FlashMD: long-stride, universal prediction of molecular dynamics

Filippo Bigi*
Institute of Materials
Ecole Polytechnique Fédérale de Lausanne
Lausanne 1015, Switzerland
filippo.bigi@epfl.ch

Sanggyu Chong*
Institute of Materials
Ecole Polytechnique Fédérale de Lausanne
Lausanne 1015, Switzerland
sanggyu.chong@epfl.ch

Agustinus Kristiadi
Department of Computer Science
Western University & Vector Institute
London, ON N6A 3K7, Canada
akristi@uwo.ca

Michele Ceriotti
Institute of Materials
Ecole Polytechnique Fédérale de Lausanne
Lausanne 1015, Switzerland
michele.ceriotti@epfl.ch

Abstract

Molecular dynamics (MD) provides insights into atomic-scale processes by integrating over time the equations that describe the motion of atoms under the action of interatomic forces. Machine learning models have substantially accelerated MD by providing inexpensive predictions of the forces, but they remain constrained to minuscule time integration steps, which are required by the fast time scale of atomic motion. In this work, we propose FlashMD, a method to predict the evolution of positions and momenta over strides that are between one and two orders of magnitude longer than typical MD time steps. We incorporate considerations on the mathematical and physical properties of Hamiltonian dynamics in the architecture, generalize the approach to allow the simulation of any thermodynamic ensemble, and carefully assess the possible failure modes of such a long-stride MD approach. We validate FlashMD's accuracy in reproducing equilibrium and time-dependent properties, using both system-specific and general-purpose models, extending the ability of MD simulation to reach the long time scales needed to model microscopic processes of high scientific and technological relevance.

1 Introduction

Simulations of atomic-scale systems are at the core of computational physics, chemistry, biology, and materials science [1]. Molecular dynamics (MD) in particular is a powerful tool for investigating the behavior of microscopic systems, from proteins [2–5] to chemical reactions [6–8] and materials [9–11]. MD elucidates atomic-scale structure and mechanisms by numerically solving the equations of atomic motion, driven by forces that can be estimated by quantum ground state electronic-structure calculations [12]. This has allowed simulating the time-evolution of various systems at the atomic scale, and probing thermodynamic and kinetic properties that are often difficult to measure experimentally.

Despite its utility, MD has long been hindered by the trade-off between computational cost and accuracy of employed methods. Machine learning interatomic potentials [13–19] (MLIPs) have remedied this in part by allowing the cheap approximation of otherwise expensive quantum mechanical

*These two authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

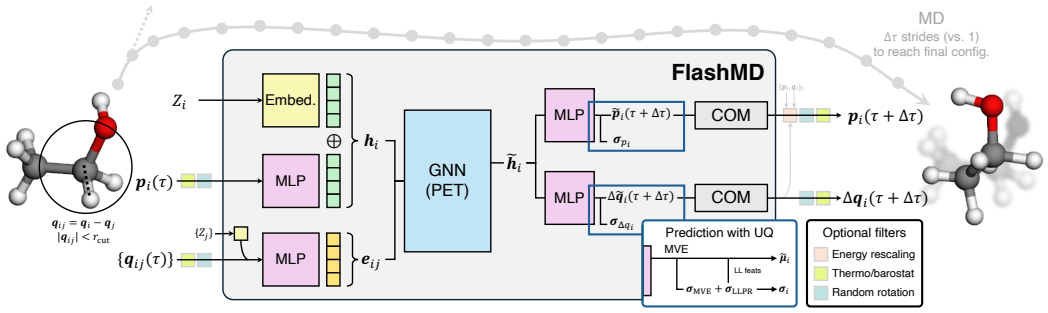

Figure 1: Schematic overview of FlashMD. Atoms of the system at time step τ are taken as inputs, with atomic numbers Z_i and momenta $\boldsymbol{p}_i(\tau)$ embedded into the node features $\boldsymbol{h}_i$, and relative coordinates $\boldsymbol{q}_{ij}(\tau)$ embedded into the edge features $\boldsymbol{e}_{ij}$ of a GNN for the system. The node outputs are used to predict the new configuration $\boldsymbol{p}_i(\tau + \Delta\tau)$ and $\Delta\boldsymbol{q}_i(\tau + \Delta\tau)$ in a multi-head manner. Center-of-mass constraints are also enforced. Uncertainty quantification can be enabled as shown in the navy inset. Optional filters for energy conservation enforcement, thermodynamic ensemble control, and random rotation are provided, as discussed below. Conventional MD would require $\Delta\tau$ explicit numerical integrations to reach the final configuration as opposed to 1 pass of FlashMD.

energies and forces. Nonetheless, MLIPs exhibit their own constraints of having to obey the physical symmetries and requiring expensive gradient computation to obtain the forces for MD propagation. Furthermore, stable and theoretically meaningful MD simulations require mathematical integration of the equations of motion with sufficiently small time steps (~ 1 fs), limiting the simulations to a regime far removed from experimentally relevant time scales.

Motivated by the latest developments in MLIPs involving symmetry breaking and direct force learning [20–22], as well as by generative approaches that construct representative atomic configurations without following the physically motivated equations of motion [23], this work will focus on the direct prediction of MD trajectories (see Fig. 1). This approach avoids both the explicit calculation of interatomic forces and the numerical integration of the equations of motion, allowing one to use much larger strides compared to traditional MD integrators – with a corresponding, dramatic extension of the time scales accessible via atomistic modeling.

Our novel contributions are summarized as follows: (i) We provide a thorough theoretical analysis of the problem of directly learning MD trajectories, discussing potential pitfalls. (ii) We introduce techniques for higher accuracy and larger time steps, e.g. by enforcing exact conservation of energy at inference time, proving its importance in stabilizing trajectories and reproducing physically correct behavior. (iii) We generalize the approach to MD in arbitrary, experimentally-relevant thermodynamic ensembles. (iv) We present universal models for direct, large-stride MD, capable of predicting trajectories across a wide range of chemical systems with diverse structure and composition.

2 Background and related work

We aim to predict the sequence of position $\boldsymbol{q}_i(\tau)$ and momentum $\boldsymbol{p}_i(\tau)$ for each atom i at the *discrete* time step τ of a MD trajectory integrated with a small time step Δt (so the actual simulation time at a time step τ is $t = \tau\Delta t$). Conventional MD evolves the dynamics using positions and momenta of all atoms at time step τ to predict the new positions and momenta at time $\tau + 1$. Our goal here is to be able to take longer strides $\Delta\tau$, skipping all intermediate steps, thereby achieving a corresponding reduction in computational cost and eventually allowing practitioners to access much longer time scales in MD.

2.1 Molecular dynamics

In its simplest form, MD is the numerical solution of Hamilton's equations

$$\frac{d\boldsymbol{q}_i}{dt} = \frac{\partial H}{\partial \boldsymbol{p}_i}, \quad \frac{d\boldsymbol{p}_i}{dt} = -\frac{\partial H}{\partial \boldsymbol{q}_i}, \tag{1}$$

for an atomistic system with N atoms at positions $\{\boldsymbol{q}_i\}_{i=1}^N$ and momenta $\{\boldsymbol{p}_i\}_{i=1}^N$. The Hamiltonian function H describing the dynamics, in the absence of external perturbations, takes the form $H(\{\boldsymbol{p}_i, \boldsymbol{q}_i\}_{i=1}^N) = \sum_{i=1}^N \boldsymbol{p}_i^2/2m_i + V(\{\boldsymbol{q}_i\}_{i=1}^N)$, where m_i are the atomic masses and $V(\{\boldsymbol{q}_i\}_{i=1}^N)$ is the potential energy of the system. In practice, Eq. 1 is discretized using a time step Δt. Among the many algorithms that could be used for numerical integration, the velocity Verlet (VV) algorithm [24] has become the standard due to its simplicity and the fact it preserves exactly some of the key properties of the underlying continuous Hamiltonian dynamics (Sec. 2.2). A single VV step reads

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i - \frac{1}{2}\frac{\partial V}{\partial \boldsymbol{q}_i}\Delta t, \quad \boldsymbol{q}_i \leftarrow \boldsymbol{q}_i + \frac{\boldsymbol{p}_i}{m_i}\Delta t, \quad \boldsymbol{p}_i \leftarrow \boldsymbol{p}_i - \frac{1}{2}\frac{\partial V}{\partial \boldsymbol{q}_i}\Delta t. \tag{2}$$

If integrated with a sufficiently small Δt, the VV algorithm approximately conserves the energy of the system, making it suitable to sample the NVE thermodynamic ensemble (where the number of particles N, volume V and total energy E are fixed).

The NVE ensemble rarely corresponds to realistic experimental conditions. For this reason, variants of MD have been developed to target other types of ensembles [25, 26], accelerate their statistical sampling [27], account for nuclear quantum effects [28], etc. Practically, NVE MD can be modified via the inclusion of thermostats (e.g., [29, 30]), which allows sampling the constant-temperature (NVT) ensemble, as well as the addition of barostats to sample constant-pressure (NpT) ensemble. Specialized Monte-Carlo moves are also used to access ensembles with varying number of particles and constant chemical potential (μVT). These different variants of MD are discussed further in Appendix D, and are all built around the NVE integrator, whose accuracy is therefore of central importance to achieve sampling of configurations with the correct probabilities.

Molecular dynamics with machine learning interatomic potentials The integrator in (2) is simple and computationally inexpensive, even though it requires a small time step. The bottleneck is typically the evaluation of the force $\boldsymbol{F}_i = -\partial V/\partial \boldsymbol{q}_i$ at every step, which is traditionally done using affordable but inaccurate empirical potentials, or accurate but very demanding quantum mechanical calculations. Over the last two decades, most efforts of applying machine learning (ML) to accelerate MD have revolved around MLIPs that approximate the potential energy surface $V(\{\boldsymbol{q}_i\}_{i=1}^N)$ from quantum mechanical calculations at a much reduced cost. Though MLIPs were traditionally focused on describing a specific chemical system, the last few years have seen the development of "universal" MLIPs [31–40], which aim to provide good accuracy across the whole periodic table, in principle allowing users to simply deploy the model for the desired system without further, dedicated training.

2.2 Physical and mathematical considerations

Symmetries of the potential energy function The potential $V(\{\boldsymbol{q}_i\}_{i=1}^N)$ obeys two fundamental physical symmetries: (i) S_N-invariance: $V(\{\boldsymbol{q}_i\}_{i=1}^N)$ is invariant with respect to permutations of atom indices; (ii) E(3)-invariance: $V(\{\boldsymbol{q}_i\}_{i=1}^N)$ is invariant with respect to translations, rotations and inversions of the atomic structure. Although traditional MLIPs incorporate all these symmetries through symmetry-constrained architectures, some recent models do not directly enforce rotational (and inversion) symmetry and use instead data augmentation strategies to encourage the model to approximately capture it at training-time [18, 22]. This allows the models to avoid expensive equivariant operations and make inference more computationally efficient, without significantly affecting the physical observables in MD [20]. It should also be noted that preserving the two remaining symmetries (those with respect to translations and permutations), which are much harder to augment, naturally leads to the choice of graph neural networks (GNNs) for learning interatomic potentials, explaining their widespread adoption in last-generation MLIPs.

Conservative and non-conservative forces Using forces that are the derivatives of V in the propagation of Hamiltonian dynamics conserves the total energy H. Even though this is a requirement to sample the NVE ensemble, some recent models have implemented the direct prediction of $\boldsymbol{F}_i$ as an arbitrary vector field, leading to non-conservative dynamics [22, 32, 41]. These "direct-force" models do not even apply data augmentation to encourage energy conservation, as doing so involves computing the Jacobian of the forces, which is computationally impractical. Nevertheless, high model accuracy and the use of hybrid integration strategies [21, 42] allow non-conservative forces to be used in MD without generating noticeable unphysical artifacts, and with the efficiency benefits given by skipping the differentiation step.

Symmetries in molecular dynamics The possibility of performing stable MD with direct force predictions, and therefore without an underlying potential energy surface, suggests that an explicit potential energy function might not be necessary to predict the system evolution over time. It is therefore natural to consider the underlying symmetries in MD, and understand whether they can be incorporated efficiently into a potential-free model, or encouraged at training-time.

The VV algorithm, as presented in Eq. 2, displays two fundamental symmetries, both of which are properties of the underlying continuous solution of Hamilton's equations: (i) MD is *symplectic*. That is, if $\{\boldsymbol{p}_i\}_{i=1}^N$, $\{\boldsymbol{q}_i\}_{i=1}^N$ are the momenta and positions a time step τ_1 and $\{\boldsymbol{p}'_i\}_{i=1}^N$, $\{\boldsymbol{q}'_i\}_{i=1}^N$ are those at a different time step τ_2, then

$$\frac{\partial p'_{i,\alpha}}{\partial p_{i,\alpha}}\frac{\partial q'_{i,\alpha}}{\partial q_{i,\alpha}} - \frac{\partial p'_{i,\alpha}}{\partial q_{i,\alpha}}\frac{\partial q'_{i,\alpha}}{\partial p_{i,\alpha}} = 1, \tag{3}$$

for all $i = 1, ..., N$ and $\alpha = x, y, z$. This crucially implies conservation of volume in phase space [43]. (ii) MD is *time-reversible*. This means that, considering positions and momenta evolved from τ_1 to $\tau_2 > \tau_1$, then evolving $\{-\boldsymbol{p}'_i\}_{i=1}^N$, $\{\boldsymbol{q}'_i\}_{i=1}^N$ for $\tau_2 - \tau_1$ time steps will result in the state $\{-\boldsymbol{p}_i\}_{i=1}^N$, $\{\boldsymbol{q}_i\}_{i=1}^N$. Furthermore, (iii) MD is (approximately) *energy-conserving*. While energy conservation is exact for the solution of Eq. 1, its discretization in the form of Eq. 2 is only approximately energy-conserving due to the finite integration step Δt. Since smaller Δt values afford better energy conservation, the latter is an important metric of integrator quality and sampling accuracy, which is why conventional MD is limited to short time steps and therefore simulation times.

