



Cite this: *Chem. Sci.*, 2025, **16**, 17542

All publication charges for this article have been paid for by the Royal Society of Chemistry

Received 25th July 2025
Accepted 29th August 2025

DOI: 10.1039/d5sc05579b
rsc.li/chemical-science

1 Introduction

A deeper understanding of molecular excited states holds profound significance in organic chemistry, chemical biology, and materials science. Understanding the role of the properties in the Franck–Condon region and the subsequent photoinduced processes and reactions is needed to design molecules with nature-inspired functions (e.g., photosynthesis and

^aComputer-Chemistry-Center, Friedrich-Alexander-Universität Erlangen-Nürnberg, Nägelsbachstraße 25, 91052 Erlangen, Germany

^bInstitute of Theoretical Chemistry, Faculty of Chemistry, University of Vienna, Währinger Straße 17, 1090 Wien, Austria

^cDepartment of Physical Chemistry, University of Chemistry and Technology, Technická 5, 162 28 Prague, Czech Republic

^dResearch Platform on Accelerating Photoreaction Discovery (ViRAPID), University of Vienna, Währinger Straße 17, 1090 Vienna, Austria

^eVienna Doctoral School in Chemistry (DoSChem), University of Vienna, Währinger Straße 42, 1090 Vienna, Austria

^fDepartment of Chemistry, University College London, 20 Gordon Street, London, WC1H 0AJ, UK

^gHoffmann Institute of Advanced Materials, Shenzhen Polytechnic University, 7098 Liuxian Boulevard, Shenzhen, Guangdong 518055, P. R. China

^hAix Marseille University, CNRS, ICR, Marseille, France

ⁱDepartment of Chemistry & Chemical Biology, Northeastern University, 805 Columbus Avenue, Boston, MA, 02120, USA. E-mail: s.lopez@northeastern.edu

^jWilhelm-Ostwald Institute, Leipzig University, Linnéstraße 2, 04103 Leipzig, Germany. E-mail: julia.westermayr@uni-leipzig.de

^kCenter for Scalable Data Analytics and Artificial Intelligence (ScalDS.AI), Dresden, Leipzig, Germany

Machine learning for nonadiabatic molecular dynamics: best practices and recent progress

Carolin Müller, Štěpán Sršeň, Brigitta Bachmair, Rachel Crespo-Otero, Jingbai Li, Sascha Mausenberger, Max Pinheiro, Jr, Graham Worth, Steven A. Lopez * and Julia Westermayr*^{jk}

Exploring molecular excited states holds immense significance across organic chemistry, chemical biology, and materials science. Understanding the photophysical properties of molecular chromophores is crucial for designing nature-inspired functional molecules, with applications ranging from photosynthesis to pharmaceuticals. Non-adiabatic molecular dynamics simulations are powerful tools to investigate the photochemistry of molecules and materials, but demand extensive computing resources, especially for complex molecules and environments. To address these challenges, the integration of machine learning has emerged. Machine learning algorithms can be used to analyse vast datasets and accelerate discoveries by identifying relationships between geometrical features and ground as well as excited-state properties. However, challenges persist, including the acquisition of accurate excited-state data and managing the complexity of the data. This article provides an overview of recent and best practices in machine learning for non-adiabatic molecular dynamics, focusing on pre-processing, surface fitting, and post-processing of data.

vision).¹ Molecular discoveries can be scaled to the materials level, encompassing light-harvesting molecules, catalysts, or even drugs² that act through photochemical reactions. Beyond fundamental insights into photochemistry, there is a high likelihood of broader impacts in atmospheric chemistry, solar energy conversion, photoresponsive materials, and molecular electronics.^{3,4}

Understanding and predicting the photophysical and photochemical properties of molecular systems requires detailed knowledge of their potential energy surfaces (PESs). However, in polyatomic molecules, PESs are inherently high-dimensional, defined by numerous internal coordinates, making their full characterization computationally intractable. A powerful strategy to address this challenge is non-adiabatic molecular dynamics (NAMD) simulations, which enable the exploration of PESs by directly identifying the critical geometries visited upon photoexcitation (see also Section 2). Through this approach, NAMD facilitates the characterization of structure–property relationships that govern the excited-state processes and reactions that occur after light absorption. The trajectory data obtained from NAMD simulations serve as a basis for identifying different nonradiative decay channels, assessing their efficiency, and determining characteristic time scales – key insights that inform the rational design of novel chromophores, materials, and photonic devices. However, despite their ability to resolve real-time excited-state molecular vibrations and reaction pathways toward various photoproducts,⁵ NAMD simulations are computationally demanding. Due



to their statistical nature, achieving meaningful insights requires the propagation of numerous trajectories, each typically evolved with time steps on the order of 0.5 fs. Consequently, a single 1 ps simulation necessitates approximately 2000 quantum chemical calculations. These high computational costs constrain the scope of NAMD applications, particularly for large molecular systems, complex environments,⁶ and simulations extending beyond the sub-nanosecond regime.

This challenge has spurred the integration of machine learning (ML) techniques into the study of photodynamics, offering a promising route to overcome existing limitations of NAMD simulations and to access previously unexplored regions of chemical space.^{7–10} ML has become a powerful tool in electronic structure theory, widely used to predict a broad range of properties including Hamiltonians,¹¹ electronic energies, forces, dipole moments, and even experimental observables such as spectroscopic features.^{10,12–14} Among its many applications, ML potentials for ground-state dynamics are arguably the most successful and broadly adopted.¹⁵ More recently, efforts have expanded to include ML potentials capable of representing multiple electronic states, thereby enabling excited-state simulations.

In this setting, ML potentials serve as efficient surrogates for potential energy surfaces (PESs), either for a single state (ground state) or for multiple electronic states. Leveraging large datasets of quantum mechanical calculations or experimental data,^{10,12–14} ML models can learn complex structure–property relationships and accurately predict key quantities for NAMD simulations, such as energies, forces, non-adiabatic couplings (NACs), and spin–orbit couplings (SOCs). Their significantly lower computational cost compared to *ab initio* methods enables simulations of excited-state processes at extended timescales.^{16–18} However, ML-driven photodynamics remains a nascent field and faces several important challenges. These include (i) the limited availability of high-quality excited-state reference data, (ii) the non-uniqueness of certain properties due to wavefunction phase arbitrariness,^{19,20} and (iii) discontinuities in PESs near regions of strong coupling.^{8,21} Together with the high computational cost of generating reliable training data, these issues currently hinder broader adoption.

To address these challenges, this article presents a consolidated overview of current best practices for the development and application of ML potentials in excited-state dynamics. Drawing on insights from the CECAM workshop “Machine-learned potentials in molecular simulation: best practices and tutorials”, we focus on ML models tailored to excited-state simulations within the framework of mixed quantum-classical (MQC) NAMD^{20,22–27} – the most widely used setting for such approaches. Alternative strategies, such as the direct learning of wavepacket propagation, are not covered. Instead, the emphasis is placed on the practical aspects of supervised ML workflows, including data generation and pre-processing, model training and refinement, and the analysis of NAMD trajectories through ML-based post-processing.

The structure of this article reflects a typical ML-driven workflow for excited-state simulations (*cf.* Fig. 1). We begin with a concise introduction to NAMD, with a focus on surface

hopping techniques as the underlying framework for most current ML applications in excited-state dynamics (Section 2). This is followed by a discussion of the data foundation required to train reliable ML potentials, including the selection or computation of quantum chemical reference data (Sections 3.1 and 3.2) and the pre-processing of quantum chemical data (Section 3.3). We proceed by outlining the construction of machine learning models for excited-state dynamics, beginning with the selection of suitable molecular structure representations (Section 4.1) and regression architectures (Section 4.2). Subsequently, we highlight representative implementations of excited-state ML potentials (Section 4.3) and provide a comparative analysis of single- *versus* multi-state architectures (Section 4.4). Further, we address the incorporation of phase correction during model training (Section 4.5) and examine the critical challenges of ML for quantum dynamics (Section 4.6) and transferability across chemical compound space (Section 4.7). Finally, we address the post-processing stage, highlighting best practices for explorative trajectory analysis (Section 5.1) using methods such as dimensionality reduction (Section 5.2) and clustering (Section 5.3).

2 Fundamentals of NAMD

While comprehensive reviews on non-adiabatic molecular dynamics (NAMD) are available in the literature (see, *e.g.*, ref. 28–32 or excited-state ML reviews^{8,9,14,33}), we provide here a brief overview tailored to the context of machine learning applications, with a particular focus on surface hopping methods.

NAMD methods have evolved into various approaches, from fully quantum to semiclassical methods. Quantum methods, such as Multiconfiguration time-dependent Hartree (MCTDH)³⁴ or variational multiconfigurational Gaussian (vMCG)³⁵ propagation methods, directly consider the nuclear wavefunctions and offer insights into nuclear quantum effects. A schematic example of wavefunction propagation is shown in Fig. 2a, where a wavepacket is split along the trajectory. However, because of their computational costs, these methods usually require selecting a few degrees of freedom for the propagation, the assumption of model potential energy surfaces (PESs) or the use of linear vibronic coupling models. Trajectory-based methods, such as *ab initio* multiple spawning (AIMS),^{36–38} Ehrenfest dynamics,^{39,40} and trajectory surface hopping (TSH),^{40–42} assume that the nuclear wave function can be approximated by a swarm of classical trajectories, exemplified in Fig. 2b. Among these approaches, the trajectory surface hopping method is one of the most widely used techniques for investigating photoinduced processes and reactions of medium-sized molecules on the picosecond timescale.³⁶

TSH simulations are usually performed on the fly, considering all vibrational degrees of freedom. However, approximated PESs obtained from, for instance, the linear vibronic coupling model, can also be used.⁴³ Notably, model PESs make the simulations much more computationally efficient but introduce approximations. Accurate solutions for linear vibronic coupling systems can only be obtained for rigid systems. ML-based PESs offer higher accuracy and high



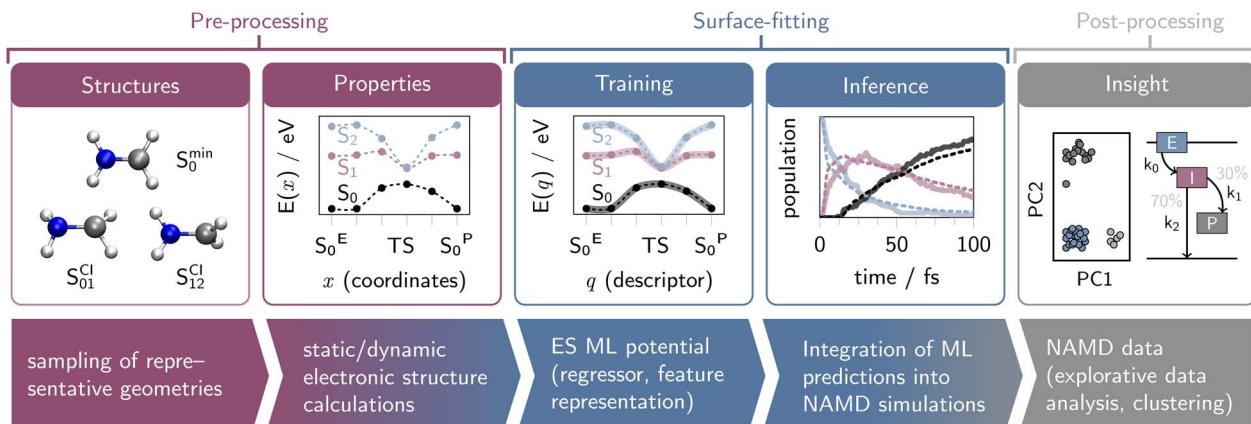


Fig. 1 Schematic illustration of a typical workflow for computing excited-state phenomena using machine learning (ML) driven non-adiabatic molecular dynamics (NAMD) simulations (see Section 2). The workflow is divided into three main areas: data pre-processing (generation and curation of data, see Section 3), surface-fitting (training, evaluation, and refinement of ML potentials for excited states, followed by their application in NAMD simulations; see Section 4), and post-processing (exploratory analysis of NAMD trajectory data, such as dimensionality reduction or clustering; see Section 5).

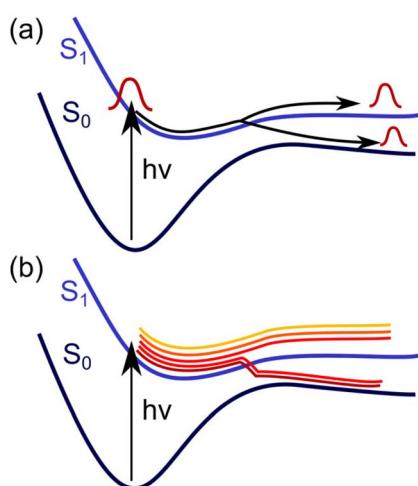


Fig. 2 Quantum (a) and classical (b) propagation of the nuclei in excited-state molecular dynamics simulations. Reproduced from ref. 8 under CC-BY 4.0.

computational efficiency by learning quantum chemical reference data.^{8,44} We will now focus on TSH trajectory data that can be fed into ML models, facilitating NAMD simulations.

2.1 Surface hopping methods

The mixed quantum-classical surface hopping^{40,42} methodologies offer favourable simplicity, efficiency, and scalability for studying non-adiabatic phenomena in molecular systems. They assume, that the nuclei move on one single PES at a time, *i.e.*, the active state. The non-adiabatic transitions are approximated with instantaneous switches between the adiabatic PESs, giving rise to “surface hopping”. Some nuclear quantum effects can be recovered *via* considering ensembles of trajectories. An example with a swarm of 5 trajectories, where two undergo a transition from the excited state to the ground state, is shown in Fig. 2b.

Several NAMD packages are available to generate the surface hopping trajectories, to name a few: SHARC,^{44,45} Newton-X,^{46,47} PyRAI²MD,¹⁶ MLatom²⁵ and JADE-NAMD.⁴⁸ Despite the various implementations of the surface hopping methods in these packages, they share the same aspects to produce the trajectory data, including the surface hopping algorithms, excited-state calculations, and generation of initial conditions used as starting structures for NAMD.

The electronic wave function in surface hopping is a linear combination of multiple electronic states with the same or different spin multiplicities at specific nuclear positions. The temporal evolution of nuclear positions, referred to as trajectory propagation, updates the nuclear coordinates and velocities according to classical dynamics; usually by integrating Newton's equation of motion. The current state for propagating the nuclear trajectory is determined stochastically *via* the coefficients of the electronic states. The square of these coefficients represents the electronic state populations, and their time derivatives indicate the tendency of non-adiabatic electronic transitions between states.³⁶

The fewest switches surface hopping (FSSH) method assumes the least number of switches between two electronic states, meaning that electronic population transfer occurs primarily from one electronic state to another.^{40–42} The transition probabilities in FSSH dynamics depend on the non-adiabatic couplings (NACs) between electronic states with the same spin multiplicity (*i.e.*, internal conversion),⁴⁹ which describe the steepest changes in the wave function as nuclear motions occur. When intersystem crossing is considered, spin-orbit couplings (SOCs) are also required to account for hops between electronic states with different spin multiplicities (*i.e.*, intersystem crossing).^{50,51} The NACs and SOCs connect the classical nuclear dynamics and electronic wave function, governing the temporal evolution of electronic populations over all considered states.

Traditional FSSH shows overcoherence between electronic states because the population transferred to an upper or lower



state follows the gradients of the current state. Various approaches have been developed to address overcoherence, including augmented FSSH (A-FSSH), decoherence-induced surface hopping (DISH), methods based on the decay of mixing (DM), and overlap decoherence correction (ODC), among others.⁵² All these methods could be used to generate training data for ML. However, due to its simplicity and effectiveness, the most commonly used approach is based on the simplified decay of mixing, which dampens the coefficients of the inactive states at each time step. Overcoherence is usually accounted for in NAMD programs and approximations such as the ones proposed by Persico and Granucci are applied.^{44,46,49,53}

3 Data as the backbone of ML-driven NAMD

At the core of every machine learning (ML) model lies its training data. The predictive performance, generalizability, and physical reliability of ML models in the context of NAMD simulations are critically dependent on the quality, diversity, and physical relevance of the underlying data. This section outlines best practices for generating and curating training data suited for ML potentials targeting excited-state dynamics. We begin by briefly discussing the selection of suitable quantum chemical (QC) reference methods – primarily for completeness – as this topic has been reviewed in detail elsewhere (see, *e.g.*, ref. 29, 54, 55 and 56). Next, we outline data sampling strategies, with particular emphasis on the active learning approach, as a means of efficiently generating data. The section then focuses on the various types of electronic structure data needed for training ML models, such as energies, forces, and coupling properties, and addresses the challenges of their computation and preprocessing. Finally, we comment on the diabatization and different bases of excited states. Additionally, we provide practical recommendations for preprocessing strategies to ensure that the data is compatible with ML models. A summary of recommended practices for the construction of (initial) training datasets for excited-state ML models is provided in Fig. 3 at the end of the section.

3.1 The data quality matters: electronic structure methods for data generation

Selecting an appropriate quantum chemical reference method is a critical step in NAMD simulations, as it directly influences the accuracy of PESs and the couplings thereof.^{21,57} The reliability of the simulations depends on how well the chosen method captures electronic structure effects, including state mixing and energy ordering. Variations in the reference method can lead to significant differences in the predicted excited-state dynamics, impacting important properties such as electronic populations, relaxation pathways, and reaction mechanisms.^{21,57–59} These methods can generally be divided into two broad categories: single-reference and multi-reference approaches, each offering distinct advantages and limitations depending on the system under study and the desired level of accuracy and efficiency.

DATA GENERATION & PRE-PROCESSING

Data Generation:

- Reference method accurately describes the chemistry/physics (*e.g.*, verify *via* comparison to experimental observables).
- Reference method is suitable for the target molecule (*e.g.*, consider multireference character).
- Required properties (*e.g.*, NACs, SOCs, energies, gradients) are available from the chosen method.
- Computational cost for generating reference data is feasible (alternatively consider Δ ML or transfer learning where needed).

