Machine Learning: Science and Technology

PAPER • OPEN ACCESS

High-fidelity molecular dynamics trajectory reconstruction with bi-directional neural networks

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
6 January 2022

REVISED
24 April 2022

ACCEPTED FOR PUBLICATION
11 May 2022

PUBLISHED
30 May 2022

PAPER

High-fidelity molecular dynamics trajectory reconstruction with bi-directional neural networks

Ludwig Winkler[1], **Klaus-Robert Müller**[1,2,3,4,5,*] **and Huziel E Sauceda**[1,6,7,*]

[1] Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany
[2] BIFOLD–Berlin Institute for the Foundations of Learning and Data, Berlin, Germany
[3] Department of Artificial Intelligence, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea
[4] Max Planck Institute for Informatics, Stuhlsatzenhausweg, 66123 Saarbrücken, Germany
[5] Google Research, Brain Team, Berlin, Germany
[6] BASLEARN – TU Berlin/BASF Joint Lab for Machine Learning, Technische Universität Berlin, Berlin, 10587, Germany
[7] Departamento de Materia Condensada, Instituto de Física, Universidad Nacional Autónoma de México, Apartado Postal 20-346, 01000 México, D.F., México
[*] Authors to whom any correspondence should be addressed.

E-mail: **klaus-robert.mueller@tu-berlin.de** **and** **huziel.sauceda@fisica.unam.mx**

Keywords: super-resolution, molecular dynamics, bi-directional recurrent neural networks, trajectory learning, LSTM, machine learning, molecular systems

Abstract

Molecular dynamics (MD) simulations are a cornerstone in science, enabling the investigation of a system's thermodynamics all the way to analyzing intricate molecular interactions. In general, creating extended molecular trajectories can be a computationally expensive process, for example, when running *ab-initio* simulations. Hence, repeating such calculations to either obtain more accurate thermodynamics or to get a higher resolution in the dynamics generated by a fine-grained quantum interaction can be time- and computational resource-consuming. In this work, we explore different machine learning methodologies to increase the resolution of MD trajectories on-demand within a post-processing step. As a proof of concept, we analyse the performance of bi-directional neural networks (NNs) such as neural ODEs, Hamiltonian networks, recurrent NNs and long short-term memories, as well as the uni-directional variants as a reference, for MD simulations (here: the MD17 dataset). We have found that Bi-LSTMs are the best performing models; by utilizing the local time-symmetry of thermostated trajectories they can even learn long-range correlations and display high robustness to noisy dynamics across molecular complexity. Our models can reach accuracies of up to 10^{-4} Å in trajectory interpolation, which leads to the faithful reconstruction of several unseen high-frequency molecular vibration cycles. This renders the comparison between the learned and reference trajectories indistinguishable. The results reported in this work can serve (1) as a baseline for larger systems, as well as (2) for the construction of better MD integrators.

1. Introduction

Computational physics and chemistry and in particular molecular dynamics (MD) simulations are fundamental tools for the understanding of molecular systems [1–3]. According to the ergodic theorem, the statistical information embedded in the MD generated trajectories is equivalent to performing thermodynamical ensemble averages [1]. This is a valid statement for very long molecular trajectories, meaning that generating predictive thermodynamical observables using *ab initio* methodologies will demand large computational resources. In order to reduce the computational load, a number of simulation parameters can be tuned to considerably reduce the amount of costly electronic structure calculations. For example, it is known that performing trustworthy thermodynamical averages from MD results requires the use of non-correlated samples from the trajectory. This in turn allows dismissing intermediate steps in the simulations. Subsequently, if the MD integration time step is $\Delta\tau$, then we can store samples every n step (i.e. $\Delta t = n * \Delta\tau$. Nevertheless, by doing so, valuable dynamical information is being dismissed, for example subtle dynamical processes and fine correlations in the system.

The parameter $\Delta\tau$ in MD simulations represent a parameter that has to be chosen in a way to maximize computational efficiency while keeping the precision required to generate physically meaningful results. On one hand, the integration step $\Delta\tau$ has to be small enough to generate energy conservative dynamics and to accurately sample high frequency modes. On the other hand, it also has to be large enough to reduce unnecessary use of computational resources, which then could also lead to non-smooth trajectories and important features could be overlooked.

Beyond the well known techniques developed within the field of statistical thermodynamics to discretize and integrate Newton's equations of motion [1], in recent years, innovative machine learning methods have appeared to accelerate and improve MD simulations ranging from force fields learning [2, 4–22] to avoid expensive *ab initio* evaluations, direct free energy sampling techniques [23–26] as well as new integrator learning [27–31]. Additionally, extensive software has been developed coupling modern machine learning models with MD techniques [32–34]. Each one of those approaches contributes to a different aspect of the task of accelerating molecular simulations, notably with the learned integrator of equations of motion the least studied up to now. This last area of trajectory and integrators' learning is precisely the topic that we will address in this article.

In this regard, some early attempts to replace conventional dynamic evolution methodologies were developed first by introducing the concept of neural ordinary differential equations (NODEs) [28], and then by modifying deep neural network (NN) architectures to include physical inductive biases. For example, Greydanus *et al* [29] introduced Hamiltonian NNs (HNN) which approximate the Hamiltonian in classical mechanics with a NN. Subsequently, they compute the velocity and acceleration by taking the partial derivatives of the approximated Hamiltonian, $\dot{q} = \partial H/\partial p$ and $\dot{p} = -\partial H/\partial q$. Then following this idea, Cranmer *et al* [30] expanded on such approach by approximating the Lagrangian of the system introducing Lagrangian NNs (LNN), which relies on computing the partial derivatives corresponding to velocity and acceleration with the Lagrangian formalism. The architectural configuration of HNN's and LNN's endow the models with the principle of energy conservation. Even though these methods give good results for ideal systems, their energy conserving bias is then a limitation when describing realistic systems such as thermostated molecular simulations.

Recently, more robust methodologies have been applied to overcome some limitations of analytically biased models, for example, Tsai *et al* [23] used modern natural language processing models to predict the probability distributions of dynamical systems allowing the direct calculation of thermodynamical observables. Another important example was presented by Kadupitiya *et al* [27] where by means of uni-directional recurrent neural networks (RNNs) they manage to forecast the positions and velocities for low-dimensional systems. The methodologies above-mentioned, even though they have made considerable advances in the field of MD simulations, are still missing a robust analysis of their applicability on realistic scenarios and their interpolation accuracy.

In this article, we introduce a series of approaches to create time-super-resolution of MD trajectories using a wide variety of NN–based methods. In particular, we focus on the family of bi-directional recurrent neural networks to precisely address this issue in the formalism of learnable differential equations. Our method is used to interpolate missing trajectory snapshots, thereby increasing the resolution of the simulations in a post-processing manner. Hence, it allows enhancing on-demand the resolution of dynamical properties such as vibrational spectra, better free energy surfaces (FESs), as well as more detailed animations. Additionally, from applying our methodology to realistic molecular systems, we have found an insightful trend (which may not come as a complete surprise): the higher the temperature of the system, the easier to learn its dynamics is. This result supports the hypothesis that the Helmholtz free energy is less complex than the underlying potential energy surface (PES). In other words, as the temperature increases, the dynamics of the system becomes smoother. Which means that by combining MD simulations with our post-processing super-resolution method, the computational cost as well as simulations times could be considerably reduced.

2. Methodology

2.1. Molecular dynamics

Performing MD simulations of a molecular system in practice requires discretizing in time Newtons' equations of motion $\ddot{\mathbf{a}} = m^{-1}\mathbf{F}$. The Velocity-Verlet algorithm is a popular choice for this discretization given a PES $U = U(\mathbf{r})$ which defines the force field $\mathbf{F} = -\nabla_{\mathbf{r}} U$ acting on each atom. Hence, as a result, we obtain a trajectory with the molecular time evolution or in other words, a time series of the atomic coordinates $\mathbf{r}$ and momentum $\mathbf{p}$: $\mathcal{S} = \{\mathbf{x}_t = (\mathbf{r}_t, \mathbf{p}_t); t = i\Delta\tau, i = 0, \ldots, N_T\}$. Here $\mathbf{x}$ is known as a point in the phase space of the system. The discretization parameter $\Delta\tau$ has to be selected according to the system and the simulation conditions. In fact, it is key to choose it much smaller than the fastest oscillation period in the system. Which, due to the nature of chemical bonds, renders $\Delta\tau$ in the order of 0.1 to 1 fs for organic systems. Now, another fundamental aspect to generate meaningful simulations is the length of the time series or the total simulation time $T = N_T * \Delta\tau$, which in contrast to the selection of $\Delta\tau$, has to be much larger than the slowest oscillation in the system. According to the ergodic theorem, in order to fully recover the thermodynamical properties of the system $N_T \to \infty$, nevertheless in practice a careful selection of a finite value of these parameters gives converged thermodynamic averages. Thereby, the selection of N_T is a more abstract task because it is tightly correlated to the type of system and the physical phenomena to be studied. For example, studying light diatomic molecules could require simulation times on the order of tens of fs ($N_T \approx 10^3$), while more interesting molecules containing a couple of dozens of atoms and a fluxional molecular structure would require simulations on the order of ns ($N_T \approx 10^7$) [20, 35]. Keeping in mind that every integration step during the MD simulation requires the explicit calculation of the potential energy and forces, which in the case of electronic structure calculations such as density functional theory (DFT) can take on the order of seconds, we can see the benefit on creating a methodology that helps to reduce the value of N_T without losing accuracy. In a straightforward manner, this implies that we take larger integration steps in the simulation.