MD as a time series The sequence of configurations generated by a MD trajectory obeys a few additional mathematical properties. (i) MD is *Markovian*. It is trivial to see that Eq. 2 defines a Markovian process, i.e. that the present time step τ contains all the necessary information to predict any future time step $\tau + \Delta\tau$. (ii) MD is *deterministic*. Even though it is possible to describe MD in a probabilistic framework (see e.g. Appendix D), barring effects due to numerical error and parallel programming, the initial conditions determine unequivocally the trajectory, both for infinitesimal and finite time steps. (iii) MD is *chaotic*. Very often, even moderately complex systems simulated with MD exhibit a positive Lyapunov exponent [44] (i.e., the speed at which trajectories diverge), meaning that the phase-space distance of two very close initial states increases exponentially fast as the simulation time progresses. Since MD is executed in finite-precision arithmetic, this implies that the targets of the learning exercise will become excessively noisy (and therefore difficult or impossible to learn) for large strides $\Delta\tau$. Furthermore, the Lyapunov exponent is highly dependent on the physical system under consideration.

2.3 ML modeling of molecular dynamics trajectories

A few previous works have applied ML techniques to model MD trajectories or their target distributions. In the following, we provide a non-exhaustive set of related works broadly categorized by their conceptual approach. We focus on methods that aim to avoid VV integration entirely, rather than on methods based on multiple time-stepping [42] that reduce the computational cost by combining the evaluation of cheap-but-inaccurate ML models and more expensive physics [45, 46] or ML-based approximations of V along the trajectory [47]. We also discuss how the nature and the practical implementation of previous approaches compare with the fundamental properties of MD.

Thermodynamic ensemble generators Generative ML techniques have been applied to directly sample the thermodynamically accessible system configurations [48–51], which is especially useful for biomolecular systems with slow conformational transitions. Such models allow cheap prediction of the thermodynamic properties that conventionally require long MD simulations, but they disregard the time-dependent behavior of the systems and hence are unsuitable for investigating the physicochemical phenomena that can only be explained via the system's *dynamics*. More recently, the implicit transfer operator (ITO) has been proposed to extend Boltzmann generators to also recover, with stochastic trajectory, a measure of long-time dynamical processes [52]. Another relevant approach is `Timewarp` [53], which employs a normalizing flow as a proposal distribution within a Markov chain Monte Carlo scheme targeting the Boltzmann distribution, achieving effective time steps on the order of 10^5–10^6 fs for molecular systems.

Time-series approaches Several works [54–57] have interpreted the MD trajectory as a time series and adopted recurrent neural network (RNN)-type architectures, particularly the long short-

term memory [58] (LSTM), for the learning task. These models take a time series of past system configurations as inputs to make predictions of the future trajectory. Some have taken a stochastic approach to predict a probable *distribution* of system states at a future time, and this has been successfully demonstrated in both all-atomic [54] and coarse-grained [56] contexts. However, in light of the Markovian nature of MD, sequence models such as LSTM use redundant information and are not necessary to learn MD trajectories. Furthermore, the deterministic nature of MD as presented in Eq. 2 makes it superfluous to use probabilistic models (VAEs [59], normalizing flows [60], diffusion models [61], etc.).

Direct MD propagators This is the class of methods that most closely resembles our approach. It is distinct from the former two approaches in that no generative approaches or multiple time step information is used. In this case, the ML models take only the positions and momenta of atoms of the system at time step τ as inputs and deterministically predict their changes at $\tau + \Delta\tau$, hence "directly propagating" the dynamics. In `MDnet` by Zheng et al. [62], the chemical system is described as a graph, with the edge features incorporating both the relative positions and momenta between the atoms. The model then predicts the changes in the positions and momenta for a fixed large time step Δt. Very recently, Thiemann et al. have demonstrated `TrajCast` [63], an autoregressive equivariant network for direct MD prediction. Their framework has been shown to achieve good accuracy in reproducing NVT trajectories for individual molecular or bulk systems at a specific thermodynamic state point. It is also worth mentioning `GICnet` [64] and its transferable, transformer-based variant `MDtrajNet-1` [65], which learns a function that takes as inputs the initial positions, velocity, and Δt to return the positions at time $t + \Delta t$, the Equivariant Graph Neural Operator [66] (EGNO) approach, which predicts the evolution of the system at multiple times using equivariant temporal convolution in Fourier space, and the Graph Network-based Simulators [67, 68] (GNS), which have been developed for arbitrary particle-based systems without the chemical context.

3 The FlashMD framework

In this section, we explain the design choices made for "FlashMD", our proposed approach for the direct learning and prediction of MD trajectories.

3.1 Learning molecular dynamics trajectories with graph neural networks

Given that MD shares with interatomic potentials the properties of $E(3)$-equivariance and the use of atomic geometries as inputs, we propose that FlashMD should be built on top of similar GNN architectures to those that have successfully been used to model machine-learning interatomic potentials (MLIPs). In this work, we choose the Point-Edge Transformer [18] (PET), although any GNN architecture could be used. Compared to the original architecture in Ref. [18], we make two physically motivated modifications to adapt it to the prediction of trajectories: (i) Each node state is also initialized using the particle momentum $\boldsymbol{p}_i$, encoded via a multi-layer perceptron, in addition to the chemical species of the atom under consideration. The initialization of the edge states remains unchanged, and it includes the interatomic vectors $\boldsymbol{q}_j - \boldsymbol{q}_i$. (ii) The outputs $\boldsymbol{p}_i(\tau + \Delta\tau)$ and $\boldsymbol{q}_i(\tau + \Delta\tau)$ are node properties and are therefore predicted via two distinct multi-layer perceptron heads starting from the node representation of PET.

It should be noted that the "raw" FlashMD predictions are chosen to be mass-scaled, i.e. $\boldsymbol{p}'_i/\sqrt{m_i}$ and $(\boldsymbol{q}'_i - \boldsymbol{q}_i)\sqrt{m_i}$. This ensures a treatment of displacements and momenta on equal footing for atoms of different mass, although a data-driven approach is also possible (App. A). Further details on the architecture and the training procedure are available in Appendices A and B, respectively.

3.2 Addressing the many pitfalls of direct molecular dynamics predictions

Despite the fact that we have identified and justified graph neural networks as a highly-suitable model to predict MD trajectories, there still exist many problematic aspects of this exercise which, if ignored, could make the resulting models practically useless. We note that these, as well as many of the theoretical considerations above, have been almost entirely ignored in previous related works.

Out-of-distribution predictions Robust *epistemic* uncertainty schemes, capable of predicting errors associated with limited data sampling, are generally highly recommended when sampling

configurations using MLIPs [69]. They become essential for models that predict MD trajectories directly, that are less physically grounded and more susceptible to exhibiting pathological and unphysical behavior when queried outside of the domain of their training data.

Chaoticity The chaoticity of MD limits the time scale that can be reached with deterministic predictions. It also introduces an *aleatoric* component to the model error, which varies in intensity depending on the system (see Sec. 2.2) and should be accounted for when building an uncertainty quantification scheme.

Time-reversibility Time-reversibility, one consequence of the symmetries of MD, can easily be incorporated by data augmentation [62]. This follows trivially from the discussion in Sec. 2.2.

Conservation of energy Conservation of energy is another consequence of the fundamental symmetries of MD, namely the translational symmetry of time. Given the radical importance of translational symmetries in 3D space, which make GNNs so effective for MLIPs, it is clear that conservation of energy should be considered a centerpiece of MD trajectory modeling, as it encodes the symmetry in the time dimension. Previous works have not carefully monitored conservation of energy in their predicted MD trajectories. In FlashMD, we implement two approaches to improve energy conservation: (i) we utilize errors in energy conservation during training, in addition to the terms corresponding to the position and momenta (see App. B), (ii) we enforce energy conservation at inference time by rescaling momenta after each FlashMD step (see App. C). The latter adjustment makes it possible to run long trajectories targeting the NVE ensemble, as these would otherwise be affected by a large energy drift (see App. F).

Symplecticity Symplectic behavior is – together with energy conservation – a necessary and sufficient condition for correct thermodynamic sampling. Unfortunately, penalizing non-symplecticity in the loss function is impractical, as evaluating Eq. (3) involves the computation of the full $3N \times 3N$ Jacobian – similar to energy conservation in direct force prediction [21]. We will discuss some numerical results on the violation of (3) by FlashMD, but mainly focus on the empirical measures of the accuracy of dynamics and sampling, through comparison with conventional MD simulations.

Symmetry breaking Although equivariant GNNs include strict rotational symmetries in the model, many GNN architectures do not enforce rotational equivariance explicitly [18, 36, 38]. Given that directly learning MD trajectories is a fundamentally more challenging problem than learning a potential energy surface, symmetry breaking might affect models for MD more than MLIPs. Therefore, if using rotationally unrestricted GNNs, we recommend correcting for rotational (and inversion) symmetry breaking at inference time, using similar techniques as those proposed for MLIPs [20]. Given that the PET architecture [18] also does not enforce rotational equivariance, we use rotational and inversion augmentation at training time, and optionally perform random rotation(s) of the system at each step of FlashMD simulations (see App. A).

3.3 Generalization to arbitrary thermodynamic ensembles

FlashMD is trained to reproduce, with a longer stride, NVE trajectories obtained with a VV integrator. However, nearly all other MD variants can be discretized (and are often implemented) using a split-operator formalism where VV is one of the components of the algorithm for a single time-step. This construction, which is further discussed in App. D, allows using FlashMD to accelerate the majority of MD variants and ensembles.

4 Results

To demonstrate the capabilities of FlashMD, we trained two types of models: *water-specific* models trained on a dataset of MD trajectories for liquid water, and general-purpose, *universal* models trained on MD trajectories of structures sampled from the MAD dataset (see Ref. [39]). All reference MD simulations were performed with the PET-MAD universal MLIP [39]. For both the water-specific and universal cases, we trained separate models targeting different time strides. Full training details are available in App. B. The following subsections will focus on the testing of such models in predicting meaningful physical observables for the corresponding systems. Ablation studies are discussed in

App. F, and timings are provided in App. G. Test set accuracy benchmarks against MDNet [62] and TrajCast [63] are reported in App. H and App. I, respectively.

4.1 Liquid water

Liquid water is central to many physical, chemical, biological and environmental processes, in great part thanks to its microscopic hydrogen-bonding network and resulting physical and chemical properties. The study of liquid water at the microscopic level with MD is therefore a very active area of research [70–73], and we use it here as an example of how FlashMD models can accurately predict physical observables at the molecular level. For consistency with the universal model, we train the water-specific model on trajectories obtained with PET-MAD, even though its reference electronic-structure method (PBEsol [74]) is known to grossly overestimate the melting point of water. For this reason, we perform simulations with a target temperature of 450 K, above the melting temperature of this model. Results for the q-TIP4P/f model [75] at room temperature are discussed in Appendix J.

We focus on the evaluation of static observables, i.e. the equilibrium, time-independent properties of a physical system which can be estimated as averages over MD trajectories. As easy-to-compute diagnostics, we investigate the mean kinetic energy and atomic radial distribution functions in NVT simulations, as well as the equilibrium density predicted by NpT simulations. The mean kinetic energies (expressed as effective temperatures) are shown in Table 1. It can be seen that, while the models without energy conservation can show pathological deviations from the target temperatures, the energy conservation enforcement approach described in App. C always recovers the correct temperatures for the overall simulation. However, significant deviations can still be observed for the global stochastic velocity rescaling [30] (SVR) thermostat in the kinetic energies resolved by atom type. This is a spurious effect (classically, each degree of freedom should have a mean kinetic energy equal to $k_B T/2$) which is also observed in non-conservative force models [21]. The link to direct force prediction (that can be seen as the $\Delta t \to 0$ of trajectory prediction) is also suggested by the fact that lack of equipartition is observed also for short-time FlashMD models, that have excellent validation accuracy. As a consequence, one needs to employ *local* thermostats, such as those based on Langevin dynamics, similar to what was done in Bigi et al. [21]. With a judicious choice, one can achieve accurate sampling of equilibrium properties, without reducing substantially sampling efficiency. However, local thermostats disrupt dynamical properties, to an extent that depends on the strength of the thermostat coupling, and a thorough quantitative analysis of dynamical properties require disentangling the effect of the long-stride sampling and that of the thermostats needed to obtain accurate equilibrium sampling (see App. J). For these reasons, we primarily focus our quantitative analysis on time-independent equilibrium properties, and discuss examples where FlashMD qualitatively captures time-dependent behavior.

The atomic radial distribution functions (for the oxygen and hydrogen atoms, respectively), are shown in the left and center panels of Fig. 2. Here, it can be seen that FlashMD is able to correctly reproduce the distribution functions from the reference MD simulations, for both water-specific and universal models. To demonstrate the behavior when simulating the constant-pressure NpT ensemble, we also compute densities at ambient pressure (Fig. 2, right panel). The water-specific FlashMD models are

Table 1: Difference between effective and target temperatures in NVT simulations of liquid water using FlashMD, using different models and thermostats, and comparing results with and without enforcement of energy conservation. Characteristic times of 100 fs and 10 fs were used for the Langevin and SVR thermostats, respectively. The "all", "O", "H" labels refer to the subset of atoms under consideration. Subscripts on the results refer to statistical sampling errors. All units are in Kelvin; numbers close to zero are better.