Pre-processing:

- Conformational Space*: Ensemble of geometries that adequately sample a relevant conformational space.
- Property Space*: Special geometries and data points in regions of discontinuities (*e.g.*, conical intersections and local minima in specific electronic states).
- All necessary quantities are computed for the geometries of interest (*e.g.*, along a trajectory and at static points).
- Account for **discontinuities**:
 - Learn 'smoothed' NACs ($\tilde{C}_{ij} = C_{ij} \cdot (E_j - E_i)$).
 - Provide sufficient data points in regions with discontinuities.
 - Consider learning in diabatic vs. adiabatic representation.
- Consider **phase problem** of coupling terms:
 - Correct all phases in advance (wavefunction overlap).
- If necessary, extend/diversify the dataset using **active learning**.

Fig. 3 Checklist for data pre-processing in the context of ML-accelerated NAMD simulations: Selection of electronic structure method and selection or generation of training data.

Single-reference methods compute excited-state electronic structures using the ground state as a reference. Common approaches include time-dependent density functional theory (TDDFT)⁶⁰ and its spin-flip variant (SF-TDDFT),^{61–63} coupled-cluster methods such as approximate second-order coupled-cluster (CC2),^{58,64} and *n*th-order algebraic diagrammatic construction methods like ADC(2), ADC(3), ADC(2)-x, SOS-ADC(2)-x.⁶⁵ Among these, (spin-flip) TDDFT and ADC(2) are the most widely used single-reference methods for TSH simulations.^{29,59} However, these methods must be used with caution, particularly in TSH simulations, as they often fail to accurately describe conical intersections between the ground and excited states.^{66,67} However, they are computationally efficient and accurate for dynamics within excited-state potentials. Especially when multiple states (more than 3 or 4) are considered, they can be beneficial compared to state-average-based multi-reference methods. This issue arises frequently in photochemical reaction pathways, where reactions may involve doubly excited states and proceed through S_1/S_0 crossing regions. In such cases, the common practice for propagating TSH trajectories is to continue the simulation until a crossing region is encountered, typically defined by an energy gap of less than 0.1 eV between states.⁴⁴ At this point, one of the following actions is usually taken: (i) stopping the simulation, or (ii) assuming an instantaneous transition to the ground state.⁶⁸ However, the assumption of an instant transition should be used with

caution, as it may lead to a significant overestimation of decay rates.⁶⁸ To address these challenges, the spin-flip technique (e.g., SF-TDDFT)^{61,63} was introduced to better describe the potential energy surfaces around conical intersections. While this method improves the accuracy of simulations in these regions, it is still susceptible to spin contamination, which requires careful monitoring throughout the simulation. Alternatively, methods such as mixed-reference spin-flip TDDFT (MRSF-TDDFT)⁶⁹ and spin-restricted ensemble-referenced Kohn-Sham (REKS)⁷⁰ can also handle conical intersections but at the cost of the simplicity associated with single-reference methods. Among these, MRSF-TDDFT has shown particular promise by addressing the spin contamination issues of SF-TDDFT, and it has been successfully applied in non-adiabatic dynamics simulations using the fewest-switches surface hopping (FSSH) algorithm.⁷¹

In contrast, multi-reference methods are essential for accurately describing regions of degeneracy and conical intersections, as they explicitly account for strong electron correlation effects. Several popular methods rely on the complete active space self-consistent field (CASSCF) approach and build upon the CASSCF reference. For example, complete active space second-order perturbation theory (CASPT2)⁷² adds second-order perturbative corrections (*i.e.*, dynamical correlation) to the CASSCF reference. Other multireference methods, such as n -electron valence state perturbation theory (NEVPT2)⁷³ and multireference configuration interaction (MRCI),^{74–76} also extend the capabilities of CASSCF to better describe excited states. Additionally, variants of CASPT2, such as extended multistate (XMS),⁷⁷ dynamically weighted (XDW),⁷⁸ and rotated multistate (RMS),⁷⁹ provide improved energy corrections, particularly near state-crossing regions. Despite their advantages, these methods cannot be used as black-box methods, and running these calculations is limited due to the need for selecting active spaces capable of describing all steps in the dynamics. Intruder states can enter the active space, affecting energy conservation. These issues make excited-state dynamics with MC calculations extremely challenging. Recent works show that the adaptive sampling configuration interaction (ASCI) method can expand the active space size beyond 50 electrons and 50 orbitals.^{80,81} The costs of MC calculations are only affordable for propagating TSH trajectories of small molecules.^{82,83} The multiconfiguration pair-density functional theory (MCPDFT) combines CASSCF for static correlations and DFT for dynamical correlations.^{84,85} It shows comparable accuracy to the CASPT2 method at the cost of CASSCF calculations, which offers another choice for obtaining TSH trajectory data.⁸⁶ In some cases, neither SC nor MC methods can propagate the TSH trajectories correctly and a combination of QM methods has to be employed.^{87,88} An alternative strategy to learn the energies, forces and NACs, is the one adopted by Booth *et al.*,⁸⁹ which builds upon the concept of eigenvector continuation. This approach allows for the interpolation of many-body wavefunctions and the analytical evaluation of forces and NACs. The method was applied to the surface hopping dynamics of a set of linear hydrogen chains within an active learning protocol. Simulations for the longest chain, H₂₈, required only 22 DMRG

training calculations. The method can also be applied using other correlated electronic structure techniques.

3.2 The sampling matters: strategies for conformational coverage

Reference data for machine learning (ML) applications in NAMD simulations are typically derived from both static and dynamic quantum chemical simulations. These datasets encompass key geometries along photochemical reaction pathways, as well as trajectory data obtained from NAMD simulations. While the selection of appropriate electronic structure methods for generating these reference data is crucial (see Section 3.1), as their inherent limitations directly influence the accuracy with which photophysical and photochemical processes are modelled, another key consideration is the coverage of conformational space provided by the reference data.

Even when comprehensive electronic structure data are available, such as the electronic properties of a set of key geometries (*cf.* Section 3.2.1) or preliminary data from a few short NAMD trajectories (*cf.* Section 3.2.2), an ML model trained on this data may not be capable of accurately predicting the electronic properties of unseen molecular conformations.²⁴ This includes conformations that are only visited during extended, long-time-scale NAMD simulations. However, the ability of the model to generalize to such unseen conformations is essential for performing ML-accelerated NAMD simulations over longer time scales.

To overcome this limitation, it is often necessary to construct databases that not only incorporate typical data points sampled *via* static and dynamic methods grounded in chemical intuition, but also to supplement these with additional data points (conformational structures) that effectively map the potential energy surface. Ideally, the inclusion of new data should focus on points that significantly differ from the existing ones, thereby enhancing the predictive capability of models.

In this section, we provide a brief overview of traditional approaches for sampling data points in computational photochemistry, both through static and dynamic methods (Sections 3.2.1 and 3.2.2), introduce the key principle of active learning for diversifying datasets (Section 3.2.3) and conclude with hybrid approaches (Section 3.2.4), emphasizing established best practices.

3.2.1 Static approach. The static approach focuses on identifying key stationary points (geometries) across electronic states using electronic structure theory, specifically solving the Schrödinger equation within the Born–Oppenheimer approximation. Key geometries include minima on the ground-state PES, such as reactants and products, and transition states that define energy barriers. Upon photon absorption, excited-state wave packets can overcome these barriers to reach crossing points where non-adiabatic population transfer occurs, breaking the Born–Oppenheimer approximation and coupling electronic and nuclear motions. This leads to processes like radiationless decay (internal conversion) or photochemical reactions, including conical intersections



between states of the same multiplicity (internal conversion) and those of different multiplicity (*e.g.*, S_1 and T_1), which mediate intersystem crossing (ISC) if spin–orbit coupling (SOC) is significant.

The identification of stationary points on PESs is guided by chemical intuition, utilizing well-established optimization methods like the global optimizer algorithm (GOAT),⁹⁰ conformer–rotamer ensemble sampling tool (CREST)^{91,92} to localize global and local minimum geometries. To locate transition states in both ground and excited states, the pysisyphus software package⁹³ can be employed. It supports traditional methods, including intrinsic reaction coordinates and chain-of-states approaches, such as the Nudged Elastic Band (NEB) and String Method (SM), to locate transition states. Minima in the conical intersection seam between two electronic states can be identified using standard geometry optimization algorithms. For these cases, single-reference methods, such as spin-flip time-dependent density functional theory (SF-TDDFT), or multireference methods are recommended for better accuracy (see also Section 3.1 and ref. 94–98). Once stationary points are located, additional static data points can be sampled through geometry interpolation techniques that connect these stationary points. This approach has been successfully applied in the construction of excited-state databases like WS22 (ref. 99) and XAlkeneDB²⁴ (see cross-marks in Fig. 4a) or in ref. 22.

Another important step is the generation of initial conditions, which serve as the foundation for performing subsequent NAMD simulations. Beyond defining positions and momenta, these conditions should also capture quantum effects in semi-classical simulations. Most NAMD studies assume vertical excitation by ultrashort pulses without explicitly modelling laser–molecule interactions.¹⁰⁰ A widely used approach is the

nuclear ensemble approach (NEA), where a distribution of ground-state geometries serves as the basis for generating an excited-state ensemble.^{101,102} If a molecule has multiple thermally accessible local minima in the ground state, such as conformers, the ensemble is typically constructed by sampling geometries from each conformer according to their Boltzmann-weighted distribution.⁸⁷ The distribution of ground-state geometries itself is generally obtained from a local minimum or a slightly displaced structure using Wigner sampling or molecular dynamics (MD) simulations. While Wigner sampling accounts for vibrational wavefunctions in position and momentum space, it struggles with anharmonic low-frequency modes ($<500\text{ cm}^{-1}$). Moreover, Wigner sampling is limited to the configurations immediately related to equilibrium geometries. In contrast, classical MD sampling generates a Boltzmann-like ensemble at room temperature but often underestimates the total energy relative to the zero-point vibrational energy (ZPVE).¹⁰³ This limitation can be mitigated by propagating dynamics at higher temperatures or applying a quantum thermostat to broaden the classical distribution.^{100,104,105} Alternatively, nuclear quantum effects can be incorporated through path integral methods.^{106–108}

3.2.2 Dynamics approach. The dynamics approach focuses on generating training data from NAMD simulations. Unlike the static approach (see Section 3.2.1), which is guided by chemical intuition, the dynamics approach benefits from directly simulating the temporal evolution of coupled nuclei and electrons. This allows the model to explore a broader range of geometries that may not be captured by static methods, preventing the potential overlooking of important relaxation mechanisms. For further details on the dynamics approach, we refer the reader to Section 2.

Since trajectory based NAMD simulations are stochastic, a larger number of trajectories is necessary to accurately capture the dynamics and inherent variability. The required number of trajectories can vary depending on factors such as system complexity, excited-state reactivity, and the specific experimental data being compared. In previous studies, typically between 50–100 trajectories (performed at the electronic structure level of theory) were used to generate training data.^{17,27,109–111} This number of trajectories was found to provide adequate conformational information for learning the ground- and excited-state PESs.^{17,27,109–111} However, propagating such a set of *ab initio* NAMD trajectories can be prohibitively expensive for training. Thus, it is usually more efficient to run additional trajectories within the ML model itself, rather than generating a large volume of trajectory data directly using electronic structure theory.^{17,24,27}

However, underrepresentation of regions with small inter-state energy gaps in trajectory data presents a challenge.^{17,110} State crossing events typically occur at these small energy gaps, which can lead to insufficient sampling of critical regions. Therefore, we recommend including a limited number of trajectories or trajectory snapshots in the training data. However, the key to generating accurate and reliable datasets lies in the choice of the electronic structure method, coupled with additional sampling of geometries near small energy gaps.

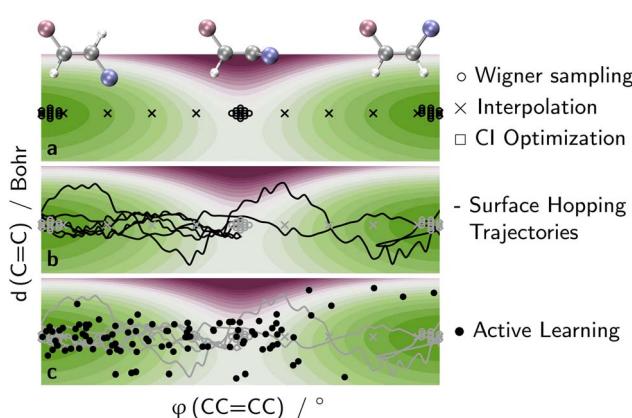


Fig. 4 Schematic illustration of a 2D-PES of 2-butene (shown for the ground state, S_0 , with excited-state minima indicated), defined along the rotation around and stretching of the central alkene bond (data is taken from ref. 24). The figure highlights key steps in constructing training databases for excited-state ML potentials. Static sampling approaches (a) include Wigner sampling around local minima (circle symbols), optimization of conical intersections (S_{01} , square symbol), and interpolation between minima (cross symbols). Surface hopping trajectories (b) expand the sampled configurational space (particularly for the *E*-isomer), while active learning (c) identifies additional critical geometries in previously unexplored regions.



This strategy ensures that critical regions, such as conical intersections (see Section 3.2.1), are adequately captured. Recently, an active learning strategy driven by inter-state energy gaps has further emphasized the importance of targeted sampling in these degeneracy regions¹¹² (see the subsequent Section 3.2.3).

Notably, a recent study compared the test errors of random and trajectory-based data splitting (see Fig. 5a vs. 5b).²⁷ While random split led to improved test performance, it underestimated the true error. This and other studies have highlighted the benefits of including trajectory snapshots rather than full trajectories.^{22,87,109–112}

3.2.3 Active learning. To enable ML-accelerated NAMD simulations over extended timescales (e.g., in the picosecond to nanosecond regime), it is crucial that the training data accurately maps the topography of the electronic-state manifold in the regions accessed during dynamics.¹⁷ Constructing such representative datasets remains a non-trivial challenge. Existing approaches often rely on manually curated data sets (see Sections 3.2.1 and 3.2.2), which may insufficiently cover critical areas near conical intersections. These regions, while central to non-adiabatic transitions, are rarely sampled in conventional NAMD trajectories and thus constitute only a small fraction of total trajectory points. This data imbalance explains why some prior studies^{24,109,111,113} – despite using large training sets (e.g., based on 100 reference trajectories) – still failed to maintain robust ML dynamics and had to revert to reference electronic-structure calculations when encountering small energy gaps during trajectory propagation. An illustration on gradual expansions of training set domains is provided in Fig. 7.

To overcome the limitations of manually curated training sets, active learning (AL) is frequently employed to iteratively refine and optimize the data – selecting only the most informative points to improve model generality and efficiency.^{17,114,115} In essence, AL involves identifying and sampling regions of configurational space where the model is uncertain, computing their *ab initio* properties, and retraining the model accordingly.¹¹⁶

Determining which configurations require recalculation depends on assessing predictive uncertainty. While uncertainty can, in principle, be estimated based on structural descriptors – such as determining whether a geometry's Coulomb matrix falls outside the 95% confidence interval of those in the training set.¹¹⁷ Most approaches in the excited-state community rely on property-based uncertainty quantification. While some models,

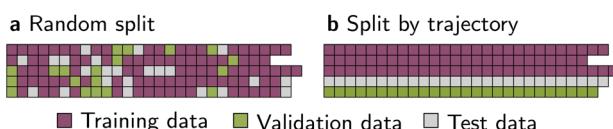


Fig. 5 Strategies for partitioning data from NAMD simulations into training, validation, and test sets, illustrated with 5 example trajectories of varying lengths. Random split (a): trajectory frames are pooled and randomly allocated to one of the three subsets. Split by trajectory (b): all frames from a single trajectory are assigned to the same subset. The figures are adapted from ref. 27.

like Gaussian processes, offer built-in uncertainty estimates, ensemble-based approaches, such as multiple independently trained neural networks, are commonly used for models that lack this feature, particularly neural networks.^{7,21} Among these, the widely adopted strategy of query-by-committee leverages at least two separately trained neural networks to evaluate disagreement in predicted molecular properties (e.g., energies, forces, dipole moments, or NACs) during NAMD simulations.^{17,112,114} High discrepancies between models signal that a geometry likely resides in an undersampled or poorly learned region of the potential energy surface.

To quantify such disagreement, uncertainty quantification (UQ) metrics are important to assess the reliability and interpretability of models.¹¹⁸ Early work by Behler *et al.*¹¹⁹ established key principles for assessing predictive uncertainty in atomistic neural networks (NNs). Building on this, Musil *et al.*¹²⁰ demonstrated the use of ensembles combined with resampling techniques to improve uncertainty estimates in machine-learned potentials. Comprehensive reviews on UQ for NNs can be found in ref. 121 and 122, covering a wide range of methodologies from Bayesian inference to ensemble methods. For Gaussian process regression (GPR), which naturally provides predictive variance, Deringer *et al.*¹²³ offer an extensive review. More recent advances include novel UQ frameworks for NNs by Kellner *et al.*,¹²⁴ which integrate uncertainty estimates with improved computational efficiency. Additionally, Ceriotti and co-workers have introduced computationally inexpensive UQ methods^{125,126} that are especially suited for high-throughput molecular simulations.

In the excited-state context, UQ measures like absolute errors¹¹² or root mean squared errors (RMSEs)^{17,114} are applied. Once the model has undergone an initial training phase, it is used to sample new geometries. If the UQ value at a given configuration exceeds a predefined threshold, the simulation is paused, the configuration is recalculated using the reference electronic structure method, and the resulting data is added to the training set. Retraining follows, and the process is repeated

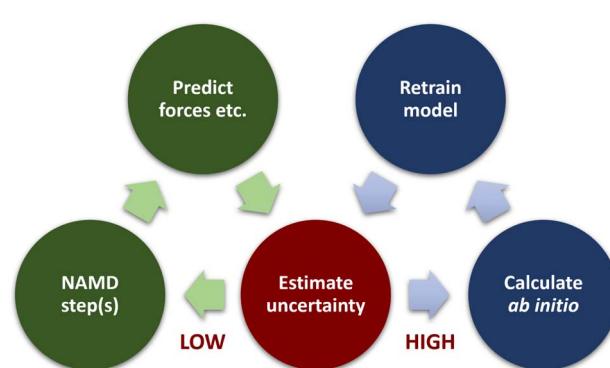


Fig. 6 Simplified diagram of an active learning procedure: during NAMD, properties are predicted with more than one ML model or KMs and the uncertainty is estimated. In case the uncertainty exceeds a certain threshold predefined by the user, the data point is recomputed with the reference method, added to the data set, models are retrained, and the procedure is carried on.



