



Machine learning heralding a new development phase in molecular dynamics simulations

Eva Prašnikar¹ · Martin Ljubič^{1,3} · Andrej Perdih^{2,3}  · Jure Borišek¹ 

Accepted: 11 February 2024 / Published online: 29 March 2024
© The Author(s) 2024

Abstract

Molecular dynamics (MD) simulations are a key computational chemistry technique that provide dynamic insight into the underlying atomic-level processes in the system under study. These insights not only improve our understanding of the molecular world, but also aid in the design of experiments and targeted interventions. Currently, MD is associated with several limitations, the most important of which are: insufficient sampling, inadequate accuracy of the atomistic models, and challenges with proper analysis and interpretation of the obtained trajectories. Although numerous efforts have been made to address these limitations, more effective solutions are still needed. The recent development of artificial intelligence, particularly machine learning (ML), offers exciting opportunities to address the challenges of MD. In this review we aim to familiarize readers with the basics of MD while highlighting its limitations. The main focus is on exploring the integration of deep learning with MD simulations. The advancements made by ML are systematically outlined, including the development of ML-based force fields, techniques for improved conformational space sampling, and innovative methods for trajectory analysis. Additionally, the challenges and implications associated with the integration of ML and artificial intelligence are discussed. While the potential of ML-MD fusion is clearly established, further applications are needed to confirm its superiority over traditional methods. This comprehensive overview of the new perspectives of MD, which ML has opened up, serves as a gentle introduction to the exciting phase of MD development.

Keywords Molecular dynamics simulations · Machine learning · Deep learning · Artificial intelligence

✉ Andrej Perdih
andrej.perdih@ki.si

✉ Jure Borišek
jure.borisek@ki.si

¹ Theory Department, Laboratory for Chemical Informatics, National Institute of Chemistry, Ljubljana, Slovenia

² Theory Department, Laboratory for Computational Biochemistry and Drug Design, National Institute of Chemistry, Ljubljana, Slovenia

³ Faculty of Pharmacy, University of Ljubljana, Ljubljana, Slovenia

Abbreviations

ABF	Adaptive biasing force method
ACSF	Atom-centred symmetry function
AE	Autoencoder
AI	Artificial intelligence
AIMD	ab initio molecular dynamics
AIMNet-NSE	Neural spin equilibration
aMD	Accelerated molecular dynamics
ANI	1-Accurate NeurAI network engINe for molecular energies
CC	Coupled cluster
CG	Coarse grained model
CMAP	Correction map
CNN	Convolutional neural network
CPU	Central processing unit
CV	Collective variable
CVAE	Convolutional variational autoencoders
D2R	Dopamine receptor subtype D2
DCF	Direct covariant forces
Deep-LDA	Deep linear discriminant analysis
Deep-TDA	Deep targeted discriminant analysis
DFT	Density functional theory
DL	Deep learning
DTNN	Deep tensor neural network
FES	Free energy surface
FFNN	Feedforward network
FFS	Forward flux sampling
FUNN	Force-biasing using neural networks
GAN	Generative adversarial network
GNN	Graph neural network
GNNFF	Graph neural network force field
GPCR	G protein-coupled receptor
GPU	Graphics processing unit
HLDA	Harmonic linear discriminant analysis
ICA	Independent component analysis
IMD	Independent Markov decomposition
ITS	Integrated tempering sampling
LSTM	Long short-term memory
MBTR	Many-body tensor representation
MD	Molecular dynamics
ML	Machine learning
MM	Molecular mechanics
MSES	Multiscale enhanced sampling
MSM	Markov state model
MTL	Multi-task learning
NLP	Natural language processing
NN	Neural network
NNFF MD	Neural network force field molecular dynamics
NNP	Neural-network potentials

OPES	Flooding approach-on-the-fly probability enhanced sampling-flooding approach
PCA	Principal component analysis
PES	Potential energy surface
QM	Quantum-mechanical
RBM	Restricted Boltzmann Machine
REAP	REinforcement learning based Adaptive samPLing
REMD	Replica exchange molecular dynamics
RMSD	Root mean square deviation
RNN	Recurrent neural network
SASA	Solvent accessible surface area
SOAP	Smooth overlap of atomic positions
SSAGES	Advanced general ensemble simulations
TICA	Time-lagged independent component analysis
TPI-Deep-TDA	Transition path informed deep-TDA
TPS	Transition path sampling
UB	Urey-Bradley
VAC	Variational approach for conformation dynamics method
VAE	Variational autoencoders
VAMP	Variational approach for Markov processes
VES	Variationally enhanced sampling
5-HT2AR	Serotonin receptor subtype 2 A

1 Introduction

In modern times, computer science has become an indispensable part of virtually all areas of human activity. The constant evolution of technology has led to new advances that continue to impact the way we live, work, and interact with the world around us. One area where computers have left a significant impact are simulations of molecular systems, with one of its flagship methods being molecular dynamics (MD). These key tools of computational chemistry have been used for some time to gain insights into the behavior of various systems under specific conditions at the atomistic level. As such, they are critical for understanding the properties of materials and biological systems and their interactions with the environment, which is essential in areas such as drug discovery, materials science, and nanotechnology (Schlick and Portillo-Ledesma 2021).

The roots of artificial intelligence (AI) can be traced even further back than the occurrence of MD simulations and in recent years significant progress has been made in this field. This claim can be further supported by the fact that this general introduction paragraph was written and improved in part by using ChatGPT (Rudolph et al. 2023), an AI language model from the generative pre-trained transformer (GPT) family of language models. Advances in AI technology have led to rapid progress in its use in a variety of domains, including MD simulations, with the increased boom occurring after 2015. AI algorithms can represent a valuable tool for optimizing simulations to run significantly faster (Thölke and De Fabritiis 2022; Galvelis et al. 2023), efficiently extracting information from generated molecular trajectories (Mardt et al. 2018), and predicting the behavior of complex macromolecular systems (Moritsugu 2021; Guo et al. 2018). However, AI implementations are not without significant challenges, such as the need for large amounts

of data, the struggle of efficiently training the algorithms to accurately capture complex systems. Additionally, in terms of speed, neural network potentials (NNPs) are generally slower than classical force fields (FF) (Behler 2016, 2021). In this context the inclusion of AI constitutes a new development phase in MD simulations. As these technologies continue to evolve, it is clear that they will play an increasingly critical role in shaping the future of scientific research and its practical applications.

In this review, we aim to provide a comprehensive coverage of the latest advancements in implementing AI within the framework of MD simulation. As researchers who apply MD simulations in our own work, we understand the challenges of staying up-to-date with the latest developments and innovations in the field. With this work we hope to equip fellow researchers with the knowledge and tools to improve the accuracy and efficiency of their MD simulations, while our review may also be informative for method developers. To begin with, a concise introduction to the MD method and an outline of its main three challenges, namely insufficient sampling, inadequate accuracy of the atomistic models, and challenging interpretation of the obtained trajectories are provided. This is followed by an overview of techniques for improved sampling of conformational space designed without the implementation of machine learning (ML). Next, we discuss most of the deep learning (DL) - based architectures to enhance the understanding of their applications in MD and present the recently developed ML-based approaches for dealing with three persisting key problems encountered in MD. We conclude with a critical discussion of some of the challenges associated with this exciting ML-MD fusion and attempt to assess its wider impact and speculate on the potential future directions that the MD field might take next. We hope that such comprehensive treatment of this developmental step in MD will ensure a better understanding and appreciation of the diverse applications of ML in MD. In addition to this classification, the integration of AI techniques in MD simulations can also be approached from an algorithm-based perspective (Zhang et al. 2020).

2 Molecular dynamics simulations: methodologies and challenges

The emergence of MD simulations in chemistry can be traced back to 1957 when Alder and Wainwright used them in a study of simple gases (Alder and Wainwright 1957). Following a subsequent development of algorithms as well as their computational applicability, the first MD simulation of a protein, bovine pancreatic trypsin Inhibitor, in a vacuum was reported by McCammon, Gelin, and Karplus in 1977 (McCammon et al. 1977). With MD's potential unraveled, it was subsequently extensively used in protein research to study conformational changes, protein-ligand interactions, and even reaction mechanisms. It became a complementary method to interpreted biochemical experiments to support research in material science as well as in computer-aided molecular design (Wu et al. 2022).

Fundamentally, there are three major challenges associated with MD simulations: (i) (in)accuracy of the used FF and/or too extreme approximations leading to systematic errors, (ii) challenging interpretation of the high-dimensional and noisy molecular trajectories, and (iii) the limited computational power that restricts the trajectory length in a given simulation runtime, leading to statistical errors due to inadequate sampling of the conformational space available to the system (Fig. 1) (Durrant and McCammon 2011; Sidky et al. 2020; Hénin et al. 2022). To address the last challenge, several algorithms were developed to allow a more efficient sampling of the conformational space of the investigated molecular system.

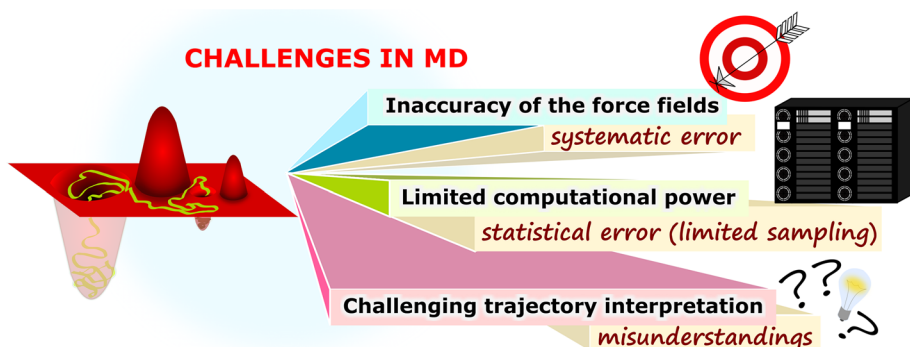


Fig. 1 Three major challenges of the molecular dynamics simulation technique: (i) imprecision of the force fields leading to systematic errors, (ii) limited computational power leading to statistical errors, (iii) interpretation of high-dimensional trajectories

2.1 Atomistic models and force fields

Performing MD simulations within a comprehensive framework of time-dependent many-body Schrödinger equation has been a long-standing challenge in quantum computational chemistry. Since nuclei have a larger mass than electrons, the Born-Oppenheimer approximation can be used, and the wave function becomes a product of a nuclear and electronic wave function with the first one explicitly dependent on time. However, quantum dynamics is still computationally too intensive for larger systems even for the most efficient and capable supercomputers (Ollitrault et al. 2021).

This significant challenge of modeling the atomic structure and its dynamics with quantum mechanics (QM) was efficiently solved by introducing another description of molecular structure-molecular mechanics (MM), and subsequent development of classical MD simulations which reduce the computational complexity by considering an approximation that movement of atoms can be described based on Newtonian (classical) mechanics. In MD simulations, Newton's equation of motion is thus used to calculate the time-dependent changes in the position and velocities of the moving atoms of the molecular systems under study (Wu et al. 2022; Durrant and McCammon 2011). Forces acting on individual atoms can be calculated using FF (i.e. molecular mechanics) which define the systems potential energy and include the contributions of bonded (i.e. chemical bonds, valence and dihedral angles) and nonbonded interactions (i.e. van der Waals and Coulomb interactions).

The FF approach is viable also because in many cases quantum effects can be globally neglected. In addition to calculating forces via classical FF, *ab initio* MD was also developed, in which the energy of the system is calculated at each time step using a selected QM method (Iftimie et al. 2005). The energy terms in FF are appropriately parameterized to reproduce QM calculations and experimental (e.g., spectroscopic) data. In this order, parameters such as stiffness and length of springs describing bonds and angles, as well as partial atomic charges and van der Waals atomic radii are determined (Fig. 2). These are then used to calculate the forces between particles and their potential energies $U(\vec{R})$ (Durrant and McCammon 2011). However, standard FF neglect several physical effects such as electronic polarization, charge transfer and many-body dispersion (Melcr and Piquemal

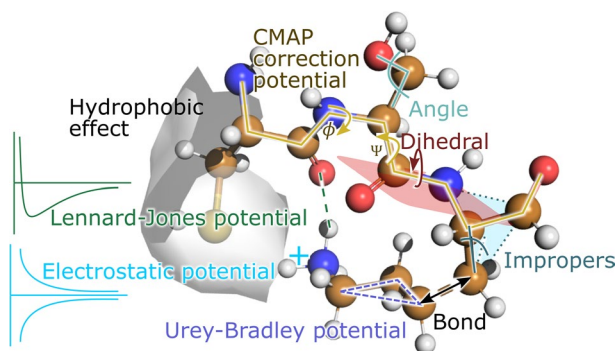


Fig. 2 Illustration of elements used in the construction of force fields. The bonded terms: bonds, which represent covalently bonded atoms, angles that account for the bending energy between every triplet of bonded atoms, dihedrals that describe the torsional energy of four sequentially bonded atoms, and improper dihedrals that involve a central atom connected to three peripheral atoms. The Urey-Bradley potential considers the coupling between bond length and bond angle, while the CMAP (Correction Map) potential corrects for the conformation of the peptide backbone (e.g., in CHARMM27 force field). The non-bonded terms: the Lennard-Jones potential, which characterizes the non-electrostatic interactions between pairs of non-bonded atoms, and the electrostatic potential, which accounts for the electrostatic interactions between charged atoms and follows Coulomb's law

2019). There has been a lot of effort dedicated to the development of polarizable FF to provide a more accurate treatment of the atomistic structure (Jing et al. 2019).

2.2 Sampling in molecular dynamics simulations

The challenges of conformational sampling in MD simulations depend heavily on the high energy barrier that must be overcome for one configuration to transition to another. To ensure a more efficient sampling of a vast conformational space of atomistic systems under study, several approaches can be utilized, including simulation of lower-level representations, coarse-grained (CG) systems (Kmieciak et al. 2016) and enhanced sampling (Liao 2020). Enhanced sampling, sometimes also referred to as accelerated sampling, is a complex concept encompassing intertwined methods, making it difficult for simple categorization. Some techniques are exploratory and aim to discover new regions of the configuration space and provide only semi-quantitative estimates of the probability distribution, while others allow estimation of probability distributions and free energies from the sampled space. In addition, some of the enhanced sampling schemes do not preserve the kinetics of the system. In general, these methods should allow sampling of larger portions of the available conformational space in a given amount of simulation time. Enhanced sampling can be achieved by sampling at higher temperatures, adding external forces or potentials, driving an adiabatically decoupled degree of freedom, or extending the ensemble under consideration (Fig. 3).

