3D Dense Convolutional Neural Networks Utilizing Molecular Topological Features for Accurate Atomization Energy Predictions

Ankur Kumar Gupta and Krishnan Raghavachari**

Department of Chemistry, Indiana University, Bloomington, Indiana 47405, United States

ABSTRACT

Deep learning methods provide a novel way to establish a correlation between two quantities. In this context, computer vision techniques like 3D-Convolutional Neural Networks (3D-CNN) become a natural choice to associate a molecular property with its structure due to the inherent three-dimensional nature of a molecule. However, traditional 3D input data structures are intrinsically sparse in nature, which tend to induce instabilities during the learning process, which in turn may lead to under-fitted results. To address this deficiency, in this project, we propose to use quantum-chemically derived molecular topological features, namely, Localized Orbital Locator (LOL) and Electron Localization Function (ELF), as molecular descriptors, which provide a relatively denser input representation in three-dimensional space. Such topological features provide a detailed picture of the atomic configuration and inter-atomic interactions in the molecule and are thus ideal for predicting properties that are highly dependent on molecular geometry. Herein, we demonstrate the efficacy of our proposed model by applying it to the task of predicting atomization energies for the QM9-G4MP2 dataset, which contains ~134-k molecules. Furthermore, we incorporated the Δ-ML approach into our model, allowing us to reach beyond benchmark accuracy levels (~1.0 kJ mol^{-1}). We consistently obtain impressive MAEs of the order 0.1 kcal mol^{-1} (~ 0.42 kJ mol^{-1}) *versus* G4(MP2) theory using relatively modest models, which could potentially be improved further using additional compute resources.

1. Introduction

A recent surge in deep learning and computer vision research has pushed this field to unprecedented heights, so much so that new state-of-the-art models are being developed and implemented every other month for 2D image recognition tasks.[1] These newly developed computer vision techniques have profoundly impacted other branches of science as well, and chemistry is no exception. Thus, taking a cue from 2D image representations, molecules, being intrinsically three-dimensional in nature, can be imagined as 3D images and, therefore, can be analogously represented in the form of a 3D grid or a multi-dimensional tensor. However, unlike 2D images, where the input features are quite well-defined, viz., red, green, and blue (RGB) color channels, there is no clear consensus on the choice of descriptors to represent a molecule, and this remains an outstanding task in the field of machine learning in chemistry. Nonetheless, a variety of molecular descriptors have been identified for representing a molecule in a 3D data structure (*vide infra*) and successfully used for a diverse set of problems ranging from protein-ligand binding affinity prediction[2-9] and receptor binding site detection and classification[10-13] to the prediction of material properties[14, 15] and NMR chemical shifts.[16] A major complication associated with 3D input representations is its high data sparsity aggravated due to its 3D grid cell structure; therefore, in this article, we advocate the use of spatially dense descriptors, especially the ones based on the electron distribution in the molecule, thus providing an alternative to mitigate the data structure sparsity. Specifically, we propose to use what are known as electron localization functions, viz., Localized Orbital Locator (LOL)[17] and Electron Localization Function (ELF),[18] which have found widespread use in elucidating molecular bonding topology. The input data structure usually dictates the architecture type of the network. Therefore, with the input data structure defined, a convolutional neural network (CNN) becomes an obvious choice for the model architecture. Among the host of CNN architectures available in the literature for image learning tasks, we chose to use the DenseNet architecture chiefly for its high parameter efficiency.

Computing molecular bond energies to high accuracy is one of the holy grails of quantum chemistry. However, the steep computational requirements of highly accurate methods such as CCSD(T)[19] and Gaussian-4,[20] preclude their use on a routine basis. A variety of noteworthy graph-based architectures (viz., SchNet[21], PhysNet[22], DimeNet[23], DeepMoleNet[24], OrbNet[25]) have been proposed for the prediction of DFT level (B3LYP/6-31G(*2df,p*)) energies on the

QM9 dataset.[26, 27] In this work, however, we aim to predict G4(MP2) level energies, a relatively cheaper alternative to the G4 method, which is typically accurate within 1.0 kcal mol^{-1} of the experimental value, and hence is a more valuable quantity to reproduce. Therefore, in the present work, we attempt to leverage and adapt some of the latest developments in fields such as computer vision for the task of predicting atomization energies at high levels of accuracy. In this context, we note that Ward *et al.*[28] have achieved a highly impressive out-of-sample mean absolute error (MAE) of the order of 0.1 kcal mol^{-1} (*versus* the G4(MP2)[29] level of theory) on the QM9-G4MP2 dataset[27, 30] using the SchNet and FCHL[31] models in conjunction with the Δ-Machine Learning approach.[32] As the name suggests, the Δ-ML strategy targets learning the energy difference between an expensive target level of theory and a cheaper baseline level of theory, thus exploiting the systematic nature of the error between the two theoretical methods. Thus, given the energy at the baseline theory, energy at the expensive level of theory could be obtained using the ML-learned additive correction term. Indeed, Δ-ML procedures have been shown to provide significantly better accuracy than models attempting to learn absolute energies directly,[28] thus allowing to reach chemical accuracy (±1.0 kcal mol^{-1}) and within striking distance of the elusive benchmark accuracy (±1.0 kJ mol^{-1}) with respect to the experimental value (or a high level of theory) through machine learning means. Therefore, we have also incorporated the Δ-ML model in our proposed machine learning protocol.