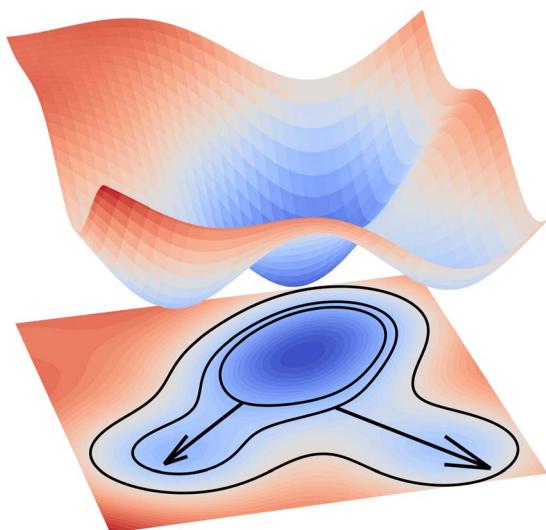


Fig. 7 Schematic illustration of the gradual expansion of the training domain (e.g., *via* manual sampling or active learning) when exploring a potential energy surface by means of molecular dynamics. Each of the arrows represents one dataset extension, such as one active learning step or the manual addition of key geometries (e.g., conical intersections or minimum energy crossing points).

until the model consistently performs across all relevant regions of the PES. A schematic of this iterative AL workflow is provided in Fig. 6.

While this iterative process has been successful in improving ML potentials for excited-state simulations,^{17,87,114} the effectiveness of AL depends on many interconnected parameters. Researchers must make critical choices regarding the initial data, the ML model (including descriptors and regression algorithms; see Section 4), simulation conditions, UQ method, and thresholds – all of which can significantly influence the final model quality.

The thresholds used to trigger reference calculations in AL loops are highly system- and model-dependent and often refined through empirical testing, whereas Hu *et al.*¹¹⁷ provide a method for an automatic statistically justified choice of thresholds. Energy-based thresholds typically range from 0.03 to 0.04 hartree, aligning with typical gap values where non-adiabatic transitions occur.¹²⁷ Different studies have adapted this general strategy to the specifics of their systems. For example, in the PyRAI²MD (see Section 4.3) study on [3]-ladderdienes undergoing [2 + 2]-photocycloadditions, uncertainty was quantified *via* the standard deviation of model predictions, with thresholds set to 0.043 hartree and 0.25 hartree/bohr for energies and forces, respectively. In the SchNarc workflow (also Section 4.3) for the small cation CH_2NH_2^+ , a more diverse set of thresholds was applied: 0.03 hartree for energies (reduced iteratively by a factor of 0.95), 0.5 debye for dipole moments (kept constant), and 0.25 for NACs, which was temporarily raised to 3.0 to enhance sampling near conical intersections.¹⁷

A more recent study targeting azobenzene and pyrene derivatives further refined this approach by integrating query-

by-committee with gap-driven molecular dynamics (gapMD), a strategy designed to focus sampling around conical intersections.¹¹² Trajectories were steered toward small-gap regions (≤ 0.03 hartree), after which dynamics proceeded on either the upper or lower surface to ensure meaningful sampling and avoid dissociative pathways. The associated AL loop employed a multi-state neural network (MS-ANI, see Section 4.3) trained on both energies and gradients. Retraining was triggered by two conditions: negative predicted energy gaps, which indicate incorrect state ordering, and elevated uncertainty values, defined as the absolute difference between predictions from a main model (trained on energies and gradients) and an auxiliary model (trained on energies alone).

3.2.4 Hybrid approaches. The previously discussed approaches – static (e.g., Wigner and normal mode sampling), dynamic (e.g., ground- and excited-state MD), and AL-based strategies – each contribute useful but incomplete coverage of the relevant regions of the PESs in photochemical systems. While early AL-based ML-NAMD strategies provided valuable insights, they often required manual adjustments and still failed to ensure robust model performance without further interventions such as PES point interpolations.^{17,114} Consequently, individual methods alone have proven insufficient for generating well-balanced training sets.

Recent work has demonstrated that a combined sampling strategy yields the most reliable and compact training data.^{22,24,99,112} Specifically, integrating Wigner sampling (to cover thermal fluctuations of reactants and products), geometrical interpolation (between optimized structures of reactants, minimum energy crossing points, and products), and short-time quantum chemical trajectories (to access relevant non-equilibrium conformations) represents current best practice. This hybrid approach efficiently captures the necessary configurational diversity for accurate ML-accelerated NAMD simulations.

3.3 The curation matters: preparing data for ML

NAMD trajectories provide detailed time-resolved information about molecular systems, including nuclear coordinates, velocities, electronic state populations, energies, energy gradients, and interstate couplings. In this section, we outline the types of data that can be extracted for ML applications and discuss strategies for transforming them into suitable representations for ML models. To structure this discussion, we distinguish between properties associated with individual electronic states (Section 3.3.1) and those that describe interactions between pairs of electronic states (Section 3.3.2). Finally, we address considerations involved in learning within diabatic *versus* adiabatic representations (Section 3.3.3).

3.3.1 Single-state properties. The energies and gradients are the most important data to train an ML potential. The TSH trajectories usually contain energy data for each included electronic state, as they can be computed in one single-point calculation. Most FSSH programs only compute the gradients of the “current” active state needed for propagating the nuclei. In this case, the gradient data only contain partial information



about the excited and ground state before and after the surface hopping. On the other hand, one can compute the incomplete gradient data for the selected structures of the TSH trajectories.

We should note that the energy data are a series of scalar values. In contrast, the gradient data are vectorial data stored in 2D arrays of N nuclei and three Cartesian coordinates or flattened 1D arrays of $3N$ values. While the energy is invariant under permutation, translation and rotation, the gradient data are rotationally covariant (*i.e.*, depending on the molecular orientation). While initial efforts addressed this problem by learning gradients as the first derivative of the energy with respect to nuclear coordinates, equivariant models have since become the state-of-the-art.^{128–130} These models enable the direct prediction of vectorial properties such as forces. However, it remains important to ensure that such direct predictions are consistent with the corresponding energy derivatives.¹³¹

3.3.2 Interstate properties. The arbitrary phase of the wave function makes coupling between different electronic states non-unique, implying an arbitrary sign of inter-state properties; the arbitrary sign must be corrected before or during training. A recent review discusses best practices in ML for chemistry and is recommended to consider before beginning the study.¹³²

Fitting couplings can be done by choosing an initial structure as a reference and correcting the phase for all other structures.¹⁷ This procedure requires computing overlap matrices between the initial structure and any other structure in the training set. Substantial overlap (>0.5 for each electronic state) can identify whether phase change has occurred by assessing the sign of the overlap. Negative overlap values suggest that a phase change has occurred; in that case the property of the new structure must be multiplied by -1 for the given state. The same procedure must be carried out for the other contributing state (*i.e.*, any inter-state property has to be multiplied by -1 or $+1$ twice) because couplings are related to two electronic states. When the differences between two molecular structures are significant, the overlap could be too small to determine the phase factor. In this case, geometry interpolation between the two structures must be done to determine the correct phase.¹⁷

3.3.2.1 Non-adiabatic couplings (NACs). The availability of analytic NAC data depends on the QM methods in the electronic structure codes. NACs are available for multireference and TDDFT methods. One alternative is the use of time-derivative couplings that can be approximated by finite differences considering the overlaps of wavefunctions between successive time steps, which is the common alternative for ADC(2) and CC2 dynamics.²⁹ When NACs are unavailable, ML models are trained to predict the energies and gradients and use TSH algorithms without NACs.

The learning of NACs poses significant challenges, primarily due to two intrinsic properties: (i) NACs exhibit a strong dependence on the inverse of the energy gap between coupled electronic states, which introduces singularities near conical intersections or avoided crossings and leads to highly non-smooth behavior; (ii) the arbitrary phase of electronic wavefunctions results in discontinuous NACs as a function of nuclear geometry. In the following, we outline best practices to

mitigate these challenges, focusing on techniques for smoothing NACs and correcting for random phase behavior.

- Smoothed NACs. NACs between the two electronic states of same spin multiplicity i and j are formally given by

$$d_{ij} = \left\langle \Psi_i \left| \frac{\partial}{\partial \mathbf{R}} \right| \Psi_j \right\rangle = \left\langle \Psi_i \left| \frac{\partial \hat{H}_{\text{el}}}{\partial \mathbf{R}} \right| \Psi_j \right\rangle \cdot (E_i - E_j)^{-1}. \quad (1)$$

This formula¹³³ highlights the inverse dependence on the energy gap $E_i - E_j$, which leads to NAC magnitudes that are near-zero across most of the PES, but reach infinity in the vicinity of conical intersections and avoided crossings where the energy gap approaches zero. This singular behaviour, illustrated by the pink dashed line in Fig. 8, poses a substantial obstacle for ML models, which rely on smooth training data.

To address this, it is advantageous to train ML models on the smooth, gap-independent numerator of eqn (1), rather than on the full NACs. This approach yields continuous learning targets across the PES¹⁷ (see blue dashed line in Fig. 8). Full NACs can then be reconstructed on the fly by dividing the learned quantities by the ML-predicted energy gaps, resulting in smoothed couplings that are more amenable to ML.¹³⁵

- Phase-corrected NACs. The wavefunction phase is randomly initialized at different nuclear positions. However, preprocessing the NAC data for ML requires that the NACs across all trajectories have the same phase. Phase correction of the couplings is typically performed by computing their overlap between two structures at consecutive time steps.^{17,19} These complexities necessitate additional steps for rigorous data preprocessing, enabling the models to handle double-valued properties effectively.^{135,136} A recently developed methodology by Richardson¹³⁶ can also deal with double-valued properties by means of phase-correction performed within the loss function of a model. This is described in more detail in Section 4.5.

The so-called Baeck–An approximation^{137,138} is one approximation from which various formulas have been derived for the evaluation of NACs.^{139,140} For instance, Westermayr *et al.* used

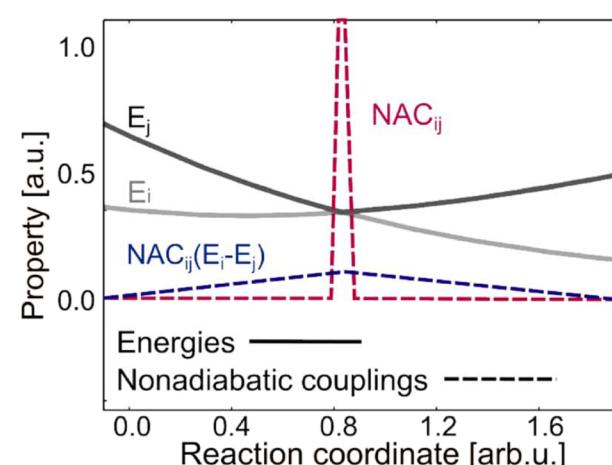


Fig. 8 Conical intersection between two states and corresponding non-adiabatic couplings (reddish, dashed lines) including smooth couplings multiplied by the energy gap (blueish, dashed lines). Reproduced from ref. 134 under CC-BY 4.0.



the Hessian of the squared energy gap potential and the gradient difference vectors, defining the branching space, to approximate NACs using ML.^{87,135,141} Baeck-An couplings, however, are only accurate in regions of the configuration space where two PESs are nearly degenerate. Thus, it is suggested to set the couplings to 0, when two PESs have a $\Delta E > 0.5$ eV.^{19,51} Alternatively, for a two-state problem the Landau–Zener scheme^{142,143} can be used to compute hopping probabilities. Landau–Zener requires the evaluation of the energy gaps and their time derivatives, obtained by finite differences along the trajectories, and originally applies to internal conversion, but was extended to account for intersystem crossing. The results from NAMD with Landau–Zener agree well for small to medium organic systems.¹⁴⁴ A similar approach relying only on the PES shape is the Zhu–Nakamura theory of surface hopping,¹⁴⁵ which uses energies and gradients of two crossing states for the hopping probability calculations. The forces are diabatized in a generalized 1D model based on three-point interpolation.¹⁴⁶ This method agrees well with the fewest switches surface hopping and applies to intersystem crossing.^{146–149}

3.3.2.2 Spin-orbit couplings (SOCs). The preprocessing of SOC data depends on the representations of the spin wave functions in TSH dynamics. Most QM programs compute SOCs in the spin-diabatic representation, where each spin component's SOCs between the singlet and triplet state are calculated. Due to the small values (*e.g.*, ≤ 100 cm^{−1}) in organic molecules, the PESs are usually less affected by the SOCs. Thus, the norm of the SOCs can be directly used to determine the surface hopping probability at intersystem crossing regions.

SOCs are known to substantially increase in molecules with heavy atoms (*e.g.*, S, Br). They introduce important off-diagonal elements to the electronic Hamiltonian, significantly altering the topology of PESs near the intersystem crossing points. Therefore, we must diagonalize the electronic Hamiltonian with SOCs to obtain PESs in a spin-adiabatic representation. It should be noted, that the SOCs lift the degenerate triplet states into different spin states ($z = 0, \pm 1$) in spin-adiabatic representation, which expands the total number of states, including all spin components of a multiplet state.

Learning SOCs in the spin-diabatic representation only requires computing the norm of SOCs without additional preprocessing. The spin-diabatic SOCs can be fitted as scalar values, similar to electronic energies (see Section 3.3.1). However, learning SOCs in the spin-adiabatic representation requires explicit calculations of the interstate couplings between the spin-adiabatic states. Such interstate couplings have the same nature as the NACs, where phase corrections are needed, and can be trained similarly to NACs (see Section 4.5).

3.3.3 Diabatization. Electronic states and corresponding energies, as obtained from electronic structure codes, are typically in the so-called adiabatic representation. These are ordered by energy, and two states of the same multiplicity and symmetry never cross. Instead, they form a conical intersection, a seam of degenerated energies. The dimensionality of such a seam is smaller by two than the system's dimensionality in internal coordinates. For example, the seam consists of just one point for a 2D system, as plotted in Fig. 9.

Conical intersections are incredibly challenging for surface fitting because of the discontinuous potentials near the seam. And, of course, NACs are singular at conical intersections. It is thus advantageous for potential fitting to switch to a different basis to enable smooth evolution along geometrical coordinates with a geometry-dependent transformation matrix $\mathbf{T}(\mathbf{R})$:

$$\Psi_i^{\text{diab}}(\mathbf{r}; \mathbf{R}) = \sum_j^{N_{\text{states}}} T_{ji}(\mathbf{R}) \Psi_j^{\text{adiab}}(\mathbf{r}; \mathbf{R}) \quad (2)$$

The corresponding potential energy matrix $\mathbf{U}(\mathbf{R})$ is given by the following transformation of the diagonal matrix of adiabatic energies $\mathbf{V}(\mathbf{R})$:

$$\mathbf{U}(\mathbf{R}) = \mathbf{T}^T(\mathbf{R}) \mathbf{V}(\mathbf{R}) \mathbf{T}(\mathbf{R}) \quad (3)$$

Note, that the resulting matrix of energies is no longer diagonal; the off-diagonal elements (diabatic couplings) must also be fitted. The basis in which the NACs vanish completely is called a diabatic basis. Unfortunately, such a basis does not generally exist for molecules with three or more atoms.¹⁵⁰ Therefore, the definition of a diabatic basis is ambiguous. The transformation matrix cannot be clearly defined, and many different approaches to diabatization have been proposed,¹⁵¹ including fitting-while-diabatizing method,^{152,153} methods based on NACs elimination,^{154–157} wavefunction similarities,^{158–161} or smooth evolution of molecular properties.^{162–164} Recently, several data-driven ML approaches have been proposed as well, mainly based on NNs.^{23,164–169} If wavepacket propagations are to be performed, surfaces and couplings in a diabatic representation are required rather than the adiabatic representation provided by standard quantum chemistry programs. The reverse transformation from a diabatic to an adiabatic basis can be achieved simply by diagonalization. Other properties apart from energies can also be predicted in the diabatic basis. While atomic forces and approximate NACs can be directly obtained from the diabatic representation, other properties can be fitted separately using the calculated transformation matrix.^{170,171}

The diabatic basis is convenient for ML because of its smoothness. The recommendation is to switch to a diabatic basis whenever possible for a given system. Some quantum

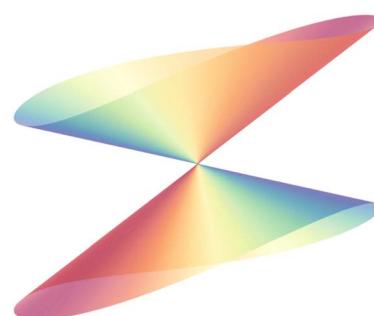


Fig. 9 Schematic illustration of a conical intersection for a two-dimensional system. The seam of the conical intersection consists of only one point. This point separates two adiabatic states from each other while the colors represent diabatic characters.



chemistry codes, such as Molpro,¹⁷² can provide quasi-diabatic energies. The Quantics package can also deliver diabatic energies and couplings using propagation diabatization.¹⁷³ Even for surface hopping, which has been shown to work better in the adiabatic basis,^{40,44} the ML training can be more efficiently performed in the diabatic basis and diagonalized to obtain the adiabatic basis. Interestingly, a new approach for smooth fitting of coupled surfaces avoiding diabatization has been proposed recently.^{174–177} Instead of the energies in either adiabatic or diabatic basis, coordinate-dependent coefficients of the characteristic polynomial are fitted. Even though it is so convenient, obtaining a diabatic representation is usually difficult so this task is often left to cases where quantum dynamics is needed. The fitting of adiabatic energies and properties for trajectory surface hopping has proved to be feasible, barring non-smooth cusps. The dynamics are reliable, because hopping events occur near conical intersections but not exactly on the seam. This simplifies the problem, since the exact seam is not reproduced as accurately as the rest of the potentials. Moreover, diabatic states can be fitted implicitly while minimizing loss function based on diagonalized adiabatic states, thanks to the versatility of NNs.¹⁶⁹ Very recently, a Deep Sets¹⁷⁸ autoencoder has been incorporated into a MACE¹⁷⁹ architecture in order to learn excited states in a permutationally invariant manner within X-MACE.²⁶

4 Learning the landscape: excited-state machine learning potentials

Excited-state machine learning (ML) potentials can address a central challenge in NAMD simulations: the prohibitive cost of computing excited-state properties, such as energies, forces, and non-adiabatic couplings (NACs), at each simulation time step. These models aim to approximate the underlying electronic structure function that maps molecular structures to relevant excited-state properties, as formalized in eqn (4).

$$f: Z, \mathbf{R} \rightarrow E_i, \mathbf{F}_i, \mathbf{C}_{ij} \quad (4)$$

By replacing quantum chemical evaluations with trained models, excited-state ML potentials enable efficient photodynamics and have been implemented in several software packages.^{16,20,24,135,180}

Their successful development hinges on three pillars: access to high-quality reference data (Section 3), an appropriate molecular structure representation (Section 4.1), and a suitable regression model (Section 4.2). Together, these elements determine the model's accuracy, generalizability, and physical reliability. A checklist for surface fitting is provided in Fig. 10.

