t)$ is an effective external, possibly time-dependent, potential where the effect on electrons in taken into account in a mean-field sense. If, same as in Eq. (8), nuclei are treated as classical particles, then forces acting on nuclei arise from the gradient of the electronic eigenvalue of the ground state plus the effective external potential:

$$M\ddot{\mathbf{R}} = -\nabla_R\left[\epsilon_0(\mathbf{R}) + \mathcal{V}^{\text{eff}}_{\text{ext}}(\mathbf{R},t)\right]. \tag{11}$$

2.1.1 Equilibrium statistical mechanics

Within the framework we just developed, any property is a function of the phase space of nuclei, which also parametrically depends on the properties of the corresponding atoms, e.g., their mass, effective charge, etc. Thus, the generic property reads $A(\mathbf{\Gamma}; \{m_\alpha\}_{\alpha=1,N}, \{q_\alpha\}_{\alpha=1,N}, \dots)$. In absence of any time-dependent external field, $A(\mathbf{\Gamma}; \{m_\alpha\}_{\alpha=1,N}, \{q_\alpha\}_{\alpha=1,N}, \dots)$ depends on time *via* $\mathbf{\Gamma}$, the point in the phase space evolving according to the equations of motion. However, on the macroscopic scales, in equilibrium conditions, these properties have no time-dependence. Thus, following Boltzmann, macroscopic observables correspond to time average of their associated microscopic ones:

$$A = \bar{A} = \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \mathrm{d}s\ \hat{A}(\mathbf{\Gamma}(s); \{m_\alpha\}_{\alpha=1,N}, \{q_\alpha\}_{\alpha=1,N}, \dots) \tag{12}$$

Hereafter, the *bar*, e.g., $\bar{A}$, denotes time average and *hat*, e.g., $\hat{A}$, the microscopic observable associated to the corresponding macroscopic one. Here and in the following, we will replace $\hat{A}(\mathbf{\Gamma}(s); \{m_\alpha\}_{\alpha=1,N}, \{q_\alpha\}_{\alpha=1,N}, \dots)$ with the shorter notation $\hat{A}(\mathbf{\Gamma}(s))$. Other quantities which will parametrically depend on $\{m_\alpha\}_{\alpha=1,N}, \{q_\alpha\}_{\alpha=1,N}$, etc. will also be denoted with a similar, shorter notation, avoiding to explicitly report the parametric dependence on these quantities in the symbol. One can transform the time average of Eq. (12) into an ensemble average, i.e., an average over the probability density to be at the $\mathbf{\Gamma}$ point in phase space. To illustrate this, one (i) first discretizes the integral of Eq. (12), (ii) reorganizes the terms of the sum according to a tessellation of the phase space, and (iii) recognizes that this sum over the tessellation of the phase space corresponds to a phase space integral.

$$\begin{aligned}\bar{A} &= \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau \mathrm{d}s\, \hat{A}(\mathbf{\Gamma}(s)) \sim \lim_{\tau\to\infty} \frac{1}{\tau}\sum_{i=1}^{N} \hat{A}(\mathbf{\Gamma}(s_i))h \\ &= \lim_{\tau\to\infty} \frac{h}{\tau}\sum_{i=1}^{N} \hat{A}(\mathbf{\Gamma}(s_i)) = \lim_{N=\tau/h\to\infty} \frac{1}{N}\sum_{i=1}^{N} \hat{A}(\mathbf{\Gamma}(s_i)) \\ &= \lim_{N\to\infty} \sum_{\alpha=1}^{N_c} \hat{A}_\alpha \frac{n_\alpha}{N} = \frac{\lim_{N\to\infty}}{\lim_{\Delta\Omega_\alpha\to 0}} \sum_{\alpha=1}^{N_c} \hat{A}_\alpha \frac{p_\alpha}{\Delta\Omega_\alpha}\Delta\Omega_\alpha = \int_\Omega \mathrm{d}\mathbf{\Gamma}\, \hat{A}(\mathbf{\Gamma})\rho(\mathbf{\Gamma}) = \langle\hat{A}\rangle.\end{aligned} \tag{13}$$

where h is the discretization step of the integral over time, with $N = \tau/h$ the corresponding number of discretization points, n_α is the number of $\mathbf{\Gamma}(s_i)$ points falling in the αth cell of the tessellation, $\hat{A}_\alpha$ is the value of $\hat{A}(\mathbf{\Gamma})$ in a suitable point within cell α, N_c the number of cells and the limit of $\frac{n_\alpha}{N\Delta\Omega_\alpha}$ is finite. Thus, for a fixed step in time, h, the limit for τ going to infinity turns into the corresponding limit on N, the elements of the sum over the trajectory. $\Delta\Omega_\alpha$ is the volume of the phase space cell of the tessellation. $p_\alpha = \lim_{N\to\infty} n_\alpha/N$ is the probability to find the system in the αth cell and $\rho_\alpha = p_\alpha/\Delta\Omega_\alpha$ is the corresponding probability density. $\hat{A}_\alpha$ is the average value of $\hat{A}(\mathbf{\Gamma})$ computed over the phase space points falling in the αth cell. Finally, $\langle\cdot\rangle$ denotes ensemble average. The equivalence between time and ensemble average is the so-called *ergodic hypothesis*. The ergodic hypothesis implies that each phase space cell is visited an infinite number of times with a probability density consistent with that of the *ensemble* simulated by the molecular dynamics setup (see below). In other words, we assume that there are no impediments, i.e., no region of the phase space is prevented to be visited in an infinitely long trajectory. Summarizing, the value of any macroscopic observable can be obtained from the ensemble average of the corresponding microscopic one:

$$A_{XYZ} = \langle\hat{A}\rangle_{XYZ} = \int \mathrm{d}\mathbf{\Gamma}\, \hat{A}(\mathbf{\Gamma})\, \rho_{XYZ}(\mathbf{\Gamma}) \tag{14}$$

Here, the indexes XYZ of the macroscopic observable, A_{XYZ}, and of the probability density function, $\rho_{XYZ}(\mathbf{\Gamma})$, highlight the conditions the macroscopic system is

subject to. For example, A_{NVT} denotes properties computed in conditions of constant number of particles, N, volume, V, and temperature, T, according to the corresponding probability density function $\rho_{NVT}(\mathbf{\Gamma})$: the so-called canonical ensemble.

Within statistical mechanics, a phase space point is a random variable and, through it, so is any microscopic observable $\hat{A}(\mathbf{\Gamma})$. Hence, one can ask what is the probability (density) to observe a given value, A^*, of $\hat{A}(\mathbf{\Gamma})$:

$$\rho^{A}_{XYZ}(A^*) = \int \mathrm{d}\mathbf{\Gamma}\, \rho_{XYZ}(\mathbf{\Gamma})\, \delta(\hat{A}(\mathbf{\Gamma}) - A^*) \tag{15}$$

The meaning of Eq. (15) is that the probability that $\hat{A}(\mathbf{\Gamma}) = A^*$ is equal to the probability to be in any phase space point consistent with the given condition. The delta function within the integral selects these phase space points.

One can also define the joint probability of observing the system at a phase space point $\mathbf{\Gamma}$ and given values of a set $\{A_i\}_{i=1,m}$ of observables

$$\rho^{\{A_i\}_{i=1,m}}_{XYZ}(\mathbf{\Gamma}, \left\{A_i^*\right\}_{i=1,m}) = \rho_{XYZ}(\mathbf{\Gamma}) \prod_{i=1}^{m} \delta(\hat{A}_i(\mathbf{\Gamma}) - A_i^*), \tag{16}$$

together with the associated conditional probability (density):

$$\rho^{\{A_i\}_{i=1,m}}_{XYZ}(\mathbf{\Gamma} | \left\{A_i^*\right\}_{i=1,m}) = \frac{\rho_{XYZ}(\mathbf{\Gamma}) \prod_{i=1}^{m} \delta(\hat{A}_i(\mathbf{\Gamma}) - A_i^*)}{\int \mathrm{d}\mathbf{\Gamma}\, \rho_{XYZ}(\mathbf{\Gamma}) \prod_{i=1}^{m} \delta(\hat{A}_i(\mathbf{\Gamma}) - A_i^*)}. \tag{17}$$

Notice that Eq. (15) can be meant as the marginal probability density $\rho^{\{A_i\}_{i=1,m}}_{XYZ}(\left\{A_i^*\right\}_{i=1,m})$ of the joint probability density $\rho^{\{A_i\}_{i=1,m}}_{XYZ}(\mathbf{\Gamma}, \left\{A_i^*\right\}_{i=1,m})$ defined in Eq. (16).

Of course, the probabilities of phase space and/or observables contain more information than just the average value. For example, they can tell whether the system contains metastabilities, i.e., large local maxima of the probability density. For some systems, these local/absolute maxima are separated by regions of low probability. This is the scenario of many chemical reactions and physical processes discussed in Sect. 5. In these cases, for a system lying in a metastable state to reach the stable (absolute maximum, if it exists,) one, it must pass through low probability regions. This makes this process unlikely, thus occurring *once in a blue moon*. This is the conceptual framework beneath the kinetics of many processes—like chemical reactions, phase transitions, conformational isomerization (including protein folding)—characterized by the presence of barriers.

2.1.2 Non-equilibrium statistical mechanics

Let us now lift the condition imposed in the previous section, that the system is at the equilibrium. We remark that this is just the declaration of non-equilibrium statistical

mechanics, with a detailed discussion reserved to Sect. 6. In non-equilibrium systems, the probability density function is, generally speaking, time-dependent: $\rho(\mathbf{\Gamma}, t)$. $\rho(\mathbf{\Gamma}, t)$ obeys a continuity equation, the Liouville equation:

$$\frac{\mathrm{d}\rho(\mathbf{\Gamma}, t)}{\mathrm{d}t} = 0 = \left(\frac{\partial}{\partial t} + \dot{\mathbf{\Gamma}} \cdot \nabla_{\mathbf{\Gamma}}\right) \rho(\mathbf{\Gamma}, t). \tag{18}$$

Notice that Eq. (18) holds only for Hamiltonian systems. Equation (18) can be proven by computing the flux of trajectories across the boundary of an arbitrary volume in the phase space. Such a flux is equal to the variation of the probability density function with time, with positive sign if the overall flux is inward, and negative otherwise. The surface integral underneath the calculation of the flux of trajectories can be turned into a volume integral over the divergence of the current through the Gauss' divergence theorem. If the system is Hamiltonian, $\partial\dot{q}/\mathrm{d}q = -\partial\dot{p}/\mathrm{d}p$ and one is left with Eq. (18).

2.2 From classical statistical mechanics to thermodynamics

In thermodynamics, any quantity can be derived from the so-called thermodynamic potentials: entropy $S(N, V, E)$, enthalpy $H(N, P, S)$, Helmholtz's $F(N, V, T)$ or Gibbs's $G(N, P, T)$ free energy, and grand potential $\Phi(\mu, V, T)$, depending on the thermodynamic variables one controls: number of particles (N), chemical potential (μ), volume (V), pressure (P), energy (E), temperature (T). For example, recalling that $V = (\partial G(N, P, T)/\partial P)_{N,T}$, the isothermal compressibility, $\beta = -1/V\,\partial V/\partial P$, can be expressed in terms of derivatives of the Gibbs' free energy with respect to pressure P. Above, we have shown that given the probability density function $\rho_{XYZ}(\mathbf{\Gamma})$, one can compute the so-called *mechanical properties*, namely those properties that can be expressed as an ensemble average of microscopic observables. These include some of the thermodynamic variables, such as P, E and T (or additional macroscopic variables in the case of more complex systems). More in general, being founded on a microscopic description of matter, statistical mechanics *contains* thermodynamics. Thus, we are left to establish an identification between thermodynamics potentials and statistical mechanics' quantities.

Let us first focus on the case of constant number of particles, constant volume and constant energy system, denoted NVE in the following. The associated thermodynamic potential is the entropy, $S(N, V, E)$. Entropy is additive, i.e., the entropy of a (macroscopic, bulk) system grows linearly with its size, and is a non-decreasing function of E. Consider the absolute frequency that a system stays at given values of N, V, and E, $P(N, V, E)$. An absolute frequency is a non-normalized probability (density) to observe an event. Under the assumption that any microscopic state is equiprobable at NVE, $P(N, V, E)$ is proportional to the number of microscopic states consistent with these macroscopic conditions: $\int_{H(\mathbf{\Gamma})=E} \mathrm{d}\mathbf{\Gamma} \equiv \int \mathrm{d}\mathbf{\Gamma}\, \delta(H(\mathbf{\Gamma}) - E)$. Apart constant terms, $1/h^{3N}N!$, this is the partition function $\Omega_S(N, V, E)$. Notice that, apart the named multiplicative constant, $\Omega_S(N, V, E)$ is the (surface) integral of the invariant measure of the $N-1$ dimensional hyper-surface

$H(\mathbf{\Gamma}) = E$: $\int_{\partial\Omega_S(N,V,E)} \mathrm{d}\sigma/|\nabla H(\Gamma)|$. For regular enough Hamiltonians, this integral is a non-decreasing quantity of E, which is one of the properties of entropy. However, $\Omega_S(N, V, E)$ scales exponentially with the number of particles in the system, while $S(N, V, E)$ being additive scales only linearly. Thus, one identifies the ln of $\Omega_S(N, V, E)$ with the entropy and so, for dimensional consistency,

$$S(N, V, E) := k_\mathrm{B} \ln \Omega_S(N, V, E) \tag{19}$$

where k_B is the Boltzmann constant. This probabilistic interpretation of entropy can be extended to the case of the other ensembles. For example, the Gibbs and Helmholtz free energies and the Gran Potential can be identified with the logarithm of the corresponding partition function:

$$F(N, V, T) := -k_\mathrm{B} T \ln \Omega_F(N, V, T) \tag{20a}$$

$$G(N, P, T) := -k_\mathrm{B} T \ln \Omega_G(N, P, T) \tag{20b}$$

$$\Phi(\mu, V, T) := -k_\mathrm{B} T \ln \Omega_\Phi(\mu, V, T) \tag{20c}$$

In all cases, thermodynamic potentials measure the non-normalized marginal probability densities to be in the given macroscopic conditions understood as random variables [7].

3 Molecular dynamics

In this article, molecular dynamics is introduced as a tool for implementing statistical mechanics in realistic cases of interest for condensed matter physics, chemistry, biochemistry, biophysics, engineering, etc. (see Sect. 2). Initially, this might look bizarre: in Sect. 2, effort has been spent to turn time into ensemble averages and, apparently, here we follow the opposite path. To illustrate why molecular dynamics can help implementing the plan of statistical mechanics, let us make some back of the envelope calculation to prove that a direct calculation of the integrals associated to ensemble averages is impossible. Imagine one has a computational sample containing 1000 atoms, representing a reasonably sized model of a real system (see below). Imagine one wants to compute the ensemble average of an observable depending only on the atomic coordinates $\boldsymbol{R}$. A possible intuitive approach is to numerically compute integrals like the one of Eq. (13) by discretizing the configuration space and transforming them into the corresponding sums (penultimate term of Eq. (13)). For each *bin*, one computes the value of the observable and of the probability density in a reference point, and average over the bins. The cost of this approach is proportional to the total number of bins partitioning the 3N-dimensional configuration space. The total number of bins, n_b^{tot}, scales exponentially with the number of bins along each degree of freedom: $n_b^{tot} = n_b^{3N}$, assuming each degree of freedom is discretized in n_b bins. Imagine, for example, that one decomposes the configuration space along each degree of freedom in (just) 10 bins, with 1000 atoms, a very small sample these days, one has to evaluate 10^{3000} times the integrand of Eq. (13). Considering the shortest time

necessary to perform an arithmetic operation on a modern computer, $\sim 1/3\ 10^{-9}$ s, to accomplish the proposed task one needs $1/3\ 10^{2991}$ s, much longer than the age of Universe. Thus, the direct approach is impossible.

In 1953, Metropolis et al. [8] proposed a method circumventing the problem of the direct calculation of ensemble averages: Metropolis Monte Carlo.[1] This method consists of generating a Markov chain, a suitable series of configurations sampling asymptotically the probability density function of the given ensemble (the canonical ensemble, in this case). This suggested that any method generating a Markov chain sampling the ensemble of interest can be used to compute ensemble averages, and, among others, a trajectory obtained by integrating Newton's/Hamilton's (or any suitable) EoM: molecular dynamics is born! [10, 11]

In this section, we describe how to perform molecular dynamics. First, we focus on the case of Newton's/Hamilton's dynamics, which is suitable to sample the constant number of particles, constant volume and constant energy ensemble, NVE. Within this context, we discuss:

- how to numerically integrate the equations of motion (Eq. (8));
- what boundary conditions the system is subject to and how they are imposed;
- how to avoid the solution of the quantum electronic problem with empirical potentials, already alluded in Sect. 2 (see also Appendix A);

Next, we move to discuss how Newton's dynamics can be modified so as time averages along the trajectory is consistent with the average of other ensemble than NVE (Sect. 3.2). Next, we discuss how molecular dynamics is performed without resorting to empirical many-body potentials (Sect. 3.3).

In the following sections, we discuss special techniques to deal with the calculation of free energies (Sect. 4), so-called rare events (Sect. 5), processes governed by (free) energy barriers, and non-equilibrium problems (Sect. 6).

3.1 Microcanonical ensemble

Let us start from the case of a microcanonical system, requiring the sampling of the ensemble at constant number of particles, constant volume and constant energy. As we said at the beginning, we are mainly discussing condensed phase bulk systems, although with due caution the technique presented in this section can be extended to the case of surfaces (2D), wires (1D), *dots* and molecules (0D). We first briefly introduce all the ingredients of molecular dynamics, which we will discuss in detail in the following. First, we focus on an atomistic model of a system. Within the computational power of modern supercomputers, we can handle models consisting of up to 10^6-10^7 atoms in the case of so-called classical molecular dynamics, the one in which forces arise from empirical force fields mentioned in Sect. 2 and discussed in detail below.

Atoms in the range of 10^6-10^7 are very few with respect to the $\sim 10^{23}$ contained in bulk systems, e.g., those contained in a glass of water. This poses the problem of

[1] Readers interested in understanding the principles of Monte Carlo, a complementary technique to address the problems considered in this review, are encouraged to read Ref. [9]. More modern and comprehensive articles exist on Monte Carlo but the one suggested is rather clear on the general principles of the technique.

boundary conditions of the computational model. To understand this aspect, imagine putting two systems in vacuum, one made of the number of atoms in the range mentioned above and one made of a mole of atoms. Of course, the smaller one has a much higher surface/volume ratio. This implies that in the smaller sample atoms (molecules) at the surface, which have a different environment than the interior ones, can significantly affect the properties of the system. This effect is much smaller in the large system. Thus, a 10^6-10^7 atoms (molecules) computational systems in vacuum cannot accurately represent the properties of a macroscopic ($\sim 10^{23}$ atoms/molecules) one. This problem is typically solved using the so-called periodic boundary conditions, where the computational system is put in a *simulation box* of suitable size, so that the density is the same as the experimental one. This box is virtually repeated an infinite number of times in space. If the characteristic length of the interatomic forces is smaller than half the box size, there is no interaction between the atoms in the box and their periodic images. Thus, the effect of the periodic nature of the computational system on its characteristics, the fact that the computational sample is essentially a *supercrystal*, is negligible. Of course, this sample lacks almost any surface effect. Instead, in macroscopic systems, surface effects on the bulk properties are really negligible.

Once the general characteristics of the computational sample for modeling bulk systems are set, we focus on solving the EoM of a system of point particles, the nuclei, that interact *via* an effective potential depending on the position of the particles, $V(\mathbf{R})$. Here, it is convenient to consider the EoM expressed in Hamilton's form:

$$\dot{\mathbf{\Gamma}} = \{\mathbf{\Gamma}, \mathcal{H}\} = \left(\nabla_{\mathbf{p}}\mathcal{H} \cdot \nabla_{\mathbf{R}}\mathbf{\Gamma} - \nabla_{\mathbf{R}}\mathcal{H} \cdot \nabla_{\mathbf{p}}\mathbf{\Gamma}\right) = i\mathcal{L}\mathbf{\Gamma} \tag{21}$$

where $i\mathcal{L}$, the Liouvillian operator (multiplied by the imaginary number), is equal to $\left(M^{-1}\mathbf{p} \cdot \nabla_{\mathbf{R}} - \nabla_{\mathbf{R}}V(\mathbf{R}) \cdot \nabla_{\mathbf{p}}\right)$, with $\mathbf{p}$ the $3N$ vector of momenta, M the diagonal matrix of atomic masses, and $\mathcal{H} = K + V$, respectively kinetic energy and the interatomic potential among point particles representing the atoms in the classical picture of the system discussed above. The Liouville operator applied to the vector $\mathbf{\Gamma}$ must be understood as operating element by element, e.g., $\dot{\mathbf{\Gamma}}_i = i\mathcal{L}\mathbf{\Gamma}_i$, where $\mathbf{\Gamma}_i$ is the i-th element of the phase space vector $\mathbf{\Gamma}$. These equations of motion can be formally solved by

$$\mathbf{\Gamma}(t) = S(t)\mathbf{\Gamma}(0) \equiv \exp[it\mathcal{L}]\mathbf{\Gamma}(0). \tag{22}$$

where $S(t)$ is the time evolution operator of the system. Generally speaking Eq. (22) has not an explicit solution. Following Tuckerman et al. [12], exploiting the Trotter theorem [13], one proves that $\exp[it\mathcal{L}] = \lim_{n\to\infty}\{\exp[it/2n\mathcal{L}_2]\exp[it/n\mathcal{L}_1]\exp[it/2n\mathcal{L}_2]\}^n$ where $i\mathcal{L}_1 = \dot{\mathbf{R}} \cdot \nabla_{\mathbf{R}}$ and $i\mathcal{L}_2 = -\nabla_{\mathbf{R}}V(\mathbf{R}) \cdot \nabla_{\mathbf{p}}$. More in detail, for a finite value of n, $\exp[it\mathcal{L}] = \{\exp[it/2n\mathcal{L}_2]\exp[it/n\mathcal{L}_1]\exp[it/2n\mathcal{L}_2]\}^n + \mathcal{O}(t^3/n^3)$. Notice that another equivalent approximation of the time evolution operator exists where $i\mathcal{L}_1$ is applied externally and $i\mathcal{L}_2$ internally. However, the writing reported above, leading to an algorithm for integrating the EoMs equivalent to the so-called *velocity Verlet* algorithm, is computationally convenient. The advantage of the approach presented here with respect to the direct derivation of velocity Verlet, is that *integrators* based on time evolution operators written in terms of Liouvillians can be extended to EoMs

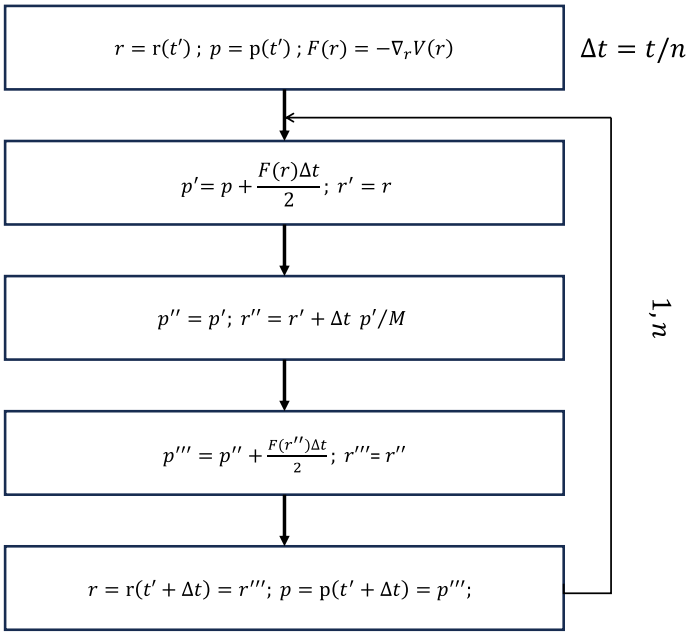

Fig. 1 Algorithm for the integration of the EoM according to the Liouvillian integrators. Notice that this algorithm corresponds to the so-called velocity Verlet

suitable to sample other ensembles (see below). Following Yoshida [14] and Suzuki [15], higher-order accuracy *decompositions* of the time evolution operator of Eq. (22) can be systematically developed. However, the corresponding integrator algorithms require to compute atomic forces $-\nabla_{\mathbf{R}} V(\mathbf{R})$ multiple times for each time step t/n, and one typically prefers to shorten the time step (increasing n).

It is important to stress a key characteristics of $\tilde{S}(t) = \{\exp[it/2n\mathcal{L}_2]\exp[it/n\mathcal{L}_1]\exp[it/2n\mathcal{L}_2]\}^n$: this is the exact time evolution operator for an unknown $\mathcal{H}'(\mathbf{\Gamma})$ Hamiltonian whose difference from the real Hamiltonian of the system is bound and can be reduced by reducing the time step (increasing the value of n). The proof of this statement is somehow involuted and is not provided here. This issue is further discussed in Ref. [16], pages 80–81. Thus, for suitable values of the time step, along molecular dynamics trajectories obtained using integrators derived from the time evolution operator $\tilde{S}(t)$, energy fluctuates but does not diverge. As a consequence, Liouvillian integrators produce trajectories properly sampling the microcanonical ensemble at the relevant energy with an accuracy that can be tuned by tuning the length of the time step.

$\tilde{S}(t)$ readily allows to derive a corresponding integrator algorithm. Notice that both $\exp[it/n\mathcal{L}_1]$ and $\exp[it/2n\mathcal{L}_2]$ have the same structure: $\exp[f(u)\nabla_v]$, where u and v are positions and momenta, or *vice versa*. The application of such an operator on a function $g(u, v)$ results in a shift of the v variable by $f(u)$: $\exp[f(u)\nabla_v]g(u, v) = g(u, v+f(u))$. Thus, the application of $\exp[it/n\mathcal{L}_1]$ or $\exp[it/2n\mathcal{L}_2]$ to $\mathbf{\Gamma}$ results into a shift of the **r** part in the first case, or of the **p** one in the second. The algorithm reported

in Fig. 1 corresponds to the velocity Verlet algorithm and is the typical symmetric algorithm implemented in most of the modern codes. Note that in the inner loop of the procedure, the force is computed only once as $F(r'')$ at time step t is identical to $F(r)$ at the subsequent time step $t + \Delta t$.

It is seen that the integration algorithm requires initial positions and momenta. This, indeed, is an obvious consequence of the EoM written in the Hamiltonian formalism, which are first-order differential equations in $\boldsymbol{R}$ and $\boldsymbol{p}$. Initial positions can be set in many different ways, e.g., placing the atoms or molecules at the position of one of the crystalline structures of the chemical species, or by randomly distributing them within the simulation box under the prescription that their distance is larger than some predetermined value, consistent with the typical distance in the given thermodynamic conditions (see, e.g., [17]). Initial momenta can be set from a Maxwellian distribution.

The last *ingredient* of molecular dynamics that is left to be discussed is the choice of the atomic forces $-\nabla_R V(\mathbf{R})$ and their calculation. In the previous section, it was explained that under the typical conditions of condensed matter, when the Born–Oppenheimer approximation holds, one can use an approximated form of the forces with an explicit dependence on the atomic positions. This avoids to perform a quantum mechanical ground state calculation at each time step of the integration algorithm of Fig. 1. (Conservative) atomic forces are obtained from the gradient of the potential $V(\mathbf{R})$. Generally speaking, this potential is *many-body*, i.e., the value of the potential depends on a function or sum of contribution of terms depending on the position of several atoms. This is computationally very expensive as the calculation of the potential scales with n-power of the number of atoms, where n is the order of the many-body potential. It turned out that in several cases, one can have a good representation of the interatomic interactions by pair additive potentials, i.e., $V(\mathbf{R})$ can be adequately expressed as sum of terms depending on the positions of two atoms, which makes the calculation to scale with the second power of the number of atoms.[2] Of course, pairwise additive potentials are not general and are unsuitable to describe many relevant systems, such as metals, for which the embedded atom model [21] is often used. Going beyond simple atomistic systems, i.e., considering a system made of molecules, typical forms of $V(\mathbf{R})$ consist of *bonded* and *non-bonded* terms, the former represents the covalently bonded atoms of molecules and the latter intra- and intermolecular terms between non-covalently bonded pairs. This representation is suitable for a broad class of systems, systems ranging from biological matter (e.g., proteins) to polymers to molecular liquids (e.g., water). These typically depend on (i) distances between pairs of atoms, modeling the stretching/compression of bonds, (ii) angles between three atoms, modeling bending, (iii) dihedral angles between four atoms, modeling rotations around *single* bonds, the process leading to conformational isomerization, and (iv) so-

[2] Smart approaches have been developed over the years to make the calculation of $V(\mathbf{R})$ faster. They are based on the observation that the pair interactions between non-covalently bonded atoms can but cut beyond some prescribed distance. Thus, at each time step, each atom *interacts* only with a subset of all atoms. Thus, at each time step, the number of pair force calculations does not scale as $\mathcal{O}(N^2)$ with the size of the system but only as $\mathcal{O}(Nm)$, with m the number of interacting neighbors. This observation led to the development, for example, of the so-called Verlet lists, which are still $\mathcal{O}(N^2)$ but with a very low *coefficient*, hence very efficient, or of the linked cells, where interacting pairs are sought after in cells of the size of the interaction length partitioning the simulations box. A detailed description of these methods and some more advanced techniques can be found in Refs. [16, 18–20].

called improper dihedrals, controlling the planarity of molecules like aromatic rings.

$$
\begin{aligned}
V_b(\mathbf{R}) = & \sum_{i=1}^{n_{\text{bonds}}} \frac{k_i^b}{2}(\Delta R_i - \Delta R_i^*)^2 \\
& + \sum_{i=1}^{n_{\text{angles}}} \frac{k_i^a}{2}(\theta_i - \theta_i^*)^2 \\
& + \sum_{i=1}^{n_{\text{dihedrals}}} k_i^d(1 + \cos(m\phi_i - \phi_i^*)) \\
& + \sum_{i=1}^{n_{\text{impropers}}} \frac{k_i^a}{2}(\xi_i - \xi_i^*)^2
\end{aligned}
\tag{23}
$$

where ΔR_i, θ_i, ϕ_i, and ξ_i are the values of the i-th bond length, angle, dihedral and improper dihedral, respectively. The corresponding symbols with asterisks represent the target value, e.g., ΔR_i^* is the equilibrium bond length of the i-th bond. The strength of each term, e.g., how much a given elongation of a bond from its equilibrium value causes an increase of energy, depends on the value of the corresponding force constant, k_i^b in the case of bond stretching/compression. The index i in the target values and force constants does not mean that one has to define a value for each specific bond. Typically, these values are defined for classes of bonds; For example, single, double and triple carbon–carbon bonds are characterized by decreasing bond lengths and increasing force constants along the series. Notice the difference between improper and *ordinary* dihedral terms: the first is associated to a harmonic potential, whose effect is to restore the equilibrium value of the angle; the second presents m equivalent minima separated by a corresponding number of equivalent maxima, which is the typical energy profile observed for the rotation around a single bond. This is insufficient, for example, to represent rotations around bonds in alkanes (C_nH_{2n+2}), were minima and maxima are not all equivalent. Alternative forms, such as the Ryckaert–Bellemans form [22], have been proposed.

Non-bonded interactions account for electrostatic and van der Waals interactions. Typically, electrostatic interactions are represented within the fixed point particle approximation, with charges at atomic positions. van der Waals interactions can be approximated by several representations, a very common one being the so-called Lennard–Jones pair potential [23]. Overall, the non-bonded interactions are typically expressed in the following form:

$$V_{nb}(\mathbf{R}) = \sum_{i>j} \frac{q_i q_j}{\Delta R_{ij}} + \sum_{i>j} 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{\Delta R_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{\Delta R_{ij}} \right)^{6} \right) \tag{24}$$

where ΔR_{ij} is the distance between atoms i and j, q_i and q_j are their charges, and ϵ_{ij} and σ_{ij} and the characteristic energy and length of the van der Waals interaction between the chemical species of atoms i and j as modeled in the Lennard–Jones potential. We reiterate that, same as in the case of bonded interactions, q_i and q_j, and ϵ_{ij} and σ_{ij} depend on the chemical species and the chemical environment, e.g., whether an oxygen atoms belongs to water or an ether molecules, rather than only on the specific atom. Thus, for example, all the oxygen atoms of a computational sample of water have the same charge and their Lennard–Jones parameters for the interaction with any other oxygen atom of the same sample are all the same. Given the high negative power dependence on interatomic distances, the Lennard–Jones potential is *short ranged*. Thus, though this function has not a compact support it can still accurately be approximated by its short-range analog obtained by cutting (and shifting to zero) the Lennard–Jones potential beyond some prescribed distance, typically for $\Delta R_{ij} > 2.5 - 3\sigma_{ij}$.

Non-bonded interactions with limited spatial range are typically named *short range*, to put in evidence the difference with other types of interactions, such as electrostatics, which cannot be cut out at a given distance. Indeed, electrostatic interactions require a special attention and the form reported in Eq. (24) will be revised and discussed more in detail in the following. Lennard–Jones contribution to the overall potential scales quadratically with the number of particles because, at variance with bonded interactions, where sets of interacting atoms remain the same all along the simulation, one has to identify which pairs of atoms fall within the $R_c = 2.5 - 3\sigma_{ij}$ cutoff distance at each time step. This, if one does not use any special technique referenced in the footnote above (see Refs. [18], [16, 19], [20]), requires the calculation of distances between all pairs of atoms in the simulation box and, for the short ranged one (link cell method), its 27 (in 3D) neighboring replica.

Atoms falling within the R_c from a given one must be searched taking into account the periodic boundary conditions. This is illustrated in Fig. 2, where the water molecule denoted *b'*, periodic image of the molecule *b*, is within the cutoff distance molecule *a*, while the molecule *b* within the computational box is not. More in general, a convention is adopted, the so-called minimum image convention, that only the pairs of molecules of minimum distance found either in the simulation box or within its periodic images are considered to test whether this interaction falls within or beyond R_c. This minimum image convention is implemented by the formula:

$$\left(\mathbf{R}_i - \mathbf{R}_j\right)' = \left(\mathbf{R}_i - \mathbf{R}_j\right) - H \left[H^{-1} \left(\mathbf{R}_i - \mathbf{R}_j\right) \right]_{nint} \tag{25}$$

where H is the 3×3 matrix of the vectors defining the simulation box, with each row containing respectively the x, y and z components of the vectors of the edges of the simulation box. To illustrate the effect of Eq. (25), consider the case of a cubic simulation box. In this case, H is a diagonal matrix whose diagonal elements are L,

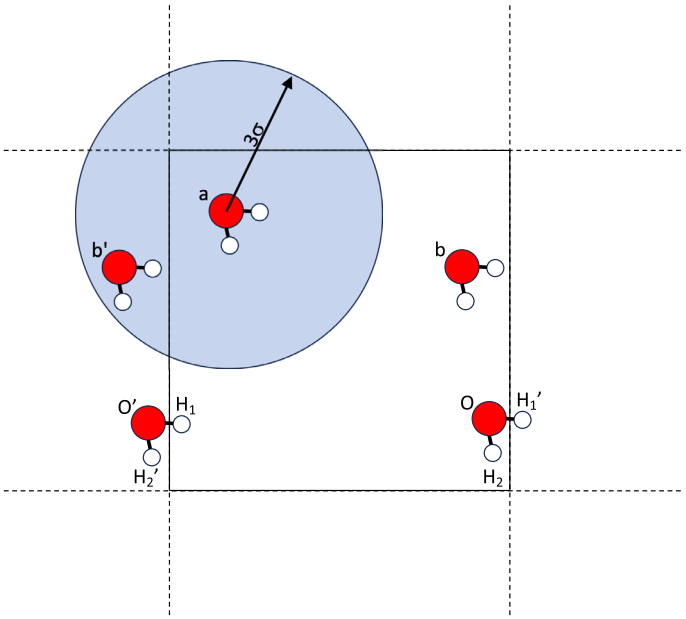

Fig. 2 Cartoon of a simulation box with periodic boundary conditions. The blue circle represents the cutoff distance for *short-range* non-bonded interactions

where L is the length of the edge of the cube. H^{-1} is the (diagonal) inverse of matrix H of elements $1/L$; thus, $H^{-1}\left(\mathbf{R}_i - \mathbf{R}_j\right)$ is the vector distance between atoms i and j in *reduced coordinates*. $[\cdot]_{nint}$ is the nearest neighbor operation. Thus, when a component of $H^{-1}\left(\mathbf{R}_i - \mathbf{R}_j\right)$ is larger than 0.5 or smaller than -0.5, along the corresponding simulation box direction a corresponding length is subtracted or added, respectively, to the given component of the vector distance. For example, if $H^{-1}\left(\mathbf{R}_i - \mathbf{R}_j\right) = 0.8$, its nearest integer value is 1 and, according to Eq. (25) L is subtracted from the corresponding element of the vector $\mathbf{R}_i - \mathbf{R}_j$, returning the value corresponding to the closest pair among all periodic images. The minimum image convention holds for bonded interaction as well. Once more, this is illustrated in Fig. 2, where clearly atoms O and H_1 of the water molecule in the bottom right corner are not in a bonded configuration, while e.g., O and H_1', the latter being the periodic image of the named hydrogen atom, are.

