

Benchmark study on deep neural network potentials for small organic molecules

Rohit Modee  | Siddhartha Laghuvarapu | U. Deva Priyakumar 

Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad, India

Correspondence

U. Deva Priyakumar, Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India.
Email: deva@iiit.ac.in

Funding information

IHub-Data, IIIT Hyderabad; Tata Consultancy Services, Grant/Award Number: TCS RESEARCH SCHOLAR PROGRAM

Abstract

There has been tremendous advancement in machine learning (ML) applications in computational chemistry, particularly in neural network potentials (NNP). NNPs can approximate potential energy surface (PES) as a high dimensional function by learning from existing reference data, thereby circumventing the need to solve the electronic Schrödinger equation explicitly. As a result, ML accelerates chemical space exploration and property prediction compared to quantum mechanical methods. Novel ML methods have the potential to provide efficient means for predicting the properties of molecules. However, this potential has been limited by the lack of standard comparative evaluations. In this work, we compare four selected models, that is, ANI, PhysNet, SchNet, and BAND-NN, developed to represent the PES of small organic molecules. We evaluate these models for their accuracy and transferability on two different test sets (i) Small organic molecules of up to eight-heavy atoms on which ANI and SchNet achieve root mean square error (RMSE) of 0.55 and 0.60 kcal/mol, respectively. (ii) On random selection of molecules from the GDB-11 database with 10-heavy atoms, ANI achieves RMSE of 1.17 kcal/mol and SchNet achieves RMSE of 1.89 kcal/mol. We examine their ability to produce smooth meaningful surface by performing PES scans for bond stretch, angle bend, and dihedral rotations on relatively large molecules to assess their possible application in molecular dynamics simulations. We also evaluate their performance for yielding minimum energy structures via geometry optimization using various minimization algorithms. All these models were also able to accurately differentiate different isomers of the same empirical formula $C_{10}H_{20}$. ANI and PhysNet achieve an RMSE of 0.29 and 0.52 kcal/mol, respectively, on $C_{10}H_{20}$ isomers.

KEYWORDS

energy prediction, machine learning, neural network potentials, potential energy surface, small organic molecules

1 | INTRODUCTION

Accurate modeling of biological and chemical processes requires precise estimation of energies and physiochemical properties of molecules. Density functional theory (DFT) method, even though is very accurate and has a central role in computational chemistry, are

computationally prohibitive in high-throughput applications. On the other hand, machine learning (ML) algorithms, trained on known examples, can be used to predict the properties of new molecules at much reduced computational cost with comparable accuracy as DFT.^{1,2} ML algorithms such as generative adversarial networks,³ coupled with reinforced learning,^{4,5} have succeeded in finding new

molecules with the desired drug-like properties.^{6–8} ML models have been successfully trained and used as predictive models for interatomic potential energy surfaces (PES),^{9–14} molecular force field (FF),^{15,16} electron densities,¹⁷ density functionals,¹⁸ protein structure prediction,^{19–21} protein–protein interactions,²² material property prediction,^{23–25} retrosynthesis,²⁶ and drug discovery.^{6,7,27,28}

Researchers use molecular modeling methods to understand biological and chemical processes. Biological and chemical molecular modeling requires an accurate estimation of the energies and physiochemical properties of the molecules. DFT methods are popular quantum mechanical (QM) methods of choice for calculating accurate molecular energies and physiochemical properties. They are computationally expensive and are impractical for larger systems,²⁹ especially at a large scale. Hence, to model larger systems and to perform high-throughput exercises, classical FF are used with significantly reduced computational cost but with reduced accuracy compared to QM. FFs simplify the description of interatomic interactions by summing components of the bonded, angular, dihedral, and nonbonded contributions fitted to a simple analytical form. FFs often do not account for polarizability. Furthermore, FFs have to be parameterized for modeling various biological and chemical processes.^{30–34} Hence, they lack “out-of-the-box” transferability.

In recent years, ML methods have been used to circumvent the problem of solving Schrödinger equation altogether. ML methods learn the high dimensional function (HDF), f by training the computer on large amounts of precomputed data (properties like energy, force, etc.). The ability to learn the HDF (f) which can then be used to map function $f(Z_i, r_i) \rightarrow E$, which returns the energy (E). Hence, to estimate properties for unknown compounds or structures, one needs to provide only the given set of nuclear charges Z_i and atomic positions r_i of the system to get the required property. The HDF is similar to PES in the sense that when given a set of nuclear charges Z_i and atomic positions r_i of a molecule, it can predict the value of the energy.³⁵

A host of supervised ML algorithms are available to train and optimize HDF, f to approximate the output value for a given input. One such class of algorithms are known as the artificial neural networks (ANN).^{36,37} These techniques have been used to tackle various complex problems in natural language processing³⁸ and computer vision.³⁹ Feed-forward neural networks are a class of ANN that have been proven to be general function approximators,⁴⁰ as they have the ability to transform the input into a new feature (latent) space, in which it becomes correlated with the output. The transformation is done sequentially through several layers and is typically highly nonlinear. Hence, feed-forward NNs are suitable for learning $f(Z_i, r_i) \rightarrow E$, which approximates PES.

Previously, NNs have been used to approximate PES for small molecules with the idea of many-body expansion.⁴¹ While being accurate, these methods scale poorly due to a large number of individual NN involved, typically, one for each term in many-body expansion. Various feature vectors have been developed recently, such as the smooth overlap of atomic positions, which was introduced by Bartók et al.¹² Nuclear charges (Z) and a matrix of interatomic distances were used as input to the model for prediction of the energy of the

molecule by Schütt et al.^{42,43} List of bonds, angles, nonbonded, and dihedrals (BAND) is used as a feature vector in BAND-NN by Siddhartha et al.⁴⁴ Atomic environment vector (AEV) was developed by Smith et al.,¹⁰ which is a modified version of Behler–Parrinello symmetry function¹¹ (BPSF).

$$E_{\text{tot}} = \sum_{i=1}^N E_i. \quad (1)$$

As it was later realized that energy is an extensive property and can be decomposed into atomic contributions. As shown in Equation (1), where E_i is the energy contribution of an atom i of the molecule with a total number of N atoms. Hence emerged high dimensional atomic NN (HDNN), which allowed one to use the same size network for different size molecules.^{10,11,42,43} HDNN can be broadly classified into two types. One is “descriptor-based,” and the other is the “message-passing” variant.