2.2. MD trajectory interpolation with NNs

In this article, we propose to employ machine learning algorithms to integrate Newton's laws of motion directly from the phase space vector representation of molecules. To that end, we train a machine learning integrator to interpolate the phase space trajectory of the molecules over a finite time horizon. In general, $\mathbf{r}$, $\mathbf{p} \in \mathbb{R}^{3N}$ where N is the number of atoms in the molecular system, but for the sake of simplicity here we will analyse the one dimensional case, $r, p \in \mathbb{R}$ and later generalize to $3N$ dimensions.

A time-dependent variable can be described by the differential equation $\dot{x}_t = f(x_t, t)$ for a general time index t, which is derived from the phase space vector, which then has the discrete solution,

$$x_{t+\Delta t} = x_t + \int_t^{t+\Delta t} f(x_{t'}, t')dt'. \tag{1}$$

Now, instead of performing a 'big' jump by Δt, we would like to interpolate in between the two subsequent phase space vectors $x_{t+\Delta t}$ and x_t to achieve a higher resolution. Hence, we choose to integrate a parameterized dynamics described by, $\dot{x}_\tau = f_\theta(x_\tau, \tau)$ which has a higher temporal resolution τ between any two subsequent states in the simulation x_t (i.e. $\tau \in [t, t+\Delta t]$). Thereby, two subsequent phase space vectors in the coarse simulation separated by a time Δt takes the form,

$$x_{t+\Delta t} = x_t + \int_t^{t+\Delta t} f_\theta(x_\tau, \tau)d\tau \tag{2}$$

where the second term is the NN integrator that propagates the dynamics. To infer the correct set of parameters θ of the integrator network, we have to optimize the scalar loss function,

$$\mathcal{L}(f_\theta(x_\tau, \tau), f(x_\tau, \tau), t, t+\Delta t) \\ = \frac{1}{\Delta t}\left\|\int_t^{t+\Delta t}\left[f_\theta(x_\tau, \tau) - f(x_\tau, \tau)\right]d\tau\right\|^2. \tag{3}$$

The loss function is evaluated on randomly sampled sections of the trajectories in mini batches.

2.2.1. Bi-directional interpolation

The coarse simulation x_t provides a trajectory at discrete time steps Δt, from which the learned intermediate dynamics $f_\theta(x_\tau, \tau)$ reconstructs the missing trajectory segment at a higher resolution (see figure 1(B)). This means obtaining x_τ, where $\tau \in \{t, t+\Delta\tau, \ldots, t+(n-1)\Delta\tau, t+\Delta t\}$ and $\Delta\tau/\Delta t$ gives the enhanced resolution. The coarse trajectory at time steps x_t and $x_{t+\Delta t}$ thus provides the initial and the final condition for the dynamics $f_\theta(x_\tau, \tau)$.

Now, for better error control we can make use of the fundamental property of time reversibility of Newtons' equations. We obtain the same trajectory if we start from the initial conditions x_0 and get to the final state x_T and if we start from x_T and propagate the system backwards in time to x_0. With the provision of the initial condition x_t and the final condition $x_{t+\Delta t}$, we can compute the forward $\overrightarrow{x}_\tau$ and backward $\overleftarrow{x}_\tau$ trajectories for $\tau \in \{t, t+\Delta t\}$ (see figure 1),

$$\overrightarrow{x}_\tau = x_t + \int_t^\tau f_\theta(x_s, s)ds \tag{4}$$

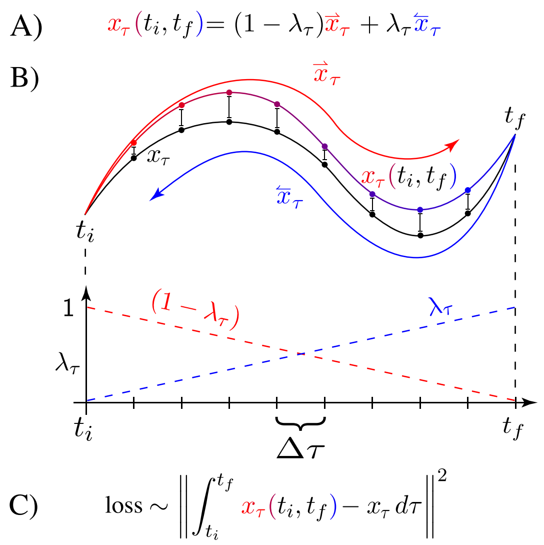
A) $x_\tau(t_i,t_f) = (1-\lambda_\tau)\overrightarrow{x}_\tau + \lambda_\tau\overleftarrow{x}_\tau$
B)
$\overrightarrow{x}_\tau$
t_f
x_τ
$x_\tau(t_i,t_f)$
$\overleftarrow{x}_\tau$
t_i
1
$(1-\lambda_\tau)$
λ_τ
λ_τ
t_i
$\Delta\tau$
t_f
C) loss $\sim \left\| \int_{t_i}^{t_f} x_\tau(t_i,t_f) - x_\tau \, d\tau \right\|^2$

Figure 1. Interpolation of the trajectory using a bi-directional NNs. (A) Linear combination of the forward $\overrightarrow{x}_\tau$ (red) and backward $\overleftarrow{x}_\tau$ (blue) solutions allows for better control of the error by using the information of both the initial and the final condition, $x(t_i)$ and $x(t_f)$, respectively. λ_τ is the mixing parameter defined by equation (6). (B) Graphical description of the interpolation process by combining the forward and backward predictors. (C) Loss function.

$$\overleftarrow{x}_\tau = x_{t+\Delta t} - \int_{t+\Delta t}^{\tau} f_\theta(x_s,s)ds. \tag{5}$$

The interpolation of the two trajectories is achieved via the interpolation parameter $\lambda_\tau \in [0,1], \tau \in [0,\Delta t]$,

$$\lambda_\tau = \frac{\int_{s=0}^{\tau} s\, ds}{\Delta t} \tag{6}$$

which is monotonically increasing for the duration $\tau \in [0,\Delta t]$ and is designed to shift the weight from the forward trajectory to the backward trajectory over the course of the interpolation. The interpolated trajectory can then be recovered via,

$$x_\tau = (1-\lambda_\tau)\overrightarrow{x}_\tau + \lambda_\tau\overleftarrow{x}_\tau. \tag{7}$$

This equation was inspired by the very insightful concept of thermodynamic integration from the FES computation's toolbox [1]. The interpolation is visualized in figure 1, which highlights the shifting interpolation parameter λ_τ.

2.2.2. NN integrators

In order to assess which NN architecture is the most suitable to interpolate MD trajectories, we considered four NN integrators: neural ODEs, HNNs, recurrent NNs and long short-term memory (LSTM) [36]. A pictorial description of the considered architectures is presented in figure 2.

2.2.2.1. Differential architectures

The most accessible approach to modelling the approximate dynamics f_θ is by directly computing the differentials with a fully-connected NN. The Euler discretization is often sufficient for convergence but more sophisticated, adaptive solvers are applicable such as Runge-Kutta and Dormand-Prince solvers for differential equations [37]. In order to be applicable to deep NNs trained with the backpropagation algorithms, these solvers require the use of the adjoint sensitivity method, which backpropagates an adjoint quantity as a surrogate gradient through time [28]. The adjoint sensitivity method backpropagates the error through adaptive solvers with a constant memory cost, which is highly suitable for adaptive solvers with potentially large number of evaluations. Once the adjoint is backpropagated to all evaluations, the gradients of the parameters θ can be obtained and gradient descent training is eligible.