To be effective, excited-state ML potentials must satisfy several essential criteria. They must preserve key physical symmetries: energy predictions should be invariant to translation and permutation of identical atoms, and vectorial properties (e.g., forces, NACs, and transition dipoles) must transform equivariantly under rotation (Section 4.1). Differentiability is required for computing forces as energy gradients, constraining model architectures (see Section 4.5). Inference

SURFACE-FITTING

Model Selection

ML regressor:

- Neural network
- Kernel method (for ≤ 2000 data points)

ML-NAMD interface:

- Use an existing software packages interfacing ML and NAMD: **MLatom**, **DANN**, **PyRAI²MD**, **SchNarc**, **SPaINN**, or **FieldSchNet**
- Set up a python library or software package designed for ground state potentials to work for excited states.

Model Setup

Descriptor:

- Ensure basic symmetry constraints (SE(3) and permutations).
- Neural networks:
 - Local, learned $>$ local, fixed representation.
 - Equivariant $>$ invariant architecture.
- Kernel methods
- Global $>$ local representation

Consider phase problem of coupling terms:

- Account for the phases during training (*via* a phaseless loss function or double-valued properties).
- Correct all phases in advance (wavefunction overlap).

Model Evaluation

Quantify and evaluate model performance (e.g. uncertainty quantification, query by committee, etc.)

Use active learning to assure model reliability, and to diversify training data.

Fig. 10 Checklist for surface fitting in the context of ML-accelerated NAMD simulations: selection, setup and evaluation of ML potentials for excited states.

should be fast and accurate, with sufficient model flexibility to capture excited-state complexity (see Sections 4.5 and 3.3.3). Additionally, models should scale to larger systems – often achieved through atom-wise descriptors (see also Section 4.1) – and generalize to unseen chemical environments (see Section 4.7). While these criteria are the same as for fitting ground-state properties, excited states introduce new problems that need to be considered and are discussed in the following sections. We begin by introducing the two major design components of ML potentials: molecular structure representations (Section 4.1) and regression models (Section 4.2). Next, we discuss different approaches for treating the manifold of excited states, comparing single-*versus* multi-state approaches (Section 4.4). This is followed by a discussion of training and validation strategies (Section 4.5). We subsequently review existing excited-state ML potentials in Section 4.3, distinguishing between models with fixed *versus* learned representations and models using local *versus* global descriptors. Finally, we comment on the transferability of excited-state ML potentials (Section 4.7).

While reference data (Section 3) forms the foundation for training ML models, the choice of molecular descriptor is equally crucial for performance. In the context of ML potentials,



descriptors are typically categorized as global or local representations, which are discussed in Section 4.1. Equally important is the regression model, which maps descriptors to target properties. In Section 4.2, we introduce the main regression methods: kernel methods and neural networks.

4.1 Molecular representations

Most ML models for excited states learn the relationship between the molecular structure and their excited-state properties. In surface hopping dynamics, molecular geometries are typically represented using Cartesian coordinates of nuclear positions. However, Cartesian coordinates are not uniquely defined, as they depend on the arbitrary choice of the origin and orientation of the molecular system in Euclidean space. To ensure physically meaningful predictions, these coordinates must be transformed into machine-learnable representations that are invariant to translation and rotation.^{181,182}

To this end, molecular structure representations used in ML models must satisfy several key requirements. These include: (i) translational, rotational, and permutational invariance, ensuring that scalar properties such as energy remain unchanged under coordinate transformations or atom indexing; (ii) rotational equivariance for directional properties like forces, dipole moments, and non-adiabatic couplings (NACs), which must transform consistently with molecular orientation; (iii) smoothness and differentiability with respect to molecular geometry; (iv) a one-to-one (biunique) mapping to molecular structure; (v) transferability across chemical compound space; and (vi) computational efficiency.¹⁸³ While symmetries can in principle be learned from data, explicitly encoding them into the descriptors significantly improves learning efficiency and model generalization. For instance, although atom sorting can be used to enforce permutational invariance, it often introduces discontinuities,¹⁸⁴ whereas inherently invariant descriptors avoid this issue.^{185,186}

Molecular representations can be broadly categorized as global, where the entire structure is encoded into a single descriptor, or local, where individual atoms are described based on their local environments. While early ML approaches often used global representations, recent developments – especially ground-state ML potentials – have focussed on local descriptors due to their scalability and ability to model systems of arbitrary size and composition, enhancing transferability and flexibility.

In the following, we briefly outline representative examples of both global and local descriptors employed in excited-state ML potentials. A summary of the key models and descriptors used in the excited-state community is provided in Fig. 11, while Section 4.3 offers an overview of representative software packages, focusing on their application domains, such as multi-state learning or the prediction of NACs, and highlighting how they have been used and evaluated in practice.

- Global descriptors encode the entire molecular geometry into a single vector or matrix, enabling ML models to predict total energies directly as a function of all atomic coordinates $E = f(Z, R)$, without partitioning into atomic contributions (local descriptors).

The most common and straightforward representations include the Coulomb matrix (CM),^{114,134} which captures nuclear charges and interatomic distances; bag of bonds (BoB),¹⁸⁷ is based on the sorting of CM elements; the inverse distance descriptor, which simplifies CM by using only pairwise inverse distances; and the relative-to-equilibrium (RE) descriptor, which normalizes these distances by their equilibrium values.¹⁸⁵ The latter descriptors are central to models like (s)GDML¹⁸⁸ and KREG,^{185,189} which are commonly used ground-state ML potentials.

Global descriptors are typically complete and compatible with standard regression techniques, allowing them to capture all interactions regardless of atomic separation. However, they lack built-in permutational invariance and struggle with transferability to systems of varying size or composition, limiting their scalability. These limitations motivate the shift toward local descriptor approaches in larger or more diverse chemical systems.

- Local descriptors represent the chemical environment of each individual atom within a molecule and encode atom-wise information in vectors or matrices. These descriptors enable ML models to predict molecular properties by aggregating atomic contributions. For example, the total energy can be decomposed as a sum of atomic energies $E = \sum_i E_i(Z_i, \mathbf{R}_i)$.

The representation of the molecular structure can be handled as fixed or learned during training with NNs (see example architectures in Fig. 11). Some of the popular state-of-the-art local and pre-defined (fixed) representations are *e.g.* the smooth overlap of atomic positions (SOAP),¹⁹² atom-centered symmetry functions (ACSF)¹⁹³ and the Faber–Christensen–Huang–Lilienfeld (FCHL) representations.¹⁹⁴ Examples, of learned representations involve SchNet,¹⁸¹ PaiNN¹³⁰ or MACE.¹⁷⁹

Additional hyperparameters must be optimized through a thorough hyperparameter search for fixed representations and require modest computational resources. The number of requisite hyperparameters is lower for learned representation, but training is more expensive. The training process refines the representation and requires additional computational resources for the model to learn this data pattern. Thus, the choice between fixed and learned representations depends on the task's requirements, available resources, and requisite model complexity.⁸

4.2 Regressors

The first choice when it comes to fitting potential energy surfaces and related properties is the selection of the ML regressor: Two basic classes of ML regression models are currently omnipresent: neural networks (NNs) and kernel methods (KMs), such as kernel ridge regression (KRR), Gaussian processes regression (GPR), or support vector machines (SVM). One of the aspects when making the choice is the size of the training ensemble of nuclear geometries.^{134,191} While KMs are relatively efficient with small datasets, they suffer from the fact that training scales cubically whereas the size of the kernel matrix scales quadratically in memory with number of training points. Thus, the usage of KMs for <2000



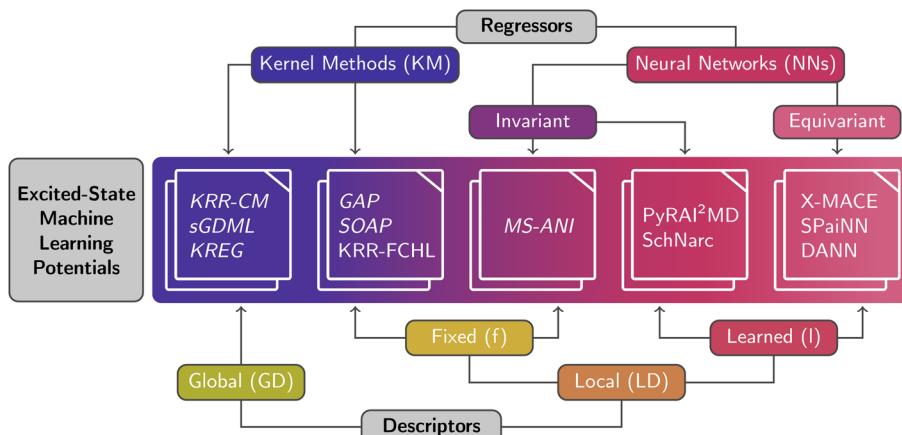


Fig. 11 Overview of excited-state machine learning potentials, categorized into Kernel Methods and Neural Networks, based on both global and local descriptors (fixed and learned). Models handled by MLAtom are highlighted in italic font²⁵ and are interfaced with Newton-X,^{46,47} such as MS-ANI.¹¹² Models interfacing with the SHARC software suite include SchNarc,¹³⁵ FieldSchNet,²⁷ SPaINN,²⁴ X-MACE,²⁶ and KRR-FCHL.¹⁹⁰ Implementations interfacing to other MD drivers include PyRAI²MD²² and DANN.¹⁶⁹ The scheme is adapted from ref. 191.

data points is usually more efficient, while NNs are more appropriate for larger samples. However, there are some approximative ways to make KMs efficient even for large samples.¹⁹⁵

NNs currently present the state-of-the-art for NAMD as relatively large training samples are often required. In addition, NNs allow for simultaneous treatment of all involved electronic states, which is usually both more efficient and accurate. Techniques such as diabatization (see Section 3.3.3) can increase the accuracy of kernel methods as well. While rarely applied in quantum chemistry, NNs can be combined with KMs to get the best of both worlds.^{196,197}

- Kernel methods (KMs). KMs use the Kernel trick, allowing linear regression algorithms to fit non-linear functions by implicitly transforming the input data into a higher-dimensional space.¹⁹⁸ KRR is straightforward and popular in quantum chemistry.^{199,200} KMs represent a category of elementary ML methods that are relatively simple to implement; there are readily available frameworks and libraries. For instance, the scikit-learn Python library can be used to train employing an arbitrary kernel method in just a few lines of code.²⁰¹

Within the KRR method, the properties of interest are modelled as a linear combination of the kernel functions k (*i.e.*, a similarity functions) between the geometry \mathbf{x} of interest and the training points \mathbf{x}_i , written as

$$f(x) = \sum_{i=1}^{N_{\text{train}}} \alpha_i k(\mathbf{x}, \mathbf{x}_i). \quad (5)$$

While the regression coefficients α_i are determined during the training procedure, the kernel is typically predefined. The most common choices are the Laplacian kernel, the Gaussian kernel and its generalization in the form of the Matérn kernel.^{198,199} The Gaussian kernel is usually a good starting point when learning in the configuration space as it is smooth.^{198,202} It is defined as

$$k(x, x_i) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|_2^2\right). \quad (6)$$

with σ being a hyperparameter called the bandwidth or the length scale, defining the locality of the kernel. Alternatively, the best-performing kernel or a linear combination of kernels can be selected during the training phase.

Note, the performance of KMs depends on the molecular representation, that is, how we input the molecular structure into the ML model (see Section 4.1). Therefore, it might be advantageous to use one of the packages developed especially for chemical applications with several representations readily available, such as MLAtom,^{199,203,204} GAP,^{205,206} or QML code.²⁰⁷ A special kernel-based method called symmetric gradient-domain machine learning (sGMDL²⁰⁸) was designed especially for molecular dynamics.^{208–210} Kernel methods together with molecular representations are still being developed and we have seen accurate applications to ground-state MD.²¹¹

- Neural Networks (NNs). The fundamental unit of an artificial NN is a single neuron, which applies weights and bias to the inputs, and outputs the result after possibly applying a non-linear, so-called activation function. The neural units are connected in layers, linked through trainable parameters. A NN typically consists of an input layer, one or more hidden layers, and an output layer. The interconnected node architecture is thus analogous to the brain; each connection has adjustable parameters that allow the network to learn from prior information.^{116,212}

Various NNs are widely used to learn ground-state electronic properties of molecular systems, particularly energies and gradients. Many advanced methods leverage the local nature of chemical interactions by representing structures on a per-atom basis. These atom-wise descriptors predict extensive properties, like total energy, as a sum of atomic contributions. This approach offers key benefits: linear scaling with system size, preservation of size-extensivity, and permutational invariance of energy. Examples include Behler–Parrinello NNs (BPNNs),²¹³

ANI,²¹⁴ and ACE²¹⁵ for fixed, rotationally invariant descriptors, and PaiNN,¹³⁰ Nequip,²¹⁶ MACE,¹⁷⁹ and So3krates²¹⁷ for learned, rotationally equivariant descriptors (*cf.* right side in Fig. 11).

In those examples, users do not necessarily have to develop the ML codebase on their own to train and predict properties. SchNetPack²¹⁸ is a prominent example, which includes several models like SchNet,¹⁸¹ PaiNN,¹³⁰ SchNOrb,²¹⁹ or FieldSchNet,²²⁰ the latter allowing the treatment of electric fields and thus environmental effects. It offers various modules to predict ground state properties (*e.g.*, energy, forces, dipole moments). The default settings of these models suffice for most tasks, but additional parameter adjustments enable more complex tasks.^{20,24,135}

4.3 Existing excited-state implementations

In the following, we briefly outline existing software code implementations that allow for the training of excited-state ML potentials, highlighting their applicability in the prediction of single-state or interstate coupling properties. We start with kernel methods and subsequently turn to NN architectures.

- Kernel methods.^{25,134,221} In the area of Kernel methods, there are no chemistry-tailored codes that were primarily designed for excited-state dynamics. That means that they do not natively support the fitting of the coupling properties or multiple PESs at once. However, they can be easily utilized for training on individual states, especially when using curvature-driven NAMD such as Landau–Zener NAMD.

In one of the first approaches the inverse distance and FCHL molecular structure representation were employed in an NN and KRR approach.¹³⁴ All models were shown to learn the relation between either energies and forces or NACs and the molecular structure, when the properties are treated separately from each other. However, it was found that training a single ML model to predict energies, forces, and NACs simultaneously reduces accuracy compared to learning each property separately. Including all properties in a shared loss function can even hinder the learning of individual properties due to conflicting learning signals.

Encoding of the energy levels (state numbers) in the representation was shown to improve results and make multiple outputs for KRR possible, while multi-outputs are achieved more straightforwardly for NNs. In both cases the modification of the representation to yield multi-state outputs was accompanied by an enlargement of the ML model, a larger kernel matrix in the case of KRR, and a larger input layer in the case of NNs.¹³⁴

The MLatom package^{25,203,204} provides a suite of kernel- and NN based models, which have recently been extended to support the prediction of excited-state properties.^{112,221,222} These extensions involve constructing separate machine learning models for each electronic state of interest,^{112,221} enabling the prediction of state-specific quantities such as energies, gradients (and oscillator strengths). Supported kernel-based approaches include KREG, KRR with Coulomb matrix descriptors, SGML, and GAP-SOAP. Very recently, Dral *et al.*²²¹ utilized the relative-to-equilibrium (RE) representation¹⁸⁵ within the

MLatom framework to develop KRR-based KREG models capable of accurately predicting non-adiabatic coupling (NAC) vectors.