2. Methods

2.1 Data

The QM9-G4MP2 dataset is a collection of 133,296 molecules composed of C, N, O, F, and H atoms, with each molecule containing up to nine heavy atoms.[27, 30] The dataset provides the atomization energies of the molecules at B3LYP/6-31G(*2df,p*) (precursor for G4(MP2) computations) and G4(MP2) levels of theory, and thus is ideally suited to be used for the Δ-ML approach. Ward *et al.*[28] used a total of 130,258 molecules from the QM9-G4MP2 dataset, excluding the ones whose bond connectivity was found to be ambiguous. In their study, a random selection of 10% of molecules from the entire dataset (13,026 molecules) was chosen as the test set to validate the working of their machine learning models, viz., Schnet[21] and

FCHL.[28, 31, 33] To make a fair comparison with their results, we have also chosen the same training and test split.

2.1.1 Data Representation

The three-dimensional space (where a molecule 'lives') can be imagined as a cubic grid composed of *voxels*. Given the cartesian coordinates of a molecule, its atomic positions can be mapped onto the voxelized grid. In addition, any property associated with an atom viz., atom type (based on atomic number, aromaticity/aliphaticity, etc.), charge (or population), spin density, valence, hybridization, etc., can be directly embedded into one of the voxels based on its position in the 3D space. Formally, the 3D input representation of a molecule is a four-dimensional tensor (say, N×N×N×C), with the three equal indices (or dimensions) representing the voxel grid length (N) of the cube confining a molecule, and the remaining one representing the number of different features (or channels (C) in the context of convolutional neural networks (CNN)) associated with a given molecule. Thus, the embedded properties can act as molecular descriptors for a machine learning model to predict a chemical property of interest. However, a naïve mapping of the discreet atomic attributes to their corresponding voxels leads to a highly sparse tensor (or input representation) (Figure 1), which in turn may lead to an under-fitted model due to the lack of enough information to learn from, in the input representation. Such a performance degradation is caused due to sparse gradients being propagated through the network. Sparsity can be reduced to some extent by convolving the 3D molecular image with a gaussian or a wave-transform kernel, which imparts a smoothing (or blurring) effect to the input representation, thus approximately capturing inter-atomic interactions, while also providing a continuous feature representation.[34, 35] 3D sparse data is most efficiently represented through an octree data structure, where only the non-sparse regions (voxels) of a cubic volume is recursively partitioned into octants. Following this algorithm, a uniformly spaced voxelized data structure can be converted to one with numerically dense regions represented at fine resolutions and sparse spaces at low resolutions. The octree-based CNN[36, 37] proposed by Liu *et al*.[8] for the prediction of protein-ligand binding energies showed incredible performance gains in terms of memory usage and computation time; however, the model accuracy did not improve at high resolutions (< 1.0 Å), potentially due to the (quality of) molecular descriptors used being unsuitable for high resolutions. Therefore, molecular descriptors that are intrinsically dense in nature and contain meaningful

information at fine resolutions are needed. Naturally, a well-defined volumetric function depending on the atomic spatial positions would be an obvious choice, for example, the electrostatic potential due to nuclear charges. Alternatively, a molecular descriptor based on the electron probability distribution (electronic structure) of the molecule can also provide a non-sparse way of encoding molecular features into a spatial grid, and is the main focus of this paper.

Electron density, a scalar-valued function depending on the three spatial coordinates, is the primary observable associated with a molecule's electronic state. A plethora of electron density-based functions are available in the literature to extract physically interpretable information from a molecule's electronic structure. For the problem at hand of predicting atomization energies, which are highly dependent on the molecular geometry, an accurate picture of the bonding patterns in the molecule must be provided to the machine learning model. Therefore, in the present work, we have mainly explored the performance of the so-called electron localization functions, viz., LOL (Localized Orbital Locator),[17] and ELF (Electron Localization Function),[18] which are known to provide comprehensive topological information of a molecule.