Let us now come back to electrostatic interactions. Since this is long range, in a periodic system it must more properly be written as:

$$V_{\mathrm{el}}(\mathbf{R}) = \frac{1}{2}\sum_{\mathbf{l}}\sum_{i,j}{}' \frac{q_i q_j}{|\Delta\mathbf{R}_{ij'} + H\mathbf{l}|} \tag{26}$$

where $|\Delta\mathbf{R}_{ij} + H\mathbf{l}|$ is the distance between atoms i and j, including all periodic images, which is taken into account by the term $H\mathbf{l}$, with $\mathbf{l}$ a vector of (positive and negative) integer numbers. The prime on the sum symbol remarks that from the sum

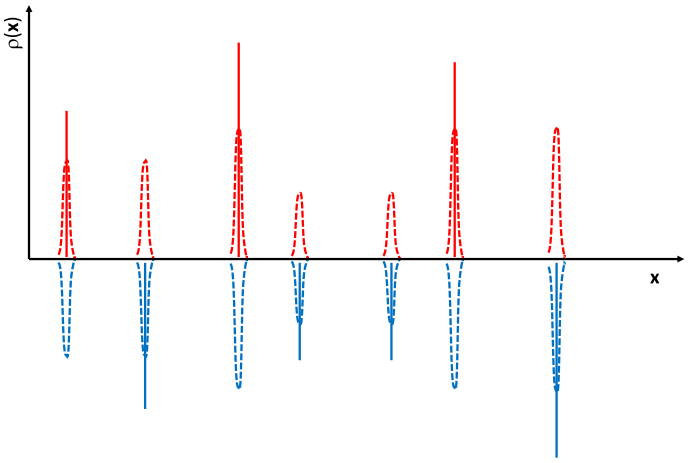

Fig. 3 Physical (vertical bar representing a delta-like distribution) and additional (Gaussian) charge density

over j it is excluded the atom i when $\mathbf{l} = \mathbf{0}$, i.e., atoms can interact with their periodic images but not with themselves. Notice the $1/2$ in Eq. (26), introduced to avoid double counting. When the total charge of the system is zero the series in Eq. (26) converges, but only conditionally. Hence, one cannot simply truncate the series for its evaluation. To address this problem, a technique due to Ewald [24], who developed it to compute the electrostatic energy in ionic solids, is used.[3] The starting point is recognizing that Eq. (26) can be written as $V_{\mathrm{el}}(\mathbf{R}) = \sum_i q_i \phi_i^{\mathrm{el}}(\mathbf{R}_i)$, where $\phi_i^{\mathrm{el}}(\mathbf{x}) = 1/2 \sum_{\mathbf{l}} \sum_j q_j/|\mathbf{x} - \mathbf{R}_j + H\mathbf{l}| - q_i/2|\mathbf{x} - \mathbf{R}_i|$ is the electrostatic potential generated by all atoms and their periodic images on atom i. This writing, which is slightly different from the one of Eq. (26), is obtained by adding and subtracting the self-interaction term. Since there are now no more constraints on the values of j in the definition of $\phi_i^{\mathrm{el}}(\mathbf{x})$. The first term in $\phi_i^{\mathrm{el}}(\mathbf{x})$ is the electrostatic potential associated to a delta-like charge density: $\rho^p(\mathbf{x}) = \sum_{\mathbf{l}} \sum_{\mathbf{j}} \mathbf{q_j} \delta(\mathbf{x} - \mathbf{R_j} + \mathbf{Hl})$. The fundamental idea of the Ewald sum is to add and subtract a *smooth* charge density with maxima/minima in correspondence of the particles' position: $\rho^p(\mathbf{x}) = \rho^{\mathbf{p}}(\mathbf{x}) - \rho^{\mathbf{s}}(\mathbf{x}) + \rho^{\mathbf{s}}(\mathbf{x})$, where $\rho^s(\mathbf{x}) = \sum_{\mathbf{l}} \sum_{\mathbf{j}} \mathbf{q_j} \mathbf{G}(\mathbf{x} - \mathbf{R_j} + \mathbf{Hl}; \mathbf{s})$ is a sum of smooth of Gaussian-like charge densities (see Fig. 3), with s standard deviation of the Gaussians. The operation of adding and subtracting ρ^s does not change the overall charge density, hence the electrostatic potential is not affected by this operation. Let us introduce $\phi_{\mathrm{short}}(\mathbf{x})$ and $\phi_{\mathrm{long}}(\mathbf{x})$, short and long range electrostatic potentials associated to $\rho^p(\mathbf{x}) - \rho^{\mathbf{s}}(\mathbf{x})$ and $\rho^s(\mathbf{x})$, respectively. The origin of the denominations *short* and *long* will be clarified shortly. One understands that at a distance d from an atom larger than $3s$, three times the standard deviation of the Gaussian compensating charge (see Fig. 3), the charge corresponding to the density $\rho^p(\mathbf{x}) - \rho^{\mathbf{s}}(\mathbf{x})$ enclosed by a sphere of radius d is negligible. According to the Gauss theorem, the electric field $\mathbf{E}$ is proportional to the charge q_d enclosed within the sphere of radius d, $\mathbf{E}(d) \propto q_d$. Thus, at a distance longer than

[3] Other techniques, e.g., the particle–particle/particle–mesh approach [25], inspired to the Ewald sum were developed over the years to further improve the efficiency in the calculation of electrostatic interactions.

$3s$ $\mathbf{E} \sim 0$. This suggests that only atoms close to $\mathbf{R}_i$ contribute to $\phi_{\text{short}}(\mathbf{R}_i)$, which also explains the attribute *short range*. ϕ_{short} can be computed by a direct sum on the point and Gaussian compensating charges associated to particles laying within a prescribed cutoff radius from $\mathbf{R}_i$ considering the minimum image convention, similar to the approach used for the Lennard–Jones potential (Fig. 2). $\phi_{\text{long}}(\mathbf{x})$ is obtained by solving the corresponding Poisson equation, $\nabla^2\phi_{\text{long}}(\mathbf{x}) = -\rho^s(\mathbf{x})/\epsilon$, in the Fourier space, which, for a neutral system reads:

$$\phi_{\text{long}}(\mathbf{x}) = \frac{1}{\Omega}\sum_{\mathbf{k}\neq 0}\sum_j \frac{q_j}{k^2}\exp[i\mathbf{k}\cdot(\mathbf{x}-\mathbf{R}_j)\exp[-s^2k^2/2]] \tag{27}$$

where $\mathbf{k}$ is a vector of the lattice of the reciprocal space of the simulation box, k^2 is its absolute square, and Ω is the volume of the simulation box. Contributions to $\phi_{\text{long}}(\mathbf{x})$ decrease with the inverses of k^2 and one can cut them off after a threshold, which depends on how wide is the reciprocal space Gaussian $\exp[-s^2k^2/2]]$. Notice that the standard deviation of a Gaussian in the Fourier space is proportional to the inverse of its value in the real space. Thus, a suitable choice of the value of s, neither too big to keep the ϕ_{short} short range, nor too small, to make the calculation of ϕ_{long} in the Fourier space fast, is necessary for an efficient algorithm to compute the electrostatic potential energy $V_{\text{el}}(\mathbf{R})$.

Systems and problems exist where covalent bonds are formed and broken along the dynamics. This is the case, for example, of liquid or amorphous silicon, where atoms can have a different number of nearest neighbors, and this may change along the trajectory. This case cannot be represented by potentials of the kind mentioned above. Of course, one can perform ab initio simulations (Sect. 3.3), where one solve the quantum problem of electrons together with classical nuclei. This is computationally very expensive. For selected cases, one can still use parametric potentials. Many examples of these potentials have been proposed, the most extensively used probably being the the Stillinger–Weber [26] and Tersoff [27] potentials. Some of them, like the Tersoff potential [27], take into account the present bonding environment of atoms in the system (see Sect. A for more modern approaches to this problem). Just to illustrate the idea, in the Tersoff potential a pair interaction potential is typically written as

$$V(\boldsymbol{R}_{ij}) = V^{\text{rep}}(\boldsymbol{R}_{ij}) + b\,V^{\text{att}}(\boldsymbol{R}_{ij}) \tag{28}$$

where the pair interaction potential $V(\boldsymbol{R}_{ij})$ is the sum of a repulsive, $V^{\text{rep}}(\boldsymbol{R}_{ij})$, and an attractive, $V^{\text{att}}(\boldsymbol{R}_{ij})$ term, the latter being multiplied by the bond order *weight* b depending on the chemical environment of the atoms involved in the bonding, e.g., the number of atoms within some cutoff distance from atoms i and j, the angle formed among these atoms, etc. Thus, the strength of forces acting on the atoms depends on the number of *bonds* dynamically formed/broken.

Let us conclude this section shortly discussing the case of systems subjected to holonomic constraints. Here, we refer to constraints like fixed values of interatomic distances or bond angles. In Sects. 4.1, 5.1, and 5.2, we will consider more general cases. The reason of introducing constraints in molecular dynamics is that the char-

acteristic times of the movement of internal degrees of freedom of molecules might be very different, e.g., dihedral angles modes are typically much slower than bond stretching. Since the length of the timestep Δt (Fig. 1) is determined by the shorter characteristic times, fast modes imply short timesteps. Thus, imposing constraints on fast modes, which keep them out from the list of movements to be integrated with a suitable timestep, allows one to use longer Δt, implying that one needs a smaller number of molecular dynamics timesteps to simulate a trajectory of a prescribed duration. A holonomic constraint is represented by the relation $\sigma(\boldsymbol{R}) = z$. The equations of motion of a system subjected to a (or more) constraint(s) can be obtained by the method of the Lagrange multipliers, consisting of adding the $\lambda(\sigma(\boldsymbol{R}) - z)$ term to the Lagrangian/Hamiltonian function from which EoM are derived, where λ is the [28] Lagrange multiplier. Of course, on a system may (and typically do) act multiple constraints. In this case, one adds multiple terms of the previous kind: $\sum_{i=1}^{N_{\text{constr}}} \lambda_i (\sigma_i(\boldsymbol{R}) - z_i)$, resulting in the corresponding force terms $-\sum_{i=1}^{N_{\text{constr}}} \lambda_i (\boldsymbol{\nabla}_{\boldsymbol{R}} \sigma_i(\boldsymbol{R}))$. The values of λs are fixed by the condition that the constraints are respected at all time. Ryckaert et al. [28] recognized that even if the exact value of λs were known, since integrating algorithms are not exact, constraints would be not satisfied exactly and errors would accumulate along dynamics, bringing to a departure from the constrained value. An alternative approach has been proposed by Ryckaert and coworkers, the SHAKE algorithm [28], in which EoM are first solved without imposing the constraints, and then the *exact* values of the Lagrange multipliers consistent with the accuracy of the integration algorithm is obtained requiring that λs' values are such that the constraints at the next timestep are satisfied.

Let's call the generally vectorized equation corresponding to this condition for the Lagrange multipliers, the SHAKE equation:

$$\sigma(\mathbf{R}(\mathbf{t} + \Delta\mathbf{t}; \lambda) - \mathbf{z} = \mathbf{0} \tag{29}$$

Many different algorithms can be introduced to solve it. The first [28] precedes iteratively solving successively one scalar element of (29) at time; when all constraints have been considered, one iterates until convergence. In case of non-convergence, one has to reduce the integration step. Several extensions to the original method have been introduced. Andersen [29] extended the original algorithm, developed for the Verlet integrator, to the velocity Verlet case of Fig. 1. Elber and coworkers [30, 31] developed parallel versions of the SHAKE algorithm, which are better suited for massively parallel machines. An additional version was developed efficiently running on hybrid CPU/GPU machines [32].

3.2 Canonical, constant-pressure constant-temperature and other ensembles

In Sect. 3.1, we discussed MD driven by the Newton's EoM, which sample the microcanonical ensemble. Here, we discuss how one can modify these EoM so that the associated MD samples, for example, the canonical ensemble. The theoretical framework within which this extension of Newton's dynamics is developed is the statistical treatment of the so-called non-Hamiltonian dynamics [33]. The key obser-

vation of this framework is the *equal probability density* of the phase space points satisfying the conservation laws of the given EoM. In the case of Newton's dynamics, energy is conserved, $H(\mathbf{\Gamma}) = E$, and the probability density function reads $\rho(\mathbf{\Gamma}; E) = \delta(H(\mathbf{\Gamma}) - E)/\int \mathrm{d}\mathbf{\Gamma}\, \delta(H(\mathbf{\Gamma}) - E)$, where E is the (conserved) energy of the system. The intuitive meaning of $\rho(\mathbf{\Gamma}; E)$ is that (i) one has zero probability (density) to be in a phase space point inconsistent with the conservation law $H(\mathbf{\Gamma}) = E$ and (ii) the same probability (density) among the phase space points consistent with the constraint. The phase space associated to the Newton's EoM is equipped with the invariant measure $\mathrm{d}\mathbf{\Gamma}$. Hence, the average value of the generic observable can be computed as $\langle B \rangle_E = \int \mathrm{d}\mathbf{\Gamma}\, B(\mathbf{\Gamma})\rho(\mathbf{\Gamma}; E)$

Consider an extended phase space $\boldsymbol{\chi} = (\mathbf{\Gamma}, \boldsymbol{\eta}, \boldsymbol{\pi}_{\boldsymbol{\eta}})$, in which $\mathbf{\Gamma}$ may be augmented by other variables, position-like variables $\boldsymbol{\eta}$ and associated momenta $\boldsymbol{\pi}_{\boldsymbol{\eta}}$. For example, anticipating the discussion reported in the following, in constant temperature simulations, one extends the particles' phase space with degrees of freedom representing a thermostat, allowing energy to fluctuate. Let us generally represent these EoM in a form analogous to the Hamiltonian form:

$$\dot{\boldsymbol{\chi}} = \mathbf{g}(\boldsymbol{\chi}) \tag{30}$$

where $\mathbf{g}(\boldsymbol{\chi})$ is a suitable (vectorial) function. In the Hamiltonian form, Newton's EoM read:

$$\begin{cases} \dot{\mathbf{R}} = \mathbf{g}_{\mathbf{R}}(\boldsymbol{p}) = \mathbf{p}/M \\ \dot{\mathbf{p}} = \mathbf{g}_{\mathbf{p}}(\boldsymbol{R}) = -\nabla V(\mathbf{R}) \end{cases} \tag{31}$$

where $\mathbf{p}/M$ is the $3N$ vector of the velocities of the particles and $-\nabla V(\mathbf{R})$ is the $3N$ vector of forces. While Hamilton's dynamics conserves energy (and, depending on the chosen boundary conditions total linear and angular momenta), a generic set of EoM can have other conservation laws, generally denoted by the set $\{A_i(\boldsymbol{\chi}) = z_i\}_{i=1,m}$, where $A_i(\boldsymbol{\chi})$ is the ith conserved observable and z_i is the corresponding (conserved) value in the realization of the dynamics, and m is the number of conservation laws. In EoM where the set $\{A_i(\boldsymbol{\chi}) = z_i\}_{i=1,m}$ of constraints is satisfied, the trajectory moves on the intersect of the hypersurfaces $\{A_i(\boldsymbol{\chi}) = z_i\}_{i=1,m}$. The associated probability density function $\rho(\boldsymbol{\chi}; \{z_i\}_{i=1,m})$ can be considered the probability density function of a *microcanonical ensemble* in the extended phase space $\boldsymbol{\chi}$. However, a key difference from Hamiltonian systems is that with generic EoM $\mathrm{d}\boldsymbol{\chi}$ may not be an invariant measure, i.e., the phase space may compress or expand along the trajectory: $\mathrm{d}\boldsymbol{\chi}(t) \neq \mathrm{d}\boldsymbol{\chi}(t')$. The invariant measure can be obtained from the Jacobian, $J\big(\boldsymbol{\chi}(t), \boldsymbol{\chi}(t_0)\big)$, of the transformation between the extended phase space variables at different times, noticing that $\mathrm{d}\boldsymbol{\chi}(t) = J\big(\boldsymbol{\chi}(t), \boldsymbol{\chi}(t_0)\big)\, \mathrm{d}\boldsymbol{\chi}(t_0)$. Developing this argument, Tuckerman et al. [33, 34] have shown that a general invariant measure exists, $\sqrt{g(\boldsymbol{\chi}(t), t)}\mathrm{d}\boldsymbol{\chi}(t)$, $\sqrt{g(\boldsymbol{\chi}(t), t)}$ being related to the *compressibility* of the phase space, $\kappa(\boldsymbol{\chi}(t), t) = \nabla_{\boldsymbol{\chi}(t)}\dot{\boldsymbol{\chi}}(t)$: $\sqrt{g(\boldsymbol{\chi}(t), t)} = \big(\exp[\int \mathrm{d}s\, \kappa(\boldsymbol{\chi}(s), s)]\big)^{1/2}$. In these systems, the average value of a (microscopic) observable $B(\mathbf{\Gamma})$ (remember that $\boldsymbol{\chi} = (\mathbf{\Gamma}, \boldsymbol{\eta}, \boldsymbol{\pi}_{\boldsymbol{\eta}})$ and so $\mathbf{\Gamma}$ refers to the phase space (physical) variables) is:

$$\langle B\rangle_{\{z_i\}_{i=1,m}} = \int \sqrt{g(\boldsymbol{\chi},t)}\,\mathrm{d}\boldsymbol{\chi}\; B(\boldsymbol{\Gamma})\rho(\boldsymbol{\chi};\{z_i\}_{i=1,m}) \tag{32}$$

Consider Eq. (32), consider performing first the integration over the non-physical subspace of variables $\boldsymbol{\eta}$ and $\boldsymbol{\pi}_{\boldsymbol{\eta}}$ of the extended phase space $\boldsymbol{\chi} = (\boldsymbol{\Gamma}, \boldsymbol{\eta}, \boldsymbol{\pi}_{\boldsymbol{\eta}})$:

$$\begin{aligned}\langle B\rangle_{\{z_i\}_{i=1,m}} &= \int \mathrm{d}\boldsymbol{\Gamma}\; B(\boldsymbol{\Gamma}) \int \mathrm{d}\boldsymbol{\eta}\,\mathrm{d}\boldsymbol{\pi}_{\boldsymbol{\eta}}\,\sqrt{g(\boldsymbol{\chi},t)}\,\rho(\boldsymbol{\chi};\{z_i\}_{i=1,m}) \\ &= \int \mathrm{d}\boldsymbol{\Gamma}\; B(\boldsymbol{\Gamma})\rho(\boldsymbol{\Gamma})\end{aligned} \tag{33}$$

where

$$\rho(\boldsymbol{\Gamma}) = \int \mathrm{d}\boldsymbol{\eta}\,\mathrm{d}\boldsymbol{\pi}_{\boldsymbol{\eta}}\,\sqrt{g(\boldsymbol{\chi},t)}\,\rho(\boldsymbol{\chi};\{z_i\}_{i=1,m}) \tag{34}$$

is the marginal probability density on the physical phase subspace of the overall probability density function sampled by the (extended) EoM. Thus, while the probability density of the entire space is *microcanonical*, $\rho(\boldsymbol{\chi};\{z_i\}_{i=1,m}) = \prod_{i=1,m}\delta(A_i(\boldsymbol{\chi})-z_i)$, the reduced or marginal probability density may be, e.g., canonical (constant number of particles, constant volume and constant temperature). Indeed, as we will show in the following, this allows to derive extended non-Hamiltonian EoM sampling for the physical subset non-microcanonical ensembles. In this way, one can address the problem of computing ensemble averages in conditions consistent with experiments.

The statistical mechanics of non-Hamiltonian systems summarized above allows one to test the ensemble sampled by a set of EoM. The analysis proceeds as follows:

1. One must eliminate linearly dependent variables, i.e., those observables obeying rules like $\chi_i(t) = C\,\chi_j(t)$, where C is a constant. Of course χ_i is not an independent degree of freedom of the dynamical system.
2. One must also exclude driven, trivial or uncoupled variables. These are variables χ_js satisfying EoM like $\dot{\chi}_i(t) = g_i(\chi_i)$ and $\dot{\chi}_j(t) = g_j(\chi_i,\chi_j)$, with $\chi_i(t)$ a variable of primary importance. Moreover, conservation laws must not mix χ_i and χ_j—$A_\alpha(\chi_i) = z_\alpha$ and $A_\beta(\chi_j) = z_\beta$—and χ_is are variables of primary importance, e.g., physical degrees or freedom, elements of $\boldsymbol{\Gamma}$. If these conditions are met, (i) χ_js do not affect the dynamics (hence statistics) of χ_is, and (ii) the decoupling of χ_js and χ_is implies that the probability density function of the latter is not affected by the former.
3. Identify all the conservation laws satisfied by the EoM, $\dot{A}_i(\boldsymbol{\chi}) = \nabla_{\boldsymbol{\chi}} A_i(\boldsymbol{\chi})\cdot\dot{\boldsymbol{\chi}} = 0$, to give to the probability density function $\rho(\boldsymbol{\chi})$ the form $\prod_{i=1,m}\delta(A_i(\boldsymbol{\chi}) - z_i)$.
4. Based on the EoM, one can now compute the metric determinant factor $\sqrt{g(\boldsymbol{\chi})}$ relative to the relevant variables remaining after the elimination steps of points 2 and 3.
5. Following Eq. (34) and using the metric determinant factor $\sqrt{g(\boldsymbol{\chi})}$ computed in point 4, one can now integrate over the non-physical degrees of freedom. The resulting marginal or reduced probability density function represents the ensemble that is sampled by the dynamics generated by the given EoM.

To illustrate both the analysis outlined above and how it can be used to prove what ensemble is sampled by a set of EoM, let us apply it to the so-called Nosé–Hoover chain EoM. [35]

$$\begin{cases} \dot{\mathbf{R}} &= \mathbf{p}/M \\ \dot{\mathbf{p}} &= -\nabla V(\mathbf{R}) - p_{\eta_1}/Q_1\,\mathbf{p} \\ \dot{\eta}_i &= p_{\eta_i}/Q_i \\ \dot{p}_{\eta_1} &= \left[2\mathcal{K}(\mathbf{p}) - dNk_{\mathrm{B}}T\right] - p_{\eta_1}\,p_{\eta_2}/Q_2 \\ \dot{p}_{\eta_i} &= \left[p_{\eta_{i-1}}/Q_{i-1} - k_{\mathrm{B}}T\right] - p_{\eta_i}\,p_{\eta_{i+1}}/Q_{i+1} \\ \dot{p}_{\eta_M} &= \left[p_{\eta_{M-1}}/Q_{M-1} - k_{\mathrm{B}}T\right] \end{cases} \tag{35}$$

where d is the dimensionality of the space, e.g., $d = 3$ in the ordinary 3D space. This extended dynamical system consists of the phase space $\boldsymbol{\Gamma} = (\mathbf{R}, \mathbf{p})$ and the additional degrees of freedom $\left\{\eta_i, p_{\eta_i}\right\}_{i=1,M}$. The EoM of the physical system are augmented of a term connecting physical momenta $\mathbf{p}$ with the momentum p_{η_1} associated to η_1. In the following, we will show that the EoM of Eq. (35) sample in the reduced $\boldsymbol{\Gamma}$ space the canonical ensemble. Hence, here and in the following, we dub p_{η_i} and η_i *thermostat variables*. In a dynamical system, each pair of position-like/momentum-like variables must have an inertia associated; Q_i is this inertia of the thermostat variables. One notices that the EoM of physical momenta looks like the equations of a system with friction. However, the friction parameter p_{η_1}/Q_1 is dynamical, changes along the dynamics depending on the value of p_{η_1} and is not constrained to be positive. Thus, physical momenta can be dragged or pushed depending on the sign of the thermostat momentum. To better understand the effect of thermostat variables on the physical subsystem, imagine that at time t_0 $p_{\eta_1} = 0$ and that at this time, the kinetic energy of the physical system $\mathcal{K}(\mathbf{p})$ is lower than the target thermal energy $Nk_{\mathrm{B}}T$. The force-like term acting on p_{η_1} drags p_{η_1} to negative values, which makes the dynamical friction coefficient $p_{\eta_1}/Q_1 < 0$. Thus, the friction-like term in the EoM pushes the physical momenta toward an increase of the kinetic energy, i.e., toward lower absolute values of $2\mathcal{K}(\mathbf{p}) - dNk_{\mathrm{B}}T$. Of course, if $2\mathcal{K}(\mathbf{p}) > Nk_{\mathrm{B}}T$ the effect is the opposite, the dynamical friction coefficient will be positive and it will act as a drag, by reducing the kinetic energy of the particles. In both cases, the thermostat will act as a feedback at the target value of the kinetic energy. Notice that a similar argument may be developed for the EoM of the thermostat momenta apart last one: mathematically, the EoM of $\left\{p_{\eta_i}\right\}_{i=1,m-1}$ have a structure analogous to the one of physical momenta. Thus, any thermostat momentum but last one is thermostatted. This is the origin of the term *chain* in the name of the method, as opposed to the Nosé–Hoover method, which had only one pair thermostat variables.

The above EoM are non-Hamiltonian, i.e., they cannot be derived from Hamilton-like equations $\dot{\mathbf{x}} = \nabla_{\mathbf{p}_x}\mathcal{H}(\mathbf{x}, \mathbf{p}_x)$ and $\dot{\mathbf{p}_x} = -\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}, \mathbf{p}_x)$, with $\mathbf{x}$ and $\mathbf{p}_x$ generalized coordinates and momenta. Using EoM (35), one can prove that the following quantity is conserved along the dynamics:

$$\mathcal{H}'(\mathbf{R}, \mathbf{p}, \mathbf{p}_\eta, \boldsymbol{\eta}) = \mathcal{H}(\mathbf{R}, \mathbf{p}) + \sum_{k=1,M} p_{\eta_k}^2/Q_k + dNk_{\mathrm{B}}T\,\eta_1 + k_{\mathrm{B}}T \sum_{k=2,M} \eta_k \tag{36}$$

where $\mathcal{H}(\mathbf{R}, \mathbf{p})$ is the *conventional* Hamiltonian of the physical subsystem, the sum of the kinetic and potential energy of the particles. One notices that $\mathbf{R}$, $\mathbf{p}$ and $\mathbf{p}_\eta$ are genuine variables (neither driven nor trivial—see above). Concerning the ηs, only η_1 and $\eta_{\mathrm{CoM}} = \sum_{k=2,M} \eta_k$ enter the analysis via $\mathcal{H}'(\mathbf{R}, \mathbf{p}, \mathbf{p}_\eta, \boldsymbol{\eta})$. The index CoM of η_{CoM} recalls that this variable looks like the *center of mass* of the thermostat position-like coordinates, if one takes out from these the one directly coupled to the physical system. The compressibility of the extended phase space in the relevant variable is $\kappa = \nabla_{\mathbf{x}}\dot{\mathbf{x}} = \nabla_{\mathbf{R}}\dot{\mathbf{R}} + \nabla_{\mathbf{p}}\dot{\mathbf{p}} + \nabla_{\eta_1}\dot{\eta}_1 + \nabla_{\eta_{\mathrm{CoM}}}\dot{\eta}_{\mathrm{CoM}} = -dN\dot{\eta}_1 + \dot{\eta}_{\mathrm{CoM}}$. Hence, the metric determinant factor is $\sqrt{g(\mathbf{x}, t)} = \exp[(dN - (d-1))\eta_1 + \eta_{\mathrm{CoM}}]$.

If there are no external forces acting on the physical system, there is another (vectorial) conserved quantity $\mathbf{p}_{\mathrm{CoM}} \exp[\eta_1] = \mathbf{z}_{\mathbf{p}_{\mathrm{CoM}}}$, where $\mathbf{p}_{\mathrm{CoM}}$ is the momentum of the center of mass and $\mathbf{z}_{\mathbf{p}_{\mathrm{CoM}}}$ is a vector of constants. This conservation law can be rewritten as $\exp[\eta_1] = z_{p_{\mathrm{CoM},i}}/p_{\mathrm{CoM},i} = \sqrt{z^2_{p_{\mathrm{CoM},1}} + z^2_{p_{\mathrm{CoM},2}} + z^2_{p_{\mathrm{CoM},3}}}/p_{\mathrm{CoM}}$, where the index i denotes one of the components of the momentum/constant vector, and p_{CoM} its magnitude of $\mathbf{p}_{\mathrm{CoM}}$. A consequence is that there is a linear dependence among the components of the momentum, or between a component and it magnitude: $p_{\mathrm{CoM},i} = z_{p_{\mathrm{CoM},i}}/z_{p_{\mathrm{CoM},j}}\, p_{\mathrm{CoM},j} = z_{p_{\mathrm{CoM},i}}/\sqrt{z^2_{p_{\mathrm{CoM},1}} + z^2_{p_{\mathrm{CoM},2}} + z^2_{p_{\mathrm{CoM},3}}}\, p_{\mathrm{CoM}}$. Hence, two components of the momentum of the center of mass are linearly dependent on the third or all of them depend on the magnitude of the vector. As a further consequence, momenta and positions can be redefined as relative to the center of mass value, $\mathbf{R}' = \mathbf{R} - \mathbf{R}_{\mathrm{CoM}}$ $\mathbf{p}' = \mathbf{p} - \mathbf{p}_{\mathrm{CoM}}$.

One can now compute the probability density function sampled by the Nosé–Hoover chain EoM as follows:

$$\begin{aligned}\rho(\boldsymbol{\Gamma}) = \int \mathrm{d}\eta_1\, \mathrm{d}\eta_{\mathrm{CoM}}\, \mathrm{d}p_\eta\, e^{[(dN-(d-1))\eta_1+\eta_{\mathrm{CoM}}]} \\ \times\delta\left(\mathcal{H}'(\mathbf{R}', \mathbf{p}', p_{\mathrm{CoM}}, \mathbf{p}_\eta, \boldsymbol{\eta}) - z_{\mathcal{H}'}\right)\delta\left(p_{\mathrm{CoM}}\, e^{\eta_1} - z_{p_{\mathrm{CoM}}}\right).\end{aligned} \tag{37}$$

Here, $z_{\mathcal{H}'}$ and $z_{p_{\mathrm{CoM}}}$ are the values of the conserved quantities $\mathcal{H}'(\mathbf{R}, \mathbf{p}, \mathbf{p}_\eta, \boldsymbol{\eta})$ and $p_{\mathrm{CoM}} \exp[\eta_1]$. The integral over η_1 is carried out first, using $\delta(p_{\mathrm{CoM}} \exp[\eta_1] - z_{p_{\mathrm{CoM}}})$ to replace the integration variable where needed:

$$\begin{aligned}\rho(\boldsymbol{\Gamma}) = \int \mathrm{d}\eta_{\mathrm{CoM}}\, \mathrm{d}p_\eta \left(\frac{z_{p_{\mathrm{CoM}}}}{p_{\mathrm{CoM}}}\right)^{dN-(d-1)} e^{\eta_{\mathrm{CoM}}}\delta\left(\mathcal{H}(\mathbf{R}', \mathbf{p}', p_{\mathrm{CoM}}) + \sum_{k=1,M} p^2_{\eta_k}/Q_k\right. \\ \left. + dNk_{\mathrm{B}}T \ln\left(\frac{z_{p_{\mathrm{CoM}}}}{p_{\mathrm{CoM}}}\right) + k_{\mathrm{B}}T\eta_{\mathrm{CoM}} - z_{\mathcal{H}'}\right).\end{aligned} \tag{38}$$

Next we integrate over $d\eta_{\mathrm{CoM}}$ taking advantage of the remaining delta function:

$$\begin{aligned}\rho(\boldsymbol{\Gamma}) &= e^{-\mathcal{H}(\mathbf{R}', \mathbf{p}', p_{\mathrm{CoM}})/k_{\mathrm{B}}T}\left(\frac{z_{p_{\mathrm{CoM}}}}{p_{\mathrm{CoM}}}\right)^{-(d-1)} \int \mathrm{d}p_\eta\, e^{\left(z_{\mathcal{H}'} - \sum_{k=1,M} p^2_{\eta_k}/Q_k\right)} \\ &\propto e^{-\mathcal{H}(\mathbf{R}', \mathbf{p}', p_{\mathrm{CoM}})/k_{\mathrm{B}}T}\,(p_{\mathrm{CoM}})^{d-1}.\end{aligned} \tag{39}$$

Thus, the (physical) phase space probability density function $\rho(\mathbf{\Gamma})$ sampled along the Nosé–Hoover chain MD is the canonical distribution.

Often one refers to molecular dynamics driven by the Nosé–Hoover chain EoM (35) as *NVT-MD*. Notice that what one means is just that this MD samples the canonical distribution, not that particles follow the same dynamics as that of atoms in a sample attached a thermostat.

To integrate the Nosé–Hoover chain EoM (35), one can use the Liouvillian integrators approach discussed in Sect. 3.1.

An analysis similar to the one reported above for the Nosé–Hoover chain EoM can be performed, e.g., the Martyna–Tobias–Klein EoM [36], showing that it samples the constant number of particles, constant pressure, constant temperature (NPT) ensemble. The procedure illustrated above can be used to identify the ensemble corresponding to any given set of EoMs.

Above we illustrated an analysis for determining what ensemble is sampled by a (extended) deterministic molecular dynamics. Deterministic molecular dynamics is not the only possible approach to sample ensemble beyond the NVE one. In the above, we have already mentioned Monte Carlo. However, there are also stochastic approaches rooted within molecular dynamics. Here, we refrain to further extend our analysis to these approaches. We also refrain from providing a complete literature on these complementary approaches as this is against the spirit of this article, analyzing in detail a limited number of approaches. Thus, we suggest to interested readers a possible starting point for their further reading. In particular, we suggest Ref. [1] for Monte Carlo and Ref. [37] for stochastic MD methods.

3.3 Beyond empirical potentials: ab initio molecular dynamics

In Sect. 3.1, we introduced MD exploiting empirical potentials, i.e., forces are assumed to depend on particles position *via* explicit functions of the atomic positions. As we discussed in that section, this is computationally very effective as one does not have to compute the fixed nuclei *electronic* ground state at each timestep. Nevertheless, as shortly mentioned in the previous section, this approach has the limitation to be unsuitable to describe processes when one observes a significant change in the electronic structure of the system along the dynamics, e.g., if bond breaking or formation takes place (apart the specific cases discussed few lines above). In the Appendix A, we will show machine learning approaches to achieve an accuracy comparable with the most rigorous approach based on the electronic ground state together with an efficiency comparable with empirical potentials. Here, we focus on the ab initio approach. In Sect. 2.1, we have shown that within the mixed quantum-classical treatment introduced in that section, nuclei can be treated as classical particles subjected to a potential corresponding to the nuclear position-dependent ground state eigenvalue of the *electronic* Hamiltonian. The particle's EoM to implement this approach reads:

$$M\ddot{\mathbf{R}} = -\langle \nabla_R \mathcal{H}(\mathbf{r};\mathbf{R})\rangle_0 \,. \tag{40}$$

where $\langle\cdot\rangle_0$ denotes the *quantum* expectation value over the electronic ground state (at the *present* nuclear position). Considering the EoM integration algorithm shown in Fig. 1, implementation of ab initio molecular dynamics according to Eq. (40) implies computing the ground state wave function after the second step in the loop, after atomic positions are *updated* (denoted $\nabla_{\mathbf{R}} V(t' + \Delta t)$). This implies that at each step of MD one must perform a quantum mechanical calculation to compute the ground state electronic wave function.