Based on the utilization of the collective variables (CVs), enhanced sampling techniques can be broadly divided into two groups: (1) CV-based, constrained enhanced sampling and (2) CV-free enhanced sampling. The first group includes approaches such as: umbrella sampling (Fig. 3), steered MD, metadynamics (Fig. 3), potential smoothing methods, J-walking, local elevation, conformational flooding, hyperdynamics, conformational space annealing, adaptive biasing force method, local elevation umbrella sampling, and variationally enhanced sampling (VES) (Yang et al. 2019). The main methods of the second group

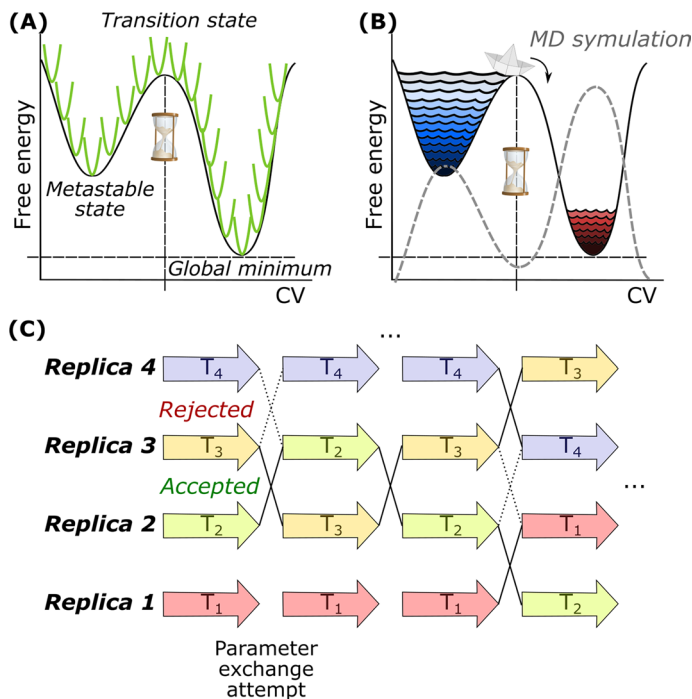


Fig. 3 Outline of some of the techniques developed to improve sampling of the available conformational space in MD simulations. **A** Umbrella sampling method—green lines represent the harmonic bias potentials added to the system Hamiltonian at different CV points along the CV space. **B** Metadynamics: history-dependent Gaussian-type biases are applied across the CV space. When the first basin (blue) is filled, the MD simulation is allowed to traverse high transition barrier and explore the red basin. Once the red basin is also filled, the accumulated biases are summed, enabling estimation of the negative free energy landscape (represented by the grey dashed line). **C** Replica exchange method: several replicas are simulated in parallel at different temperatures. In regular intervals exchange in temperatures/configurations between replicas is attempted. This exchange is allowed only when Metropolis criterion is satisfied (Liao 2020). (Color figure online)

are replica exchange MD (REMD) (parallel tempering) (Fig. 3), accelerated MD (aMD) (Liao 2020), simulated tempering, multicanonical simulation, temperature-accelerated dynamics, Wang-Landau algorithm, statistical temperature sampling, temperature-accelerated MD, enveloping distribution sampling, integrated tempering sampling (ITS), and accelerated enveloping distribution sampling (Yang et al. 2019). Several methods for analysis of MD trajectories as well as methods of enhanced sampling are related to the concept of collective variables (CVs), which are a functional mappings of full 3 N-dimensional configurations to a lower-dimensional representation (Hénin et al. 2022), which then serves as a coarse-grained description of a system.

In CV-free enhanced sampling techniques predefined CVs are not required as well as prior knowledge about the studied process (Liao 2020). For example, in popular REMD, multiple simulations run simultaneously at different temperatures or potential energy functions, with correlation reduced by exchanging conformations/temperatures between replicates at regular time intervals to avoid trapping the simulation in a stable conformation (Fig. 3) (Chen 2021; Liao 2020). The effectiveness of REMD depends strongly on the

activation enthalpy and the choice of maximum temperature (Bernardi et al. 2015). However, the REMD method is not limited to changing the temperature, as any control parameter can be changed and even the expression of the Hamiltonian can be modified (Abrams and Bussi 2013; Liao 2020). Another popular method is aMD where a boost potential (acting on torsional potential, the whole potential or both) is added to the system potential energy surface (PES) when it is below the threshold energy to promote crossing the energy barriers. In addition, the boost potential can be added only in certain regions of the system to promote specific conformations. aMD is a powerful approach to conformational sampling, but reweighting methods can also restore the free energy landscape (Liao 2020).

Moreover, in CV-based enhanced sampling techniques, a bias potential is introduced into the system along the defined CVs to overcome the energy barrier (Yang et al. 2019). Here, as with aMD, the free energy landscape can be restored later by post analysis (Chen 2021). The bias can be introduced, for example, by changing the temperature or potential energy function. The bias itself can be used either to constrain the dynamics around a conformation like in umbrella sampling (Fig. 3) technique (Yang et al. 2019; Torrie and Valleau 1977; Liao 2020; Hénin et al. 2022) or to fill the minima on PES to allow barrier crossing as it is implemented in a popular metadynamics (Abrams and Bussi 2013; Hénin et al. 2022; Bernardi et al. 2015) (Fig. 3). These two techniques can also be combined into the so called well-sliced metadynamics (Awasthi et al. 2016). The principle limitation of these techniques is that prior knowledge of the process under study (i.e. free energy surface) is crucial. Instead of the bias potential, a bias force can be applied directly to the mean force felt by the CVs, leading to adaptive bias force methods (Chen 2021).

When particularly interested in the detailed dynamics of the transition pathways between well-defined metastable states path sampling techniques (Chong et al. 2017) can be adopted. These methods include for example transition path sampling (TPS) and forward flux sampling (FFS) (Chong et al. 2017) and facilitates rare events (e.g. transition states) (Zwier and Chong 2010). For conducting path sampling techniques initial and final states of the studied system must be known (Zwier and Chong 2010).

Interestingly, a multiscale approach can also be used to provide enhanced conformational sampling. In multiscale enhanced sampling (MSES), the sampling of the all-atom protein is enhanced by accelerated dynamics of the associated coarse-grained model (CG).

Within enhanced sampling techniques we can also find a subgroup of methods designated as adaptive sampling. Their primary goal is to efficiently utilize computational resources by continuously monitoring the simulation. The information obtained from analyzing already-simulated part of the trajectory helps decide where to sample next. This allows for broader sampling of the conformational space and increases the likelihood of discovering new and interesting conformations. Adaptive sampling strategies primarily vary in the analysis step, where approaches such as Markov state models (MSM) can be utilized (Hruska et al. 2020). Briefly, the Markov state based enhanced sampling uses CVs to construct the MSM and bias the simulations by restarting at less sampled states (Doerr and De Fabritiis 2012; Pérez et al. 2020). More in depth description of this particular approach is given in the Chapter. 4.3.

2.3 Trajectory analysis in molecular dynamics simulations

The result of the MD simulation is a trajectory that captures a time-dependent evolution of the conformations of the system. The number of calculations required for the MD simulation increases with the length of the trajectory and the size of the system under study.

There are several approaches available that can be used when analyzing MD trajectories including visual inspection, calculation of energies, measuring geometric parameters such as interatomic distances, angles and similar, performing cluster analysis, generating cross-correlation matrices, root mean square deviation and fluctuation analysis, principal component analysis (PCA), generating contact maps, checking for specific interactions like H-bonds, hydrophobic interaction, solvent accessible surface area (SASA) analysis, to name a few (Likhachev et al. 2016; Baltrukovich and Podlowska 2022). All these methods represent invaluable tools that enable an interpretation of obtained data and the establishment of the connection with the experiment.

3 Machine learning and neural networks

AI can be broadly defined as a branch of computer science dealing with the development of systems that exhibit characteristics we associate with intelligence. One of the earliest papers in this field was published in 1943 by Warren McCulloch and Walter Pitts, introducing the model of artificial neurons (McCulloch and Pitts 1943). Early foundational ideas, like Turing's Test in the 1950s, set the stage for AI research. The 1956 Dartmouth Conference marked AI's official emergence as a research field, followed by significant advancements in the 1960 and 1970 s, and the development of expert systems. The 1980 and 1990 s saw a shift towards connectionism and the emergence of ML, leading to the creation of algorithms that enabled computers to learn from data. The 2000s ushered in the era of DL (Hinton et al. 2006), transforming AI capabilities in areas like image recognition and natural language processing. From its beginnings to the present, AI experienced some golden years and also part of the so-called "*AI winters*," characterized by a shortage of funds because initial high expectations of this technology were not met. This was related to the still underdeveloped field and the low computing power available at the time (Xu et al. 2021). Later, the development of capable central processing units (CPUs) and graphics processing units (GPUs) provided the necessary computing power, and since the early to mid-1990s enable the computer science to thrive (Zhang and Lu 2021; Hwang 2018).

To list some AI success, *AlphaGo* defeated the world go champion in 2016 (Zhang and Lu 2021), the social humanoid robot *Sophia* (Retto 2017) made its first public appearance in the same year, while one of the most advanced chatbots *chatGPT* (Rudolph et al. 2023) appeared in 2022. A major breakthrough in life science was made by *Alphafold* (Jumper et al. 2021) in 2021, which uses DL techniques to make highly accurate protein structure predictions. Still, Alphafold faces some limitations, such as limited ability to predict the outcome of point mutations or, structures of complexes with small-molecule ligands, and model induced fit (Spiwok et al. 2022). AI is a collective term that encompasses several areas such as vision, speech, expert systems, robotics, planning, ML, and natural language processing (NLP). AI applications are often based on ML methods that implement the core idea of AI and can be divided into several classes: (un)supervised learning, dimensionality reduction, semi-supervised learning, reinforcement learning, and DL (Mukhamediev et al. 2022). Here, we mainly focus on the fastest growing subfield of ML-DL.

To sum up, ML involves teaching computers to make predictions or decisions based on data. This is accomplished by using algorithms that adjust parameters such as weights and biases in a model that maps inputs to outputs through multiple layers of artificial neurons. These algorithms can be supervised or unsupervised, use reinforcement learning, etc. A loss function is a measure of how well a ML model fits the training data. It calculates the

difference between the predicted output and the actual value. The goal of training a model is to minimize the loss function, which in turn helps improve the accuracy of the model's predictions (Vapnik 1999; Goodfellow et al. 2016).

Traditional ML techniques, such as decision trees, linear regression, and support vector machines, involve algorithms that learn from data to make predictions or decisions. They are generally less complex than DL models and require less computational power. However, they may struggle with very large datasets and complex problems. On the other hand, DL, a subset of ML, uses neural networks with multiple layers to model complex patterns in data. DL excels in tasks like image and speech recognition, natural language processing, and can handle large and complex datasets. However, DL models require substantial computational resources and large amounts of data to train effectively. DL can also be seen as “black box,” providing little insight into how decisions are made. While traditional ML methods are less resource-intensive and more interpretable, they may lack the sophistication needed for complex tasks. On the other hand, DL offers powerful tools for handling large and intricate datasets, but at the cost of increased computational demands and less interpretability (Janiesch et al. 2021).

There are some platforms that offer open-source tools for ML/DL that are used by data scientists, ML engineers, and researchers to develop and train models for a wide range of applications such as image recognition, NLP, speech recognition, etc. (Table 1). These tools provide a range of algorithms, application programming interfaces, and frameworks,

Table 1 Examples of open-source ML/DL tools (Latif et al. 2021)

Name	Short description
SciKit-learn (Pedregosa et al. 2011)	A ML library for Python, offering algorithms for classification, regression, clustering, and more.
TensorFlow (Abadi et al. 2016, 2016)	An open-source platform for building and deploying ML models.
Microsoft CNTK (Microsoft Cognitive Toolkit 2016)	A DL framework, designed to train and evaluate deep neural networks.
Keras (Chollet et al. 2015)	A high-level neural networks application programming interface, capable of running on top of TensorFlow or CNTK.
PyTorch (Paszke et al. 2019)	An open-source DL framework that provides fast, dynamic, and intuitive development of deep neural networks.
Mxnet (Chen et al. 2015)	A scalable DL framework that allows you to train and deploy neural networks on a variety of devices.
Chainer (Tokui et al. 2019)	A Python-based DL framework that emphasizes flexibility and computational efficiency.
Caffe (Jia et al. 2014)	A DL framework, with a focus on speed and modularity.
PyTorch VAE (Subramanian 2020)	A PyTorch implementation of a variational autoencoder (VAE), a type of deep generative model.
DL4J (Eclipse DeepLearning4j Development Team)	DL library written for Java and JVM (Java Virtual Machine)
PyCaret (Ali 2020) JAX (Bradbury et al. 2018)	An open-source, low-code ML library in Python that automates ML workflows. A DL framework developed by Google, which couples a modified version of autograd and TensorFlow

that help developers build, train, and deploy ML models efficiently. Each tool has its own strengths and weaknesses, and the choice of tool depends on the specific requirements of the project at hand.

In terms of architecture DL models can be categorized into different groups. There are some basic architectures/building blocks such as: (i) feedforward neural networks (FFNNs) (Goodfellow et al. 2016), (ii) convolutional neural networks (CNNs) (LeCun et al. 2015; Cong and Zhou 2023), (iii) recurrent neural networks (RNNs) (Goodfellow et al. 2016; Vaswani et al. 2017; Medsker and Jain 1999), and (iv) restricted Boltzmann machines (RBMs) (Latif et al. 2021; Upadhy and Sastry 2019). These form the basis for many other more complex designs, such as: (v) generative adversarial networks (GANs) (Goodfellow et al. 2016; Aggarwal et al. 2021; Vint et al. 2021), (vi) autoencoders (AEs) (Tian et al. 2021; Goodfellow et al. 2016), (vii) variational autoencoders (VAEs) (Tian et al. 2021; Goodfellow et al. 2016), (viii) transformers (Lin et al. 2022; Vaswani et al. 2017), and (ix) graph neural networks (GNNs) (Zhou et al. 2020; Sanchez-Lengeling et al. 2021; Mukhamediev et al. 2022).

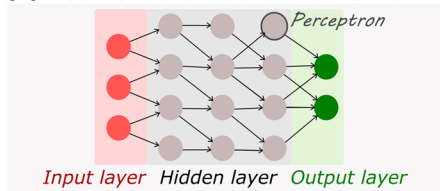
FFNNs are the most basic artificial neural networks that have an input layer, at least one hidden layer, and an output layer. Input data is passed through the network and transformed by weights and biases in each layer to produce an output that can be used for prediction or classification tasks (Goodfellow et al. 2016). In case some feedback connections are included, we obtain the RNN (Goodfellow et al. 2016; Vaswani et al. 2017; Medsker and Jain 1999). RBMs consists of two layers, one visible and one hidden. While the visible layer receives and encodes input data, the hidden layer builds its latent representation that captures the underlying structure of the data. Once an RBM is properly trained, it can generate new samples by drawing from the probability distribution represented by the hidden layer (Upadhy and Sastry 2019; Latif et al. 2021). CNNs can function as computer vision. They use convolutional layers to scan an image, extract features through convolution operations, and apply activation functions to the output of each convolution operation, in order to add nonlinearity. The pooling operation then reduces the spatial dimensions of the convolution layer output to control computation and overfitting. The features are then fed into fully connected layers for classification (Fig. 4) (LeCun et al. 2015; Cong and Zhou 2023).