ELF and LOL, developed by Becke and coworkers, are scalar functions providing a quantitative value to the degree of electron localization in space for a molecule. The idea behind the concept of localization functions is built on the premise of Pauli's exclusion principle, or more precisely, on the conditional probability of finding an electron with a given spin in the immediate vicinity of a reference electron with the same spin. The corresponding spherically averaged probability could be further shown to be directly proportional to the non-interacting kinetic energy density using the Taylor series expansion.[38, 39] For interpretation purposes, the expression for the conditional pair probability density in terms of kinetic energy density is scaled with respect to the kinetic energy density for the uniform electron gas and then mapped to a range of [0,1]. Physically, a low probability of finding another like-spin electron in the neighborhood of a reference electron implies high localizability of the reference electron in that region, which can also be interpreted as the reference electron being low in kinetic energy, and hence is termed as a "slow" electron. Such electrons are said to be highly localized within a region and are associated with those found in the core, bonding, and lone-

pair regions. Whereas a high pair probability corresponds to a high delocalizability of the reference electron, implying a high associated kinetic energy ("fast" electrons), and refers to the delocalized regions such as those found near orbital boundaries. Thus, a localization function cleanly partitions the molecular topology into electronically dense and diffuse regions, thus providing a chemically interpretable picture of a molecule akin to VSEPR and Lewis-dot theory. For the sake of visual comparison, a discrete voxel-based representation of a simple molecule (C_2N_2) is shown in Figure 1, and a voxelized LOL profile for the same molecule is shown in Figure 2. The two contrasting images depict the difference in sparsity levels in the two representations.

The information needed to compute localization functions or any other wavefunction-dependent molecular descriptor, viz., orbital coefficients, is usually stored in large data files (viz., checkpoint or *wfx* files in Gaussian 16), and hence are not included in curated datasets, potentially due to huge memory requirements. Therefore, to obtain the requisite descriptors, an additional electronic structure calculation on the full dataset is needed, which is probably one of the reasons why there has been a reluctance to use wavefunction-based descriptors in the machine learning models. Fortunately, localization functions depend only on the symmetry and nodal properties of the orbitals, making them topologically invariant with respect to the level of theory used.[40] In contrast to most population analysis methods, the level of theory does not change the qualitative nature of the molecular topology and, by extension, localization functions. Therefore, a simple computation such as a single-determinant small basis set or a semi-empirical method would be sufficient to provide learnable topological features of a molecule. In the present work, the localization functions used for molecular representation are generated using B3LYP/6-31G, a relatively cheap level of theory. Nevertheless, for the sake of comparison, the model's efficacy was tested with a large basis set (B3LYP/6-31G(*2df,p*)) generated localization functions as well. Additionally, we have also analyzed the performance of nuclear electrostatic potential (as a molecular descriptor) due to its dense nature and being independent of any electronic structure computation.

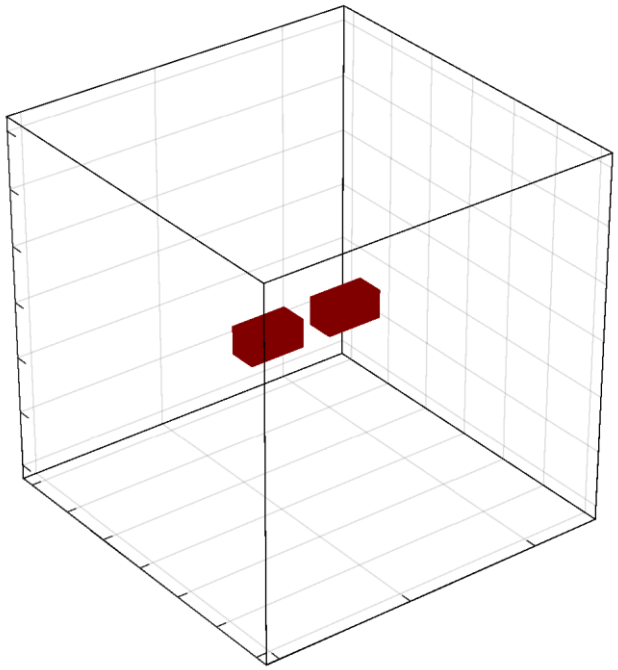

Figure 1. Discrete voxelized representation of C_2N_2, with the occupied voxels representing the atomic positions.

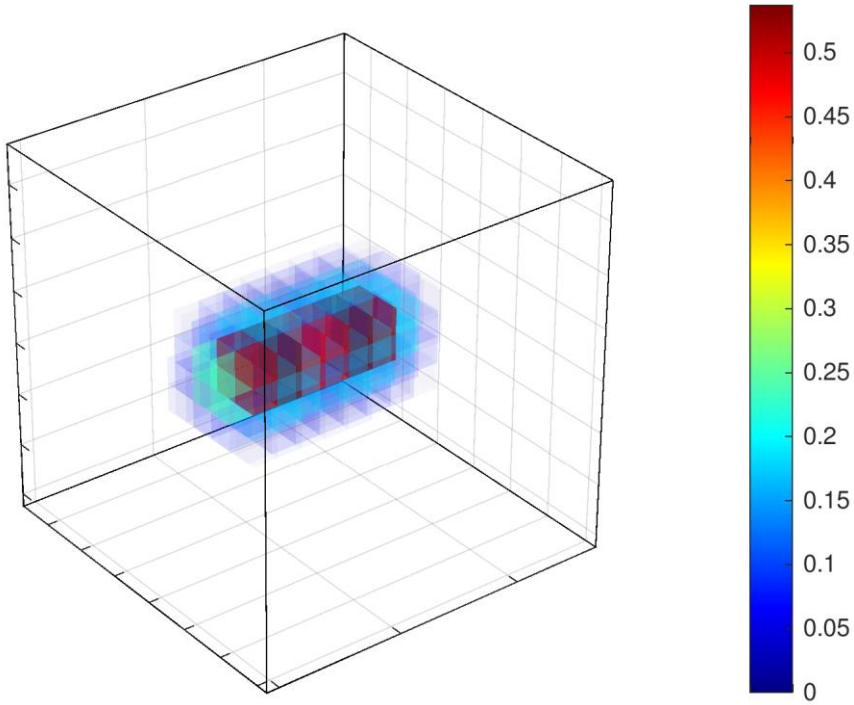

Figure 2. Voxelized LOL profile of C_2N_2. Larger values represent electron localized regions, while smaller values represent electron diffuse regions.