The focus of this review is not discussing in detail ab initio calculations. Still, to explain the principles of ab initio molecular dynamics, we must introduce some important ingredients allowing the calculation of the electronic ground state wave function. Therefore, we limit the presentation to the details relevant for the purposes of this article. Consider Eq. (5), the eigenvalue equation associated with the electronic Hamiltonian and its *ground state* eigenfunction $\chi_0(\mathbf{r}; \mathbf{R})$. $\chi_0(\mathbf{r}; \mathbf{R})$ depends on the 3N electronic positions $\mathbf{r}$ and parametrically on the nuclear positions $\mathbf{R}$. Its determination is a many-body problem (the N-interacting electrons). Approximated methods are typically based on one particle functions, typically called *orbitals*, $\phi_\alpha(\vec{r}; \mathbf{R})$, functions of the position $\vec{r}$ of a single electron (the overhead arrow is used in place of bold font to distinguish between a 3D vector and a 3ND one). To set a reference, we recall some of the best-known methods based on orbitals: Hartree–Fock (see Ref. [38]) and the the density functional theory in the Kohn–Sham formulation (DFT) [4, 39]. Within these methods, the orbitals $\{\phi_\alpha(\vec{r}; \mathbf{R})\}_\alpha$ must satisfy single-particle eigenvalue equations:

$$\hat{h}(\vec{r}; \mathbf{R}, \{\phi_\beta\}_\beta)\phi_\alpha(\vec{r}; \mathbf{R}) = e_\alpha \phi_\alpha(\vec{r}; \mathbf{R}), \quad \alpha = 1, N_{\text{orb}}. \tag{41}$$

Typically, $\hat{h}(\vec{r}; \mathbf{R}, \{\phi_\beta\}_\beta)$, the single-particle *effective* Hamiltonian, depends on the set of *occupied* orbitals $\{\phi_\beta(\vec{r}; \mathbf{R})\}_\beta$. This makes any component of Eq. (41) non-closed. Then the whole Eq. (41) is solved iteratively: (i) from a first guess of the occupied orbitals one obtains a first guess $\hat{h}(\vec{r}; \mathbf{R}, \{\phi_\beta\}_\beta)$; (ii) the corresponding eigenvalue equation is solved and (iii) with the new set of orbitals one obtains a next guess $\hat{h}(\vec{r}; \mathbf{R}, \{\phi_\beta\}_\beta)$; (iv) this procedure is repeated until a convergence is reached. The final solution is *self-consistent* (within a prescribed accuracy).

Consider a basis set for the one-particle functional space, $\{g_i(\vec{r})\}_{i=1,\infty}$. The orbitals used to approximate the ground state wavefunction (or density, in DFT) can be expanded on the basis set: $\phi_\alpha(\vec{r}; \mathbf{R}) = \sum_{i=1,\infty} c_{\alpha,i}(\mathbf{R})\, g_i(\vec{r})$. Here, the expansion coefficients $\{c_{\alpha,i}(\mathbf{R})\}_{\alpha=1,N_{\text{orb}};i=1,\infty}$ are unknown. If one insert the orbitals projected in the basis set in Eq. (41), the original functional problem is transformed into an algebraic one. An obvious but important consequence, as we shall see shortly, is that the energy of the system within the given approximation is a function of the expansion coefficients $\{c_{\alpha,i}(\mathbf{R})\}_{\alpha=1,m;i=1,\infty}$ (and, parametrically, on the nuclear positions): $E(\{c_{\alpha,i}(\mathbf{R})\}_{\alpha=1,N_{\text{orb}};i=1,\infty})$.

It is well-known [40] that solving the eigenvalues problem for the search of the ground state wavefunction/density is equivalent to minimize a corresponding (energy) functional. Indeed, methods based on orbitals are derived by searching for the orbitals minimizing the energy functional (see, e.g., Szabo and Ostlund for the derivation of the

Hartree–Fock equations [38], and Dreizler and Gross for the derivation of the Kohn–Sham equations [4]). Thus, one can recast the problem of determining the orbitals of Eq. (41) into the following:

$$\left\{\hat{c}_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty} = \text{argmin}_{\{c_{\alpha,i}\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty}) \tag{42}$$

where the hat symbol denotes the set of optimal coefficients, which minimize $E(\cdot)$, and in the index of argmin, we neglected the limits of the coefficients' set to keep the notation compact. For brevity, in Eq. (42), we did not report Lagrange multipliers necessary to impose orthonormality of $\{\phi_\alpha(\vec{r};\mathbf{R})\}_\alpha$.

Many methods have been developed over the years to efficiently solve Eq. (42) (see, e.g., Refs. [41–45]). Here, we report perhaps the simplest one, the *steepest descent method*, which, being very simple, allows one to discuss some general aspects of ab initio MD rather than delving in the technical details. The steepest descent method corresponds to the so-called Aristotelian dynamics, in which the velocity on an object is proportional to the force acting on it. Here, the degrees of freedom subjected to the Aristotelian dynamics are the expansion coefficients $\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty}$, and the force is the gradient of $E(c_{\alpha,i}(\mathbf{R})_{\alpha=1,m;i=1,\infty})$ with respect to them (at a given, fixed, nuclear configuration $\mathbf{R}$):

$$\dot{c}_{\alpha,i} = \gamma\left(-\frac{\partial E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})}{\partial c_{\alpha,i}}\right) \tag{43}$$

where γ is understood as the rate of change of the expansion coefficients. Equation (43) can be numerically solved following an iterative algorithm:

$$c_{\alpha,i}(t+\Delta t) = c_{\alpha,i}(t) + \gamma\left(-\frac{\partial E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})}{\partial c_{\alpha,i}}\right)\Delta t \tag{44}$$

Here, of course t is not the physical time, it is just an additional, *artificial* variable which, under suitable mathematical hypotheses, in its infinity limit guarantees that the coefficients converge to the ones minimizing the energy functional. In fact, notice that:

1. the value of the coefficients $\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty}$ evolves following the (pseudo) force $-\nabla_{\{c_{\alpha,i}(\mathbf{R})\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})$, hence the name of the method steepest descent;
2. when the force is zero, $\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty}$ does not change any longer, and
3. the set of coefficients minimizing the energy has been found.
4. Of course, numerically one never reaches $\nabla_{\{c_{\alpha,i}(\mathbf{R})\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty}) = 0$, rather one stops the procedure when $\left|\nabla_{\{c_{\alpha,i}(\mathbf{R})\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})\right|$ is smaller than a prescribed value.

The interpretation of $\nabla_{\{c_{\alpha,i}(\mathbf{R})\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})$ as an effective force acting on the expansion coefficients, and $\dot{c}_{\alpha,i}$ as their velocities, serves as the idea for an

innovative approach to modeling systems within and beyond the Born–Oppenheimer approximation, where electrons are treated as quantum particles in (close to—see below) their ground state and nuclei as classical ones at finite temperature. Car and Parrinello [46] introduced the extended dynamical system $\left(\mathbf{R}, \dot{\mathbf{R}}, \left\{c_{\alpha,i}\right\}, \left\{\dot{c}_{\alpha,i}\right\}\right)$ taking advantage of the above interpretation of $\dot{c}_{\alpha,i}$ and $\nabla_{\{c_{\alpha,i}(\mathbf{R})\}} E(\left\{c_{\alpha,i}(\mathbf{R})\right\}_{\alpha=1,m;i=1,\infty})$. Notice that at a variance with above, within the Car–Parrinello approach the expansion coefficients are considered independent variables, not function of the nuclear positions $\mathbf{R}$. The dynamics of this extended system is formulated within the Lagrange formalism as:

$$\mathcal{L}\left(\mathbf{R}, \dot{\mathbf{R}}, \left\{c_{\alpha,i}\right\}, \left\{\dot{c}_{\alpha,i}\right\}\right) = \sum_{i=1,N} \frac{1}{2} M_i \left|\vec{R}_i\right|^2 + \sum_{\alpha=1,m;i=1,\infty} \frac{1}{2}\gamma \left|\dot{c}_{\alpha,i}\right|^2 + E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right) \tag{45}$$

where $\sum_{\alpha=1,m;i=1,\infty} \frac{1}{2}\gamma \left|\dot{c}_{\alpha,i}\right|^2$ is the *fictitious* kinetic energy of the expansion coefficients, with γs playing here the role of an *inertia* and $E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right)$ plays the role of the potential energy of the extended system. It is worth remarking that in the steepest descent approach, the mathematical form of $E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right)$ is that of the electronic structure method of reference, e.g., DFT, computed at $\{\hat{c}_{\alpha,i}(\mathbf{R})\}_{\alpha,\mathbf{i}}$, while here this is computed on the basis of the current value of the $\left\{c_{\alpha,i}\right\}$ set. From the so-called Car–Parrinello Lagrangian, one can derive EoM for the variables of the extended system and one can evolve simultaneously nuclei and orbitals. The value of *gamma* is set such that $\left\{c_{\alpha,i}\right\}$ have a much shorter characteristic time than $\mathbf{R}$. This time scale separation (adiabaticity) implies that nuclei move driven by a force *averaged* over the distribution of the expansion coefficient. We will return on this point in the following.

So far, we did not discuss whether the forces acting on nuclei along Car–Parrinello MD would be consistent with the Born–Oppenheimer one. This has been formally proven by Bornemann and Schütte [47]. Here, we illustrate this aspect following an intuitive approach. If the fictitious kinetic energy is low, the values of the expansion coefficients $\left\{c_{\alpha,i}\right\}$ are close to the values $\left\{c^{BO}_{\alpha,i}(\mathbf{R})\right\}$ corresponding to the minimum of $E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right)$ at the current nuclear positions $\mathbf{R}$. In this condition, expansion coefficients are subjected to a quasi-harmonic (symmetric) restoring potential: $E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right) \sim E\left(\left\{c^{BO}_{\alpha,i}\right\}; \mathbf{R}\right) + \sum_{\alpha,\beta}\sum_{i,j} 1/2 E''_{\alpha,\beta,i,j}\left(\left\{c^{BO}_{\alpha,i}\right\}; \mathbf{R}\right)(c_{\alpha,i} - c^{BO}_{\alpha,i})(c_{\beta,j} - c^{BO}_{\beta,j})$. For small displacements of $\left\{c_{\alpha,i}\right\}$ from their optimal value, the nuclear forces have a linear dependence on $\left\{c_{\alpha,i}\right\}$: $-\boldsymbol{\nabla}_{\mathbf{R}} E\left(\left\{c_{\alpha,i}\right\}; \mathbf{R}\right) \sim -\boldsymbol{\nabla}_{\mathbf{R}} E\left(\left\{c^{BO}_{\alpha,i}\right\}; \mathbf{R}\right) - \sum_{\alpha}\sum_{i} \boldsymbol{\nabla}_{\mathbf{R}} E'_{\alpha,i}\left(\left\{c^{BO}_{\alpha,i}\right\}; \mathbf{R}\right)(c_{\alpha,i} - c^{BO}_{\alpha,i})$. As mentioned above, thanks to adiabatic separation, nuclei move according to mean forces averaged over the distribution of faster expansion coefficients. Owing to the symmetric potential acting on $\left\{c_{\alpha,i}\right\}$ in the low fictitious kinetic energy regime, the distribution of the expansion coefficients is symmetric and, owing to the linearity of the atomic force on $\left\{c_{\alpha,i}\right\}$, the average force acting on nuclei is the same as the Born–Oppenheimer one.

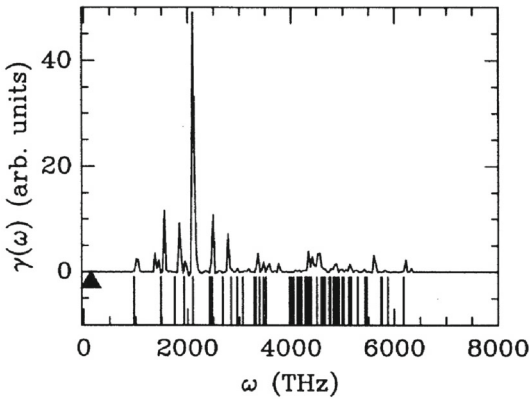

Fig. 4 Power spectrum, i.e., the modulus square of the Fourier transform of the velocity–velocity autocorrelation function of nuclei and $\{\dot{c}_{\alpha,i}\}$, in crystalline silicon. The triangle on the left represents the typical highest value of nuclei vibrational frequency. One notices that the nuclear vibrational frequencies are much lower than the (pseudo)vibrational frequencies of $\{c_{\alpha,i}\}$ in semiconductors. Figure adapted from Ref. [48]

We are left to discuss the condition for adiabaticity of the nuclear and expansion coefficients. This was first investigated by Pastore, Smargiassi and Buda [48]. They introduced a linear approximation of the Car–Parrinello dynamics, revealing that (i) $\{c_{\alpha,i}\}$ have an oscillatory-like dynamics, (ii) the characteristic frequencies are lower bounded by the square root of the band gap, ΔE_g, divided by inertia: $\omega \geq \sqrt{\Delta E_g/\gamma}$. Considering the typical value of the band gap in semiconductors ($\Delta E_g/ \geq 1$ eV) or the so-called HOMO–LUMO (Higher Occupied Molecular Orbital/Lowest Unoccupied Molecular Orbital) molecular gap, the characteristic frequencies of expansion coefficients are orders of magnitude higher than those of the nuclei (see Fig. 4). The lack of overlapping of vibrational frequency range of nuclei and the expansion coefficient prevent an *efficient* energy transfer between the two sub-systems, which makes them adiabatic over the timescale of simulations. Of course, the smaller the gap, the less the system is adiabatic. In these cases, special approaches, such as thermostating the expansion coefficient subsystem at low temperatures [49], are necessary.

4 Free energy calculations

In Sect. 2.2, we established a correspondence between thermodynamic potentials and the partition function, which can be interpreted as the limit histogram (i.e. non-normalized probability) of realizing given values of the thermodynamic variables, e.g., N, V and T. This concept can be generalized to deal with the case of any additional macroscopic observable. Let us introduce this idea considering the example of protein folding, for instance the simple case of alanine dipeptide. The alanine dipeptide, the dimer of the alanine amino acid, can exist in different *conformers*, corresponding to different values of the dihedral angles θ, ϕ, ψ, and ζ shown in Fig. 5. Here, one can ask what is the probability density $P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)$,[4] that in thermodynamic conditions N, V and T, the dipeptide is in any configuration consistent with

[4] In Eq. (46) and the text above this equation, $P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)$ is the conditional probability density to observe given values of the collective variables at given thermodynamic conditions N, V and T (or corresponding conditions in other ensembles). Here, we explain why this is a conditional probability density. Consider $\Omega(\theta^*\phi^*, \psi^*, \zeta^*, N, V, T) = \int d\mathbf{\Gamma} \exp[-H(\mathbf{\Gamma})/k_B T] \ \delta(\theta(\boldsymbol{r}) - \theta^*)\delta(\phi(\boldsymbol{r}) - \phi^*)\delta(\psi(\boldsymbol{r}) -$

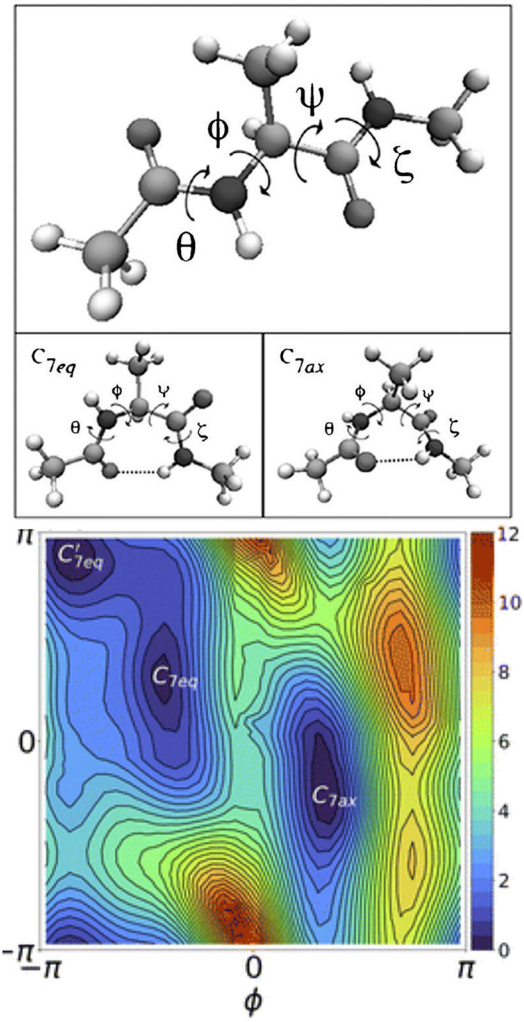

Fig. 5 Alanine dipeptide. In the top panel, the structure is illustrated in its unwrapped configuration to better show the θ, ϕ, ψ, and ζ dihedral angles. In the central panels, the same molecule is presented in the configurations favoring intramolecular hydrogen bonding (adapted from Ref. [51]). Free energy landscape in the subspace of two dihedral angles out of the four listed above when the molecule is in the vacuum (adapted from Ref. [52])

the conditions $\theta(\boldsymbol{r}) = \theta^*$, $\phi(\boldsymbol{r}) = \phi^*$, $\psi(\boldsymbol{r}) = \psi^*$, and $\zeta(\boldsymbol{r}) = \zeta^*$:

$$P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T) = \int \mathrm{d}\boldsymbol{\Gamma}\, \rho(\boldsymbol{\Gamma}) \delta(\theta(\boldsymbol{r}) - \theta^*) \delta(\phi(\boldsymbol{r}) - \phi^*)$$

$\psi^*)\delta(\zeta(\boldsymbol{r}) - \zeta^*)$, which is analogous to Eq. (46) apart that we replace the *normalized* probability density function $\rho(\boldsymbol{\Gamma})$ with its non-normalized $\exp[-H(\boldsymbol{\Gamma})/k_\mathrm{B}T]$ counterpart. As discussed in the Sect. 2.2, Ω is an absolute frequency, i.e., a non-normalized probability, here the probability to observe given values of the collective variables and of the volume, number of particles and temperature: $\Omega(\theta^*, \phi^*, \psi^*, \zeta^*, N, V, T)$. $P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)$ of Eq. (46) is nothing else than $\Omega(\theta^*, \phi^*, \psi^*, \zeta^*, N, V, T)/\Omega(N, V, T)$, the latter being the normalization of $\exp[-H(\boldsymbol{\Gamma})/k_\mathrm{B}T]$: $\rho(\boldsymbol{\Gamma}) = \exp[-H(\boldsymbol{\Gamma})/k_\mathrm{B}T]/\int \boldsymbol{\Gamma} \exp[-H(\boldsymbol{\Gamma})/k_\mathrm{B}T] = \exp[-H(\boldsymbol{\Gamma})/k_\mathrm{B}T]/\Omega(N, V, T)$. Noticing that $\Omega(N, V, T)$ is the non-normalized (marginal) probability density of N, V and T (Sect. 2.2), one concludes that $P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)$ is the ratio between the joint probability $\Omega(\theta^*, \phi^*, \psi^*, \zeta^*, N, V, T)$ and the marginal probability $\Omega(N, V, T)$, hence is the conditional probability $P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)$.

$$\times\delta(\psi(\boldsymbol{r}) - \psi^*)\delta(\zeta(\boldsymbol{r}) - \zeta^*) \tag{46}$$

Given the equivalence of the probability of Eq. (46) and the absolute frequencies of Eqs. (19) and (20), one can introduce a thermodynamic potential associated with $P(\theta^*, \phi^*, \psi^*, \xi^*; N, V, T)$:

$$L(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T) = -k_B T \ln[P(\theta^*, \phi^*, \psi^*, \zeta^*; N, V, T)], \tag{47}$$

which is usually called the Landau free energy. Indeed, there is no fundamental difference between the usual thermodynamic potentials and the Landau free energy: both are a convenient representation of the probability to observe some prescribed values of some observables. Notice that the negative sign in the relation between the probability density and the (Landau) free energy, together with the monotonic nature of the logarithmic function, implies that maxima of the former are associated to minima of the latter. For the case of alanine dipeptide *configurational* free energy, this is illustrated in the bottom panel of Fig. 5.

We remark that the Landau free energy is not restricted to the case of conformational free energy or to any internal degree of freedom of molecules. For example, the Landau free energy has been used to study phenomena focusing on more macroscopic observables, such as the density field of a multiphase system in confined bubble nucleation [50]. In the following, we will use the notation:

$$L(\theta^*; N, V, T) = -k_B T \ln[p(\theta^*; N, V, T)] = -k_B T \ln\left[\int \mathrm{d}\boldsymbol{\Gamma}\, \rho(\boldsymbol{\Gamma})\delta(\theta(\boldsymbol{r}) - \theta^*)\right]. \tag{48}$$

where $\rho(\Gamma) = \exp[-\mathcal{H}/\mathbf{k_B T}]/\mathbf{Q}$ and $\theta(\boldsymbol{r})$, usually called a *collective variable*, in general depends on all the (configurational) degrees of freedom of the system. Other names frequently used for $\theta(\boldsymbol{r})$, depending on the context, are *order parameter* and *reaction coordinate*. For the sake of more agile notation, here and in the following, we will use a single collective variable. However, all concepts discussed in the following can be extended to any number of collective variables. In the following for simplicity when discussing Landau free energy, we will come back to the general symbol F for function.

4.1 Methods for computing free energy

Apparently, for computing both the traditional thermodynamic potentials and the Landau free energy, one has to compute the partition function. Indeed, molecular dynamics (and Monte Carlo) can help efficiently sampling the probability density to be at a point in the phase space and to compute the associated average values, but not values that cannot be expressed as mean values over a suitable regular and smooth function. It was instead soon recognized that the variation of free energy between two macroscopic states can, indeed, be expressed as a ratio between expectation values of suitable observables over the relevant ensemble. Consider a system in the canonical ensemble and the variation of free energy between two temperatures, initial and final, T_i and T_f

$$F^f - F^i \propto -k_B T \log\left[\frac{\Omega_f}{\Omega_i}\right]. \tag{49}$$

one can always simplify for the kinetic energy part, multiply and divide the argument of the integrals in the logarithm by $\int d\mathbf{\Gamma} \exp[-V(\boldsymbol{R})/k_B T_i] \exp[-V(\boldsymbol{R})/k_B T_f]$, after which Eq. (49) can be read:

$$F^f - F^i \propto -k_B T \log\left[\frac{\langle \exp[-V(\boldsymbol{R})/k_B T_f]\rangle_i}{\langle \exp[-V(\boldsymbol{R})/k_B T_i]\rangle_f}\right]. \tag{50}$$

Equation (50) shows that the variation of the free energy between two states can be expressed as ratio between expectation values of suitable (although not too smooth) observables over the proper ensemble average, a quantity that can be computed by molecular dynamics (or Monte Carlo). The terms $\langle \exp[-V(\mathbf{R})/T_\alpha]\rangle_\beta$ (with temperatures α and β, being i or f) also reveal another important fact, that the probability density distributions under conditions i and f must overlap for this approach to be valid for the estimation of $F^f - F^i$. In fact, if they do not overlap, the integrand is large in those regions in which the probability density is small and hard to sample, and *vice versa*. This problem can be addressed by splitting the calculation of $\langle \exp[-V(\boldsymbol{R})/k_B T_f]\rangle_i / \langle \exp[-V(\boldsymbol{R})/k_B T_i]\rangle_f$ into intermediate *overlapping steps*:

$$\prod_{\alpha=1,N-1} \frac{\langle \exp[-V(\mathbf{R})/k_B T_{\alpha+1}]\rangle_\alpha}{\langle \exp[-V(\mathbf{R})/k_B T_\alpha]\rangle_{\alpha+1}}, \tag{51}$$

where $\alpha = 1$ in the initial state i and $\alpha = N$ the final one. This *discrete path* in temperature allows one to compute the ratio of partition functions efficiently even when the probability density function of the initial and final states do not overlap sufficiently.

Let us now generalize the question of the calculation of the variation of free energy to the more general case, including the case of Landau free energies. Consider Eq. (48); here, we are interested in the case in which one wants to compute the relative free energy on an interval of values of the variable $\theta(\mathbf{R})$. In principle, one could bin the relevant interval, run molecular dynamics, and build the histogram of the collective variable. This is an estimate of the probability density $\rho(z)$, where z is a realization (a possible value) of $\theta(\mathbf{R})$: $\theta(\mathbf{R}) = z$. In principle, this estimate of the probability density can systematically be improved by reducing the spacing of the binning of the θ range and making the dynamics longer. Consider the case in which the system presents metastabilities, i.e., two (or more) attractive basins such that the passage from one to the other is infrequent, a *rare event*. These metastabilities are associated with the presence of multiple minima separated by relatively large maxima in the free energy profile of a suitable collective variable (Fig. 6). These concepts will be formalized in the following, in Sect. 5 in the context of the discussion of the theory and methods for studying *Reactive Processes*. For the time being, we rely on the intuitive idea that if the system must overcome a large free energy barrier one has to wait for an infrequent energy fluctuation allowing it to stay on the high free energy state corresponding to

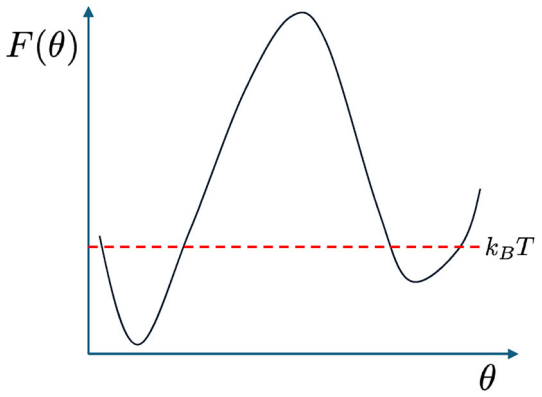

Fig. 6 Cartoon of the free energy profile of the variable θ. Here, we purposely chosen the case of a system presenting metastabilities, and assumed that the collective variable θ is suitable to characterize them, i.e., the free energy along θ presents two minima: an absolute minimum, corresponding to the stable state, and a local minimum, associated to the metastable one. The stable and metastable states are separated by a free energy maximum. Among all possible collective variables, the *reaction coordinate*, $\bar{\theta}$, has the characteristics that a trajectory initiated at free energy maximum has 0.5 probability to reach the either (meta)stable state before than the other (see below). When the free energy along the reaction coordinate presents a large maximum, much larger than the thermal energy $k_{\mathrm{B}}T$ available to the system, the passage from one (meta)stable state to the other is infrequent, it is a *rare event*

the maximum separating two minima. If the time the system takes to pass from one metastable state to another is long, longer than the duration of the trajectory one can afford, the *binning and histogramming* is unfeasible. An early idea to handle this problem is *Umbrella sampling* [53]. Although umbrella sampling was introduced in the context of Monte Carlo, its extension to molecular dynamics is straightforward. In practice, one biases the system to force it to visit infrequent states; The effect of the bias is removed from the histogram of θ a posteriori. Imagine adding the biasing potential $V_b(\theta(\mathbf{R}))$, depending on the atomic positions through the collective variable $\theta(\mathbf{R})$, to the physical potential $V(\mathbf{R})$. With such an extended potential, the system samples the (biased) probability density function:

$$\begin{aligned}\rho_b(z) &= \frac{\int \mathrm{d}\mathbf{\Gamma}\ \exp[-(V(\mathbf{R}) + V_b(\theta(\mathbf{R})))/k_{\mathrm{B}}T]\delta(\theta(\boldsymbol{r}) - z)}{\int \mathrm{d}\mathbf{\Gamma}\ \exp[-(V(\mathbf{R}) + V_b(\theta(\mathbf{R})))/k_{\mathrm{B}}T]} \\ &= \frac{\exp[-V_b(z)/k_{\mathrm{B}}T]}{\langle \exp[-V_b(\theta(\mathbf{R}))/k_{\mathrm{B}}T]\rangle}\rho(z). \end{aligned} \tag{52}$$

Where we have used the simple fact that $f(\theta)\delta(\theta - z) = f(z)\delta(\theta - z)$. The calculation of the $\langle \exp[-V_b(\theta(\mathbf{R}))/k_{\mathrm{B}}T]\rangle$ term poses challenges in systems presenting metastabilities as it requires to sample well the configuration space where the probability is small and the observable $\exp[-V_b(z)/k_{\mathrm{B}}T]$ large. We will return on this question shortly. For the time being, we stress that with a suitable choice of the biasing potential, e.g., the pseudo-harmonic potential $V_b(\theta(\mathbf{R})) = \kappa/2\,(\theta(\mathbf{R}) - z^{\dagger})^2$ with $z^{\dagger}$ close to z, sampling $\rho_b(z)$ by molecular dynamics (or Monte Carlo) is easier than sampling $\rho(z)$. To compute the free energy profile over a broad interval spanning, for

example, more than one (meta)stable state, one can repeat the procedure of Eq. (52) for several values of $z^\dagger$, denoted in the following $\left\{z_i^\dagger\right\}_{i=1,n}$.[5] This allows to obtain a corresponding number of probability distributions of z, denoted by $\left\{\rho_b(z;z_i^\dagger)\right\}_{i=1,n}$. Each $\rho_b(z;z_i^\dagger)$ provides an estimation of the unbiased distribution, denoted by $\rho(z;z_i^\dagger)$. These estimated distributions may overlap, especially those corresponding to close-by values $z_i^\dagger$. In the weighted histogram analysis method (WHAM) [54, 55], this overlap is exploited to circumvent the problem of calculation of $\langle \exp[-V_b(\theta(\mathbf{R}))/k_\mathrm{B}T]\rangle$ in Eq. (52). Let us define the weighted average of the $\rho(z;z_i^\dagger)$ distribution:

$$\rho_{\mathrm{WHAM}}(z) = \sum_{i=1,n} \alpha_i(z)\rho(z;z_i^\dagger) \tag{53}$$

The best estimation of $\rho(z)$ is obtained by the $\left\{\hat{\alpha}_i(z)\right\}_{i=1,n}$ values minimizing the statistical error of $\rho_{\mathrm{WHAM}}(z)$. The derivation of optimal $\hat{\alpha}_i(z)$ is involved and the reader is referred to the original articles, Refs. [54] and [55]. There are other methods to reconstruct the free energy profile beyond WHAM; among others we mention the umbrella integration method [56] which is remarkably numerically stable allowing to get smooth free energy functions.

Umbrella sampling provides a mean of sampling states of low probability but it requires non-trivial post-processing. An alternative approach of sampling states of low probability is based on constrained molecular dynamics, leading to *blue moon ensemble* [57, 58] (see also Ref. [59]). First, consider a canonical variable transformation $(\boldsymbol{\Gamma}) \rightarrow (\boldsymbol{q}, \boldsymbol{p}_q)$, where $\theta(\boldsymbol{R})$ is one of the $\boldsymbol{q}$ variables. We recall that the Jacobian of a canonical transformation is 1 (see, e.g., Ref. [60]), a property that will be exploited in the following. Consider the derivative of the free energy (Eq. (48)) with respect to z

$$\begin{aligned}\frac{dF(z)}{dz} &= -k_\mathrm{B}T\frac{\int \mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}_q\,\rho(\boldsymbol{q},\boldsymbol{p}_q)\;\mathrm{d}\delta(\theta-z)/\mathrm{d}z}{p(z)} \\ &= k_\mathrm{B}T\frac{\int \mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}_q\,\partial\rho(\boldsymbol{q},\boldsymbol{p}_q)/\partial\theta\;\delta(\theta-z)}{p(z)}.\end{aligned} \tag{54}$$

where we exploited the relation $\int \mathrm{d}x f(x)\;\mathrm{d}\delta(\theta(x)-z)/\mathrm{d}z = -\int \mathrm{d}x\,(\mathrm{d}f\,(x(\theta))/\mathrm{d}\theta)\,\delta(\theta(x)-z)$. Let us assume that the system is at constant temperature, hence $\rho(\boldsymbol{q},\boldsymbol{p}_q) \propto \exp[-\tilde{\mathcal{H}}(\boldsymbol{q},\boldsymbol{p}_q)/k_\mathrm{B}T]$, where $\tilde{\mathcal{H}}(\cdot)$ is the Hamiltonian expressed in the transformed variables $(\boldsymbol{q},\boldsymbol{p}_q)$. In this case, Eq. (54) reads:

$$\begin{aligned}\frac{\mathrm{d}F(z)}{\mathrm{d}z} &= \frac{\int \mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}_q\,\partial\tilde{\mathcal{H}}(\boldsymbol{q},\boldsymbol{p}_q)/\partial\theta\;\rho(\boldsymbol{q},\boldsymbol{p}_q)\,\delta(\theta-z)}{p(z)} \\ &\equiv \langle\partial\tilde{\mathcal{H}}(\boldsymbol{q},\boldsymbol{p}_q)/\partial\theta\rangle_z.\end{aligned} \tag{55}$$

[5] Here, we assume that $z_i^\dagger$ values are ordered, e.g., they are reported in ascending order of the value of the collective variable.

Before proceeding with the discussion, it is important to remark two aspects. First, from $\langle \partial \tilde{\mathcal{H}}(\boldsymbol{q}, \boldsymbol{p}_q)/\partial\theta \rangle_z$ by integration over z, one can compute the variation of the free energy between any two states in the relevant range. Thus, by numerical integration with a fixed extreme and the other extreme spanning the relevant range, one can reconstruct the free energy profile associated to a given collective variable (Fig. 6). Second, $\langle \partial \tilde{\mathcal{H}}(\boldsymbol{q}, \boldsymbol{p}_q)/\partial\theta \rangle_z$ is the ensemble average over a conditional probability density function at the current value of the collective variable. Molecular dynamics is suitable for this purpose, provided that one can force the sampling of a conditional probability density. Thus, one can circumvent the problem of computing the partition function, for which molecular dynamics is unsuitable, with the *thermodynamic* integration of an expectation value, combining molecular dynamics and straightforward numerical integration of the mean force.

Let us now move to discussing how to estimate $\langle \partial \tilde{\mathcal{H}}(\boldsymbol{q}, \boldsymbol{p}_q)/\partial\theta \rangle_z$ by molecular dynamics. In principle, one can reformulate molecular dynamics into $(\boldsymbol{q}, \boldsymbol{p}_q)$, but this is cumbersome and must be adapted to the specific θ. Thus, one prefers to work in the Cartesian coordinates, $\boldsymbol{\Gamma}$, which requires obtaining the equivalent of Eq. (55) in these variables. The first observation is that we consider collective variables θ depending only on atomic positions, $\boldsymbol{R}$. Thus, Eq. (55) can be reformulated in terms of the conditional probability density $\rho(\boldsymbol{R}; z) = \rho(\boldsymbol{R})\,\delta(\theta(\boldsymbol{R}) - z)/\int \mathrm{d}\boldsymbol{R}\,\rho(\boldsymbol{R})\,\delta(\theta(\boldsymbol{R}) - z)$

$$\frac{\mathrm{d}F(z)}{\mathrm{d}z} = \int \mathrm{d}\boldsymbol{R} \left[\frac{\partial V(\boldsymbol{R})}{\partial\theta} - k_B T \frac{\partial \ln J(\theta)}{\partial\theta} \right] \rho(\boldsymbol{R}; z). \tag{56}$$

where $\partial V(\boldsymbol{R})/\partial\theta$ can be computed in Cartesian space by applying the chain rule over $\boldsymbol{R}$. $J = \partial(\boldsymbol{u})/\partial(\boldsymbol{R})$ is the Jacobian of the coordinate transformation $\boldsymbol{R} \to \boldsymbol{u} = (\theta, \boldsymbol{q})$, where $\boldsymbol{q}$ is a set of generalized coordinates suitably chosen. If one compares the conditional probability $\rho(\boldsymbol{R}; z)$ with the configuration space probability sampled along constrained MD discussed in Sect. 3, $\rho_z^C(\boldsymbol{R})$, one notices that:

$$\rho(\boldsymbol{R}; z) = \frac{|Z(\boldsymbol{R})|^{-1/2}}{\int \mathrm{d}\boldsymbol{R}\,|Z(\boldsymbol{R})|^{-1/2}\rho_z^C(\boldsymbol{R})} \rho_z^C(\boldsymbol{R}). \tag{57}$$

where $Z(\boldsymbol{R}) = \sum_{i=1,N} 1/M_i\, \partial\theta/\partial\vec{R}_i \cdot \partial\theta/\partial\vec{R}_i$. In the case of multiple conditions, $|Z|$ is replaced with the determinant of the corresponding matrix of elements $\hat{Z}_{\alpha,\beta}(\boldsymbol{\Gamma}) = \sum_{i=1,N} 1/M_i\, \partial\theta_\alpha/\partial\vec{R}_i\, \partial\theta_\beta/\partial\vec{R}_i$. $|Z|$ arises from the variable transformation in the $(\boldsymbol{q}, \boldsymbol{p}_q)$ coordinates. The $|Z(\boldsymbol{R})|^{-1/2}$ term arises from the Gaussian integral associated to the momenta distribution in the $(\boldsymbol{q}, \boldsymbol{p}_q)$ variables due to a $\boldsymbol{q}(\boldsymbol{R})$-dependent term in the kinetic energy in these variables. The detailed derivation of Eq. (57) can be found in Ref. [59] (Secs. 4 and 5).