Model	Without energy conservation						With energy conservation					
	Langevin			SVR			Langevin			SVR		
	ΔT_{all}	ΔT_{O}	ΔT_{H}	ΔT_{all}	ΔT_{O}	ΔT_{H}	ΔT_{all}	ΔT_{O}	ΔT_{H}	ΔT_{all}	ΔT_{O}	ΔT_{H}
Water, 1 fs	$-1.3_{(0.9)}$	$-1.1_{(1.3)}$	$-1.4_{(0.9)}$	$-0.4_{(0.3)}$	$13.8_{(7.0)}$	$-7.4_{(2.3)}$	$-0.3_{(0.8)}$	$-1.6_{(1.4)}$	$0.3_{(0.9)}$	$-0.3_{(0.3)}$	$-5.4_{(4.3)}$	$2.2_{(2.2)}$
Water, 4 fs	$1.4_{(0.9)}$	$-1.4_{(1.2)}$	$2.8_{(1.2)}$	$0.1_{(0.3)}$	$-21.4_{(2.9)}$	$10.8_{(1.4)}$	$-0.4_{(0.9)}$	$-3.6_{(1.3)}$	$1.2_{(1.1)}$	$-0.1_{(0.3)}$	$-20.4_{(2.7)}$	$10.0_{(1.1)}$
Water, 16 fs	$-0.2_{(0.8)}$	$-1.1_{(1.2)}$	$0.2_{(0.8)}$	$-0.1_{(0.4)}$	$-16.0_{(2.9)}$	$7.8_{(1.3)}$	$1.3_{(0.9)}$	$1.2_{(1.0)}$	$1.4_{(1.1)}$	$0.1_{(0.4)}$	$-10.7_{(2.1)}$	$5.4_{(1.2)}$
Universal, 1 fs	$33.8_{(1.0)}$	$36.2_{(1.9)}$	$32.8_{(1.1)}$	$8.0_{(0.4)}$	$-57.5_{(5.5)}$	$40.4_{(1.9)}$	$0.2_{(0.8)}$	$-1.4_{(1.1)}$	$0.9_{(1.0)}$	$-0.5_{(0.4)}$	$-58.6_{(2.8)}$	$28.2_{(1.6)}$
Universal, 4 fs	$10.7_{(1.0)}$	$7.3_{(1.3)}$	$12.5_{(1.3)}$	$9.2_{(1.4)}$	$79.2_{(3.1)}$	$-25.3_{(2.2)}$	$-0.7_{(0.8)}$	$-1.9_{(1.4)}$	$-0.1_{(1.0)}$	$0.4_{(0.4)}$	$-31.2_{(4.6)}$	$16.0_{(2.0)}$
Universal, 16 fs	$-22.5_{(0.7)}$	$-20.9_{(1.1)}$	$-23.5_{(0.9)}$	$-4.0_{(0.4)}$	$7.6_{(2.3)}$	$-9.8_{(1.4)}$	$0.4_{(0.9)}$	$2.9_{(1.3)}$	$0.8_{(1.0)}$	$0.1_{(0.4)}$	$8.6_{(3.0)}$	$-4.1_{(1.4)}$

able to reproduce densities similar (although not statistically equivalent) to the reference MD. The universal models show significant deviations from the reference calculation, although smaller than the typical discrepancy expected when varying the details of the electronic structure calculation.

4.2 Universal long-stride sampling

Having compared and validated both system-specific and universal FlashMD models on the liquid water system, we now proceed to demonstrate the accuracy of the universal FlashMD models for more complex and chemically-diverse systems. We consider three archetypal examples that showcase the relevance for different classes of chemical and materials science problems: (i) We estimate the distribution of Ramachandran angles for alanine dipeptide, a system that exhibits the basic features of protein dynamics and that is often used to benchmark sampling methodologies in biomolecular simulations; (ii) We model the finite-temperature dynamics of the Al(110) surface, a deceptively simple system that exhibits spontaneous formation of defects and surface pre-melting [76]; (iii) We compute the temperature-dependent diffusion coefficient of Li atoms in lithium thiophosphate (LiPS) a material that is being investigated as an electrolyte for solid-state batteries [77, 78], and that allows us to demonstrate the accuracy in capturing time-dependent properties.

Solvated alanine dipeptide We extend upon the previous liquid water simulations, generating constant-pressure trajectories of a single alanine dipetide molecule in solution, following closely the setup described in Morrone et al. [79]. The energy landscape is probed in three different simulations: MD using the PET-MAD potential with 0.5 fs stride, and universal FlashMD with 8 fs and 16 fs strides. Note that for this system 0.5 fs is a limit above which MD simulations exhibit severe instabilities. The Ramachandran plots (Figure 3a) show the characteristic distribution of molecular conformers in terms of the backbone dihedral angles. This is a model for the backbone flexibility of proteins and demonstrates the ability of FlashMD to recover major features of the Ramachandran plot (particularly the low-energy basins in the $-\pi \leq \phi \leq 0$ range) with strides up to 32 times larger.

Metal surface As discussed in a previous work [76], the premelting behavior in Al(110) is characterized by two soft vibrational modes of the surface atoms: the top layer atoms show softening in the [001], or x component, and the second layer atoms show softening in the [110], or z component. Figure 3b shows that the universal Flash MD models correctly describe this trend, with the mean square displacement (MSD) larger for the corresponding surface atoms and their associated softer vibrational modes at 500 K. The dynamical consequences of premelting is presented in the trajectory traces of 3b, which is generated from the FlashMD simulation with 64 fs strides at 600 K. The trajectory traces provide a visual representation of the anisotropic softening of vibration for first and second layer atoms, as well as the dynamic adatom formation pathways involving cooperative migration of both the surface and second layer atoms and effective adatom diffusion via exchange with nearby surface atoms. This demonstrates the ability of FlashMD to not only capture material specific trends, but also describe meaningful dynamical behavior, despite the much larger strides.

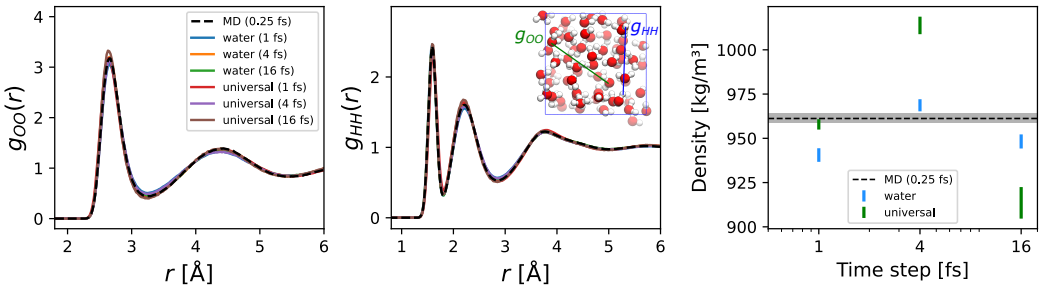

Figure 2: Comparison of physical observables obtained from MD (black) and FlashMD (other colors). Left and Center: radial distribution functions for oxygen and hydrogen atoms, respectively, from simulations in the NVT ensemble using the Langevin thermostat. Right: densities from simulations in the NpT ensemble.

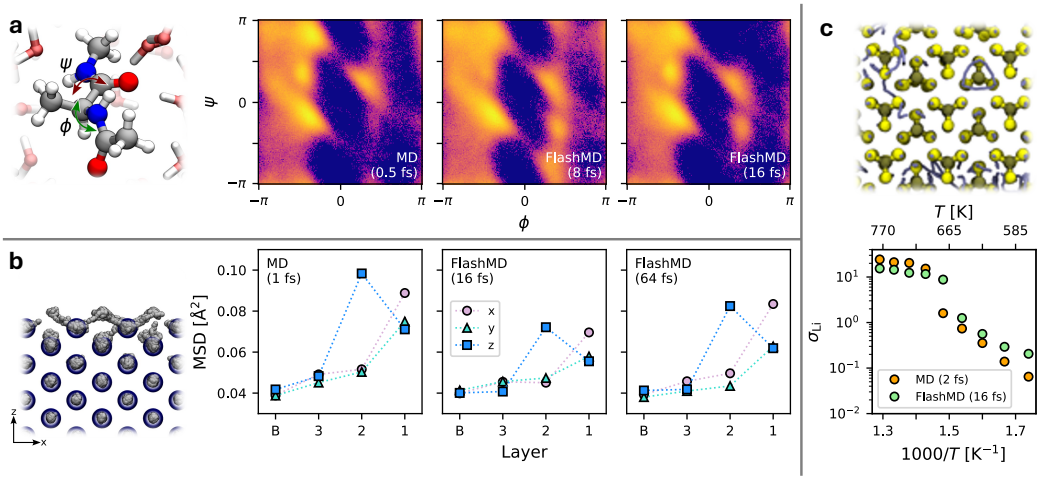

Figure 3: Results of case studies conducted for the universal FlashMD models. (a) Ramachandran plots of the main backbone dihedrals for a simulation of solvated alanine dipeptide at 450 K. (b) Mean square displacement (MSD) of the Al (110) surface atoms at 500 K, at different layers from the surface (B indicates the limiting value for the bulk). The premelting and defect formation phenomena are also visualized as traces of atomic positions from a FlashMD simulation at 600 K, run with $\Delta\tau = 64$ fs. The ideal atomic positions are also shown for reference. (c) Li conductivities of $\gamma-Li_3PS_4$ at varying T, along with the initial system configuration overlaid with traces of the Li atom positions from the FlashMD trajectory at 700 K. In both (b) and (c), traces are obtained with a moving average in time to remove thermal fluctuations and visualize more clearly the diffusive behavior.

Solid-state electrolyte At high temperatures, the γ phase of lithium thiophosphate undergoes a phase transition to a superionic state that exhibits much higher conductivity. We reproduce simulations analogous to those in Ref. [80], computing the Nernst-Einstein conductivity of a LiPS cell as a function of temperature. Results in Figure 3c show that the universal FlashMD model successfully describes the superionic transition of γ Li_3PS_4 and predicts for it to take place at 675 K, within the established transition temperature range determined in previous simulations using PET-MAD [39]. Li conductivities are reasonably matched with the reference MD simulations, albeit with systematic over- and under-estimations in the low and high T regimes, respectively.

5 Discussion

Machine learning has been quietly revolutionizing the atomistic modeling of matter, accelerating the most time-consuming parts of physics-based calculations while striving to retain as much as possible of the underlying physical symmetries and constraints. As datasets and models grow in scale, there is increasing interest in more radical approaches that trade the physical grounding of established practices for computational efficiency. Our work demonstrates that there is enormous potential in constructing GNN models that predict directly the evolution of atomic coordinates and momenta, allowing MD simulations to propagate with long strides, each replacing tens of costly force evaluations with miniscule time steps. Contrary to the very few previous works in this direction, which were restricted to reproducing MD trajectories for a specific system in prescribed thermodynamic conditions, we show that our FlashMD architecture allows one to obtain a *universal* direct MD model that can be applicable to different thermodynamic conditions and ensembles, and to wildly diverse atomic structures and compositions.

This is not to say that circumventing Hamiltonian dynamics is without problems. We highlight several ways an architecture similar to FlashMD could fail, by breaking some of the fundamental symmetries and conservation laws that are obeyed (at least approximately) by conventional integrators. We show how these shortcomings can be mitigated, e.g., by performing energy rescaling at inference time, or by including thermostats to control systematic drifts in the constant-energy trajectories. While the design choices of FlashMD deviate from Hamiltonian dynamics, an alternative framework that preserves symplecticity (and therefore Hamiltonian dynamics) in the direct learning of trajectories is

proposed elsewhere [81]. Many of the shortcomings of non-symplectic FlashMD are shared by non-conservative force models, which have also become fashionable as a tool to accelerate MD. We argue that the transformative speed-up afforded by FlashMD makes direct MD trajectory prediction a more promising approach, at least when performing exploratory studies that require simulating long time scales. As shown concretely by challenging examples that simulate with semi-quantitative accuracy three archetypal systems for biochemistry, surface science and energy technologies, FlashMD can already be applied to realistic simulation problems, capturing the essential equilibrium and dynamical processes while accelerating sampling significantly in all cases (App. G).

In considering potential directions for further development, one should keep in mind that, contrary to the case of ML interatomic potentials that have been studied in great detail and brought to scale over the past decade, there is very little existing research on direct MD prediction. We recognize the possibility of incorporating further constraints in the model architecture, or refinements to the training details, that can better enforce the conservation laws obeyed by the fine-grained VV integrator. One could also investigate scaling up the FlashMD universal model to more parameters and a larger trajectory datasets, or implement a modified architecture that targets multiple time strides with a single model. Given that training relies on short MD trajectories built from a universal MLIP, it is relatively affordable to increase the dataset size by at least one order of magnitude. Moving forward, we envisage a future in which every MLIP would come with its own FlashMD-style long-stride MD companion models, increasing even further the time scales within reach of ML-driven atomic-scale simulations.

Software and data

The FlashMD models presented in this work were trained with the `metatrain` package [82], and they support inference in multiple simulation engines (including ASE [83], i-PI [84], LAMMPS [85]) through `metatomic` [82].

Helper functions to download universal FlashMD models and to prepare simulations are distributed with the `flashmd` package available on PyPI. Further information and instructions can be found at `https://flashmd.org`, including links to the training datasets and scripts to reproduce the reported results on `HuggingFace` and Materials Cloud [86]. An example of the use of FlashMD is also available at `https://atomistic-cookbook.org/examples/flashmd/flashmd-demo.html`.

Besides the universal models presented in this work, which were trained to reproduce molecular dynamics at the PBEsol level of theory (through the PET-MAD universal potential), we also make available FlashMD models based on the more accurate r^{2}SCAN functional which we recommend for scientific use.

Acknowledgments and Disclosure of Funding

The authors would like to thank Davide Tisi and Federico Grasselli for help in processing LiPS trajectories. FB and MC acknowledge support from the NCCR MARVEL, funded by the Swiss National Science Foundation (SNSF, grant number 182892) and from the Swiss Platform for Advanced Scientific Computing (PASC). MC and SC acknowledge funding from the Swiss National Science Foundation (Project 200020_214879).