2.1.2 Data Preparation

A molecule can attain multiple orientations in three dimensions; therefore, to remove any ambiguity in the orientations between different molecules in the dataset, a unique orientation for each structure is needed. Although, it should be noted that multiple orientations for the same molecule can be incorporated into the dataset to increase the number of training samples. This is a well-known data augmentation technique often used in the field of computer vision if the dataset is scarce. Such data augmentations are feasible only if the input representations are rotationally invariant in space, thus uniquely setting 3D representations apart from other input types. Hence, while dealing with datasets with insufficient training samples, designing a machine learning framework based on 3D input representations could be advantageous compared to other options. However, the QM9-G4MP2 is a reasonably large dataset (~130,000 data points); therefore, no data augmentation procedure was incorporated during data pre-processing. A unique molecular orientation for every molecule was obtained using the Principal Component Analysis (PCA) algorithm, which could be used to provide a new set of molecular coordinates at which the variance in the (heavy atom) x-coordinates is maximum. Thus, the molecule is oriented along the first principal component (or x-axis in this case). The reoriented molecules thus obtained could be enclosed in a cubic box of dimension 10.4 Å (with the geometric center of the cube taken as the origin), which is large enough to encompass all the heavy atoms of any molecule in the dataset.

Grid resolution determines how finely the topological details of a molecule are encoded in the voxelized grid, and can be formally defined as the dimension of a single voxel cell. The number of uniformly spaced voxels along a grid dimension is known as the grid length (N) and determines the size of the 4D-input tensor. For a cube of fixed dimensions, the larger the grid length, the higher would be the grid resolution, and thus more would be the topological information embedded into the grid, which should theoretically improve model accuracy. However, the computational cost of training a convolutional neural network (CNN) roughly scales as the cubic power of the grid length. Therefore, grid length (or grid resolution) should be carefully chosen, keeping in mind the available computational resources. We used a grid length of 14 (grid resolution=0.743 Å) to construct the input tensors for model training;

however, we also experimented with multiple grid lengths (or equivalently, grid resolutions) to ascertain their correlation with the model performance.

The requisite molecular descriptors (viz., LOL, ELF, and nuclear electrostatic potential (NEP)) were obtained using the Multiwfn program[41] in a 3D grid format from the Gaussian 16[42] generated *wfx* files. The generated data was then converted to a 4D tensor (N×N×N×1), suitable to be used as an input for a 3D-CNN. Data preparation scripts are available in the paper's GitHub repository.

2.2 Model

2.2.1 Architecture

DenseNet[43] model architecture, known for its parameter efficiency and ease of training, was employed for the task of learning the molecular topology. Most computer vision architectures, including DenseNets, were developed for 2D image recognition tasks where the input shape is a three-dimensional tensor. Thus, we modified the standard DenseNet architecture accordingly to make it compatible with 3D input representations. A schematic diagram of the basic DenseNet architecture used is shown in Figure 3. DenseNet introduces what is known as dense-blocks into the network architecture, which are composed of the so-called dense-layers, which in turn is a stack of 1×1 and 3×3 convolution layers, reminiscent of the bottleneck-block in ResNets.[44] The defining trait of a DenseNet architecture is the dense connectivity pattern within a dense block, wherein every dense layer is directly connected to every other dense layer through a concatenation operation. Mathematically, the feature maps generated by a dense layer are concatenated with those produced by all the preceding layers, which are then passed as an input to the next layer in the architectural hierarchy. In this way, the features learned by the shallower layers are transferred to the deeper layers, thus enhancing the learning process. Since features are being reused throughout the network, only a small number of new features (or channels) need to be added by every dense layer, making DenseNets, parameter efficient by design and hence less susceptible to overfitting. Although quite simple in concept, the densely connected topology of DenseNets makes it robust to the vanishing gradient problem, and boosts information and gradient flow. For the sake of simplicity, only four dense layers are included in the dense block shown in Figure 3. Any two adjacent dense blocks are connected through a transition layer, which downsamples the feature