Newton's equation for dynamical systems can be generalized into the Hamiltonian mechanics framework. The canonical coordinate positions r_t and momentum p_t can be obtained through the partial derivatives of the Hamiltonian $\mathcal{H}(r_t,p_t)$. HNNs [29] predict an approximate Hamiltonian $\mathcal{H}_\theta \approx \mathcal{H}$

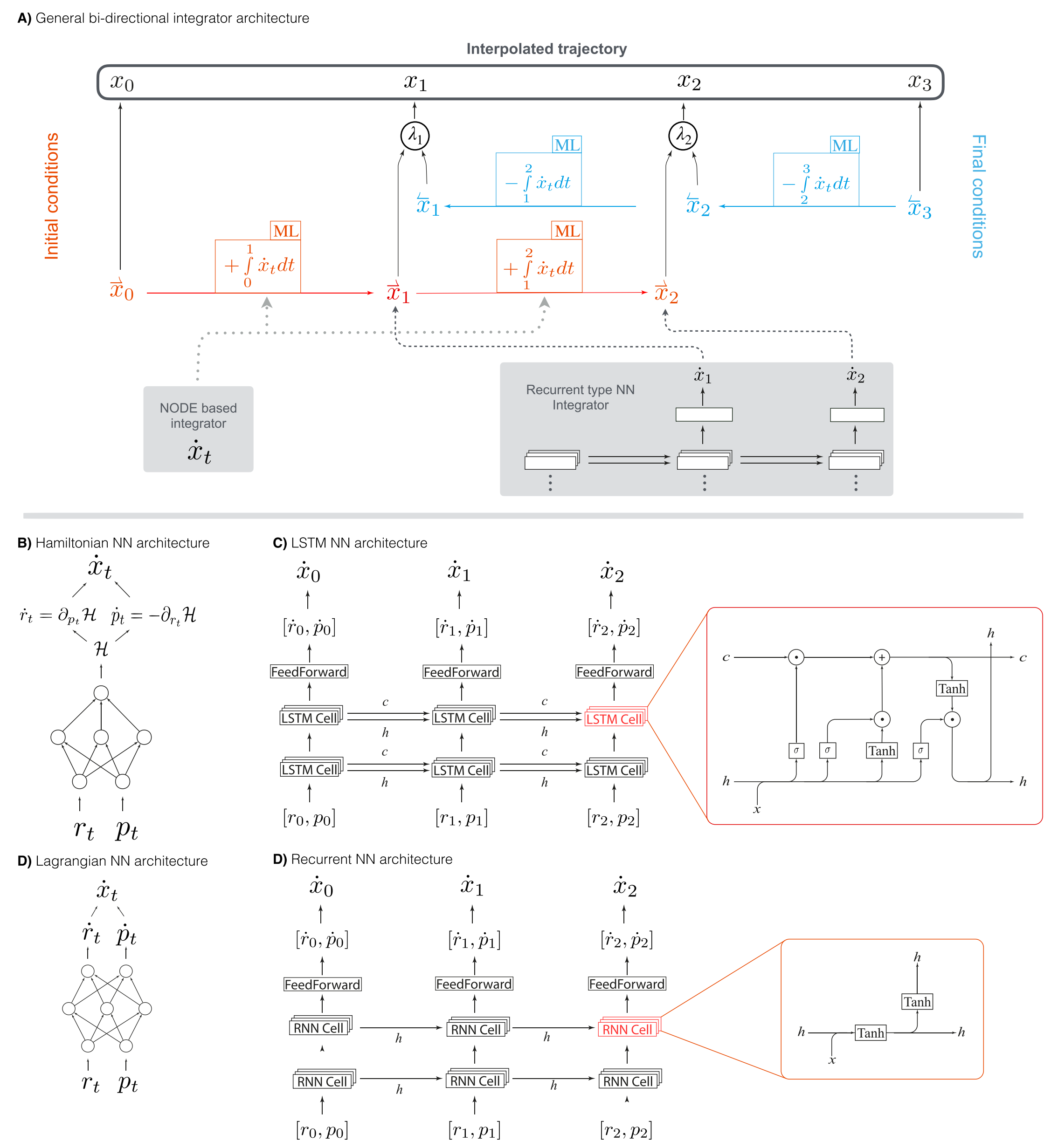
A) General bi-directional integrator architecture
Interpolated trajectory
x_0
x_1
x_2
x_3
λ_1
λ_2
Initial conditions
Final conditions
ML
$+\int_0^1 \dot{x}_t dt$
$+\int_1^2 \dot{x}_t dt$
$-\int_1^2 \dot{x}_t dt$
$-\int_2^3 \dot{x}_t dt$
$\vec{x}_0$
$\vec{x}_1$
$\vec{x}_2$
$\overleftarrow{x}_1$
$\overleftarrow{x}_2$
$\overleftarrow{x}_3$
NODE based integrator
$\dot{x}_t$
Recurrent type NN Integrator
$\dot{x}_1$
$\dot{x}_2$
B) Hamiltonian NN architecture
$\dot{x}_t$
$\dot{r}_t = \partial_{p_t}\mathcal{H}$
$\dot{p}_t = -\partial_{r_t}\mathcal{H}$
$\mathcal{H}$
r_t p_t
C) LSTM NN architecture
$\dot{x}_0$ $\dot{x}_1$ $\dot{x}_2$
$[\dot{r}_0, \dot{p}_0]$ $[\dot{r}_1, \dot{p}_1]$ $[\dot{r}_2, \dot{p}_2]$
FeedForward
LSTM Cell
c
h
$[r_0, p_0]$ $[r_1, p_1]$ $[r_2, p_2]$
Tanh
σ
x
D) Lagrangian NN architecture
$\dot{x}_t$
$\dot{r}_t$ $\dot{p}_t$
r_t p_t
D) Recurrent NN architecture
$\dot{x}_0$ $\dot{x}_1$ $\dot{x}_2$
$[\dot{r}_0, \dot{p}_0]$ $[\dot{r}_1, \dot{p}_1]$ $[\dot{r}_2, \dot{p}_2]$
FeedForward
RNN Cell
h
$[r_0, p_0]$ $[r_1, p_1]$ $[r_2, p_2]$
Tanh
x

Figure 2. (A) The information flow in a bi-directional integrator, which is trained to interpolate the differential equation that governs the MD. The same NN differential equation is utilized to predict the solution from the initial condition $[r_0, p_0]$ at $t = 0$ forward in time and from the final condition $[r_3, p_3]$ at $t = 3$ backward in time. Subsequently, the forward and backward solution is interpolated with λ_τ to obtain the interpolated values. (B) The HNN architecture which uses partial derivatives of the network output in the forward pass. (C) The information propagation of LSTM NNs and the inner dynamics of each LSTM cell consists of aggregating the input information x with the memory cell c and computing the output h. (D) NODEs use NNs in black box adaptive ODE solvers. (E) RNN model architecture showcasing how the NN integrators propagate and aggregate information through time in their hidden state h.

parameterized as a deep NN and compute the time derivatives $\dot{r}_t$ and $\dot{p}_t$ during the forward pass. In terms of functional analysis, one has to be careful in using an architecture which is differentiable at least twice, since the network is trained through backpropagation which requires a second differentiation of the model. In practice, this amounts to using continuously differentiable activation functions such as tangent hyperbolic or sigmoid and refrain from using piece-wise linear activation functions such as rectified linear units. By virtue of their construction, Hamiltonian networks exhibit energy conserving properties, such that the total energy of a dynamical system remains constant. This is of interest for energy-based systems such as MD, in which energy is shifted between potential and kinetic energy but never lost.

2.2.2.2. Recurrent architectures

RNNs extend feed-forward NNs through recurrent connections through time. They offer the ability to explicitly model time dependent relationships by incorporating the neuron activations of the previous time step. LSTM networks are recurrent architectures that resolve some important issues on general RNNs by using memory cells that can selectively read and write to them. LSTM's are widely used in modelling time series and provide a remedy for the vanishing and exploding gradient problem of classical RNN's due to excessive or miniscule eigenvalues in the recurrent weight matrices [38].

Whereas Ordinary Differential Networks and Hamiltonian Networks are considered Markovian in the sense that they only use the current state x_t of the dynamical system to predict the time derivative $\dot{x}_t$, RNNs and LSTMs are capable of modelling long-distance dependencies through their hidden states in memory cells. In the next section, we analyse in detail the performance of these two different approaches, as well as their bi-directional variants on the task of trajectory reconstruction.