References

[1] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation*. Academic Press, London, second edition, 2002.

[2] David E. Shaw, Kevin J. Bowers, Edmond Chow, Michael P. Eastwood, Douglas J. Ierardi, John L. Klepeis, Jeffrey S. Kuskin, Richard H. Larson, Kresten Lindorff-Larsen, Paul Maragakis, Mark A. Moraes, Ron O. Dror, Stefano Piana, Yibing Shan, Brian Towles, John K. Salmon, J. P. Grossman, Kenneth M. Mackenzie, Joseph A. Bank, Cliff Young, Martin M. Deneroff, and Brannon Batson. Millisecond-scale molecular dynamics simulations on Anton. In *Proc. Conf. High Perform. Comput. Netw. Storage Anal. - SC 09*, page 1, New York, New York, USA, 2009. ACM Press. ISBN 978-1-60558-744-8. doi: 10.1145/1654059.1654099.

[3] Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. doi: 10.1126/science.1208351.

[4] Paul Robustelli, Stefano Piana, and David E. Shaw. Mechanism of coupled folding-upon-binding of an intrinsically disordered protein. *Journal of the American Chemical Society*, 142 (25):11092–11101, 2020. doi: 10.1021/jacs.0c03217.

[5] Yohan Lee, Sanggyu Chong, Chiwoo Lee, Jihan Kim, and Siyoung Q. Choi. Structural determinants of chirally selective transport of amino acids through the α-hemolysin protein nanopores of free-standing planar lipid membranes. *Nano Letters*, 24(2):681–687, 2024. doi: 10.1021/acs.nanolett.3c03976.

[6] Yun Kyung Shin, Hyunwook Kwak, Alex V. Vasenkov, Debasis Sengupta, and Adri C.T. van Duin. Development of a reaxff reactive force field for fe/cr/o/s and application to oxidation of butane over a pyrite-covered cr2o3 catalyst. *ACS Catalysis*, 5(12):7226–7236, 2015. doi: 10.1021/acscatal.5b01766.

[7] Pietro Vidossich, Agustí Lledós, and Gregori Ujaque. First-principles molecular dynamics studies of organometallic complexes and homogeneous catalytic processes. *Accounts of Chemical Research*, 49(6):1271–1278, 2016. doi: 10.1021/acs.accounts.6b00054.

[8] Christoph K. Jung, Laura Braunwarth, and Timo Jacob. Grand canonical reaxff molecular dynamics simulations for catalytic reactions. *Journal of Chemical Theory and Computation*, 15(11):5810–5816, 2019. doi: 10.1021/acs.jctc.9b00687.

[9] Norman J. Wagner, Brad Lee Holian, and Arthur F. Voter. Molecular-dynamics simulations of two-dimensional materials at high strain rates. *Phys. Rev. A*, 45:8457–8470, Jun 1992. doi: 10.1103/PhysRevA.45.8457.

[10] Thomas E. III Gartner and Arthi Jayaraman. Modeling and simulations of polymers: A roadmap. *Macromolecules*, 52(3):755–786, 2019. doi: 10.1021/acs.macromol.8b01836.

[11] Sanggyu Chong, Sven M. J. Rogge, and Jihan Kim. Tunable electrical conductivity of flexible metal–organic frameworks. *Chemistry of Materials*, 34(1):254–265, 2022. doi: 10.1021/acs.chemmater.1c03236.

[12] R Car and M Parrinello. Unified Aproach for Molecular Dynamics and Density-Functional theory. R. Car and M. Parrinello.pdf. *Phys. Rev. Lett.*, 55(22):2471–2474, 1985.

[13] Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.*, 98(14):146401, April 2007. ISSN 0031-9007. doi: 10.1103/PhysRevLett.98.146401.

[14] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.*, 104 (13):136403, April 2010. ISSN 0031-9007. doi: 10.1103/PhysRevLett.104.136403.

[15] Han Wang, Linfeng Zhang, Jiequn Han, and Weinan E. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications*, 228:178–184, July 2018. ISSN 00104655. doi: 10.1016/j.cpc.2018.03.016.

[16] Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Adv. Neural Inf. Process. Syst.*, 2022.

[17] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J. Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nat Commun*, 14(1):579, February 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36329-y.

[18] Sergey Pozdnyakov and Michele Ceriotti. Smooth, exact rotational symmetrization for deep learning on point clouds. In *Adv. Neural Inf. Process. Syst.*, volume 36, pages 79469–79501. Curran Associates, Inc., 2023.

[19] Ryan Jacobs, Dane Morgan, Siamak Attarian, Jun Meng, Chen Shen, Zhenghao Wu, Clare Yijia Xie, Julia H. Yang, Nongnuch Artrith, Ben Blaiszik, Gerbrand Ceder, Kamal Choudhary, Gabor Csanyi, Ekin Dogus Cubuk, Bowen Deng, Ralf Drautz, Xiang Fu, Jonathan Godwin, Vasant Honavar, Olexandr Isayev, Anders Johansson, Boris Kozinsky, Stefano Martiniani, Shyue Ping Ong, Igor Poltavsky, KJ Schmidt, So Takamoto, Aidan P. Thompson, Julia Westermayr, and Brandon M. Wood. A practical guide to machine learning interatomic potentials – status and future. *Current Opinion in Solid State and Materials Science*, 35:101214, 2025. doi: https://doi.org/10.1016/j.cossms.2025.101214.

[20] Marcel F Langer, Sergey N Pozdnyakov, and Michele Ceriotti. Probing the effects of broken symmetries in machine learning. *Mach. Learn.: Sci. Technol.*, 5(4):04LT01, December 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad86a0.

[21] Filippo Bigi, Marcel Langer, and Michele Ceriotti. The dark side of the forces: assessing non-conservative force models for atomistic machine learning. *arXiv preprint arXiv:2412.11569*, 2024.

[22] Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

[23] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, September 2019. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aaw1147.

[24] Loup Verlet. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159(1):98–103, July 1967. ISSN 0031-899X. doi: 10.1103/PhysRev.159.98.

[25] Hans C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.*, 72(4):2384–2393, 1980. ISSN 00219606. doi: dU/dV.

[26] M. Parrinello and A. Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.*, 52(12):7182–7190, 1981. ISSN 00218979. doi: 10.1063/1.328693.

[27] A Laio and M Parrinello. Escaping free-energy minima. *Proc. Natl. Acad. Sci.*, 99(20):12562–12566, 2002.

[28] M Parrinello and A Rahman. Study of an F center in molten KCl. *J. Chem. Phys.*, 80:860, 1984.

[29] G Bussi and M Parrinello. Accurate sampling using Langevin dynamics. *Phys. Rev. E*, 75(5):56707, 2007. doi: 10.1103/PhysRevE.75.056707.

[30] G Bussi, D Donadio, and M Parrinello. Canonical sampling through velocity rescaling. *J. Chem. Phys.*, 126(1):14101, 2007.

[31] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.*, 31(9):3564–3572, May 2019. ISSN 0897-4756, 1520-5002. doi: 10.1021/acs.chemmater.9b01294.

[32] Johannes Klicpera, Florian Becker, and Stephan Günnemann. GemNet: Universal directional graph neural networks for molecules. *arxiv:2106.08903*, 2021.

[33] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, Nov 2022. doi: 10.1038/s43588-022-00349-3.

[34] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J. Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, Sep 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00716-3.

[35] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Fabio Falcioni, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2024.

[36] Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential. *arXiv preprint arXiv:2410.22570*, 2024.

[37] Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, Matthew Horton, Robert Pinsler, Andrew Fowler, Daniel Zügner, Tian Xie, Jake Smith, Lixin Sun, Qian Wang, Lingyu Kong, Chang Liu, Hongxia Hao, and Ziheng Lu. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

[38] Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of Chemical Theory and Computation*, 20(11):4857–4868, 2024. doi: 10.1021/acs.jctc.4c00190.

[39] Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, and Michele Ceriotti. Pet-mad, a universal interatomic potential for advanced materials modeling. *arXiv preprint arXiv:2503.14118*, 2025.

[40] Haochen Yu, Matteo Giantomassi, Giuliana Materzanini, Junjie Wang, and Gian-Marco Rignanese. Systematic assessment of various universal machine-learning interatomic potentials. *arXiv preprint arXiv:2403.05729*, 2024.

[41] Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

[42] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97(3):1990, 1992. ISSN 00219606. doi: 10.1063/1.463137.

[43] Mark Tuckerman. *Statistical Mechanics and Molecular Simulations*. Oxford University Press, 2008.

[44] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, Mar 1980. ISSN 1572-9648. doi: 10.1007/BF02128236. URL https://doi.org/10.1007/BF02128236.

[45] Venkat Kapil, Jörg Behler, and Michele Ceriotti. High order path integrals made easy. *J. Chem. Phys.*, 145(23):234103, December 2016. ISSN 00219606. doi: 10.1063/1.4971438.

[46] Kevin Rossi, Veronika Jurásková, Raphael Wischert, Laurent Garel, Clémence Corminboeuf, and Michele Ceriotti. Simulating Solvation and Acidity in Complex Mixtures with First-Principles Accuracy: The Case of CH_3SO_3H and H_2O_2 in Phenol. *J. Chem. Theory Comput.*, 16(8):5139–5149, August 2020. ISSN 1549-9618, 1549-9626. doi: 10.1021/acs.jctc.0c00362.

[47] Lars L. Schaaf, Ilyes Batatia, Christoph Brunken, Thomas D. Barrett, and Jules Tilly. Boostmd: Accelerating molecular sampling by leveraging ml force field features from previous time-steps. *arXiv preprint arXiv:2412.18633*, 2024.

[48] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[49] Giacomo Janson, Gilberto Valdes-Garcia, Lim Heo, and Michael Feig. Direct generation of protein conformational ensembles via machine learning. *Nature Communications*, 14(1):774, Feb 2023.

[50] Leon Klein and Frank Noé. Transferable boltzmann generators. *arXiv preprint arXiv:2406.14426*, 2024.

[51] Selma Moqvist, Weilong Chen, Mathias Schreiner, Feliks Nüske, and Simon Olsson. Thermodynamic interpolation: A generative approach to molecular thermodynamics and kinetics. *Journal of Chemical Theory and Computation*, 21(5):2535–2545, Mar 2025.

[52] Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit Transfer Operator Learning: Multiple Time-Resolution Models for Molecular Dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 36449–36462. Curran Associates, Inc., 2023.

[53] Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52863–52883. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a598c367280f9054434fdcc227ce4d38-Paper-Conference.pdf.

[54] Sun-Ting Tsai, En-Jui Kuo, and Pratyush Tiwary. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature Communications*, 11(1):5115, Oct 2020.

[55] J C S Kadupitiya, Geoffrey C Fox, and Vikram Jadhao. Solving newton's equations of motion with large timesteps using recurrent neural networks based operators. *Machine Learning: Science and Technology*, 3(2):025002, apr 2022.

[56] Xiang Fu, Tian Xie, Nathan J. Rebello, Bradley D. Olsen, and Tommi Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *arXiv preprint arXiv:2204.10348*, 2022.

[57] Guy Dayhoff and Sameer Varma. Mlmd: Machine learning velocities to propagate molecular dynamics simulations. *ChemRxiv*, 2025. doi: 10.26434/chemrxiv-2025-ds2pb.

[58] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[59] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[60] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[61] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

[62] Tianze Zheng, Weihao Gao, and Chong Wang. Learning large-time-step molecular dynamics with graph neural networks. *arXiv preprint arXiv:2111.15176*, 2021.

[63] Fabian L. Thiemann, Thiago Reschützegger, Massimiliano Esposito, Tseden Taddese, Juan D. Olarte-Plata, and Fausto Martelli. Force-free molecular dynamics through autoregressive equivariant networks. *arXiv preprint arXiv:2503.23794*, 2025.

[64] Fuchun Ge, Lina Zhang, Yi-Fan Hou, Yuxinxin Chen, Arif Ullah, and Pavlo O. Dral. Four-dimensional-spacetime atomistic artificial intelligence models. *The Journal of Physical Chemistry Letters*, 14(34):7732–7743, 2023. doi: 10.1021/acs.jpclett.3c01592.

[65] Fuchun Ge and Pavlo O. Dral. Artificial intelligence for direct prediction of molecular dynamics across chemical space. *arXiv preprint arXiv:2505.16301*, 2025.

[66] Minkai Xu, Jiaqi Han, Aaron Lou, Jean Kossaifi, Arvind Ramanathan, Kamyar Azizzadenesheli, Jure Leskovec, Stefano Ermon, and Anima Anandkumar. Equivariant graph neural operator for modeling 3d dynamics. *arXiv preprint arXiv:2401.11037*, 2024.

[67] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. *arXiv preprint arXiv:2002.09405*, 2020.

[68] Yulia Rubanova, Alvaro Sanchez-Gonzalez, Tobias Pfaff, and Peter Battaglia. Constraint-based graph network simulator. *arXiv preprint arXiv:2112.09161*, 2021.

[69] Federico Grasselli, Sanggyu Chong, Venkat Kapil, Silvia Bonfanti, and Kevin Rossi. Uncertainty in the era of machine learning for atomistic modeling. *arXiv preprint arXiv:2503.09196*, 2025.

[70] Michele Ceriotti, Jér'me Cuny, Michele Parrinello, and David E. Manolopoulos. Nuclear quantum effects and hydrogen bond fluctuations in water. *Proc. Natl. Acad. Sci. U. S. A.*, 110 (39):15591–15596, September 2013. ISSN 00278424. doi: 10.1073/pnas.1308560110.

[71] Gregory R. Medders, Andreas W. Götz, Miguel A. Morales, Pushp Bajaj, and Francesco Paesani. On the representation of many-body interactions in water. *J. Chem. Phys.*, 143(10): 104102, September 2015. ISSN 0021-9606. doi: 10.1063/1.4930194.