In the descriptor-based variant, the environment information about an atom is encoded in a handcrafted feature vector, for example, BAND, AEV, and BPSF. In feature vectors such as BAND, an atom's local environment is encoded using a list of BAND interactions. For feature vectors such as BPSF and AEV, the local environment of atom i is encoded as an array of values. These values are calculated using symmetry functions, which mathematically combine distances and/or angles between an atom of interest i and all other atoms in its neighborhood. This kind of atom-wise feature vector formulation makes these feature vectors invariant to translation, rotational, and permutation of equivalent atoms. The energy of the molecules is calculated as shown in Equation (1) where E_i is the energy contribution of the atom of interest. These invariances are desired property for feature vectors because NNs are numerical algorithms, which will produce different output values if input changes due to such transformation, which in principle does not change the molecules' conformation. Examples of models, which use handcrafted feature vector are BAND-NN, ANI, and TensorMol. ANI and TensorMol approaches are the extensions of the original approach proposed by Behler and Parrinello¹¹ or the deep potential molecular dynamics model.⁴⁵

The second type is the “message-passing” variant, which takes nuclear charges and Cartesian coordinates as input, and tries to learn a meaningful representation of the chemical environment by exchanging information between individual atoms using a deep neural network, also called as deep learning. This approach was first introduced by the deep tensor neural network (DTNN) and has since been refined in other DNN architectures, for example, SchNet,⁴³ hierarchically interacting particle neural network (HIP-NN),⁴⁶ and PhysNet.⁴⁷

It is clear that novel ML methods have the potential to provide efficient means for predicting the properties of molecules. However, this potential has been limited by the lack of standard comparative evaluations. Recently, Folmsbee et al. have compared the performance of neural network potentials (NNP) for representing PES and geometry optimization⁴⁸ whereas Gastegger et al. compared the accuracy of NNP on *trans* alkanes.⁴⁹ Algorithmic papers often benchmark proposed methods on disjoint dataset collections and highlight a

different kind of applicability, making it a challenge to gauge whether a proposed technique does, in fact, improve performance and applicability. In this study, we compare four NNP models, namely, ANI, PhysNet, BAND-NN, and SchNet to evaluate their performance using a common dataset across different test cases: (i) Comparison of the goodness of fit and transferability among these NNPs. It checks how well these models perform when trained on N atom molecules and are tested on molecules with more than N atoms; (ii) Their performance on structural and geometric isomers; (iii) Their ability to produce smooth, physically meaningful surface with respect to bond, angles, and torsional angle changes; (iv) The applicability of these models for geometry optimization and evaluates their performance using various optimization algorithms.

2 | METHODS

2.1 | Data selection

The ANI-1 dataset was chosen for the experiments, as they cover a large conformational space so that the models, when trained, are not confined to equilibrium structures only. This makes it possible to evaluate ML models for calculating energies of nonequilibrium structures, and subsequently, geometry optimization. The ANI-1 dataset was built from an exhaustive sampling of a subset of the GDB-11 database containing molecules upto eight heavy atoms and limiting the atomic species to C, N, O, and H, which gives 57,947 molecules. These 57,947 molecules were DFT optimized with ω B97x density functional and 6-31G(d) basis set which gives 57,462 DFT optimized starting structures. Smith and coworkers then generated tens of thousands of nonequilibrium structures for each 57,462 equilibrium structure, using normal-mode sampling. For this work, we chose a subset of the ANI-1 dataset that includes all the equilibrium structures and nonequilibrium structures whose relative energies with respect to the corresponding minimum energy structure is less than 30 kcal/mol.^{44,50} The justification for choosing this subset is that software such as Gaussview⁵¹ and Rdkit⁵² can produce initial geometries near the energy minima. In addition, in most of the cases, the aim of drug design/bio-molecular simulations is to model processes, which are near to energy minima. Hence, the chosen subset of the dataset is deemed sufficient.

2.2 | Training

All four models were trained from the scratch on the subset of the ANI-1 dataset described in the previous section. A train:test:validation data split in the ratio of 80:10:10 was used. This results in ~ 7 million data points in the training set and ~ 885 k data points each in the test and validation sets. All the hyper-parameters were taken as default as stated in original papers of ANI-1,¹⁰ PhysNet,⁴⁷ BAND-NN,⁴⁴ and SchNet.⁴³ The training code for each of the models is obtained from the code repositories provided by respective authors.

$$L = |E - E_{\text{ref}}| + L_{nh}, \quad (2)$$

$$L_{nh} = \frac{\lambda_{nh}}{N} \sum_{i=1}^N \sum_{m=2}^{N_{\text{modules}}} \frac{(E_i^m)^2}{(E_i^m)^2 + (E_i^{m-1})^2}. \quad (3)$$

All the models were trained on atomization energy with criteria being minimization of the mean squared error loss function. In PhysNet, a different loss function is used as shown in Equations (2) and (3) where E is predicted energy, E_{ref} is DFT reference energy, L_{nh} is “nonhierarchicality penalty,” λ_{nh} is regularization hyperparameter set to 10^{-2} , E_i^m is energy contribution of atom i from module m , the total number of modules used in the PhysNet model is N_{modules} and N is total number of atoms in a molecule.⁴⁷

For ANI, SchNet, and BAND model's ADAM,⁵³ optimizer is used to minimize the loss function, whereas, for PhysNet, we used ASMGrad.⁵⁴ The batch size for ANI is 1024, and 32 for BAND-NN, SchNet, and PhysNet. The learning rate of 0.01 was used for BAND, 0.0001 for SchNet, and 0.001 for ANI and PhysNet. Learning rate decay of 0.5, 0.1, 0.1, and exponential decay of 0.96 was used in ANI, BAND, PhysNet, and SchNet, respectively.

These four models incorporate a different type of feature vectors. ANI uses AEVs, which are modified Behler and Parrinello symmetry functions.¹¹ SchNet and PhysNet take nuclear charges and positions as input, and BAND-NN uses feature vectors inspired from classical FF equation. While ANI and BAND-NN use simple multilayered neural network, SchNet and PhysNet have modular architectures to allow for the calculation of interactions between atoms and learn atomic features in a hierarchical manner. These four descriptors demonstrate the evolution of models from simple hand-engineered descriptor and multilayer NN to complex message-passing models with embedding and cfconv ⁵⁵ layers. These varied sophistications within these models piqued our interest to see how well they perform when compared to each other.

2.2.1 | ANI

The feature vectors used in ANI are called AEVs. They represent the atomic environment around each atom. AEVs are constructed from “symmetry functions” which probe the radial and angular environment for each atom. These are modified versions of the original BPSF.¹¹ These AEV's are then fed into a multilayered feed-forward neural network, and the weights of the neural network are optimized during the training process. The energy is calculated as the sum of contributions E_i from each atom of a molecule, as shown in Equation (1). The architecture of ANI-model is described in Figure 1.

2.2.2 | BAND-NN

BAND-NN is inspired from the FF equation, where the energies are calculated as sum of the contributions from BAND as described in

Equation (4). The feature vectors are thus created from the list of BAND of a molecule denoting the atoms that make up these terms, distances, angles, and dihedral angles. BAND-NN uses four different neural networks for each of BAND. The BAND-NN architecture is described in Figure 2.

$$E_{\text{total}} = \sum E_{\text{bonds}} + \sum E_{\text{angles}} + \sum E_{\text{non-bonds}} + \sum E_{\text{dihedrals}} \quad (4)$$

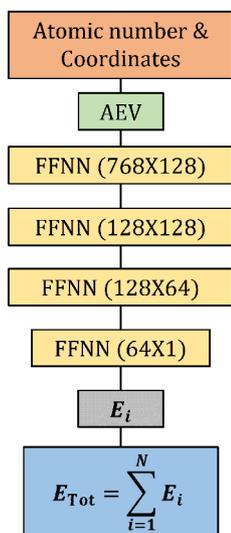


FIGURE 1 Overview of ANI architecture showing coordinates (orange box), atomic environment vector (AEV in green box). Feed forward neural network (FFNN) (yellow boxes) with hidden layers specified in round brackets, E_i (gray box) is the energy contribution of atom i and E_{Tot} (blue box) gives the total energy of the molecule calculated as shown in Equation (1)¹⁰