2.3. NN architectures and optimization

The networks were trained to predict the change in the $3N$ position and $3N$ momentum of every atom for a total input and output dimensions of $6N$ where N is the number of atoms in the respective molecule. For all networks we employed a standardized number hidden layers and the number of neurons in the hidden layers was chosen as a integer multiple of the input dimension. For our experiments we chose five hidden layers and a multiple of 5 which resulted i.e. for a molecule of $N = 10$ atoms in $10 \times 3 \times 2 = 60$ input features and $60 \times 5 = 300$ neurons per hidden layer.

The NNs were trained on predicting entire trajectories of varying length. For each mini-batch a sub-trajectory of length Δt with n time steps was sampled and the first value was used as the initial condition for the integration of the differential equation. For bi-directional models, the final time step was used for the backward integration. The recurrent architectures such as RNN and LSTMs allowed for a trajectory as the initial respectively final condition. This gave better results but made an unfair comparison vis-a-vis the NeuralODE and HNN models which by design can only take a single time step as their respective initial condition.

The optimization was done on mini-batches of 200 samples with the default parameters of the ADAM optimizer [39]. Furthermore, a plateau learning rate scheduler was employed which halved the learning rate every epoch if the criterion had not improved every epoch by a minimum of 0.001 until a minimal learning rate of 0.00001 was reached with a patience of three epochs. The validation samples were sampled from the final 10% of the training data set by the same subtrajectory sampling as described above.

In order to speed up the training and inference speed in bi-directional models, the integration both forward and backward in time was combined into a single combined mini-batch. The NNs predicted the forward and backward integration in a single evaluation and the backward integration was appropriately prepared by reversing the momentum in time.

The experiments were done on the extended-MD17 database [8, 40]. This consists in a set of MD simulations carried out for nine molecules with different levels of fluxionality computed at the DFT level of theory using the PBE functional together with the Tkatchenko-Scheffler dispersion correction scheme [41, 42]. These molecules are shown in figure 3. This database is available at the quantum-machine website. The length of the trajectories range from 100 000 time steps to almost 1 000 000 time steps with a step size of one femtosecond. The position and momentum was normalized by subtracting the centre of the molecule at each time step and rescaling the momenta such that they were standard normally distributed. The original characteristics was easily recovered by reversing the normalization.

3. Results and discussion

3.1. Training and validation of the model

The general goal of this study is to establish the applicability of the presented methodologies on realistic dynamics from physico-chemical simulations. To this end, we have selected the well established extended-MD17 dataset [8, 35] which contains middle-sized molecules with various dynamical complexities. In this section, we analyse the performance of the bi-directional approaches (i.e. LSTM, RNN, HNN, and NODE) compared to their uni-directional counterparts while considering different resolution for the interpolation tasks. These results are shown in figure 3.

3.1.1. Uni- vs bi-directional NNs

Common methods for sequence and trajectory learning are usually deployed using the natural arrow of time, nevertheless it has been demonstrated in other areas of machine learning, such as language modelling [43–45], that bi-directional learning substantially improve the results. Here, we exploit the regularity of the physical trajectories and local time reversal symmetry by using bi-directional versions of the architectures shown in figure 2. In order to directly show the benefit of this approach, in figure 3 we compare the performance of the two approaches for all the molecules from the extended-MD17 dataset [8]. From this

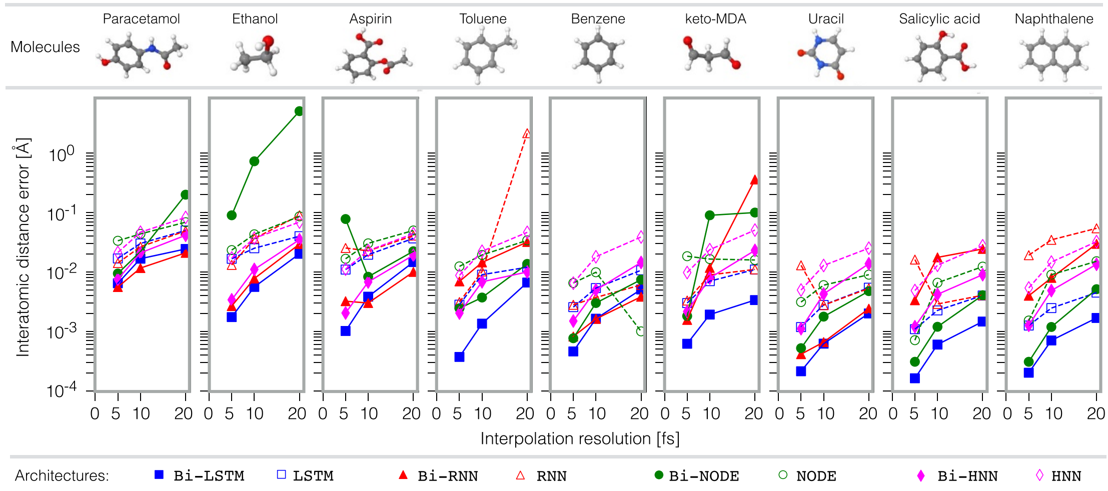

Figure 3. Performance comparison of uni- and bi-directional models over multiple integration lengths Δt on the extended-MD17 dataset [8, 35]. Uni-directional and bi-directional methods are represented by empty and full symbols, respectively. The considered interpolation resolutions are 5, 10, and 20 in femtoseconds.

figure, we can quickly see that using bi-directional NNs can boost the accuracy of the model by up to two orders of magnitude. This reveals that including both directions of the arrow of time goes beyond simply duplicating the amount of data. Instead, this strongly encodes the trajectory's regularity in the model, which results in a more accurate description without increasing the number of parameters in the architecture. Additionally, we can see that this phenomenon is not architecture specific, since all the tested NNs obtained a similar increase in accuracy and similar learning curves when considering bi-directionality. Additionally, such behaviour is also independent of the interpolation resolution.

3.1.2. Bi-LSTM performance

Now, comparing the different bi-directional methods, we can see that RNNs (i.e. RNN and LSTM) always perform better than neural ODE based approaches. This is due to the fact that recurrent based NNs are more robust against noise, as will be the case when dealing with thermostated MD simulations. Furthermore, figure 3 shows that the Bi-LSTM architecture (see figure 2) gives more consistent results across molecular structures with different complexities compared to the rest of the architectures, reaching interpolation errors of up to one order of magnitude lower. This advantage of Bi-LSTMs over other architectures could be related to the fact that LSTM cells propagate information more efficiently and keep track of the many subtleties in the molecular trajectories for longer periods of time. It is worth to remark that the accuracies reached by Bi-LSTMs in real space (i.e. coordinates) range on the order of 10^{-3} Å, which in practical terms is indistinguishable from the reference trajectory. Given these results, from now on, we will focus our analysis only on Bi-LSTM networks.

3.1.3. Molecular complexity and degrees of freedom

As an additional observation, we have noticed that molecules with higher complexity (e.g. larger number of rotors and/or higher fluxionality) generate more intricate trajectories that render the reconstruction process more challenging. This is only evident for the cases of paracetamol, ethanol and aspirin, nevertheless, there are cases where the correlation between interpolation accuracy and molecular morphology is non-intuitive. For example, even though the keto-MDA molecule (see figure 3) presents a complex PES with two rotors as main degrees of freedom [46], the reconstruction precision of its trajectory is similar to benzene and uracil, molecules that have no rotors and are formed only by an aromatic ring. Furthermore, the best accuracy reached by the Bi-LSTM model was for salicylic acid, a molecule that has two rotors coupled by a complex hydrogen bond where the proton is dynamically being transferred between the two functional groups.

The physical interpretation of these results is that the reconstruction process occurs on the FES of the molecular system, which means that the molecule is moving on its FES instead of the PES. Hence, moving on the molecular FES at a given temperature can result on non-trivial behaviour originated from the entropic contributions. Thereby, a molecular system containing complex interaction such as salicylic acid which has a H–bond can generate a smoother dynamics compared to uracil or naphthalene (see figure 3 for molecular structure reference).

The insight behind these results is that, for the considered molecules, thermal fluctuations considerably reduce the dynamical complexity of molecular systems with high fluxionality. Thereby, simplifying the learning process.

3.2. Achieving super-resolution in MD trajectories

In the previous section, we have demonstrated that Bi-LSTMs are a suitable architecture for trajectory learning of realistic molecular simulations. In this section, we continue analysing what it implies to reconstruct molecular trajectories, but now from the vibrational point of view, first from the normal frequencies framework and then from the real dynamics perspective.