[72] Mariana Rossi, Michele Ceriotti, and David E. Manolopoulos. Nuclear Quantum Effects in H+ and OH- Diffusion along Confined Water Wires. *J. Phys. Chem. Lett.*, 7(15):3001–3007, August 2016. ISSN 19487185. doi: 10.1021/acs.jpclett.6b01093.

[73] Bingqing Cheng, Mandy Bethkenhagen, Chris J. Pickard, and Sebastien Hamel. Phase behaviours of superionic water at planetary conditions. *Nat. Phys.*, 17(11):1228–1232, November 2021. ISSN 1745-2473, 1745-2481. doi: 10.1038/s41567-021-01334-9.

[74] John P. Perdew, Adrienn Ruzsinszky, Gábor I. Csonka, Oleg A. Vydrov, Gustavo E. Scuseria, Lucian A. Constantin, Xiaolan Zhou, and Kieron Burke. Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces. *Phys. Rev. Lett.*, 100(13):136406, April 2008. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.100.136406.

[75] Scott Habershon, Thomas E Markland, and David E Manolopoulos. Competing quantum effects in the dynamics of a flexible water model. *J. Chem. Phys.*, 131(2):24501, July 2009. ISSN 1089-7690. doi: 10.1063/1.3167790.

[76] Nicola Marzari, David Vanderbilt, Alessandro De Vita, and M. C. Payne. Thermal Contraction and Disordering of the Al(110) Surface. *Phys. Rev. Lett.*, 82(16):3296–3299, April 1999. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.82.3296.

[77] Li-Peng Hou, Xue-Qiang Zhang, Nan Yao, Xiang Chen, Bo-Quan Li, Peng Shi, Cheng-Bin Jin, Jia-Qi Huang, and Qiang Zhang. An encapsulating lithium-polysulfide electrolyte for practical lithium–sulfur batteries. *Chem*, 8(4):1083–1098, Apr 2022. ISSN 2451-9294. doi: 10.1016/j.chempr.2021.12.023. URL https://doi.org/10.1016/j.chempr.2021.12.023.

[78] Xin Gao, Zhiao Yu, Jingyang Wang, Xueli Zheng, Yusheng Ye, Huaxin Gong, Xin Xiao, Yufei Yang, Yuelang Chen, Sharon E. Bone, Louisa C. Greenburg, Pu Zhang, Hance Su, Jordan Affeld, Zhenan Bao, and Yi Cui. Electrolytes with moderate lithium polysulfide solubility for high-performance long-calendar-life lithium–sulfur batteries. *Proceedings of the National Academy of Sciences*, 120(31):e2301260120, 2023. doi: 10.1073/pnas.2301260120.

[79] Joseph A. Morrone, Thomas E. Markland, Michele Ceriotti, and B. J. Berne. Efficient multiple time scale molecular dynamics: Using colored noise thermostats to stabilize resonances. *J. Chem. Phys.*, 134(1):14103, January 2011. ISSN 00219606. doi: 10.1063/1.3518369.

[80] Lorenzo Gigli, Davide Tisi, Federico Grasselli, and Michele Ceriotti. Mechanism of Charge Transport in Lithium Thiophosphate. *Chem. Mater.*, 36(3):1482–1496, February 2024. ISSN 0897-4756, 1520-5002. doi: 10.1021/acs.chemmater.3c02726.

[81] Filippo Bigi and Michele Ceriotti. Learning the action for long-time-step simulations of molecular dynamics. *arXiv preprint arXiv:2508.01068*, 2025.

[82] Filippo Bigi, Joseph W Abbott, Philip Loche, Arslan Mazitov, Davide Tisi, Marcel F Langer, Alexander Goscinski, Paolo Pegolo, Sanggyu Chong, Rohit Goswami, et al. Metatensor and metatomic: foundational libraries for interoperable atomistic machine learning. *arXiv preprint arXiv:2508.15704*, 2025.

[83] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

[84] Yair Litman, Venkat Kapil, Yotam M. Y. Feldman, Davide Tisi, Tomislav Begušić, Karen Fidanyan, Guillaume Fraux, Jacob Higer, Matthias Kellner, Tao E. Li, Eszter S. Pós, Elia Stocco, George Trenins, Barak Hirshberg, Mariana Rossi, and Michele Ceriotti. I-PI 3.0: A flexible and efficient framework for advanced atomistic simulations. *J. Chem. Phys.*, 161(6): 062504, August 2024. ISSN 0021-9606, 1089-7690. doi: 10.1063/5.0215869.

[85] Aidan P Thompson, H Metin Aktulga, Richard Berger, Dan S Bolintineanu, W Michael Brown, Paul S Crozier, Pieter J In't Veld, Axel Kohlmeyer, Stan G Moore, Trung Dac Nguyen, et al. Lammps-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer physics communications*, 271:108171, 2022.

[86] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper Welzel Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. Materials Cloud, a platform for open computational science. *Sci Data*, 7(1):299, December 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-00637-5.

[87] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[88] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[89] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter*, 29(27):273002, July 2017. ISSN 0953-8984, 1361-648X. doi: 10.1088/1361-648X/aa680e.

[90] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[91] Benedict Leimkuhler and Charles Matthews. Robust and efficient configurational molecular sampling via Langevin dynamics. *J. Chem. Phys.*, 138(17), 2013. ISSN 00219606. doi: 10.1063/1.4802990.

[92] Giovanni Bussi, Tatyana Zykova-Timan, and Michele Parrinello. Isothermal-isobaric molecular dynamics using stochastic velocity rescaling. *J. Chem. Phys.*, 130(7):074101, February 2009. ISSN 0021-9606. doi: 10.1063/1.3073889.

[93] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[94] Michael F. Herbst, Antoine Levitt, and Eric Cancès. *A Posteriori* error estimation for the non-self-consistent Kohn–Sham equations. *Faraday Discuss.*, 224:227–246, 2020. ISSN 1359-6640, 1364-5498. doi: 10.1039/D0FD00048E.

[95] Michael F. Herbst and Thomas Fransson. Quantifying the error of the core–valence separation approximation. *J. Chem. Phys.*, 153(5):054114, August 2020. ISSN 0021-9606, 1089-7690. doi: 10.1063/5.0013538.

[96] Michael F Herbst and Antoine Levitt. Black-box inhomogeneous preconditioning for self-consistent field iterations in density functional theory. *J. Phys.: Condens. Matter*, 33(8): 085503, February 2021. ISSN 0953-8984, 1361-648X. doi: 10.1088/1361-648X/abcbdb.

[97] Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia Vogt. Effective bayesian heteroscedastic regression with deep neural networks. *Advances in Neural Information Processing Systems*, 36:53996–54019, 2023.

[98] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.

[99] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable Bayesian optimization using deep neural networks. In *ICML*, 2015.

[100] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *ICML*, 2020.

[101] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in neural information processing systems*, 34:20089–20103, 2021.

[102] Filippo Bigi, Sanggyu Chong, Michele Ceriotti, and Federico Grasselli. A prediction rigidity formalism for low-cost uncertainties in trained neural networks. *Mach. Learn.: Sci. Technol.*, 5(4):045018, December 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad805f.

[103] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.

[104] Mahdi Hijazi, David M. Wilkins, and Michele Ceriotti. Fast-forward Langevin dynamics with momentum flips. *J. Chem. Phys.*, 148(18):184109, May 2018. ISSN 00219606. doi: 10.1063/1.5029833.

[105] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry*, 30(13):2157–2164, 2009. doi: https://doi.org/10.1002/jcc.21224.

[106] Zoran Šućur and Vojtěch Spiwok. Sampling enhancement and free energy prediction by the flying gaussian method. *Journal of Chemical Theory and Computation*, 12(9):4644–4650, 2016. doi: 10.1021/acs.jctc.6b00551.

[107] Michele Ceriotti, Giovanni Bussi, and Michele Parrinello. Langevin equation with colored noise for constant-temperature molecular dynamics simulations. *Phys. Rev. Lett.*, 102(2): 020601, January 2009. ISSN 00319007. doi: 10.1103/PhysRevLett.102.020601.

A Model details

A.1 Why predict $q'_i - q_i$ and p'_i: small- and large-time limits

It should be noted that the models for liquid argon in the early work by Zheng et al. [62] effectively predict $\boldsymbol{q}'_i - \boldsymbol{q}_i - \boldsymbol{p}_i \Delta t / m_i$ and $\boldsymbol{p}'_i - \boldsymbol{p}_i$. While the additional terms with respect to FlashMD ($-\boldsymbol{p}_i \Delta t / m_i$ and $-\boldsymbol{p}_i$, respectively) reduce the variance of the targets for small time steps and/or very smooth potential energy surfaces (such as Lennard-Jones argon in Ref. [62]), they instead increase it for more complex systems and larger predicted time steps, which are the focus of the present work. This is a consequence of more complex systems generally having smaller position and momentum correlation times. As a result, we do not shift the targets by these additional terms in our work.

A.2 Predicting mass-scaled positions and momenta

All models shown in this work are trained on, and therefore predict, mass-scaled displacements and momenta defined as $\Delta\tilde{\boldsymbol{q}}_i = \Delta\boldsymbol{q}_i \sqrt{m_i}$ and $\tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i / \sqrt{m_i}$, respectively, where m_i is the mass of atom i. This is aimed at making the scales of the training targets uniform across atoms of potentially very different mass, and it is of fundamental importance for models trained on the whole periodic table. This prevents, for example, the displacement of light atoms or the momenta of heavy atoms from dominating the loss, instead leading to good predictions for all atoms, independent of their mass. At prediction time, a simple scaling using the masses recovers the conventional displacement and momenta. We found that a similar, but data-driven, approach can provide the same benefits. This consists of using different standardization factors for different chemical elements, so that training displacements and momenta are scaled to unit standard deviation during training, *for each chemical element* (i.e., two scaling factors are used for every single chemical element in the dataset: one for displacements, one for momenta).

A.3 Center-of-mass enforcement

Using Eqs. 2 to evolve a system without external forces naturally leads to conservation of total momentum of the system, i.e.,

$$\sum_{i=1}^{N} \boldsymbol{p}'_i - \sum_{i=1}^{N} \boldsymbol{p}_i = \boldsymbol{0}. \tag{4}$$

Since the total momentum is constant, the center of mass of the system follows a uniform linear motion, i.e.,

$$\sum_{i=1}^{N} m_i \boldsymbol{q}'_i - \sum_{i=1}^{N} m_i \boldsymbol{q}_i = \Delta t \sum_{i=1}^{N} \boldsymbol{p}_i. \tag{5}$$

Both conditions are enforced within the model to avoid unphysical drift effects during molecular dynamics simulations (Fig. 1). We note that removing the center-of-mass motion entirely would not be correct in the general case, although many MD simulations are performed with this additional constraint. Although we enforced these contraints within the model in this work, we recommend applying them at inference time only in order not to break the locality assumption that underpins our approach.

A.4 Optional inference-time filters

Within FlashMD, we have implemented "filters" (Fig. 1) that can be employed at inference-time to mitigate the artifacts of direct MD prediction. We refer to the dedicated sections for energy conservation enforcement (App. C) and thermodynamic ensemble control (App. D). Here we only provide a discussion of the random rotation filter.

Since equivariance is not exactly preserved and only learned via data augmentation in the case of unrestricted architectures such as PET [18], simulations performed with the resulting FlashMD model would be prone to spurious effects. To mitigate this, we adopt the strategy proposed Langer et al. [20], where random rotations of the simulated system are performed to average out any artifacts of non-equivariance along the MD trajectory. Implementation details of the random rotation filter is shown in Fig. 4. We note that in case of GNNs that preserve rotational symmetry, this filter is not needed – and would actually have no effect if applied.

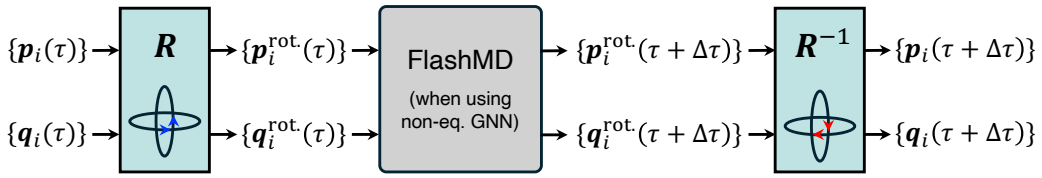

Figure 4: Implementation of the random rotation filter. A random rotation matrix $\boldsymbol{R}$ is sampled and applied on all coordinates and momenta before the rotated inputs are provided to FlashMD. After model inference, $\boldsymbol{R}^{-1}$ is applied to rotate the system back to the original coordinate reference. Random rotation filter is only relevant for rotationally unconstrained GNNs.

B Training details

B.1 Loss function, optimization, normalization

All models are implemented in PyTorch [87] and, unless specified otherwise (see App. E), are trained using a loss function given by the sum of two mean square error terms, for the mass-scaled momenta and the positions respectively:

$$\mathcal{L} = \sum_{s=1}^{N_{\text{train}}} \frac{1}{3N_s} \sum_{i=1}^{N_s} (\tilde{\boldsymbol{p}}'_i - \tilde{\boldsymbol{p}}'_{i,\text{ref}})^2 + (\Delta\tilde{\boldsymbol{q}}'_i - \Delta\tilde{\boldsymbol{q}}'_{i,\text{ref}})^2, \tag{6}$$

where s is an index for structures in the training set and N_s is the number of atoms in structure s. In order to ensure similar weight in the loss function between position and momentum terms, the mass-scaled positions and momenta are scaled by their standard deviation across the dataset before training.

Optimization is carried out using the Adam [88] optimizer with an initial learning rate of $3 \cdot 10^{-4}$. Learning rate decay is applied at a regular intervals of 100 and 50 epochs for the water and universal models, respectively. Training-time rotational augmentation for vectorial targets is carried out in the same way as in Pozdnyakov and Ceriotti [18].