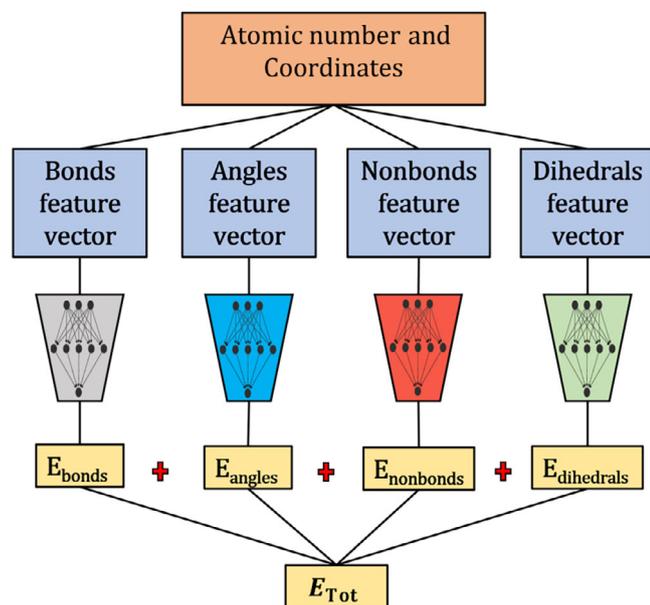


FIGURE 2 Overview of the BAND-NN architecture showing four different neural networks (trapezoids) for bonds (gray), angles (blue), non-bonds (red), and dihedral (green) inspired from force fields⁴⁴

2.2.3 | SchNet

SchNet is a variant of DTNNs.⁴² This allows SchNet to learn representations for molecules that are invariant to rotation, translation, and atom indexing. SchNet uses continuous-filter convolutions with filter-generating networks^{55,56} to model the atomic interactions inside the interaction block (blue boxes). As shown in Figure 3, SchNet has a modular architecture, atom embedding (green box), interaction refinements (blue boxes), and atom-wise energy contributions E_i . At each layer, the atomic representation is refined to better model the atomic interaction with the surrounding environment. SchNet takes nuclear charges (Z) and positions (R) as input during training.

2.2.4 | PhysNet

PhysNet⁴⁷ is inspired from SchNet⁴³ and HIP-NN.⁴⁶ The architecture is modular and is shown in the Figure 4. To model the atomic interactions, PhysNet uses learnable distance based attention masks that select different features based on the pairwise distance r_{ij} between atoms inside the interaction block. In addition, to circumvent the vanishing gradients problem arising due to vanishing of gradients in deeper neural nets, PhysNet uses pre-activation residual blocks, which skips one or several layers while training. In the first module, the input is an atom-wise embedding vector. For other lower modules, the feature vector is obtained as output from the respective previous

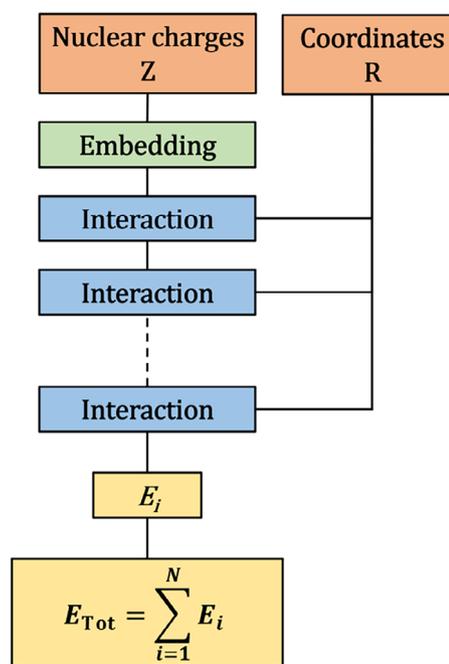


FIGURE 3 Overview of SchNet architecture with nuclear charge Z and coordinates R as input to embedding (green box) and interaction block (blue box) respectively. There are total of six interaction blocks, represented as dotted line. E_i is the atomic energy contribution of atom i and E_{Tot} is the total energy of a molecule with N atoms⁴³

module. Feature vector gets updated as it moves down the five modules. As the feature vector passes through each module it captures higher-order interactions. Energy from each module is added to obtain the atomic energy E_i . These energies are further summed to calculate the total energy E_{Tot} of a molecule with “N” atoms.

3 | RESULTS AND DISCUSSION

In this section, we present the comparative results of all four models, namely, ANI, PhysNet, BAND-NN, and SchNet across four test cases. The first test case evaluates the goodness of fit, transferability, and overall performance of the models. For this, we tested the models using 885 k molecules test-set. We also used one more test-set called random GDB, which has 1617 molecules with ten heavy atoms. The atom counts for random GDB test systems are ten heavy atoms, and total atom count ranges from 15 to 32 atoms.¹⁰ In the second test case, we show the accuracy of the models in predicting the relative energies of DFT energy minimized $C_{10}H_{20}$ isomers with respect to the lowest energy isomer. The third test case is to evaluate the ability of the models to produce a smooth PES. Hence, PES scans for bond stretch, angle bend, and two dihedral rotations on relatively large molecules are carried out using NNPs and are compared with reference DFT results. In the fourth test case, we perform geometry optimization on the following four molecules, namely, methamphetamine (eight conformers), decane (eight conformers), fentanyl (eight conformers), and retinol (eight conformers) using various optimization methods, 32 conformer in total and use NNPs to predict energy during the optimization.

3.1 | Accuracy and transferability

As mentioned in the *Data selection* section, all the conformers that were under 30 kcal/mol in the ANI-1 data set from the corresponding

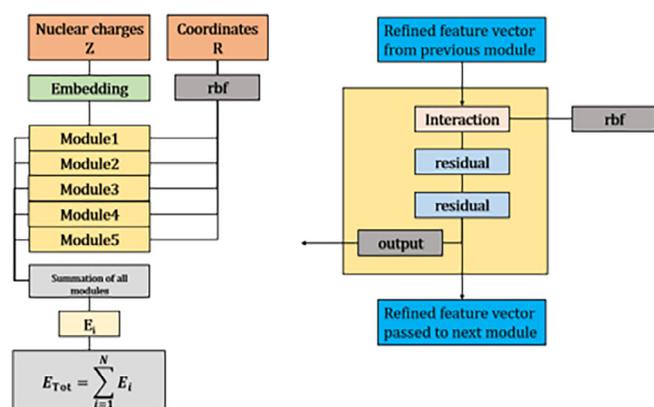


FIGURE 4 Overview of the PhysNet architecture (left). Embedding block, five module blocks (yellow boxes). The schema of module block (right). Atom-wise energy contribution E_i and their sum gives total energy E_{Tot} of a molecule with “N” atoms⁴⁷