Summarizing,

$$\frac{\mathrm{d}F(z)}{\mathrm{d}z} = \frac{\int \mathrm{d}\boldsymbol{R} \left[\frac{\partial V(\boldsymbol{R})}{\partial\theta} - k_B T \frac{\partial \ln J(\theta)}{\partial\theta} \right] |Z|^{-1/2} \rho_z^C(\boldsymbol{R})}{\int \mathrm{d}\boldsymbol{R}\,|Z|^{-1/2}\rho_z^C(\boldsymbol{R})}, \tag{58}$$

and numerator and denominator can be estimated by constrained molecular dynamics. A related approach has been proposed in the context of the accelerated exploration of the configuration space [61]. Here, instead of imposing a constraint one imposes a *restraint*, similar to the pseudo-harmonic biasing potential $V_b(\theta(\mathbf{R})) = \kappa/2\,(\theta(\mathbf{R}) - z)^2$ discussed in the context of umbrella sampling. The difference with umbrella sampling is twofold:

(i) the coupling force is stronger, and essentially imposes the system to stay in configurations almost satisfying the condition $\theta(\mathbf{R}) = z$;
(ii) the analysis of results follows the approach of the Blue Moon ensemble, extracting from simulations an estimate of the mean force $\mathrm{d}F(z)/\mathrm{d}z$, which is then integrated over the relevant θ range.

In restrained molecular dynamics, one imposes no constraints on the momenta. Hence, the trajectory samples the conditional distribution:

$$\begin{aligned}\rho_\kappa(\boldsymbol{R}; z) &= \frac{\exp[-(V(\mathbf{R}) + \kappa/2\,(\theta(\mathbf{R}) - z)^2)/k_\mathrm{B}T]}{\int \mathrm{d}\mathbf{R}\, \exp[-(V(\mathbf{R}) + \kappa/2\,(\theta(\mathbf{R}) - z)^2)/k_\mathrm{B}T]} \\ &= \frac{\exp[-V(\mathbf{R})/k_\mathrm{B}T]\exp[-\kappa/2\,(\theta(\mathbf{R}) - z)^2/k_\mathrm{B}T]}{\int \mathrm{d}\mathbf{R}\, \exp[-V(\mathbf{R})/k_\mathrm{B}T]\exp[-\kappa/2\,(\theta(\mathbf{R}) - z)^2/k_\mathrm{B}T]}.\end{aligned} \tag{59}$$

In the limit $\kappa \to \infty$, the (normalized) Gaussian $\exp[-\kappa/2\,(\theta(\mathbf{R}) - z)^2/k_\mathrm{B}T]$ goes to $\delta(\theta(\mathbf{R}) - z)$, and $\rho_\kappa(\boldsymbol{R}; z) \to \rho_\infty(\boldsymbol{R}; z)$, the latter being the conditional probability at $\theta = z$. When κ is large but finite the probability density sampled along he restrained trajectory is an approximation of the limiting one. With restrained molecular dynamics, however, one can take the explicit derivative of the Landau free energy with respect to θ

$$\begin{aligned}\frac{\mathrm{d}F(z)}{\mathrm{d}z} &= -k_\mathrm{B}T\frac{\int \mathrm{d}\boldsymbol{\Gamma}\, \mathrm{d}\rho_R(\boldsymbol{\Gamma}, \theta)/\mathrm{d}\theta}{\int \mathrm{d}\boldsymbol{\Gamma}\, \rho_R(\boldsymbol{\Gamma}, \theta)} \\ &= \frac{\int \mathrm{d}\boldsymbol{\Gamma}\, \kappa\,(\theta(\mathbf{R}) - z)\rho_R(\boldsymbol{\Gamma}, \theta)}{\int \mathrm{d}\boldsymbol{\Gamma}\, \rho_R(\boldsymbol{\Gamma}, \theta)},\end{aligned} \tag{60}$$

i.e., the expectation value of $\kappa\,(\theta(\mathbf{R}) - z)$ over the ensemble sampled by restrained molecular dynamics. In other words, restrained molecular dynamics requires no unbiasing, and $\kappa\,(\theta(\mathbf{R}) - z)$ is an (asymptotically) unbiased estimator of the mean force.

As it emerges clearly from the beginning of this section, the research community has been developing methods for computing (differences of) free energy for long time. A work that inspired a family of methods anticipates the appearance of computers. [62] As above, in the concluding part of this section, we aim at providing to interested readers a concise bibliography of other methods that were not discussed in this article. We tried our best to identify the key articles (non-necessarily the original ones), but, as in the spirit of this work, this paragraph (and analogous paragraphs in the following) is not meant to be *complete*. A class of methods has been devised with the objective of exploring the coordinate space of the system and identify metastabilities. Some of these methods were meant to explore the configuration space, [63–67] other the collective

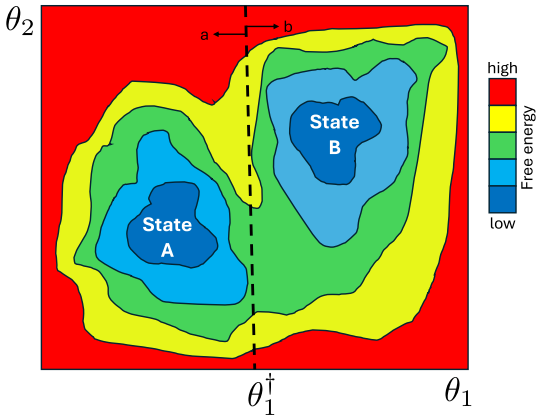

Fig. 7 Free energy landscape according to the collective variables θ_1 and θ_2. The landscape is represented as a color map, with cold and warm colors indicating low and high free energies, respectively. One notices that the free energy presents two minima, corresponding to the (meta)stable states A and B. The dashed line is an example of (hyper)surfaces used for the calculation of the TST transition frequency

variable space. [68–71] The latter methods also allow to determine the Landau free energy in the selected set of collective variables.

5 Reactive processes

So far, we have applied statistical mechanics and molecular dynamics to investigate the *static properties* of a system, e.g., equilibrium thermodynamics discussed in the previous section. Here, we move to the investigation of dynamical properties of systems. First, we focus on the transition between (meta)stable states of a system in equilibrium conditions and the rate associated to this process. Here, *equilibrium conditions* means that a system can visit all (meta)stable states over and over, i.e., it cannot be trapped forever in one of them. In practice, we can assume ergodicity, i.e., that time average is equivalent to (a suitable) ensemble average. Before proceeding, let us illustrate a couple of examples of the processes we consider in this section. Chemical reactions, with the system transiting between *reactants* and *products*, are examples of the processes that we are interested in. Transitions between conformational states of molecules, e.g., protein folding mentioned in the previous section, is another example. Physical processes, e.g., nucleation of crystals or vapor bubbles, are also examples of processes that can be investigated by the methods discussed in this section. In all cases, we can describe the (meta)stable states characterizing a system by the values of suitable order parameters. This is shown in Fig. 7, where the collective variables θ_1 is suitable to identify two free energy minima, corresponding to the (meta)stable states A and B. The second collective variable, θ_2, represents all other degrees of freedom.

Here, we focus on the calculation of the rate of the A $\rightarrow$ B process. First, we discuss the approaches to investigate dynamical problems within the transition state theory (TST). Next, we will discuss the limits of TST. Finally, we provide elements of the transition path theory (TPT) with the related method of the string in collective variables.

5.1 Transition state theory

TST has been originally formulated by Eyring in 1935 [72], with crucial modifications introduced by several authors [73–75] to include the so-called *dynamical corrections*. Here, we discuss TST in the formulation proposed by Vanden-Eijnden and Tal [76], which is more coherent with the theoretical framework of this article.

Let us start by defining the rate of the process A $\rightarrow$ B, k_{AB}, corresponding to the number of trajectories leaving A and reaching B per unit time. We assume that the system follows a (stochastic) dynamics at constant number of particles, volume and temperature. Extensions to other ensembles are possible. We also assume that the system spends most of the time in A and B, i.e., in A and B is concentrated most of the probability to find the system. In formulas, we assume that $P(A) + P(B) \sim 1$, where $P(A) = \int_A \mathrm{d}\boldsymbol{R} \int \mathrm{d}\boldsymbol{p}\, \rho(\boldsymbol{\Gamma})$ and similar for $P(B)$. We also assume that *reactive* trajectories, those leaving A and reaching B, are uncorrelated. This happens if the system looses *memory* between successive reactive trajectories, which is guaranteed when the waiting time within A and B is long with respect to the molecular timescale. This hypothesis is coherent with (i) the hypothesis that the system spends almost all time in A and B and (ii) is guaranteed when the (meta)stable states are separated by barriers much larger than the thermal energy $k_{\mathrm{B}}T$, i.e., when it is unlikely to visit the states separating (meta)stable states.

Within the above hypotheses, the number of reactive trajectories from A to B per unit time is the inverse of the average time spent in A, τ_A: $k_{AB} = 1/\tau_A$. In turn, τ_A is given by $\tau\, P(A)$, with τ the total time, divided by the number of transitions from A to B, N_{AB}. N_{AB} is half the total number of transition N_T between the two (meta)stable states. Thus, $\tau_A = \tau P(A)/(N_T/2) = 2\tau P(A)/N_T$, and $k_{AB} = N_T/2\tau P(A) = \nu/2P(A)$, where ν is the crossing frequency between A and B.[6]

Indeed, finding an explicit formula for ν is non-trivial and we first search for the frequency of the related problem of crossing a (hyper)surface separating A and B. This is illustrated by the dashed line on Fig. 7. This line separates the state a and b, the former containing state A and the latter state B. Let us define the surface by an order parameter, $\theta(\boldsymbol{R}) = \theta_1(\boldsymbol{R}) - \theta_1^{\dagger}$ such that when $\theta(\boldsymbol{R}) < 0$ the system is in a, and when $\theta(\boldsymbol{R}) > 0$ the system is in b. Let us call this (hyper)surface the transition state of the system. Within this partitioning of the space, one measures the rate of going from a to b (instead of going from A to B), which consistently with the formulas derived above, reads

$$k_{ab}^{\mathrm{TST}} = N_T^{\mathrm{TST}}/2\tau P(a) = \nu^{\mathrm{TST}}/2P(a). \tag{61}$$

where the superscript TST recalls that the rate and the other relevant quantities refers to the transition from a to b, rather than from A to B.

We now focus on the methods to compute ν^{TST} and $P(a)$ (Eq. (61)). Let us introduce the Heaviside step function of $\theta(\boldsymbol{R})$:

[6] Note that ν is the crossing frequency between A and B, including both the transitions *from A to B* and *from B to A*.

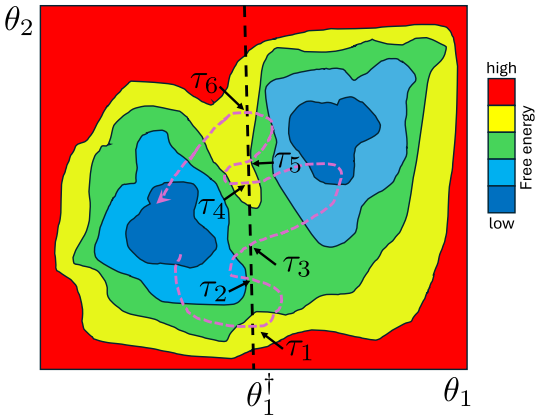

Fig. 8 Free energy landscape same as in Fig. 7. A cartoon trajectory is also reported (purple dashed terminating with an arrow). This trajectory crosses the (hyper)surface separating the a and b subsystems at times τ_i, also reported in the figure

$$\Theta\left(\theta(\boldsymbol{R})\right) = \begin{cases} 0, & \theta(\boldsymbol{R}) < 0 \\ 1, & \theta(\boldsymbol{R} \geq 0 \end{cases}$$

Consider the integral

$$\frac{1}{\tau}\int_0^{\tau} \mathrm{d}s \ \left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right| \tag{62}$$

the $\left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right|$ function (of time s) is a sequence of Dirac's delta function, in correspondence of the times τ_i when the system crosses the boundary between a and b: $\left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right| = \sum \delta(s - \tau_i)$ (see Fig. 8). Thus, the integral of Eq. (62) is the sum of integrals of delta functions, each integral being equal to 1. Hence, Eq. (62) counts the number of transitions between a and b in the time τ. Hence,

$$\nu^{\mathrm{TST}} = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} \mathrm{d}s \ \left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right|. \tag{63}$$

Next, we manipulate Eq. (63) to express it in terms of ensemble average. First, one notices that

$$\begin{aligned} \left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right| &= \left|\dot{\boldsymbol{R}}(s)\cdot\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})\frac{\mathrm{d}\Theta(\theta)}{\mathrm{d}\theta}\right| \\ &= \left|\dot{\boldsymbol{R}}(s)\cdot\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})\right|\delta(\theta) \end{aligned} \tag{64}$$

Hence,

$$\begin{aligned} \nu^{\mathrm{TST}} &= \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} \mathrm{d}s \ \left|\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right)\right| \\ &= \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} \mathrm{d}s \ \left|\dot{\boldsymbol{R}}\cdot\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})\right|\delta(\theta) \\ &= \int \mathrm{d}\boldsymbol{\Gamma} \left|\dot{\boldsymbol{R}}\cdot\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})\right|\delta(\theta)\rho(\boldsymbol{\Gamma}) \end{aligned} \tag{65}$$

the last step resulting from ergodicity and $\rho(\mathbf{\Gamma})$ being the probability density function sampled by the dynamics followed by the system. By multiplying and dividing Eq. (65) by $P(0) = P(\theta = 0) = \int \mathrm{d}\mathbf{\Gamma}\delta(\theta)\rho(\mathbf{\Gamma})$, ν^{TST} can be expressed as

$$\nu^{\mathrm{TST}} = \left\langle \left| \dot{\boldsymbol{R}} \cdot \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| \right\rangle_0 P(0). \tag{66}$$

where $\langle \cdot \rangle_0$ denotes the canonical average conditional to $\theta(\boldsymbol{R}) = 0$. The TST rate (Eq. (61)) is then

$$k_{ab}^{\mathrm{TST}} = \left\langle \left| \dot{\boldsymbol{R}} \cdot \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| \right\rangle_0 \frac{P(0)}{2\,P(a)}. \tag{67}$$

Note that $P(0)$ is the **probability density** to be at $\theta(\boldsymbol{R}) = 0$ while $P(a)$ is the **probability** of the domain a. In Eq. (70), we will manipulate $P(a)$ so as to express it in terms of a suitable **probability density**.

$\left\langle \left| \dot{\boldsymbol{R}} \cdot \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| \right\rangle_0$ can be written so as to separate the momentum and configuration integrals. One starts noticing that

$$\left| \dot{\boldsymbol{R}} \cdot \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| = \left| \dot{\boldsymbol{R}}_n \right| \left| \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| \tag{68}$$

with $\dot{\boldsymbol{R}}_n$ the component of the velocity orthogonal to $\theta(\boldsymbol{R}) = 0$. The integral over momenta can be performed explicitly (Vanden-Eijnden and Tal, Eq. 21 and 22 of Ref. [76], and other authors before them) obtaining:

$$k_{ab}^{\mathrm{TST}} = \sqrt{\frac{2k_{\mathrm{B}}T}{\pi}} \left\langle \left| \nabla_{\boldsymbol{R}} \theta(\boldsymbol{R}) \right| \right\rangle_0 \frac{P(0)}{P(a)}. \tag{69}$$

In Eq. (69), the conditional ensemble average $\langle \cdot \rangle_{\theta=0}$ is referred only to the configuration space.

One can further simplify Eq. (69). Following Eq. (48), $P(\theta) = \exp[-F(\theta)/k_{\mathrm{B}}T]$, and $P(a) = \int_a \mathrm{d}\theta\, P(\theta) = \int_{-\infty}^{0} \mathrm{d}\theta\, \exp[-F(\theta)/k_{\mathrm{B}}T]$. Within the subdomain a, $F(\theta)$ presents a single (deep) minimum and the integral $\int_{-\infty}^{0} \mathrm{d}\theta\, \exp[-F(\theta)/k_{\mathrm{B}}T]$ can be solved with the Laplace method, consisting in expanding $F(\theta)$ to the second order in θ around θ_A, the position of the minimum. Since $F'(A) = \mathrm{d}F(\theta)/\mathrm{d}\theta|_{\theta_A} = 0$, within the second-order approximation of the free energy one is left with solving a Gaussian integral, and

$$P(a) \sim \sqrt{\frac{2\pi k_{\mathrm{B}}T}{F''(A)}} \exp[-F(A)/k_{\mathrm{B}}T] = \sqrt{\frac{2\pi k_{\mathrm{B}}T}{F''(A)}}\, P(A). \tag{70}$$

where $F''(A) = \mathrm{d}^2F(\theta)/\mathrm{d}\theta^2|_{\theta_A}$ is the curvature of the free energy at the reactant minimum. Within this approximation of $P(a)$, the TST rate reads

$$
\begin{aligned}
k_{ab}^{\mathrm{TST}} &= \sqrt{\frac{F''(A)}{\pi^2}}\,\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0\,\frac{P(0)}{P(A)} \\
&= \sqrt{\frac{F''(A)}{\pi^2}}\,\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0\,\exp\left[-\frac{F(0)-F(A)}{k_{\mathrm{B}}T}\right] \\
&= \sqrt{\frac{F''(A)}{\pi^2}}\,\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0\,\exp\left[-\frac{\Delta F^{\dagger}}{k_{\mathrm{B}}T}\right]
\end{aligned}
\tag{71}
$$

Equation (71) shows that to determine the TST rate one needs to (i) compute the free energy difference between the minimum associated to the reactant (a) and the transition state, (ii) the curvature of the free energy at the reactant and (iii) the conditional average $\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0$ at the transition state ($\theta = 0$). Points (i) and (ii) can be readily solved with the techniques discussed in Sect. 4.1. The estimate of the curvature of the free energy can be obtained by a parabolic fitting of the free energy profile around θ_A. Thus, we are left with computing the conditional average of point (iii). Equation (57) relates the conditional probability density to the probability density sampled along constrained molecular dynamics. Thus, one can compute $\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0$ by constrained molecular dynamics performing the same unbiasing discussed previously. One can also compute $\langle|\nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})|\rangle_0$ by restrained molecular dynamics; this time, no unbiasing is necessary, as discussed above.

Figure 8 shows a problem related to the estimate of the rate by Eq. (71). One notices that the branch of trajectory exiting from A crosses the transition state several times before reaching B. Hence, computing the frequency of switching between A and B by counting the frequency of crossing of the transition state results in an over-counting. As mentioned at the beginning of this section, several authors have proposed approaches to avoid this multiple counting [73–75]. Here we present an approach proposed by Vanden-Eijnden and Tal [76]. We anticipate that (i) a computational method based on this approach is suitable to handle the case in which correlation time is much shorter than the average time the system spends in the basins a or b, i.e., no correlated recrossing occurs. This is not the case, for example, when thermalization time is very long, e.g., in gas phase reactions in the energy diffusion regime (low friction regime). For For processes not suffering from this problem, (ii) the approach presented in the following is instructive to understand the meaning of the so-called transmission coefficient, the coefficient counting the number of crossing of the transition state surface which are reactive, in terms of other relevant probabilistic quantities. We remark that methods exist to estimate the transmission coefficient that do not suffer of the limitations described in the point (i) (see, e.g., the classical works of Bennet [74] and Chandler [75], or more modern approaches improving over the past methods, e.g., Refs. [77, 78]).

The exact rate can be reformulated starting from Eq. (63):

$$
\nu = \lim_{\tau\to\infty}\frac{2}{\tau}\int_0^{\tau} \mathrm{d}s\ \chi_B(\boldsymbol{R}(s_{BS}^{+}(s)))\chi_A(\boldsymbol{R}(s_{AB}^{-}(s)))\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right). \tag{72}
$$

where $\chi_A(\cdot)$ and $\chi_B(\cdot)$ are characteristic functions, amounting to 1 if its argument lay within the domain associated to the function and zero otherwise. s^+_{BS} is defined as the minimum time successive to time s when the system is either in B or on the TST surface (here denoted by S). s^-_{AB} is defined as the maximum time before time s when the system was either in A or B. With these definitions, $\chi_A(\boldsymbol{R}(s^-_{AB}(s))) = 1$ if the trajectory comes from the reactant (A—Fig. 7), $\chi_B(\boldsymbol{R}(s^+_{BS}(s))) = 1$ if the trajectory goes to the product (B) next, without recrossing the surface separating a from b (S). As a result, $\chi_B(\boldsymbol{R}(s^+_{BS}(s)))\chi_A(\boldsymbol{R}(s^-_{AB}(s))) = 1$ if and only if the branch of trajectory comes from A and goes next to B before any recrossing. Thus, contributions to integral of Eq. (72) arise only for (i) reactive trajectories (coming from A) and that (ii) are considered only once in the case of multiple crossing. The factor 2 in front of the integral in Eq. (72) is because in Eq. (61), we use the frequency of transition between A and B, both ways between the two (meta)stable states.

Now we move formulating Eq. (72) in terms of ensemble averages. Recall Eq. (13), the equation defining ergodicity. Here, it is assumed that the trajectory passes an infinite number of times in a volume around a phase space point. We assume that the dynamics is stochastic and, as a consequence, each time the system passes by a phase space point it goes in future and comes from the past along a different path, and so the path may or may not come from A last, and may or may not go in future to B next. Thus, the time average implies an expectation value over the different paths passing by each point $\boldsymbol{\Gamma}$: $\mathbb{E}(\chi_B(\boldsymbol{R}(s^+_{BS}(s)))\chi_A(\boldsymbol{R}(s^-_{AB}(s))))$. Since the dynamics is assumed to be Markovian, i.e., what happen in future depends only on the present and not on the past, $\mathbb{E}(\chi_B(\boldsymbol{R}(s^+_{BS}(s)))\chi_A(\boldsymbol{R}(s^-_{AB}(s)))) = \mathbb{E}(\chi_B(\boldsymbol{R}(s^+_{BS}(s))))\mathbb{E}(\chi_A(\boldsymbol{R}(s^-_{AB}(s))))$. Let us call $\zeta_{B/S}(\boldsymbol{R}, \boldsymbol{p}) = \mathbb{E}(\chi_B(\boldsymbol{R}(s^+_{BS}(s))))$ and $\zeta_{A/B}(\boldsymbol{R}, -\boldsymbol{p}) = \mathbb{E}(\chi_A(\boldsymbol{R}(s^-_{AB}(s))))$. $\zeta_{B/S}(\boldsymbol{R}, \boldsymbol{p})$ is the probability that a trajectory at the phase space point $\boldsymbol{R}, \boldsymbol{p}$ on the hypersurface $\theta(\boldsymbol{R}) = 0$ goes next to B rather than crossing the surface S, and $\zeta_{A/B}(\boldsymbol{R}, -\boldsymbol{p})$ is the probability that a trajectory comes from A rather than B. Thus, Eq. (72) reads

$$\begin{aligned} \nu &= \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathrm{d}s \ \chi_B(\boldsymbol{R}(s^+_{BS}(s)))\chi_A(\boldsymbol{R}(s^-_{AB}(s)))\dot{\Theta}\left(\theta(\boldsymbol{R}(s))\right) \\ &= \int \mathrm{d}\boldsymbol{\Gamma} \ \zeta_{B/S}((\boldsymbol{R}, \boldsymbol{p})\zeta_{A/B}(\boldsymbol{R}, -\boldsymbol{p}) \ \left[\dot{\boldsymbol{R}} \cdot \nabla_{\boldsymbol{R}}\theta(\boldsymbol{R})\right]_+ \delta(\theta)\, \rho(\boldsymbol{\Gamma}) \\ &= \kappa_t \ \nu^{\mathrm{TST}} \end{aligned} \tag{73}$$

where κ_t, the transmission coefficient, is defined as the ratio between the true and the transition state theory approximation of the rate (Eq. (65)): $\kappa_t = \nu/\nu^{\mathrm{TST}}$.

Equation (73) brings to a straightforward computational method to evaluate the *exact rate*, i.e., the rate including the transmission coefficient. First, one computes the transition state frequency as discussed previously. Then, at each configuration of a sample extracted from the (configuration) transition state ensemble, one starts trajectories with momenta $\boldsymbol{p}$ and $-\boldsymbol{p}$, with $\boldsymbol{p}$ extracted from a Maxwell distribution. From these trajectories, one estimates $\zeta_{B/S}((\boldsymbol{R}, \boldsymbol{p})$ and $\zeta_{A/B}(\boldsymbol{R}, -\boldsymbol{p})$ by counting the fraction of them reaching B before recrossing S, and A before B, respectively. From these, one obtains κ_t. Since states A and B are attractive points on the configuration space, one expect that short trajectories are sufficient for this part of the calculation.

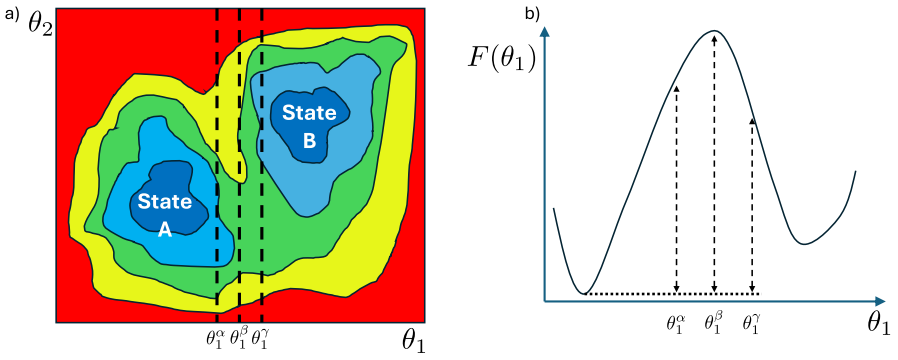

Fig. 9 **a** Free energy landscape same as in Fig. 7. Three dashed lines are displayed, representing different values of the collective variables corresponding to different separating surfaces to be used as transition state. **b** Free energy profile along the collective variable θ_1, with the barriers corresponding to the three values of the collective variables associated to the dashed lines of landscape of panel **a**

Nevertheless, the number of these trajectories can be very large if $\zeta_{B/S}((\boldsymbol{R}, \boldsymbol{p})$ and/or $\zeta_{A/B}(\boldsymbol{R}, -\boldsymbol{p})$ are small.

Equation (73) shows that, in principle, the choice of the collective variable used to identify the transition state, as well as the value of the given collective variable for setting the transition state, is not crucial for computing the value of the rate as far as the collective variable and its value are suitable to distinguish between the reactant and product states, A and B. To illustrate this aspect, consider Fig. 9, where three values of the collective variable θ_1 are reported, θ_1^α, θ_1^β, and θ_1^γ. The barrier depends on the value of the collective variable, and thus does the term $\exp[-\Delta F^\dagger / k_B T]$. For example, the barrier at θ_1^α and θ_1^γ are both smaller than at θ_1^β, and the exponential term of the rate goes in the opposite way. Nevertheless, this effect is balanced by the transmission coefficient. For example, at θ_1^α $\zeta_{BS}(\boldsymbol{R}, \boldsymbol{p})$ is smaller than at θ_1^β, compensating for the barrier term. Complementary, at θ_1^γ the term $\zeta_{AB}(\boldsymbol{R}, -\boldsymbol{p})$ is smaller than at θ_1^β, in this case, this term compensating for the higher value of the barrier term at θ_1^γ. Despite the fact that the rate associated to any of the transition states is the same, the difficulty in their estimation changes. To make the point simpler to understand, consider a dividing surface very close to reactant. In this case, the transmission coefficient is very small because most of the trajectories starting from this surface come back to the surface rather than reaching the product B, i.e., $\zeta_{B/S}((\boldsymbol{R}, \boldsymbol{p})$ is very small. Thus, the evaluation of this term, which can be obtained by starting a number of molecular dynamics from this surface and computing the frequency of trajectories reaching the product, requires shooting very many trajectories for an accurate estimate of $\zeta_{B/S}((\boldsymbol{R}, \boldsymbol{p})$. How to select a suitable separating surface is discussed in the next section.

The approach illustrated above to compute the exact rate, including the transmission coefficient, is not the only possible one. Indeed, pioneering works in this field date back to mid 1970s. [74, 75] Other methods for computing the rate of a process which are not based on the transition state theory. [79–85] These methods typically assume one knows the initial and final state of a process. However, there are cases in which the final state in unknown, and one looks for possible transition states allowing to escape

from the present metastable state (see, e.g., Refs. [86, 87] and [88]). Finally, one must consider the case of a system characterized by many (intermediate) metastable states, with reactant and products that can be connected by more than one multi-step path. These cases can be analyzed within Markov State Models. Readers interested in this subject might want to consider Refs. [89] and [90].

5.2 Committor function and (elements of) the transition path theory

The potential difficulty in estimating the transmission coefficient just mentioned, can be more systematically analyzed. Figure 10 shows four typically topologies of free energy landscapes. In this figure, the landscape is represented as a function of two collective variables but conclusions can be extended to any number of degrees of freedom. To the landscape of each panel are associated two additional quantities. The first is the *projection* of the free energy landscape on the variable θ_2 in correspondence of the putative transition state $\theta_1(\boldsymbol{R}) = z_1^\dagger$, $F(z_2; z_1^\dagger)$ (as before, z_2 are the value of the collective variable $\theta_2(\boldsymbol{R})$). The second is another probabilistic concept connected to $\zeta_{AB}(\boldsymbol{R}, -\boldsymbol{p})$ and $\zeta_{BS}(\boldsymbol{R}, \boldsymbol{p})$: the *committor* function, first introduced in this context by Onsager [91], and more recently discussed by Geissler et al. [92].

The committor function $q_A(\boldsymbol{\Gamma})$ measures the probability that a trajectory passing by the phase space point Γ reaches the metastable state A next (before any other one). In other words, it measures how much a trajectory passing by the given phase space point is committed to a given state (in future). Committor $q_A(\boldsymbol{\Gamma})$ ranges between 0, when the system at $\boldsymbol{\Gamma}$ is fully non-committed to reach the target state, and 1, when it is fully committed to reach that state. With reference to Fig. 10 since a trajectory passing by $\boldsymbol{\Gamma}$ certainly reaches either A or B in future, $q_A(\boldsymbol{\Gamma}) + q_B(\boldsymbol{\Gamma}) = 1$. The concept of committor can be applied also to *reduced* spaces, e.g., collective variables. Thus, one can ask what is the probability that trajectories passing by any configuration consistent with a prescribed value of $\theta_1(\boldsymbol{\Gamma})$, z_1 and $\theta_2(\boldsymbol{\Gamma})$, z_2, of Fig. 10 reaches A, $q_A(z_1, z_2)$.

Regardless whether the committor is defined in the phase space or any reduced space, it is intuitive that the transition state of a chemical reaction or a physical process is the isocommittor hypersurface $q_A = 0.5$, meaning that 50 % of trajectories go to products next and 50 % to reactants. Given a generic (hyper)surface S, points belonging to S possibly have different values of the committor. Thus, one can ask what is the distribution of the committor on S, i.e., what is the probability that a point belonging to S has a given value q_A. For the case of the surface (a line) $\theta_1(\boldsymbol{\Gamma}) = z_1^\dagger$ of Fig. 10, the distribution of the committor, $P(q_A(z_2, z_1^\dagger))$, is shown in bottom right chart of any panel. Obviously, the distribution of committor on the transition state of a given reaction/process is delta-like centered at 0.5. Thus, for example, in the case of the system shown in Fig. 10a $\theta_1(\boldsymbol{\Gamma}) = z_1^\dagger$ is a suitable approximation of the transition state. Let us develop a qualitative argument leading to this conclusion. In panel a, the putative transition state does not cross the two basins. Rather, it is (almost) tangent to the free energy isolines. Thus, $F(z_2; z_1^\dagger)$ presents a single minimum in correspondence of both the attractive basins of A and B. The symmetry of the free energy landscape at any point of the line $\theta_1(\boldsymbol{R}) = z_1^\dagger$ makes it such that trajectories departing from the line toward right in the figure end up in B, and those departing toward left end up

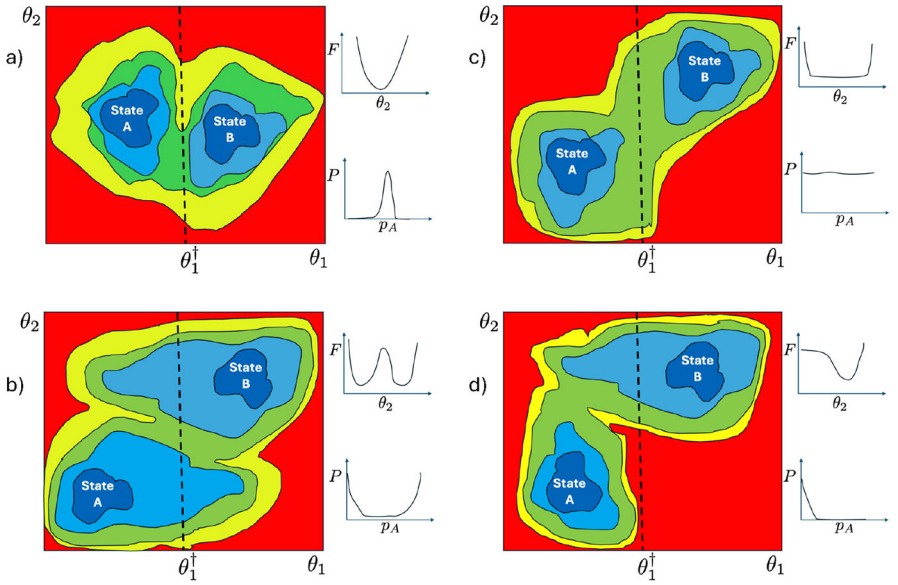

Fig. 10 Four classes of shapes of free energy landscapes. Here, for obvious reason of graphical representation, we consider a free energy of two variables, but the same arguments can be extended to higher dimensional cases. For each panel, the 2D free energy is represented in the main chart, with its projection along the second collective variable and the distribution of the committor function reported in the top and bottom secondary charts, respectively. The image is inspired to Fig. 9 of Ref. [79]

in A. Since momenta are symmetrically distributed (see Sect. 5.1), one expect that committor of points laying on this line is 0.5.

Let us now consider the cases of panels b–d. In all cases, we assume to define the transition state according to the condition $\theta_1(\mathbf{\Gamma})) = z_1^\dagger$. The case of panel b is when the putative transition state crosses the attractive basins of reactants and products. In this case, $F(z_2; z_1^\dagger)$ presents two minima, in correspondence of the attractive basins of A and B. Trajectories started from z_2 values associated to A and B mostly end up in the corresponding basins. Consider, for example, starting a trajectory from a point with a value z_2 close to the value corresponding to B. If the initial momentum points toward B, the trajectory ends up in this metastable state. If the initial momentum points in the opposite direction, the system experiences a forces pushing it again toward B. Thus, the value of q_A of points in this region is ~ 0. A complementary argument can be developed for points on $\theta_1(\mathbf{\Gamma}) = z_1^\dagger$ in correspondence of A: in this case $q_A \sim 1$. Overall, the committor distribution is bimodal, with two peaks at 0 and 1 and $\theta_1(\mathbf{\Gamma}) = z_1^\dagger$ is a poor approximation of the transition state. In the case of panel c, $F(z_2; z_1^\dagger)$ presents a broad, flat, low free energy region. In fact, in this case in the z_2 region intermediate between the values corresponding to A and B, q_A takes values intermediate between 0 and 1. Finally, in the case of panel d, when the transition state crosses mostly one attractive basin, the committor distribution presents a single peak, either at 0 or 1.