One of the key aspects in trajectory interpolation is the time resolution that can be achieved by the model. Given that the MD17 dataset has integration steps of 1 fs, here the task was to skip a number of frames n for each molecule and then reconstruct the missing fragments on the trajectory. Hence, for this dataset, $\Delta\tau = 1$ fs and $\Delta t = n$ fs. In this regard, figure 3 shows the performance of all the models for $\Delta t = 5$, 10 and 20 fs. Let's analyse the results for each value of Δt in the context of its physical implications for the vibrational normal modes (i.e. harmonic analysis). As a reference, the fastest atomic oscillation periods in paracetamol and in benzene are $\approx$9.3 and $\approx$11 fs, respectively [32]. Such vibrations are mainly due to fast oscillations of the hydrogen atoms in the molecule. Hence, sampling a trajectory every 5 fs in the context of molecular vibrations means that we are skipping half the period of the fastest oscillation in the molecule, which implies that, at most, the model has to interpolate half of the oscillation cycle. This fact is reflected on the validations errors shown in figure 3. More challenging cases are sampling molecular trajectories every 10 and 20 fs, given that in these two cases the model has to reconstruct at least one full period (in the case of Δt =10 fs). Furthermore, the most interesting resolution to analyse in more detail is Δt =20 fs, given that in such case the model has to reconstruct at least one full cycle of all the molecular oscillations with frequencies larger than $\approx$1600 cm^{-1}. Which in the case of the benzene molecule, there are eight out of 30 normal modes with frequency values larger than such value (i.e. ~27% of the normal modes). More interestingly, six of them have oscillation periods of $\approx$10 fs, meaning that the Δt =20 fs model has to reconstruct two full oscillations periods for the six vibrational modes using as inputs only the initial and final states (i.e. $(\mathbf{r}_t, \mathbf{p}_t)$ and $(\mathbf{r}_{t+21}, \mathbf{p}_{t+21})$). In the supporting information, we have animated such example for the toluene molecule case, where the true dynamics is represented by blue atoms and the interpolated dynamics appears in red. As expected from figure 3, the dynamics is practically indistinguishable.

In order to visualize this analysis, in figure 4 we present the reconstructed trajectory for the benzene molecule using the Bi-LSTM architecture for Δt =20 fs, and we show its dynamics in terms of its main internal degrees of freedom: interatomic distances, angles and dihedral angles. In this figure, the blue curve is the ground truth and the red dashed line represents the predicted dynamics. A quick glance over the plot shows that, even though the Δt =20 fs case in figure 3 is the one that gives the largest error, *such accuracy still corresponds to an excellent agreement between the reference trajectory and the ML prediction.*

As mentioned before, the fastest oscillation in a molecule are due to hydrogen atoms oscillations, which in the benzene case it can be tracked by plotting the interatomic distance $d_{\text{C-H}}$ (figure 4 top panel). Here, the measured oscillation period from the signal is $\approx$11 fs, slightly larger than the normal mode value. The origin of this red shift is the fact that at finite temperatures the system visits the anharmonic region of the PES, which then generate slower oscillations. As a reference, a green rectangular window is used in figure 4 to show examples of interpolated intervals of trajectories. Hence, in the case of $d_{\text{C-H}}$ we can clearly see that the method is successfully reconstructing two full cycles of the variable, rendering indistinguishable the comparison to the reference trajectory. Another important internal variable is the first neighbour carbon-carbon distance $d_{\text{C-C}}$ (second panel in figure 4). $d_{\text{C-C}}$ oscillation period is in general considerably slower, but because such interatomic distance is part of many anharmonically-coupled vibrational modes, it can create apparent high-frequency diatomic oscillations (as short as $\approx$9 fs). Here, despite the complex dynamics of this internal variable, the model manages to faithfully recover its behaviour. Now, in order to incorporate a highly non-linear and weakly correlated interatomic distance, we considered two opposite hydrogen atoms, $d_{\text{H-H}}$ (third panel in figure 4). In principle, this variable should amplify small errors in the reconstructed trajectory, nevertheless the results are still in excellent agreement.

Another important aspect in molecular fluctuations is the analysis of internal shearing deformations as well as out of plane deformations, which represent a global mechanical property of the system. These mechanical deformations can be analysed by measuring shearing angles and dihedral angles, as shown in the lower half of figure 4. The oscillations in these variables are considerably slower compared to interatomic distance fluctuations, but they can contain highly anharmonic contributions, making them a good measure for reconstruction accuracy. Again, here the interpolation accuracy of the model is very high, which makes the predicted trajectories indistinguishable from the reference curve. In fact, the MAE reconstruction

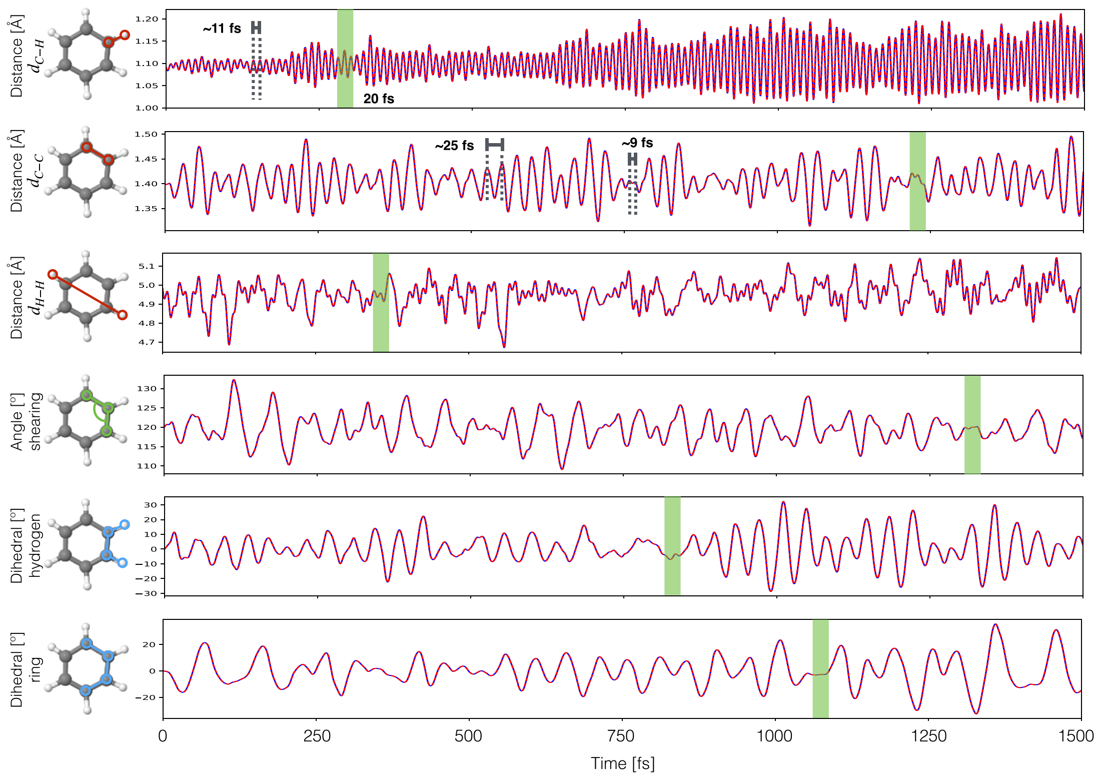

Figure 4. Analysis of the benzene dynamics and its interpolation for the 20 fs interpolation case. The six internal variables displayed are representative feature of the molecule, including interatomic distances (in Å), internal angles (in degrees), and dihedral angles (in degrees) (represented in red, green and blue in the left-most molecular structure, respectively). The true dynamics is displayed by the blue curve and the predicted results are given by the dashed red lines. In grey dotted lines we show some relevant periods for some variables, and in a green window we represent the 20 fs interpolation interval.

accuracy of the Bi-LSTM model for the benzene molecule's trajectory is $\sim 10^{-4}$ Å, a value that is close to the accuracy of the actual numerical integrators such as the Verlet algorithm.

The results obtained in this section show that the methods presented here, and in particular Bi-LSTMs, give excellent trajectories' reconstruction accuracies, rendering their predictions indistinguishable from full *ab-initio* MD results. Hence, this allows to confidently use these techniques in a diverse range of applications such as data augmentation, super-resolution generation, or even storage capacity reduction. In the next section, we use Bi-LSTMs to extract some physical insights from learning trajectories at different temperatures, where we demonstrate that we can faithfully recover its molecular FES.