minimum energy structure were chosen for this study. We did 80:10:10 split; this resulted in 7 million data points in the training set and 885 k data points each in the test and validation sets. We have summarized the performance of all four models on test-set in Table 1. A summary of the performances of all four models on a random GDB test set, which has 1617 molecules with 10 heavy atoms, is given in Table 2. This shows the overall accuracy and transferability of these models as they are tested on larger molecules as compared to molecules in the training set. It is also important to know the performance of these models from their respective papers. ANI reported a mean absolute error (MAE) of 0.83 kcal/mol, and root means squared error (RMSE) of 1.12 kcal/mol on GDB-10 test set with structures that are 30 kcal/mol away from the energy minima and RMSE of 1.91 kcal/mol on all of GDB-10 test set.¹⁰ Unke and Meuwly, in 2019, proposed PhysNet, which achieved an MAE of 0.19 kcal/mol on QM9 dataset.⁴⁷ SchNet model by Schütt et al. achieved MAE of 0.31 kcal/mol on QM9 dataset.⁴³ Both PhysNet and SchNet were trained on the QM9 dataset, which has only ~134 k small organic molecules, which is small as compared to the 7 million training set which has been used in this work. BAND-NN reported an MAE of 1.45 kcal/mol on the test-set and MAE of 2.1 kcal/mol on the GDB-10 dataset.

ANI, which uses a hand-crafted feature vector, achieves MAE of 0.39 kcal/mol and RMSE of 0.55 kcal/mol on test set. In contrast, PhysNet, which learns the feature vector directly from the data, achieves MAE of 0.40 kcal/mol and RMSE of 0.62 kcal/mol on the test set with the same size molecules as in the training set, that is, molecules with eight heavy atoms. So, when it concerns only accuracy, ANI and PhysNet have similar accuracy. But, when it concerns accuracy and transferability to bigger molecules, ANI achieves the best performance out of all four NNP's with MAE of 0.83 kcal/mol and RMSE of 1.17 kcal/mol on a random GDB test set which has larger molecules with 10 heavy atoms in contrast to the training set which has molecules up to eight heavy atoms.

3.2 | Structural and geometric isomers

In this section, the models ability to accurately predict energies of geometric isomers with the empirical formula $C_{10}H_{20}$ is evaluated. Thirteen geometric isomers spanning diverse structural and geometric space are chosen, including linear chains, *cis-trans*, and ring containing isomers, namely, *p*-menthane, *n*-butylcyclohexane, *t*-butylcyclohexane,

TABLE 1 Summary of performance of the models on test set (kcal/mol)

| Model | MAE | RMSE |
|---------|------|------|
| ANI | 0.39 | 0.55 |
| PhysNet | 0.40 | 0.62 |
| BAND-NN | 1.45 | 1.82 |
| SchNet | 0.48 | 0.60 |

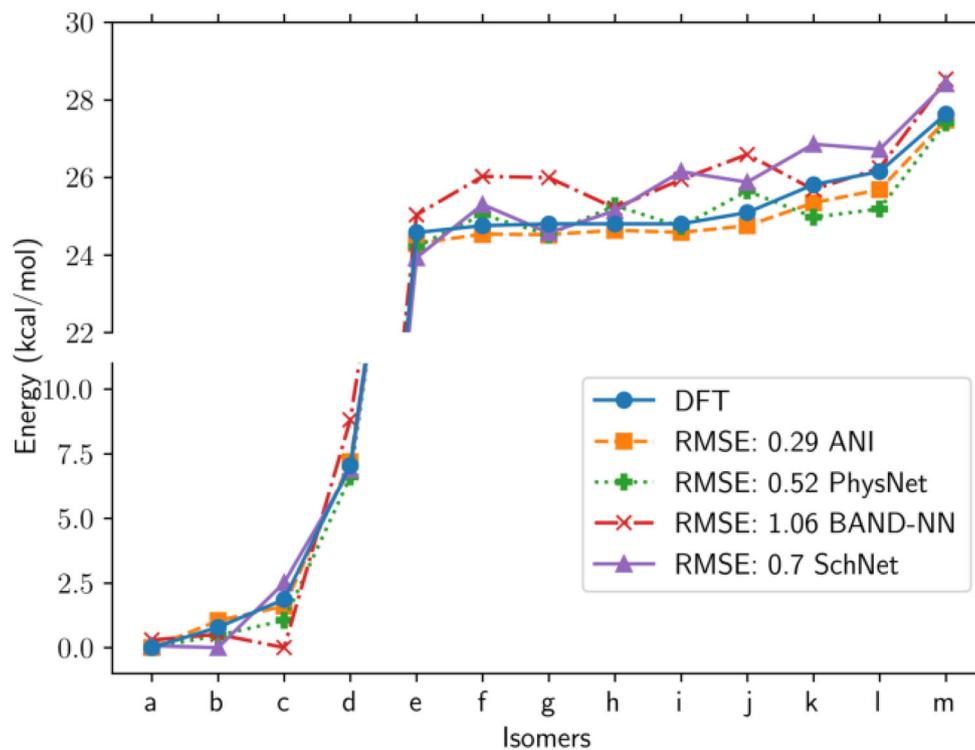
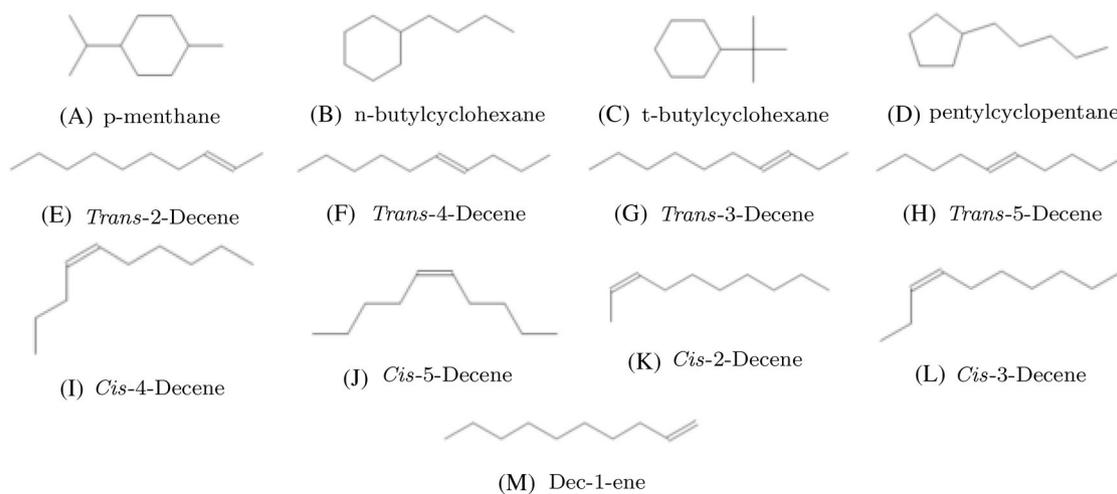
Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

TABLE 2 Summary of performance of the models on test set from GDB-10 dataset (kcal/mol)

| Model | MAE | RMSE |
|---------|------|------|
| ANI | 0.83 | 1.17 |
| PhysNet | 1.26 | 1.65 |
| BAND-NN | 2.1 | 2.68 |
| SchNet | 1.51 | 1.89 |