maps through the average-pooling operation. Data downsampling is often necessary to train deeper networks without proliferating the number of FLOPs, and hence training time, albeit at the cost of some loss in resolution (or information). The compactness of the DenseNet architecture is further increased by reducing the number of incoming channels (C_{in}) in the transition-block by a factor, called the compression factor (*cf*), with its domain being $0 < cf \leq 1$, which provides further computational efficiency to the model without compromising accuracy to a large extent. A *cf* value of 0.5 was found to be optimal for providing a reasonable balance between model cost and accuracy.[43] All the architectural hyperparameters except the number of dense layers in each of the dense blocks are depicted in Figure 3. We experimented with different dense-layer configurations in the two dense blocks, which determines the overall depth of the architecture, and is one of the primary factors dictating the model's overall performance and cost. Henceforth, the number of dense-layers in the first and second dense-block are referenced as d_1 and d_2, respectively, and is collectively denoted as (d_1, d_2), representing the dense-block configuration of a DenseNet architecture. For example, a dense-block configuration of (16, 8) implies 16 dense layers in the first dense block and 8 dense layers in the second dense block.

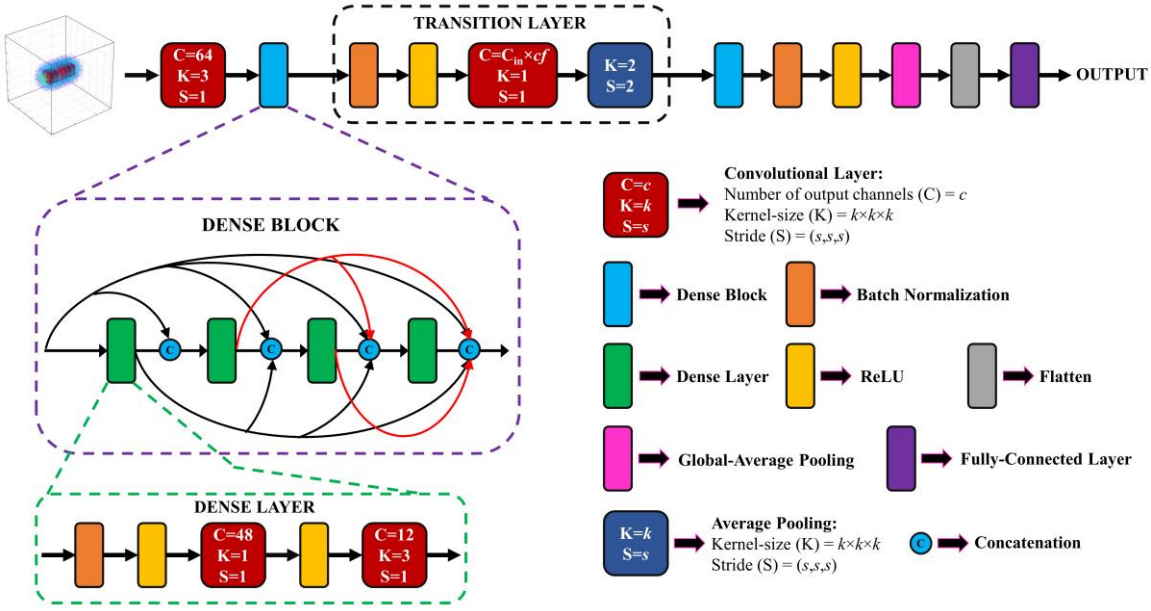

Figure 3. Schematic diagram of the DenseNet architecture.

2.2.2 Training

The entire machine learning workflow was implemented in PyTorch-Lightning,[45] with PyTorch[46] as backend. The 3D-DenseNet code was adapted from the publicly available memory-efficient version of Densenet implemented in PyTorch by Pleiss *et al.*.[47] All of the models were trained in parallel on four NVIDIA V100 GPUs with a combined batch size of 128. The mean absolute error (MAE) is chosen as the loss function for model training. The model parameters were optimized using the SGD (Stochastic Gradient Descent) algorithm in conjunction with Nesterov momentum (0.9) using a weight decay parameter (L2 penalty) of 1.0×10^{-4}. During the optimization procedure, the learning rate is controlled through a learning rate schedule that decreases the learning rate by a factor of 0.75 whenever the training loss plateaus within a certain threshold (0.005 kcal mol^{-1}). The model optimization was initialized with a starting learning rate of 0.1 to run for 250 epochs, enough for both training and test metric values to converge comfortably. While benchmarking the performance of a hyperparameter of interest, the corresponding trials were run under fixed random seed conditions to eliminate any variability whatsoever due to dissimilar weight initializations. However, due to the non-deterministic nature of certain GPU algorithms, a small degree of variance is still inevitably introduced between different runs; hence all the reported metrics were obtained using an average of five different runs.

3. Results and Discussion

Following the Δ-ML philosophy, the proposed machine learning model is trained to reproduce the difference in the atomization energies between the G4(MP2) and B3LYP/6-31G(*2df,p*) levels of theory. The optimized model could then be used to predict the Δ-atomization-energy values for out-of-sample cases, which in turn could be used to predict their absolute atomization energies at the G4(MP2) level of theory, provided the corresponding atomization energies at B3LYP/6-31G(*2df,p*) level are known. The predicted values for an out-of-sample dataset by the model network, however, must be within a reasonable error threshold to be of any practical use, and is indicative of the quality of a model. Therefore, the mean absolute error (MAE) between the ML-predicted values and the exact values over the test set (13,026 molecules) is used as the metric to quantify the performance of a given model. The model performance usually depends on a number of model and data-related

hyperparameters; therefore, the effect of varying a few seemingly important hyperparameters is reported in this section, thus gleaning insight into different ways that systematically improve model performance.