- PyRAI²MD.⁹ Another package that includes NNs for excited state properties is Python Rapid Artificial Intelligence *Ab Initio* Molecular Dynamics (PyRAI²MD).⁹ PyRAI²MD has enabled mechanistic understanding of complex photochemical organic transformations such as electrocyclizations, *E/Z*-isomerizations, and cycloadditions.^{16,180} It implements a fully-connected feed-forward NNs using inverse distance descriptors with multiple output nodes for predicting several excited-state energy and gradients at once. To avoid rapid growth of the inverse distance descriptors, it allows user to define the input distances between atoms pairs in their local atomic environments. It can also learn symmetric reaction pathways by defining the permutations of symmetric atoms, such as hexafluorobenzene.²²³

Besides the excited-state potential, PyRAI²MD provides a virtual potential model like SchNarc to predict the NAC vectors and a scalar output NNs for predicting the norm of SOCs for NAMD in the spin-diabatic representation. In addition, PyRAI²MD support both Zhu–Nakamura²²⁴ surface hopping and FSSH with curvature-driven time-dependent couplings, which run NAMD without using NACs. Recent updates of PyRAI²MD incorporated the NNs into the ONIOM approach with semi-empirical methods, like GFN2-xTB, which successfully revealed blocked non-radiative mechanisms in molecular aggregates²²⁵ and singlet fission mechanism in molecular crystals.²²⁶

- SchNarc^{20,135} & SpaiNN.²⁴ There are two software packages, SchNarc^{20,135} and SpaiNN,²⁴ that integrate SchNetPack 1.0 and 2.0, respectively, with SHARC 3.0 (ref. 44) and SHARC 4.0,⁴⁵ enabling ML-accelerated surface hopping simulations. SchNarc, interfaced with SchNetPack 1.0, relies on the rotationally invariant SchNet¹⁸¹ descriptor and, therefore, cannot directly predict vectorial properties. To overcome this, SchNarc applies a trick: it first predicts a “virtual” property and then derives the vectorial property by taking its derivative with respect to nuclear coordinates. This method allows SchNarc to predict properties for multiple singlet or triplet states, transition dipole moments, and spin-orbit couplings (SOCS). SpaiNN, a follow-up to SchNarc, integrates with SchNetPack 2.0 and takes advantage of the rotationally equivariant PaiNN¹³⁰ representation, enabling direct prediction of vectorial properties, including NACs. This direct approach improves prediction quality, as SpaiNN also combines scalar predictions, multiplying them with nuclear coordinates and adding the vectorial prediction to enhance the accuracy of the final NAC vectors.²⁴

- DANN.¹⁶⁹ Also building on the PaiNN¹³⁰ architecture, Axelrod *et al.*¹⁶⁹ have implemented the diabatic artificial neural network (DANN) model, to predict the quantum yields for the photoinduced *E/Z*-isomerization of azobenzene derivatives based on NAMD trajectories. This model implicitly incorporates diabatization into the NN architecture. This diabatization is used to ease the fitting of adiabatic states across chemical space. In particular, it addresses the issue of gap overestimation near conical intersections of unseen species.^{17,22,110,113,135}



Since diabatic energies are smooth even at conical intersections (unlike adiabatic energies, which exhibit non-differentiable cusps), they are easier to approximate using ML models (see also Section 3.3.2). DANN enforces this smoothness through a specialized loss function related to the NAC vector. This loss penalizes the magnitude of the NAC vector after rotation from the adiabatic to the diabatic basis and includes contributions from the NAC forces and a phase-correction term for each geometry. In addition to standard energy and force loss terms, DANN includes a gap error loss and an application-specific term that discriminates between *E*- and *Z*-isomers of azobenzene. The latter is implemented using the root-mean-square deviation (RMSD) between the input geometry and its aligned equilibrium structure.

- FieldSchNarc²⁷ and Field-MACE.²²⁷ FieldSchNarc,²⁷ built upon FieldSchNet²²⁰ for ground-state simulations, and Field-MACE²²⁷ both enable the inclusion of environmental effects in ML-driven photodynamics simulations using a hybrid quantum mechanics/molecular mechanics (QM/MM) approach. In QM/MM simulations, the system is divided into two distinct regions: a QM region and an MM region. The QM region is treated using high-level quantum chemical methods or, in this context, ML potentials, while the MM region is described by classical force fields. This setup facilitates computationally efficient simulations of large systems or solvated molecules, preserving high accuracy where it matters most.

Specifically, FieldSchNarc incorporates environmental interactions by accepting additional inputs representing the electrostatic field derived from MM atom point charges. This allows the model to effectively respond to dynamic changes in the environment, accurately capturing electronic properties of the QM region influenced by adaptive MM surroundings. However, accurate ML potentials from FieldSchNarc require training data that include the same QM region embedded in diverse MM environments. In contrast, Field-MACE models environmental interactions using a multipole expansion integrated within the MACE¹⁷⁹ and X-MACE²⁶ architectures, tailored respectively for ground- and excited-state simulations. A notable advantage of Field-MACE is its ability to initialize parameters from foundational MACE models,^{228,229} significantly enhancing data efficiency. Nevertheless, a common drawback of employing either model type is the sparsity of QM/MM datasets, necessitating additional effort in data curation and preparation.²⁷

- MS-ANI. The multi-state ANI (MS-ANI) architecture¹¹² was recently introduced. MS-ANI employs separate NNs for each atom type (Z_i), where atom-wise energy contributions are summed to yield the total molecular energy for a given state E_i . The adiabatic electronic state is included in the input for each element-wise NN. Notably, the number of electronic states is decoupled from the network architecture, allowing MS-ANI to handle an arbitrary number of states. This flexibility is achieved by incorporating the state index (i) into the model's input features alongside geometric descriptors. The state information is passed through all hidden and output layers of the network, enabling the model to distinguish between different states during training. This design not only allows electronic state information to propagate through the entire network, but also

facilitates training on incomplete datasets—such as those containing molecules with varying numbers of labelled electronic states. The MS-ANI architecture was recently employed in the OMNI-P2x,²²² a universal excited-state neural network potential that enabled real-time photodynamical simulations and rational design of the visible-light-absorbing azobenzene systems.²²²

- Excited-state MACE versions.^{26,177} Recently, MACE has been used to predict multiple potential energy surfaces and improves on the prediction of regions around conical intersections.¹⁷⁷ The architecture is designed to reconstruct intersecting energy surfaces from value-sorted data using smooth invariants. It involves approximating elementary symmetric polynomials or related invariants, which remain smooth despite surface intersections causing cusps. The original surfaces are recovered by solving a polynomial root-finding problem through companion matrices, specifically Frobenius, Schmeisser (symmetric tridiagonal), or Chebyshev colleague matrices. Each method has distinct numerical properties, advantages, and sensitivity to perturbations, influencing their stability and accuracy. This method has been shown to improve accuracy in regions where electronic states are close to each other and has been tested in the prediction of valence and conduction bands of graphene and electronically excited states of organic molecules.

In addition, X-MACE²⁶ combines the MACE framework with a DeepSets autoencoder, enabling smooth representation of inherently non-smooth excited-state surfaces. The model encodes adiabatic energies into permutationally invariant functions, reconstructing them through a Hermitian companion matrix decoder to ensure physically meaningful, real-valued eigenvalues. X-MACE significantly improves prediction accuracy for excited-state energies, forces, and non-adiabatic couplings compared to existing models, and notably allows transfer learning from foundational ground-state models to excited states, enhancing data efficiency and generalization to previously unseen molecular systems.

4.4 Single-state *vs.* multi-state *vs.* multi-output models

A major obstacle to the development of ML approaches for NAMD is the accurate representation of multiple PESs, particularly in regions where these surfaces are closely spaced or strongly coupled. Capturing the full multidimensional PES manifold, along with the complex interstate correlations, is essential to achieve reliable and transferable ML-accelerated NAMD simulations.

Early efforts predominantly employed single-state ML models, wherein independent models were trained for each electronic state.^{112,134} While conceptually straightforward, this approach inherently disregards correlations among electronic states, thereby limiting its efficacy in describing interstate couplings and correlated dynamics.

To address the limitations of single-state models, multi-output architectures have been developed.^{22,24,134,169} In this approach, a single neural network is trained to predict multiple potential energy surfaces (PESs) simultaneously. The model employs shared hidden layers and a final output layer with one



neuron per electronic state.²⁴ This shared representation allows the network to capture inter-state correlations implicitly, often leading to improved accuracy in surface hopping simulations compared to separate, state-specific models.^{134,135}

However, despite these advantages, such models can still struggle in scenarios involving unseen species, particularly near conical intersections, where subtle differences in energy gaps are crucial. As a result, these models may overestimate energy gaps and fail to consistently deliver better predictions across all systems.^{17,22,110,113,135,169}

More recently, multi-state models have been proposed, which incorporate the electronic state index (*e.g.*, 0 for the ground state, 1 for the first excited state, *etc.*) as an additional input feature to the network.^{112,222} In this architecture, hidden and output neurons process state-specific information, enabling a unified treatment of all electronic states. This approach offers increased flexibility relative to multi-output models, as it allows for the seamless handling of an arbitrary number of states. Furthermore, it alleviates limitations associated with loss function scaling in multi-output architectures, which may affect the quality of predicted coupling properties.²⁴

Another effective strategy for modeling excited-state potentials involves an internal ML-based diabatization scheme, where diagonalization is directly integrated within the model architecture. In this approach, the model predicts an internal ML-diabatic Hamiltonian matrix, which does not necessarily need to be physically meaningful and is subsequently diagonalized to obtain the adiabatic energies. The loss function explicitly considers these adiabatic energies. Several methods utilizing this strategy have been developed, consistently showing superior performance compared to models that directly predict adiabatic energies.^{87,169,230,231}

4.5 Phase-correction with ML

Generally, the training of excited-state ML potentials can be performed as for ground state potentials for energies and forces for which we refer the reader to ground-state ML potential overview articles (*e.g.* ref. 13, 128, and 232) but specific loss functions for phase properties. Fitting of excited state properties poses additional challenges besides fitting ground-state properties. More properties and states must be learned, and the arbitrary phase factor has to be considered. The arbitrary phase factor of the wave function makes coupling between different electronic states non-unique, implying an arbitrary sign of inter-state properties; the arbitrary sign²³³ must be corrected before or during training.

Pre-processing of photodynamics data is typically cumbersome; accounting for the phase factor during training is advised. To avoid explicit phase alignment during preprocessing, phase-free (or phaseless) loss functions have been developed and are commonly used in training NNs potentials for excited-state properties.^{22,24,135,169}

The key idea behind phase-free training is to incorporate phase correction directly into the loss function L_p . One approach evaluates the loss for all possible sign combinations of the N coupled electronic states, *i.e.*, 2^{N-1} permutations, and

selects the combination with the lowest error relative to the target.¹³⁵ However, this method scales exponentially as $O(2^N)$, making it computationally expensive for systems with many states.

To address this, more efficient alternatives treat each coupling vector independently by evaluating only two sign options per coupling, multiplying it by +1 and -1, and selecting the one with the lower loss.^{22,24,169} This reduces the computational complexity to linear scaling, $O(N)$, while maintaining predictive accuracy.²⁴

By selecting the sign combination that yields the lowest fitting error, this method effectively performs implicit phase correction, streamlining the training process without sacrificing performance.

4.6 ML quantum dynamics

The ML tools described above are being developed primarily to help accelerate and enable the running and analysis of NAMD using trajectory-based methods such as TSH. Full quantum dynamics methods that propagate a delocalized wavepacket also stand to gain much from these techniques. However, complications resulting from their computationally intensive nature point towards slower advances in this subfield.

Full quantum dynamics methods solve the time-dependent Schrödinger equation by propagating the complete wavepacket on a grid. The Multi-configuration Time-Dependent Hartree (MCTDH) method is an example of this approach.³⁴ Unlike trajectory-based methods, however, global surfaces are needed. A diabatic representation is also required (Section 3.3.3). While the surfaces only need to be accurately described in the region of space occupied by the wavepacket, their delocalized nature means that it is sensitive to “holes” in the potential, *i.e.*, regions where the potential drops to large negative energies due to poor fitting. This is a likely problem, if sufficient data is not used when learning surfaces, particularly at the boundaries of the space to be covered.

One can provide the data samples by using the surface hopping data. However, the classical nature of these trajectories may mean that they do not sample the full space needed to describe the wavepacket. Gaussian wavepacket methods, such as Multiple Spawning³⁸ or vMCG,²³⁴ may provide the answer by collecting data along trajectories that can be related directly to the wavepacket motion. This is in particular for the case with vMCG where the trajectories are not classical and spread faster to cover the needed phase space. vMCG has been shown to provide the points for Shepard Interpolated surfaces during direct dynamics simulations^{173,235,236} and use more advanced sampling techniques in the GROW methodology.²³⁷ More recently, it has been used with Gaussian Process Regression to learn the potentials for MCTDH calculations.²³⁸ However, the latter simulations also showed the challenges as many points were required for accurate surfaces.

ML techniques can take a complicated multi-dimensional potential function and put it in the form required for efficient quantum dynamics simulations. An ongoing challenge is the high computational cost of multi-dimensional integrals; for



efficiency the potential must be in a “sum-of-products” form. The sum-of-products formalism is written as a sum of terms comprising low-dimensional functions multiplied together. NNs are suitable for this,^{239,240} but have only been applied to small systems.

More and more research is directed towards post-processing (Section 5). Unlike surface hopping, which gives easily visualizable results with chemical structures moving along trajectories, wavepackets have all the information in a complex multi-dimensional function. Extracting trajectories of molecules evolving in time across excited- and ground-state PESs would help to create unprecedented knowledge of molecular excited states. Sampling and clustering techniques (see Sections 5.3 and 5.4) to find the significant regions are useful, but visualizing correlations is needed to demonstrate how wavepacket components evolve coherently.

4.7 Transferability and generalization

Ideally, a pre-trained machine learning (ML) model would generalize well enough to be reused for NAMD simulations of different molecular systems. However, achieving such transferability across chemical compound space remains a significant challenge for excited-state ML potentials.

Unlike classical force fields, widely used for ground-state dynamics, ML models for excited states often require retraining for each specific system. Global representations, though efficient for predicting properties relevant to NAMD, tend to generalize poorly across diverse chemical structures compared to local descriptors. Even state-of-the-art models frequently struggle to deliver accurate predictions when trained on a single system.²¹

While many ML architectures are in principle transferable, excited-state coupling properties, such as NACs, are highly sensitive and system-specific, further limiting generalization. Nonetheless, recent work has demonstrated promising progress: models trained on multiple molecules and conformations have been applied to UV/vis spectra prediction²⁴¹ and excited-state dynamics.¹⁶⁹

Notable are the OMNI-P2x model by Dral *et al.* (2025)²²² and X-MACE by Westermayr and co-workers,²⁶ which are the first universal ML potentials capable of predicting excited-state energies, forces, and other properties across a diverse set of organic molecules. The models are tested on out-of-the-box ML-driven NAMD simulations and allow for screening of derivatives of organic chromophores.

5 Extracting insights: post-processing and analysing NAMD trajectories

ML-driven excited-state molecular dynamics have revolutionized photochemistry by enabling the simulation of thousands of trajectories over nanosecond timescales. Consequently, the analysis of the resulting data has transformed into a big data problem, surpassing the capabilities of traditional visual inspection and trajectory analysis methods. Novel techniques are required to comprehensively analyze the data and extract

pertinent information and more and more often, unsupervised learning is used to assist analysis.^{242,243} We describe best practices for investigating ML- and QM non-adiabatic trajectories, incorporating various recent approaches in the literature. To illustrate these techniques, we will present examples involving the analysis of TSH dynamics data for the methylenimmonium cation.¹⁷

The section will be divided into four parts: prior considerations, dimensionality reduction, and clustering analyses of static data, which involves analyzing all the data simultaneously, considering each point individually, and dynamics analysis, which focuses on analyzing data as a time series. We will not delve into the analysis of individual trajectories, as comparisons to experimental data necessitate the examination of statistical averages rather than individual events.

An example Jupyter notebook for the analysis of trajectory data from excited-state calculations can be found at the following link: https://colab.research.google.com/drive/14W3zqhvSjHxUkSM_JrSbQiU_G6cSBmQK.

5.1 Prerequisites and preparative considerations

Before diving into the details of excited-state trajectory data analysis using ML-based techniques, we want to highlight the importance of the data quality. Especially if the NAMD is based on ML potentials, the trajectories might sometimes be error-prone. We urge users to search for undefined or missing values and outliers. For example, if an NVE ensemble is used, the user should check for validating constant total energy during each trajectory simulation (Fig. 12).

The choice of features included, and the representation are crucial for the outcome of any ML method, as already emphasized in Section 4.1. For postprocessing analyses, this depends on – and is limited by – the NAMD data aspects, the research question, and the extent of knowledge about the chemical problem. Many possible representations exist for geometrical

POST-PROCESSING

General Aspects

- Check for missing data and outliers before processing.
- Select relevant features and choose an appropriate data representation for analysis.
- Prioritize interpretable representations.
- Scale data to prevent biases, especially when combining different quantities like atomic distances and angles.

Algorithm Selection for NAMD Data Analysis

- Apply dimensionality reduction (e.g., PCA, MDS) to extract key features and reduce complexity.
- Use clustering algorithms (e.g., DBSCAN, agglomerative clustering) to identify common patterns in phase space.
- Perform time-series clustering to analyze dynamical behavior and identify representative trajectories.
- Visualize results to facilitate interpretation and insight extraction.

Fig. 12 Checklist for post-processing in ML-accelerated NAMD simulations: data quality checks and analysis of NAMD results.



information. However, to maintain interpretability, intuitive representations and features are chosen, such as the pairwise distances,²⁴⁴ any combination of interatomic distances, angles, and dihedrals,²⁴⁵ based on the pairwise dissimilarity matrix between two configurations,²⁴⁶ or as normal mode coordinates representations.^{247,248}

The data should be normalized or scaled to avoid biases from different descriptors. The possibilities depend on the nature of the data and include a mere shift of the mean to zero, min–max normalization (subtraction of the minimum followed by dividing by the maximum–minimum difference), z-score scaling (subtraction of mean followed by division by standard deviation), and the division by the respective quantities of a reference data point (*e.g.*, global minimum structure).

5.2 Dimensionality reduction

Molecular data is complex because of the substantial number of degrees of freedom; it cannot be easily represented in one or two dimensions for visualization. Reducing the data's dimensionality becomes crucial while retaining its fundamental structure. Dimensionality reduction techniques are invaluable in such scenarios, because they address the challenges posed by the high dimensional data and aid in visualization, exploration, and interpretation.

Principal Component Analysis (PCA) is one of the most widely used linear dimensionality reduction methods. It transforms the original high-dimensional data into a lower-dimensional space by identifying the directions of maximum variance in the data. The obtained set of uncorrelated variables is known as the principal components. The input to PCA is standardized descriptors that are pre-defined geometric and/or property-based quantities that represent the molecular system under investigation. The covariance matrix of the input features is computed to capture the correlation between the variables. This matrix is decomposed into its eigenvectors and eigenvalues, where the eigenvectors with the largest eigenvalues correspond to the principal components containing most of the data's variance. These principal components determine the new coordinate system for the transformed data. The first few main components will ideally include most of the variance in the data, enabling dimensionality reduction to visualize the data in a few dimensions. This is most effective on jointly Gaussian-distributed data because no correlation between components implies independence.²⁴⁹

PCA enables the interpretation of the transformed features by investigating how much each pre-defined descriptor contributes to each principal component, revealing the relative importance of each feature. Kernel PCA is an extension of PCA that allows for nonlinear dimensionality reduction. It uses a kernel trick technique to implicitly map the data into a higher-dimensional feature space, where linear PCA is performed. By utilizing a nonlinear mapping, kernel PCA can capture more complex and nonlinear patterns in the data.²⁵⁰

Multidimensional scaling (MDS) is another common category of nonlinear dimensionality reduction methods, especially useful for visualisation.²⁵¹ MDS encompasses several algorithms

reconstructing a low-dimensional spatial representation while preserving given pairwise distances or dissimilarities among data. Unlike other methods, MDS can handle any dissimilarity measure, making it applicable to a broad range of data types in chemistry, such as bond distances, angles, or torsional angles. The primary goal of MDS is to create a map with coordinates in a lower-dimensional space (usually 2 or 3 dimensions) that reflects the pairwise dissimilarities among the data points. These coordinates place each data point in the new space in such a way that the Euclidean distance between them closely matches the original dissimilarity measure.