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

pentacyclopentane, *trans*-2-decene, *trans*-4-decene, *trans*-3-decene, *trans*-5-decene, *cis*-4-decene, *cis*-5-decene, *cis*-2-decene, *cis*-3-decene, and dec-1-ene. The structures of all 13 isomers are shown in Figure 5. All isomers were optimized using the ω B97X/6-31G(d) level of theory using Gaussian 09 software.⁵⁷ The accuracy of these models in predicting the energies of the DFT optimized $C_{10}H_{20}$ isomers with respect to the lowest energy isomer are compared. Figure 5 shows that all the NNPs accurately predict the minimum energy structure and continue to accurately order the energies across

(N) Relative energies of $C_{10}H_{20}$ isomers**FIGURE 5** (A–M) Are all the structural and geometric isomers used to generate the data for the isomer case study and (n) plot of relative energies of structures given in (A–M)

ring containing structures, linear alkenes structures, and linear alkanes structures. ANI achieves the best performance with an RMSE of 0.29 kcal/mol followed by PhysNet, with an RMSE of 0.52 kcal/mol. This experiment is indicative of the ability of these models to capture higher-order atomic interactions and differentiate between the isomers.

3.3 | Potential energy surface

In this section, the models' ability to generate smooth and accurate PES is evaluated. From the experiments reported above, it is clear that these models can accurately predict the atomization energy of small organic molecules. However, it is important that these models not only accurately predict atomization energy but also be able to produce meaningful PES for a given molecule. Such behavior is necessary for these models to be applicable in energy minimization and force calculations in molecular dynamics simulations.

To examine how well these models produce PES, we performed PES scans of various molecules that are significantly larger than those in the training set. We generated PES profiles for N1—C2 bond stretch and C2—C3—C4 angle bend in methamphetamine, shown in Figure 6, N1—C2 bond stretch and C3—C4—C5 angle bend in fentanyl, shown in Figure 7, C1—C2—C3—C4 dihedral scan in decane and 4-cyclohexyl butanol, shown in Figures 8 and 9 respectively. C1—C2 bond stretch, C2—C3—C4 angle bend, and C1—C2—C3—C4 torsion angle in pentadecane shown in Figure 10.

As evident from Figures 6A, 7A, and 10A for 3 C—N, C—N, and C—C bond stretch scans respectively, ANI and PhysNet produce smoother curves and have predicted accurate bond equilibrium distances when compared to others. SchNet misses the bond equilibrium distance by 0.04 Å in Figure 7A. For PES scan along angle bend, as seen in Figures 6B, 7B, and 10B, all the models have good agreement with the DFT curve, and PhysNet has the best performance in all the three cases. For torsion angle, as shown in Figures 8A, 9A, and 10C, ANI does better than the other models and is able to get an accurate estimate of the dihedral rotation barrier.

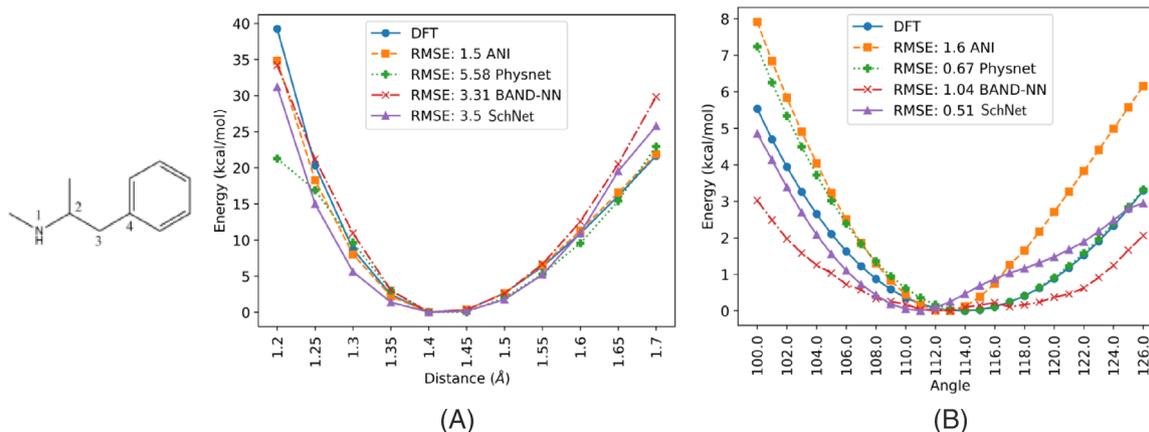


FIGURE 6 Methamphetamine structure and specific atoms used for potential energy surface (PES) are shown (left). PES scan of (A) N1—C2 bond stretch and (B) C2—C3—C4 angle bend are shown

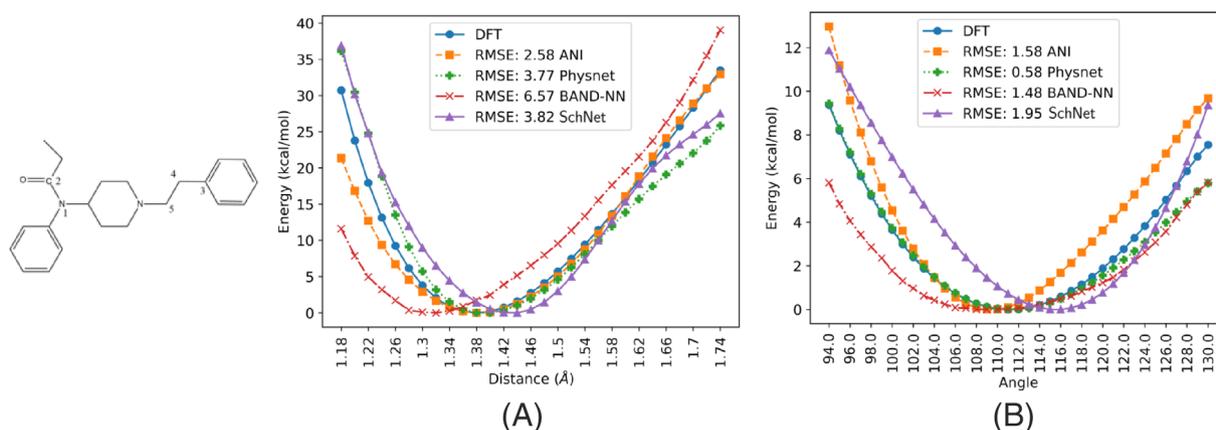


FIGURE 7 Fentanyl structure and specific atoms used for potential energy surface (PES) are shown (left). PES scan of (a) N1—C2 bond stretch and (b) C3—C4—C5 angle bend are shown

3.4 | Geometry optimization

Along with property prediction and accurate representation of PES, it is also desirable for these models to be used with optimization algorithms to perform geometry optimization of non-optimized structures, circumventing the need for expensive DFT-based optimization. The combination of NNPs with optimization algorithms will not only allow us to accelerate the optimization process, but also give us access to optimize larger molecules which cannot be optimized using DFT. Four molecules, which are larger than the molecules in the training set, were considered for optimization, namely, decane ($C_{10}H_{22}$), fentanyl ($C_{22}H_{28}N_2O$), retinol ($C_{20}H_{30}O$), and methamphetamine ($C_{10}H_{15}N$). Eight conformers for each of the four molecules were created using the RDKit⁵² package. These were then optimized to obtain their respective DFT energy minima. Eight different conformers were generated for each molecule so as to have a different starting point for each optimization and hence probe more of the conformational space.