GANs consist of two neural networks, the generator and the discriminator. The first takes data from the latent space and generates a new image. The second takes the generated image and the real image and decides whether the image is authentic or generated by the generator. Both networks are trained against each other, and in order to obtain a generator that provides images that the discriminator perceives as authentic (Goodfellow et al. 2016; Aggarwal et al. 2021; Vint et al. 2021). AEs consist of an encoder that converts the input data into a low-dimensional representation (bottleneck or latent code) and a decoder that is able to recover the input-like output from the low-dimensional representation (Tian et al. 2021; Goodfellow et al. 2016). VAEs are essentially similar to autoencoders but model the latent space as a probability distribution (Tian et al. 2021; Goodfellow et al. 2016).

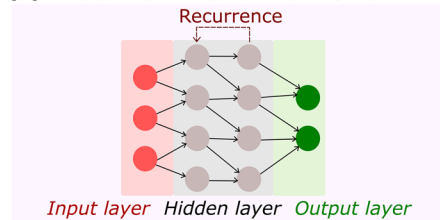
The core of the transformer is the implementation of the self-attention mechanisms that allow them to weigh the importance of different parts of the input sequence in making predictions. In its original architecture, this model consists of an encoder and a decoder. The input is first embedded, and a position encoding is performed. Then, the embedding is processed by a stack of encoders consisting of a multi-head attention mechanism and a feed-forward network, where each of these sub-layers is connected by a residual connection and normalized by layer normalization. The information is then passed to the decoder, which has a similar architecture with two multi-head-attention layers and a feed-forward network.

BASIC ARCHITECTURES

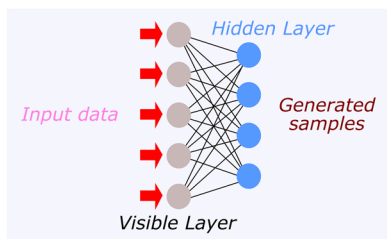
(A) Feedforward Neural Networks



(B) Recurrent Neural Networks



(C) Restricted Boltzmann Machines



(D) Convolutional Neural Networks

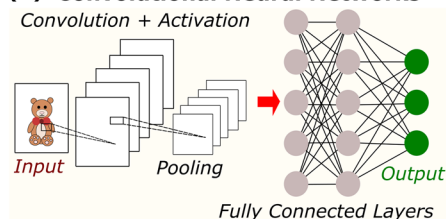


Fig. 4 Basic deep learning architectures. **A** feedforward neural network (FFNN), **B** the recurrent neural network (RNN), **C** restricted Boltzmann machines (RBM), and **D** convolutional neural networks (CNN)

Finally, linearization and softmax activation are applied to the output before it is produced as the final result (Vaswani et al. 2017; Lin et al. 2022).

Finally, GNNs take a graph representation as input. In a graph, the data can be represented as global context, nodes, edges, and the connectivity of the graph. Before applying the GNN block, an embedding step is performed for each data type to represent them as vectors or embeddings. The GNN core block consists of several operations, including message passing and aggregation, which propagate information between neighboring nodes and refine their embeddings. An update function is applied to create new embeddings for each node based on the aggregated information. After the GNN block processes the entire graph, a pooling function is used to aggregate the updated node embeddings into a graph-level representation. Then, a classification layer is applied to produce a final classification or regression output based on the graph-level representation. Finally, an activation function is applied to the output to produce the prediction of the model (Fig. 5) (Zhou et al. 2020; Sanchez-Lengeling et al. 2021).

4 Machine learning and molecular dynamics simulations

One of the common aims of MD simulations is to accurately construct the free energy surface (FES) from the well-converged simulations (Moritsugu 2021). As mentioned, before, three main problems are faced when performing MD simulations: accuracy (FF), efficiency (sampling) and the challenging interpretation of the trajectories (analysis), all of which could be tackled by AI methods. Traditionally used methods developed to address these challenges often struggle to efficiently capture complex nonlinear relationships in high-dimensional data involving intricate molecular interactions. For example, user-defined reaction coordinates to describe the progression of large-scale conformational changes or chemical reactions are one such simplified description of reduced dimensionality that

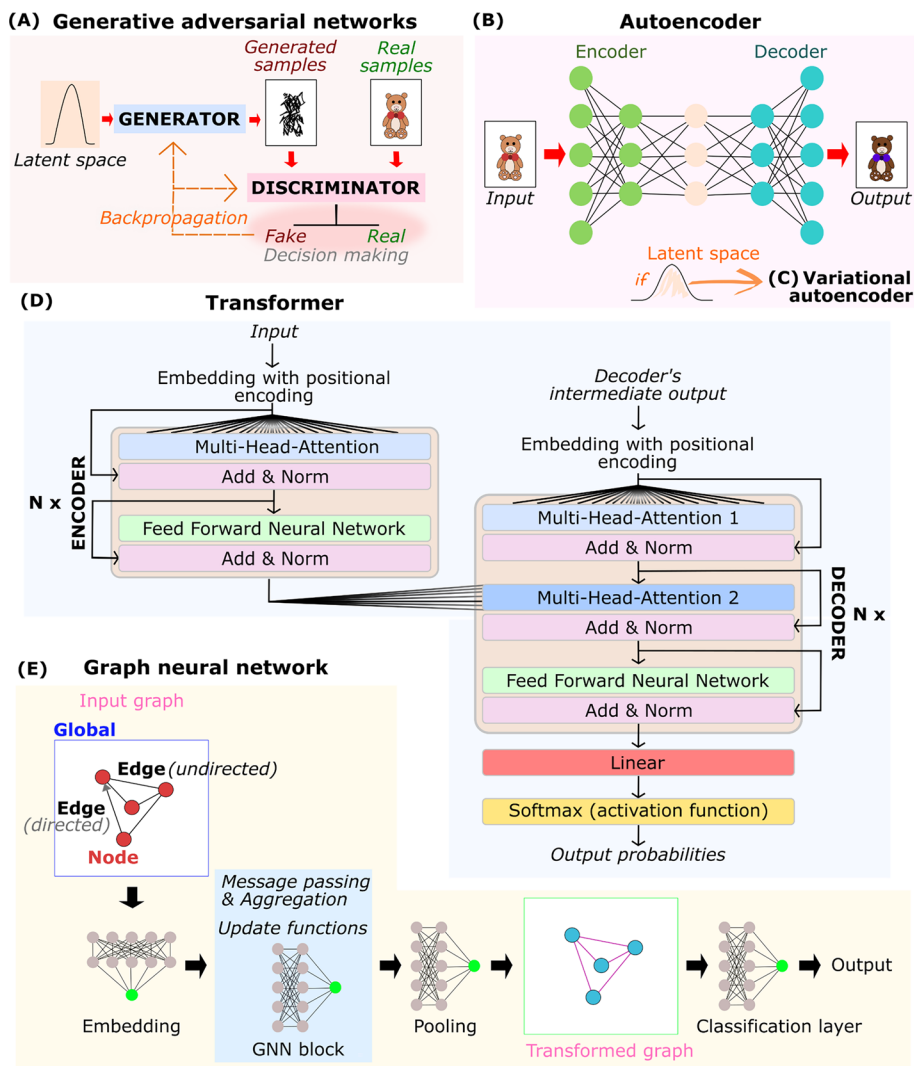


Fig. 5 Selected, more complex deep learning architectures. **A** Generative adversarial network (GAN), **B** autoencoder (AE), **C** variational autoencoder (VAE), **D** transformer, **E** graph neural network (GNN)

suffers from predefined biases (Best and Hummer 2005). In addition, conventional FF often rely on predefined functional forms and parameters, which limits their accuracy in capturing diverse and dynamic molecular behaviors (Zhang et al. 2023). Efforts have already been made to implement AI in MD calculations and in MD trajectory analysis (Noé et al. 2020; Mouvet et al. 2022; Mudedla et al. 2022; Behler and Parrinello 2007). ML has been used, for example, to extract classical potential energy surfaces (PES) from QM calculations to perform MD simulations with quantum effects (Behler and Parrinello 2007). In trajectory analysis, ML has been integrated into the construction of MSMs (Kononov et al. 2021). By learning from existing trajectories, ML algorithms can also guide simulations towards

relevant regions of the configurational space, accelerating convergence and improving the exploration of rare events. Overall the application of ML provides more flexibility, tractability and scales much better when studying high-dimensional data such a MD simulation. A more detailed overview of recent advances in MD using AI is provided in the following subsections.

4.1 Machine learning-based force fields

In this section, we focus on the methods, frameworks and libraries for ML-based FF, in particular DCF, SchNet, GNNFF, TorchMD, ACEMD3, CGSchNet, ANI and np2p NNP. ML in MD is usually utilized to replace QM calculations with ML-potentials for FF-like dynamics, allowing faster simulations with *ab initio* QM accuracy (Behler 2016, 2021). Neural networks and kernel methods are typically trained with data obtained from coupled cluster (CC) calculations or density-functional theory (DFT) and then used to predict potential energies and/or forces (Mouvet et al. 2022). In this case Cartesian coordinates of the atomic positions are not a good choice for system representation, since the output of the numerical fitting method such as neural networks depends on the absolute values of the input coordinates, while translations, rotations and permutations (i.e. physical symmetries) do not change the energy of the molecules but do change Cartesian coordinates. Structures, however, can be described by internal coordinates such as interatomic distances (Behler 2011). The input information for the evaluation of the potential should also not include information about the atom type, except for the specification of the nuclear charge, so that the chemical environment and bonding can change in the simulation. In addition, to enable bond creation and breaking, all predefined atomic connections and bonds should be disregarded (Behler 2016, 2021). Dscribe is an example of freely available software that supports the conversion of atomic structures into ML-input features (Coulomb matrix, Ewald sum matrix, sine matrix, many-body tensor representation (MBTR), atom-centered symmetry function (ACSF), and smooth overlap of atomic positions (SOAP) (Table 2) (Himanan et al. 2020).

Once the representation of the molecular system is ready and appropriate neural network is designed, a training program is required to adjust the parameters in the neural network to obtain reproducible results (i.e. potentials) with respect to the data usually derived from *ab-initio* simulations (Singraber et al. 2019). After neural network training, computer code is required to run it and calculate the forces needed in the MD simulation (Singraber et al. 2019). Here, the total potential energy can be calculated as the sum of individual atomic contributions that depend on their local environment (Singraber et al. 2019; Behler 2011; Behler and Parrinello 2007). This allows the use of atomic potentials to calculate the many-body potential of systems of arbitrary size (Fig. 6) (Behler and Parrinello 2007). Using the NNPs to carry out MD simulations results in faster development of the system over time, compared to traditional methods with proposedly same level of accuracy as present in the training QM-based data. This is due to faster prediction of chemical properties since ML models do not have to solve any complex QM formalisms (Unke et al. 2021).

Several neural network-based approaches for MD simulations as described above have been developed recently, some of which we will discuss herein. Symmetry functions (Behler 2011) can be used to describe the local atomic environment around each atom in the system. This description can then be used as input for ML-based MD methods such as neural network FF MD (NFF MD), where PES is first predicted, followed by the computation of atomic forces required for MD simulations, which allows the system to evolve

Table 2 Selected tools for ML-supported MD simulations

Name	Short descriptions
Dscribe (Himanan et al. 2020)	Python package for transforming atomic structures into neural network input (including Atom-centered Symmetry Functions)
N2p2 (Thompson et al. 2022; Singraber et al. 2019)	Ready-to-use software for high-dimensional neural network potential
LAMMPS (Thompson et al. 2022; Singraber et al. 2019)	A classical MD simulation code
OpenMM (Eastman et al. 2023b)	A GPU-based MD code
TorchMD (Doerr et al. 2021)	A DL framework for molecular simulations
TorchMD-NET (Thölke and De Fabritiis 2022)	State-of-the-art graph neural networks and equivariant transformers potentials for learning molecular potentials
CGSchNet (Husic et al. 2020)	A transferable coarse-grained force field
ANI-1 (Smith et al. 2017)	Computational DFT database
ANI-2x (Devereux et al. 2020)	Trained NNP model
AIMNet2 (Anstine et al. 2023)	Current state-of-the-art NNP
DeepPMD-kit (Wang et al. 2018)	Python/C++ package designed to minimize the effort required to build DL-based models of interatomic potential energy and FF and to perform MD
JAX MD (Schoenholz and Cubuk 2021)	JAX MD is an end-to-end differentiable MD package that enables an easy incorporation of ML models

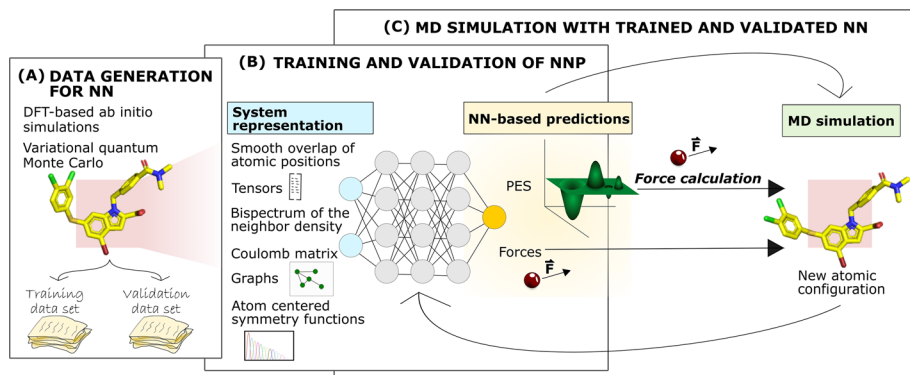


Fig. 6 Basic workflow of MD simulations with neural network-based potentials (NNP). **A** First, data is obtained, for example, from DFT or variational quantum Monte Carlo methods, and divided into training and validation sets. **B** Next, the simulated system has to be adequately represented by descriptors such as tensors, symmetry functions, graphs, etc. Then neural network is appropriately trained and validated to predict potentials or forces. If atomic potentials/potential energy surface (PES) are the final output of the neural network, then the forces acting on the atoms of the system must be calculated and are **C** finally used in the MD simulations to evolve system over time. At each step new atomic configuration is obtained that is fed to the trained and validated neural network to predict new forces acting on the atoms

over time (Behler 2011; Mailoa et al. 2019). Later, the neural network-based approach for direct covariant forces (DCF) was introduced. In DCF, Cartesian force vectors in extended solid-state systems are predicted directly from multi-element local atomic environments

without the need for any prior calculation of the potential energy of the system. DCF force prediction accuracy was evaluated by simulating polyethylene oxide and amorphous lithium phosphate oxide and comparing the predicted forces with data obtained by DFT calculations, the performance of classical FF was added as reference. OPLS 2005 FF was used for polyethylene oxide and FF from the literature developed for oxide systems for lithium phosphate oxide. DCF showed lower mean absolute error of the force error prediction compared to the classical FF. Computational speed was also evaluated for the lithium phosphate oxide system, where DFT-based MD was $\sim 10^6$ times slower than classical FF-based MD. Standard NNFF MD, where ‘atomic fingerprints’ by Behler, Parrinello are used (B-P MD) written in the PROPhet plugin for LAMMPS (C++) was ~ 100 times slower than classical FF, while the current DCF MD (in Python with Fortran acceleration) is ~ 2 times faster than the standard B-P MD (in C++) and ~ 800 times faster than standard B-P MD implementation using the Python AMP package with Fortran acceleration (Mailoa et al. 2019).