The algorithms most frequently applied are classical MDS, metric MDS, and non-metric MDS. Classical MDS is particularly useful when the original dissimilarities are based on metric measurements, such as bond lengths or angles. Metric MDS is designed to handle metric dissimilarities explicitly, ensuring that the distances in the reconstructed map satisfy the properties of a metric space. It is a valuable approach when the dissimilarity measure is derived from real distances with a consistent scale. Non-metric MDS can accommodate non-metric dissimilarities, which may not satisfy the triangle inequality. Non-metric MDS is often employed when direct metric measurements are unavailable and only ordinal relationships between molecules are known.²⁵²

Another nonlinear dimensionality reduction technique is isomap, which focuses on preserving the intrinsic geometry and manifold structure of the data. It leverages the concept of geodesic distances, measuring the shortest path along the manifold between data points. The critical steps in isomap include the construction of neighborhood graphs, computing the geodesic distances between data points connected *via* the neighborhood graph, and embedding the data into a lower-dimensional space. The latter transformation uses MDS to preserve the geodesic distances as much as possible. Isomap is particularly useful for datasets with nontrivial geometric or topological properties that are difficult to analyze with linear methods.^{253,254}

Other popular nonlinear dimensionality reduction techniques are t-SNE (t-Distributed Stochastic Neighbor Embedding) and UMAP (Uniform Manifold Approximation and Projection). t-SNE captures local structure and similarities between data points by constructing a probability distribution in the high-dimensional space, thus representing pairwise similarities between data points and a corresponding distribution in the low-dimensional space. It is especially effective at preserving local structures, enabling cluster visualization for identifying patterns within complex molecular datasets.²⁵⁵

UMAP is another nonlinear technique that focuses on preserving the global structure and connectivity of the data. It constructs a high-dimensional topological representation of the data and then optimizes a low-dimensional representation that preserves the topological relationships. UMAP is based on manifold learning and is particularly useful for capturing local and global structures in high-dimensional data. It can handle large datasets more efficiently than t-SNE while producing similar embedding results.²⁴⁹



5.3 Pattern recognition and point-based data clustering

Pattern recognition encompasses a broad array of tools designed to identify patterns within data and group them accordingly. These tools can be categorized into clustering and classification methods. Classification techniques are part of supervised learning, where labelled data is available, allowing data points to be assigned to predefined classes. In contrast, clustering techniques belong to unsupervised learning, where data points are grouped into clusters based on their inherent similarities. The key distinction lies in the presence or absence of labels, making unsupervised learning approaches, particularly clustering, well-suited for analysing molecular dynamics simulations where labels are typically unavailable. Labelling data before analysis is often impractical with large and complex datasets. Consequently, unsupervised techniques offer a more straightforward starting point for analysing and exploring patterns within dynamic data, making them the primary focus of this section. Clustering algorithms can thus uncover hidden patterns and structures within datasets.²⁴⁹

5.3.1 Selecting a proper clustering algorithm. The nature of the data, the desired outcome, and the specific problem factors should be considered when choosing a clustering method. We advise gaining a basic understanding of the data. This understanding can be obtained by examining the data's dimensionality, the flexibility of the system, and any assumptions about the expected processes found in the literature. Once some understanding of the data is established, the objective of the clustering analysis needs to be defined. We suggest considering the following questions: Are you aiming to identify outliers? Is the primary goal to reduce the data dimensionality? Do you try to discover natural groups representing structures near mechanistic critical points on potential energy surfaces or photo-products? Different clustering techniques may be selected depending on the case. We suggest comparing various clustering algorithms to make an informed decision. If the dataset is too large, using a randomly generated subset can be beneficial for comparing different algorithms.^{256–258}

K-means clustering partitions the data into a predefined number of clusters (*K*) by minimizing the within-cluster sum of squares. *K*-means clustering requires a distance metric and is suitable for continuous variables. It assumes clusters of similar sizes and spherical shapes.²⁵⁹

Hierarchical clustering, as the name implies, creates a hierarchical structure of clusters by merging or splitting clusters based on their similarity or dissimilarity. It does not require prior specification of the number of clusters and can handle both continuous and categorical variables.²⁰¹

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) identifies clusters based on regions of high density separated by regions of low density. It is suitable for data with irregularly shaped clusters and effectively handles noise and outliers.²⁶⁰

Gaussian Mixture Models (GMM) are useful when the number of clusters is difficult to determine. While it starts with an initial guess of the number of clusters, it provides suggestions for a suitable number. GMM assumes that the data points

are generated from a mixture of Gaussian distributions. It can estimate the distribution parameters and assign probabilities to data points belonging to each cluster.²⁶¹

We recommend exploring the scikit-learn library for more clustering techniques, which provides a comprehensive overview of different clustering methods and their use cases.²⁰¹ Note that experimenting with various methods of clustering or parameter settings can be highly valuable because there is usually no general solution. Still, many processes might lead to similar results or can provide complementary information. Selecting a clustering method may require iterative refinement based on the data's specific characteristics and the desired outcome.

5.3.2 Evaluating the performance of clustering. Assessing the performance of clustering algorithms presents distinct challenges due to the absence of labelled data in unsupervised learning and, thus, the absence of a metric to evaluate the performance. This paper explores various extrinsic and intrinsic evaluation measures to overcome this limitation.

The distinction can be made between extrinsic and intrinsic measures. Extrinsic measures rely on externally provided labels or known class assignments to evaluate clustering performance. Thus, they are often not practicable when clustering dynamics data of which no labels are accessible before analysis. Examples of extrinsic measures are the rand index (measuring the similarity between two clusters by comparing pairs of samples), mutual information (quantifying mutual dependence between clusters), *V*-measure (assessing homogeneity and completeness of clusters), or Folkes–Mallows score (determining the similarity between two clusters by comparing geometric means).

More practicable for our use case are intrinsic measures that assess clustering quality based solely on the characteristics of the clustering itself and the underlying data without requiring any external labels. They thus aim to capture how well the data points within each cluster are grouped and how distinct the clusters are from one another. Intrinsic measures include the Silhouette coefficient, Calinski–Harabasz index, and Davies–Bouldin index.

The Silhouette coefficient or Silhouette score evaluates the performance based on the distance of data points within clusters *a* (average intra-cluster distance) and the distance of different clusters *b* (average inter-cluster distance),²⁶² which reads as

$$\text{Silhouette score} = \frac{b - a}{\max(a, b)}. \quad (7)$$

The silhouette score ranges from -1 to 1 , where 1 means that clusters are well separated, 0 means no significant distance between clusters, and -1 implies that clusters are not assigned correctly.

The Calinski–Harabasz index uses the ratio of between-cluster variance to within-cluster variance. The higher the index, the better the clusters are defined. The Davies–Bouldin score measures the average similarity of a cluster with respect to its most similar cluster. Lower values indicate better clustering performance.^{201,263}



5.3.3 Cluster visualization and data extraction. After executing and evaluating the clustering, visualizing the results is needed to extract chemical insights. One common approach involves a scatter plot; two or three descriptors are used to plot the data in the feature space. The inputs to the scatter plot are features known to be important for the reaction under investigation, or high-dimensional inputs already transformed into lower-dimensional representation with dimensionality reduction techniques, as discussed in Section 5.2. Using colors, shapes, or markers to indicate the different clusters while plotting the data is useful. Scatter plots allow for an overview of the distribution and separation of clusters in the feature space. MDS is a very common technique to plot a low-dimensional representation of high-dimensional data. t-SNE and UMAP are also frequently applied to visualize high-dimensional molecular data as these methods aim to preserve the local structure of the data and reveal clusters and patterns in the molecular dataset.

Exploring molecular interactions and connectivity patterns can be achieved with network visualization techniques such as Cytoscape.²⁶⁴ Molecular data are represented as a network or graph, where molecules are nodes, and their relationships are edges. Clusters can be visualized as subgraphs or communities within the larger network.

Besides these visualization techniques, additionally inspecting the clusters visually is often inevitable. Therefore, cluster centroids can be computed and serve as representatives of each cluster, reducing the complexity and time required for the visual inspection.^{87,201}

5.4 Time series clustering

Analysing molecular dynamics data with time series techniques can provide valuable insights into the system's dynamic behaviour. Some of the most useful time series techniques include descriptive statistics, time plots, autocorrelation, and partial autocorrelation analysis. Descriptive statistics can help characterize the behaviour of molecular properties over time. Calculating statistics such as mean, standard deviation, or autocorrelation of molecular properties (e.g., bond lengths, angles, dihedral angles) can provide insights into their stability, fluctuations, and correlations.

Time plots are probably the most intuitive approach because they demonstrate how molecular properties evolve throughout the simulation. Plotting molecular properties against time allows for identifying trends, fluctuations, and potential patterns or events in the system. Autocorrelation analysis can reveal the presence of correlations or dependencies in molecular data over different time lags. It helps identify any time-dependent patterns or autocorrelated behaviour within the system. Fourier Transform applied to molecular dynamics data can identify dominant frequencies or periodicities in molecular motions. It helps uncover characteristic vibrational modes, collective motions, or oscillatory behaviour within the system.

Tslearn is a Python library designed for time series analysis.²⁶⁵ It provides many tools and algorithms for analysing, modelling, and visualizing time series data. Some of the key features and techniques offered by tslearn include time series

clustering and time series classification, which enable clustering and classification of time series into predefined classes or categories, respectively, time series forecasting, or time series distance metrics to allow for quantification of similarities and dissimilarities between time series sequences.

6 Perspectives and conclusion

ML for excited states is much less explored, than ML for ground-state molecular properties. While ML ground-state potential energy surfaces can be obtained with high accuracy, errors for fitting excited-state properties are generally higher. The benefit of ML potentials that give very low errors with respect to QM methods, as it is often discussed in ground-state simulations,^{128–130,266,267} provide record-setting timescales for NAMD simulations. The benefit of using equivariant features in ML for fitting potential energy surfaces and properties thereof is in its infancy. Much work is still needed to push this field forward and more benchmark data sets are required to allow for developing and testing new ML methods for excited states.²⁶⁸ Efforts in the future should thus be further devoted to data set creation in addition to ML development and testing.

A major challenge that remains to be solved is transferability. Notably, the number and character of electronic states within a certain energy range can vary dramatically even between seemingly similar molecules. ML offers the possibility to carry out many hundreds of NAMD trajectories that are needed to achieve statistically significant results. Therefore, active learning techniques are often indispensable when carrying out ML-driven NAMD. In addition, the high amount of data produced by ML-driven NAMD will likely require other advanced ML-based methods for postprocessing and analysis in the future.

Author contributions

All authors contributed to the discussion that led to this manuscript and the initial draft. SAL and JW oversaw the writing of the manuscript. Individual groups formed that prepared the initial draft of each section. CM, SAL, and RCO focused on Section 3. CM, ŠS, SM, GW and JL focused on Section 4. BB, MPJ, and JW focused on Section 5. All authors were involved in the revision process. CM and ŠS prepared the original figures. BB prepared the Jupyter notebook on unsupervised learning.

Conflicts of interest

There are no conflicts to declare.

Data availability

No primary research results, software or code have been included and no new data were generated or analysed as part of this review. An example Jupyter notebook for the analysis of trajectory data from excited-state calculations can be found at

the following link: https://colab.research.google.com/drive/14W3zqhvSjHxUkSM_JrSbQiU_G6cSBmQK.

Acknowledgements

ŠS gratefully acknowledges the support of Czech Science Foundation, project no. 22-13489O. RCO acknowledges funding from the Leverhulme Trust Leverhulme Trust (RPG-2019-122) and the Royal Society for funding through an International Exchange Grant (IES|R2|222057). SAL acknowledges funding from the U.S. National Science Foundation (NSF-CHE-2144556). MPJ acknowledges the financial support of the European Research Council (ERC) under the advanced grant SubNano (No. 832237). GW acknowledges financial support from the EPSRC under grant EP/X026973/1. BA and SM acknowledge funding from the University of Vienna in the frameworks of the research platform (ViRAPID) and the Doctoral School of Chemistry (DoSChem), respectively.

Notes and references

- 1 S. Gómez, I. F. Galván, R. Lindh and L. González, in *Motivation and Basic Concepts*, John Wiley & Sons, Ltd, 2020, ch. 1, pp. 1–12.
- 2 L. Cwiklik, M. Pederzoli, M. W. Baig, M. Kývala and J. J. Pittner, *Biophys. J.*, 2020, **118**, 79a.
- 3 S. Mathew, A. Yella, P. Gao, R. Humphry-Baker, B. F. E. Curchod, N. Ashari-Astani, I. Tavernelli, U. Rothlisberger, M. K. Nazeeruddin and M. Grätzel, *Nat. Chem.*, 2014, **6**, 242–247.
- 4 J. Westermayr, J. Gilkes, R. Barrett and R. J. Maurer, *Nat. Comput. Sci.*, 2023, **3**, 139–148.
- 5 S. Mai, P. Marquetand and L. González, in *Surface hopping molecular dynamics*, John Wiley & Sons, Ltd, 2020, pp. 499–530.
- 6 J. M. Toldo, M. T. do Casal, E. Ventura, S. A. do Monte and M. Barbatti, *Phys. Chem. Chem. Phys.*, 2023, **25**, 8293–8316.
- 7 J. Westermayr and P. Marquetand, *Machine Learning in Chemistry: The Impact of Artificial Intelligence*, The Royal Society of Chemistry, 2020, pp. 76–108.
- 8 J. Westermayr and P. Marquetand, *Chem. Rev.*, 2020, **121**, 9873–9926.
- 9 J. Li and S. A. Lopez, *Acc. Chem. Res.*, 2022, **55**, 1972–1984.
- 10 *Quantum Chemistry in the Age of Machine Learning*, ed. P. O. Dral, Elsevier, 1st edn, 2022.
- 11 C. Zhang, Y. Zhong, Z.-G. Tao, X. Qin, H. Shang, Z. Lan, O. V. Prezhdo, X.-G. Gong, W. Chu and H. Xiang, *Nat. Commun.*, 2025, **16**, 2033.
- 12 Editorials, *Nat. Synth.*, 2023, **2**, 459.
- 13 O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko and K.-R. Müller, *Chem. Rev.*, 2021, **121**, 10142.
- 14 P. O. Dral and M. Barbatti, *Nat. Rev. Chem.*, 2021, **5**, 388–405.
- 15 O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko and K.-R. Müller, *Chem. Rev.*, 2021, **121**, 10142–10186.
- 16 J. Li, P. Reiser, B. R. Boswell, A. Eberhard, N. Z. Burns, P. Friederich and S. A. Lopez, *Chem. Sci.*, 2021, **12**, 5302–5314.
- 17 J. Westermayr, M. Gastegger, M. F. S. J. Menger, S. Mai, L. González and P. Marquetand, *Chem. Sci.*, 2019, **10**, 8100–8107.
- 18 A. V. Akimov, *J. Phys. Chem. Lett.*, 2021, **12**, 12119–12128.
- 19 A. V. Akimov, *J. Phys. Chem. Lett.*, 2018, **9**, 6096–6102.
- 20 J. Westermayr, M. Gastegger and P. Marquetand, SchNarc, 2021, <https://github.com/schnarc/SchNarc>.
- 21 J. Westermayr and P. Marquetand, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 043001.
- 22 J. Li, P. Reiser, B. R. Boswell, A. Eberhard, N. Z. Burns, P. Friederich and S. A. Lopez, *Chem. Sci.*, 2021, **12**, 5302–5314.
- 23 C. Li, S. Hou and C. Xie, *J. Chem. Theory Comput.*, 2023, **19**, 3063–3079.
- 24 S. Mausenberger, C. Müller, A. Tkatchenko, P. Marquetand, L. González and J. Westermayr, *Chem. Sci.*, 2024, **15**, 15880–15890.
- 25 L. Zhang, S. V. Pios, M. Martyka, F. Ge, Y.-F. Hou, Y. Chen, L. Chen, J. Jankowska, M. Barbatti and P. O. Dral, *J. Chem. Theory Comput.*, 2024, **20**, 5043–5057.
- 26 R. Barrett, C. Ortner and J. Westermayr, *arXiv*, 2025, arXiv:2502.12870, DOI: [10.48550/arXiv.2502.12870](https://doi.org/10.48550/arXiv.2502.12870).
- 27 M. X. Tiefenbacher, B. Bachmair, C. G. Chen, J. Westermayr, P. Marquetand, J. C. B. Dietschreit and L. González, *Digital Discovery*, 2025, **4**, 1478–1491.
- 28 B. F. E. Curchod and T. J. Martínez, *Chem. Rev.*, 2018, **118**, 3305–3336.
- 29 R. Crespo-Otero and M. Barbatti, *Chem. Rev.*, 2018, **118**, 7026–7068.
- 30 T. R. Nelson, A. J. White, J. A. Bjorgaard, A. E. Sifain, Y. Zhang, B. Nebgen, S. Fernandez-Alberti, D. Mozyrsky, A. E. Roitberg and S. Tretiak, *Chem. Rev.*, 2020, **120**, 2215–2287.
- 31 E. Tapavicza, G. D. Bellchambers, J. C. Vincent and F. Furche, *Phys. Chem. Chem. Phys.*, 2013, **15**, 18336–18348.
- 32 L. L. E. Cigrang, B. F. E. Curchod, R. A. Ingle, A. Kelly, J. R. Mannouch, D. Accomasso, A. Alijah, M. Barbatti, W. Chebbi, N. Došlić, E. C. Eklund, S. Fernandez-Alberti, A. Freibert, L. González, G. Granucci, F. J. Hernández, J. Hernández-Rodríguez, A. Jain, J. Janoš, I. Kassal, A. Kirrander, Z. Lan, H. R. Larsson, D. Lauvergnat, B. Le Dé, Y. Lee, N. T. Maitra, S. K. Min, D. Peláez, D. Picconi, Z. Qiu, U. Raucci, P. Robertson, E. Sangiogo Gil, M. Sapunar, P. Schürger, P. Sinnott, S. Tretiak, A. Tikku, P. Vindel-Zandbergen, G. A. Worth, F. Agostini, S. Gómez, L. M. Ibele and A. Prlj, *J. Phys. Chem. A*, 2025, **129**(31), 7023–7050.
- 33 J. Li and S. A. Lopez, *Chem. Phys. Rev.*, 2023, **4**, 031309.
- 34 M. H. Beck, A. Jäckle, G. A. Worth and H. D. Meyer, *Phys. Rep.*, 2000, **324**, 1–105.
- 35 G. A. Worth and B. Lasorne, *Quantum Chemistry and Dynamics of Excited States*, John Wiley & Sons Ltd, 2020, pp. 413–433.