We used 14 different optimization algorithms spanning across different paradigms of gradient-free optimization techniques. To name

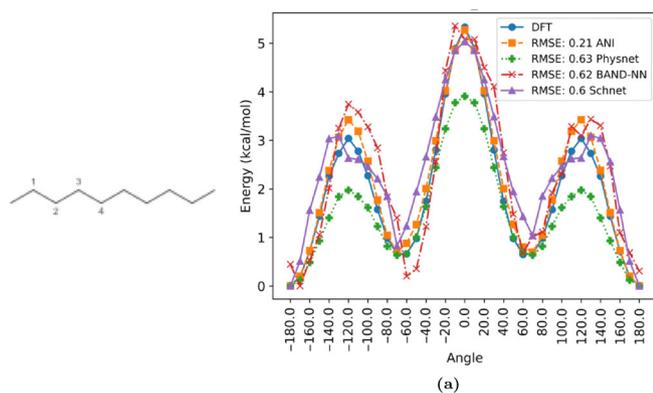


FIGURE 8 Decane structure and specific atoms used for potential energy surface (PES) are shown (left). PES scan of (a) C1–C2–C3–C4 dihedral angle is shown

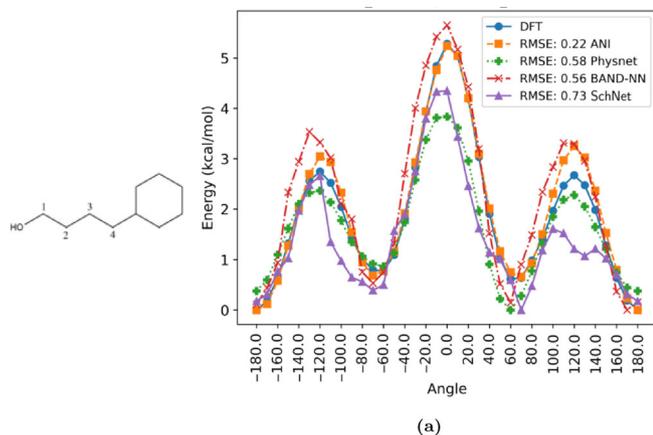


FIGURE 9 4-Cyclohexylbutanol structure and specific atoms used for potential energy surface (PES) are shown (left). PES scan of (a) C1–C2–C3–C4 dihedral angle is shown

a few, we have used the BFGS optimizer, a quasi-Newton method that searches for the stationary point on the optimization landscape using first-order gradients or gradients computed approximately. The Nelder–Mead algorithm is based on the iterative computation of nonlinear simplices. The OnePlusOne is a subclass of evolutionary algorithms. The covariance matrix adaptation belongs to a class of evolutionary strategies using Gaussian sampling. Powell is a line search method built using iterative computation of search vectors. Cobyla works through iterative linear approximations of the objective function. For this work, the BFGS optimizer implementation in the scipy python package was used. Other non-gradient-based methods are implemented in the Nevergrad⁵⁸ python package. The results of optimization have been summarized in Tables S1–S55. Each table in Data S1 represents data of one optimization method for a single molecule out of four molecules.

We have 14 optimization methods and four molecules, which make up a total of 56 tables. One such set of optimization results are summarized in Table 3, where column number “(1)–(8)” represent eight conformers of a single molecule, and models are present in rows. The values in each cell represents energy ΔE_{opt} in kcal/mol calculated as shown in Equation (5), where $E_{\text{opt}}^{\text{Model}}$ is the model optimized energy and $E_{\text{opt}}^{\text{DFT}}$ is DFT optimized energy (ground truth). ΔE_{opt} indicates how far away is the model optimized conformer from the DFT optimized conformer. Hence smaller values of ΔE_{opt} are better as they indicate the ability of the models to predict optimized structures in very close agreement to that of the DFT optimized structures (ground truth). The last column, called “Best count”, represents the number of counts out of eight conformers for which the model got the best ΔE_{opt} value (smaller is better).

Table 3 shows optimization results for decane molecules using the CManAS3 method. As ANI gets the best ΔE_{opt} values (smaller is better) for conformers 2, 4, and 6 as compared to other models, we put “Best count” as 3 for ANI. Similarly, PhysNet gets the best values for five conformers out of eight; hence, we give five to PhysNet in the “Best count” column. Similarly, for all the 55 tables in Data S1

$$\Delta E_{\text{opt}} = E_{\text{opt}}^{\text{Model}} - E_{\text{opt}}^{\text{DFT}} \quad (5)$$

The experiments are performed on four molecules, namely, methamphetamine (eight conformers), decane (eight conformers), fentanyl (eight conformers), and retinol (eight conformers) and 14 different optimization methods. Altogether, there are $4 \times 8 \times 14 = 448$ different optimization procedures. For decane, ANI achieves 55 best-optimized conformers out of 112 as compared to all other models, whereas PhysNet achieves 48 best-optimized conformers out of 112 (Table 4). For decane, the best-optimized structure is only 0.61 kcal/mol away from DFT energy minima (Table S9). Similarly, the best-optimized conformer for fentanyl, retinol, and methamphetamine are 5.37, 3.51, and 1.18 kcal/mol away from DFT energy minima. All the Optimization results are given in Data S1. For the three molecules, namely, decane, fentanyl, and retinol, the ANI model has the highest number of best-optimized conformers count, whereas, for methamphetamine, PhysNet has a higher number of best-optimized