3.1 Effect of Varying the Voxel Grid Length (or Grid Resolution)

The input tensor's shape and size depend on the grid length (or equivalently, grid resolution), which ultimately governs the quality of topological information encoded in the grid. However, the number of FLOPs associated with a convolutional neural network formally scales as the cubic power of the grid length (Figure 4b). Therefore, selecting an appropriate grid length is imperative if the computational resources are scarce. To assess the performance of the model as a function of the change in the grid length, the topological descriptors are generated with different grid lengths (N) viz., 12, 14, and 16, with the corresponding grid resolutions being 0.867 Å, 0.743 Å, and 0.650 Å, respectively. These input representations are then used to train the base DenseNet architecture (Figure 3) with a fixed dense block configuration of (16, 16). The obtained metrics summarized in Figure 4a clearly show an improvement in model performance with an increase in grid length. All three molecular descriptors improve model generalizability at finer grid resolutions. Indeed, the higher the input grid resolution, the more detailed interatomic features would be available to the model to help it discern between different molecular patterns. Comparing the two localization functions' performance, LOL provides superior results than ELF in all cases. Additionally, the performance of nuclear electrostatic potential (NEP) is comparable to that of LOL. More importantly, all the errors are well below the desired benchmark accuracy of 1.0 kJ mol^{-1}, with the N=16 errors being comparable to the best result obtained by Ward *et al.*[28], i.e., 4.5 meV (= 0.43 kJ mol^{-1} or 0.104 kcal mol^{-1}).

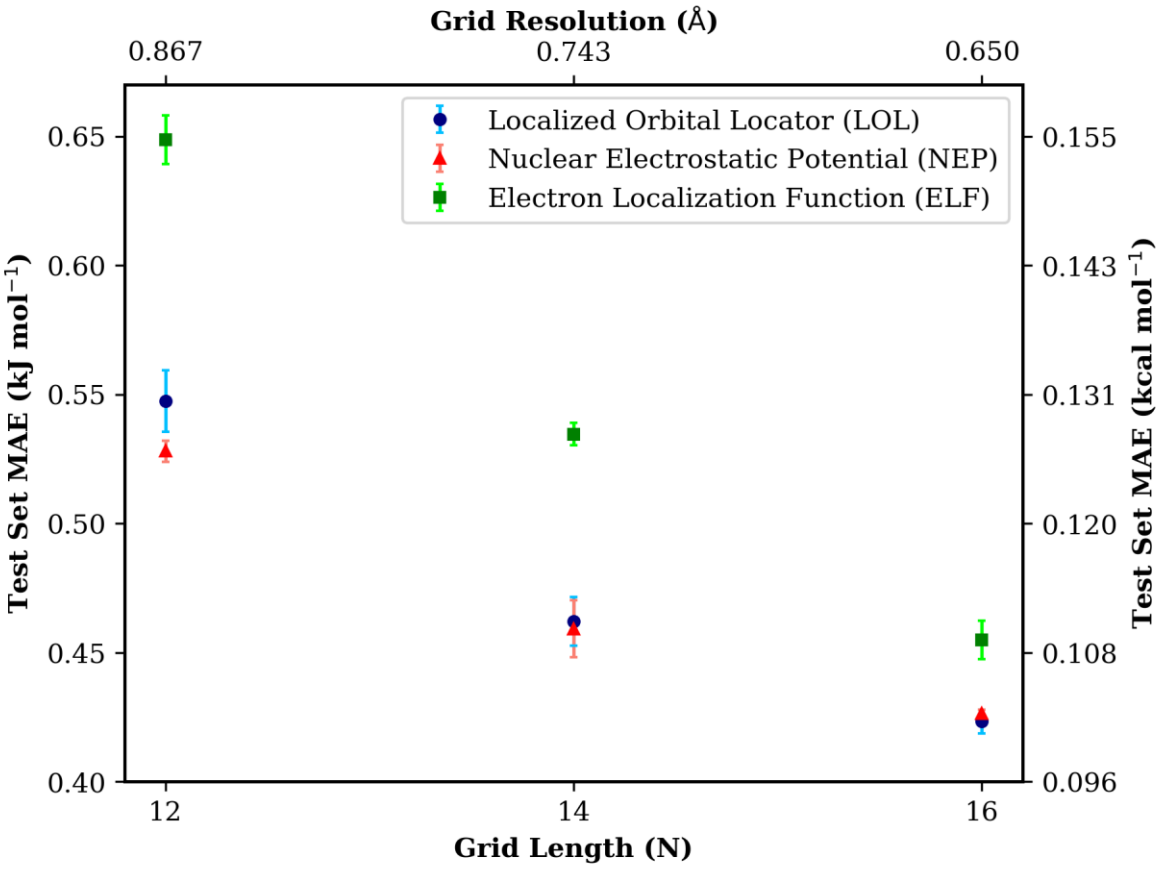

Figure 4a. Effect of varying grid length (N) on the MAE of the test set. All results were obtained using the base DenseNet architecture (Figure 3) with a dense block configuration of (16,16).