Local atomic environment representations can also be graph-based. SchNet, for example, is a ML model that uses a GNN architecture to learn PES of a molecule directly from its atomic positions. The potential energy function is learned from a set of training data, typically consisting of large-scale QM calculations of molecular energies. The accuracy of SchNet was evaluated on the MD17 dataset—a collection of MD simulations of small organic molecules—in which the mean absolute errors of energy and force predictions were below 0.12 kcal/mol and 0.33 kcal/(mol·Å), respectively. SchNet was also tested on the C₂₀-fullerene system, where the normal mode analysis of the fullerene dynamics showed the largest error of $\sim 1\%$ when comparing SchNet with DFT-based reference results. Additionally, SchNet approach enabled 1.25 ns of path-integral MD, reducing the runtime by 3–4 orders of magnitude compared to DFT: from about 7 years to less than 7 h with much less computational resources (Schütt et al. 2018).

Another example using a GNN architecture to reduce the computational costs that constrain ab initio MD is a graph neural network FF (GNNFF). Here, atomic forces are predicted directly from automatically extracted structural features that are translationally invariant, but rotationally covariant, to the coordinate space of atomic positions without explicit calculation of PES. The latter contributes to faster prediction time, while the former contributes to higher accuracy. Indeed, the more recent GNNFF outperformed SchNet, in terms of force prediction accuracy and prediction speed. The mean absolute error of the Cartesian force components derived with GNNFF in respect to the DFT calculations was 0.036 eVÅ^{-1} when considering ISO17 database—a collection of MD trajectories of 129 organic isomers with the composition of C₇O₂H₁₀ and distinct structures. Furthermore, the mean absolute error for molecules there were not considered in training set was 0.088 eVÅ^{-1} , indicating that GNNFF is general enough to enable accurate prediction of forces for new molecular structures (with same chemical composition as the molecules used in training). GNNFF cannot be used to perform micro-canonical simulations or measure properties related to energy of the system because its forces are not derived from the system’s PES and are not energy conserving. However, GNNFF can be used in NVT MD simulations with set thermostat. GNNFF was trained on ab initio MD (AIMD) trajectory forces for smaller Li_{7-x}P₃S₁₁ system while AIMD simulations were also performed for a system with larger simulation cell and higher atom number. The comparison of GNNFF performance for “Small” and “Large” systems showed only 3% difference in accuracy, evaluated for each element separately. Next, NVT MD simulation for the Large Li_{7-x}P₃S₁₁ system, using atomic forces calculated by the GNNFF trained on the small system was performed and compared with AIMD. The comparison showed highly consistent radial and

angular distribution functions (Park et al. 2021). Depending on the element type and system, GNNFF also outperformed the DCF approach mentioned above (Mailoa et al. 2019).

Recently, ML and DL have been used for accurate prediction of FF parameters and topologies of small drug-like molecules. Specifically, the ML random forest regressor model was first trained on the atomic charges of molecules, calculated based on the DFT method, and then used to predict partial charges on molecules in a much shorter time (i.e., less than one minute). Meanwhile, neural network models (neural network classification model in Scikit-learn package) were developed to assign atom types, phase angles and periodicities (Mudedla et al. 2022).

TorchMD represents a framework for molecular simulations that provides mixed classical and ML potentials, where all force computations are expressed as PyTorch arrays and operations (Table 2). Additionally, TorchMD also enables learning and simulating NNPs (Doerr et al. 2021). The equivariant transformer (ET) architecture was also implemented in the TorchMD-NET framework (Table 2) where attention mechanism is used to predict QM properties (Thölke and De Fabritiis 2022). The application of TorchMD was first demonstrated with typical MD use cases (e.g. water box, alanine dipeptide, and trypsin with bound ligand benzamidine) for the evaluation of speed and energy conservation. Next, the QM9 data set was used to validate the training procedure, and finally a CG-simulation for miniprotein chignolin was performed using NNP trained on all-atom MD simulation data. Due to the lack of neighbor lists for nonbonded interactions in TorchMD, this method is 60-fold slower compared to ACEMD (a high-performance MD code) (Harvey et al. 2009; Galvelis et al. 2023). Neighbor list issue also makes TorchMD prohibitive for much larger systems, however it is still a suitable method for the treatment of CG systems (Doerr et al. 2021).

CG systems by themselves allow longer simulations of larger molecular systems. ML can also be applied in this area, where continuous filter convolutions on a GNN architecture (CGSchNet) (Husic et al. 2020) (Table 2) were used to obtain the ML CG FF. CGSchNet's performance was demonstrated on two model systems: capped alanine and the miniprotein chignolin. In both demonstrations CGSchNet simulations captured the same basins on two-dimensional free energy surface (FES) that were observed in FES calculated for initial all-atom simulation (Husic et al. 2020).

Another approach for calculating the transferable NNPs is ANI (Accurate Neural network engine for Molecular Energies) (Smith et al. 2017), which uses a modified version of the symmetry function to build single-atom atomic environment vectors as a molecular representation while training the DNN on QM DFT calculations (Table 2). ANI was used for creating potential called ANI-1, where GDB database was used in initial training. Despite ANI-1 being trained on small molecules counting only eight heavy atoms it demonstrated chemical accuracy compared to the reference DFT-based calculations also on much larger molecular systems (up to 54 atoms) suggesting its transferability (Smith et al. 2017).

JAX MD software that can also be used to conduct MD is based on JAX DL framework developed by Google, which couples a modified version of autograd (automatic obtaining of the gradient function through differentiation of a function) and TensorFlow. At its core, JAX MD comprises several primitive operations that can be used in molecular simulations. Building on these primitives, JAX MD further includes simulation environments and interaction potentials that can be integrated with several architectures of neural networks. The simulations in JAX MD are differentiable, allowing for meta-optimization through minimization of particle packings. With this JAX MD enables simulations with hundreds-of-thousands of particles on a single GPU (Schoenholz and Cubuk 2021).

One of the challenges in creating ML-based potential is that a large amount of reference data is required for neural network training to ensure accurate calculations. Therefore, the authors of ANI-1 have also provided access to a large computational DFT database of over 20 million off-equilibrium conformations of 57462 small molecules that can be used to compare current and future methods in the field of ML potential (Smith et al. 2017). Moreover, there are also additional open databases available for NNP training, namely QMugs and SPICE (Eastman et al. 2023a; Isert et al. 2022).

Several libraries of NNPs have been presented to date. One of them is a library of high-dimensional NNPs (n2p2 NNP package) that can be used together with MD packages such as LAMMPS (Thompson et al. 2022) (Table 2). This particular combination enabled the massively parallelized MD simulations of 2880 water molecules with a DFT NNP parametrization, achieving a speed of approximately 100 time steps per second (Singraber et al. 2019). One of the most influential trained NNP model is ANI-2x (Devereux et al. 2020), the extension of the ANI-1x model, trained to seven elements (H, C, N, O, F, Cl, S) and with improved prediction of molecular torsion profiles. Recently, a more current state-of-the-art NNP emerged-AIMNet2 (Anstine et al. 2023), that is applicable for structures of up to 14 chemical elements in neutral or charged states. These and similar NNPs can be utilized in the simulations via MD codes, like the above mentioned LAMMPS (Thompson et al. 2022; Singraber et al. 2019) or OpenMM (Eastman et al. 2023b).

4.2 Machine learning for improved sampling

In this section, we focus on the integration of ML to support enhanced and adaptive sampling methods in MD, in particular Deep-LDA, Deep-TDA, TPI-Deep-TDA, multitask learning, AlphaFold-inspired CVs, VAE-driven MSES, FUNN, DEEP-VES, REAP, VAEs, DeepDriveMD and NN-based generative models methods. By leveraging ML algorithms to learn from MD trajectories, enhanced and adaptive sampling methods can be further developed to explore and describe the high-dimensional conformational space of complex molecular systems more efficiently. In this respect ML techniques can, among others, aid in designing CVs, applying biases, predicting FES, and generating/selecting new starting conformations (Fig. 7).

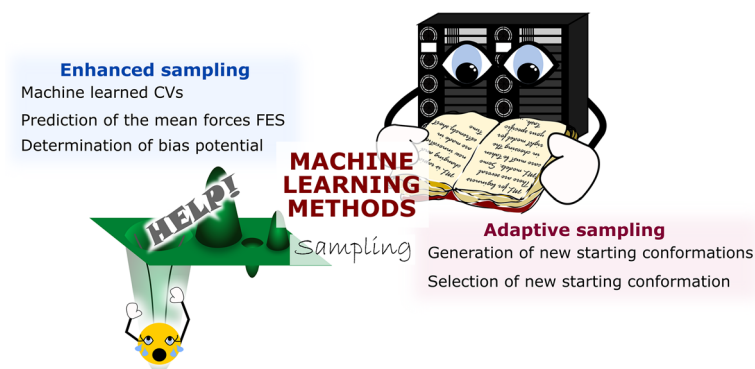


Fig. 7 Machine learning (ML) approaches to support enhanced and adaptive sampling during molecular dynamics simulations. (left) Machine-learned collective variables (CVs) can be used in enhanced sampling while mean forces of FES and bias potential can also be predicted and determined by ML. (right) In adaptive sampling, the generation or selection of new starting conformations can be addressed by ML

Appropriate CVs and FES accuracy are cornerstones for accurate and efficient CV-based enhanced sampling, and ML can be an excellent support to overcome both challenges. In principle, CVs represent a dimensional reduction of a high-dimensional space into a low-dimensional space, and methods for dimensional reduction are extensively studied by ML (Table 3) (Chen 2021).

CVs can be classified as high variance CVs or slow CVs. High variance CVs capture local motions that contribute significantly to the overall configurational variability (e.g., bond stretching, angle bending), whereas slow CVs capture large-scale conformational changes (e.g., protein folding, ligand binding) that contribute significantly to the overall kinetic content (Sidky et al. 2020).

A high-dimensional bias potential method (NN2B), was one of the first methods using neural networks to bias along CVs. This approach is based on two ML algorithms: (1) the nearest neighbor density estimator (NNDE), which estimates density and corresponding free energy, and (2) artificial neural network (ANN), which is used for bias potential approximation. NN2B performs short biased MD runs and updates a multidimensional bias potential iteratively based on the sampled distributions. Validation of this method was carried out with alanine and tryptophan polypeptides simulations in vacuum and water, where 2–8 dihedral angles are used as CVs (Galvelis and Sugita 2017).

CVs are traditionally designed as functions of only a few degrees of freedom (e.g., interatomic distances, torsion angles, and coordination numbers), which can lead to an inadequate representation of the complex behaviour taking place in a molecular system. Neural networks can be used to create CVs that capture many more variables and are based on the slowest modes of the systems. CVs can be designed using classification methods that can define coordinates to distinguish between different metastable states of interest, for which data from unbiased simulations in the different metastable states are used. A more classical approach to this is harmonic linear discriminant analysis (HLDA) (Mendels et al. 2018) while improved neural networks-based methods like deep linear discriminant analysis (Deep-LDA) (Bonati et al. 2020) and deep targeted discriminant analysis (Deep-TDA) (Trizio and Parrinello 2021) have been developed (Ray et al. 2023).

In Deep-LDA, a set of physical descriptors derived from various unbiased simulations of metastable basins is used as input to FFNN, where a nonlinear transformation is performed. In the last layer, linear discriminant analysis (LDA) is applied and the direction of maximal separation between classes is determined and ultimately the CV is obtained (Fig. 8) (Bonati et al. 2020). Deep-LDA method was implemented in the study of alanine dipeptide. This system has two metastable states which are well described by a pair of Ramachandran angles ϕ and ψ which represents almost ideal CVs for this particular system. However, a general set of descriptors (distance-based descriptors only) was used in this demonstration to create a situation similar to what one would face with more complicated system and to demonstrate Deep-LDA's ability to handle a large number of descriptors. Deep-LDA CV was then used in enhanced sampling which performed similar to simulations where the pair of Ramachandran angles are biased. The performance of Deep-LDA was also demonstrated with aldol reaction between vinyl alcohol and formaldehyde. Descriptors based on interatomic distances were calculated from the unbiased simulations of reactant and product. The resulting CV was used in on-the-fly probability enhanced sampling (OPES) (Invernizzi and Parrinello 2020; Hénin et al. 2022) where the direction along which the system was driven was correlated with the minimum free-energy path (Bonati et al. 2020).

The basic idea of the Deep-TDA method comes from Deep-LDA, with the main difference being that the linear step is skipped altogether and the CV is expressed directly

Table 3 DL models that can be used for collective variable identification

DL models for CV identification	Enhanced sampling techniques	Methods for analysis
Variational Autoencoder (VAE)	VAE-driven MSES (Moritsugu et al. 2014)	
Graph Neural Networks (GNNs)		
Feedforward Neural Network (FNN)	Deep Linear Discriminant Analysis (Deep-LDA) (Trizio and Parrinello 2021) Deep Targeted Discriminant Analysis (Deep-TDA) (Trizio and Parrinello 2021) Transition Path Informed Deep-TDA (TPI-Deep-TDA) (Ray et al. 2023) Force-biasing Using Neural Networks (FUNN) (Guo et al. 2018)	VAMPnets (Mardt et al. 2018) iVAMPnets (Mardt et al. 2022)
Multi-task learning (MTL)	MTL of CVs for Enhanced sampling of rare events (Sun et al. 2022)	

as a neural network output, using topological data analysis. In Deep-TDA, a set of physical descriptors that are invariant with respect to the symmetries of the system, and originate from different unbiased simulations of metastable basins represents a data set that is projected by FFNN into a low-dimensional representation. In this representation, data from different basins is discriminated, which is achieved by the loss function that ensures that the projected data follow a predetermined distribution (Fig. 8). The performance of this approach was tested in cases of alanine dipeptide in vacuum, hydrobromination of propene and in double proton transfer in diamino-benzoquinone. Using Deep-TDA CV and OPES encouraged transitions between the two metastable basins in alanine dipeptide and gave similar results as Deep-LDA CV. In other two cases the use of one-dimensional Deep-TDA CV was demonstrated to successfully promote different reaction steps (Trizio and Parrinello 2021).