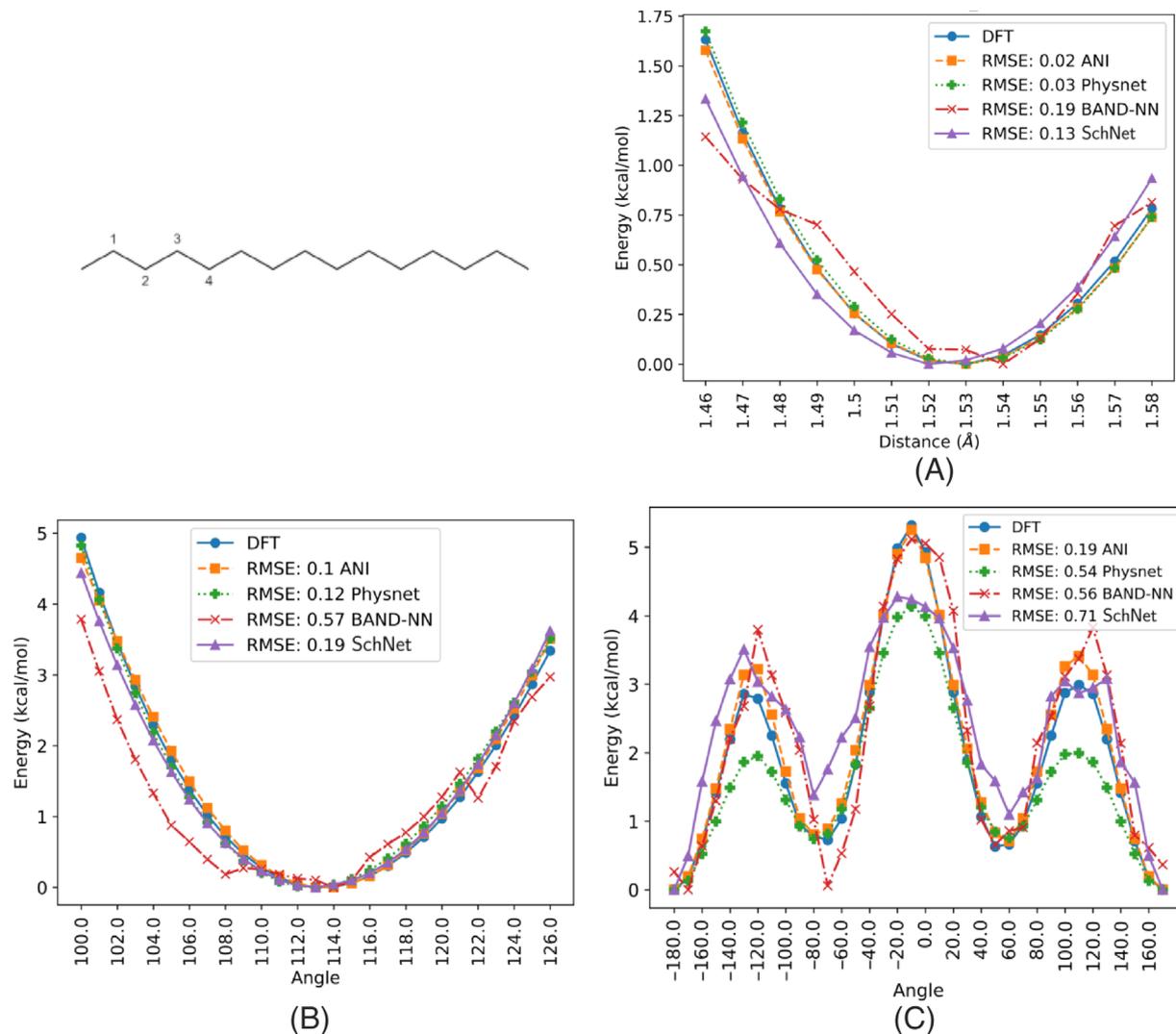


FIGURE 10 Pentadecane structure and specific atoms used for potential energy surface (PES) are shown (top left). PES scan of (a) C1–C2 bond stretch, (b) C2–C3–C4 angle bend, and (c) C1–C2–C3–C4 torsion angle are shown

| Structures | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Best count |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|
| ANI | 0.43 | 0.56 | 0.62 | 0.64 | 0.44 | 0.56 | 0.96 | 0.58 | 3 |
| PhysNet | 0.42 | 0.57 | 0.59 | 0.66 | 0.44 | 0.57 | 0.95 | 0.57 | 5 |
| SchNet | 0.5 | 0.63 | 0.67 | 0.64 | 0.48 | 0.64 | 1.12 | 0.62 | 0 |
| BAND-NN | 0.57 | 0.68 | 0.79 | 0.83 | 0.57 | 0.65 | 1.14 | 0.73 | 0 |

Note: Values shown are the differences between optimized DFT and ML energies (kcal/mol). The best value (smallest) in each column is highlighted in bold indicating which model performs best for each of the eight structures.

Abbreviations: DFT, density functional theory; ML, machine learning.

TABLE 4 Count of the best-optimized conformers

| Molecules | Decane | Fentanyl | Retinol | Methamphetamine |
|-----------|--------|----------|---------|-----------------|
| ANI | 55 | 66 | 85 | 39 |
| PhysNet | 48 | 15 | 18 | 56 |
| SchNet | 4 | 19 | 9 | 5 |
| BAND-NN | 5 | 12 | 0 | 12 |
| Total | 112 | 112 | 112 | 112 |

TABLE 3 Optimization of eight decane structures using CMandAS3 optimization method

conformers count of 56 out of 112. ANI performs better than the other three models as it gets the best optimization for 245 times out of 448. PhysNet gets 137 best-optimized conformers out of 448. Although results for all the conformers are not perfect, this could be further improved using appropriate molecular representations. We have also used gradient-based optimization BFGS, which gives the best-optimized conformer for decane, fentanyl, retinol, and methamphetamine, which are 0.49 (ANI), 4.30(ANI), 2.93(SchNet), and 0.95

(PhysNet) kcal/mol away from DFT energy minima respectively. Overall, ANI performs better than other three models.

4 | CONCLUSIONS

In this work, we evaluate and compare four models, that is, ANI, PhysNet, SchNet, and BAND-NN, on their accuracy in energy prediction, transferability to larger molecules, ability to produce accurate PES, and applicability in geometry optimization. These models were originally trained and tested on different data sets, which makes their applicability and comparison difficult. This work provides a standard comparative evaluation of these models by training and testing them on the same sets. For accuracy and transferability, we report that the ANI model performs best with RMSE of 0.55 and 1.17 kcal/mol on ~885 k molecules and on the GDB-10 test set, which has 1617 molecules with 10 heavy atoms randomly selected from the GDB-11 database. We also notice that both ANI (descriptor-based) and PhysNet (message-passing) produce smooth and meaningful surfaces and are potentially applicable in Molecular dynamics simulations. All these models were also able to accurately differentiate different isomers of the same empirical formula $C_{10}H_{20}$. ANI and PhysNet achieve an RMSE of 0.29 kcal/mol and 0.52 kcal/mol, respectively. All these models are trained on DFT energies, but can also be extended to higher-level ab initio QM methods and larger basis sets. These models also show their potential for geometry optimization. ANI outperformed all models by optimizing 245 structures more accurately than others in 448 test structures. PhysNet stood next with 137 best-optimized conformers.

ANI is a variant of descriptor-based HDNNs where features are manually constructed. On the contrary, PhysNet is an end-to-end data-driven model. It is clear that ANI outperforms all the models in most of the experiments. The PhysNet model, which takes only distances and atom types as input without requiring any additional hand engineering, is not far behind ANI. ANI and PhysNet show promising results in producing smooth and accurate PES and can perform geometry optimization. While deep neural network have demonstrated great potential in achieving accurate energies, systematic benchmarking of these are necessary for wider applicability and transferability.

ACKNOWLEDGMENT

Rohit Modee thanks TCS for a research fellowship. We thank IHub-Data for financial assistance.