Before proceeding the analysis, let us clarify some notational and definition aspects. The committor function $q_B^+(\mathbf{\Gamma})$ ($q_A^+(\mathbf{\Gamma})$) measures the probability that a system in $\mathbf{\Gamma}$ goes next to B (A) before A (B). This is the same definition we gave above with the addition of the index + in the symbol to stress that this concerns future events. One can also introduce the committor function about the probability to come from a given set, e.g., $q_B^-(\mathbf{\Gamma})$ gives the probability that the trajectory passing through the point $\mathbf{\Gamma}$ comes last from B.[7]

Usually, one describes the state of a system and its processes in the configuration space $\boldsymbol{R}$. Thus, the committor function is also restricted to this space: $q_B^{\pm}(\boldsymbol{R})$. Consider the isocommittor surface $S_c(\boldsymbol{R}) = c$, i.e., $S_c(\boldsymbol{R}) = \left[q_B^+(\boldsymbol{R}) = c\right.$, with $c \in [0, 1]]$. S_c foliates the space, i.e., the entire (configuration) space can be split into (an infinite number of) surfaces S_c. Any trajectory passing by the points belonging to the surface S_c has a probability c to reach B next. Within the restriction that the committor depends only on the configuration subspace, $q_B^+(\boldsymbol{R}) + q_B^-(\boldsymbol{R}) = 1$, and since $q_A^-(\boldsymbol{R}) + q_B^-(\boldsymbol{R}) = 1$ then $q_B^+(\boldsymbol{R}) = q_A^-(\boldsymbol{R})$. Thus, trajectories passing through $\boldsymbol{R}$ have a probability $q_A^-(\boldsymbol{R}) = q_B^+(\boldsymbol{R})$ to come from A last and $q_B^+(\boldsymbol{R})$ to go to B next. In the case of Markovian dynamics, the probability that a trajectory comes from A and go to B is $q_A^-(\boldsymbol{R}) \times q_B^+(\boldsymbol{R}) = q_B^+(\boldsymbol{R})^2$, i.e., the value of committor function allows to determine how probable is that a trajectory passing by a point is reactive. From the intuitive point of view, isocommittor surfaces measure the *degree of progress* of a process: low values of $q_B^+(\boldsymbol{R})$ correspond to the locus of point (the surface) where it is unlikely to reach the products, i.e., one is in an *early stage of the reaction*, and large values of $q_B^+(\boldsymbol{R})$ to surfaces where it is highly likely to reach them, toward the *end of the reaction*. More in general, if one knows the committor function one can in principle compute any quantity relative to a reaction, as discussed in detail in Ref. [93] and references therein.

Unfortunately, isocommittor surfaces are very difficult to identify. Maragliano et al. [51] have developed a method, the string method in collective variable, allowing to determine a planar approximation to isocomittor surfaces close to the most likely reactive path. In the following, we present this method but, given the mathematical complexity, and contrary to the style of this review, we refrain from giving detailed derivations. Let us first formulate the objectives of the method and introduce its key ingredients. One of the objectives is reducing the dimensionality of the variable space from $6N$ phase space variables $\mathbf{\Gamma}$ to the m collective variables $\{\theta_i(\boldsymbol{R})\}_{i=1,m}$ depending only on the configuration space. $\theta_i(\boldsymbol{R})$ are thought to include all the relevant degrees of freedom needed to describe the reaction at hand. Notice that the method does not require m to be small. For the sake of brevity of notation, in the following, the set of collective variable will be represented in the vectorial notation $\boldsymbol{\theta}(\boldsymbol{R})$. As above, $\boldsymbol{z}$ are the realization of the collective variables, the values they take. Also in this case, we use a vectorial representation for the values of all collective variables. Consistently with the reduction of the dimensionality of the variable space, the committor functions

[7] Notice that for a deterministic dynamics, e.g., Newton's dynamics, the committor has *pathological* values, either 1 or 0. Thus, here we assume that the system follows a stochastic dynamics. We remark that the dynamics of a non-isolated subsystem is stochastic though the dynamics of the overall isolated system is Newtonian. Hence, the assumption of considering that the system evolves according of a stochastic dynamics is not nonphysical.

are approximated by functions of the collective variables $\boldsymbol{\theta}$: $q_B^+(\boldsymbol{R}) = f\Big(\boldsymbol{\theta}_i(\boldsymbol{R})\Big)$. Within the collective variable space, one searches for the most likely transition path connecting reactants and products (within the given *reactive channel*, in the case reactants and products are connected *via* multiple reactive channels). Close to the most likely transition path, where most of the reactive trajectories pass by, isocommittor surfaces can be approximated as a plane. One aims at finding the equation of such a plane. In particular, one wants to identify the equation for the plane corresponding to $q_B^+(\boldsymbol{R}) = 0.5$, the approximation of the *transition state* hypersurface nearby the intersection with the most likely path.

Let us denote by $\boldsymbol{z}(\lambda)$ the minimum free energy path, which is here parametrized by its fractional arc-length, $\lambda \in [0, 1]$. Notice that $\boldsymbol{z}(\lambda)|_{\lambda=0} \in$A and $\boldsymbol{z}(\lambda)|_{\lambda=1} \in$B. The equation to be satisfied by $\boldsymbol{z}(\lambda)$ is:

$$[A(\boldsymbol{z}(\lambda))\,(-\boldsymbol{\nabla}_{\boldsymbol{z}} F(\boldsymbol{z}(\lambda)))]_{\perp} = 0 \tag{74}$$

where the metric matrix $A(\cdot)$ of the collective variable space, of elements $A_{i,j}(\cdot) = \left\langle \boldsymbol{\nabla}_{\boldsymbol{R}}\theta_i(\boldsymbol{R}) \cdot \boldsymbol{\nabla}_{\boldsymbol{R}}\theta_j(\boldsymbol{R}) \right\rangle_{\boldsymbol{z}}$, the conditional expectation value of $\boldsymbol{\nabla}_{\boldsymbol{R}}\theta_i(\boldsymbol{R}) \cdot \boldsymbol{\nabla}_{\boldsymbol{R}}\theta_j(\boldsymbol{R})$, results from the reduction of the dimensionality of the phase space from $6N$ to m. $F(\mathbf{z})$ is the Landau free energy and $-\boldsymbol{\nabla}_{\mathbf{z}} F(\mathbf{z})$ is the corresponding mean force. The meaning of Eq. (74) is that the most likely path is the one with zero orthogonal component of the effective force to the path, as determined by the metric matrix A.. Note that the planar approximation to isocommittor surfaces passing by $\boldsymbol{z}$ is orthogonal to $\boldsymbol{\nabla}_{\boldsymbol{z}} F(\boldsymbol{z})$. In particular, the $q_B^+(\boldsymbol{R}) = 0.5$ isocommittor surface crosses the minimum free energy path at the point corresponding to the maximum of the free energy along the path itself. Thus, the local approximation of the transition state is the plane orthogonal to $\boldsymbol{z}(\lambda)$ crossing the path at the point corresponding to the maximum of the free energy.

Thus, we are left with finding an algorithm to determine the minimum free energy path. Equation (74) presents an (iterative) algorithm. The basic idea is that, starting from a discretized first guess path $\left\{\boldsymbol{z}^0(\lambda_i)\right\}_{i=1,n}$, the images (discretized points belonging to the path) along the *string* are evolved so as to satisfy, in a suitable number of iterations, the condition of Eq. (74) (Fig. 11a). Consider the first-order dynamics in the pseudo-time s:

$$\dot{\boldsymbol{z}}(\lambda_i, s) = \mu\,[A(\boldsymbol{z}(\lambda_i, s))\,(-\boldsymbol{\nabla}_{\boldsymbol{z}} F(\boldsymbol{z}(\lambda_i, s)))]_{\perp} \tag{75}$$

where $\boldsymbol{z}(\lambda_i, s)$ is the value of the i-th image of $\boldsymbol{z}$ along the guess path at *time* s. μ is a weight: it guarantees the dimensional consistency of the equation and allows to control the speed of evolution of $\dot{\boldsymbol{z}}$. We must immediately remark that Eq. (75) is just a pseudo-dynamics whose property is that in the limit $s \to \infty$ $\boldsymbol{z}(\lambda_i, s)$ goes to the minimum free energy path. Here, either constrained or restrained molecular dynamics could be used to compute the conditional averages $\langle\cdot\rangle_{\boldsymbol{z}(\lambda_i,s)}$ associated to $A(\boldsymbol{z}(\lambda_i, s))$ and $\boldsymbol{\nabla}_{\boldsymbol{z}} F(\boldsymbol{z}(\lambda_i, s)$. Equation (75) can be numerically solved with an algorithm in spirit analogous to molecular dynamics, by discretizing the pseudo-time s. In this case, a value of s corresponds to a given iteration of the string optimization procedure, with $\left\{\boldsymbol{z}^0(\lambda_i, s = 0)\right\}_{i=1,n} = \left\{\boldsymbol{z}^0(\lambda_i)\right\}_{i=1,n}$.

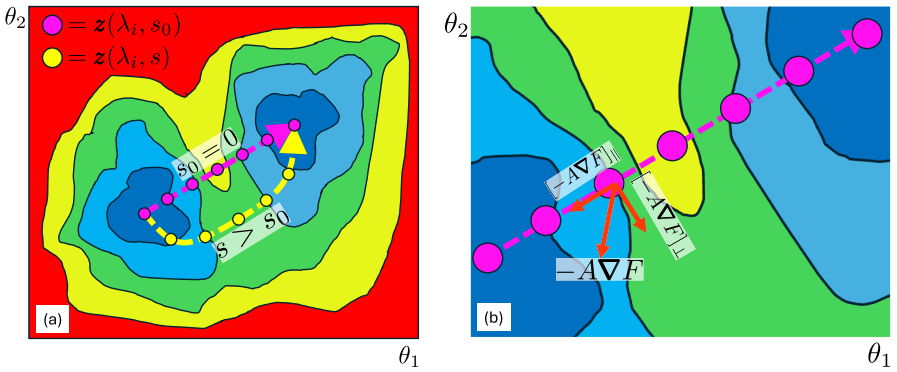

Fig. 11 Illustration of the string method (**a**) and the principle beneath its simplified, $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$, version. In the string method, a discretized path is evolved according to the pseudo-dynamics Eq. (75), or equivalent evolution equations. This dynamics evolves toward the minimum free energy path. **b** In its simplified version, one recognizes that the effect of the tangential component of the generalized force is to change the relative distance between images, an effect that can be removed at the end of each iteration without the need of projecting the tangential component of $[A(\mathbf{z})\,(-\nabla_{\mathbf{z}} F(\mathbf{z}))]$ out

The orthogonal generalized force term, $[A(\boldsymbol{z}(\lambda_i, s))\,(-\nabla_{\mathbf{z}} F(\boldsymbol{z}(\lambda_i, s)))]_\perp$ can in principle be obtained by projecting the tangential term out of the overall generalized force:

$$[A(\mathbf{z})\,(-\nabla_{\mathbf{z}} F(\mathbf{z}))]_\perp = \left(\mathbb{1} - \mathbf{t}(\mathbf{z})\mathbf{t}^T(\mathbf{z})\right) \cdot [A(\mathbf{z})\,(-\nabla_{\mathbf{z}} F(\mathbf{z}))]\,, \tag{76}$$

with $\mathbb{1}$ identity matrix and $\mathbf{t}(\boldsymbol{z})$ the tangent to the path at the position $\boldsymbol{z}$ and at the current pseudo-time. Within the discretized path $\{\boldsymbol{z}(\lambda_i)\}_{i=1,n}$, the tangent can be computed by finite differences between successive points along the path. However, this approach typically results in an approximation of $\mathbf{t}(\boldsymbol{z})$ of limited accuracy, which is critical when the path is discretized in a few images. This, in turn, typically results in a slow convergence of the algorithm. Maragliano et al. [51] have identified an approach to overcome this problem, which has then been further improved by Ren et al. [94]. The fundamental observation beneath this approach is that the effect of the tangential component of the force is to change the distance between successive images (Fig. 11b). Thus, instead of solving the pseudo-dynamics of Eq. (75), one implements a two-step algorithm:

1. One first integrates the dynamics $\dot{\boldsymbol{z}}(\lambda_i, s) = \mu\,[A(\boldsymbol{z}(\lambda_i, s))\,(-\nabla_{\mathbf{z}} F(\boldsymbol{z}(\lambda_i, s)))]$ for one step, i.e., one uses the entire generalized force $[A(\boldsymbol{z}(\lambda_i, s))\,(-\nabla_{\mathbf{z}} F(\boldsymbol{z}(\lambda_i, s)))]$;
2. Then, one reparametrizes the path, moving the images along the polygonal line associated to the points $\{\boldsymbol{z}(\lambda_i, s)\}_{i=1,n}$. After reparametrization, the images $\{\boldsymbol{z}(\lambda_i)\}_{i=1,n}$ are at the original fractional arc-length distance, e.g., equidistant.

Not only the string method allows us to identify the most likely reactive path, illustrating the *mechanism* of chemical the mechanism of chemical reactions or physical processes. It also provides a means of improving the efficiency of the calculation of the rate of the process. Indeed, as discussed above, the 0.5 isocommittor surface is

the one maximizing the transmission coefficient. Thus, one can (i) first compute the string in a suitable set of collective variables, then (ii) compute the free energy terms (barrier and curvature) of Eq. (71) at the maximum and reactant minimum along the minimum free energy path, and, finally, (iii) compute the transmission coefficient from trajectories shot from a sample extracted from the ensemble conditioned to the plane $n(\boldsymbol{z}(\lambda^{\dagger}))$.

As above, we conclude this section providing to readers interested to more broadly explore methods for investigating reaction paths a brief bibliography. Several review articles have been published discussing the theoretical framework of reactive processes, e.g., Refs. [79] and [95]. Same as in the case of *exploration methods*, techniques (in addition to those discussed in the manuscript) to investigate reactive paths can be rooted in the configuration [79–82, 85, 96, 97] or collective variable space. [98] Finally, above we discussed the importance of the committor function for the description of reactive events. Indeed, the committor function is **the** *reaction coordinate*. However, we also mentioned that its identification is possible only in simple cases, and we have shown a method to determine a local approximation to reactive paths. Some authors have developed approaches (based on the maximum likelihood) representing the committor function as a suitable combination of collective variables [99, 100]. These methods also provide a criterion to test whether a given collective variable is a statistically relevant degree of freedom to represent a reactive process.

6 Dynamical properties and non-equilibrium systems

So far we have considered equilibrium systems, i.e., system whose macroscopic properties can be expressed as an ensemble average over an equilibrium probability density function, and used molecular dynamics to sample equilibrium ensemble averages. In this section, we consider the case of dynamical properties and non-equilibrium systems, i.e., (i) time correlation functions in equilibrium, (ii) systems whose probability density function is either time-dependent or stationary under the action of external perturbations. (iii) An additional interesting case is that of active matter (see, e.g., Refs. [101–103]), which can also be studied by molecular dynamics approaches. In the following, we mainly focus on the first two cases.

Given the deep connection between the classes of properties (i) and (ii) provided by the fluctuation–dissipation theorem, [104, 105] in the following, we will concentrate on the non-equilibrium case. The problem with this case is sampling a probability density function that, usually, is of unknown mathematical form. Here, we will first derive a general formalism for handling these cases and then we will apply it in two contexts. Consider the case in which the system is initially at equilibrium, or in a stationary state that can be sampled by molecular dynamics. At a given time, denoted $t = 0$ in the following, a perturbation starts acting on the system. For those cases in which the perturbation is weak enough (with respect to microscopic interactions), one can resort to the so-called linear response theory. Within linear response theory, responses can be expressed in terms of equilibrium ensemble average of suitable dynamical quantities.

Let us start by recalling some general results discussed in Sect. 2. Consider a system governed by the time-dependent Hamiltonian $H(\boldsymbol{\Gamma}, t) = H_0(\boldsymbol{\Gamma}) + H_p(\boldsymbol{\Gamma}, t)$, where

$H_0(\mathbf{\Gamma})$ is the time-independent equilibrium Hamiltonian containing, for example, the usual kinetic and interatomic potential terms (see Sect. 3). A generic *microscopic* property $\hat{O}(\mathbf{\Gamma})$, which has a time-dependence only through the phase space variables $\mathbf{\Gamma}$, evolves according to the following equation:

$$\begin{aligned}\dot{\hat{O}}(\mathbf{\Gamma}) &= \left\{H(\mathbf{\Gamma}, t), \hat{O}(\mathbf{\Gamma})\right\} \\ &\equiv i\mathcal{L}(t)\hat{O}(\mathbf{\Gamma}),\end{aligned} \tag{77}$$

where $\{\cdot, \cdot\}$ are the Poisson brackets and $i\mathcal{L}(t)$ is the Liouvillian operator associated to the Hamiltonian $H(\mathbf{\Gamma}, t)$. In Sect. 2, Eq. (18), we have shown that the probability density function, for Hamiltonian system, satisfies the related equation

$$\frac{\partial \rho(\mathbf{\Gamma}, t)}{\partial t} = \{\rho(\mathbf{\Gamma}, t), H(\mathbf{\Gamma}, t)\} = -i\mathcal{L}(t)\rho(\mathbf{\Gamma}, t). \tag{78}$$

Thus, following Eq. (22), the time evolution operator of any phase space observable $\hat{O}(\mathbf{\Gamma})$, $S(t)$,[8] is the adjoint of the time evolution operator of the probability density function: $\rho(\mathbf{\Gamma}, t) = S^*(t)\rho(\mathbf{\Gamma}, 0)$. Consider now the expectation value of observable $O(\mathbf{\Gamma})$ at time t:

$$\begin{aligned}O(t) &= \langle \hat{O}(\mathbf{\Gamma})\rangle_t \equiv \int \mathrm{d}\mathbf{\Gamma}\ \hat{O}(\mathbf{\Gamma})\, \rho(\mathbf{\Gamma}, t) = \int \mathrm{d}\mathbf{\Gamma}\ \hat{O}(\mathbf{\Gamma})\, S^*(t)\rho(\mathbf{\Gamma}, 0) \\ &= \int \mathrm{d}\mathbf{\Gamma}\, [S(t)\hat{O}(\mathbf{\Gamma})]\, \rho(\mathbf{\Gamma}, 0) = \int \mathrm{d}\mathbf{\Gamma}_0\ \hat{O}(\mathbf{\Gamma}(t; \mathbf{\Gamma}_0))\, \rho(\mathbf{\Gamma}_0, 0).\end{aligned} \tag{79}$$

This relation means that in a non-equilibrium system, one can compute the value of a macroscopic observables at time t, $O(t)$, by taking the expectation value of the microscopic observable at the time-evolved phase space point $\mathbf{\Gamma}(t; \mathbf{\Gamma}_0)$ over the initial probability density function $\rho(\mathbf{\Gamma}_0, 0)$.

Equation (79) has been more commonly used when the initial probability density function is an equilibrium distribution. Based on Eq. (79), computational methods were developed for computing properties of non-equilibrium systems starting from equilibrium states. Notice that the initial equilibrium state can also be any stationary one, e.g. an ensemble where the phase space accessible is limited to the hypersurface consistent with a given value of some observable.

We start the discussion on non-equilibrium systems first considering the case of calculation of *transport coefficients*. Consider a macroscopic flow J caused by a field E. Just to fix ideas, this can be heat flux caused by temperature gradient between two walls confining the system. Empirical, linear, *constitutive relations* have been identified between the field and the flow when the system is under the action of a weak, stationary field:

$$J = \alpha E \tag{80}$$

[8] Here, the time evolution operator is relative to an initial time t_0 that, for the sake of limiting heavy notation, is dropped here and in the following.

where α is the (scalar or tensorial) *transport coefficient* relating J and E. As just mentioned, this linear relation holds only for weak fields, i.e., $\alpha = \lim_{E\to 0} J/E$. Although Eq. (80) has been written here for the simpler case of a scalar field in a single species system, it can be extended to much more general cases, e.g., the effect of a vectorial field. From the point of view of statistical mechanics, J is the ensemble average of the microscopic observable corresponding to the macroscopic phenomenon under investigation over the stationary, non-equilibrium probability density function $\rho_\infty(\mathbf{\Gamma})$, where the index ∞ stresses that the probability density function is stationary (though non-equilibrium).

In the following, we will consider a special, nevertheless paradigmatic, case: an external field, $E(\boldsymbol{x}, t)$, coupled to an observable microscopic field $\mu(\boldsymbol{x}) = \sum_{i=1}^{N} \mu_i(\mathbf{\Gamma})\delta(\boldsymbol{R}_i - \boldsymbol{x})$, sum of atomistic contributions. We assume also that the field has the form $E(\boldsymbol{x}, t) = W(\boldsymbol{x})A(t)$. In this case, the Hamiltonian perturbation reads:

$$\begin{aligned} H_p(\mathbf{\Gamma}, t) &= -\int \mathrm{d}\boldsymbol{x} \sum_{i=1}^{N} \mu_i(\mathbf{\Gamma})\delta(\boldsymbol{R}_i - \boldsymbol{x})W(\boldsymbol{x})A(t) \\ &= -\sum_{i=1}^{N} \mu_i(\mathbf{\Gamma})W(\boldsymbol{R}_i)A(t) = -F(\mathbf{\Gamma})A(t) \end{aligned} \tag{81}$$

where $F(\mathbf{\Gamma}) = \sum_{i=1}^{N} \mu_i(\mathbf{\Gamma})W(\boldsymbol{R}_i)$. The time evolution operator satisfies the Dyson relationship:

$$S(t) = S_0(t) + \int_0^t \mathrm{d}\tau \; S(\tau) i\mathcal{L}_p(\tau) S_0(t-\tau), \tag{82}$$

where, as above, $S(t)$ is the time evolution operator associated to the total Hamiltonian $H(\mathbf{\Gamma}, t) = H_0(\mathbf{\Gamma}) + H_p(\mathbf{\Gamma}, t)$, $S_0(t)$ is the time evolution operator of the unperturbed Hamiltonian, and $i\mathcal{L}_p(\tau) = \{\cdot, H_p(\mathbf{\Gamma}, \tau)\}$ is the Liouvillian associated with $H_p(\mathbf{\Gamma}, t)$. Here, we focus on the case of weak perturbation, so that we can express Eq. (82) as follows:

$$S(t) \sim S_0(t) + \int_0^t \mathrm{d}\tau \; S_0(\tau) i\mathcal{L}_p(\tau) S_0(t-\tau). \tag{83}$$

Recalling Eq. (79), the value of a macroscopic observable at time t is

$$\begin{aligned} O(t) = \left\langle \hat{O} \right\rangle_t &= \left\langle \hat{O} \right\rangle_0 + \int_0^t \mathrm{d}\tau \int \mathrm{d}\mathbf{\Gamma} \left[S_0(\tau)\, i\mathcal{L}_p(\tau)\, S_0(t-\tau)\, \hat{O}(\mathbf{\Gamma}) \right] \rho_0(\mathbf{\Gamma}) \\ &= \left\langle \hat{O} \right\rangle_0 - \int_0^t \mathrm{d}\tau \int \mathrm{d}\mathbf{\Gamma} \left[S_0(t-\tau)\, \hat{O}(\mathbf{\Gamma}) \right] i\mathcal{L}_p(\tau)\, S_0^*(\tau)\, \rho_0(\mathbf{\Gamma}) \\ &= \left\langle \hat{O} \right\rangle_0 + \frac{1}{k_\mathrm{B}T} \int_0^t \mathrm{d}\tau \left\langle \hat{O}(t-\tau)\, \dot{\hat{F}}(\tau) \right\rangle_0 A(\tau). \end{aligned} \tag{84}$$

where $\rho_0(\mathbf{\Gamma})$ is the initial probability density function, assumed, as stated above, to be the equilibrium one. The third equality stems from the integration by part related to the operator $i\mathcal{L}_p(\tau)$. The equilibrium ensemble average $\langle\cdot\rangle_0$ within the time integral in the r.h.s term results from the application of the unperturbed time evolution $S_0^*(\tau)$ to the equilibrium probability density function $\rho_0(\mathbf{\Gamma})$. Additionally, the $\dot{F}(\tau)$ term results from the application of the perturbation Liouvillian to the equilibrium probability density function: $i\mathcal{L}_p\rho_0(\mathbf{\Gamma}) = \left\{\rho_0(\mathbf{\Gamma}), H_p(\mathbf{\Gamma}, t)\right\} = -\rho_0(\mathbf{\Gamma})/k_\mathrm{B}T\left\{H_0(\mathbf{\Gamma}), H_p(\mathbf{\Gamma}, t)\right\} = -\rho_0(\mathbf{\Gamma})/k_\mathrm{B}T\left(i\mathcal{L}_0 H_p(\mathbf{\Gamma}, t)\right)$. Recalling that $H_p(\mathbf{\Gamma}, t) = -F(\mathbf{\Gamma})A(t)$, it follows $i\mathcal{L}_0 H_p(\mathbf{\Gamma}, t) = -\dot{F}(\tau)A(\tau)$.

Consider now the case of a perturbation arising from a stationary field starting to act on an equilibrium system from a given time $t_0 = 0$, $A(t) = A_0\, h(t)$, with $h(\cdot)$ the Heaviside step function. As stressed above, we are considering a future time when the system has reached stationarity. Finally, here we focus on observables associated to the perturbation field such that $\dot{\hat{O}}(\mathbf{\Gamma}) = \dot{F}(\mathbf{\Gamma}) = \hat{J}(\mathbf{\Gamma})$. The latter equality stresses that here we focus on the case of observables whose time derivative is the microscopic field associated to the response under study. For the transport coefficient (Eq. (80)), one gets:

$$\alpha = \lim_{A_0\to 0}\frac{J}{A_0} = \frac{1}{k_\mathrm{B}T}\int_0^\infty \mathrm{d}\tau \left\langle \hat{J}(\tau)\hat{J}(0)\right\rangle, \tag{85}$$

one of the time-honored Kubo formulas. Equation (85) shows that transport coefficients can be computed from time integrals of the time (auto)correlation functions of microscopic observables associated to the flux. Equation (85) can be computed by numerical integration of $\left\langle \hat{J}(\tau)\hat{J}(0)\right\rangle$. In principle, the latter can be computed by equilibrium molecular dynamics: one runs molecular dynamics in the relevant ensemble and computes the ensemble average of the autocorrelation function by computing $1/M\sum_{i=1}^{M}\hat{J}(\mathbf{\Gamma}(t+\tau_i;\,\mathbf{\Gamma}(\tau_i)))\times\hat{J}(\mathbf{\Gamma}(\tau_i))$, where τ_i is a time along the molecular dynamics. This approach, however, is computationally inefficient as it requires *to scan* the trajectory back and forth. A more efficient approach is based on the convolution theorem, which transform a convolution in a given space in a simple product in the Fourier transformed variable [106].

As mentioned above, the linear response theory summarized in Eq. (85) is based on the hypothesis that the perturbation field is weak. The question is what is meant by weak. Numerical experiments have shows that the linearity range reaches values of fields unexpectedly macroscopically high, e.g., for argon in the case of thermal transport, the temperature gradient can go up to 10^8 K/cm, and for momentum transfer, shear rate up to 10^{12} s^{-1}.

Next, we consider the calculation of properties in non-equilibrium systems beyond transport coefficients. We still focus on the case in which the system is initially in stationary conditions, and an external perturbation is applied or released at time t_0. As above, to make notation lighter, in the following, we denote the initial time $t_0 = 0$. Notice that many interesting problems fall in this class [107–110]. As in the case of linear response theory, here one is possibly more interested in time-dependent responses. Also in this case, to compute the macroscopic time-dependent observable

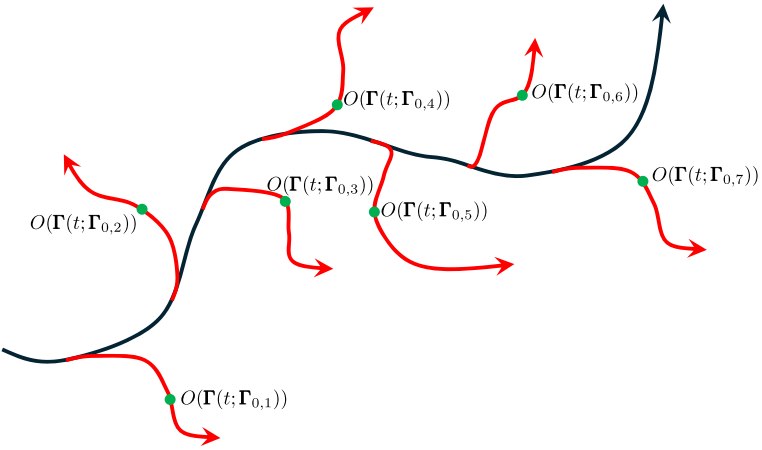

Fig. 12 Cartoon of non-equilibrium molecular dynamics for computing the value of the macroscopic observable $O(t)$ according to the estimator of Eq. (86). The black line represents the stationary trajectory of molecular dynamics before an external perturbation is applied. The red arrows represent trajectories starting from $\mathbf{\Gamma}_{0,i}$ after perturbation is applied, each branch an element of the sampled perturbed trajectories. Green points represent the phase space points moving along the perturbed trajectories after perturbation in applied

$O(t)$, we take advantage of the relation between the time evolution operator of phase space microscopic observables and probability density function (cfr. Eqs. (77) and (78)) as implemented in Eq. (79). This equation immediately provides a mean for computing $O(t)$:

1. One samples initial conditions, $\mathbf{\Gamma}_0$ [108, 111] (black line in Fig. 12), using one of the methods discussed in the previous sections, either along the *standard* molecular dynamics trajectory in the relevant initial ensemble, using methods for sampling conditional ensemble, e.g., constrained [28] or restrained [61] molecular dynamics. In the following, this sample of initial (microscopic) conditions is denoted $\left\{\mathbf{\Gamma}_{0,i}\right\}_{i=1,M}$, with M the number of elements in the sample.
2. From each of the initial conditions $\mathbf{\Gamma}_{0,i}$ in the sample, one runs molecular dynamics in the relevant conditions, including the effect of the external perturbation(red arrows in Fig. 12). Let us denote by $\mathbf{\Gamma}(t;\mathbf{\Gamma}_{0,i})$ the time evolution of initial condition $\mathbf{\Gamma}_{0,i}$ at time t. The *microscopic observable* at this time is $\hat{O}(\mathbf{\Gamma}(t;\mathbf{\Gamma}_{0,i}))$
3. Finally, the microscopic observable at time t along molecular dynamics of the previous point is averaged over initial conditions

$$O(t) \sim \frac{1}{M}\sum_{i=1}^{M}\hat{O}(\mathbf{\Gamma}(t;\mathbf{\Gamma}_{0,i})). \tag{86}$$

A final comment is in order. At variance with regular equilibrium distributions (we mean, e.g. in presence of metastabilities or phase transitions), the time-dependent probability density function can be multimodal. In such a case, taking the averages according to Eq. (79), without any special prescription, might lead to oversee phe-

nomena one is interested in. Consider, for example, the convection flux induced by the gravitational field and an orthogonal temperature gradient. The system is initially at thermal *equilibrium* and at time t_0, one turns on a temperature gradient. A convective flux is produced which can run *clockwise* or *counterclockwise*. If one computes the velocity field for this problem as obtained from Eq. (79) with the computational method discussed above, one could get a null velocity field, as two opposite convective motions could be generated that cancel each other. In reality, only one of the two motions is realized for each $\mathbf{\Gamma}_{0,i}$, but the *selection* of one of the two is not embodied in the direct application of the method just introduced. In principle, one should select initial conditions leading to relevant phenomena (e.g., *clockwise* and *counterclockwise* convective modes). Unfortunately, how to identify which microscopic initial conditions will bring to which of the possible future macroscopic phenomena is not generally easy.

7 Conclusions

Our aim has been to provide an overview of the theory and methods for investigating the properties of condensed matter by molecular dynamics. We started by providing a justification of the use of a classical description of nuclei, subjected to forces arising from the electronic degrees of freedom. Within this framework, we introduced a statistical mechanics description of the system. Next, we have established a connection between statistical mechanics and thermodynamics, and generalized the concept of thermodynamic potentials to the case of observables beyond the usual thermodynamic variables, i.e., we introduced so-called Landau free energies. Next, we introduced molecular dynamics as a tool for computing the quantities of interest for the Statistical Mechanics treatment of our systems. First we discussed *standard* molecular dynamics. Then, we discussed special techniques for computing thermodynamic potentials. Among the processes of interest for condensed matter systems, special techniques are required for investigating problems occurring on free energy landscapes presenting barriers sizeably higher than the thermal energy. We concluded the main text discussing the theory and computational methods for studying non-equilibrium systems. A large Appendix is devoted to machine learning methods.

In this review, we refrained from providing a complete reference to all possible methods to investigate condensed matter systems, privileging a detailed discussion of a limited number of approaches. Indeed, despite the size of this review, many aspects have been left completely out. For example, we did not discuss theory and methods to investigate systems in their excited electronic state, and we did not discuss theory and methods to take into account the quantum nature of nuclei. At the same time, we avoided discussing any method related to Monte Carlo. Paraphrasing the words of one of us in closing the summer school "Computer Simulation in Condensed Matter: From Materials to Chemical Biology": the field has become too wide to aim at comprehensively discussing progresses in the field in a single text. Still, we hope this work can be the starting point for young researchers to enter the field, or for more accomplished scientists to explore domains of computer atomistic simulations beyond their research domain.

During the last 20–30 years atomistic simulations, in general, and molecular dynamics, in particular, were very successful at investigating the properties of condensed matter systems of interest for physics, e.g., computing the properties of liquids, the phase diagram of relatively simple systems, defect dynamics, etc. More and more, molecular dynamics is having an impact on chemistry, e.g., to study chemical reactions in solution and solid phase, including the development of novel materials for energy harvesting, catalysts, etc. Also biochemistry and biophysics are fields where molecular dynamics made great progress contributing, for example, to understand biocatalysis and enzymatic reactions, protein folding and amyloid fibrils aggregation. We envisage that in future simulations can contribute to establishing the foundation and, at the same times, pushing the boundaries of even more applicative fields, such as engineering. Nanofluidics is one of the fields that might act as a playground. Indeed, the length and timescales of this problem, though large/long, is within the reach of modern atomistic simulations. Should atomistic simulations be successful in this field, novel frontiers will open up. This could led to the development of continuum/atomistic multiscale simulations and, on the atomistic side, to another level of multiscaling allowing, for example, the design of fluidized bed nanoreactors. We are anxious to see progresses along these and other directions and hope the readers of this review feel inspired to become part of this Community and contribute to the field.

Appendix A Machine learning and atomistic simulations

Machine learning (ML) is a field at the intersection of computer science, numerical mathematics, probability and statistics, that empowers systems to detect patterns and make decisions by learning from data. Unlike traditional rule-based programming, where humans explicitly define the rules for a system, machine learning allows computers to learn from data and improve their performance over time. This ability to learn and adapt is what sets machine learning apart as a powerful tool for solving complex problems and making sense of vast datasets. More concretely, machine learning (ML) aims to solve many problems including but not limited to: function approximation, classification, clustering, ranking, and manifold learning among others [112]. Over the past two decades, ML, and particularly deep learning, has yielded a plethora of noteworthy outcomes [113] and is becoming more and more an enabling tool for the computational physicist. The inherent flexibility of machine learning methodologies enables their application across a wide array of sub-domains within physics itself. Machine learning methods play nowadays a pivotal role in supporting atomistic simulations across various levels and tasks, allowing analyzing simulative data [114] and accelerate ab initio simulations by orders of magnitude [115].

In the subsequent sections of this Appendix, we succinctly guide the reader through the diverse possibilities offered by machine learning methods for atomistic simulations. We concentrate on paradigmatic and simple applications by presenting for each discussed task one machine learning approach/method which we consider as prototypical and explanatory. We start with an introduction on the machine learning main features and tasks; we introduce also to one of the core assets of machine learning methods namely neural networks. Next we discuss applications to important aspects

of a simulation campaign, such as solving partial differential equations and emulating potential energy functions. Subsequently, we shift our focus to free energy, followed by a discussion of methods for the analysis and post-processing of simulations trajectories. Finally, we address the roles of software, data, high-performance computing, and present some open challenges.

A.1 Machine learning basics

The machine learning field is characterized by some distinct features and commonly found nomenclatures:

- At the core of machine learning is the concept of learning from data. Algorithms are designed to identify patterns, relationships, and trends from datasets to support predictions or decisions on new, unseen data. Hence, it is an inductive process.
- Input data often are numerical and they appear to the algorithm as a matrix X of n samples (rows) each characterized by m *features* (columns). Each sample x_i is hence a point, or vector, in $\mathbb{R}^m$.
- When to each input sample x_i is associated a response variable, y_i, either a scalar or a vector, one talks of *supervised learning* i.e. with the predefined goal of predicting y for x. In absence of such response variables the algorithm is said *unsupervised* and other tasks are possible.
- Machine learning algorithms serve as computational tools for learning. These include (linear) regression, decision trees, support vector machines, neural networks, and many others [112]. Models built using these algorithms adapt to the data they are exposed to, capturing complex relationships and making predictions or classifications.
- The *training* phase involves exposing the machine learning model to examples to learn the underlying patterns. Evaluation is performed on separate datasets to assess the model's ability to work with new, unseen data. Iterative refinement is common to enhance performance. This phase is often realized through the optimization of a loss function, $\mathcal{L}$, a function which measures the error associated to the chosen task.
- Machine learning encompasses various types of learning approaches beyond supervised and unsupervised learning. For instance Reinforcement learning involves training models to make decisions through trial and error, receiving feedback in the form of rewards or penalties.