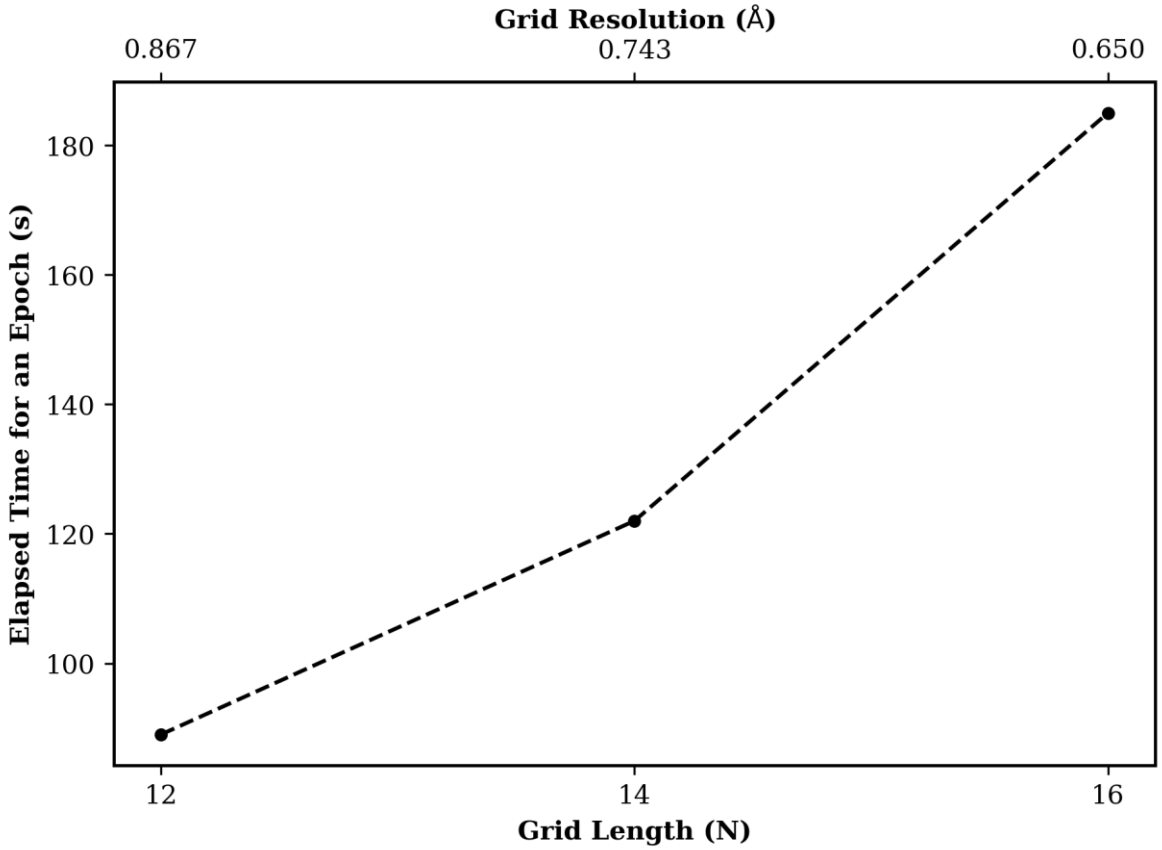

Figure 4b. Increase in elapsed time for an epoch with change in grid length. All results were obtained using the base DenseNet architecture (Figure 3) with a dense block configuration of (16,16).

3.2 Effect of Varying the Training Set Size

High volume and quality of data are essential to increase the generalization capability of a machine learning model. To decipher the extent of correlation between the amount of data and model accuracy, we experimented with multiple training set sizes keeping the architectural and data hyperparameters fixed. To be more precise, we prepared training sets of various sizes, viz., 25,000, 50,000, 75,000, and 117,232 (full train set), keeping the grid length for the input representation at 14, which provides a reasonable balance between cost and accuracy. The (16,16)-DenseNet architecture was used to test the variation in the model performance. The mean absolute error (MAE) of the test set as a function of the training set size is depicted in Figure 5. As expected, the model performance improves with an increase in the training set size. Moreover, even with a relatively small training set composed of only 25,000 samples, the model achieves a respectable accuracy of approximately 1.2 kJ mol^{-1} (=0.287 kcal mol^{-1}),

which could be useful in situations with limited compute availability. Interestingly, both LOL and nuclear electrostatic potential (NEP) provide similar quantitative results with respect to change in the training set size, indicating further similarity between the efficacies of the two descriptors.