B.2 Training-time energy conservation

We found that the degree of energy conservation on structures of the validation set correlates well with the quality of the models during molecular dynamics runs. As a result, during training, we choose the best model as the one having the lowest product of three terms, evaluated across the validation set: (i) RMSE on the predictions of $\tilde{\boldsymbol{p}}'$, (ii) RMSE on the predictions of $\tilde{\boldsymbol{q}}'$, and (iii) RMSE on the energy of the resulting structure when compared to the energy of the target structure. While an energy term might also be included in the loss function, we found that it slows down training significantly (due to evaluations of the energy model and its gradients), without improving the quality of the FlashMD models in any measurable way.

Indeed, as shown in App. F, models with similar accuracy on positions and momenta can predict MD states with highly varying degrees of energy conservation. In particular, energy misalignment is often observed if the error on the energies is ignored. Although we found this approach to improve the quality of the simulations afforded by our water models (both PET-MAD and q-TIP4P/f), we found its impact on universal models to be less dramatic.

We also found it useful to compare errors in total energies with familiar metrics such as "chemical accuracy" or the accuracy of the underlying energy model. For all models tested in this work, such comparisons correlate extremely well with the quality of the models in the resulting MD simulations.

B.3 Reference MD trajectory generation

All reference MD trajectories were obtained from simulations performed with PET-MAD [39] (version 1.0), a universal MLIP capable of making reasonably accurate predictions of the potential energy surface across the entire periodic table of elements. All simulations were performed using the Atomic Simulation Environment [89] (ASE) software (version 3.24).

Water-specific models A water structure at experimental density (at 298 K and 1 atm) was equilibrated with PET-MAD (or q-TIP4P/f for the q-TIP4P/f-based water models discussed in Appendix J). Subsequently, two more structures were generated by increasing and decreasing the volume of the cell by 10%, scaling the atomic positions accordingly. For each of the three resulting structures, NVT equilibration runs were performed at all temperatures between 20 and 1000 K, in steps of 20 K, with a time step of 0.5 fs and a duration of 5 ps, using a Langevin thermostat with a characteristic time of 10 fs. Subsequently, each equilibrated structure was used to produce an NVE MD trajectory of 2 ps with a time step of 0.25 fs. Structures for training were extracted from these trajectories every 100 fs, and augmented with their time-reversed version, for a total of 5400 structures.

Universal models 10,000 structures from the MAD dataset, used in the training of PET-MAD [39], baseline MLIP, were randomly selected for reference MD trajectory generation (see Ref. [39] for further details on the MAD dataset). The initial geometry was first energetically optimized with the BFGS algorithm until the maximum force component threshold of 0.01 eV/Å was reached. The energy-optimized system was put through equilibration under the NVT ensemble for 10 ps with timesteps of 0.5 fs. A characteristic time of 100 fs was used in the Langevin thermostat. The final configuration from NVT equilibration was then taken for trajectory production under the NVE ensemble for 2.5 ps with finer timesteps of 0.25 fs. Positions and momenta were recorded every timestep for FlashMD training. Simulations were repeated 10 times for each structure, with a randomly selected temperature between 0 and 1500 K. Structures for training were chosen from these trajectories every 500 fs (5 samples per NVE trajectory to avoid time-correlated samples), and augmented with their time-reversed version, for a total of 1 million structures.

C Enforcing conservation of energy by momentum rescaling

Especially for NVE simulations, it is important to avoid excessive energy drift. In practice, we found that enforcing total energy conservation after each step is beneficial (even for thermostatted runs, see Sec. 4). This is achieved by rescaling the momenta in the following way:

$$\boldsymbol{p}_i' \leftarrow \alpha \boldsymbol{p}_i', \quad \alpha = \sqrt{1 - \frac{E' - E}{K'}}, \tag{7}$$

where E' is the total energy after the step, E is the total energy before the step, and K' is the kinetic energy after the step.

It should be noted, however, that enforcing energy conservation requires one or two energy evaluations per step (depending on the ensemble and integration scheme), potentially introducing significant overhead. Implementation details of the energy conservation enforcement filter in FlashMD is visualized in Fig. 5.

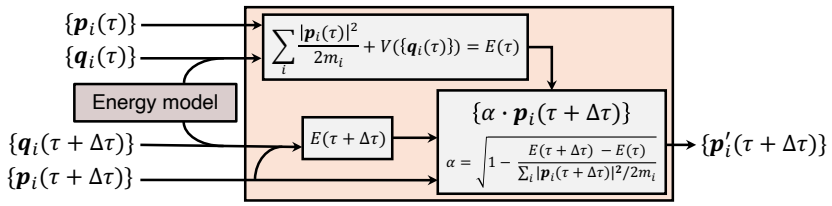

Figure 5: Implementation details of the energy conservation enforcement filter in FlashMD. Energy model can be any model of the interatomic potential (e.g. MLIP, classical force field, etc.) that can be used to compute $V(\{\boldsymbol{q}_i\})$.

D Arbitrary thermodynamic ensembles with FlashMD

Molecular dynamics trajectories conserve, at least approximately, the classical total energy of the system, which makes it appropriate for sampling configurations under constant energy, volume and particle number (NVE) conditions. Several modifications to Hamiltonian dynamics have been proposed [25, 26] to generate configurations consistent with other thermodynamic conditions, such as constant temperature (NVT), constant pressure (NpT), or constant chemical potential (μVT). These ensembles are often more relevant to compare with experimental conditions, that usually do not involve closed, isolated setups – especially not on the small length scales that are used in simulations. Here we discuss two approaches that are routinely used to this end in MD, and how they can be can be combined with FlashMD to extend its constant-energy long-stride integration to sample other ensembles.

First, given that NVE trajectories conserve not only the total energy, but also the probability measures associated with most other ensembles, it is possible to alternate segments of NVE trajectories with discrete Monte Carlo moves changing the particle velocities, the simulation cell size, or the nature of the atoms, using a Metropolis-Hastings criterion [90] to accept or reject them in a way that is consistent with the desired ensemble. This approach can be applied straightforwardly to FlashMD, and its reliability depends on the assumption that segments of FlashMD trajectories are symplectic and energy-preserving to a high degree of accuracy, which is why we give much emphasis to these diagnostics in our study.

The second approach is slightly more subtle and requires some additional technical background, and we discuss and test it in more detail. Integrators for Hamiltonian dynamics can be expressed in a Liouvillian formalism, in which the trajectory density $P(\boldsymbol{q}, \boldsymbol{p})$ is evolved in time according to an operator that combines the time evolution of the different variables, e.g. for Hamiltonian dynamics

$$\mathrm{i}\hat{L} = \sum_i \frac{\partial H}{\partial \boldsymbol{p}_i} \cdot \frac{\partial \square}{\partial \boldsymbol{q}_i} - \frac{\partial H}{\partial \boldsymbol{q}_i} \cdot \frac{\partial \square}{\partial \boldsymbol{p}_i} = \mathrm{i}\hat{L}_q + \mathrm{i}\hat{L}_p \tag{8}$$

Finite-time propagation of the trajectory density can be formally achieved with an exponential operator $e^{\mathrm{i}(\hat{L}_q + \hat{L}_p)\Delta t}$, and the difficulty in developing an exact propagation algorithm for $(\boldsymbol{q}, \boldsymbol{p})$ can be understood as a consequence of the fact that $\hat{L}_q$ and $\hat{L}_p$ do not commute. The error grows with the time step Δt, and can be reduced using symmetric Trotter factorizations such as $e^{\mathrm{i}\hat{L}_p \Delta t/2} e^{\mathrm{i}\hat{L}_q \Delta t} e^{\mathrm{i}\hat{L}_q \Delta t/2}$. This splitting corresponds, in the trajectory picture, to the VV integrator in Eq. (2). Continuous equations of motion that describe other thermodynamic ensembles can be derived with an extended Lagrangian formalism, in which additional structural parameters (e.g. the simulation cell volume V or shape) are associated with fictitious masses and momenta. Starting from the Lagrangian, one can derive a Liouvillian that describes their time evolution as an additional term, e.g. $\hat{L}_V$.

Langevin-type thermostats [29] can also be described with an associated Liouvillian $\hat{L}_\xi$ and are a good example to explain the formalism. There are in fact multiple possible ways to factorize the overall Liouvillian: the so-called OBABO splitting reads $e^{\mathrm{i}\hat{L}_\xi \Delta t/2} e^{\mathrm{i}\hat{L}_p \Delta t/2} e^{\mathrm{i}\hat{L}_q \Delta t} e^{\mathrm{i}\hat{L}_q \Delta t/2} e^{\mathrm{i}\hat{L}_\xi \Delta t/2}$ and corresponds to bracketing a velocity Verlet integrator (BAB) between two finite-time propagators for an Ornstein-Uhlenbeck process (O, the Langevin equation for a free particle with inertia)

$$\begin{aligned}
\boldsymbol{p}_i &\leftarrow e^{-\gamma \Delta t/2} \boldsymbol{p}_i + \sqrt{m_i k_B T (1 - e^{-\gamma \Delta t})} \boldsymbol{\xi}_1 \\
\boldsymbol{p}_i &\leftarrow \boldsymbol{p}_i - \frac{1}{2} \frac{\partial V}{\partial \boldsymbol{q}_i} \Delta t \\
\boldsymbol{q}_i &\leftarrow \boldsymbol{q}_i + \frac{\boldsymbol{p}_i}{m_i} \Delta t \\
\boldsymbol{p}_i &\leftarrow \boldsymbol{p}_i - \frac{1}{2} \frac{\partial V}{\partial \boldsymbol{q}_i} \Delta t \\
\boldsymbol{p}_i &\leftarrow e^{-\gamma \Delta t/2} \boldsymbol{p}_i + \sqrt{m_i k_B T (1 - e^{-\gamma \Delta t})} \boldsymbol{\xi}_2
\end{aligned} \tag{9}$$

where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are vectors of uncorrelated, unit-variance Gaussian random numbers. Note that this splitting preserves the symmetry of the VV integrator. Many other splittings are possible, such as BAOAB (with the Ornstein-Uhlenbeck propagator sandwiched between two half-VV integrators), which has been found to be more accurate for position-dependent observables [91].

From this extensive overview of the Liouville operator formalism and its connection to the theory of integrators, one can see how to incorporate FlashMD into the integration schemes that are used for other thermodynamic ensembles. If the full Liouvillian for a given integrator is $\hat{L}' + \hat{L}_q + \hat{L}_p$, one can factor an integration over $\Delta\tau\Delta t$ as

$$e^{i\hat{L}'\Delta\tau\Delta t/2}(e^{i\hat{L}_p\Delta t/2}e^{i\hat{L}_q\Delta t}e^{i\hat{L}_q\Delta t/2})^{\Delta\tau}e^{i\hat{L}'\Delta\tau\Delta t/2}. \tag{10}$$

The central term is precisely the evolution that FlashMD aims to approximate, and can therefore be readily replaced with one large step on $(\boldsymbol{q}, \boldsymbol{p})$.

There are a few considerations that should be made when designing one of these extended integrators for FlashMD. First, if there are multiple splittings available for the base integrator, one has to choose those that involve a VV core. For instance OBABO can be used, but not BAOAB, and we cannot use the Bussi-Zykova-Parrinello splitting [92], but a more naive one that does not simultaneously update atomic positions and cell vectors. Second, if one wants to further split $e^{i\hat{L}'\Delta\tau\Delta t/2}$, it should be done in a symmetric way, applying the factors in opposite order before and after the FlashMD step. Last, and most importantly, the integration of $e^{i\hat{L}'\Delta\tau\Delta t/2}$ should be accurate also with a large time step, and the factorization with the VV core be similarly accurate, or at least preserve the target ensemble. This implies choosing large effective masses for extended Lagrangian terms (e.g. the cell volume in a constant-pressure integrator). Long time scales for Langevin-type thermostats should also be chosen if one wants to preserve time-dependent properties of the original Langevin dynamics – but this is a lesser concern for sampling accuracy, because usually Langevin-type free-particle integrators preserve the velocity distribution for any time step. The workflow proposed herein for the integration of thermostats and barostats with FlashMD is shown in Fig. 6.

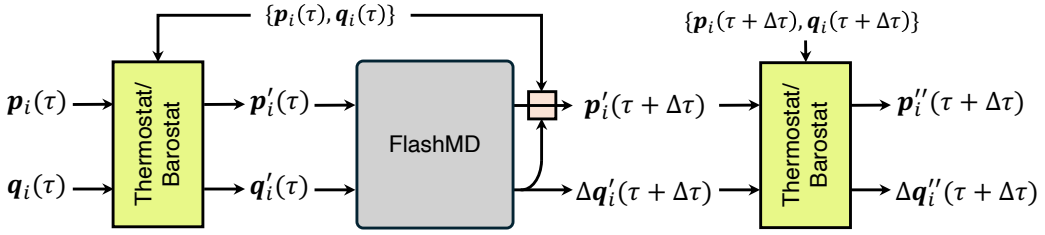

Figure 6: Integration of thermostats and barostats with FlashMD for thermodynamic ensemble control. Note that the integration is performed for the full stride in FlashMD, and the split operators on both ends are applied for half strides.

E Uncertainty quantification

ML models are inherently statistical, introducing different types of error at prediction time. In the context of FlashMD, uncertainty quantification (UQ) becomes an even bigger necessity as the potential error accumulation along a trajectory makes FlashMD simulations prone to many undesirable artifacts, leading to the incorrect sampling of the thermodynamic ensemble, which can manifest itself, for example, as unphysical bond forming/breaking behavior, uncontrolled expansion of the simulated system, etc. In this section, we identify the design choices that are required for robust UQ for MD prediction models and provide simple demonstrations.