CVs trained only to discriminate between metastable states are often not optimal for providing a meaningful description of the transition state region. The recently improved deep-TDA method, transition path informed deep-TDA (TPI-Deep-TDA), enables the identification of CVs that can distinguish between initial and final (metastable) states while can also pass through the lowest free energy transition pathways. In TPI-Deep-TDA first a set of descriptors is collected from the unbiased simulations in the metastable basins of the system, which are then used to generate standard Deep-TDA CV. The latter is used in a set of simulations performed using the on-the-fly probability enhanced sampling-flooding approach (OPES-flooding approach) (Ray et al. 2022). This enhanced sampling method avoids the deposition of bias in the transition regions, while the bias introduced in the non-excluded regions accelerates the probability of observing a transition (Ray et al. 2023). In this way, reactive trajectories are obtained, from which only configurations outside the metastable basins are collected, which are added to the original dataset used for deep-TDA CV generation. The target distribution of Deep-TDA is modified to account for the additional data. Then, FFNN is trained to generate TPI-Deep-TDA CV, which can be used in OPES (Invernizzi and Parrinello 2020) simulations to calculate the free energy landscape (Fig. 8).

The method was tested in folding/unfolding of chignolin and ligand-receptor binding study on G2 guest-OAMe octa-acid host. In comparison to Deep-TDA, TPI-Deep-TDA improved CV performance and sped up convergence. Thus, TPI-Deep-TDA CV follows

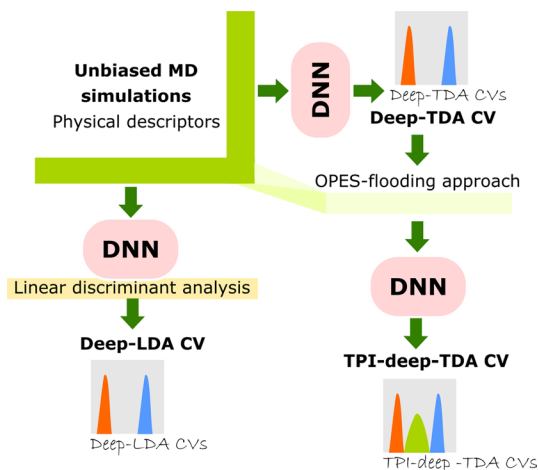


Fig. 8 Overview of the workflows of methods based on discriminant analysis: deep-LDA, deep-TDA, and TPI-deep-TDA methods (adapted from (Bonati et al. 2020; Ray et al. 2023; Trizio and Parrinello 2021)). In deep-LDA, physical descriptors from unbiased MD simulations of metastable states that do not allow discrimination between two states, are fed into a neural network. DL is used to uncover hidden components that allow discrimination between states. In the last layer, linear discriminant analysis is performed to obtain Deep-LDA CVs. The Deep-TDA method follows essentially the same workflow as the Deep-LDA method, but omits the linear discriminant analysis. In TPI-deep-TDA, deep-TDA CVs are used to guide MD simulations, e.g., using the OPES-flooding approach, where rare events such as transitions between metastable states are sampled. The descriptors obtained from the transition trajectories are then used to refine the CVs, which now include information about the transition path

the free energy gradient more closely than Deep-TDA CV, meaning that the sampled points are closer to the minimum free energy path. In the folding/unfolding study of chigoline TPI-Deep-TDA CV in biased simulation enabled estimation of the free energy difference between the folded and unfolded states in less than 200 ns of simulation time while similar convergence in unbiased simulations was reached only after $\sim 100 \mu\text{s}$. In case of biased simulations using Deep-TDA CV convergence was reached after 500 ns with larger uncertainty in the free energy difference compared to TPI-Deep-TDA CV. Fast convergence was also demonstrated in the last example case (Ray et al. 2023).

Recently, a multitask learning method was used to obtain CVs for enhanced sampling of rare events. This method addresses three tasks simultaneously: (i) dimensionality reduction in the form of a latent space is performed by a common upstream encoder, then separate downstream parts use this latent space to (ii) assign a basin class labels using a basin classifier, and (iii) predict potential energy using a potential energy predictor. The model is trained by combining short MD trajectories confined to the basins or containing transitions. A joint loss function is used for training, which combines the loss functions for each task. To obtain free energy landscapes, an iterative training procedure is followed. First, the initial configurations from unbiased MD simulations are collected and used for training. Then, the exploration of the configuration space is extended with biased simulations (umbrella sampling simulations) using the latent space as CV. Based on the obtained umbrella sampling simulations, short unbiased simulations are initiated and run until a known basin is reached, or until the predefined maximum time step is reached. Structures within the simulations are assigned destination basin label or label “unknown”, in the first

and second cases, respectively, and the assigned basin and potential energy labels are finally collected. Then, convergence is checked by calculating the misclassification rate, potential energy error, and free energy landscape on the latent space. In cases where the error is high and the free energy landscape is very different from the last iteration, convergence has not yet been achieved. In this case, the newly obtained basin classes and potential energy labels are added to the training data and the training and configuration explorations with CVs, followed by the recovery of new configuration labels, are repeated. When convergence is achieved, obtained CV is used to estimate the free energy with umbrella sampling or another enhanced sampling method (Fig. 9). The performance of this CV was tested on a model alanine dipeptide system in vacuum (Sun et al. 2022).

AlphaFold-inspired CVs were used in metadynamics and parallel tempering metadynamics simulations. AlphaFold 2 generates a tensor that stores the probability that two residues are at a given distance, which allows evaluation of fitness between a given protein conformation and the AlphaFold prediction. The level of this fitness was used to drive the MD simulations, biasing the system towards the conformations that matched the AlphaFold prediction. This allowed exploration of different conformations and prediction of their equilibrium probabilities (Fig. 10). The AlphaFold-inspired CV was used in cases of miniprotein Trp-cage and β -hairpin folding simulations, where the combination of parallel tempering with metadynamics enabled accurate prediction of FESs at different temperatures and observation of multiple folding events (Spiwok et al. 2022).

The MSES method mentioned in the first chapter was recently coupled with the ML approach - variational autoencoder (VAE). In VAE-driven MSES, two MD trajectories of the opened and closed ribose-binding protein were used to extract the distances between

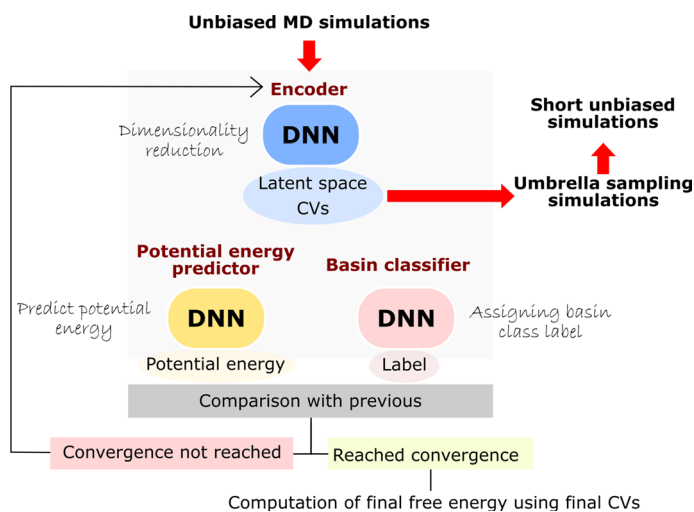


Fig. 9 A simplified workflow diagram of the multitask learning method for obtaining CVs (adapted from (Sun et al. 2022)). The general DL architecture includes an encoder that performs dimensionality reduction. The output of the encoder (latent space) then serves as input to a potential energy predictor and a basin classifier, that predict potential energy and assign basin class, respectively. A basic workflow is as follows: First, CVs are learned by the encoder using data from initial unbiased MD simulations. Then, an umbrella sampling technique is performed using the extracted CVs to obtain initial conformations from which short unbiased simulations are performed. The potential energy and basin class labels obtained from these simulations are then compared to the results of the previous iteration to assess the convergence status. If convergence is not achieved, the process is repeated, while if convergence is achieved, the final free energy is calculated using the final CVs

the residues as structural features, which were then normalized to represent values between 0 and 1. These normalized structural features were then used as inputs for encoding into latent space, from which the decoder can generate an output that is used for distance restraint applied to the all-atom simulations in MSES. This method has the advantage over classical MSES in that it eliminates the difficult and time-consuming construction of a CG model and proper parameter selection (Moritsugu 2021).

Another example of the use of neural network for enhanced sampling is the force-biasing using neural networks (FUNN) method (Guo et al. 2018). The adaptive biasing force method (ABF) is one of the CV-based methods that results in a biased FES with lower energy barriers by contrasting the calculated mean force of the native FES with the biasing force, resulting in shorter transition time scales that can be well sampled in simulation (Hénin et al. 2022). FUNN combined with ABF has been shown to improve the performance of ABF by an order or two in computer time. The combination uses the estimates of mean force stored on a grid, as in ABF, to train the neural network, to generate a continuous estimate of mean force over the entire space of CVs, even in regions that may not have been explored. These estimates are then used to calculate bias forces that are eventually applied to the ABF method to drive the simulation. FUNN approach was tested on model alanine dipeptide in explicit water, and Rouse modes of a CG ideal Gaussian polymer chain (Guo et al. 2018). FUNN is part of a Software Suite for advanced general ensemble simulations (SSAGES) framework that works with multiple MD engines and contains a variety of CVs and advanced sampling methods (Sidky et al. 2018) (Table 4).

In DEEP-VES method (Bonati et al. 2019) a variationally enhanced sampling (VES) (Valsson and Parrinello 2014) is used in combination with DL approach. In VES, a convex functional of the bias potential is introduced that includes a chosen target probability distribution related to CVs. Variational principle is then used to minimize this functional and acquire bias potential which is related in a simple way to the FES (Valsson

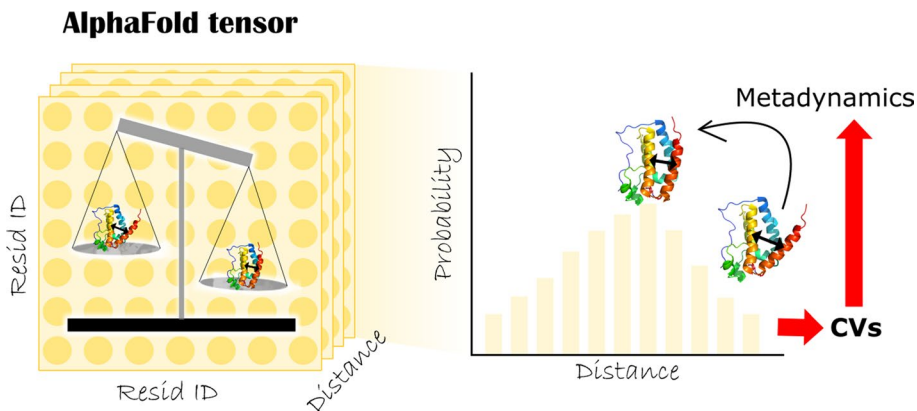


Fig. 10 The representation of AlphaFold-inspired CVs (adapted from (Spiwok et al. 2022)). AlphaFold generates tensors with information about the probability that two residues are at a given distance. The fitness between the given conformation and the AlphaFold prediction is then used as CV to guide MD simulations

and Parrinello 2014; Bonati et al. 2019). In DEEP-VES, however, the bias potential is determined with a DNN that uses the chosen CVs as inputs. The method was applied to

several systems including alanine dipeptide and alanine tetrapeptide in vacuum, as well as phase transition of silicon from liquid to solid (Bonati et al. 2019).

ML can also be applied to adaptive sampling, for which the REinforcement learning based Adaptive samPLing (REAP) algorithm has been developed. This approach takes advantage of reinforcement learning, which rewards sampling along important degrees of freedom and disregards sampling that does not facilitate exploration. First, several short MD simulations are run from a set of different initial conformations. Proteins are then clustered based on the selected CVs, while a new conformation is selected for a new simulation based on the reward function. The reward function depends on weights that reflect the importance of a CV, that may change in different basins, as well as on the sampling performance of the new simulation in the landscape compared to the current data. The main advantage of the method is on-the-fly estimation of the importance of CV, which makes it useful for systems with limited structural information. REAP method's performance was tested on model alanine dipeptide as well as Src kinase, where it demonstrated faster exploration of conformational space compared to a single continuous MD simulation or adaptive sampling technique (Shamsi et al. 2018).

In addition, VAEs can be used to explore the conformational space of proteins through a MD simulation, while the learned latent space can be used to generate unsampled protein conformations. These new conformations are then used as restarters for new MD simulations, greatly speeding up the sampling process and helping to identify hidden spaces. Enzyme adenosine kinase was used to study the transition between closed and open states. After a series of MD simulations were performed using crystal structures of the closed and open protein, the obtained trajectories were subsequently used for model training. Random points in the latent space were decoded into new conformations, that were used as new starting points for additional MD simulations. The initial and new trajectories captured a complete transition between closed and open conformations (Tian et al. 2021).

DeepDriveMD is another method for accelerated sampling of the available conformational space that is not limited to any particular learning method (e.g., convolutional variational autoencoders (CVAE), but can support any DL-driven methods and high-performance computing simulations. In the presented workflow, the latent space is learned from performed MD simulations and used to drive adaptive simulations. More specifically, data about the conformational dynamics of the system is collected during MD simulations, which is then used to train a ML model. This model learns a low-dimensional

Table 4 Selected tools for ML-supported collective variable generation and enhanced sampling

Name	Short description
mlcvs: ML Collective Variables (Chollet et al. 2015)	A Python package for creating collective variables from molecular simulation data, useful for enhanced sampling and dimensionality reduction in ML applications.
SSAGES: Software Suite for Advanced General Ensemble Simulations (Sidky et al. 2018)	A suite of software tools for performing advanced ensemble simulations in MD, including rare event sampling, free energy calculations, and more.

representation of the conformational space that can be used to generate new starting points for subsequent MD simulations, and it can also be used to terminate unproductive MD simulations that have become stuck in metastable states. To demonstrate its performance, DeepDriveMD was applied to investigate folding of Fs-peptide and villin head piece. Compared to traditional MD-based approaches, ML based approach offers about two-fold gain in the effective performance in sampling the folded states (Lee et al. 2019).

Conformational ensembles of a studied system can also be modeled directly with NN-based generative models trained on a dataset of molecular conformations from MD simulations. Recently, idpGAN was developed to generate physically realistic conformational ensembles of proteins. For this purpose, a conditional generative adversarial network (GAN) was used, whose generator is based on a transformer architecture and uses the peptide sequence as a condition. The model was trained using MD simulation data of intrinsically disordered peptides of varying lengths or α -synuclein. idpGAN was successful in predicting sequence-dependent ensembles for sequences of peptides not present in the training set, demonstrating transferability beyond the limited training data (Janson et al. 2023).