3.3. Physical insights from the reconstruction process: application to keto-MDA as case of study

The keto-MDA molecule is a challenging molecule given that it presets a wide variety of interactions that generate a PES with strong electrostatic interactions but also regions with particle-in-a-box behaviour [40, 46]. Hence, such intricate energy landscape generates interesting dynamics from which, for example, force field reconstruction is not a straightforward task and requires the use of sophisticated ML methodologies [6, 9, 19, 47]. Interestingly, there is evidence that the force field learning process gets easier when the temperature of the generated training data gets lower [46], given that, in general, they explore smaller regions of the configurational space.

From the statistical point of view, the MD trajectories at a given temperature sample the configuration space in such a way that the molecular FES can be estimated by $\sim \ln P(\theta_1,\theta_2)$ where $P(\theta_1,\theta_2)$ is the integrated configurational probability density, and θ_1 and θ_2 are the main degrees of freedom of the system (see figure 7). In general, the FES is known to get smoother as we increase the temperature due to entropic effects. This is represented in figure 6, where we can see that as the temperature increases, the curved regions of the FES get smoother until they progressively flatten. This also means that the molecular trajectory in phase space becomes more stable, and hence it should be easier to reconstruct. In order to get some insights about this effect as well as to further validate the performance of our models, we have run three simulations of the keto-MDA molecule at different temperatures (100, 300 and 500 K) using a pretrained sGDML model [32] and analysed the generated trajectories. In table 1, we summarize the results for the learning procedure for the three temperatures.

Table 1. Experimental results for evaluating the influence of varying training integration time T_{tr} of the Bi-LSTM architecture on (A) inter-atomic distances and (B) velocity field. This was done for the keto-MDA molecule as a case of study for simulations run at three different temperatures. The interatomic distances and velocity field were normalized to follow the first two moments of a standard normal distribution.

(A) Interatomic distances $[10^{-3}$ Å]			
Temperature [K]	$T_{\mathrm{tr}}=5$	$T_{\mathrm{tr}}=10$	$T_{\mathrm{tr}}=20$
100	0.259	0.930	3.287
300	0.417	1.500	5.321
500	0.501	1.801	6.399

(B) Velocity field $[10^{-3}$ Å fs^{-1}]			
Temperature [K]	$T_{\mathrm{tr}}=5$	$T_{\mathrm{tr}}=10$	$T_{\mathrm{tr}}=20$
100	11.7100	38.390	132.700
300	11.320	36.820	127.210
500	10.380	33.070	113.60

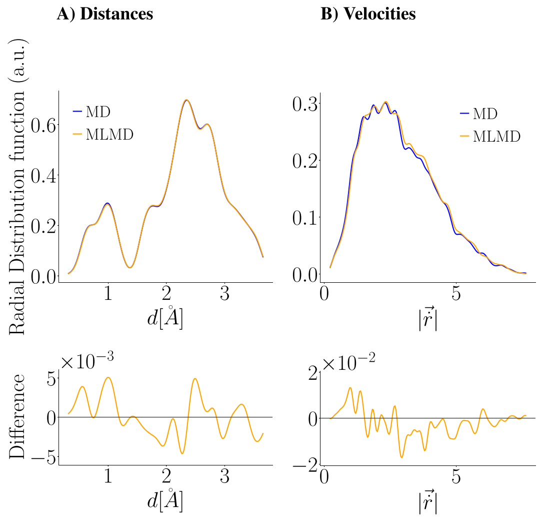

Figure 5. Comparison of between the interpolated (yellow) and ground truth (blue) trajectories for keto-MDA molecule. (A) Interatomic distance histogram and (B) speed histogram. The interpolation was done with a Bi-LSTM integrator with $\Delta t = 10$. The bottom panels provide the difference between the two curves.

3.3.1. Phase-space histograms

The frameworks presented in this study work on phase space, (r, p), meaning that during the reconstruction task the trajectory and the velocity field are recovered. The velocity field refers to the atomic velocities. Hence, here, we analyse Bi-LSTM's predictive power in terms of physically meaningful distributions from MD simulations by examining the speed and interatomic distances distributions. In previous sections, we have focused on spacial accuracy for trajectory reconstruction (up to $\sim 10^{-4}$ Å), but another important aspect is to recover an accurate velocity field. In table 1, we report the interpolation accuracy for both of these variables, displaying an excellent agreement with the reference data. In particular, in figure 5 we show the explicit comparison of the reconstructed probability distributions for the interatomic pair distance distribution function, $h(r)$, and the speed distribution function for the case of keto-MDA's trajectory at 300 K. From figures 5 and 7 and from table 1, we can see that the acquired velocity field precision is maintained for the three different temperatures.

3.3.2. Temperature dependent learning

It is worth noticing that the temperature has indeed an effect during the learning process. From the dynamical point of view, we know that the FES depends on the temperature, meaning that the trajectory generated will follow different patterns and statistically sample differently the phase-space. A pictorial representation of this effect is shown in figure 6. Figure 7 shows this effect for the keto-MDA molecule, where the left panels show the sampling for the different temperatures and the right panels show the pair distance distribution.

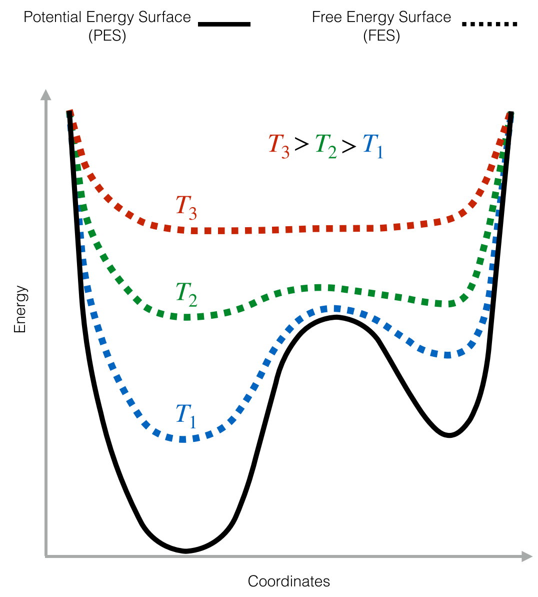

Figure 6. Pictorial representation of the effect of the temperature on the FES (dotted lines) and its underlying PES (black continuous line). Here, the FES is shown at three different temperatures ($T_1 < T_2 < T_3$) to show how increasing the temperature generates smoother surfaces.

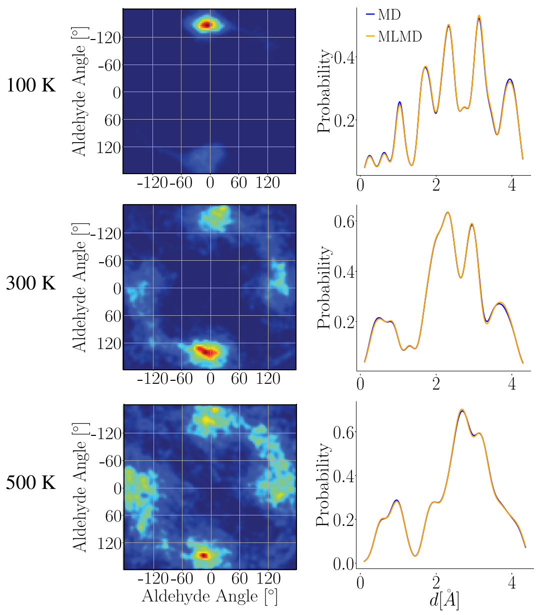

Figure 7. The distribution of interatomic distances d[Å] of keto-MDA at 100, 300 and 500 K. The predicted distribution of interatomic distances is shown in red and the target distribution is shown in blue. The distributions become less multi-modal as the temperature increases.

From here, we can see how the pair distance distribution function evolves from a multimodal histogram at 100 K to a less complex function as the temperature increases. In terms of trajectory reconstruction, from table 1, we can quickly see that when increasing the temperature of the reference data, the accuracy in the interatomic distances marginally decreases but the accuracy of the velocity field reconstruction considerably increases. Furthermore, in the case of the Bi-LSTM model with $\Delta t = 20$, the prediction accuracy increases roughly by a factor of 4 when training on data generated at 500 K relative to the 100 K case.