To clarify better these aspects, we can discuss some learning tasks. In function approximation, which is often referred to as regression in the ML Community, for instance one wants to *learn* (approximate) a function which is able, given a certain set of input samples to estimate an output, for instance a scalar value. In the simplest case, the function to be learnt is a linear one.

In classification instead, one wants to distinguish between two and more classes; hence, classification is a regression problem where the predicted outputs are numerical labels (without any order relation).

In clustering, one does not have explicit labels (or they are ignored) and data are grouped according to a metric (a measure of coherence among samples) to create new labels according to the found groups.

Further in data projection, visualization, and manifold learning, one wants to find a sub-space which well recapitulates the data for computational or visualization purposes or both.

More recently, generative learning has been also proposed; in this paradigm, samples are not classified nor regressed. Instead, they are generated. In other words, during the learning phase, the algorithm learns the distribution of the data. Then upon training, the learnt model is able to draw samples from the learnt distribution. This is useful because the samples can be complex objects belonging to an intricate distribution, whereas the sampling for generating the samples, in such methods, can come from a much simpler distribution (e.g. Gaussian) whose samples are a posteriori transformed to the ones of interest.

At the technical level, often, several of these tasks can be completely or partially addressed by means of the so-called deep neural networks. Hence it is worth describing them as a first example of machine learning algorithm. A neural network, essentially, is a function approximator (to be defined below). The attribute *neural* comes from the resemblance of the networks nodes to biological neurons. The term *deep learning* instead stems from the fact that multiple subset of nodes, called *layers*, are used together to approximate the function of interest.

To understand what a neural network is, one can start from a simple example. Let assume that the function $f(x) : \mathbb{R}^m \rightarrow \mathbb{R}$ to construct the approximation is linear, $f(x; w) = wx + b$, where w is a vector of m free parameters, called *weights* and b is a scalar offset which is dubbed *bias* and is again a free parameter. By wx, we mean the dot product of x and w. In order to solve this linear regression task, one can solve the following optimization problem:

$$\min_w \mathcal{L}(X, y; w) \equiv \min_w \sum_{u=1}^{n} \left(\left(\sum_{j=1}^{m} x_{uj} w_j + b \right) - y_u \right)^2 \tag{A1}$$

where $\mathcal{L}(X, y; w)$ is the previously mentioned loss function, it measures the error between the prediction on the u-th sample $f(x_u) = wx_u + b$ and the observed, possibly noise affected, output y_u. Note our convention: when only one index appears on x, we will mean the sample index not the feature index. Upon optimization, one obtains the optimal b^* and w^* that can be used to perform predictions on new data, e.g. $\hat{x}$ not belonging to the training set, as per $f(\hat{x}) = w^*\hat{x} + b^*$. By this parametric inference, we have solved a first elementary machine learning problem.

We can generalize the prediction function $f(x)$ to a much more powerful form, but this will require a more complex construction of $f(x)$ and a significantly different optimization problem. We first discuss the construction of $f(x)$ and next how to optimize its parameters.

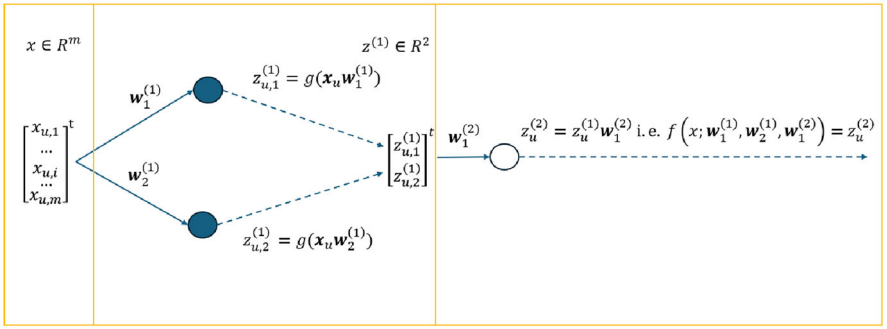

Fig. 13 A simple feed-forward neural network with evidenced neurons (blue/white circles), their weights, connections and layers. A blue filled circle indicates a neuron with non-linear activation function, a white one without non-linear activation. A continuous arrow indicates that a vector is transferred, a dashed arrow indicates a scalar quantity. Squared brackets indicate that the scalar output of the neurons are concatenated into a single vector (i.e. $z^{(1)}$). The row vector x_u (belonging to the matrix X, represented as a column for easy of reading) enters from the left, after the first layer it gets transformed in a 2d vector $z^{(1)}$, lastly a linear output layer delivers $f(x; w_1^{(1)}, w_2^{(1)}, w_2^{(2)})$

The first ingredient in the construction of our new $f(x)$ is the main building block which we will call neuron. A neuron is defined by the following function:

$$g(x; w) = g(wx) = g(\tilde{w}\tilde{x} + b) \tag{A2}$$

where the bias term, without loss of generality, has been absorbed in w, as one can always concatenate the representation $\tilde{x} \in \mathbb{R}^{m-1}$ with an extra feature of constant value 1. Moreover $g(x; w) : \mathbb{R}^m \to \mathbb{R}$ can be perfectly a non-linear function applied on wx (often a sigmoid as per $g(x; w) = 1/(1 + \exp(-wx))$. This last expression is called *neuron* in the ML community because this function represents a very approximate functional representation of a biological neuron; indeed, each of the m input features (input stimuli to the neuron) is weighted by a component of a vector w (synaptic weights) and in turn summed and processed by the non-linear function (the *activation* function).

A neural network, in its most general definition, is a graph of neurons that are connected through the inputs and the outputs of the g functions. The approach at the basis of the feed-forward neural network, a widely used subclass of neural networks, is to ingest the u-th input sample $x_u \in \mathbb{R}^m$ and through successive vectorial transformations arrive to a last representation which ultimately is a scalar and is the output of the function $f(x)$. The sample x_u enters the network and gets transformed into intermediate (latent) representations $z_u^{(1)}, z_u^{(2)}, \ldots$ several times through subsequent stages, called layers. The final layer allows to get the output of our function $f(x)$.

To understand how $f(x)$ is obtained, it is beneficial to make a simple example before making the feed-forward network formalization fully general. Suppose we have a network with a single layer and two neurons (see Fig. 13).

The sample x enters in the first layer and in the first layer two neurons process this datum; those neurons compute $g(x; w_1)$ and $g(x; w_2)$. Those two functions, as we have

seen, output each one, one scalar value. Let's call these scalar values $z_1^{(1)} \equiv g(x; w_1)$, $z_2^{(1)} \equiv g(x; w_2)$. Now we concatenate these two values as $z^{(1)} = [z_1^{(1)}, z_2^{(1)}]$ where the superscript means layer index. Now $z^{(1)} \in \mathbb{R}^2$. This $z^{(1)}$ is the result of the first layer. At this stage this vector is a 2d version of the original m-size vector x. How to obtain a scalar output? It is sufficient to add a so-called output neuron whose g function is simply wz:

$$f(x; \mathcal{W}) \equiv w_1^{(2)} z^{(1)} = w_1^{(2)}[g(x; w_1^{(1)}), g(x; w_2^{(1)})] \tag{A3}$$

where $w_1^{(2)}$ is the vector of size 2 of this just added output neuron, and $\mathcal{W} \equiv \{w_1^{(1)}, w_2^{(1)}, w_1^{(2)}\}$, namely the full set of variables which parameterize the function f of x. This simple example teaches us already some key aspects of a neural network:

- Neural networks use latent maps. In our example, we start from x and obtain $z^{(1)}$. The hope is that (see later) we can choose $\mathcal{W}$ in a so smart way, such that $z^{(1)}$ is easier to be used to get the scalar output of $f(x)$ similar to y rather than using x directly (that is doing a direct regression).
- The effect of a neuron, such as $g(x; w)$, is to create indeed a non-linear scalar projection of x. Indeed xw computes the projection of x over w and g applies a non-linearity. If we use more neurons in a layer, we can build several of these projections. Concatenating these projections, create our intermediate layer which brings us from x to $z^{(1)}$.
- The function $g(x; w)$ can be interpreted as an operation which quantifies the similarity of x and w. Indeed the scalar product xw measures the similarity (alignment) of two vectors. This gives us an interesting interpretation of a layer. A representation obtained on a layer is the concatenation of projections, and each projection, if $\mathcal{W}$ is chosen smartly, can capture different aspects of x. Indeed if w_1, w_2 are different enough, the projections xw will capture different combinations of features of x. This means that $g(w; x)$ are powerful tools to capture different aspects of x. This is beneficial because combinations of features of x (or z) are possibly more expressive than x itself to finally obtain a function $f(x)$ which fits the reference values y_i.
- The output neuron is a simple dot product $w_1^{(2)} z^{(1)}$. That is, very often, as output neuron, we can have a neuron where there is no need of a last non-linearity. A non-linearity as a sigmoid can be useful to interpret the output as a probability value. For instance, in bi-class classification where $y_u = 1$ or 0 is useful to deliver the probability that a sample x_u belongs to a given class.

Now we are ready to devise a general feed-forward neural network. Figure 14 shows such a general example of a feed-forward neural network and how we assemble neurons. From now on, we will denote through a superscript in brackets the layer index, by the first lower index the neuron weights index, and by $m^{(k)}$ the k-th layer size (i.e. the number of neurons). In general, a layer takes as input the data representation $z^{(k-1)}$ of size $m^{(k-1)}$ at the stage k and outputs a new representation of size $m^(k)$, call it $z^{(k)}$. This transformation is obtained using the previously defined neuronal units; indeed, each layer has $m^{(k)}$ neurons. We recall that a neuron is a unit g which accepts as input

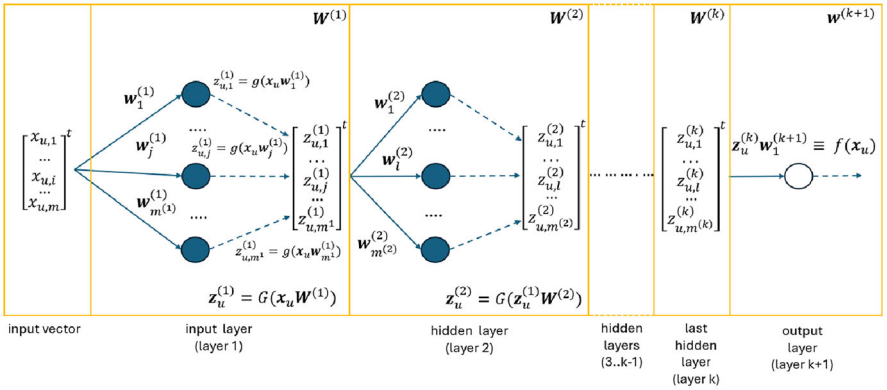

Fig. 14 A feed-forward neural network with evidenced neurons (blue/white circles), their weights, connections and layers. A blue filled circle denotes a neuron equipped with a non-linear activation function, a white one without non-linearity (i.e. the output one). A continuous arrow indicates that a vector is transferred, a dashed arrow indicates a scalar quantity. Squared brackets indicate that the scalar output of the neurons are concatenated. We report also the weights matrix on top (see text) and the layer-wise transformations in matrix form (see text). The sample x_u is the u-th row of the data matrix X, hence in the graph is transposed

x (or z), is parameterized by w, and returns a scalar value. Hence, by concatenating (i.e. aggregating the scalars in a unique vector), the output of all neurons belonging to the same layer, one can get a new representation of a datum x. This layer operation can be repeated an arbitrary number of times. As we have seen on our example, one finally generates $f(x)$ by performing the scalar product of the last layer representation z with a unique weight vector without any non-linearity (see Fig. 14).

Note that at this point, we don't know which are the optimal values for w; for the time being, we just observe that they exist and can be used to produce in principle a highly parametric function $f(x; \mathcal{W})$.

In a feed-forward neural network, the first layer is called input layer and the last one is called output layer. The ones in the middle are called hidden layers. Formally, the j-th component of the k-th layer representation for the u-th sample x_u (datum) is:

$$z_{u,j}^{(k)} = g\left(\sum_{l=1}^{m^{(k-1)}} z_{u,l}^{(k-1)} w_{j,l}^{(k)}\right) = g\left(\sum_{l=1}^{m^{(k-1)}} w_{j,l}^{(k)} g(z_u^{(k-2)}; w^{(k-1)})\right) \tag{A4}$$

where in the first layer by definition $z_u^0 \equiv x_u$ and assuming the output neuron (we recall we don't apply g in this last step and $m^{k+1} = 1$) in the $k+1$ layer, one has in the end:

$$z_{u,1}^{(k+1)} = \sum_{l=1}^{m^{(k)}} z_{u,l}^{(k)} w_{1,l}^{(k+1)} \tag{A5}$$

We can switch also to a matrix, more compact, notation to define our transformations. We remind that each weight vector of the layer k is of size $m^{(k-1)}$. Hence, we can

assemble a weight matrix $W^{(k)}$ for each k-th layer where each column is a weight vector of the neurons of the layer. Such matrix, hence, has size $(m^{(k-1)}, m^{(k)})$. Also we define $G : \mathbb{R}^{m^{(k)}} \to \mathbb{R}^{m^{(k)}}$, a function which accepts as input a vector and for each component of the vector, it outputs the value of g for that component. This is a compact way to obtain the output of all g functions together in an output vector from G which is already the concatenation of the g outputs which we need. For example, as said, w_1 and w_2 the first and second column of the m by 2 matrix W, $G(xW)$ is equal to $[g(x; w_1), g(x; w_2)]$. With these premises, we can compactly write a neural network until the output as:

$$f(x_u; \mathcal{W}) \equiv z_{u,1}^{(k+1)} = G_{m^k}^k(\dots G_{m^2}^2(G_{m^1}^1(x_u W_{mm^1}^1) W_{m^1 m^2}^2) \dots W_{m^{k-1},m^k}^k) W_{m^k,m^{k+1}}^{k+1} \tag{A6}$$

where W^{k+1} is of dimension m^k by 1 and is equal to the column vector $w_1^{(k+1)}$ (see Fig. 14), our output single weight vector.

A neural network is hence the systematic computation of these representations where the last layer is interpreted as the output, which in Fig. 14 is assumed to be scalar.

Now it remains open the problem of obtaining good parameters $\mathcal{W}$. This is obtained again via optimization:

$$\min_{\mathcal{W}} \mathcal{L}(X, y; \mathcal{W}) = \min_{\mathcal{W}} \sum_{u=1}^{n} (f(x_u; \mathcal{W}) - y_u)^2 \tag{A7}$$

As a very last note, we can use directly the matrix form of x, namely X. This is very convenient computationally. Indeed, so far we have analyzed one datum only at time, $x \in \mathbb{R}^m$. That datum goes through the network to output $f(x; \mathcal{W})$. However, we can define a function $F(X_{nm}; \mathcal{W}) : \mathbb{R}^{nm} \to \mathbb{R}^n$, where the input is the full data matrix X_{nm} and the output is a vector of size n where the u-th entry is the output of f evaluated on the u-th input point. This means that $F(X; \mathcal{W})$ gives $[f(x_1; \mathcal{W}), \dots, f(x_n; \mathcal{W})]$. We have also (note the bold character) $\mathbf{G}^k : \mathbb{R}^{(n,m^{(k)})} \to \mathbb{R}^{(n,m^{(k)})}$ can accept as input an entire matrix of size n by $m^{(k)}$ (e.g. $Z_{nm^1} = X_{nm} W_{mm^1}^1$) and gives as output a matrix of the same size where each element is the output of g evaluated on each entry of the input matrix. This rather convoluted definition actually boils down to the following matrix $\mathbf{Z}_{nm^k}$ output of the function $\mathbf{G}_{nm^k}^k(\cdot)$:

$$\begin{bmatrix} g(z_1^{(k-1)}; w_1) & \cdots & g(z_1^{(k-1)}; w_{m^k}) \\ \vdots & \ddots & \vdots \\ g(z_n^{(k-1)}; w_1) & \cdots & g(z_n^{(k-1)}; w_{m^k}) \end{bmatrix} \tag{A8}$$

Within this setting, we can write:

$$F(X;\mathcal{W}) = \mathbf{G}^k_{nm^k}(\ldots \mathbf{G}^2_{nm^2}(\mathbf{G}^1_{nm^1}(X_{nm}W^1_{mm^1})W^2_{m^1m^2})\ldots W^k_{m^{k-1}m^k})W^{k+1}_{m^km^{k+1}} \tag{A9}$$

The computational advantage of this form is that we can express several operations as matrix multiplications which are very efficient, particularly on modern graphical processors units (GPUs). The optimization problem could be even more compactly written as:

$$\min_{\mathcal{W}} \mathcal{L}(X, y; \mathcal{W}) = \min_{\mathcal{W}} ||F(X;\mathcal{W}) - y||^2 \tag{A10}$$

where y is the vector of components y_u.

At this stage, it should be intuitive that neural networks are powerful approximators as they can stack layers of neurons where each neuron is hierarchically expanded. Such hierarchical expansion confers to the network a remarkable flexibility in representing the desired function. Also, at this point, it should emerge that it is rather difficult to solve the related optimization problem for several reasons including but not limited to the non-convexity of the loss function, the possible high number of layers and neurons and the possibly high dimensionality of the training set X both in terms of number of rows and columns. All these aspects already bring to the stage (see later) the need of very-high-performing computers. Without the proper machinery, solving such problems efficiently is hopeless.

Going back to the representation ability of our network, there are rigorous results [116] which show that a neural network is a so-called *universal function approximator*. This means that it can approximate any function to arbitrary precision. It can be shown that a neural network can provide functions able to perform correct predictions on new data if proper conditions hold true [116, 117]; this result has been proved for one layer neural networks, but it holds also for more complex ones.

A word of wisdom is needed at this stage; indeed the universal approximation capability of a network tells nothing about the predictive capability, or as said in the ML community *generalization* capability, of a specifically parameterized network. That is obtaining zero error at training time, for any finite sample size, is by far not sufficient for giving a warranty on the error on unseen samples. One can easily understand such a situation via an example. Suppose one has a learning machine that *learns* a sampled function by placing a delta of Dirac at each training point. This machine will get exactly 0 training error, but has no generalization capability outside of training points. This machine is memorizing, not learning.

The only case in which zero error on the training set always implies generalization is the one for which the training set has an infinite number of samples, namely when we know everything about the data distribution. When the training set size is finite instead, there is always a risk of *over-fitting*, that is fitting a model that works on the training set only. Hence, minimizing a loss function only on the training set is most often not sufficient. Indeed one often combines the loss function with a so-called regularization term; this term encodes an assumption, a restriction, on the space of obtainable representations of the network which renders the learning process robust.

For a function approximation problem, for instance, the smoothness with respect to the input could be a desirable property. This can be achieved, for instance, by penalizing the approximating function derivatives. Hence, in this case, the derivative would be the desired regularization term. For an arbitrary regularizer, one hence solves the following augmented minimization problem:

$$\min_{\mathcal{W}} \mathcal{C}(X, \mathcal{W}) = \min_{\mathcal{W}} \mathcal{L}(X, \mathcal{W}) + \lambda \mathcal{R}(\mathcal{W}) \tag{A11}$$

where we minimize a cost function $\mathcal{C}$ composed of a loss term which accounts for the adherence of the model to the given data X (e.g. a square loss) plus a regularization term $\mathcal{R}$ and where $\mathcal{W}$ represents the whole set of weights of all the layers of the network; lastly λ is a positive scalar which rules the importance of the regularization with respect to adhering to the observed data (the loss). This λ is often called, hyper-parameter, as at variance of w it does not vary during the minimization. In detail, $\lambda = 0$ means ignoring regularization and λ tending to infinity means ignoring input data. For instance *weight decay* [118] is an effective regularization procedure which allows to avoid over-fitting. By *weight decay*, we indicate a technique to shrink the absolute value of the weights; this can be achieved for instance by jointly minimizing the L_2 norm of the weights vectors together with the loss. The reason why this regularization technique renders overfitting less probable is because it avoids the weights to perfectly fit the loss function and the y. This is particularly true if y are affected by noise. Hence, we ask to shrink the weights toward 0 to avoid that the weights fit too much (indeed over-fit) the values y. As an example, for a two-layers neural network the optimization problem, assuming a squared loss and weight decay regularization, would result in the following problem:

$$\min_{\{w^{(1)}\}, w^{(2)}} \sum_{(x,y)\in(X,Y)} \left(\sum_{l=1}^{m^{(1)}} w_{1,l}^{(2)} g(x^{(0)}; w_l^{(1)}) - y \right)^2 + \lambda \left(\sum_{l=1}^{m^{(1)}} ||w_l^{(1)}||^2 + ||w^{(2)}||^2 \right) \tag{A12}$$

where the sum in the loss is taken over the matched pairs of samples and output (x, y) in the training set.

To understand that weight decay, and particularly L_2 regularization is beneficial also numerically, here we discuss the simple case of a regularized linear model. This problem is a so-called Tikhonov regularization problem [119]:

$$\min_{w} \sum_{u=1}^{n} (w x_u - y_u)^2 + \lambda ||w||^2 \tag{A13}$$

where n is the training set size. Tikhonov in 1977 originally created that tool to stably solve systems of discretized differential equations arising in inverse physical problems. The fact that regularization is beneficial numerically can be seen by looking at the closed form solution expression of the previous problem. The cost function in matrix form is $||Xw - y||^2 + \lambda ||w||^2$; to minimize it, one takes the gradient and sets

it to zero. In this lucky case, everything can be done analytically. Indeed, the gradient is $2X^t(Xw - y) + 2\lambda w$ and setting it to zero leads to:

$$w = (X^t X + \lambda I)^{-1} X^t y \tag{A14}$$

where I is the identity matrix and X is the data matrix. The higher λ the better the inversion stability (as $X^t X$ becomes diagonally dominant) but also the different will be the result from the original inversion problem. Hence, it is evident here that a $\lambda > 0$ is highly beneficial numerically.

As we have said before, regularization dramatically reduces the probability of overfitting. Hence, the interesting fact here is that $\lambda > 0$ from the Tikhonov view point is a numerical stabilizer only, whereas from the machine learning perspective, it allows to get a well-predicting function [117]. Indeed, it is known [112] that, in practically any real-world case, in order to get a robust neural model, one needs a not null value of λ. In other words, numerical stabilization means also allowing a learning process to happen properly. Tikhonov perspective was not the machine learning one, but paradoxically, it predates much later devised regularization techniques (e.g. weight decay) for machine learning problems [120].

Before concluding this introductory section, it is worth noting that solving numerically the proposed cost functions can be by far not trivial. In the case of Tikhonov regularization, the cost function is convex and the solution is unique. Hence, as we have seen, a linear system emerges. When we switch to multi-layers networks, the problem solution becomes more intricate and there is no way to solve the problem exactly. In that case, one has a system of non-linear equations and the problem has many local minima; this requires iterative gradient-based methods. To support these methods, one can show that by the chain rule of derivatives, one can express analytically the gradients of the cost function with respect to all the weights in the network. This approach is called *back-propagation* [121]. The main iterative methods used to minimize the cost are from the so-called stochastic gradient descent family. In this family of methods, one performs gradient descent by approximating the true gradient, $\nabla\left(\sum_{i=1}^{n} \mathcal{L}(x_i, y_i; \mathcal{W})\right)$, via a random subset of the training set of size $\hat{n}$, called *batch*, i.e. $\nabla \sum_{i=1}^{\hat{n}} \mathcal{L}(x_i, y_i; \mathcal{W})$, at each iterative step. This is reasonable because loss functions are additive with respect to training samples; this assures that the error with respect to the true gradient can be bounded. The minimization proceeds using this approximate gradient; the cycle repeats until all the batches of samples are processed. This concludes one iteration (*epoch*) of the stochastic gradient method. The cycling over the batches is repeated until convergence. As the network can contain billions of parameters $\mathcal{W}$ and require processing millions of samples, it is often necessary this kind of approximate minimization procedure to limit the request of memory resources.

In the following, we will often recall these basic concepts. One should consider that there are many (important) details and significant variants which bring to each method its own peculiar flavor. However, this first shallow introduction should be enough to grasp the main applications and concepts we expose from now on.

A.2 Learning PDEs and dynamical systems

In this section, we show how machine learning contributes to solving ordinary (ODE) or partial (PDEs) differential equations and how it can be used to simplify the sampling of a complex distribution as the Boltzmann one. Solving ODEs or PDEs is useful in many cases in atomistic simulation. The Poisson–Boltzmann equation is a differential equation and is a typical example; it is solved to compute the electrostatic potential field in space of an atomistic system immersed in a solvent where this one is dealt via a continuum approximation (hence not representing each solvent-water-molecule explicitly). Next we examine how machine learning can efficiently emulate ab initio potentials so that standard MD can predict material properties or study biological processes at quantum accuracy and molecular mechanics speed.

Finally, we provide an overview of the utilization of normalizing flows [122, 123] for sampling the Boltzmann distribution; in these methods, one finds a map (a neural network) from a simple (normal, from which the name) distribution to a complex one (Boltzmann distribution) to accelerate the sampling process.

A.2.1 Solving a PDE via machine learning

Let's assume that one desires to solve the following PDE:

$$\frac{\partial u(x,t)}{\partial t} + N[u(x,t);\lambda] = 0 \tag{A15}$$

where

- $u(x,t)$ is the unknown solution.
- $N[u(x,t);\lambda]$ is a, possibly non-linear, known differential operator, which rules the evolution of the system and is parameterized by the scalar λ.
- $x \in \Omega$ which is the x domain.
- $t \in [0,T]$.

Given the numerical complexity and computational intensity involved in solving the aforementioned equation, one can use neural networks to enhance the efficiency of evaluating and solving the PDE for $u(x,t)$. This entails the crucial task of *learning* (approximating) the behavior of the state function $u(x,t)$, a practice commonly referred to as *physics-informed* machine learning [124]. To achieve this, a well-defined cost (error) function is established by requesting that the approximate *neural* $u(x,t)$ accurately adheres to the boundary conditions and to the solutions of the PDE. Denoting $v(x,t;\mathcal{W})$ the learned function and $\mathcal{W}$ the set of neural network weights to be determined, $v(x,t;\mathcal{W})$ can be learned by minimizing the following cost function:

$$\min_{\mathcal{W}} \frac{1}{N_u}\sum_{i=1}^{N_u}(v(x_i,t_i;\mathcal{W}) - u(x_i,t_i))^2 + \frac{1}{N_f}\sum_{i=1}^{N_f}\left(\frac{\partial v(x_i,t_i;\mathcal{W})}{\partial t} + N[v(x_i,t_i);\lambda]\right)^2 \tag{A16}$$

where the first term ensures compliance with known boundary and initial conditions (N_u training data), while the second term assures the satisfaction of the differential equation in N_f points. From this second term derives the nomenclature *physically informed*, in that the second term embeds the physical knowledge and plays the role of a regularization operator. In fact, we have seen before regularization encodes a restriction on the space of the possible solutions since one is restricting the space of feasible solutions to the ones which complies with the differential equation. Neural networks can in principle accurately approximate any solution $u(x, t)$ under any boundary and initial condition, provided the minimization process succeeds. Overall, this approach corresponds in transforming a differential equation into a variational problem (the neural network training process).

Interestingly, it is known that neural networks and differential equations are closely related to each other [125]. For a specific subclass of neural networks, called residual [125, 126], actually the correspondence is complete as it is found that training a residual network is equivalent to finding the parameters of a discretized differential equation. The name residual comes from the fact that if instead of the conventional $z = G(x; w)$, we use $z = G(x; w)+x$ where $x, z \in \mathbb{R}^m$ we can employ much deeper networks as the cost function minimization problem becomes easier. As $z-x = G(x; w)$, this quantity is called residue. The intuition behind this is that one is modeling the perturbations of the function and not the function directly. The fact that these networks are indeed discretized differential equations comes from observing that the step $z = G(x; w)+x$ can be interpreted as an Euler discretization $x_{t+1} = G(x_t; w) + x_t$ where now the layer index represents the time advancement [125]:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = G(x(t), \mathcal{W}) \tag{A17}$$

Hence, curiously, the approach of solving PDE/ODE via neural networks, for residual networks, maps differential equations to other differential equations of which now we desire to find the parameters.

Going back to our original problem, we report now an illustrative 1d example of this methodology, namely the solution of the Burger equation which arises in fluido-dynamics [124]. In one-dimensional space, the equation together with Dirichlet boundary conditions is:

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} - (0.01/\pi)\frac{\partial^2 u(x,t)}{\partial x^2} = 0 \\ &u(0,x) = -\sin(\pi x) \\ &u(t,-1) = u(t,1) = 0 \end{aligned} \tag{A18}$$

where $x \in [-1, +1]$ and $t \in [0, 1]$.

One can define the function:

$$f(t,x) := \frac{\partial v(x,t)}{\partial t} + v(x,t)\frac{\partial v(x,t)}{\partial x} - (0.01/\pi)\frac{\partial^2 v(x,t)}{\partial x^2} \tag{A19}$$

where $v(x,t)$ is our neural network. As for the general case, we optimize the neural network via:

$$\min_{\mathcal{W}} \frac{1}{N_u} \sum_{i=1}^{N_u} (v(x_i,t_i;\mathcal{W}) - u(x_i^b,t_i^b))^2 + \frac{1}{N_f} \sum_{i=1}^{N_f} f(x_i^f,t_i^f;\mathcal{W})^2 \qquad \text{(A20)}$$

where $u(x_i^b,t_i^b)$ are boundary conditions values and x_i^f, t_i^f are points used to enforce the network to adhere to the differential equation. The authors in [124] perform the minimization using between hundred and thousand points N_u, and employing the Limited Memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) [127] method, a quasi-Newton, exact gradient optimization algorithm. It is possible in this case to use such algorithm because the problem is not demanding computationally; bigger in size, problems would require batched stochastic gradient. The authors also found that if the differential equation is well-posed and its solution is unique, the method is able to solve, as it is often said, to generalize. Obviously, this in turn requires enough x_i^f, t_i^f points and sufficiently expressive (deep enough) neural network. Employing $N_u = 100$ and a 9-layer, 20 neurons each, deep neural network with hyperbolic tangent activation functions, the solution was correctly found. A nice feature of this approach is that the regularization term is rather restrictive compared to the usually mild assumptions for regular neural network. Indeed, in regular neural networks, one usually requires a smooth function; this restricts the space of the feasible solutions moderately. Having an entire differential equation enforced is much more restrictive; and as it always happens in neural networks when one can afford a strong regularization in the task, this leads to requiring less data.

A.2.2 Learning potential functions and properties

It is possible to summon machine learning for sampling the phase space through the emulation of the potential of a molecular system of interest [128, 129]. ML is used as a computational accelerator. The most compelling case in this scenario is quantum mechanical computations where it is well known that for instance, evaluating a density functional theory (DFT) [130] potential is extremely time-expensive. Evidently predicting the potential is just one of the possibilities; indeed, one could predict other molecular properties, even vectorial ones using the same or similar machinery. This flexibility, also, renders these approaches of interest to industrial applications[131].

There are countless technical possibilities for devising such potentials/properties approximations [132]; however, most of them consist of learning a scalar or vectorial function which takes as input a conveniently described feature space, z, namely a vectorial space which represents the molecular configuration of interest. An important aspect of the learning strategy is the ability of embedding invariances and equivariances in the molecular representations used to feed the approximating network/function. This last aspect is key as it corresponds to embedding the physical a priori knowledge about the problem.

Invariance for instance to roto-translations is necessary for energy, that is $E = f(x) = f(RTx)$ where R and T are roto and translations matrices, x is the molecular

configuration and f is the neural network which approximates the energy E. Equivariance instead is useful for properties as the dipole moment; for instance, v the dipole moment vector corresponding to configuration x, $v = f(x)$, one would like having $RTv = f(RTx)$, that is the roto-translation on x, leads to the corresponding effect into the output v. Other desirable properties are non-ambiguity and smoothness. By non-ambiguity, we mean that a given molecular configuration x should map to a given feature representation z and it cannot happen that two different configurations x_1, x_2 collapse to the same representation z. By smoothness, we indicate the capability of the approximating function to be continuous together with its first derivatives with respect to positions.

To make things clearer now, we explain, as an example, the network known as ANAKIN [133] which approximates DFT-derived potentials and is an enhancement of the seminal work from Behler and Parrinello [134]. In this proposal first, one writes the total energy E of the system as the sum of atom-centered energies E_i:

$$E = \sum_{i=1}^{n} E_i(\hat{x}) \tag{A21}$$

where $\hat{x}$ is the configuration around the i-th atom. This decomposition is always possible in principle in terms of its n-body contributions. Unfortunately, E_i is something we cannot usually directly fit; hence, one can try to fit E and hope that the learned E_i have some degree of generality. In principle, E_i should depend on all the atoms of the system. However, assuming only short range forces, one takes advantage of only the neighboring atoms of the i-th one. In the original Behler and Parrinello proposal, one had one single element type and a single network, in ANAKIN each atomic energy is predicted by a neural network which is atom-type specific [133, 134], that is one has one network for each element type. The ANAKIN problem is clearly much harder to solve.

To make an example, suppose we want to predict the total energy of one H_2O water molecule through already trained networks. First, we have to remind that we need one neural network for each atom type. Hence, we will have two networks, call them $f_O(\hat{x})$ and $f_H(\hat{x})$. The total energy of the system, E_W, will be predicted as:

$$E_W = f_H(\hat{x}_{H1}) + f_H(\hat{x}_{H2}) + f_O(\hat{x}_O) \tag{A22}$$

Now a not trivial aspect that we have to consider is that it is very inconvenient to directly feed $\hat{x}$ coordinates to the neural network. This would be detrimental for several reasons; it is not invariant to roto-translations, it does not encode in any way the *centrality* of the i-th atom with respect to the other neighbors. Hence a smarter solution is needed. We search hence for a procedure to bring $\hat{x}$ into a new vector $x \in \mathbb{R}^m$ which will facilitate the learning process for the network. The activity, in machine learning, of devising a good representation x starting from non-vectorial data (or in general from data) is called *feature engineering*. That is we need to find closed form functions, or a procedure, which brings us from the raw entity (here the atom and the surrounding ones) to the input vector of the network. These specific networks take as input a

feature vector called Atomic Environment Vector (AEV). One AEV is associated and computed from a given atom (and its neighbor) of which the energy is desired. As said, these features are not learned, instead they are carefully manually designed to embed the desired invariances. The AEV is composed of a radial and angular component. The first part is instrumental to support the estimation of two-body interactions while the second for three-body ones. Note that the following relations create a restriction on the possible learned functions, this however greatly simplifies the learning process.

Given an atom i, the l component of the AEV radial part is:

$$x_{i,l(\eta,R_0,\Phi)} = V^i_{l(\eta,R_0,\Phi)} = \sum_{j\in\Phi} \exp(-\eta(R_{ij} - R_0)^2)\ f_C(R_{ij}) \tag{A23}$$

where R_{ij} is the distance between atom j and atom i, $f_C(R_{ij})$ is a switch-off function ruled by a cut-off distance, Φ is the set of neighbor atoms for a specific atomic species (e.g. Oxygen), η and R_0 are hyper-parameters. By writing $l(\eta, R_0, \Phi)$, we underline that the l component is determined by the combination of the attained η, R_0, Φ values. Indeed, the full radial AEV is built through the concatenation of scalars $V^i_{l(\eta,R_0,\Phi)}$ by varying over the species Φ and pre-defined set of values for η and R_0. The switch-off function $f_C(R)$ is:

$$f_C(R) = \begin{cases} \frac{1}{2}\cos\left(\frac{\pi R}{R_C}\right) + \frac{1}{2} & \text{if} R \le R_C \\ 0 & \text{if} R > R_C \end{cases} \tag{A24}$$

with R_C that is a predetermined cut-off distance.

Analogously, one can define an angular part, where the l component is:

$$x_{i,l(\theta,\zeta,\eta,\Phi,\Lambda,R_0)} = H^i_{l(\theta,\zeta,\eta,\Phi,\Lambda,R_0)} = 2^{1-\zeta} \sum_{j,k\in(\Phi\times\Lambda)} (1 + \cos(\theta_i - \theta))^\zeta \times \exp\left[-\eta\left(\frac{R_{ij} + R_{ik}}{2} - R_0\right)^2\right] f_C(R_{ij})\ f_C(R_{ik}) \tag{A25}$$

Again we vary along all the possible combinations of θ, ζ, η, Φ, Λ, R_0 to get the full angular component. The complete AEV is hence assembled as the concatenation of $V^i_{l(\eta,R_0,\Phi)}$ and $H^i_{l(\theta,\zeta,\eta,\Phi,\Lambda,R_0)}$ varied over the above cited free hyper-parameters. In Fig. 15, we show such networks and the prediction of the energy for a water molecule.