So far, we have focused on discussing methods that are based on CVs, especially neural network-based CVs. They represent the most widely used approach of using ML for enhanced sampling. Another conceptually different paradigm that was developed to address the sampling challenges in MD, are Boltzmann generators. These can generate unbiased equilibrium samples from diverse metastable states. Their advantage, compared to the more traditional enhanced sampling methods, is that they do not require predefined reaction coordinates/variables to steer between these states. Structures from metastable states are used to determine their free-energy differences, enabling the comparison of their stabilities. Furthermore, these generators can even identify physically realistic, low-energy transition pathways through linear interpolations in latent space (Noé et al. 2019).

4.3 Machine learning for improved analysis of simulation data

In this section we focus on the applicability of ML to better classify simulated models from which descriptors for a given class can be obtained as well as examine their utility to assist the MSM (Fig. 11). Once the trajectory is obtained by a MD simulation, challenges of extracting the relevant information and interpreting it arise. As we mentioned, there are several methods available, however the interpretation is not always trivial. Automatic extraction of statistically relevant information is generally aimed for in this context (Mardt et al. 2018). Application of ML approaches promises less manual intervention required to analyze this vast amount of generated data of the molecular system under investigation, and often provide better accuracy and ease or expand the interpretation of data.

DL has been successfully used, for example, to determine the enzymatic (enantio) selectivity of an ω -transaminase toward a range of ligands, without the need for hand-crafted criteria. Supervised CNN and semi-supervised RNN (long short-term memory, LSTM) architectures were used for this task. For training (80%) and validation (20%), a dataset consisting of 100 examples per class (with the preferred/non-preferred enantiomer) and per ligand was used, giving a total of 9800 examples. This dataset was obtained by a combination of ligand docking and running short MD simulations of the docked complexes. In addition to accurate classification, CNN also provided a visual description of the decision process, allowing the identification of descriptors that guided the classification. These descriptors represent geometric criteria that define binding

poses or identify interesting events in the trajectories that could characterize the classes (Ramírez-Palacios and Marrink 2023).

Another example of analyzing MD trajectories by performing classification task using the DL approach was performed in the context of G protein-coupled receptor (GPCR) structure and function. The main objective was to identify ligand-dependent conformational ensemble differences in GPCRs. Two receptor families were used, serotonin receptor subtype 2 A (5-HT_{2A}R) and dopamine receptor subtype D2 (D₂R), to which full, partial, or inverse agonists were bound. First, the MD trajectories were pre-processed to eliminate positional and orientational information. Then, the trajectories were transformed (frame by frame) into pixel representations that could be read by deep neural networks, where the values of red, green, and blue (RGB) corresponded to the x, y, and z coordinates of the atoms. Convolution neural network (CNN) was used for the classification task, from which the determinants of the classification were extracted. CNN was first trained, validated, and tested for 5-HT_{2A}R complexes, while the generalizability of the methods was subsequently tested for D₂R complexes. The molecular features that contributed most to the classification decision were then identified based on a sensitivity analysis approach in the category of visual saliency (Plante et al. 2019).

A popular approach for trajectory analysis is the application of MSMs, which allows the integration of short, distributed MD simulations into models of long-timescale molecular kinetics. This method can provide information about the probabilities for the occurrence of a particular state and the probabilities for the transition from one state to another. We should mention also here that MSM can be used in enhanced sampling to bias the simulation towards the conformational states or transitions of interest or in adaptive sampling. The typical pipeline for MSM would be as follows: (i) featurization; MD coordinates are aligned or transformed to internal coordinates, (ii) dimension reduction; reduction to much less slow CVs (e.g. variational approach, conformation dynamics, time-lagged independent component analysis (TICA), blind source separation, dynamic mode decomposition), (iii) scaling to a metric space, which is later (iv) discretized by clustering the projected data into discrete states, (v) obtaining the transition matrix describing the transition probabilities between discrete states in some

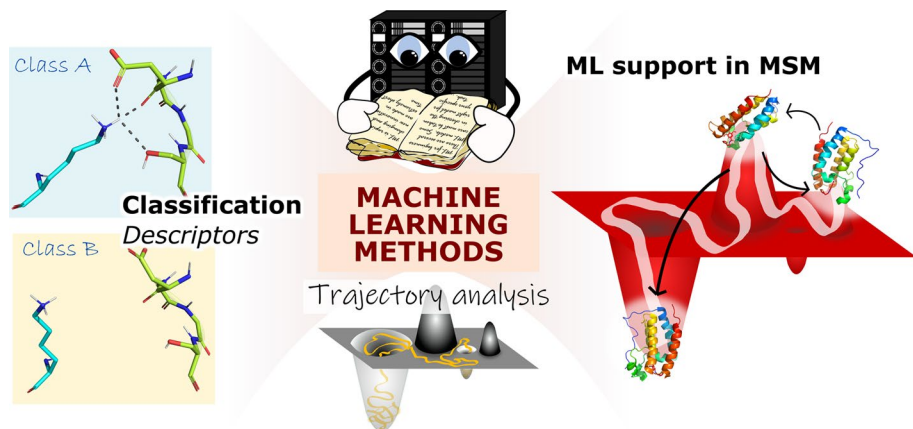


Fig. 11 ML to support efficient analysis of MD trajectories, namely classification of simulated models to obtain descriptors for a given class and assistance in Markov state modelling (MSM)

lag time τ or building a Koopman model follows before finally (vi) coarsening the MSM to a few states (Mardt et al. 2018; Lazim et al. 2020).

In the scope of Markov state modelling, software, such as PyEMMA (Scherer et al. 2015), has been developed that follows the pipeline described. However, the optimal pipeline for Markov modelling varies from case to case, and the variational approach for conformation dynamics method (VAC) can guide modelers by providing them with scores that compare the accuracy of a kinetic model with the unknown MD operator responsible for the true kinetics in the data (Mardt et al. 2018). VAC has been incorporated into the variational approach for Markov processes (VAMP), which allows identification of optimal feature mappings and optimal Markovian models of dynamics from given trajectories (Wu and Noé 2020).

Finally, a VAMPnets was developed with the implementation of AI (Table 5). This approach uses the DL framework, which replaces the MSM-based processing pipeline and encodes the entire mapping of coordinates to Markov states. Here, two FFNNs architectures were used, the first processing the coordinates of the system at time t and the second processing the coordinates of the system at time step $t + \tau$. The outputs of the two network lobes were then merged and a variational score was calculated, which was then maximized to optimize the network. Similar to the classical approach, the main derivatives of the VAMPnets are easily interpretable kinetic models with few states (Mardt et al. 2018).

For larger molecular systems, the number of independent subsystems and metastable states increases, so that capturing all the different global states becomes problematic due to a combinatorial explosion. This problem has been addressed by the proposal of independent Markov decomposition (IMD) (Hempel et al. 2021). Here, the system under study is first decomposed into subsystems for which MSMs are calculated independently and can be later coupled in order to obtain the behavior of the global system. There is no general rule for how to define protein subsystems. However, a dependency score was proposed that assess the coupling of two substructures, provides information on the quality of the IMD model approximation, and helps to determine the optimal partitioning of unknown systems (Hempel et al. 2021).

IMD was then extended by combining it with VAMPnets resulting in so-called iVAMPnets (Table 5). Here, the decomposition into subdomains and their individual MSMs are learned simultaneously. Furthermore, a training objective was set to quantify how well a given decomposition of the molecular system into independent subdomains with Markovian dynamics approximates the overall dynamics. However, learning of dynamical coupling of Markovian subdomains remains an open question (Mardt et al. 2022).

Table 5 Selected tools for ML-supported analysis of MD trajectory

Name	Short description
VAMPnets (Mardt et al. 2018)	VAMPnets are DL models for analyzing time-series data that use variational inference to estimate the best Markov model.
iVAMPnets (Mardt et al. 2022)	iVAMPnets are a variation of VAMPnets that can handle incomplete time-series data by using a different variational inference scheme that accounts for missing data.
Deeptimes (Hoffmann et al. 2021)	Deeptime is a Python library for time-series analysis that includes implementations of VAMPnets, iVAMPnets, and other models, as well as tools for preprocessing data and visualizing results.

5 AI integration into molecular dynamics simulations: limitations, challenges, impact, and future trends

The integration of ML in molecular simulations has resulted in numerous success stories and it is expected to make even greater advances in the future. However, there are some notable challenges that can hinder this progress. ML-based FF have generally been developed and used only for small molecules and ordered solutions (Noé et al. 2020). Therefore, ML-FF is currently not at the stage for straightforward application in simulations of large systems, such as protein complexes. In addition, ML approaches learn the physical shape of the PES from the sample data provided, so based on the ML potentials is limited by input reference data. The complexity of the configuration space presents a challenge in ensuring that the reference data is not biased or incomplete. Consequently, relatively large reference data sets are required to construct meaningful ML potentials, making the method computationally intensive and time consuming, especially given the time-consuming nature of MD simulations. Although ML potentials speed up the simulations, they do not provide new insights beyond those observed, and may even have been hidden in the training set (Behler 2016).

Overall, neural networks perform well in tasks where interpolation of data is carried out, but perform poorly when used for extrapolations (Ray et al. 2023). Some other challenges associated with ML methods also include overfitting and the computational effort required for model training (Fig. 12). There are some regularization methods that can be used to circumvent the overfitting issue (Latif et al. 2021).

More generally, AI technology has indeed finally made the long-awaited breakthrough that has impressed scientists and the public, and expectations are high in virtually all areas. Therefore, it is very easy to (un)intentionally misuse AI-, ML-based approaches and push them into certain areas where they ultimately serve merely as buzzwords and do not convey any advantages over the “traditional” well established methods. This can be potentially risky, as these AI-based methods fail to deliver the expected improvements and thus discourage end users from adopting them. Combined with the potentially slow progress of AI in certain areas, unrealistic expectations, and the mistrust that normally exists, this can ultimately hinder the development of the field. A critique was, for example, made for ML-supported CV identification, which highlighted the use of ML algorithms for the CV selection of systems that can be represented by simple geometric CVs (Bhakat 2022). And while it is true that selecting appropriate CVs can be challenging for novices, it is also worth noting that it is a streamlined and well-defined process and that the “automatic identification” that AI methods promise can be a trap that leads to uninterpretable and poor CV selection (Bhakat 2022).

Another noteworthy topic would also be the commonly perceived black-box nature of the ML models, more precisely DL models. Interpretability is indeed an important aspect, because understanding why the model made a certain prediction is crucial for predictions credibility and gaining knowledge on connections between certain attributes and functions (e.g. identifying key structural features distinguishing between activated and inhibited enzyme or between different metastable states). Transparency can also help to uncover hidden biases or errors in the model and assist in model development and optimization. For this purpose, interpretable ML (IML), or explainable AI could be exploited. There are two major groups of IML: (1) model-specific approaches that use the knowledge of a certain interpretable model for understanding predictions, and (2) model-agnostic methods that assess the predictive response of DL model (Bai et al. 2022). Some of these methods have

already been implemented in MD field (Xie et al. 2023; Vandermause et al. 2020; Li et al. 2022).

Furthermore, recent reviews by Zang et al. highlighted several other issues that, if properly addressed, would lead to a better fusion of AI and MD. First, the existing discrepancy between AI and MD programming calls for development of modules and libraries that are compatible with both frameworks. Next, very limited transferability of any generated deep potential energy surface (PES) model means that models previously trained by others cannot be directly reused. The potential of differentiable simulations and meta-optimization techniques is currently not fully exploited, and their development could lead to faster and more convenient applications. Also, two approaches of data-driven AI and physically informed AI (e.g. Boltzmann generators) should be used synergistically to improve generalizability and accelerate learning in AI-enhanced molecular simulations (Zhang et al. 2020, 2023). Finally, the utilization of ML methods to address the challenges in preparing molecular systems for MD is still in its early stages. This is a critical factor for achieving meaningful results, as it facilitates among others accurate assignment of rotamers, protonation (pKa) (Johnston et al. 2023), and metal coordination states, as well as identifies inaccurately placed water molecules.

Despite these challenges ML has already established its potential to afford more efficient and comprehensive MD simulations. Prior to the simulations themselves, ML can help refine the experimental structures of the systems under study, improving their accuracy and providing more reliable starting points for MD simulations (Hiranuma et al. 2021). As discussed here, technologies of AI, especially ML, can help MD in speeding up simulations, enabling improved sampling of conformational space, and providing support for analysis of the trajectories obtained (Fig. 12). Additionally, AI-assisted MD simulations can better

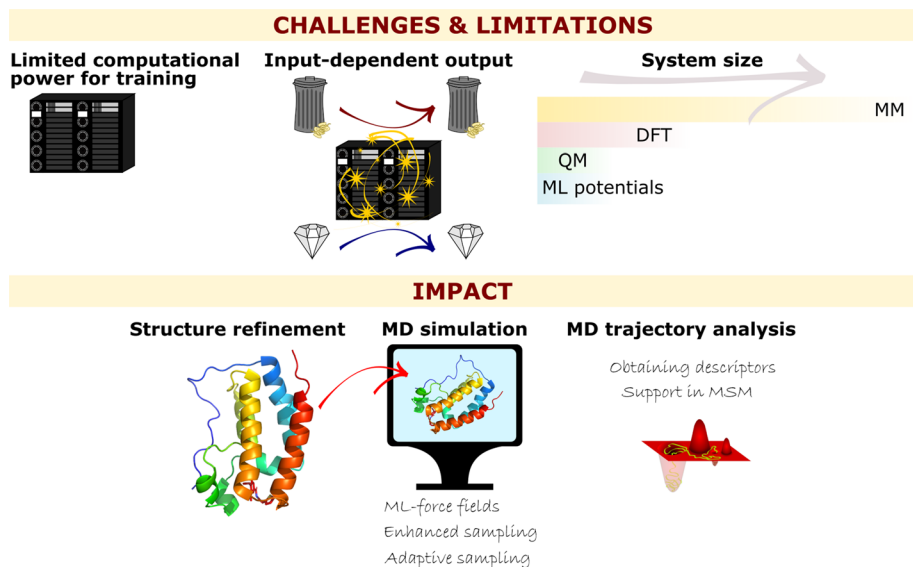


Fig. 12 Challenges, limitations, and impact of machine learning (ML) in molecular dynamics (MD). One of the limitations is the insufficient computational power for model training, making the ML potentials suitable only for small systems. Moreover, the quality of the results cannot and should not exceed the quality of the input data. Meanwhile, ML can help with structural refinement, MD simulations (enhanced and adaptive sampling and ML force field construction), and MD trajectory analysis

model the funnel energy landscape of proteins, leading to a better understanding of protein folding mechanisms (Zhang et al. 2019). They could also eliminate the need to calculate gradients of potential energy functions (i.e. atomic forces) and enable more comprehensive treatment of high-dimensional data to better construct reaction coordinates. All these advances suggest that ML is heralding a new development phase in MD simulations for addressing long-standing challenges in chemistry and biophysics (Zhang et al. 2020, 2023).