In general, there largely exist two different sources of uncertainty for the model: *aleatoric*, irreducible uncertainty stemming from the "noise" in data, and *epistemic*, reducible uncertainty from the model's lack of knowledge [93]. In the training of MLIPs, it is widely assumed (although not necessarily true [94–96]) that the reference data is noise-free, and that it is therefore appropriate to only account for epistemic uncertainty in UQ approaches for MLIPs. In the learning formulation for FlashMD that directly targets time-evolved positions and momenta, we note the potentially significant presence of aleatoric uncertainty due to the chaoticity of the underlying physical problem. This aleatoric contribution to the uncertainty is expected to be strongly heteroscedastic, since different chemical systems exhibit very different degrees of chaotic behavior in molecular dynamics (in other words, the Lyapunov exponent can vary significantly based on the system, see Sec. 2.2). Last but not least, the UQ scheme of choice should not result in significant prediction time overhead in the simulation, as

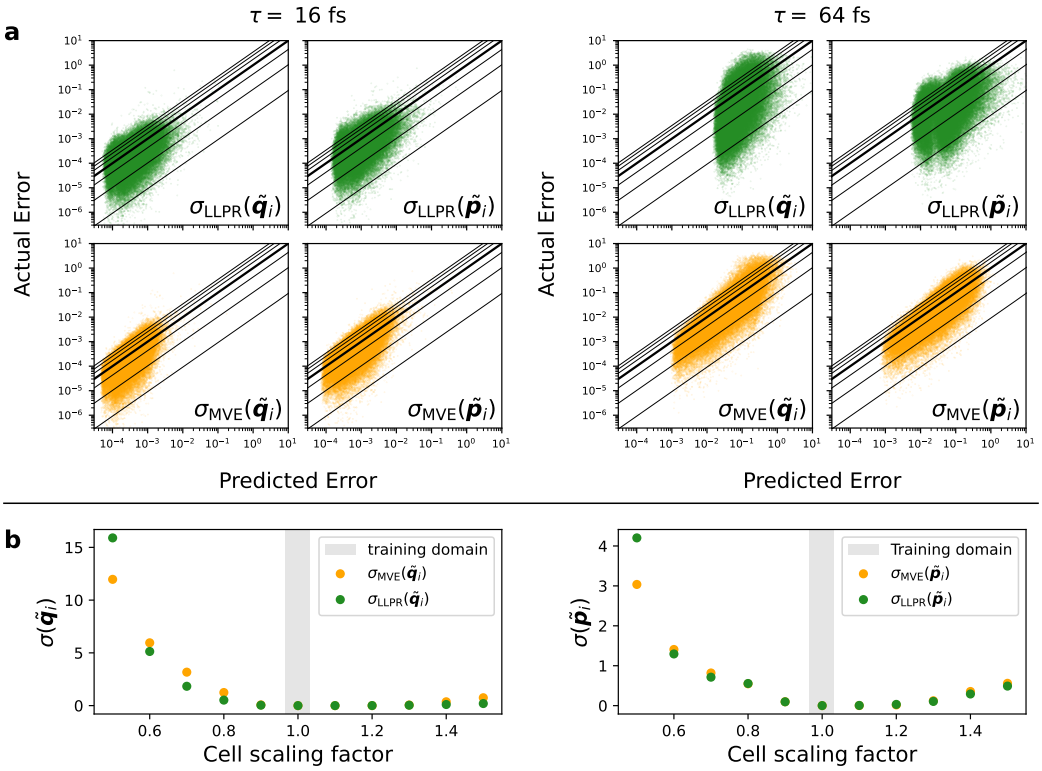

Figure 7: Uncertainty quantification diagnostics for the water-specific FlashMD model on its test set. (a) shows the uncertainty plots (predicted variance on the x axis vs squared residual on the y axis) on mass-scaled positions and momenta ($\tilde{\boldsymbol{q}}_i$ and $\tilde{\boldsymbol{p}}_i$) for two models trained on the water dataset (with 16 fs and 64 fs strides, respectively). Units are eV for ($\tilde{\boldsymbol{p}}_i$) and Å^{2}u for ($\tilde{\boldsymbol{q}}_i$). σ_{LLPR} is shown in green, and σ_{MVE} is shown in orange. The black lines corresponds to the parity line, as well as pairs of iso-probability lines of the ideal distribution containing density equivalent to that contained within 1σ, 2σ, 3σ of a Gaussian distribution. (b) shows the predicted uncertainty, in the same units, for out-of-distribution predictions using the 16 fs model, as a function of the scaling of the cell and the atoms within it.

this would undermine the key advantages of FlashMD. For these reasons, we adopt a UQ scheme that quantifies both types of uncertainties with a near-zero computational overhead. The method is sketched in the blue inset of Fig. 1.

Taking inspiration from Immer et al. [97], we assume the overall uncertainty to arise from a sum of an epistemic and an aleatoric term

$$\sigma^2 = \sigma_a^2 + \sigma_e^2, \tag{11}$$

which are predicted by different types of UQ estimator. For the aleatoric component, the prediction heads are modified to yield mean-variance estimators (MVEs), in which the model predicts mean and variance of the target predictions. FlashMD in this mode is trained to the negative log-likelihood loss

$$\mathcal{L} = \frac{1}{2}\left(\ln \sigma_a^2 + \frac{(y - y_{\mathrm{ref}})^2}{\sigma_a^2}\right), \tag{12}$$

where σ_a^2 is the variance afforded by the mean-variance estimator, parametrized as described in Lakshminarayanan et al. [98]. In our case, the overall loss is obtained by summing one of such terms for mass-scaled positions and one for mass-scaled momenta. A Laplace approximation is usually considered a good model for the epistemic uncertainty σ_e^2, and we implement it as a last-layer approximation (LLPR) [99–102] on the mean part of the MVE, i.e., on y.

We now demonstrate the UQ capabilities of FlashMD, with a special focus on the need for aleatoric uncertainty within direct MD prediction models as discussed in Sec. 3. To do so, we slightly deviate

from Eq. (11) and consider *separately* the MVE and LLPR uncertainty predictions for liquid water and analyze their behavior (Fig. 7a). The $\Delta\tau = 64$ fs model, which is very difficult to learn for the liquid water system (due to the fast correlation times of the physical system), displays the failure of epistemic uncertainty in isolation, exhibiting a narrow spread in values that does not provide meaningful insights. In contrast, the mean-variance estimator provides good uncertainty estimates. Perhaps more surprisingly, uncertainties of the 16 fs model are predicted reasonably by both uncertainty estimators. Since $\Delta\tau = 16$ fs is a more learnable regime for water, where epistemic uncertainty is expected to dominate, this implies that the mean-variance estimator is also capable of capturing epistemic uncertainties to good accuracy, at least within the training distribution. As an out-of-domain example, Fig. 7b shows the average LLPR and MVE uncertainties as a function of the compression (or expansion) factor of a water cell. Even for very compressed or very stretched cells (up to a change of 50% in the cell length), the mean-variance estimator provides a qualitatively correct uncertainty profile, suggesting that a MVE alone might be sufficient to quantify uncertainties in direct MD predictions. We leave a more thorough analysis of combining the two UQ metrics for future work.

F Ablation studies

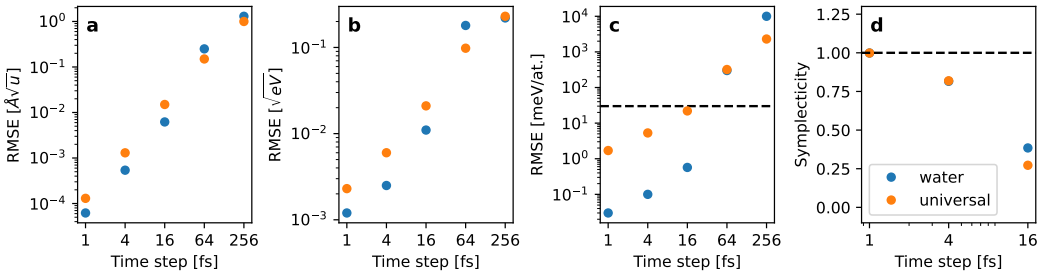

Figure 8: Comparison of validation root mean square errors (RMSEs) in (a) mass-scaled positions, (b) mass-scaled momenta, (c) energy conservation, and (d) symplecticity as a function of the stride for different FlashMD models. All errors are calculated on the respective validation sets, except for the symplecticity errors, which are evaluated using the left-hand side of Eq. 3 for 100 degrees of freedom of a liquid water structure. Dotted line in (c) is the RMSE of PET-MAD, an indirect metric of energy accuracy, and in (d) marks perfect symplecticity.

F.1 Effect of the stride length on training

As a result of the chaoticity effects described in Sec. 2, it is desirable to examine the errors in the training as the predicted time stride increases. Fig. 8 shows the errors on energies and symplecticity, as well as mass-scaled positions and momenta, for the training runs on the water and universal datasets for different time steps. As expected, training becomes extremely difficult after a certain number of steps. From these plots, it would appear that the increase in the machine learning error is not exponential with the time step, but rather polynomial. While the interpretation of this observation is not trivial, it is potentially promising as it would make longer time scales accessible by simply improving the accuracy of the models using more data and/or learnable parameters, without the presence of hard limits to the accuracy. However, the rapid increase in the non-symplectic behavior of the predictions is alarming and worth future investigation.

F.2 Enforcing energy conservation

We will now illustrate the effect of the procedure to enforce energy conservation presented in App. C on a simulation targeting the NVE ensemble. Fig. 9 shows that, while FlashMD exhibits a total energy drift that would quickly lead to a unstable trajectories and large sampling errors, the proposed energy conservation enforcement technique allows for exact conservation of the total energy along the simulation, producing a stable trajectory (even thought it is not guaranteed to sample the NVE ensemble because it is not exactly symplectic). We further show in Sec. 4 that this technique improves the temperature control in NVT simulations.

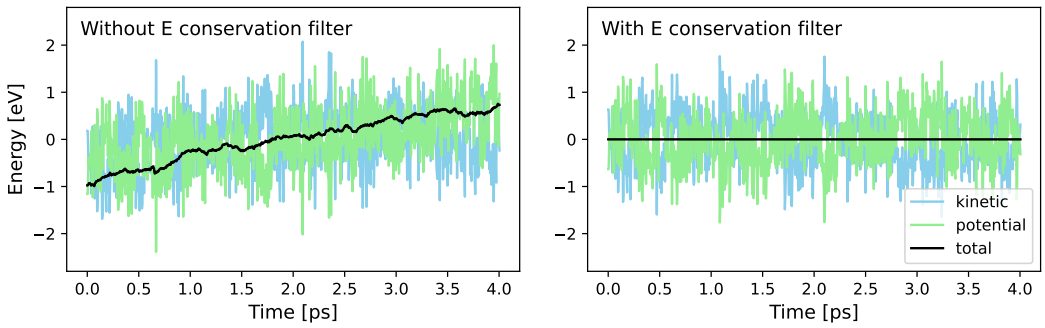

Figure 9: Comparison of the potential, kinetic, and total energies along short trajectories of water FlashMD model with 4 fs strides, with and without the energy conservation enforcement filter. The mean potential/kinetic/total energy is subtracted from all energy profiles.

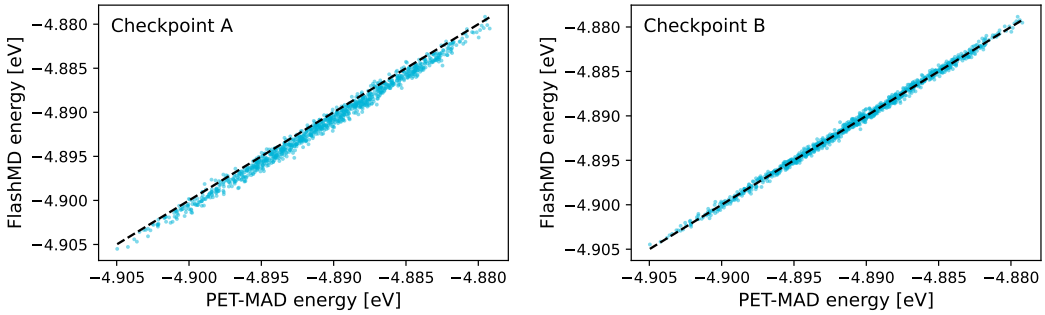

Figure 10: Parity plots of the predicted energy for two water-specific model checkpoints belonging to the same training run. The energies of the structure predicted by the model on the left are misaligned.

F.3 Energy criterion for model selection

App. B describes how the error on the energy of the predicted structures is also used during training to choose good models. Here, to illustrate the utility of such approach, we show models from different epochs in the same training run (a water-specific model trained to predict over a time stride of 16 fs). Despite the two models having similar position and momentum errors, the energies of the first model are misaligned (Fig. 10). In a simulation, such an offset in model prediction would induce a progressive and unphysical cooling of the system.

G Timings

Table 2: Timings, strides, and acceleration factors of FlashMD compared to conventional MD on the systems investigated in this work. For completeness, we also include the speed-up factors we obtain without energy conservation enforcement (ECE), which would slightly degrade the quality of the simulations.

System (# atoms)	Liquid water (192)	Al(110) slab (560)	Solvated alanine dipeptide (622)	Li_3PS_4 (768)
MD timing [stride]	$2.0 \cdot 10^4$ [0.25 fs]	$3.2 \cdot 10^4$ [1 fs]	$2.1 \cdot 10^5$ [0.5 fs]	$1.4 \cdot 10^5$ [2 fs]
FlashMD timing [stride]	$4.0 \cdot 10^2$ [16 fs]	$5.1 \cdot 10^2$ [64 fs]	$1.8 \cdot 10^4$ [16 fs]	$4.1 \cdot 10^4$ [16 fs]
Acceleration factor	**50**	**48**	**12***	**3.4***
Acceleration factor (no ECE)	**195**	**186**	**12***	**3.4***

*These simulations are run in the NpT ensemble, leading to more energy model evaluations in order to compute stresses. Re-using these energy evaluation for energy conservation enforcement and printing the energy to output, as opposed to recomputing them, would reduce the overhead, but it is not exploited in the present implementation.