In a broader picture of the physical problem, these results tell us that the generated dynamics at higher temperatures are smoother even though the atomic speeds are higher. This is actually because the molecular system spends more time in anharmonic regions, which then generates the well-known frequency red shift of most of the vibrational frequencies [35]. In other words, increasing the system's temperature reduces the

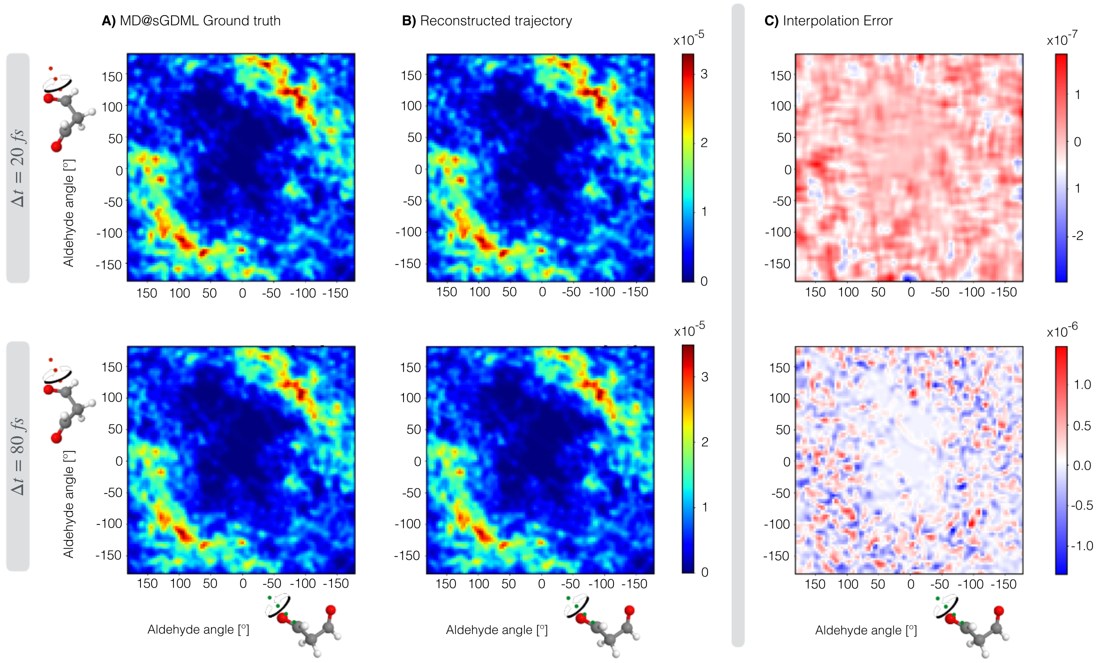

Figure 8. Sampling generated from MD using (A) the sGDML force field trained on DFT (MD17 database [8]) for the keto-MDA, (B) and its interpolation generated using bi-directional LSTM models for $\Delta t = 20$ fs and $\Delta t = 80$ fs shown in the upper and bottom rows, respectively. (C) Difference between the reference and the interpolated trajectories.

oscillation periods and generates larger oscillation amplitudes, thereby reducing the complexity of the learning problem.

3.3.3. FES reconstruction

To complement the results from the previous sections, here we correlate those results to the FES. The FES can be estimated by integrating the probability distribution generated by the molecular trajectories, hence smoother trajectories in principle should lead to less complex FES as shown in figure 7. Now, this actually has a beneficial connection to the learning problem, because given the previous analysis, we can accurately reconstruct the molecular FES from a reduced amount of data or sparse trajectories. In order to corroborate such accuracy, in figure 8 we show a comparison between the reference and Bi-LSTM-predicted configurations' sampling for the keto-MDA molecule for its main degrees of freedom i.e. the two aldehyde dihedral angles. As mentioned before, the accuracy in trajectory reconstruction is $\sim 10^{-4}$ Å, hence renders the differences between the two FES barely noticeable.

3.3.4. Testing the limits of the methodology: larger interpolation intervals

In order to assess the accuracy of the Bi-LSTM models as a function of the sampling frequency, in figure 9 we show their different models with an interpolation capacity of 5, 20 and 80 fs, where the resolution of the training data is 1 fs. This means that during training, the 80 fs model for example, was trained using time series sampled every 80 fs, given the initial and final conditions, the model will predict the intermediate 78 points in the series. If we analyse the information encoded in a time series sampled every 80 fs, we can clearly see that the training data set only contain information regarding molecular oscillations with frequencies of $\sim$500 cm^{-1} at most, and the rest of the spectrum is not explicitly embedded in the data. The frequency spectrum shown to each model during the training process is highlighted in red in figure 9. From this figure we obtain that in the three cases, the reconstructed vibrational spectrum is indistinguishable from the ground truth calculations (see bottom plot in each of the panels), despite the wide variability of the interpolation parameter. Nevertheless, in the 80 fs case (inset in figure 9(C)), we can appreciate that some slight deviations start to appear.

These results immediately bring up the question of how the model can still reconstruct very high frequencies when these are not explicitly given. The answer lies on the memory cells of the LSTM architectures. It does not matter that high frequencies are not shown to the model, since in each pass of the training process, the model is getting different configurations that sample the high frequency normal modes. Hence, by keeping in memory all that information, the model can infer the existence of all the vibrations in the system.

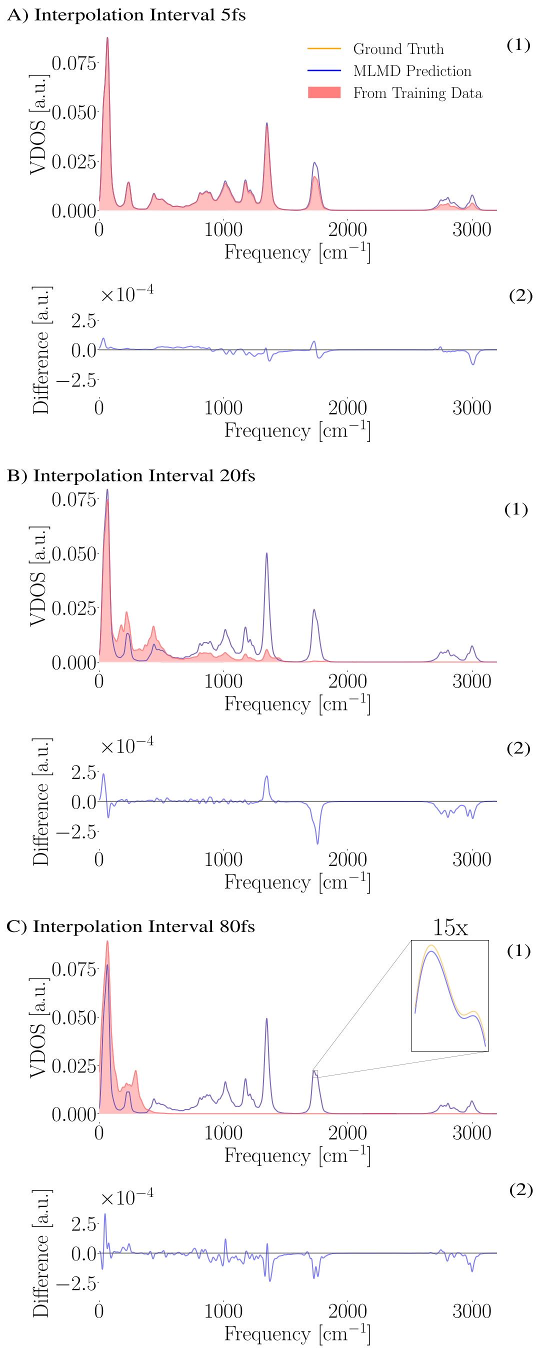

Figure 9. Reconstruction of the vibrational density of states (VDOS) using the interpolation intervals: (A) 5 fs, (B) 20 fs, and (C) 80 fs. In panel (1) are the ground truth VDOS (yellow), the predicted VDOS (blue), and the VDOS obtained from the training data (shaded red). In Panel (2) is the differences between the ground truth and predicted VDOSs.

From the results in this section and the previous one, we obtain that bi-LSTM models not only manage to accurately reconstruct the spacial components of the trajectories, but also the velocity field. Thereby, ensuring their application to reconstruct the phase-apace dynamics as well as the physical properties such as free energies and vibrational spectra.

4. Conclusions

In this work we have introduced a series of bi-directional NN integrators (i.e. NODE, HNN, RNN and LSTM) to increase the resolution of MD trajectories for a variety of molecular systems. An extensive validation process of these models on the MD17 dataset and different interpolation resolutions were presented (see figure 3). Here, we found that the Bi-LSTM model is the better performing method generating more stable results across all the extended-MD17 dataset and better accuracies, reaching errors as low as

10^{-4} Å. These errors render the interpolated and ground truth trajectories indistinguishable. The overall higher performance of Bi-LSTM networks is due to the robustness against noisy (thermostated) reference data, as well as their capacity to retain long time correlations' during the learning process.