However, these techniques still need to further demonstrate their superiority over traditional MD methods, especially for larger simulated systems, while in the meantime caution is needed in deciding which method to use for a given research task. We believe that introduction of AI technology in MD simulations has surpassed the phase of “trough of disillusionment” of the well-known Gartner Hype Cycle, and is now on the “Slope of enlightenment” where the benefits of technology are clearly visible. We hope that AI-MD hybrid technologies will reach the full “Plateau of productivity” in the forthcoming years. One of the very last challenges in the AI-integration into MD simulations would be imparting of these methods to the scientists who apply MD simulations to study specific systems. Our review thus aims to bridge the gap between method developers and such scientists to inform fellow researchers on the latest advances in MD and how they can be applied to their specific systems.

Acknowledgements JB, AP, EP and ML thank the Slovenian Research Agency (grants P1-0017, P1-0012, J1-3019, J1-4402, N1-0300 and young researcher’s program number 39012) for the financial support.

Author contributions E.P.: Investigation; writing—original draft; writing—review and editing. M.L.: writing—original draft. A.P.: Conceptualization; supervision; writing—review and editing. J.B.: Conceptualization; supervision; writing—review and editing. All authors reviewed the manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: a system for large-scale machine learning. in *Osdi*, pp. 265–283
- Abrams C, Bussi G (2013) Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration. *Entropy* 16(1):163–199. <https://doi.org/10.3390/e16010163>
- Aggarwal A, Mittal M, Battineni G (2021) Generative adversarial network: an overview of theory and applications. *Int J Inf Manage* 1(1):100004. <https://doi.org/10.1016/j.jjimei.2020.100004>
- Alder BJ, Wainwright TE (1957) Phase transition for a hard sphere system. *J Chem Phys* 27(5):1208. <https://doi.org/10.1063/1.1743957>
- Ali M (2020) Home-PyCaret. Available at: <https://pycaret.org/>
- Anstine D, Zubatyuk R, Isayev O (2023) AIMNet2: a neural network potential to meet your neutral, charged, organic, and elemental-organic needs. <https://doi.org/10.26434/chemrxiv-2023-296ch>

- Awasthi S, Kapil V, Nair NN (2016) Sampling free energy surfaces as slices by combining umbrella sampling and metadynamics. *J Comput Chem* 37(16):1413–1424. <https://onlinelibrary.wiley.com/doi/10.1002/jcc.24349>
- Bai Q, Liu S, Tian Y, Xu T, Banegas-Luna AJ, Pérez-Sánchez H, Huang J, Liu H, Yao X (2022) Application advances of deep learning methods for de novo drug design and molecular dynamics simulation. *Comput Mol Sci* 12(3):e1581. <https://doi.org/10.1002/wcms.1581>
- Baltrukovich H, Podlowska S (2022) From data to knowledge: systematic review of tools for automatic analysis of molecular dynamics output. *Front Pharmacol* 13:844293. <https://doi.org/10.3389/fphar.2022.844293>
- Behler J (2011) Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J Chem Phys* 134(7):074106. <https://doi.org/10.1063/1.3553717>
- Behler J (2016) Perspective: machine learning potentials for atomistic simulations. *J Chem Phys* 145(17):170901. <https://doi.org/10.1063/1.4966192>
- Behler J (2021) Four generations of high-dimensional neural network potentials. *Chem Rev* 121(16):10037–10072. <https://doi.org/10.1021/acs.chemrev.0c00868>
- Behler J, Parrinello M (2007) Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys Rev Lett* 98(14):146401. <https://doi.org/10.1103/PhysRevLett.98.146401>
- Bernardi RC, Melo MC, Schulten K (2015) Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochim Biophys Acta* 1850(5):872–877. <https://doi.org/10.1016/j.bba-gen.2014.10.019>
- Best RB, Hummer G (2005) Reaction coordinates and rates from transition paths. *Proc Natl Acad Sci U.S.A.* 102(19):6732–6737. <https://doi.org/10.1073/pnas.0408098102>
- Bhakat S (2022) Collective variable discovery in the age of machine learning: reality, hype and everything in between. *RSC Adv* 12(38):25010–25024. <https://doi.org/10.1039/d2ra03660f>
- Bonati L, Zhang Y-Y, Parrinello M (2019) Neural networks-based variationally enhanced sampling. *PNAS* 116(36):17641–17647. <https://doi.org/10.1073/pnas.1907975116>
- Bonati L, Rizzi V, Parrinello M (2020) Data-driven collective variables for enhanced sampling. *J Phys Chem Lett* 11(8):2998–3004. <https://doi.org/10.1021/acs.jpcclett.0c00535>
- Bradbury J, Frostig R, Hawkins P, Johnson MJ, Leary C, Maclaurin D, Necula G, Paszke A, VanderPlas J, Wanderman-Milne S et al (2018) JAX: composable transformations of Python+NumPy programs. Available at: <http://github.com/google/jax>
- Chen M (2021) Collective variable-based enhanced sampling and machine learning. *Eur Phys J B* 94(10):211. <https://doi.org/10.1140/epjb/s10051-021-00220-w>
- Chen T, Li M, Li Y, Lin M, Wang N, Wang M, Xiao T, Xu B, Zhang C, Zhang Z (2015) Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*
- Chollet F et al. (2015) Keras. Available at: <https://github.com/fchollet/keras>
- Chong LT, Saglam AS, Zuckerman DM (2017) Path-sampling strategies for simulating rare events in biomolecular systems. *Curr Opin Struct Bio* 43:88–94. <https://doi.org/10.1016/j.sbi.2016.11.019>
- Cong S, Zhou Y (2023) A review of convolutional neural network architectures and their optimizations. *Artif Intell Rev* 56(3):1905–1969. <https://doi.org/10.1007/s10462-022-10213-5>
- Devereux C, Smith JS, Huddleston KK, Barros K, Zubatyuk R, Isayev O, Roitberg AE (2020) Extending the applicability of the ANI deep learning molecular potential to sulfur and halogens. *J Chem Theory Comput* 16(7):4192–4202. <https://doi.org/10.1021/acs.jctc.0c00121>
- Doerr S, De Fabritiis G (2012) On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations. *J Chem Theory Comput* 10(5):2064–2069. <https://doi.org/10.1021/ct400919u>
- Doerr S, Majewski M, Pérez A, Kramer A, Clementi C, Noe F, Giorgino T, De Fabritiis G (2021) Torchmd: a deep learning framework for molecular simulations. *J Chem Theory Comput* 17(4):2355–2363. <https://doi.org/10.1021/acs.jctc.0c01343>
- Durrant JD, McCammon JA (2011) Molecular dynamics simulations and drug discovery. *BMC Biol* 9(1):71. <https://doi.org/10.1186/1741-7007-9-71>
- Eastman P, Behara PK, Dotson DL, Galvelis R, Herr JE, Horton JT, Mao Y, Chodera JD, Pritchard BP, Wang Y et al (2023a) Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Sci Data* 10(1):11. <https://doi.org/10.1038/s41597-022-01882-6>
- Eastman P, Galvelis R, Peláez RP, Abreu CRA, Farr SE, Gallicchio E, Gorenko A, Henry MM, Hu F, Huang J et al (2023b) OpenMM 8: molecular dynamics simulation with machine learning potentials. <https://doi.org/10.48550/arXiv.2310.03121>

- Eclipse DeepLearning4j Development Team. DeepLearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. Available at: <http://deeplearning4j.org>. Accessed 12 2023
- Galvelis R, Sugita Y (2017) Neural network and nearest neighbor algorithms for enhancing sampling of molecular dynamics. *J Chem Theory Comput* 13(6):2489–2500. <https://doi.org/10.1021/acs.jctc.7b00188>
- Galvelis R, Varela-Rial A, Doerr S, Fino R, Eastman P, Markland TE, Chodera JD, De Fabritiis G (2023) NNP/MM: accelerating molecular dynamics simulations with machine learning potentials and molecular mechanics. *J Chem Inf Model* 63(18):5701–5708. <https://doi.org/10.1021/acs.jcim.3c00773>
- Goodfellow IJ, Bengio Y, Courville A (2016) Deep learning
- Guo AZ, Sevgen E, Sidky H, Whitmer JK, Hubbell JA, de Pablo JJ (2018) Adaptive enhanced sampling by force-biasing using neural networks. *J Chem Phys* 148(13):134108. <https://doi.org/10.1063/1.5020733>
- Harvey MJ, Giupponi G, De Fabritiis G (2009) ACEMD: accelerating biomolecular dynamics in the microsecond time scale. *J Chem Theory Comput* 5(6):1632–1639. <https://doi.org/10.1021/ct9000685>
- Hempel T, Del Razo MJ, Lee CT, Taylor BC, Amaro RE, Noé F (2021) Independent Markov decomposition: toward modeling kinetics of biomolecular complexes. *PNAS* 118(31):e2105230118. <https://doi.org/10.1073/pnas.2105230118>
- Hénin J, Lelièvre T, Shirts MR, Valsson O, Delemotte L (2022) Enhanced sampling methods for molecular dynamics simulations. *LivingJ Comp Mol Sci* 4(1):1583. <https://doi.org/10.33011/livecoms.4.1.1583>
- Himanen L, Jäger MO, Morooka EV, Canova FF, Ranawat YS, Gao DZ, Rinke P, Foster AS (2020) DScribe: library of descriptors for machine learning in materials science. *Comput Phys Commun* 247:106949. <https://doi.org/10.1016/j.cpc.2019.106949>
- Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
- Hiranuma N, Park H, Baek M, Anishchenko I, Dauparas J, Baker D (2021) Improved protein structure refinement guided by deep learning based accuracy estimation. *Nat Commun* 12(1):1340. <https://doi.org/10.1038/s41467-021-21511-x>
- Hoffmann M, Scherer M, Hempel T, Mardt A, de Silva B, Husic BE, Klus S, Wu H, Kutz N, Brunton SL et al (2021) Deeptime: a python library for machine learning dynamical models from time series data. *Mach Learn: Sci Technol* 3(1):015009. <https://doi.org/10.1088/2632-2153/ac3de0>
- Hruska E, Balasubramanian V, Lee H, Jha S, Clementi C (2020) Extensible and scalable adaptive sampling on supercomputers. *J Chem Theory Comput* 16(12):7915–7925. <https://doi.org/10.1021/acs.jctc.0c00991>
- Husic BE, Charron NE, Lemm D, Wang J, Pérez A, Majewski M, Krämer A, Chen Y, Olsson S, de Fabritiis G et al (2020) Coarse graining molecular dynamics with graph neural networks. *J Chem Phys* 153(19):194101. <https://doi.org/10.1063/5.0026133>
- Hwang T (2018) Computational power and the social impact of artificial intelligence. *arXiv preprint arXiv:1803.08971* <https://doi.org/10.48550/arXiv.1803.08971>
- Iftimie R, Minary P, Tuckerman ME (2005) Ab initio molecular dynamics: concepts, recent developments, and future trends. *Proc Natl Acad Sci U.S.A.* 102(19):6654–9. <https://doi.org/10.1073/pnas.0500193102>
- Invernizzi M, Parrinello M (2020) Rethinking metadynamics: from bias potentials to probability distributions. *J Phys Chem Lett* 11(7):2731–2736. <https://doi.org/10.1021/acs.jpcclett.0c00497>
- Iserl C, Atz K, Jiménez-Luna J, Schneider G (2022) QMugs, quantum mechanical properties of drug-like molecules. *Sci Data*. <https://doi.org/10.1038/s41597-022-01390-7>
- Janiesch C, Zszech P, Heinrich K (2021) Machine learning and deep learning. *Electron Markets* 31(3):685–695. <https://doi.org/10.1007/s12525-021-00475-2>
- Janson G, Valdes-Garcia G, Heo L, Feig M (2023) Direct generation of protein conformational ensembles via machine learning. *Nat Commun* 14(1):774. <https://doi.org/10.1038/s41467-023-36443-x>
- Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. in *arXiv preprint arXiv:1408.5093*
- Jing Z, Liu C, Cheng SY, Qi R, Walker BD, Piquemal J-P, Ren P (2019) Polarizable force fields for biomolecular simulations: recent advances and applications. *Annu Rev Biophys* 48:371–394. <https://doi.org/10.1146/annurev-biophys-070317-033349>
- Johnston RC, Yao K, Kaplan Z, Chelliah M, Leswing K, Seekins S, Watts S, Calkins D, Chief Elk J, Jerome SV et al (2023) Epik: pKa and protonation state prediction through machine learning. *J Chem Theory Comput* 19(8):2380–2388. <https://doi.org/10.1021/acs.jctc.3c00044>

- Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, \wZ\ idek A, Potapenko A et al (2021) Highly accurate protein structure prediction with AlphaFold. *Nature* 596(7873):583–589. <https://doi.org/10.1038/s41586-021-03819-2>
- Kmieciek S, Gront D, Kolinski M, Wieteska L, Dawid AE, Kolinski A (2016) Coarse-grained protein models and their applications. *Chem Rev* 116(14):7898–7936
- Kononov KA, Unarta IC, Cao S, Goonetilleke EC, Huang X (2021) Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au* 1(9):1330–1341. <https://doi.org/10.1021/jacsau.1c00254>
- Latif S, Driss M, Boulila W, Jamal SS, Idrees Z, Ahmad J et al (2021) Deep learning for the industrial internet of things (IIoT): a comprehensive survey of techniques, implementation frameworks, potential applications, and future directions. *Sensors* 21(22):7518. <https://doi.org/10.3390/s21227518>
- Lazim R, Suh D, Choi S (2020) Advances in molecular dynamics simulations and enhanced sampling methods for the study of protein systems. *Int J Mol Sci* 21(17):6339. <https://doi.org/10.3390/ijms21176339>
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
- Lee H, Turilli M, Jha S, Bhowmik D, Ma H, Ramanathan A (2019) DeepDriveMD: deep-learning driven adaptive molecular simulations for protein folding. 2019 IEEE/ACM Third Workshop Deep Learn Supercomput (DLS). <https://doi.org/10.1109/DLS49591.2019.00007>
- Li C, Liu J, Chen J, Yuan Y, Yu J, Gou Y, Guo Y, Pu X (2022) An interpretable convolutional neural network framework for analyzing molecular dynamics trajectories: a case study on functional states for G-protein-coupled receptors. *J Chem Inf Model* 62(6):1399–1410. <https://doi.org/10.1021/acs.jcim.2c00085>
- Liao Q (2020) Enhanced sampling and free energy calculations for protein simulations. *Prog Mol Biol Transl Sci* 170:177–213. <https://doi.org/10.1016/bs.pmbts.2020.01.006>
- Likhachev I, Balabaev N, Galzitskaya O (2016) Available instruments for analyzing molecular dynamics trajectories. *Open Biochem J* 10:1. <https://doi.org/10.2174/1874091X01610010001>
- Lin T, Wang Y, Liu X, Qiu X (2022) A survey of transformers. *AI Open* 3:111–132. <https://doi.org/10.1016/j.aiopen.2022.10.001>
- Mailoa JP, Kornbluth M, Batzner S, Samsonidze G, Lam ST, Vandermause J, Ablitt C, Molinari N, Kozinsky B (2019) A fast neural network approach for direct covariant forces prediction in complex multi-element extended systems. *Nat Mach Intell* 1(10):471–479. <https://doi.org/10.1038/s42256-019-0098-0>
- Mardt A, Pasquali L, Wu H, Noé F (2018) VAMPnets for deep learning of molecular kinetics. *Nat Commun* 9(1):1–11. <https://doi.org/10.1038/s41467-017-02388-1>
- Mardt A, Hempel T, Clementi C, Noé F (2022) Deep learning to decompose macromolecules into independent Markovian domains. *Nat Commun* 13:7101. <https://doi.org/10.1038/s41467-022-34603-z>
- McCammom JA, Gelin BR, Karplus M (1977) Dynamics of folded proteins. *Nature* 267(5612):585–590. <https://doi.org/10.1038/267585a0>
- McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133. <https://doi.org/10.1007/BF02478259>
- Medsker L, Jain LC (1999) Recurrent neural networks: design and applications. CRC, Boca Raton
- Melcr J, Piquemal J-P (2019) Accurate biomolecular simulations account for electronic polarization. *Front Mol Biosci* 6:143. <https://doi.org/10.3389/fmolb.2019.00143>
- Mendels D, Piccini G, Parrinello M (2018) Collective variables from local fluctuations. *J Phys Chem Lett* 9(11):2776–2781. <https://doi.org/10.1021/acs.jpcllett.8b00733>
- Microsoft Cognitive Toolkit (2016) *Microsoft Corporation* Available at: <https://github.com/Microsoft/CNTK>. Accessed 3, 2023
- Moritsugu K (2021) Multiscale enhanced sampling using machine learning. *Life* 11(10):1076. <https://doi.org/10.3390/life11101076>
- Moritsugu K, Terada T, Kidera A (2014) Multiscale enhanced sampling driven by multiple coarse-grained models. *Chem Phys Lett* 616:20–24. <https://doi.org/10.1016/j.cplett.2014.10.009>
- Mouvet F, Villard J, Bolnykh V, Rothlisberger U (2022) Recent advances in first-principles based molecular dynamics. *Acc Chem Res* 55(3):221–230. <https://doi.org/10.1021/acs.accounts.1c00503>
- Mudedla SK, Braka A, Wu S (2022) Quantum-based machine learning and AI models to generate force field parameters for drug-like small molecules. *Front Mol Biosci*. <https://doi.org/10.3389/fmolb.2022.1002535>
- Mukhamediev RI, Popova Y, Kuchin Y, Zaitseva E, Kalimoldayev A, Symagulov A, Levashenko V, Abdoldina F, Gopejkeno V, Yakunin K et al (2022) Review of artificial intelligence and machine learning technologies: classification, restrictions, opportunities and challenges. *Mathematics* 10(15):2552. <https://doi.org/10.3390/math10152552>

- Noé F, Olsson S, Köhler J, Wu H (2019) Boltzmann generators: sampling equilibrium states of many-body systems with deep learning. *Science* 365(6457):eaaw1147. <https://doi.org/10.1126/science.aaw1147>
- Noé F, Tkatchenko A, Müller K-R, Clementi C (2020) Machine learning for molecular simulation. *Annu Rev Phys Chem* 71:361–390. <https://doi.org/10.1146/annurev-physchem-042018-052331>
- Ollitrault PJ, Miessen A, Tavernelli I (2021) Molecular quantum dynamics: a quantum computing perspective. *Acc Chem Res* 54(23):4229–4238. <https://doi.org/10.1021/acs.accounts.1c00514>
- Park CW, Kornbluth M, Vandermause J, Wolverson C, Kozinsky B, Mailoa JP (2021) Accurate and scalable graph neural network force field and molecular dynamics with direct force architecture. *Npj Comput Mater* 7(1):73. <https://doi.org/10.1038/s41524-021-00543-3>
- Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al (2019) Pytorch: an imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst* 32:8024–8035
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
- Pérez A, Herrera-Nieto P, Doerr S, De Fabritiis G (2020) AdaptiveBandit: a multi-armed bandit framework for adaptive sampling in molecular simulations. *J Chem Theory Comput* 16(7):4685–4693. <https://doi.org/10.1021/acs.jctc.0c00205>
- Plante A, Shore DM, Morra G, Khelashvili G, Weinstein H (2019) A machine learning approach for the discovery of ligand-specific functional mechanisms of GPCRs. *Molecules* 24(11):2097. <https://doi.org/10.3390/molecules24112097>
- Ramírez-Palacios C, Marrink SJ (2023) Computational prediction of ω -transaminase selectivity by deep learning analysis of molecular dynamics trajectories. *QRB Discov* 4:e1. <https://doi.org/10.1017/qrd.2022.22>
- Ray D, Ansari N, Rizzi V, Invernizzi M, Parrinello M (2022) Rare event kinetics from adaptive bias enhanced sampling. *J Chem Theory Comput* 18(11):6500–6509. <https://doi.org/10.1021/acs.jctc.2c00806>
- Ray D, Trizio E, Parrinello M (2023) Deep learning collective variables from transition path ensemble. *arXiv preprint arXiv:2303.01629* <https://doi.org/10.48550/arXiv.2303.01629>
- Retto J (2017) Sophia, first citizen robot of the world. ResearchGate URL: <https://www.researchgate.net>
- Rudolph J, Tan S, Tan S (2023) ChatGPT: bullshit spewer or the end of traditional assessments in higher education? *J Educ Teach*. <https://doi.org/10.37074/jalt.2023.6.1.9>
- Sanchez-Lengeling B, Reif E, Pearce A, Wiltchko AB (2021) A gentle introduction to graph neural networks. *Distill*. <https://doi.org/10.23915/distill.00033>
- Scherer MK, Trendelkamp-Schroer B, Paul F, Pérez-Hernández G, Hoffmann M, Plattner N, Wehmeyer C, Prinz J-H, Noé F (2015) PyEMMA 2: a software package for estimation, validation, and analysis of Markov models. *J Chem Theory Comput* 11(11):5525–5542. <https://doi.org/10.1021/acs.jctc.5b00743>
- Schlick T, Portillo-Ledesma S (2021) Biomolecular modeling thrives in the age of technology. *Nat Comput Sci* 1(5):321–331. <https://doi.org/10.1038/s43588-021-00060-9>
- Schoenholz SS, Cubuk ED (2021) JAX, MD a framework for differentiable physics. *J Stat Mech*. <https://doi.org/10.1088/1742-5468/ac3ae9>
- Schütt KT, Sauceda HE, Kindermans P-J, Tkatchenko A, Müller K-R (2018) Schnet-a deep learning architecture for molecules and materials. *J Chem Phys* 148(24):241722. <https://doi.org/10.1063/1.5019779>
- Shamsi Z, Cheng KJ, Shukla D (2018) Reinforcement learning based adaptive sampling: REAPing rewards by exploring protein conformational landscapes. *J Phys Chem B* 122(35):8386–8395. <https://doi.org/10.1021/acs.jpcc.8b06521>
- Sidky H, Colón YJ, Helfferich J, Sikora BJ, Bezik C, Chu W, Giberti F, Guo AZ, Jiang X, Lequieu J et al (2018) SSAGES: software suite for advanced general ensemble simulations. *J Chem Phys* 148(4):044104. <https://doi.org/10.1063/1.5008853>
- Sidky H, Chen W, Ferguson AL (2020) Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Mol Phys* 118(5):e1737742. <https://doi.org/10.1080/00268976.2020.1737742>
- Singraber A, Behler J, Dellago C (2019) Library-based LAMMPS implementation of high-dimensional neural network potentials. *J Chem Theory Comput* 15(3):1827–1840. <https://doi.org/10.1021/acs.jctc.8b00770>
- Smith JS, Isayev O, Roitberg AE (2017) ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem Sci* 8(4):3192. <https://doi.org/10.1039/c6sc05720a>
- Spiwok V, Kurečka M, Křenek A (2022) Collective variable for Metadynamics Derived from AlphaFold output. *Front Mol Biosci* 9:878133. <https://doi.org/10.3389/fmolb.2022.878133>

- Subramanian AK (2020) PyTorch-VAE. *GitHub repository* Available at: <https://github.com/AntixK/PyTorch-VAE>. Accessed 3, 2023
- Sun L, Vandermause J, Batzner S, Xie Y, Clark D, Chen W, Kozinsky B (2022) Multitask machine learning of collective variables for enhanced sampling of rare events. *J Chem Theory Comput* 18(4):2341–2353. <https://doi.org/10.1021/acs.jctc.1c00143>
- Thompson AP, Aktulga HM, Berger R, Bolintineanu DS, Brown WM, Crozier PS, Veld PJ, Kohlmeyer A, Moore SG, Nguyen TD et al (2022) LAMMPS—a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput Phys Commun*. <https://doi.org/10.1016/j.cpc.2021.108171>
- Thölke P, De Fabritiis G (2022) Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*. <https://doi.org/10.48550/arXiv.2202.02541>
- Tian H, Jiang X, Trozzi F, Xiao S, Larson EC, Tao P (2021) Explore protein conformational space with variational autoencoder. *Front Mol Biosci* 8:781635. <https://doi.org/10.3389/fmolb.2021.781635>
- Tokui S, Okuta R, Akiba T, Niitani Y, Ogawa T, Saito S, Suzuki S, Uenishi K, Vogel B, Yamazaki Vincent H (2019) Chainer: a deep learning framework for accelerating the research cycle. *Proc 25th ACM SIGKDD Int Conf Know Discov Data Mining, 2002–2011*. <https://doi.org/10.1145/3292500.3330756>
- Torrie GM, Valleau JP (1977) Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling. *J Comput Phys* 23(2):187–199. [https://doi.org/10.1016/0021-9991\(77\)90121-8](https://doi.org/10.1016/0021-9991(77)90121-8)
- Trizio E, Parrinello M (2021) From enhanced sampling to reaction profiles. *J Phys Chem Lett* 12(35):8621–8626. <https://doi.org/10.1021/acs.jpcclett.1c02317>
- Unke OT, Chmiela S, Sauceda HE, Gastegger M, Poltavsky I, Schütt KT, Tkatchenko A, Müller K-R (2021) Machine learning force fields. *Chem Rev* 121(16):10142–10186. <https://doi.org/10.1021/acs.chemrev.0c01111>
- Upadhyay V, Sastry P (2019) An overview of restricted Boltzmann machines. *J Indian Inst Sci* 99:225–236. <https://doi.org/10.1007/s41745-019-0102-z>
- Valsson O, Parrinello M (2014) Variational approach to enhanced sampling and free energy calculations. *Phys Rev Lett* 113(9):090601. <https://doi.org/10.1103/PhysRevLett.113.090601>
- Vandermause J, Torrisi SB, Batzner S, Xie Y, Sun L, Kolpak AM, Kozinsky B (2020) On-the-fly active learning of interpretable bayesian force fields for atomistic rare events. *NPJ Comput Mater* 6(1):20. <https://doi.org/10.1038/s41524-020-0283-z>
- Vapnik VN (1999) An overview of statistical learning theory. *IEEE Trans Neural netw* 10(5):988–999. <https://doi.org/10.1109/72.788640>
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Łukasz, Polosukhin I (2017) Attention is all you need. *Adv Neural Inf Process Syst* 30:5998–6008
- Vint D, Anderson M, Yang Y, Ilioudis C, Di Caterina G, Clemente C (2021) Automatic target recognition for low resolution foliage penetrating SAR images using CNNs and GANs. *Remote Sens* 13(4):596. <https://doi.org/10.3390/rs13040596>
- Wang H, Zhang L, Han J, Weinan E (2018) DeepPMD-kit: a deep learning package for many-body potential energy representation and molecular dynamics. *Comput Phys Commun* 228:178–184. <https://doi.org/10.1016/j.cpc.2018.03.016>
- Wu H, Noé F (2020) Variational approach for learning Markov processes from time series data. *J Nonlinear Sci* 30(1):23–66. <https://doi.org/10.1007/s00332-019-09567-y>
- Wu X, Xu L-Y, Li E-M, Dong G (2022) Application of molecular dynamics simulation in biomedicine. *Chem Biol Drug Des* 99(5):789–800. <https://doi.org/10.1111/cbdd.14038>
- Xie SR, Rupp M, Hennig RG (2023) Ultra-fast interpretable machine-learning potentials. *NPJ Comput Mater* 9(1):162. <https://doi.org/10.1038/s41524-023-01092-7>
- Xu Y, Liu X, Cao X, Huang C, Liu E, Qian S, Liu X, Wu Y, Dong F, Qiu C-W et al (2021) Artificial intelligence: a powerful paradigm for scientific research. *Innov* 2(4):100179. <https://doi.org/10.1016/j.xinn.2021.100179>
- Yang YI, Shao Q, Zhang J, Yang L, Gao YQ (2019) Enhanced sampling in molecular dynamics. *J Chem Phys* 151(7):070902. <https://doi.org/10.1063/1.5109531>
- Zhang C, Lu Y (2021) Study on artificial intelligence: the state of the art and future prospects. *J Industrial Inform Integr* 23:100224
- Zhang J, Lei Y-KCX, Zhang Z, Yang YI, Gao YQ (2019) Deep representation learning for complex free-energy landscapes. *J Phys Chem Lett* 10:5571–5576. <https://doi.org/10.1021/acs.jpcclett.9b02012>
- Zhang J, Lei Y-K, Zhang Z, Chang J, Li M, Han X, Yang L, Yang YI, Gao YQ (2020) A perspective on deep learning for molecular modeling and simulations. *J Phys Chem A* 124(34):6745–6763. <https://doi.org/10.1021/acs.jpca.0c04473>

- Zhang J, Chen D, Xia Y, Huang Y-P, Lin X, Han X, Ni N, Wang Z, Yu F, Yang L et al (2023) Artificial intelligence enhanced molecular simulations. *J Chem Theory Comput* 19(14):4338–4350. <https://doi.org/10.1021/acs.jctc.3c00214>
- Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. *AI Open* 1:57–81. <https://doi.org/10.1016/j.aiopen.2021.01.001>
- Zwier MC, Chong LT (2010) Reaching biological timescales with all-atom molecular dynamics simulations. *Curr Opin Pharmacol* 10(6):745–752. <https://doi.org/10.1016/j.coph.2010.09.008>

Publisher's Note Springer nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.