36 B. F. E. Curchod and T. J. Martínez, *Chem. Rev.*, 2018, **118**, 3305–3336.

37 B. G. Levine, J. D. Coe, A. M. Virshup and T. J. Martínez, *Chem. Phys.*, 2008, **347**, 3–16.

38 M. Ben-Nun and T. J. Martínez, *Chem. Phys. Lett.*, 1998, **298**, 57–65.

39 A. Kirrander and M. Vacher, *Quantum Chemistry and Dynamics of Excited States*, John Wiley & Sons Ltd, 2020, pp. 469–497.

40 J. C. Tully, *Faraday Discuss.*, 1998, **110**, 407–419.

41 J. C. Tully, *J. Chem. Phys.*, 2012, **137**, 22A301.

42 J. C. Tully, *Catal. Lett.*, 1991, **9**, 205–217.

43 F. Plasser, S. Gómez, M. F. S. J. Menger, S. Mai and L. González, *Phys. Chem. Chem. Phys.*, 2019, **21**, 57–69.

44 S. Mai, P. Marquetand and L. González, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2018, **8**, e1370.

45 S. Mai, B. Bachmair, L. Gagliardi, H. G. Gallmetzer, L. Grünwald, M. Hennefarth, N. Machholdt Hoyer, F. A. Korsaye, S. Mausenberger, M. Oppel, T. Piteša, S. Polonius, E. Sangiogo Gil, Y. Shu, N. K. Singer, M. X. Tiefenbacher, D. Truhlar, D. Vörös, L. Zhang and L. Gonzalez, sharec-md/sharec4: SHARC Release 4.0, 2025, DOI: [10.5281/zenodo.15496427](https://doi.org/10.5281/zenodo.15496427).

46 M. Barbatti, M. Ruckenbauer, F. Plasser, J. Pittner, G. Granucci, M. Persico and H. Lischka, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2014, **4**, 26–33.

47 M. Barbatti, M. Bondanza, R. Crespo-Otero, B. Demoulin, P. O. Dral, G. Granucci, F. Koskoski, H. Lischka, B. Mennucci, S. Mukherjee, M. Pederzoli, M. Persico, M. Pinheiro Jr, J. Pittner, F. Plasser, E. Sangiogo Gil and L. Stojanović, *J. Chem. Theory Comput.*, 2022, **18**, 6851–6865.

48 L. Du and Z. Lan, *J. Chem. Theory Comput.*, 2015, **11**, 1360–1374.

49 F. Plasser, G. Granucci, J. Pittner, M. Barbatti, M. Persico and H. Lischka, *J. Chem. Phys.*, 2012, **137**, 22A514.

50 S. Mai, P. Marquetand and L. González, *Int. J. Quantum Chem.*, 2015, **115**, 1215–1231.

51 G. Cui and W. Thiel, *J. Chem. Phys.*, 2014, **141**, 124101.

52 H. M. Jaeger, S. Fischer and O. V. Prezhdo, *J. Chem. Phys.*, 2012, **137**, 22A545.

53 M. Persico and G. Granucci, *Photochemistry: A modern theoretical perspective*, Springer, 2018.

54 S. Mai and L. González, *Angew. Chem., Int. Ed.*, 2020, **59**, 16832–16846.

55 T. V. Papineau, D. Jacquemin and M. Vacher, *J. Phys. Chem. Lett.*, 2024, **15**, 636–643.

56 J. Janoš and P. Sláviček, *J. Chem. Theory Comput.*, 2023, **19**, 8273–8284.

57 J. Westermayr, M. Gastegger, K. T. Schütt and R. J. Maurer, *J. Chem. Phys.*, 2021, **154**, 230903.

58 F. Plasser, R. Crespo-Otero, M. Pederzoli, J. Pittner, H. Lischka and M. Barbatti, *J. Chem. Theory Comput.*, 2014, **10**, 1395–1405.

59 M. Barbatti and R. Crespo-Otero, *Density-Functional Methods for Excited States*, Springer International Publishing, Cham, 2016, pp. 415–444.

60 M. Huix-Rotllant, N. Ferré and M. Barbatti, *Quantum Chemistry and Dynamics of Excited States*, John Wiley & Sons Ltd, 2020, pp. 13–46.

61 D. Casanova and A. I. Krylov, *Phys. Chem. Chem. Phys.*, 2020, **22**, 4326–4342.

62 D. Lefrancois, D. Tuna, T. J. Martínez and A. Dreuw, *J. Chem. Theory Comput.*, 2017, **13**, 4436–4441.

63 S. Lee, S. Shostak, M. Filatov and C. H. Choi, *J. Phys. Chem. A*, 2019, **123**, 6455–6462.

64 O. Christiansen, H. Koch and P. Jørgensen, *Chem. Phys. Lett.*, 1995, **243**, 409–418.

65 A. Dreuw and M. Wormit, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2015, **5**, 82–95.

66 B. G. Levine, C. Ko, J. Quenneville and T. J. Martínez, *Mol. Phys.*, 2006, **104**, 1039–1051.

67 M. Huix-Rotllant, A. Nikiforov, W. Thiel and M. Filatov, *Density-Functional Methods for Excited States*, Springer International Publishing, Cham, 2016, pp. 445–476.

68 L. Stojanović, S. Bai, J. Nagesh, A. F. Izmaylov, R. Crespo-Otero, H. Lischka and M. Barbatti, *Molecules*, 2016, **21**, 1603.

69 Y. Horbatenko, S. Sadiq, S. Lee, M. Filatov and C. H. Choi, *J. Chem. Theory Comput.*, 2021, **17**, 848–859.

70 I. S. Lee, M. Filatov and S. K. Min, *J. Chem. Theory Comput.*, 2019, **15**, 3021–3032.

71 W. Park, S. Lee, M. Huix-Rotllant, M. Filatov and C. H. Choi, *J. Phys. Chem. Lett.*, 2021, **12**, 4339–4346.

72 K. Andersson, P. A. Malmqvist, B. O. Roos, A. J. Sadlej and K. Wolinski, *J. Phys. Chem.*, 1990, **94**, 5483–5488.

73 C. Angeli, R. Cimiraglia and J.-P. Malrieu, *J. Chem. Phys.*, 2002, **117**, 9138–9153.

74 H. Lischka, D. Nachtigallová, A. J. A. Aquino, P. G. Szalay, F. Plasser, F. B. C. Machado and M. Barbatti, *Chem. Rev.*, 2018, **118**, 7293–7361.

75 P. G. Szalay, T. Müller, G. Gidofalvi, H. Lischka and R. Shepard, *Chem. Rev.*, 2012, **112**, 108–181.

76 C. David Sherrill and H. F. Schaefer, *Adv. Quant. Chem.*, 1999, **34**, 143–269.

77 T. Shiozaki, W. Győrffy, P. Celani and H.-J. Werner, *J. Chem. Phys.*, 2011, **135**, 081106.

78 S. Battaglia and R. Lindh, *J. Chem. Theory Comput.*, 2020, **16**, 1555–1567.

79 S. Battaglia and R. Lindh, *J. Chem. Phys.*, 2021, **154**, 034102.

80 D. S. Levine, D. Hait, N. M. Tubman, S. Lehtola, K. B. Whaley and M. Head-Gordon, *J. Chem. Theory Comput.*, 2020, **16**, 2340–2354.

81 J. W. Park, *J. Chem. Theory Comput.*, 2021, **17**, 4092–4104.

82 I. Polyak, L. Hutton, R. Crespo-Otero, M. Barbatti and P. J. Knowles, *J. Chem. Theory Comput.*, 2019, **15**, 3929–3940.

83 S. Gómez, L. M. Ibele and L. González, *Phys. Chem. Chem. Phys.*, 2019, **21**, 4871–4878.

84 G. Li Manni, R. K. Carlson, S. Luo, D. Ma, J. Olsen, D. G. Truhlar and L. Gagliardi, *J. Chem. Theory Comput.*, 2014, **10**, 3669–3680.

85 L. Gagliardi, D. G. Truhlar, G. Li Manni, R. K. Carlson, C. E. Hoyer and J. L. Bao, *Acc. Chem. Res.*, 2017, **50**, 66–73.



86 A. O. Lykhin, D. G. Truhlar and L. Gagliardi, *J. Am. Chem. Soc.*, 2021, **143**, 5878–5889.

87 J. Westermayr, M. Gastegger, D. Vörös, L. Panzenboeck, F. Joerg, L. González and P. Marquetand, *Nat. Chem.*, 2022, **14**, 914–919.

88 N. M. Kidwell, H. Li, X. Wang, J. M. Bowman and M. I. Lester, *Nat. Chem.*, 2016, **8**, 509–514.

89 K. Atalar, Y. Rath, R. Crespo-Otero and G. H. Booth, *Faraday Discuss.*, 2024, **254**, 542–569.

90 B. de Souza, *Angew. Chem., Int. Ed.*, 2025, e202500393.

91 S. Grimme, *J. Chem. Theory Comput.*, 2019, **15**, 2847–2862.

92 P. Pracht, S. Grimme, C. Bannwarth, F. Bohle, S. Ehlert, G. Feldmann, J. Gorges, M. Müller, T. Neudecker, C. Plett, S. Spicher, P. Steinbach, P. A. Wesolowski and F. Zeller, *J. Chem. Phys.*, 2024, **160**, 114110.

93 J. Steinmetzer, S. Kupfer and S. Gräfe, *Int. J. Quantum Chem.*, 2021, **121**, e26390.

94 J. Sanz García, R. Maskri, A. Mitrushchenkov and L. Joubert-Doriol, *J. Chem. Theory Comput.*, 2024, **20**, 5643–5654.

95 P. Pracht and C. Bannwarth, *J. Chem. Theory Comput.*, 2022, **18**, 6370–6385.

96 E. Vandaele, M. Mališ and S. Luber, *J. Chem. Theory Comput.*, 2024, **20**, 856–872.

97 T. W. Keal, A. Koslowski and W. Thiel, *Theor. Chem. Acc.*, 2007, **118**, 837–844.

98 I. F. Galvan, M. G. Delcey, T. B. Pedersen, F. Aquilante and R. Lindh, *J. Chem. Theory Comput.*, 2016, **12**, 3636–3653.

99 M. Pinheiro Jr, S. Zhang, P. O. Dral and M. Barbatti, *Sci. Data*, 2023, **10**, 95.

100 J. Suchan, D. Hollas, B. F. E. Curchod and P. Slavíček, *Faraday Discuss.*, 2018, **212**, 307–330.

101 R. Crespo-Otero and M. Barbatti, *Theor. Chem. Acc.*, 2012, **131**, 7026–7068.

102 Š. Sršen, J. Sita, P. Slavíček, V. Ladányi and D. Heger, *J. Chem. Theory Comput.*, 2020, **16**, 6428–6438.

103 M. Barbatti and K. Sen, *Int. J. Quantum Chem.*, 2016, **116**, 762–771.

104 M. Ceriotti, G. Bussi and M. Parrinello, *J. Chem. Theory Comput.*, 2010, **6**, 1170–1180.

105 J. P. Zobel, J. J. Nogueira and L. González, *Phys. Chem. Chem. Phys.*, 2019, **21**, 13906–13915.

106 C. Li and G. A. Voth, *J. Chem. Theory Comput.*, 2022, **18**, 599–604.

107 S. C. Althorpe, *Eur. Phys. J. B*, 2021, **94**, 155.

108 M. Ceriotti, D. E. Manolopoulos and M. Parrinello, *J. Chem. Phys.*, 2011, **134**, 84104.

109 S. V. Pios, M. F. Gelin, A. Ullah, P. O. Dral and L. Chen, *J. Phys. Chem. Lett.*, 2024, **15**, 2325–2331.

110 W.-K. Chen, X.-Y. Liu, W.-H. Fang, P. O. Dral and G. Cui, *J. Phys. Chem. Lett.*, 2018, **9**, 6702–6708.

111 D. Hu, Y. Xie, X. Li, L. Li and Z. Lan, *J. Phys. Chem. Lett.*, 2018, **9**, 2725–2732.

112 M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti and P. O. Dral, *npj Comput. Mater.*, 2025, **11**, 1–12.

113 P. O. Dral, M. Barbatti and W. Thiel, *J. Phys. Chem. Lett.*, 2018, **9**, 5660–5663.

114 J. Li, R. Stein, D. M. Adrián and S. A. Lopez, *J. Am. Chem. Soc.*, 2021, **143**, 20166–20175.

115 J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev and A. E. Roitberg, *J. Chem. Phys.*, 2018, **148**, 241733.

116 J. Behler, *Int. J. Quantum Chem.*, 2015, **115**, 1032–1050.

117 D. Hu, Y. Xie, X. Li, L. Li and Z. Lan, *J. Phys. Chem. Lett.*, 2018, **9**, 2725–2732.

118 L. Zhang, Y.-F. Hou, F. Ge and P. O. Dral, *Phys. Chem. Chem. Phys.*, 2023, **25**, 23467–23476.

119 J. Behler, *J. Phys.: Condens. Matter*, 2014, **26**, 183001.

120 F. Musil, M. J. Willatt, M. A. Langovoy and M. Ceriotti, *J. Chem. Theory Comput.*, 2019, **15**, 906–915.

121 M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.*, *Inf. Fusion*, 2021, **76**, 243–297.

122 J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, *et al.*, *Artif. Intell. Rev.*, 2023, **56**, 1513–1589.

123 V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti and G. Csányi, *Chem. Rev.*, 2021, **121**, 10073–10141.

124 M. Kellner and M. Ceriotti, *Mach. Learn.: Sci. Technol.*, 2024, **5**, 035006.

125 F. Bigi, S. Chong, M. Ceriotti and F. Grasselli, *Mach. Learn.: Sci. Technol.*, 2024, **5**, 045018.

126 S. Chong, F. Bigi, F. Grasselli, P. Loche, M. Kellner and M. Ceriotti, *Faraday Discuss.*, 2025, **256**, 322–344.

127 S. Mukherjee and M. Barbatti, *Res. Chem.*, 2022, **4**, 100521.

128 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt and B. Kozinsky, *Nat. Commun.*, 2022, **13**, 2453.

129 I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. Simm, R. Drautz, C. Ortner, B. Kozinsky and G. Csányi, *arXiv*, 2022, arXiv:2205.06643, DOI: [10.48550/arXiv.2205.06643](https://doi.org/10.48550/arXiv.2205.06643).

130 K. Schütt, O. Unke and M. Gastegger, *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 9377–9388.

131 A. S. Christensen and O. A. von Lilienfeld, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 045018.

132 N. Artrith, K. T. Butler, F.-X. Coudert, S. Han, O. Isayev, A. Jain and A. Walsh, *Nat. Chem.*, 2021, **13**, 505–508.

133 *Conical Intersections: Electronic Structure, Dynamics & Spectroscopy*, ed. W. Domcke, D. Yarkony and H. Köppel, World Scientific, 2004.

134 J. Westermayr, F. A. Faber, A. S. Christensen, O. A. von Lilienfeld and P. Marquetand, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 025009.