As the AEV computation only involves distances, and being angles obtainable from them, the resulting molecular representation is roto-translationally invariant. Finally, these AEVs are used as input vectors to multi-layers, atom-type specific neural networks which are requested to fit DFT energy and forces. Given a molecular configuration x, the fit is obtained through a squared, λ-weighted, loss function:

$$\mathcal{L}(x) \equiv \frac{(E(x;\mathcal{W}) - E(x))^2}{\sqrt{N}} + \lambda \frac{\sum_{i=1}^N \sum_{j=1}^3 (\nabla_{x_{ij}} E(x;\mathcal{W}) + f_{ij}(x))^2}{N} \tag{A26}$$

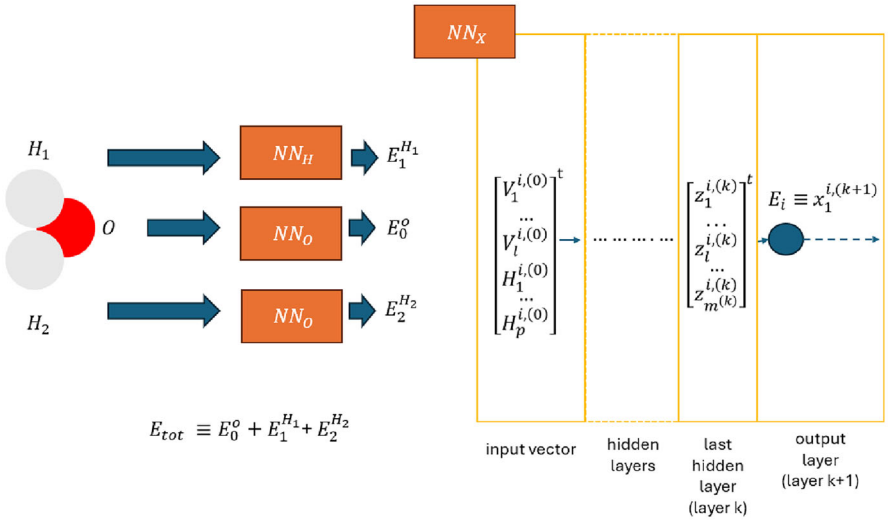

Fig. 15 Architecture of ANAKIN and example energy estimation. Each atom and its neighbor produces an atomic environment made by the V^i (Eq. (A23)) and H^i (Eq. (A25)) vectors. An element-specific neural network predicts E^i for the i-th atom. The energy associated to a molecular configuration is finally estimated as the sum of the atomic energies E^i. At training time, instead, energies and possibly forces are fitted via a squared loss to DFT values

where $E(x; \mathcal{W})$ is the estimated energy as sum of atomic energies, N is the number of atoms in the molecule, and f_{ij} are DFT forces; when these are not available in the training dataset the forces pertinent part of the cost is removed. Note that forces are predicted by direct derivation of the network which can be obtained analytically and automatically through modern software libraries. ANAKIN, but in general, this class of methods, was initially mainly concerned with the modeling of short range interactions due to the inherently local nature of the approach. Hence, electrostatics has been dealt only more recently [129] and still represents a significant challenge. To clarify the problem we can mention one of the possible approaches for electrostatics. To properly estimate long-range electrostatic interactions, one can predict atomic point charges and then add the electrostatic component explicitly through the Coulomb equation, taking care of avoiding double counting electrostatics in the short-range component. This solution is conceptually easy, yet estimating point charges is far from trivial as it is an inverse problem, which is traditionally solved only by proper regularization procedures [135]. Paradoxically, one possibility is to use further neural networks, that, given the local atomic environment predict the point-wise, time-fluctuating, local charge. These example solutions should show the significant difficulty in applying the proper electrostatic and in general long-range corrections to energy [136].

There are also important purely computational aspects to be taken into account when designing such networks. First, once learned, the function predicting the energy (or property) and forces should be fast and numerically stable over atomic displacements (e.g. for molecular dynamics). Second, as in principle, the training procedure is life-long [137] because of the virtually infinite training set size (i.e. chemical space), the

method should be able to retain past experience when updated, at least to some extent. This is a rather crucial aspect as it is not realistic to train such models all at once if there is any ambition of generality in terms of structures or chemical elements. It is evidently a process that goes through time and many training sessions. This as opposed to specialized applications where a once-and-for-all strategy may prove sufficient, for instance when a limited set of elements is needed. This last scenario corresponds to the vast majority of papers in the literature. So-called *foundational models*, which are universal reference trained networks, are starting to appear recently [138–140]. These networks should allow to be quickly (relatively few samples) customized to specific problems taking advantage of a large previous pre-training phase.

A.2.3 Sampling the Boltzmann distribution via neural networks

The Boltzmann distribution, despite being its form analytically known, poses tremendous challenges when one wants to estimate equilibrium properties. Formally, the probability density is:

$$\mu(x, p) = \frac{\exp(-\mathcal{H}(x, p)))}{\int_{\mathcal{X}\times\mathcal{Y}} \exp(-\mathcal{H}(x, p))\mathrm{d}p\mathrm{d}x} \tag{A27}$$

where $x \in \mathcal{X}$ is the molecular configuration, $p \in \mathcal{Y}$ is momentum, $\mathcal{H}$ is the adimensional Hamiltonian (the potential plus kinetic energy divided by $\beta = 1/(k_\mathrm{B}T)$) of the system, $\mathcal{X} \times \mathcal{Y}$ is the accessible phase space. Note that for self-consistency with the whole Appendix we use this slightly different notation with respect to the main text. In detail (x, p) is $(\mathbf{R}, \mathbf{p})$ and $\mu(x, p)$ is $\rho(\mathbf{R}, \mathbf{p})$ respectively in the Appendix and the main text. As $\mathcal{X} \times \mathcal{Y}$ can be huge, $\mu(x, p)$ is remarkably complex and can include high barriers which hinder the sampling of various minima energy basins. Also the source of difficulty is sampling x and not p which is distributed as a Gaussian. For this reason, neither plain Molecular Dynamics nor brute force Monte Carlo is sufficient to sample thoroughly $\mu(x, p)$. This last aspect calls for smart strategies to sample the phase space of a physical system. This can be partially achieved by taking advantage of so-called enhanced sampling methods. These methods allow to sample areas of $\mathcal{X} \times \mathcal{Y}$ of low probability by, for instance, adding biasing forces in molecular dynamics (see Section 4 in the main text).

Beside these biasing methods, recently another strategy emerged [122, 123]. The main, and general, idea is to simplify sampling by learning a mapping function, $x = f(z)$, between data sampled from a simple distribution (e.g. a Gaussian one, in the domain z) and a complex one (e.g. the Boltzmann one with domain x). One, hence, samples from the simple distribution getting a z sample and converts it to the x representation; this allows to efficiently sample the complex distribution. This class of methods is called normalizing flows [122] as they map a complex distribution to a normal one. As the function bringing z to x can be quite complex, here is where neural networks can be useful. At variance of our previous networks, now the desired output is vectorial (z) and not scalar; this implies that we have many output neurons, not just one as before. The space associated to z is often termed *latent* as it is learned and hence discovered upon training.

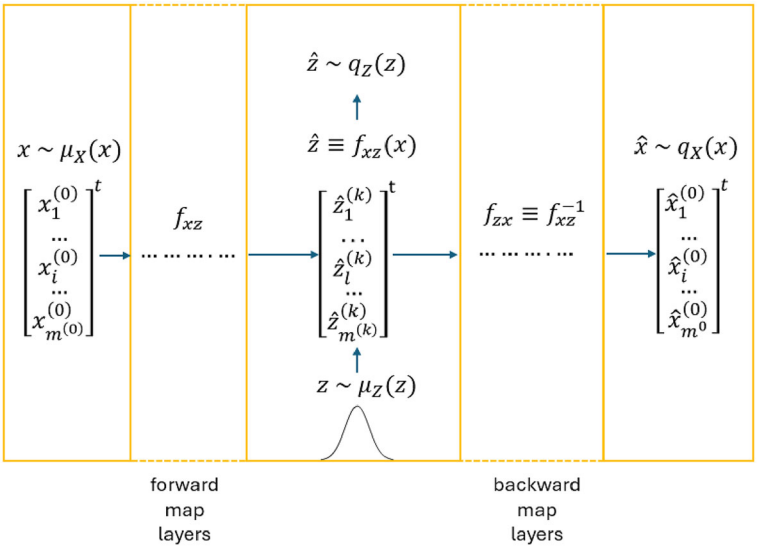

Fig. 16 Architecture of a Boltzmann generator. Data, x, is sampled from the Boltzmann distribution and converted to the latent representation z via the invertible, learned, transformation $z = f_{xz}(x)$. We underlined the fact that k-layer transformation is interpreted as z. By $\hat{x}, \hat{z}$ we denote the random variables associated to the learned distributions which are approximations of the true ones. The backward transformation is realized by the same network but used in the reverted way. For training one matches the distribution $\mu_X(x)$ with $q_X(x)$ and $\mu_Z(z)$ with $q_Z(z)$ respectively

In most machine learning applications of normalizing flows, the target distribution (x domain) is unknown, whereas in the physical case, we know the target distribution, namely the Boltzmann one. One can hence take advantage of this knowledge explicitly providing further interesting properties. The application of learned normalizing flows to the Boltzmann distribution case has been given the name *Boltzmann generator* [123]. Here we formally introduce such Boltzmann generators. We define the following symbols:

- z is a Gaussian random variable (the simple distribution).
- x is the Boltzmann distributed random variable (the target distribution).
- $\mu_X(x)$ is the exact Boltzmann distribution, as before, and $q_X(x; \mathcal{W})$ is its learned approximation, i.e. the the distribution associated to the neural network transformation $z = f_{xz}(x; \mathcal{W})$ which brings the data x to the their latent z counterpart; $\mathcal{W}$ are the associated weights.
- $\mu_Z(z)$ is the exact Gaussian distribution and $q_Z(z; \mathcal{W})$ is the learned approximation of the Gaussian distribution, i.e. the distribution associated to the neural network transformation $x = f_{zx}(z, \mathcal{W})$ which brings from the latent domain z to the the original data.
- $u(x)$ the adimensional potential function.

In Fig. 16, we report the scheme of a Boltzmann Generator. As we desire to exactly transform x to z and vice versa, our neural network layers must be invertible. Such layers exist [141] and they are significantly different from what we discussed till now,

for our discussion is sufficient to know that is possible to get structurally invertible neural layers.

Given these premises, we have to define a cost function to optimize the parameters $\mathcal{W}$. Note that in this new setting, we don't have any specific value to be fit (e.g. an energy value). Hence, the reasoning is rather different. To define our generator, we can request to make $\mu_X(x)$ and $q_X(x)$ similar as this is the desired outcome. To this aim, we minimize the Kullback–Leibler (KL) divergence (a measure of distance between probability distributions) between them over the weights $\mathcal{W}$.

$$KL(\mu_X(x), q_X(x;\mathcal{W})) = \int_x \mu_X(x)(\log \mu_X(x) - \log q_X(x;\mathcal{W}))\mathrm{d}x \tag{A28}$$

The first addend in the integral does not depend on $\mathcal{W}$. Hence, it is irrelevant in the minimization process. Using the formula $p_X(x) = p_Z(z)|\det J(x;\mathcal{W})|$, where J is the Jacobian matrix of the neural network transformation f_{xz} and $|\det J(x;\mathcal{W})|$ is the absolute value of its determinant, for the second addend, one has:

$$\begin{aligned}
&-\int_x \mu_X(x)\log q_X(x;\mathcal{W})\mathrm{d}x \\
&\quad = -\int_x \mu_X(x)\left[\log \mu_Z(f_{xz}(x;\mathcal{W})) + \log|\det J(x;\mathcal{W}|)\right]\mathrm{d}x \\
&\quad\quad -\int_x \mu_X(x)\left[\log \frac{\exp(-u(f_{xz}(x;\mathcal{W})))}{Z_Z} + \log|\det J(x;\mathcal{W}|)\right]\mathrm{d}x
\end{aligned}$$

Note that to get this result, the invertibility of the network is fundamental because this allows to get advantage of the formula $p_X(x) = p_Z(z)|\det J(x)|$. Noting that the partition function Z_Z does not depend on weights $\mathcal{W}$, one gets the final cost:

$$\mathcal{C}_{\mathrm{ML}} = \mathbb{E}_{x\sim\mu(x)}\left\{\frac{1}{\sigma^2}||f_{xz}(x;\mathcal{W})||^2 - \log|\det J(x;\mathcal{W})|\right\} \tag{A29}$$

where σ is the standard deviation from the Gaussian distribution. The optimization of this cost function is called training by example; note that this is the classical way normalizing flows are learned in machine learning [122]. This cost function should be sufficient to train our network. However, the fact the we know explicitly the Boltzmann distribution allows also to formally define another cost function. In this case, we request $\mu_Z(z)$ and $q_Z(z)$ to be as similar as possible again via the KL-divergence. For the $q_Z(z)$ case, we have:

$$\mathrm{KL}(\mu_Z(z), q_Z(z;\mathcal{W})) = \int_z \mu_Z(z)(\log \mu_Z(z) - \log q_Z(z;\mathcal{W}))\mathrm{d}z \tag{A30}$$

As before, the first addend in the integral does not depend on $\mathcal{W}$. Hence, it is irrelevant in the minimization process. Using the formula $p_Z(z) = p_X(x)|\det J(z;\mathcal{W})|$, where J is the Jacobian matrix of the neural network transformation f_{zx} and $|\det J(z;\mathcal{W})|$

is the absolute value of its determinant, for the second addend, one has:

$$
\begin{aligned}
&-\int_z \mu_Z(z) \log q_Z(z;\mathcal{W}) \mathrm{d}z \\
&\quad = -\int_z \mu_Z(z) \left[\log \mu_X(f_{zx}(z;\mathcal{W})) + \log|\det J(z;\mathcal{W}|)\right] \mathrm{d}z \\
&\qquad -\int_z \mu_Z(z) \left[\log \frac{\exp(-u(f_{zx}(z;\mathcal{W})))}{Z_X} + \log|\det J(z;\mathcal{W}|)\right] \mathrm{d}z
\end{aligned}
$$

Removing the partition function Z_X from $\mu_X(f_{zx}(z;\mathcal{W}))$, one gets the final cost:

$$
\mathcal{C}_{\mathrm{E}} = \mathbb{E}_{z\sim\mu(z)}\{u(f_{zx}(z;\mathcal{W})) - \log|\det J(z;\mathcal{W})|\} \tag{A31}
$$

Optimizing this cost is termed training by energy. This cost function is new in the normalizing flows realm as usually the functional form of the μ_X distribution is unknown. While there is no formal reason to have such additional cost function to train the network, overall and practically, the complete cost function formed by the weighted sum of $\mathcal{C}_{\mathrm{E}}$ and $\mathcal{C}_{\mathrm{ML}}$ is found to be convenient [123]. Further extensions are possible by taking advantage of collective variables but we will not discuss them. Additionally in [123], it is noted that the learned $q_X(x)$ may still present a certain degree of statistical bias, hence they suggest a technique to de-bias it [123].

The Boltzmann Generator method was applied to toy benchmarks and to an implicitly solvated bio-molecular system. In this last case, internal coordinates were used for learning the latent representation and remove roto-translational degrees of freedom. It was found that paths in the latent space tend to map to minimum free energy paths in physical space and temperature changes boil down to different Gaussian variances. As $\mathcal{C}_{\mathrm{E}}$ is the free energy difference of the system with respect to the reference Gaussian, the authors have shown that it is very efficient to compute free energy differences of the same system in two different, not easily interconverting, conformations: this is achieved by simply training two independent Boltzmann generators and than taking the difference in the respective $\mathcal{C}_{\mathrm{E}}$ values upon training. Note that this is correct as the reference gaussian system is the same for both. Essentially this method of computing free energy is rather similar to the much older Frenkel and Ladd method [142]. In this absolute free energy method, one first approximates the system of interest through the Einstein crystal and then estimates the anharmonic contribution by means of thermodynamic integration from the the reference harmonic crystal to the true force field. The key advantage is that the reference Einstein crystal free energy can be analytically evaluated. In Boltzmann generators, there is no harmonic approximation, strictly speaking, rather we refer to each physical system as a unique, shared, latent harmonic system. This has a significant advantage for which there is no need to compute the free energy difference between different harmonic reference systems. Albeit this quantity is analytically computable, its large-scale computation can be problematic as it requires the diagonalization of the Hessian matrix when one desires the much more accurate Debye crystal instead of the Einstein one [143]. This can be challenging when using an explicit solvent which dramatically increases the number of degrees of freedom [143].

Hence, the Boltzmann generators approach offers an elegant solution to this problem, namely learning the mapping between a unique harmonic, small dimensional, reference, and the systems of interest. Also, interestingly, in Frenkel and Ladd, one has, obviously, the same system sizes of the harmonic and original system; here thanks to the methodological flexibility, the reference harmonic system can live in a smaller vector space becoming, at least in principle, independent in size from the system of interest.

A.3 Free energy estimation

Machine learning can also find a significant application in estimating free energy, a quantity particularly relevant [144, 145]. For instance, by employing learned potential functions emulating Quantum Mechanical calculations, one can next derive a free energy much faster than using ab initio potentials. As we already described machine learning potentials, here we discuss other aspects:

- the identification of collective variables through machine learning methods. By collective variables, $\theta \in \mathbb{R}^m$ from now on, we mean scalar or vectorial functions of the coordinates along which one may estimate a potential of mean force (free energy as a function of θ).
- The reconstruction of the potential of mean force $F(\theta)$ by approaching the problem as a function approximation problem.
- The accelerated estimate of free energy differences between two Hamiltonians through the transport of distributions, i.e. the so-called targeted Free Energy Perturbation (FEP) method [146].

In the following subsections, we briefly discuss these three possibilities:

A.3.1 Data-driven collective variables

As we have seen, Machine Learning methods are concerned in many cases with a (possibly highly complex and non-linear) function approximation problem. In the case of collective variables discovery, however, one is concerned with a manifold learning problem: detecting the sub-manifolds (ruled by θ) of the configurational space where the phenomenon of interest mainly occurs. By *mainly*, we mean the observables of the systems (or their combinations) which bear the most important, and recapitulating information, of the studied phenomenon. For instance, in a protein ligand-binding problem, the distance between the center of mass of the ligand and the center of mass of the binding pocket in the protein is a reasonable, although poor, collective variable θ over which one can be interested in estimating the change in free energy.

A first distinction needed to orient the reader is between methods that take advantage of labels of states (e.g. ligand-bound or unbound) and methods that don't require this additional information. The first are supervised methods, whereas the second are unsupervised, according to our initial definition. Also, some methods operate directly on coordinates, whereas others assume a wide set of existing collective variables from which a subset, or a combination, is selected.

The classical prototypical approach to identify collective variables in the unsupervised class of methods is the Principal Component Analysis method [147]. It discovers a linear operator T which projects coordinates to a new set $z = Tx$. A subset of z can be identified with our collective variables θ. In PCA, we indicate with $x \in \mathbb{R}^n$ the column vector of all the coordinates (where n is the number of chosen degrees of freedom), by $\langle x \rangle$ time averages assumed existent, and by xx^t the product of the column vector x by x^t, its transposed. For instance, PCA can identify the most relevant coordinates of a molecule, a solute, often a protein or a nucleic acid. The analysis is useful for not diffusive species as it analyzes motions around a mean configuration. In PCA, one employs the covariance matrix C constructed from coordinates deviations over their mean along a trajectory:

$$C = \langle (x - \langle x \rangle)(x - \langle x \rangle)^t \rangle \tag{A32}$$

The symmetric matrix C can always be diagonalized by an orthogonal coordinate transformation T:

$$z = T^t (x - \langle x \rangle) \tag{A33}$$

An eigenvalue decomposition of the covariance matrix C leads indeed to a complete set of orthogonal collective modes (eigenvectors in the T matrix), each equipped with a corresponding eigenvalue (λ_i):

$$C = T \Lambda T^t \tag{A34}$$

where Λ is the diagonal matrix containing the eigenvalues.

The following sum represents the total atomic displacement:

$$\sum_{i=1}^{n} \langle (x_i - \langle x_i \rangle)^2 \rangle \tag{A35}$$

$$= \langle (x - \langle x \rangle)^t (x - \langle x \rangle) \rangle \tag{A36}$$

$$= \langle z^t T^t T z \rangle = \langle z^t z \rangle = \sum_{i=1}^{n} \langle z_i^2 \rangle = \sum_{i=1}^{n} \lambda_i \tag{A37}$$

From this equation, one sees that the largest eigenvalues identify the directions of the largest fluctuations. The projection operator (we mentioned at the beginning of this section) results to be T^t. Hence, if one is interested in identifying as collective variables θ the subset of most varying coordinates, then one can select them as $\theta = T_k^t (x - \langle x \rangle)$ where T_k is the rectangular matrix obtained by taking the first k rows eigenvectors associated to the largest eigenvalues. This subset is called the principal component set. How many eigenvalues to keep is largely arbitrary. However, typical choices are based on the ratio of the sum of the kept eigenvalues versus the total sum, namely $\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n} \lambda_i$. We lastly note that while PCA in machine learning is performed

typically in space, in Statistical Mechanics time averages are instead employed as we have seen.

At variance of PCA, which stems from the concept of coordinates displacements, Time-Independent Component Analysis [148] searches for a projection matrix, T, which leads to $z = Tx$ where z are now slow degrees of freedom. The final result is that TICA could be a much more meaningful tool to be used in atomistic simulations rather then PCA as it aims at identifying long time scales. Those are typically the objective of enhanced sampling. Also it has been suggested that TICA [148] could be a better tool in supporting Markov State Models [149]. As the derivation of TICA is involved, given the introductory character of our Appendix, we will not go into its details.

Another strategy to obtain collective variables is to use the so-called *autoencoders* (see for instance [150] for an application). These neural networks aim to project a sample, x, (one sample is one molecular configuration) into a low-dimensional space z (encoding) which can then be back-converted to the original space (decoding). The space z is identified as the collective variables space. These networks are structurally analogous to a Boltzmann generator; they bear an *encoder* network $z = E(x; \mathcal{W}_1)$ and a *decoder* one $x = D(z; \mathcal{W}_2)$, yet without requiring exact invertibility: $E = D^{-1}$ is not guaranteed. The capability of approximately inverting the transformation is learned by searching for the minimum over $\mathcal{W}_1, \mathcal{W}_2$ of $\sum_{i=1}^{n} ||D(E(x_i; \mathcal{W}_1); \mathcal{W}_2) - x_i||^2$. Once learned, as the encoder can provide the projection, z, one could employ this as collective variable. Up till now, this technique has been explored only in few cases. However, it offers a potentially powerful method to characterize phenomena with a strong reduction of the dimensionality and to be able to see from which part of the dimensionality they come.

Lastly, we mention a peculiar [151] application of manifold learning to CVs; in this protocol, a curve (i.e. a one dimensional manifold) learning algorithm dubbed Principal Path [152] is used to refine a rough initial path describing a transition process of interest with given and known starting and ending points in coordinates space. This algorithm tries to obtain a so-called minimum free energy path between an initial and final state; a minimum free energy path is a path in collective variable space θ which minimizes the barriers to move from the initial to the final state. For instance, in protein and ligand binding, one may be able to generate a putative binding or unbinding path via enhanced sampling. These kinds of trajectories are however affected by the noise generated by the thermostat; hence, they don't represent smooth paths from one state to another albeit they may pass through minimum free energy barriers and regions. As such it is useful to have algorithms able to *clean* an initial trajectory and obtain a smooth collective variable, θ, which could be next employed for free energy computations. In particular, in order to learn a smooth transition path connecting a starting point (molecular configuration) w_0 to an ending point w_{n_c+1} (where n_c is the number of intermediate stations w_j between the start and end point), one solves the following problem:

$$\min_{\{w\},u} \sum_{i=1}^{n} \sum_{j=1}^{n_c} \|x_i - w_j\|^2 \delta(u(x_i), j) + \lambda \sum_{i=0}^{n_c} \|w_{i+1} - w_i\|^2 \tag{A38}$$

where $\delta(u(x_i), j)$ is the Kronecker delta, u is an indicator function which gives the index of the nearest w vector to x_i and λ is a regularization constant. By this optimization problem, one searches for intermediate molecular configurations w which are jointly near to the original trajectory configurations, x_i, (the first term in the sum), and also tries to get a smooth path (the second term). Once the w are available, then one has a series of landmarks molecular configurations, w_j, which can be used as the discretized evolution of θ from the initial to the final state. Interestingly, it can be proved [152] that the cost function just presented is a variational formulation of the so-called enhanced sampling string method [51].

So far we cited methods that take advantage of coordinates only and automatically discover collective modes, hence unsupervised methods. However, if one a priori knows that a subset of candidate collective variables is important for the phenomenon of interest and also the states of interest can be sampled, then one can use classifiers to understand from the candidate variables which are the most relevant. Candidate CV values are used as features for the input x together with state labels. The classifier builds a classification model and isolates the most contributing variables. For example, suppose one has a two state phenomenon, with labels $y_i = +1$ and $y_i = -1$ respectively. One molecular configuration x_i can hence be labeled with one of the two, and we suppose its features are the values of the candidate collective variables. One may for instance employ, in the simplest case, a linear classification model as per $f(x) = wx$ where $f(x) > 0$ means we are in state $+1$ and conversely $f(x) < 0$ means we are in state $y_i = -1$. Upon obtaining the optimal w^* by optimizing as usual $\sum_{i=1}^{n}(f(x_i; w) - y_i)^2$, one can find the most important collective variables by looking at the values of the components of $|w^*|$. The highest the value of the components of $|w^*|$, the highest the importance of that CV for discriminating the two states. Hence, one can single out the most expressive collective variables from the initial superset. This supervised scenario is a much simplified one than the previous unsupervised one where putatively interesting CVs are not known; here a significant amount of a priori information is employed. In literature, some [153, 154] classification methods from machine learning were applied to simulations to identify CVs. These include for instance Support Vector Machines for classification [155], Linear Discriminant Analysis and its variants [156–159].

Overall, these methodologies can help guiding the user in detecting which are the most relevant (often slow) modes or selecting among a pool of hypotheses the most relevant variables.

A.3.2 Potential of mean force reconstruction as a function approximation problem

Once some CVs (or a single one) have been deemed as important, then one can proceed with reconstructing the potential of mean force (PMF), or Landau free energy, $F(\theta)$. Note that from a purely mathematical viewpoint rebuilding the PMF is indeed a function approximation problem: given observations (the sampling of the phase space) one wants to find $F(\theta)$. To our knowledge, the PMF estimation task is never introduced via this viewpoint in computational chemistry/physics textbooks perhaps as it is admittedly indebted to a computer science perspective. At this stage of our Appendix, this perspective, however, becomes completely natural in light of neural networks and

their relation to inverse problems. Indeed, neural networks, as we observed several times now, are excellent function approximators, as such they are the perfect candidate for this task.

Several existing methods which reconstruct the free energy profile can be well explained by this framework even if in the original papers they were not declined in this way. For instance, in the Single Sweep method [160], the authors propose a three-step procedure: first, they explore the θ space via Temperature Accelerated Molecular Dynamics (TAMD) [61], next, they perform umbrella sampling [53] (see section 4 on free energy calculations on the main text for both methods), and finally, reconstruct the PMF (1d or 2d) via an expansion in a Gaussian basis set. For our discussion, regarding TAMD, it is sufficient to recall that is a technique which allows to explore the θ space efficiently. By umbrella sampling instead, we recall that is an enhanced sampling technique. In this method, given a point of interest θ^*, one runs a simulation with the following potential:

$$V(x) = U(x) + \frac{1}{2}k(\theta(x) - \theta^*)^2 \tag{A39}$$

where $U(x)$ is potential of the sole molecular system and $\frac{1}{2}k(\theta(x) - \theta^*)^2$ is a pseudo-harmonic bias. In this ensemble, one can show that the mean force at θ^* in the stiff spring approximation (big k) is estimated (for the demonstration see [56]):

$$f(\theta^*) = \frac{1}{T}\int_0^T k(\theta^* - \theta(x(t)))\mathrm{d}t \tag{A40}$$

next, in the Single Sweep method, one assumes that the PMF can be fitted by a weighted superposition of Gaussian functions with same variance, a parameter to be determined, and centered at, the various computed θ_i^*, with $i = 1, \ldots n$:

$$F_{NN}(\theta) = \sum_{i=1}^{n} w_i g(\theta, \theta_i^*) \tag{A41}$$

Then one has that the mean force function is estimated by:

$$f_{NN}(\theta) = -\sum_{i=1}^{n} w_i \frac{\partial g(\theta, \theta_i^*)}{\partial \theta} \tag{A42}$$

As we have the observations at θ_i^* of the mean force, then we can fit the previous model equation to these observed values of mean forces. In our language, this corresponds to learning the function $f_{NN}(\theta)$ via a shallow neural network (one layer) where activation functions are the derivative of Gaussians and where the number of neurons is equal to the number n of *probed* points. The only major difference with a traditional neural network is that as we collect mean forces samples, we now ask $f_{NN}(\theta) = -\frac{\partial F_{NN}(\theta)}{\partial \theta}$ to fit the estimated mean forces $f(\theta_i^*)$. In reference to this just suggested method [160], the neural network approach has at least two possible advantages. First one

could have used a neural network, possibly employing more layers to increase the representation power. Second, we can introduce a regularization procedure in the cost function stabilizing the capping suggested in [160].

A similar path is followed in [161] where, starting from umbrella sampling simulations, they rebuild the free energy profile via the so-called Gaussian Process Regression. GPR is rather similar to the previous procedure [160] as it employs a Gaussian basis set, apart that it gives a probabilistic procedure to determine Gaussian variances [161, 162]. For the reasons given before, both methods would benefit from the neural network approach outlined previously.

From this point of view, it should be clear that despite these methods are different in many details, they however reconstruct the PMF using the same hint, namely using a basis set of Gaussian functions centered at prescribed points.

Lastly, we dutifully remind that the most difficult step of free energy estimation is sampling and possibly correctly and stably estimating the mean force; hence, rebuilding the PMF through a neural network is still relevant but largely not the most important step of the whole procedure.

A.3.3 Targeted free energy perturbation

So far we analyzed the case where a CV is defined. In this section, instead we look for the free energy difference, ΔF_{AB}, between two Hamiltonian systems $\mathcal{H}_A$ to $\mathcal{H}_B$ [144] without any explicit CV. There are several classical *alchemical* methods to this aim [144] but we limit our discussion to the so-called Targeted Free Energy Perturbation (FEP) which is of interest for machine learning. Targeted FEP was introduced by Jarzynski [146] and the key step of the method is building a map between two probability distributions to hopefully accelerate the free energy estimation process with respect to more traditional methods. Suppose microstates x or y are sampled from the canonical ensembles A or B, $\rho(x)$ or $\eta(y)$, whose distributions is e.g. $\rho(x) = \exp(-\beta U_A(x))/Z_A$, where $\beta = 1/kT$, where k is the Boltzmann constant and T is temperature. If we suppose that there is an invertible transformation $\mathcal{M} : x \to y(x)$, then the two distributions are related by $J(x)\eta(y(x)) = \rho(x)$ where $J(x)$ is the Jacobian of the mapping $\mathcal{M}$. Explicitly we have $J(x)\exp(-\beta U_B(y))/Z_B = \exp(-\beta U_A(x))/Z_A$. Taking the logarithm, we apparently get:

$$\Delta F_{AB} = U_B(y(x)) - U_A(x) - \frac{1}{\beta}\ln J(x) \tag{A43}$$

having realized this is a probabilistic relation and not a pointwise one, we redefine the apparent ΔF_{AB} as the random variable $\phi(x)$. Then

$$\phi(x) = U_B(y(x)) - U_A(x) - \frac{1}{\beta}\ln J(x) \tag{A44}$$

Multiplying by $-\beta$ and exponentiating, the expected value on the ensemble A gives

$$\langle \exp(-\beta\phi)\rangle_A = \int_X \rho(x)\exp(-\beta\phi)\mathrm{d}x = \frac{1}{Z_A}\int_X J(x)\exp(-\beta U_B(y))\mathrm{d}x = \frac{Z_B}{Z_A} \tag{A45}$$

where the last equality comes from $J(x)\mathrm{d}x = \mathrm{d}y$. This leads to

$$\langle \exp(-\beta\phi)\rangle_A = \exp(-\beta\Delta F_{AB}) \tag{A46}$$

If the map $\mathcal{M}$ is the identity, then the relation becomes the Zwanzig equation [163] which is a simple but difficult to estimate relation tried in the early times to estimate free energy differences:

$$\langle \exp(-\beta(U_B - U_A))\rangle_A = \exp(-\beta\Delta F_{AB}) \tag{A47}$$

A well-known feature of the Zwanzig relation is that it converges when the two ensembles A and B largely overlap. The intuition here is hence that if the map $\mathcal{M}$ can bring probable regions of A to probable regions of B, then one can expect a good convergence at the expense of devising such map. It is hence evident that the success of such a proposal is completely bound to our capability of finding quickly and effectively such a good $\mathcal{M}$. Again, here is where neural networks can help as we are asking to approximate a function that brings a sample in A to another in B satisfying the conditions just above. At this stage, the reader may remember that we already faced this problem. We can take advantage of the normalizing flow as again we are dealing with the problem of mapping two probability distributions. Previously for Boltzmann Generators, the primary distribution was Gaussian, whereas here both the two distributions are canonical. Using normalizing flows based on neural networks was indeed proposed in [164] and also for instance employed in [165, 166]. These applications, among others, proved successful because indeed a fast approximate way to find the mapping was devised.

A.4 Analyzing simulations

Here we discuss briefly how already performed simulations can be analyzed. Interestingly this part, even if presented as the last one in this Appendix, represents one of the first historical applications of ML to atomistic simulations. In the following, we discuss a paradigmatic ML analysis method namely the clustering of trajectories and also visualization of trajectories via projections.

As briefly mentioned at the beginning of this Appendix clustering is a classical task in machine learning. An example and widely used clustering algorithm is the k-means method [167]. In this method, given a priori $k = n_c$ as the number of searched groups, one tries to find n_c points, w_j, that by clustering recapitulate the data. To find these points, one initially fixes w_j arbitrarily and next solves iteratively the following

optimization problem:

$$\min_{\{w\},u} \sum_{i=1}^{n} \sum_{j=1}^{n_c} \|x_i - w_j\|^2 \delta(u(x_i), j) \tag{A48}$$

where $\delta(u(x_i), j)$ is the Kronecker delta, u is a membership which gives the index of the nearest w point to x_i. This cost function optimization hence tries to minimize the average distance between samples x_i and the means/centers w_j. The cost function is not convex in w_j hence it is a hard optimization problem whose solution highly depends on the initialization. Several variants of this method exist. An example is the k-medoids [168] method where the centers have the restriction of being x_i samples from the training set. Another important variant of the method can be obtained by observing that one can combine a neural network [169, 170] obtain the representation z and perform clustering in this new space. The hope is that the space z is more descriptive and hence may facilitate the clustering process also rendering it more informative. Although for clustering is not strictly needed, using an autoencoder would allow also to convert back z to x and locate w in the x space.

Clustering finds a natural application in *summarizing* the results of a simulation and finding the main basins [171]. Indeed clustering can find the most populated states which represent the long-lasting states, hence the most thermodynamically relevant. When clustering is equipped with proper post-processing statistical tools, it can lead to quantitative insights including free energy and kinetics estimations. Indeed upon clustering, one can detect the most populated states by simply counting the number of samples belonging to each cluster (the ones thermodynamically relevant). Also, by counting the inter-conversions among states, one can take interest in kinetics. For these kinetic estimations, one can take advantage of so-called Markov State Models (MSMs) [149, 172, 173]. We mention also that deep learning extensions [114, 174] of MSMs exist; however, we omit for brevity both MSMs and their extension in this Appendix as they would deserve a too large discussion.