ORCID

Rohit Modee  <https://orcid.org/0000-0002-2012-8031>

U. Deva Priyakumar  <https://orcid.org/0000-0001-7114-3955>

REFERENCES

- [1] M. Rupp, O. A. Von Lilienfeld, K. Burke, *J. Chem. Phys.* **2018**, *148*, 241401.
- [2] A. Mullard, *Nature* **2017**, *549*, 445.
- [3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *J. Japan Soc. Fuzzy Theory Intell. Inf.* **2017**, *29*, 177.
- [4] L. P. Kaelbling, M. L. Littman, A. W. Moore, *J. Artif. Intell. Res.* **1996**, *4*, 237.
- [5] M. D. Waltz Ph.D. thesis, Purdue University **1964**.
- [6] M. H. S. Segler, T. Kogej, C. Tyrchan, M. P. Waller, *ACS Cent. Sci.* **2018**, *4*, 120.
- [7] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, A. Aspuru-Guzik Objective-reinforced generative adversarial networks (ORGAN) for sequence generation Models. *arXiv* **2017**,
- [8] L. Yu, W. Zhang, J. Wang, Y. Yu. 31st AAAI Conf. Artif. Intell. **2017**, 2852–2858.
- [9] B. J. Braams, J. M. Bowman, *Int. Rev. Phys. Chem.* **2009**, *28*, 577.
- [10] J. S. Smith, O. Isayev, A. E. Roitberg, *Chem. Sci.* **2017**, *8*, 3192.
- [11] J. Behler, M. Parrinello, *Phys. Rev. Lett.* **2007**, *98*, 146401.
- [12] A. P. Bart'ok, R. Kondor, G. Cs'anyi, *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *87*, 184115.
- [13] E. V. Podryabinkin, A. V. Shapeev, *Comput. Mater. Sci.* **2017**, *140*, 171.
- [14] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, A. R. Oganov, *Phys. Rev. B* **2019**, *99*, 064114.
- [15] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K. R. Müller, *Sci. Adv.* **2017**, *3*, e1603015.
- [16] S. Chmiela, H. E. Sauceda, K.-R. Müller, A. Tkatchenko, *Nat. Commun.* **2018**, *9*, 3887.
- [17] K. Ryczko, D. A. Strubbe, I. Tamblin, *Phys. Rev. A* **2019**, *100*, 022512.
- [18] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, K. R. Müller, *Nat. Commun.* **2017**, *8*, 872.
- [19] J. Lyons, A. Dehzangi, R. Heffernan, A. Sharma, K. Paliwal, A. Sattar, Y. Zhou, Y. Yang, *J. Comput. Chem.* **2014**, *35*, 2040.
- [20] Q. Jiang, X. Jin, S. J. Lee, S. Yao, *J. Mol. Graphics Modell.* **2017**, *76*, 379.
- [21] J. Hanson, K. Paliwal, T. Litfin, Y. Yang, Y. Zhou, *Bioinformatics* **2019**, *35*, 2403.
- [22] S. Romero-Molina, Y. B. Ruiz-Blanco, M. Harms, J. Münch, E. Sanchez-Garcia, *J. Comput. Chem.* **2019**, *40*, 1233.
- [23] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, A. Walsh, *Nature* **2018**, *559*, 547.
- [24] P. Raccuglia, K. C. Elbert, P. D. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, A. J. Norquist, *Nature* **2016**, *533*, 73.
- [25] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, *npj Comput. Mater.* **2016**, *2*, 16028.
- [26] M. H. Segler, M. Preuss, M. P. Waller, *Nature* **2018**, *555*, 604.
- [27] P. O. Dral, *J. Comput. Chem.* **2019**, *40*, 2339.
- [28] Y. Pathak, S. Laghuvarapu, S. Mehta, U. D. Priyakumar, *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 873.
- [29] J. A. Pople, *Angew. Chemie Int. Ed.* **1999**, *38*, 1894.
- [30] J. Danielsson, M. Meuwly, *J. Chem. Theory Comput.* **2008**, *4*, 1083.
- [31] T. Nagy, J. Yosa Reyes, M. Meuwly, *Chem. Theory Comput.* **2014**, *10*, 1366.
- [32] B. Hartke, S. Grimme, *Phys. Chem. Chem. Phys.* **2015**, *17*, 16715.
- [33] J. Yosa Reyes, S. Brickel, O. T. Unke, T. Nagy, M. Meuwly, *Phys. Chem. Chem. Phys.* **2016**, *18*, 6780.
- [34] S. Brickel, M. Meuwly, *J. Phys. Chem. A* **2017**, *121*, 5079.
- [35] O. T. Unke, M. Meuwly, *J. Chem. Phys.* **2018**, *148*, 241708.
- [36] M. Haghightlari, J. Hachmann, *Curr. Opin. Chem. Eng.* **2019**, *23*, 51.
- [37] C. M. Bishop, *J. Electron. Imaging* **2007**, *16*, 049901.
- [38] R. Sarikaya, G. E. Hinton, A. Deoras, *IEEE Trans. Audio, Speech Lang. Process.* **2014**, *22*, 778.
- [39] A. Krizhevsky, I. Sutskever, G. E. Hinton, *Commun. ACM* **2017**, *60*, 84.
- [40] K. Hornik, M. Stinchcombe, H. White, *Neural Networks* **1989**, *2*, 359.
- [41] F. A. Faber, A. S. Christensen, B. Huang, O. A. Von Lilienfeld, *Chem. Phys.* **2018**, *148*, 241717.

- [42] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, A. Tkatchenko, *Nat. Commun.* **2017**, *8*, 6.
- [43] K. T. Schütt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko, K. R. Müller, *J. Chem. Phys.* **2018**, *148*, 241722.
- [44] S. Laghuvarapu, Y. Pathak, U. D. Priyakumar, *J. Comput. Chem.* **2020**, *41*, 790.
- [45] L. Zhang, J. Han, H. Wang, R. Car, E. Weinan, *Phys. Rev. Lett.* **2018**, *120*, 143001.
- [46] N. Lubbers, J. S. Smith, K. Barros, *J. Chem. Phys.* **2018**, *148*, 241715.
- [47] O. T. Unke, M. Meuwly, *J. Chem. Theory Comput.* **2019**, *15*, 3678.
- [48] G. R. Hutchison, D. L. Folmsbee, D. R. Koes, *J. Phys. Chem. A* **2021**, *125*, 1987.
- [49] M. Gastegger, C. Kauffmann, J. Behler, P. Marquetand, *J. Chem. Phys.* **2016**, *144*, 194110.
- [50] J. S. Smith, O. Isayev, A. E. Roitberg, *Sci. Data* **2017**, *4*, 170193.
- [51] R. Dennington, T. Keith, J. Millam. Others, GaussView, version 5. **2009**.
- [52] Landrum, G. RDKit: Open-source cheminformatics.
- [53] D. P. Kingma, J. L. Ba. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. **2015**, 1–15.
- [54] S. J. Reddi, S. Kale, S. Kumar 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc. **2018**, 1–23.
- [55] K. T. Schütt, P. J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, K. R. Müller, *Adv. Neural Inf. Process. Syst.* **2017**, *2017*, 992.
- [56] B. De Brabandere, X. Jia, T. Tuytelaars, L. Van Gool, *Adv. Neural Inf. Process. Syst.* **2016**, *30th conference*, 667.
- [57] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Starovrov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox, Gaussian~09. 2009.
- [58] J. Rapin, O. Teytaud Nevergrad - A gradient-free optimization platform. Github Repos. **2018**,

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: R. Modee, S. Laghuvarapu, U. D. Priyakumar, *J. Comput. Chem.* **2022**, *43*(5), 308. <https://doi.org/10.1002/jcc.26790>