135 J. Westermayr, M. Gastegger and P. Marquetand, *J. Phys. Chem. Lett.*, 2020, **11**, 3828–3834.

136 J. O. Richardson, *J. Chem. Phys.*, 2023, **158**, 011102.

137 H. An and K. K. Baeck, *Chem. Phys. Lett.*, 2018, **696**, 100–105.

138 K. K. Baeck and H. An, *J. Chem. Phys.*, 2017, **146**, 064107.



139 M. T. do Casal, J. M. Toldo, M. P. Jr and M. Barbatti, *Open Res. Europe*, 2021, **1**, 49.

140 Y. Shu, L. Zhang, X. Chen, S. Sun, Y. Huang and D. G. Truhlar, *J. Chem. Theory Comput.*, 2022, **18**, 1320–1328.

141 Y. Shu, L. Zhang, X. Chen, S. Sun, Y. Huang and D. G. Truhlar, *J. Chem. Theory Comput.*, 2022, **18**, 1320–1328.

142 L. Landau, *Phys. Z. Sowjetunion*, 1932, **2**, 46–51.

143 C. Zener and R. H. Fowler, *Proc. R. Soc. London, Ser. A*, 1932, **137**, 696–702.

144 J. Suchan, J. Janoš and P. Slavíček, *J. Chem. Theory Comput.*, 2020, **16**, 5809–5820.

145 T. Ishida, S. Nanbu and H. Nakamura, *Int. Rev. Phys. Chem.*, 2017, **36**, 229–285.

146 L. Yu, C. Xu, Y. Lei, C. Zhu and Z. Wen, *Phys. Chem. Chem. Phys.*, 2014, **16**, 25883–25895.

147 D. Shchepanovska, R. J. Shannon, B. F. E. Curchod and D. R. Glowacki, *J. Phys. Chem. A*, 2021, **125**, 3473–3488.

148 L. Yue, L. Yu, C. Xu, Y. Lei, Y. Liu and C. Zhu, *ChemPhysChem*, 2017, **18**, 1274–1287.

149 L. Yue, L. Yu, C. Xu, C. Zhu and Y. Liu, *Phys. Chem. Chem. Phys.*, 2020, **22**, 11440–11451.

150 C. A. Mead and D. G. Truhlar, *J. Chem. Phys.*, 1982, **77**, 6090–6098.

151 Y. Shu, Z. Varga, S. Kanchanakungwankul, L. Zhang and D. G. Truhlar, *J. Phys. Chem. A*, 2022, **126**, 992–1018.

152 X. Zhu and D. R. Yarkony, *J. Chem. Phys.*, 2014, **140**, 024112.

153 Y. Shen and D. R. Yarkony, *J. Phys. Chem. A*, 2020, **124**, 4539–4548.

154 R. Abrol and A. Kuppermann, *J. Chem. Phys.*, 2002, **116**, 1035–1062.

155 H. Köppel, *Faraday Discuss.*, 2004, **127**, 35–47.

156 X. Zhu and D. R. Yarkony, *J. Chem. Phys.*, 2010, **132**, 104101.

157 Z. Varga, K. A. Parker and D. G. Truhlar, *Phys. Chem. Chem. Phys.*, 2018, **20**, 26643–26659.

158 G. J. Atchity and K. Ruedenberg, *Theor. Chem. Acc.*, 1997, **97**, 47–58.

159 T. Pacher, H. Köppel and L. S. Cederbaum, *J. Chem. Phys.*, 1991, **95**, 6668–6680.

160 K. R. Yang, X. Xu and D. G. Truhlar, *Chem. Phys. Lett.*, 2013, **573**, 84–89.

161 N. Wittenbrink, F. Venghaus, D. Williams and W. Eisfeld, *J. Chem. Phys.*, 2016, **145**, 184108.

162 J. E. Subotnik, S. Yeganeh, R. J. Cave and M. A. Ratner, *J. Chem. Phys.*, 2008, **129**, 244101.

163 C. E. Hoyer, K. Parker, L. Gagliardi and D. G. Truhlar, *J. Chem. Phys.*, 2016, **144**, 194101.

164 Š. Sršeň, O. A. von Lilienfeld and P. Slavíček, *Phys. Chem. Chem. Phys.*, 2024, **26**, 4306–4319.

165 D. M. Williams and W. Eisfeld, *J. Chem. Phys.*, 2018, **149**, 204106.

166 Y. Guan, D. H. Zhang, H. Guo and D. R. Yarkony, *Phys. Chem. Chem. Phys.*, 2019, **21**, 14205–14213.

167 Y. Shu and D. G. Truhlar, *J. Chem. Theory Comput.*, 2020, **16**, 6456–6464.

168 Y. Shu, Z. Varga, A. G. Sampaio De Oliveira-Filho and D. G. Truhlar, *J. Chem. Theory Comput.*, 2021, **17**, 1106–1116.

169 S. Axelrod, E. Shakhnovich and R. Gómez-Bombarelli, *Nat. Commun.*, 2022, **13**, 3440.

170 Y. Shu, J. Kryven, A. G. Sampaio de Oliveira-Filho, L. Zhang, G.-L. Song, S. L. Li, R. Meana-Pañeda, B. Fu, J. M. Bowman and D. G. Truhlar, *J. Chem. Phys.*, 2019, **151**, 104311.

171 Y. Guan, H. Guo and D. R. Yarkony, *J. Chem. Theory Comput.*, 2020, **16**, 302–313.

172 H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby and M. Schütz, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2012, **2**, 242–253.

173 G. Christopoulou, A. Freibert and G. A. Worth, *J. Chem. Phys.*, 2021, **154**, 124127.

174 D. Opalka and W. Domcke, *J. Chem. Phys.*, 2013, **138**, 224103.

175 T. Y. Wang, S. P. Neville and M. S. Schuurman, *J. Phys. Chem. Lett.*, 2023, **14**, 7780–7786.

176 Y. Shu, Z. Varga, A. M. Parameswaran and D. G. Truhlar, *J. Chem. Theory Comput.*, 2024, **20**, 7042–7051.

177 T. S. Gutleb, R. Barrett, J. Westermayr and C. Ortner, *arXiv*, 2024, arXiv:2407.03731, DOI: [10.48550/arXiv.2407.03731](https://doi.org/10.48550/arXiv.2407.03731).

178 M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov and A. J. Smola, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, 3391–3401.

179 I. Batatia, D. P. Kovacs, G. Simm, C. Ortner and G. Csányi, *Adv. Neural Inf. Process. Syst.*, 2022, 11423–11436.

180 J. Li, R. Stein, D. M. Adrián and S. A. Lopez, *J. Am. Chem. Soc.*, 2021, **143**, 20166–20175.

181 K. T. Schütt, H. E. Saucedo, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, *J. Chem. Phys.*, 2018, **148**, 241722.

182 F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi and M. Ceriotti, *Chem. Rev.*, 2021, **121**, 9759–9815.

183 F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi and M. Ceriotti, *Chem. Rev.*, 2021, **121**, 9759–9815.

184 K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko and K.-R. Müller, *J. Chem. Theory Comput.*, 2013, **9**, 3404–3419.

185 P. O. Dral, A. Owens, S. N. Yurchenko and W. Thiel, *J. Chem. Phys.*, 2017, **146**, 244108.

186 A. P. Bartók and G. Csányi, *Int. J. Quantum Chem.*, 2015, **115**, 1051–1057.

187 M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, *Phys. Rev. Lett.*, 2012, **108**, 058301.

188 S. Chmiela, H. E. Saucedo, I. Poltavsky, K. R. Müller and A. Tkatchenko, *Comput. Phys. Commun.*, 2019, **240**, 38–45.

189 Y.-F. Hou, F. Ge and P. O. Dral, *J. Chem. Theory Comput.*, 2023, **19**, 2369–2379.

190 J. Westermayr, F. A. Faber, A. S. Christensen, O. A. von Lilienfeld and P. Marquetand, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 025009.

191 M. Pinheiro, F. Ge, N. Ferré, P. O. Dral and M. Barbatti, *Chem. Sci.*, 2021, **12**, 14396–14413.

192 A. P. Bartók, R. Kondor and G. Csányi, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2013, **87**, 184115.



193 J. Behler, *J. Chem. Phys.*, 2011, **134**, 074106.

194 A. S. Christensen, L. A. Bratholm, F. A. Faber and O. Anatole von Lilienfeld, *J. Chem. Phys.*, 2020, **152**, 044107.

195 A. Wilson and H. Nickisch, *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1775–1784.

196 A. G. Wilson, Z. Hu, R. Salakhutdinov and E. P. Xing, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, Cadiz, Spain, 2016, pp. 370–378.

197 S. Manzhos and M. Ihara, *J. Phys. Chem. A*, 2023, **127**, 7823–7835.

198 M. Rupp, *Int. J. Quantum Chem.*, 2015, **115**, 1058–1073.

199 P. O. Dral, F. Ge, B.-X. Xue, Y.-F. Hou, M. Pinheiro Jr, J. Huang and M. Barbatti, *Top. Curr. Chem.*, 2021, **379**, 27.

200 M. Rupp, O. A. von Lilienfeld and K. Burke, *J. Chem. Phys.*, 2018, **148**, 241401.

201 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.

202 H. Huo and M. Rupp, *Mach. Learn.: Sci. Technol.*, 2022, **3**, 045017.

203 P. O. Dral, *J. Comput. Chem.*, 2019, **40**, 2339–2347.

204 P. O. Dral, P. Zheng, B.-X. Xue, F. Ge, Y.-F. Hou, J. Max Pinheiro, Y. Su, Y. Dai and Y. Chen, *MLatom: a Package for Atomistic Simulations with Machine Learning*, version 2.3.3., Xiamen University, Xiamen, China, 2013–2023.

205 A. P. Bartók, M. C. Payne, R. Kondor and G. Csányi, *Phys. Rev. Lett.*, 2010, **104**, 136403.

206 V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti and G. Csányi, *Chem. Rev.*, 2021, **121**, 10073–10141.

207 A. S. Christensen, F. A. Faber, B. Huang, L. A. Bratholm, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, *ML: A Python Toolkit for Quantum Machine Learning*, 2017, <https://github.com/qmlecode/qml>.

208 S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Saucedo, A. Tkatchenko and K. R. Müller, *Sci. Adv.*, 2023, **9**, eadf0873.

209 H. E. Saucedo, L. E. Gálvez-González, S. Chmiela, L. O. Paz-Borbón, K.-R. Müller and A. Tkatchenko, *Nat. Commun.*, 2022, **13**, 3733.

210 S. Chmiela, H. E. Saucedo, K.-R. Müller and A. Tkatchenko, *Nat. Commun.*, 2018, **9**, 3887.

211 F. Bigi, S. N. Pozdnyakov and M. Ceriotti, *J. Chem. Phys.*, 2024, **161**, 44116.

212 J. Behler, *Chem. Rev.*, 2021, **121**, 10037–10072.

213 J. Behler and M. Parrinello, *Phys. Rev. Lett.*, 2007, **98**, 146401.

214 C. Devereux, J. S. Smith, K. K. Huddleston, K. Barros, R. Zubatyuk, O. Isayev and A. E. Roitberg, *J. Chem. Theory Comput.*, 2020, **16**, 4192–4202.

215 R. Drautz, *Phys. Rev. B*, 2019, **99**, 014104.

216 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt and B. Kozinsky, *Nat. Commun.*, 2022, **13**, 2453.

217 J. T. Frank, O. T. Unke, K.-R. Müller and S. Chmiela, *Nat. Commun.*, 2024, **15**, 6539.

218 K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko and K. R. Müller, *J. Chem. Theory Comput.*, 2019, **15**, 448–455.

219 M. Gastegger, A. McSloy, M. Luya, K. T. Schütt and R. J. Maurer, *J. Chem. Phys.*, 2020, **153**, 044123.

220 M. Gastegger, K. T. Schütt and K.-R. Müller, *Chem. Sci.*, 2021, **12**, 11473–11483.

221 J. Martinka, L. Zhang, Y.-F. Hou, M. Martyka, J. Pittner, M. Barbatti and P. O. Dral, *NPJ Comput. Mater.*, 2025, **11**(1), 132.

222 M. Martyka, X.-Y. Tong, J. Jankowska and P. O. Dral, *ChemRxiv*, 2025, DOI: [10.26434/chemrxiv-2025-j207x](https://doi.org/10.26434/chemrxiv-2025-j207x).

223 J. Li and S. A. Lopez, *Chem.-Eur. J.*, 2022, **28**, e202200651.

224 C. Zhu and H. Nakamura, *J. Chem. Phys.*, 1995, **102**, 7448–7461.

225 L. Wang, C. Salguero, S. A. Lopez and J. Li, *Chem.*, 2024, **10**, 2295–2310.

226 Z. Li, F. J. Hernández, C. Salguero, S. A. Lopez, R. Crespo-Otero and J. Li, *Nat. Commun.*, 2025, **16**, 1194.

227 R. Barrett, J. C. Dietschreit and J. Westermayr, *arXiv*, 2025, arXiv:2502.21045, DOI: [10.48550/arXiv.2502.21045](https://doi.org/10.48550/arXiv.2502.21045).

228 I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin, F. Berger, N. Bernstein, A. Bhowmik, S. M. Blau, V. Cărare, J. P. Darby, S. De, F. D. Pia, V. L. Deringer, R. Elijósius, Z. El-Machachi, F. Falcioni, E. Fako, A. C. Ferrari, A. Genreith-Schriever, J. George, R. E. A. Goodall, C. P. Grey, P. Grigorev, S. Han, W. Handley, H. H. Heenen, K. Hermansson, C. Holm, J. Jaafar, S. Hofmann, K. S. Jakob, H. Jung, V. Kapil, A. D. Kaplan, N. Karimitari, J. R. Kermode, N. Kroupa, J. Kullgren, M. C. Kuner, D. Kuryla, G. Liepuoniute, J. T. Margraf, I.-B. Magdău, A. Michaelides, J. H. Moore, A. A. Naik, S. P. Niblett, S. W. Norwood, N. O'Neill, C. Ortner, K. A. Persson, K. Reuter, A. S. Rosen, L. L. Schaaf, C. Schran, B. X. Shi, E. Sivonxay, T. K. Stenczel, V. Svahn, C. Sutton, T. D. Swinburne, J. Tilly, C. van der Oord, E. Varga-Umbrich, T. Vegge, M. Vondrák, Y. Wang, W. C. Witt, F. Zills and G. Csányi, A foundation model for atomistic materials chemistry, 2024, <https://arxiv.org/abs/2401.00096>.

229 D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole and G. Csányi, *MACE-OFF: Transferable Short Range Machine Learning Force Fields for Organic Molecules*, 2025, <https://arxiv.org/abs/2312.15211>.

230 E. Cignoni, D. Suman, J. Nigam, L. Cupellini, B. Mennucci and M. Ceriotti, *ACS Cent. Sci.*, 2024, **10**, 637–648.

231 J. Westermayr and R. J. Maurer, *Chem. Sci.*, 2021, **12**, 10755–10764.

232 J. Behler, *Chem. Rev.*, 2021, **121**, 10037–10072.

233 M. Richter, P. Marquetand, J. González-Vázquez, I. Sola and L. González, *J. Chem. Theory Comput.*, 2011, **7**, 1253–1258.



234 G. W. Richings, I. Polyak, K. E. Spinlove, G. A. Worth, I. Burghardt and B. Lasorne, *Int. Rev. Phys. Chem.*, 2015, **34**, 269–308.

235 K. E. Spinlove, G. Richings, M. A. Robb and G. A. Worth, *Faraday Discuss.*, 2018, **212**, 191–215.

236 B. Blasiak, D. Brey, W. Koch, R. Martinazzo and I. Burghardt, *Chem. Phys.*, 2022, **560**, 111542.

237 T. J. Frankcombe, M. A. Collins and G. A. Worth, *Chem. Phys. Lett.*, 2010, **489**, 242–247.

238 G. W. Richings and S. Habershon, *J. Chem. Phys.*, 2018, **148**, 134116.

239 W. Koch, M. Bonfanti, P. Eisenbrandt, A. Nandi, B. Fu, J. Bowman, D. Tannor and I. Burghardt, *J. Chem. Phys.*, 2019, **151**, 064121.

240 S. Manzhos, J. Carrington and M. J. Tucker, *J. Chem. Phys.*, 2006, **125**, 194105.

241 J. Westermayr and P. Marquetand, *J. Chem. Phys.*, 2020, **153**, 154112.

242 Y. Zhu, J. Peng, C. Xu and Z. Lan, *J. Phys. Chem. Lett.*, 2024, **15**, 9601–9619.

243 F. Häse, I. F. Galván, A. Aspuru-Guzik, R. Lindh and M. Vacher, *Chem. Sci.*, 2019, **10**, 2298–2307.

244 S. R. Hare, L. A. Bratholm, D. R. Glowacki and B. K. Carpenter, *Chem. Sci.*, 2019, **10**, 9954–9968.

245 Y. Zhu, J. Peng, X. Kang, C. Xu and Z. Lan, *Phys. Chem. Chem. Phys.*, 2022, **24**, 24362–24382.

246 X. Li, D. Hu, Y. Xie and Z. Lan, *J. Chem. Phys.*, 2018, **149**, 244104.

247 F. Häse, I. F. Galván, A. Aspuru-Guzik, R. Lindh and M. Vacher, *Chem. Sci.*, 2019, **10**, 2298–2307.

248 F. Häse, I. F. Galván, A. Aspuru-Guzik, R. Lindh and M. Vacher, *J. Phys.:Conf. Ser.*, 2020, **1412**, 042003.

249 C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, 1st edn, 2006.

250 B. Schölkopf, A. Smola and K.-R. Müller, *Artificial Neural Networks — ICANN'97*, 1997, pp. 583–588.

251 A. Mead, *J. R. Stat. Soc. Ser. D*, 1992, **41**, 27–39.

252 M. A. A. Cox and T. F. Cox, *Handbook of Data Visualization*, Springer Berlin Heidelberg, 2008, pp. 315–347.

253 B. X. Xue, M. Barbatti and P. O. Dral, *J. Phys. Chem. A*, 2020, **124**, 7199–7210.

254 B. Ghojogh, M. Crowley, F. Karay and A. Ghodsi, in *Multidimensional Scaling, Sammon Mapping, and Isomap*, Springer International Publishing, Cham, 2023, pp. 185–205.

255 N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann and A. Vilanova, *Comput. Graph. Forum*, 2016, **35**, 21–30.

256 M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, *SIGMOD Rec.*, 1999, **28**, 49–60.

257 T. Zhang, R. Ramakrishnan and M. Livny, *Data Min. Knowl. Discov.*, 1997, **1**, 141–182.

258 T. Zhang, R. Ramakrishnan and M. Livny, *SIGMOD Rec.*, 1996, **25**, 103–114.

259 X. Jin and J. Han, *Encyclopedia of Machine Learning*, Springer US, 2010, pp. 563–564.

260 E. Schubert, J. Sander, M. Ester, H. P. Kriegel and X. Xu, *ACM Trans. Database Syst.*, 2017, **42**, 1–21.

261 H. Klem, G. M. Hocky and M. McCullagh, *J. Chem. Theory Comput.*, 2022, **18**, 3218–3230.

262 P. J. Rousseeuw, *J. Comput. Appl. Math.*, 1987, **20**, 53–65.

263 L. E. Ekemeyong Awong and T. Zielinska, *Sensors*, 2023, **23**, 7925.

264 P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski and T. Ideker, *Genome Res.*, 2003, **13**, 2498–2504.

265 R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rufwurm and K. Kolar, *J. Mach. Learn. Res.*, 2020, **21**, 1–6.

266 R. Drautz, *Phys. Rev. B*, 2019, **99**, 014104.

267 K. Singh, J. Munchmeyer, L. Weber, U. Leser and A. Bande, *J. Chem. Theory Comput.*, 2022, **18**, 4408–4417.

268 R. Curth, T. E. Röhrkasten, C. Müller and J. Westermayr, *Sci. Data*, 2025, **12**, 1300.