Another relevant task where machine learning helps is summarizing simulations via a 2D projection. The projection can help give more evidence, for instance, of the clustering results of the simulation. When one can associate a physical meaning to the projection (e.g. a collective variable), these tools can give insightful information. For projecting in a low-dimensional space, a solution is represented by multi-dimensional scaling methods [175] (MDS). MDS methods build projections from a sample x (a molecular configuration) to a new, low-dimensional, sample z, where the guiding principle is that pairwise distances between molecular configurations $d(x_i, x_j)$ should be more or less approximately preserved in the new space, that is $d(z_i, z_j) \approx d(x_i, x_j)$. The simplest form of MDS is obtained by minimizing:

$$\min_{\{z\}} \sum_{i,j,i\neq j} (d(x_i, x_j) - d(z_i, z_j))^2 \tag{A49}$$

An effective example of MDS for atomistic simulations are sketch maps [176]. In this method, one is particularly interested in preserving small distances and not high

ones, indeed a distance is reliable and meaningful only when it is small. Formally, one searches for:

$$\min_{\{z\}} \frac{\sum_{i,j,i\neq j} v_i v_j (g_1(d(x_i, x_j)^2) - g_2(d(z_i, z_j)^2))^2}{\sum_{i,j,j\neq i} v_i v_j} \tag{A50}$$

where $v_i > 0$ are parameters obtained in closed form and proportional to the population of the state v_i to guarantee that most populated states are well represented, g_1 and g_2 are sigmoids manually tuned for location and shift. The role of the sigmoids is to concentrate the fitting in regions of small distances; the sigmoid plateau allows to partially ignore the error induced by big distances and concentrate the optimization for small distances. We underline that the optimization is performed on the coordinates z as it is usual in projection methods which employ an optimization principle.

A.5 Role of data, software and high-performance computing

As we have briefly seen machine learning enables or accelerates many tasks in atomistic simulations. Nevertheless, one should wonder why today ML is so widely technologically applicable. The answer consists of three pillars: data, software, and computational capabilities.

Data is the essential starting point of any machine learning activity. Current large data storage capabilities either in premise (local) or in the cloud (Internet) are a fundamental asset. A particularly relevant aspect is not only accumulating *big data* per se but also rendering it FAIR [177], namely Findable, Accessible, Interoperable, and Reusable. To this aim, one needs to enrich data with proper metadata and provide Application Program Interfaces (API), namely interfacing functions, as universal as possible [178]. Having well-structured and large databases of datasets, so-called data lakes, is becoming more and more a game changer for machine learning.

The second key ingredient is the availability of high-level software libraries. About 25 years ago, ML was mainly relegated to low-level, yet efficient languages, such as Fortran, C, and C++. Unfortunately, at that time, ML was not still perceived as a key enabling tool for physics or other disciplines. Hence, there was no urgency to make ML widely available. As of today, there exist many efficient high-level libraries and frameworks, such as Scikit-learn [179], PyTorch [180], JAX [181], and Tensorflow [182], which allow to easily build up neural networks and run ML methods as they offer a Python API. The Python language is much easier than C,C++. Hence, it dramatically lowers the entrance barrier for any user who wants to enter the ML field. Also, these libraries solve one of the key historical nuisances of neural network training, namely manually computing in closed form analytical derivatives. These recent frameworks have the capability of computing derivatives symbolically and exactly, hence greatly improving the coding speed without compromising accuracy or execution efficiency. Having automatic derivation is of paramount importance also for simulation per se; the user can now devise a new potential function or a collective variable (either based on a neural network or not) and get forces in one line of code. That is in one function call, without explicitly coding the derivative, the derivative is symbolically obtained and

computed. This remarkable flexibility prompted the development of brand new molecular dynamics engines [181, 183] or the use of the mentioned libraries for supporting enhanced sampling [184, 185].

The third enabling factor is high-performance computing. Even without enhanced sampling, we are now able to see, in plain MD, events of biological interest such as binding [171] of a ligand into a receptor. The advent of GPUs has tremendously changed our approach to simulations; trivial parallelism is now a reality and is rather important both for simulation per se and particularly for simulation coupled with machine learning as they both benefit from this enhanced computational capability.

A.6 Challenges and conclusions

In this Appendix, we briefly reported some of the most relevant applications of machine learning to atomistic simulations. Machine learning is becoming a systematic and well-established enabling tool for computational physicists. Particularly, machine learning is rather relevant in automatically making sense of the huge amount of data (e.g. trajectories) and in general in accelerating computations. Machine learning for physics, for some applications, could be considered a not orthodox form of high-performance computing. This is particularly true for the potentials case where instead of optimizing the physics-based computational code, one substitutes the routines for physical potentials computing with much faster machine learning ones.

We underline one key transversal challenge for machine learning, namely generality and hence a tautological risk for various methods. While so-called *foundation* models have emerged in large language models (e.g. ChatGPT) [186], it is not so clear if the same path could be followed for atomistic simulations. A *foundation* model is a neural model that is pre-trained on a huge dataset and that with a minor computational effort can be customized for a specific need. A *foundation* model for instance for potential functions should be able to deal with any chemical element, possibly jointly. In principle, there is no reason why this should not happen, yet, at least, this would require a significant computational effort. The other possible limitation of ab initio machine learning potentials, for instance, is the dynamical suitability of these models; one has to ensure a stable force prediction over time. Indeed, general (several elements) machine learning potential functions are not employed in molecular dynamics systematically yet. Despite these difficulties, it is encouraging that some promising attempts in this direction exist [138–140].

Along these lines, we have analogous considerations for Boltzmann Generators, where now the generality issue stems from the fact that the learned models may not be able to properly interpolate in unseen phase space regions or unable to deal with explicit solvent. Same reasoning applies for targeted free energy perturbation where the mapping problem could be as difficult as the bare, conventional, free energy estimation itself.

In summary, machine learning is a powerful tool for atomistic simulations; it is characterized by remarkable flexibility and representation ability despite some possible drawbacks, mainly related to generality. We expect that in future, ML will become more

and more a routine tool, and also we expect that the related programming paradigms will become the standard for scientific computing in general.

Author Contributions GC and SM contributed to the conception and design of the main text. SDC is responsible for the conception and design of the Appendix. The first draft of the manuscript was written by SM and SDC for the main text and Appendix, respectively, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Ferrara within the CRUI-CARE Agreement.

Declarations

Conflict of interest Sergio Decherchi is partner of BiKi Technologies s.r.l. a company which sells a computational chemistry software.

References

1. D. Landau, K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2021)
2. H. Hellmann, Einführung in die quantenchemie, 1937. Leipzig, Franz Deuticke **285** (1937)
3. R.P. Feynman, Forces in molecules. Phys. Rev. **56**(4), 340 (1939)
4. R.M. Dreizler, E.K. Gross, *Density Functional Theory: an Approach to the Quantum Many-body Problem* (Springer, Heidelberg, 2012)
5. F. Ercolessi, J.B. Adams, Interatomic potentials from first-principles calculations: the force-matching method. Europhys. Lett. **26**(8), 583 (1994)
6. B.H. Brandsen, C.J. Joachain, *Physics of Atoms and Molecules* (Longman Scientific & Technical, Harlow, 1983)
7. A. Einstein, Theorie der opaleszenz von homogenen flüssigkeiten und flüssigkeitsgemischen in der nähe des kritischen zustandes. Ann. Phys. **338**(16), 1275–1298 (1910)
8. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
9. W. Wood, F. Parker, Monte Carlo equation of state of molecules interacting with the Lennard–Jones potential. I. A supercritical isotherm at about twice the critical temperature. J. Chem. Phys. **27**(3), 720–733 (1957)
10. B.J. Alder, T.E. Wainwright, Phase transition for a hard sphere system. J. Chem. Phys. **27**(5), 1208 (1957)
11. B.J. Alder, T.E. Wainwright, Studies in molecular dynamics. I. General method. J. Chem. Phys. **31**(2), 459–466 (1959)
12. M.E. Tuckerman, B.J. Berne, G.J. Martyna, Reversible multiple time scale molecular dynamics. J. Chem. Phys. **97**(3), 1990–2001 (1992)
13. H.F. Trotter, On the product of semi-groups of operators. Proc. Am. Math. Soc. **10**(4), 545–551 (1959)
14. H. Yoshida, Construction of higher order symplectic integrators. Phys. Lett. A **150**(5–7), 262–268 (1990)

15. M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics. J. Math. Phys. **32**(2), 400–407 (1991)
16. D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Elsevier, San Diego, 2023)
17. J.M. Martínez, L. Martínez, Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking. J. Comput. Chem. **24**(7), 819–825 (2003)
18. B. Quentrec, C. Brot, New method for searching for neighbors in molecular dynamics computations. J. Comput. Phys. **13**(3), 430–432 (1973)
19. M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, New York, 2017)
20. S. Meloni, M. Rosati, L. Colombo, Efficient particle labeling in atomistic simulations. J. Chem. Phys. **126**(12) (2007)
21. M.S. Daw, M.I. Baskes, Embedded-atom method: derivation and application to impurities, surfaces, and other defects in metals. Phys. Rev. B **29**(12), 6443 (1984)
22. J.-P. Ryckaert, A. Bellemans, Molecular dynamics of liquid alkanes. Faraday Discuss. Chem. Soc. **66**, 95–106 (1978)
23. J.E. Lennard-Jones, Cohesion. Proc. Phys. Soc. **43**(5), 461 (1931)
24. P.P. Ewald, Die berechnung optischer und elektrostatischer gitterpotentiale. Ann. Phys. **369**(3), 253–287 (1921)
25. R.W. Hockney, J.W. Eastwood, *Computer Simulation Using Particles* (CRC Press, New York, 1988)
26. F.H. Stillinger, T.A. Weber, Computer simulation of local order in condensed phases of silicon. Phys. Rev. B **31**(8), 5262 (1985)
27. J. Tersoff, New empirical approach for the structure and energy of covalent systems. Phys. Rev. B **37**(12), 6991 (1988)
28. J.-P. Ryckaert, G. Ciccotti, H.J. Berendsen, Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. J. Comput. Phys. **23**(3), 327–341 (1977)
29. H.C. Andersen, Rattle: a "velocity" version of the shake algorithm for molecular dynamics calculations. J. Comput. Phys. **52**(1), 24–34 (1983)
30. Y. Weinbach, R. Elber, Revisiting and parallelizing shake. J. Comput. Phys. **209**(1), 193–206 (2005)
31. R. Elber, A.P. Ruymgaart, B. Hess, Shake parallelization. Eur. Phys. J. Spec. Top. **200**(1), 211–223 (2011)
32. A.P. Ruymgaart, R. Elber, Revisiting molecular dynamics on a CPU/GPU system: Water kernel and shake parallelization. J. Chem. Theory Comput. **8**(11), 4624–4636 (2012)
33. M.E. Tuckerman, Y. Liu, G. Ciccotti, G.J. Martyna, Non-hamiltonian molecular dynamics: Generalizing hamiltonian phase space principles to non-hamiltonian systems. J. Chem. Phys. **115**(4), 1678–1702 (2001)
34. M.E. Tuckerman, C.J. Mundy, G.J. Martyna, On the classical statistical mechanics of non-hamiltonian systems. Europhys. Lett. **45**(2), 149 (1999)
35. G.J. Martyna, M.L. Klein, M. Tuckerman, Nosé–Hoover chains: the canonical ensemble via continuous dynamics. J. Chem. Phys. **97**(4), 2635–2643 (1992)
36. G.J. Martyna, D.J. Tobias, M.L. Klein, Constant pressure molecular dynamics algorithms. J. Chem. Phys. **101**(5), 4177–4189 (1994)
37. B. Leimkuhler, C. Matthews, Molecular dynamics. Interdiscip. Appl. Math. **39**(1) (2015)
38. A. Szabo, N.S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Courier Corporation, New York, 2012)
39. W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects. Phys. Rev. **140**(4A), 1133 (1965)
40. C. Cohen-Tannoudji, B. Diu, F. Laloe, Quantum mechanics, volume 1. Quantum Mech. **1**, 898 (1986)
41. J. Verbeke, R. Cools, The Newton–Raphson method. Int. J. Math. Educ. Sci. Technol. **26**(2), 177–193 (1995)
42. M.C. Payne, M.P. Teter, D.C. Allan, T. Arias, A.J. Joannopoulos, Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. Rev. Mod. Phys. **64**(4), 1045 (1992)
43. P. Pulay, Convergence acceleration of iterative sequences. The case of SCF iteration. Chem. Phys. Lett. **73**(2), 393–398 (1980)
44. P. Pulay, Improved SCF convergence acceleration. J. Comput. Chem. **3**(4), 556–560 (1982)

45. F. Filippone, S. Meloni, M. Parrinello, A novel implicit Newton–Raphson geometry optimization method for density functional theory calculations. J. Chem. Phys. **115**(2), 636–642 (2001)
46. R. Car, M. Parrinello, Unified approach for molecular dynamics and density-functional theory. Phys. Rev. Lett. **55**(22), 2471 (1985)
47. F.A. Bornemann, C. Schütte, Adaptive accuracy control for Car–Parrinello simulations. Numer. Math. **83**(2), 179–186 (1999)
48. G. Pastore, E. Smargiassi, F. Buda, Theory of ab initio molecular-dynamics calculations. Phys. Rev. A **44**(10), 6334 (1991)
49. P.E. Blöchl, M. Parrinello, Adiabaticity in first-principles molecular dynamics. Phys. Rev. B **45**(16), 9413 (1992)
50. A. Giacomello, S. Meloni, M. Müller, C.M. Casciola, Mechanism of the Cassie–Wenzel transition via the atomistic and continuum string methods. J. Chem. Phys. **142**(10) (2015)
51. L. Maragliano, A. Fischer, E. Vanden-Eijnden, G. Ciccotti, String method in collective variables: minimum free energy paths and isocommittor surfaces. J. Chem. Phys. **125**(2) (2006)
52. S. Yao, R. Van, X. Pan, J.H. Park, Y. Mao, J. Pu, Y. Mei, Y. Shao, Machine learning based implicit solvent model for aqueous-solution alanine dipeptide molecular dynamics simulations. RSC Adv. **13**(7), 4565–4577 (2023)
53. G.M. Torrie, J.P. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling. J. Comput. Phys. **23**(2), 187–199 (1977)
54. S. Kumar, J.M. Rosenberg, D. Bouzida, R.H. Swendsen, P.A. Kollman, The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. J. Comput. Chem. **13**(8), 1011–1021 (1992)
55. M. Souaille, B. Roux, Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations. Comput. Phys. Commun. **135**(1), 40–57 (2001)
56. J. Kästner, W. Thiel, Bridging the gap between thermodynamic integration and umbrella sampling provides a novel analysis method: "umbrella integration". J. Chem. Phys. **123** (2005)
57. E.A. Carter, G. Ciccotti, J.T. Hynes, R. Kapral, Constrained reaction coordinate dynamics for the simulation of rare events. Chem. Phys. Lett. **156**(5), 472–477 (1989)
58. M. Sprik, G. Ciccotti, Free energy from constrained molecular dynamics. J. Chem. Phys. **109**(18), 7737–7744 (1998)
59. G. Ciccotti, M. Ferrario, Holonomic constraints: A case for statistical mechanics of non-hamiltonian systems. Computation **6**(1), 11 (2018)
60. H. Goldstein, *Classical Mechanics* (Pearson Education India, Noida, 2011)
61. L. Maragliano, E. Vanden-Eijnden, A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations. Chem. Phys. Lett. **426**(1–3), 168–175 (2006)
62. J.G. Kirkwood, Statistical mechanics of fluid mixtures. J. Chem. Phys. **3**(5), 300–313 (1935)
63. A.F. Voter, Hyperdynamics: accelerated molecular dynamics of infrequent events. Phys. Rev. Lett. **78**(20), 3908 (1997)
64. D.J. Earl, M.W. Deem, Parallel tempering: theory, applications, and new perspectives. Phys. Chem. Chem. Phys. **7**(23), 3910–3916 (2005)
65. A. Mitsutake, Y. Sugita, Y. Okamoto, Generalized-ensemble algorithms for molecular simulations of biopolymers. Peptide Sci. Orig. Res. Biomol. **60**(2), 96–123 (2001)
66. T. Huber, A.E. Torda, W.F. Van Gunsteren, Local elevation: a method for improving the searching properties of molecular dynamics simulation. J. Comput.-Aided Mol. Des. **8**, 695–708 (1994)
67. H. Grubmüller, Predicting slow structural transitions in macromolecular systems: conformational flooding. Phys. Rev. E **52**(3), 2893 (1995)
68. L. Rosso, P. Mináry, Z. Zhu, M.E. Tuckerman, On the use of the adiabatic molecular dynamics technique in the calculation of free energy profiles. J. Chem. Phys. **116**(11), 4389–4402 (2002)
69. A. Laio, M. Parrinello, Escaping free-energy minima. Proc. Natl. Acad. Sci. **99**(20), 12562–12566 (2002)
70. J. VandeVondele, U. Rothlisberger, Canonical adiabatic free energy sampling (cafes): a novel method for the exploration of free energy surfaces. J. Phys. Chem. B **106**(1), 203–208 (2002)
71. G. Bussi, F.L. Gervasio, A. Laio, M. Parrinello, Free-energy landscape for β hairpin folding from combined parallel tempering and metadynamics. J. Am. Chem. Soc. **128**(41), 13435–13441 (2006)
72. H. Eyring, The activated complex in chemical reactions. J. Chem. Phys. **3**(2), 107–115 (1935)

73. J. Keck, Statistical investigation of dissociation cross-sections for diatoms. Discuss. Faraday Soc. **33**, 173–182 (1962)
74. C.H. Bennett, *Molecular Dynamics and Transition State Theory: The Simulation of Infrequent Events* (ACS Publications, Washington, 1977)
75. D. Chandler, Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. J. Chem. Phys. **68**(6), 2959–2970 (1978)
76. E. Vanden-Eijnden, F.A. Tal, Transition state theory: variational formulation, dynamical corrections, and error estimates. J. Chem. Phys. **123**(18) (2005)
77. G. Menzl, A. Singraber, C. Dellago, S-shooting: a Bennett-chandler-like method for the computation of rate constants from committor trajectories. Faraday Discuss. **195**, 345–364 (2016)
78. J. Daru, A. Stirling, Divided saddle theory: a new idea for rate constant calculation. J. Chem. Theory Comput. **10**(3), 1121–1127 (2014)
79. P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Transition path sampling: Throwing ropes over rough mountain passes, in the dark. Annu. Rev. Phys. Chem. **53**(1), 291–318 (2002)
80. R.J. Allen, P.B. Warren, P.R. Ten Wolde, Sampling rare switching events in biochemical networks. Phys. Rev. Lett. **94**(1), 018104 (2005)
81. R.J. Allen, C. Valeriani, P.R. Ten Wolde, Forward flux sampling for rare event simulations. J. Phys. Condens. Matter **21**(46), 463102 (2009)
82. T.S. Van Erp, D. Moroni, P.G. Bolhuis, A novel path sampling method for the calculation of rate constants. J. Chem. Phys. **118**(17), 7762–7774 (2003)
83. D.T. Zhang, L. Baldauf, S. Roet, A. Lervik, T.S. Erp, Highly parallelizable path sampling with minimal rejections using asynchronous replica exchange and infinite swaps. Proc. Natl. Acad. Sci. **121**(7), 2318731121 (2024)
84. E. Vanden-Eijnden, M. Venturoli, Exact rate calculations by trajectory parallelization and tilting. J. Chem. Phys. **131**(4) (2009)
85. R. Elber, *Molecular Dynamics With Milestoning: Byways to Compute Kinetic* (Elsevier, Cambridge, 2023), pp.401–453
86. G. Henkelman, H. Jónsson, A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. J. Chem. Phys. **111**(15), 7010–7022 (1999)
87. M. Plasencia Gutierrez, C. Argaez, H. Jonsson, Improved minimum mode following method for finding first order saddle points. J. Chem. Theory Comput. **13**(1), 125–134 (2017)
88. W. Ren, E. Vanden-Eijnden, A climbing string method for saddle point search. J. Chem. Phys. **138**(13) (2013)
89. B.E. Husic, V.S. Pande, Markov state models: from an art to a science. J. Am. Chem. Soc. **140**(7), 2386–2396 (2018)
90. C. Schütte, M. Sarich, *Metastability and Markov State Models in Molecular Dynamics*, vol. 24 (American Mathematical Society, Providence, 2013)
91. L. Onsager, Initial recombination of ions. Phys. Rev. **54**(8), 554 (1938)
92. P.L. Geissler, C. Dellago, D. Chandler, Kinetic pathways of ion pair dissociation in water. J. Phys. Chem. B **103**(18), 3706–3710 (1999)
93. E. Vanden-Eijnden, Transition path theory, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology*, vol. 1 (Springer, Berlin, 2006), pp.453–493
94. W. Ren, E. Vanden-Eijnden et al., Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. J. Chem. Phys. **126**(16) (2007)
95. E. Vanden-Eijnden, Transition-path theory and path-finding algorithms for the study of rare events. Annu. Rev. Phys. Chem. **61**, 391–420 (2010)
96. G. Henkelman, H. Jónsson, Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. J. Chem. Phys. **113**(22), 9978–9985 (2000)
97. P. Terrier, M.-C. Marinica, M. Athènes, Using Bayes formula to estimate rates of rare events in transition path sampling simulations. J. Chem. Phys. **143**(13) (2015)
98. D. Branduardi, F.L. Gervasio, M. Parrinello, From a to b in free energy space. J. Chem. Phys. **126**(5) (2007)
99. B. Peters, B.L. Trout, Obtaining reaction coordinates by likelihood maximization. J. Chem. Phys. **125**(5) (2006)
100. W. Lechner, J. Rogal, J. Juraszek, B. Ensing, P.G. Bolhuis, Nonlinear reaction coordinate analysis in the reweighted path ensemble. J. Chem. Phys. **133**(17) (2010)

101. J. Martin-Roca, R. Martinez, L.C. Alexander, A.L. Diez, D.G. Aarts, F. Alarcon, J. Ramírez, C. Valeriani, Characterization of MIPS in a suspension of repulsive active Brownian particles through dynamical features. J. Chem. Phys. **154**(16) (2021)
102. M.R. Shaebani, A. Wysocki, R.G. Winkler, G. Gompper, H. Rieger, Computational models for active matter. Nat. Rev. Phys. **2**(4), 181–199 (2020)
103. S. Ramaswamy, The mechanics and statistics of active matter. Annu. Rev. Condens. Matter Phys. **1**(1), 323–345 (2010)
104. R. Kubo, Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems. J. Phys. Soc. Jpn. **12**(6), 570–586 (1957)
105. Kubo, R.: The fluctuation-dissipation theorem. Rep. Prog. Phys. 29, 255 (1966). https://iopscience.iop.org/article/10.1088/0034-4885/29/1/306
106. E. Kestemont, J. Van Craen, On the computation of correlation functions in molecular dynamics experiments. J. Comput. Phys. **22**(4), 451–458 (1976)
107. S. Orlandini, S. Meloni, G. Ciccotti, Hydrodynamics from dynamical non-equilibrium md, in *AIP Conference Proceedings*, vol. 1332 (American Institute of Physics, 2011), pp. 77–95
108. M.L. Mugnai, S. Caprara, G. Ciccotti, C. Pierleoni, M. Mareschal, Transient hydrodynamical behavior by dynamical nonequilibrium molecular dynamics: the formation of convective cells. J. Chem. Phys. **131**(6), 064106 (2009)
109. M. Pourali, S. Meloni, F. Magaletti, A. Maghari, C.M. Casciola, G. Ciccotti, Relaxation of a steep density gradient in a simple fluid: comparison between atomistic and continuum modeling. J. Chem. Phys. **141**(15) (2014)
110. M. Lauricella, G. Ciccotti, N.J. English, B. Peters, S. Meloni, Mechanisms and nucleation rate of methane hydrate by dynamical nonequilibrium molecular dynamics. J. Phys. Chem. C **121**(43), 24223–24234 (2017)
111. S. Orlandini, S. Meloni, G. Ciccotti, Hydrodynamics from statistical mechanics: combined dynamical-nemd and conditional sampling to relax an interface between two immiscible liquids. Phys. Chem. Chem. Phys. **13**(29), 13177–13181 (2011)
112. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, New York, 2009)
113. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**(7553), 436–444 (2015)
114. H. Wu, F. Noé, Variational approach for learning Markov processes from time series data. J. Nonlinear Sci. **30**(1), 23–66 (2019)
115. J.S. Smith, O. Isayev, A.E. Roitberg, Ani-1: an extensible neural network potential with DFT accuracy at force field computational cost. Chem. Sci. **8**, 3192–3203 (2017)
116. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
117. V.N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, New York, 2000)
118. A. Krogh, J.A. Hertz, A simple weight decay can improve generalization, in *Proceedings of the 4th International Conference on Neural Information Processing Systems. NIPS'91* (Morgan Kaufmann Publishers Inc., San Francisco, 1991), pp. 950–957
119. A.N. Tikhonov, V.Y. Arsenin, *Solutions of Ill-posed Problems* (V. H. Winston & Sons, Washington, D.C.; Wiley, New York, 1977)
120. E.D. Vito, L. Rosasco, A. Caponnetto, U.D. Giovannini, F. Odone, Learning from examples as an inverse problem. J. Mach. Learn. Res. **6**, 883–904 (2005)
121. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
122. E.G. Tabak, E. Vanden-Eijnden, Density estimation by dual ascent of the log-likelihood. Commun. Math. Sci. **8**, 217–233 (2010)
123. F. Noé, S. Olsson, J. Köhler, H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. Science **365**(6457), 1147 (2019)
124. M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
125. R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems. NIPS'18* (Curran Associates Inc., Red Hook, 2018), pp. 6572–6583

126. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778
127. D.C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization. Math. Program. **45**, 503–528 (1989)
128. P.-L. Kang, C. Shang, Z.-P. Liu, Large-scale atomic simulation via machine learning potentials constructed by global potential energy surface exploration. Acc. Chem. Res. **53**(10), 2119–2129 (2020)
129. J. Behler, G. Csányi, Machine learning potentials for extended systems: a perspective. Eur. Phys. J. B **94**(7) (2021)
130. D.S. Sholl, J.A. Steckel, *Density Functional Theory: A Practical Introduction* (Wiley, Hoboken, 2009)
131. T. Morawietz, N. Artrith, Machine learning-accelerated quantum mechanics-based atomistic simulations for industrial applications. J. Comput.-Aided Mol. Des. **35**(4), 557–586 (2020)
132. F. Musil, A. Grisafi, A.P. Bartók, C. Ortner, G. Csányi, M. Ceriotti, Physics-inspired structural representations for molecules and materials. Chem. Rev. **121**(16), 9759–9815 (2021)
133. J.S. Smith, O. Isayev, A.E. Roitberg, Ani-1: an extensible neural network potential with DFT accuracy at force field computational cost. Chem. Sci. **8**(4), 3192–3203 (2017)
134. J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys. Rev. Lett. **98**, 146401 (2007)
135. J. Wang, P. Cieplak, P.A. Kollman, How well does a restrained electrostatic potential (resp) model perform in calculating conformational energies of organic and biological molecules? J. Comput. Chem. **21**(12), 1049–1074 (2000)
136. D.M. Anstine, O. Isayev, Machine learning interatomic potentials and long-range physics. J. Phys. Chem. A **127**(11), 2417–2431 (2023)
137. M. Eckhoff, M. Reiher, Lifelong machine learning potentials. J. Chem. Theory Comput. **19**(12), 3509–3525 (2023)
138. A.E.A. Allen, N. Lubbers, S. Matin, J. Smith, R. Messerly, S. Tretiak, K. Barros, Learning together: towards foundation models for machine learning interatomic potentials with meta-learning. npj Comput. Mater. **10**(1), 154 (2024)
139. E. Gelžinytė, M. Öeren, M.D. Segall, G. Csányi, Transferable machine learning interatomic potential for bond dissociation energy prediction of drug-like molecules. J. Chem. Theory Comput. **20**(1), 164–177 (2024)
140. S. Martire, S. Decherchi, A. Cavalli, Obiwan: an element-wise scalable feed-forward neural network potential. J. Chem. Theory Comput. **20**(14), 6287–6302 (2024)
141. L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using Real NVP. arXiv (2016)
142. D. Frenkel, A.J.C. Ladd, New Monte Carlo method to compute the free energy of arbitrary solids. Application to the FCC and HCP phases of hard spheres. J. Chem. Phys. **81**(7), 3188–3193 (1984)
143. D. Gobbo, P. Ballone, S. Decherchi, A. Cavalli, Solubility advantage of amorphous ketoprofen. Thermodynamic and kinetic aspects by molecular dynamics and free energy approaches. J. Chem. Theory Comput. **16**(7), 4126–4140 (2020)
144. C. Chipot, A. Pohorille, *Free Energy Calculations: Theory and Applications in Chemistry and Biology* (Springer, Berlin, 2007)
145. S. Decherchi, A. Cavalli, Thermodynamics and kinetics of drug-target binding by molecular simulation. Chem. Rev. **120**(23), 12788–12833 (2020)
146. C. Jarzynski, Targeted free energy perturbation. Phys. Rev. E **65**, 046122 (2002)
147. A. Mackiewicz, W. Ratajczak, Principal components analysis (PCA). Comput. Geosci. **19**(3), 303–342 (1993)
148. L. Molgedey, H.G. Schuster, Separation of a mixture of independent signals using time delayed correlations. Phys. Rev. Lett. **72**, 3634–3637 (1994)
149. G.R. Bowman, V.S. Pande, F. Noé, *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation* (Springer, Netherlands, 2014)
150. Z. Belkacemi, P. Gkeka, T. Lelièvre, G. Stoltz, Chasing collective variables using autoencoders and biased trajectories. J. Chem. Theory Comput. **18**(1), 59–78 (2021)
151. M. Bertazzo, D. Gobbo, S. Decherchi, A. Cavalli, Machine learning and enhanced sampling simulations for computing the potential of mean force and standard binding free energy. J. Chem. Theory Comput. **17**(8), 5287–5300 (2021)
152. M.J. Ferrarotti, W. Rocchia, S. Decherchi, Finding principal paths in data space. IEEE Trans. Neural Netw. Learn. Syst. **30**(8), 2449–2462 (2019)

153. S. Bhakat, Collective variable discovery in the age of machine learning: reality, hype and everything in between. RSC Adv. **12**(38), 25010–25024 (2022)
154. H. Sidky, W. Chen, A.L. Ferguson, Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. Mol. Phys. **118**(5) (2020)
155. M.M. Sultan, V.S. Pande, Automated design of collective variables using supervised machine learning. J. Chem. Phys. **149**(9) (2018)
156. S. Zheng, C. Ding, F. Nie, H. Huang, Harmonic mean linear discriminant analysis. IEEE Trans. Knowl. Data Eng. **31**(8), 1520–1531 (2019)
157. D. Mendels, G. Piccini, M. Parrinello, Collective variables from local fluctuations. J. Phys. Chem. Lett. **9**(11), 2776–2781 (2018)
158. M. Dorfer, R. Kelz, G. Widmer, Deep linear discriminant analysis, in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. ed. by Y. Bengio, Y. LeCun (2016)
159. L. Bonati, V. Rizzi, M. Parrinello, Data-driven collective variables for enhanced sampling. J. Phys. Chem. Lett. **11**(8), 2998–3004 (2020)
160. L. Maragliano, E. Vanden-Eijnden, Single-sweep methods for free energy calculations. J. Chem. Phys. **128**(18) (2008)
161. T. Stecher, N. Bernstein, G. Csáanyi, Free energy surface reconstruction from umbrella samples using gaussian process regression. J. Chem. Theory Comput. **10**(9), 4079–4097 (2014)
162. C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Cambridge, 2005)
163. R.W. Zwanzig, High-temperature equation of state by a perturbation method. I. Nonpolar gases. J. Chem. Phys. **22**(8), 1420–1426 (1954)
164. P. Wirnsberger, A.J. Ballard, G. Papamakarios, S. Abercrombie, S. Racaniére, A. Pritzel, D. Jimenez Rezende, C. Blundell, Targeted free energy estimation via learned mappings. J. Chem. Phys. **153**(14) (2020)
165. A. Rizzi, P. Carloni, M. Parrinello, Targeted free energy perturbation revisited: accurate free energies from mapped reference potentials. J. Phys. Chem. Lett. **12**(39), 9449–9454 (2021)
166. A. Rizzi, P. Carloni, M. Parrinello, Free energies at QM accuracy from force fields via multimap targeted estimation. Proc. Natl. Acad. Sci. **120**(46), 2304308120 (2023)
167. S. Lloyd, Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
168. L. Kaufman, *Partitioning Around Medoids (Program PAM)* (Wiley, New York, 1990)
169. M. Moradi Fard, T. Thonet, E. Gaussier, Deep k-means: jointly clustering with k-means and learning representations. Pattern Recognit. Lett. **138**, 185–192 (2020)
170. M. Girolami, Mercer kernel-based clustering in feature space. IEEE Trans. Neural Netw. **13**(3), 780–784 (2002)
171. S. Decherchi, A. Berteotti, G. Bottegoni, W. Rocchia, A. Cavalli, The ligand binding mechanism to purine nucleoside phosphorylase elucidated via molecular dynamics and machine learning. Nat. Commun. **6**(1) (2015)
172. P. Deuflhard, W. Huisinga, A. Fischer, C. Schütte, Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. Linear Algebra Appl. **315**, 39–59 (2000)
173. E. Suarez, R.P. Wiewiora, C. Wehmeyer, F. Noé, J.D. Chodera, D.M. Zuckerman, What Markov state models can and cannot do: Correlation versus path-based observables in protein-folding models. J. Chem. Theory Comput. **17**(5), 3119–3133 (2021)
174. A. Mardt, L. Pasquali, H. Wu, F. Noé, Vampnets for deep learning of molecular kinetics. Nat. Commun. **9**(1) (2018)
175. J.D. Leeuw, Modern multidimensional scaling: theory and applications (second edition). J. Stat. Softw. **14**(Book Review 4) (2005)
176. M. Ceriotti, G.A. Tribello, M. Parrinello, Simplifying the representation of complex free-energy landscapes using sketch-map. Proc. Natl. Acad. Sci. **108**(32), 13023–13028 (2011)
177. M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. Silva Santos, P.E. Bourne, J. Bouwman, A.J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C.T. Evelo, R. Finkers, A. Gonzalez-Beltran, A.J.G. Gray, P. Groth, C. GobleJ.S., Grethe, J. Heringa, P.A.C. Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S.J. Lusher, M.E. Martone, A. Mons, A.L. Packer, B. Persson, P. Rocca-Serra, Roos, M., Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M.A., Thompson, M., Lei, J., Mulligen, E.,

Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., Mons, B., The fair guiding principles for scientific data management and stewardship. Sci. Data **3**(1) (2016)
178. C.W. Andersen, R. Armiento, E. Blokhin, G.J. Conduit, S. Dwaraknath, M.L. Evans, Á. Fekete, A. Gopakumar, S. Gražulis, A. Merkys, F. Mohamed, C. Oses, G. Pizzi, G.-M. Rignanese, M. Scheidgen, L. Talirz, C. Toher, D. Winston, R. Aversa, K. Choudhary, P. Colinet, S. Curtarolo, D. Di Stefano, C. Draxl, S. Er, M. Esters, M. Fornari, M. Giantomassi, M. Govoni, G. Hautier, V. Hegde, M.K. Horton, P. Huck, G. Huhs, J. Hummelshøj, A. Kariryaa, B. Kozinsky, S. Kumbhar, M. Liu, N. Marzari, A.J. Morris, A.A. Mostofi, K.A. Persson, G. Petretto, T. Purcell, F. Ricci, F. Rose, M. Scheffler, D. Speckhard, M. Uhrin, A. Vaitkus, P. Villars, D. Waroquiers, C. Wolverton, M. Wu, X. Yang, Optimade, an API for exchanging materials data. Sci. Data **8**(1), 217 (2021)
179. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
180. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (Curran Associates Inc., Red Hook, 2019)
181. S.S. Schoenholz, E.D. Cubuk, M.D. Jax, A framework for differentiable physics. J. Stat. Mech. Theory Exp. **2021**(12), 124016 (2021)
182. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015)
183. S. Doerr, M. Majewski, A. Pérez, A. Krämer, C. Clementi, F. Noé, T. Giorgino, G. De Fabritiis, Torchmd: a deep learning framework for molecular simulations. J. Chem. Theory Comput. **17**(4), 2355–2363 (2021)
184. P.F. Zubieta Rico, L. Schneider, G.R. Pérez-Lemus, R. Alessandri, S. Dasetty, T.D. Nguyen, C.A. Menéndez, Y. Wu, Y. Jin, Y. Xu, S. Varner, J.A. Parker, A.L. Ferguson, J.K. Whitmer, J.J. Pablo, Pysages: flexible, advanced sampling methods accelerated with gpus. npj Comput. Mater. **10**(1) (2024)
185. L. Bonati, E. Trizio, A. Rizzi, M. Parrinello, A unified framework for machine learning collective variables for enhanced sampling simulations: mlcolvar. J. Chem. Phys. **159**(1) (2023)
186. P.P. Ray, Chatgpt: a comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet Things Cyber-Phys. Syst. **3**, 121–154 (2023)