Beyond the machine learning task of high dimensional trajectory interpolation, by varying the temperature of the reference data used for training the models, we obtained important physical insights regarding the dynamical behaviour of molecular systems. There is evidence in the literature that learning molecular force fields (i.e. the underlying PES) gets more complicated as the temperature of the training data increases. Contrasting this behaviour, here we have observed that trajectory interpolation becomes easier as the temperature increases. The origin of such different temperature-dependent learning behaviour is the fact that the trajectories generated by MD simulations effectively move on the Helmholtz FES due to entropic contributions. Additionaly, going from low to high temperatures progressively pushes the FES' landscape to deviate from the underlying PES, reducing its complexity as the temperature increases. From these results we can summarize that the temperature of MD databases have opposite effects on learning the potential and FESs. Additionally, we found that Bi-LSTMs have the remarkable property of being able to learn the full spectrum of molecular vibrational frequencies (harmonic and anharmonic) even though this information is not explicitly shown during the training process. This capability is due to their memory cells.

From the results obtained in this work, we have evinced the great learning capacity of Bi-LSTMs on the reconstruction of realistic high dimensional molecular behaviour. This opens up a new set of applications for the family of RNNs on post-processing MD results. Furthermore, the results here presented can be used to set the stage for robust extrapolation techniques, either by using it as a data enhancement method for forecasting or as the basis for numerical propagator.

Data availability statement

The training data and code base used in this study are available upon reasonable request from the authors.

Acknowledgments

K R M was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University), and was partly supported by the German Ministry for Education and Research (BMBF) under Grants 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18025A and 01IS18037A; the German Research Foundation (DFG) under Grant Math+, EXC 2046/1, Project ID 390685689. L W would like to thank Jason Salomon Rinnert for software support. H E S works at the BASLEARN—TU Berlin/BASF Joint Lab for Machine Learning, co-financed by TU Berlin and BASF SE. H E S thanks the support from DGTIC-UNAM under Project LANCAD-UNAM-DGTIC-419. Correspondence should be addressed to H E S and K R M.

ORCID iDs

Ludwig Winkler https://orcid.org/0000-0002-1354-4715
Klaus-Robert Müller https://orcid.org/0000-0002-3861-7685
Huziel E Sauceda https://orcid.org/0000-0001-6091-3408

References

[1] Tuckerman M 2010 *Statistical Mechanics: Theory and Molecular Simulation* (Oxford: Oxford University Press)
[2] Noé F, Tkatchenko A, Müller K-R and Clementi C 2020 *Annu. Rev. Phys. Chem.* **71** 361
[3] Sadeghi M and Noé F 2021 *J. Phys. Chem. Lett.* **12** 10497
[4] Behler J 2017 *Angew. Chem., Int. Ed.* **56** 12828
[5] Schütt K T, Kindermans P-J, Sauceda H E, Chmiela S, Tkatchenko A and Müller K-R 2017 *Advances in Neural Information Processing Systems 30* pp 991–1001
[6] Schütt K T, Sauceda H E, Kindermans P-J, Tkatchenko A and Müller K-R 2018 *J. Chem. Phys.* **148** 241722
[7] Han J, Zhang L, Car R and Weinan E 2018 *Commun. Comput. Phys.* **23** 629
[8] Chmiela S, Tkatchenko A, Sauceda H E, Poltavsky I, Schütt K T and Müller K-R 2017 *Sci. Adv.* **3** e1603015
[9] Chmiela S, Sauceda H E, Müller K-R and Tkatchenko A 2018 *Nat. Commun.* **9** 3887
[10] Novikov I S, Gubaev K, Podryabinkin E V and Shapeev A V 2021 *Mach. Learn.: Sci. Technol.* **2** 025002
[11] Bartók A P, Payne M C, Kondor R and Csányi G 2010 *Phys. Rev. Lett.* **104** 136403
[12] Zeni C, Rossi K, Glielmo A, Fekete Á, Gaston N, Baletto F and De Vita A 2018 *J. Chem. Phys.* **148** 241739
[13] Glielmo A, Zeni C and De Vita A 2018 *Phys. Rev.* B **97** 184307
[14] Unke O T, Chmiela S, Sauceda H E, Gastegger M, Poltavsky I, Schütt K T, Tkatchenko A and Müller K-R 2021 *Chem. Rev.* **121** 10142
[15] von Lilienfeld O A, Müller K-R and Tkatchenko A 2020 *Nat. Rev. Chem.* **4** 347

[16] Tkatchenko A 2020 *Nat. Commun.* **11** 4125
[17] Musil F, Grisafi A, Bartók A P, Ortner C, Csányi G and Ceriotti M 2021 *Chem. Rev.* **121** 9759–815
[18] Keith J A, Vassilev-Galindo V, Cheng B, Chmiela S, Gastegger M, Müller K-R and Tkatchenko A 2021 *Chem. Rev.* **121** 9816–72
[19] Unke O T, Chmiela S, Gastegger M, Schütt K T, Sauceda H E and Müller K-R 2021 *Nat. Commun.* **12** 7273
[20] Sauceda H E, Gálvez-González L E, Chmiela S, Paz-Borbón L O, Müller K-R and Tkatchenko A 2021 arXiv:2106.04229
[21] Li Z, Meidani K, Yadav P and Farimani A B 2021 Graph neural networks accelerated molecular dynamics (arXiv:2112.03383 [cs.LG])
[22] Sauceda H E, Vassilev-Galindo V, Chmiela S, Müller K-R and Tkatchenko A 2021 *Nat. Commun.* **12** 442
[23] Tsai S-T, Kuo E-J and Tiwary P 2020 *Nat. Commun.* **11** 5115
[24] Thaler S and Zavadlav J 2021 arXiv:2106.01138
[25] Vlachas P R, Arampatzis G, Uhler C and Koumoutsakos P 2021 Multiscale simulations of complex systems by learning their effective dynamics (arXiv:2006.13431 [physics.comp-ph])
[26] Vlachas P R, Zavadlav J, Praprotnik M and Koumoutsakos P 2021 Accelerated simulations of molecular systems through learning of their effective dynamics (arXiv:2102.08810 [physics.comp-ph])
[27] Kadupitiya J, Fox G C and Jadhao V 2020 arXiv:2004.06493
[28] Chen R T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 *Advances in Neural Information Processing Systems* 31
[29] Greydanus S, Dzamba M and Yosinski J 2019 *Advances in Neural Information Processing Systems* pp 15379–89
[30] Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D and Ho S 2020 *Advances in Int. Conf. on Learning Representations*
[31] Ullah A and Dral P O 2021 *New J. Phys.* **23** 113019
[32] Chmiela S, Sauceda H E, Poltavsky I, Müller K-R and Tkatchenko A 2019 *Comput. Phys. Commun.* **240** 38
[33] Doerr S, Majewski M, Pérez A, Krämer A, Clementi C, Noe F, Giorgino T and De Fabritiis G 2021 *J. Chem. Theory Comput.* **17** 2355–63
[34] Schütt K T, Kessel P, Gastegger M, Nicoli K A, Tkatchenko A and Müller K-R 2019 *J. Chem. Theory Comput.* **15** 448
[35] Sauceda H E, Chmiela S, Poltavsky I, Müller K-R and Tkatchenko A 2019 *J. Chem. Phys.* **150** 114102
[36] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735
[37] Dormand J R and Prince P J 1980 *J. Comput. Appl. Math.* **6** 19
[38] Pascanu R, Mikolov T and Bengio Y 2013 *Int. Conf. on Machine Learning* (PMLR) pp 1310–8
[39] Kingma D P and Ba J 2014 arXiv:1412.6980
[40] Sauceda H E, Gastegger M, Chmiela S, Müller K-R and Tkatchenko A 2020 *J. Chem. Phys.* **153** 124109
[41] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[42] Tkatchenko A and Scheffler M 2009 *Phys. Rev. Lett.* **102** 073005
[43] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 *Advances in Neural Information Processing Systems* pp 5998–6008
[44] Bahdanau D Cho K and Bengio Y 2014 arXiv:1409.0473
[45] Wu Y *et al* 2016 arXiv:1609.08144
[46] Sauceda H E, Chmiela S, Poltavsky I, Müller K-R and Tkatchenko A 2020 *Construction of Machine Learned Force Fields with Quantum Chemical Accuracy: Applications and Chemical Insights* (Cham: Springer) pp 277–307
[47] Schütt K, Unke O and Gastegger M 2021 *Int. Conf. on Machine Learning* (PMLR) pp 9377–88