In this appendix, we report the overall timings for the simulations that were performed in this work. When multiple FlashMD models with different time steps were used, we report the largest-stride model that affords qualitatively accurate results. The timings obtained in this way are compiled in Table 2. In comparing these results with the theoretical speed-ups given by $\Delta\tau$, one should keep several issues into considerations: (i) We did not fine tune the time step of the base MD runs, nor attempt a fine-grained grid of acceleration factors for the FlashMD models. This might skew results in either directions: for instance, water can be run stably with 0.5 fs MD, and LiPS could have probably been run stably with a 32 fs FlashMD model. (ii) We did not systematically explore the Pareto frontier of FlashMD models in its architecture setup. However, in the current version, one evaluation of FlashMD is faster than a MLIP energy evaluation due to hyperparameter differences. (iii) Some of the extensions require additional energy evaluations; for instance energy scaling or the NpT integrator require one or two energy evaluations each. There are obvious optimizations, such as only rescaling energy every few steps, which we did not consider in order to keep the analysis clearer for this first demonstration of a universal FlashMD model.

H Liquid argon benchmark (MDNet)

The early work by Zheng et al. [62] presents MDNet, a simple architecture for fitting molecular dynamics trajectories in the NVE ensemble. Here, we compare FlashMD against MDNet, as well as Equivariant Graph Neural Networks [103] (EGNN, which was also explored as an alternative in Zheng et al. [62]). The lack of code and sufficient description makes it difficult to reproduce the workflow of the authors exactly; nonetheless, we attempt a similar set-up to Zheng et al. [62]:

- All reference simulations are run using LAMMPS, using a system of 256 argon atoms and a Lennard-Jones potential. The time step for the reference simulations is 1 fs.
- Ten runs are performed (eight for training, one for validation, one for testing), initially equilibrating in the NpT ensemble for 100 ps, and then performing a production run for 10 ps in the NVE ensemble using the velocity Verlet algorithm.
- From each NVE trajectory, 25 equally spaced configurations are selected to be part of the training/validation/test set.
- A stride of 128 fs is used for training. Even though smaller strides are also explored in Zheng et al. [62], we find that this physical system is not particularly challenging due to its very simple and smooth potential energy surface, and therefore we only test predictions on the largest stride investigated in the original publication. Time-reversed targets are also added to the dataset, following Zheng et al. [62].

Using this set-up, we were able to reproduce the large-stride velocity Verlet results in Zheng et al. [62] exactly. The accuracies of velocity Verlet, EGNN, MDNet and FlashMD are shown in Table 3, where FlashMD is shown to outperform all methods. Note that our set-up involves less than one tenth of the training data used in Zheng et al. [62], and that FlashMD's accuracy is likely to be underestimated as a result.

Table 3: Accuracies of different methods for molecular dynamics trajectory predictions, on a liquid argon system with a predictive stride of 128 fs. EGNN and MDNet results are from Zheng et al. [62]. Positions errors are in units of Å, velocity errors are in units of Å/fs.

Method	Velocity Verlet	EGNN	MDNet	FlashMD
RMSE (q)	$3.9 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$	$\mathbf{5.4 \cdot 10^{-4}}$
RMSE (v)	$1.8 \cdot 10^{-3}$	$1.9 \cdot 10^{-4}$	$5.7 \cdot 10^{-5}$	$\mathbf{8.4 \cdot 10^{-6}}$

I SPC/E water benchmark (TrajCast)

TrajCast [63] published three datasets for the direct learning of molecular dynamics. Here, we compare our accuracies with the accuracies reported in Thiemann et al. [63], using the SPC/E water dataset they provide.

Table 4: Test-set accuracies of FlashMD and TrajCast [63] when trained on a SPC/E water dataset [63]. TrajCast results are reproduced from [63]. All errors are given in percentage MAE.

Architecture	TrajCast	FlashMD
Displacements	0.17	0.17
Momenta	0.37	0.22

The FlashMD results here make use of a slightly improved architecture compared to the one used to generate the results shown in this work. The same architecture was used to train the r^{2}SCAN FlashMD models we currently recommend, and it corresponds to the FlashMD implementation in `metatrain` [82]. The architecture used to produce the results in this work, for example, yields a momentum MAE of 0.52%, indicating that FlashMD and TrajCast have similar accuracies and that relatively minor tweaks can tip the numbers in favor of one or the other. In general, we always found our models to show clear signs of underfitting, showing that larger models and/or longer training times are generally beneficial when training models for the direct prediction of molecular dynamics trajectories, especially when compared to machine-learned interatomic potentials. The design of more accurate models will be a crucial challenge to achieve near-quantitative results using direct models for molecular dynamics in the future.

J Water simulations based on the q-TIP4P/f model

Due to the inaccurate description of water by PBEsol (the DFT functional used in the training of PET-MAD), we also train models on the q-TIP4P/f empirical water model [75] to investigate time-dependent properties in liquid water without raising the temperature, which in turns produces artifacts such as frequent bond dissociations that significantly affect the dynamics.

The dynamical properties we focus on are the mean square displacement (MSD) of oxygen atoms, as well as the dipole-dipole correlation function, both as a function of time. In order to avoid the large temperature deviations shown in Sec. 4 for the SVR thermostat, we instead use a fast-forward Langevin [104] thermostat, which is a modification of the Langevin thermostat aimed at reducing the effect of Langevin dynamics on dynamical properties, while being applied locally to each atom.

Fig. 11 shows the MSD and dipole-dipole correlation function for MD run with the q-TIP4P/f model, as well as FlashMD models of various strides. The results, shown in Fig. 11, establish the need for strong local thermostatting (time constant $\tau_L < 100$ fs) in order to obtain consistent statistical sampling between MD and FlashMD. Unfortunately, such thermostatting leads to an underestimation of the diffusion coefficient of water (which is proportional to the slope of the MSD curve) which is evident comparing the reference MD results for the $\tau_L = 100$ fs with the gentler $\tau_L = 1000$ fs. The tradeoff between thermostat strength and accuracy in enforcing correct sampling is similar to what was observed for explicit MD simulations based on non-conservative direct-force models [21]. This suggests that, as in that case, improving model accuracy and refining the thermostatting strategy might mitigate but not cure the underlying problems, and that one should also investigate methods to recover some of the conservation laws that are obeyed by MD trajectories, in order to achieve guaranteed quantitative accuracy in dynamics and sampling.

K Simulation details

All MD simulations in Sec. 4 were performed with `i-PI` [84], employing the internal implementation of the thermostats and barostats in the case of reference MD, and using custom, FlashMD-compatible integrators that adopt the integration schemes explained in Sec. D. In FlashMD simulations, inference was made with both the energy conservation enforcement filter and the random rotation filter, unless specified otherwise.

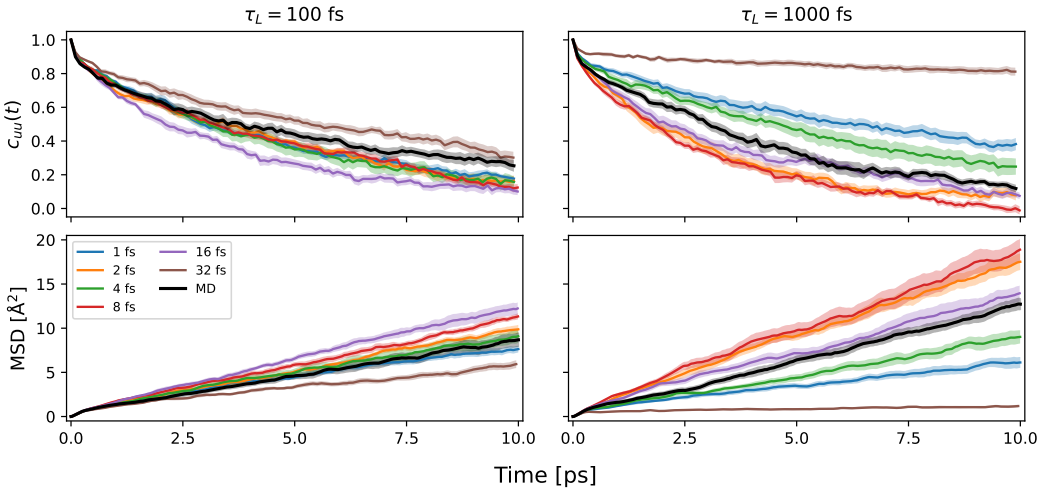

Figure 11: Dipole-dipole correlation functions (top) and mean-square displacement (bottom) for simulation of water using fast-forward Langevin thermostats with time constants $\tau_L = 100$ fs (left) and $\tau_L = 1000$ fs (right). The stronger thermostat leads to results from the different models that approach those of VV trajectories, but introduces its own spurious effects on dynamics.

Water All PET-MAD-based water-specific model runs were performed for 100 ps using a periodic box of 64 water molecules and starting from a structure equilibrated with PET-MAD in the NVT ensemble at 450 K, using a volume corresponding to the experimental density of liquid water at 300 K. NpT trajectories were started from the same structure, using a temperature of 450 K and a pressure of 1 bar. The q-TIP4P/f-based models were equilibrated and run in the NVT ensemble in the same way, but at 300 K and using the q-TIP4P/f for both equilibration and energy evaluations.

Solvated alanine dipeptide NpT simulations at 450 K and 1 bar were performed with a single dipeptide solvated by 200 molecules of water in a cubic cell of $20 \times 20 \times 20$ Å^3, as done in Morrone et al. [79]. The initial configuration of the system was randomly initialized with `packmol` [105], and the starting conformations of the dipeptide were taken from Ref. [106]. Simulations were performed for PET-MAD MLIP at 0.5 fs strides, universal FlashMD with 8 fs strides, and universal FlashMD with 16 fs strides, for a total duration of 1 ns. Langevin thermostat was coupled to the system with $\tau = 100$ fs, and an isotropic Bussi-Zykova-Parrinello (BZP) barostat [92] was used with $\tau = 400$ fs, including a Langevin thermostat coupled to the cell parameters with $\tau = 200$ fs. For each MD engine, 10 simulations were parallelly performed with different starting configurations to optimize sampling. Despite the difference in time strides, equivalent number of snapshots were sampled across the simulation setups to ensure a fair comparison of the resulting free energy surfaces.

Al(110) surface Al(110) slab configurations were generated with ASE from a $5 \times 6 \times 8$ supercell and 20 Å vacuum in z direction. Following Marzari et al. [76], NVT simulations were performed for the system for the temperature range between 400 K and 900 K, for 0.5 ns. The system was coupled to a SVR thermostat [30] with τ =10 fs. PET-MAD MLIP at 1 fs strides, universal FlashMD at 16 fs strides, and universal FlashMD at 64 fs strides were used for the simulations.

$\gamma-Li_3PS_4$ Simulations details closely follow those of the original work by Gigli et al. [80] and the starting configuration was also obtained from the reference. 3 ns NpT simulations at 0 bar were performed for temperatures between 575 and 725 K at 25 K intervals, using PET-MAD MLIP at 2 fs strides and universal FlashMD at 16 fs strides. SVR thermostat [30] was coupled to the system with $\tau = 10$ fs, and the BZP barostat [92] was used with $\tau = 1000$ fs, including a Generalized Langevin Equation [107] (GLE) thermostat coupled to the cell parameters at the same value of τ. From the resulting MD trajectories, MSD of Li was first computed and used to calculate the Li conductivity with the Nernst-Einstein equation.

L Computational resources

The use of computational resources in this work mainly stems from the generation of the universal dataset of NVE trajectories, which employed 20,000 GPU hours on an Nvidia GH200 cluster. Model training was performed on Nvidia H100 GPUs, for a total of around 3,000 GPU hours. All other experiments, mostly molecular dynamics, were run either on H100 or L40S GPUs, and they do not contribute to the overall total compute in a significant way. Overall, we estimate our total usage of computational resources as slightly under 25000 H100 GPU hours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims in the abstract are justified by our analysis and findings. The time steps generally employed in MD simulations range from 0.5 fs to 2 fs, therefore our models (which are generally tested in the 16 fs to 64 fs range) provide a longer stride by a factor between one and two orders of magnitude. Our justification for the architecture, generalization to arbitrary thermodynamic ensembles and analysis of failure modes can be found in the Theory section of this work. Our experiments and relative discussion, on which we base the last sentence of the abstract, are in the Results section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Although we do not provide a standalone Limitations section, we have been manifest about the potential shortcomings of our models, especially regarding their interaction with the global SVR thermostat (Results section) and their occasional failures in reproducing time-dependent properties (in the Appendix, using q-TIP4P/f water models).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce the results in the paper is available in the main text and in the Appendices. Based on these alone, a reader would be able to reproduce all of our results (quantitatively, in most cases), reaching the same scientific conclusions as us.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A repository is made available with all the code and instructions to reproduce all results in the paper. This is included as Supplementary Material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Our training details are fully explained in the corresponding Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All results that are affected by significant statistical error are reported together with error bounds. These include the average water temperatures and pressures in the Results section, as well as the mean square displacements and dipole autocorrelation functions discussed in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This information is reported in the corresponding Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All authors have reviewed, adhered to, and respected the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: It is difficult to estimate the societal impact of this paper, as its main result is the acceleration of *existing* modeling techniques – empowering practitioners to access longer time scales in their computational studies. To the best of our knowledge, these modeling techniques are overwhelmingly used for constructive purposes (drug discovery, modeling of new materials) and so we would expect the outcomes to have an overall positive societal impact. We have expressed this view in the paper, by highlighting that our method would allow practitioners to access longer time scales in their experiments.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the employed software has been used properly within its terms, acknowledged and credited. We have not used any other assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces a few general-purpose models for molecular dynamics prediction. These are already accompanied by excellent documentation, which will be made public along with the models.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We did not involve LLMs in any non-standard or original way.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.