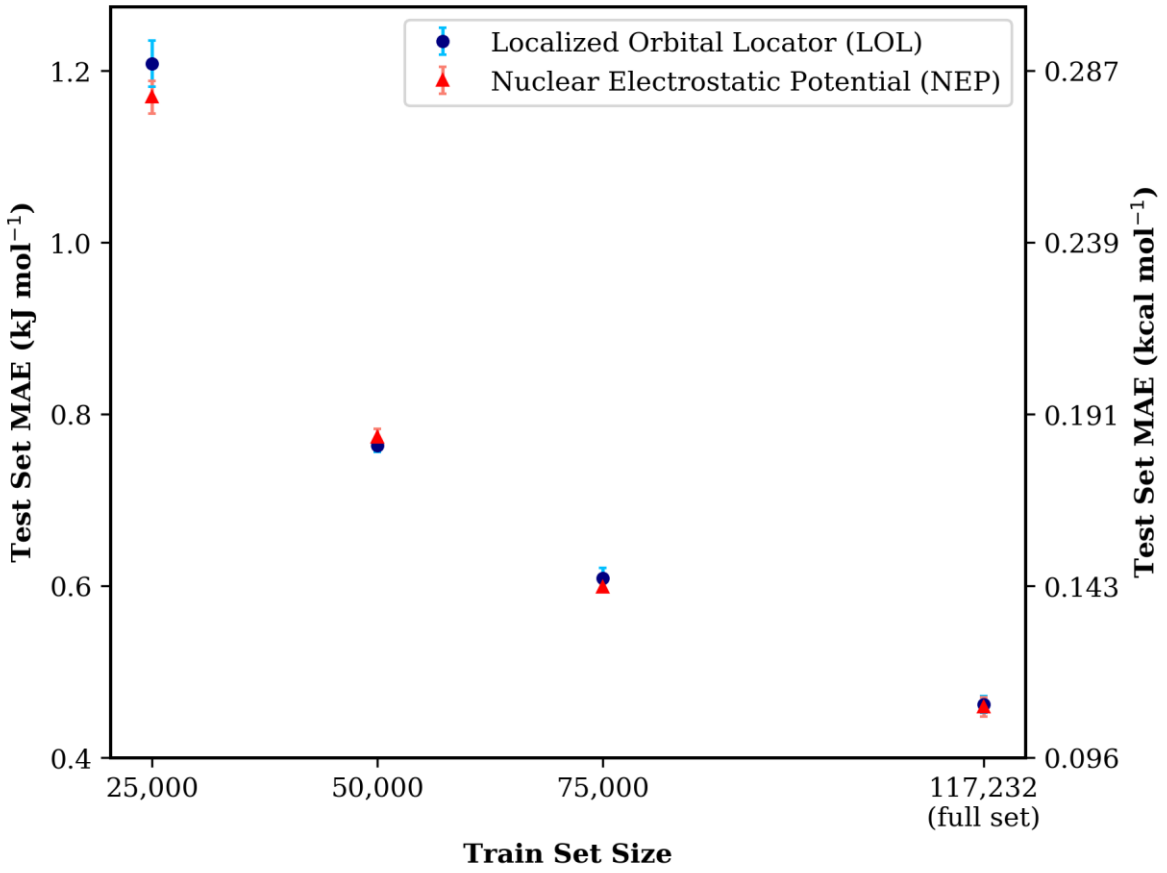

Figure 5. Effect of varying the training set size on the MAE of the test set. All results were obtained using the base DenseNet architecture (Figure 3) with a dense block configuration of (16,16).

3.3 Effect of Varying the Dense Block Configuration

The depth of a dense block refers to the number of dense layers (or convolution layers) it is composed of. The depth of the *first* dense block (d_1) is a hyperparameter of critical importance since it is one of the primary determining factors of the computational cost associated with a DenseNet architecture. All the convolutional layers in the first dense block act on the full *un-downsampled* input tensor, making its associated FLOP count substantially larger than that for the second dense block. The latter operates only on a downsampled version of the data, thus losing some of the topological information. Therefore, for the model to learn

as many high-level input features as possible, the first dense block needs to be as deep as computationally feasible. In short, increasing the number of layers in the first dense block should theoretically improve model accuracy but at an associated computational cost. To measure model sensitivity as a function of the change in the first dense block's depth, DenseNet models with different d_1 values (viz., 12, 16, 20, 24) were prepared, keeping the total depth (d_1+d_2) of the network fixed at 32. The number of dense layers in the second dense block (d_2) is varied accordingly to keep the overall depth constant across all the models. It should be noted that even though the dense block configuration is different, the total number of trainable parameters remains the same across the different models, as it depends only on the total depth of the model. As predicted, the test error decreases with an increase in the depth of the first dense block (Figure 6a).

Furthermore, we also experimented varying d_1 (viz., 8, 16, 24, 32) while keeping d_2 constant (at 8) (Figure 6b). Finally, we also show results obtained by simultaneously doubling the number of layers in each of the dense blocks (Figure 6d). In both of these cases, the overall depth of the network is being increased, causing a lowering of test set MAE. Clearly, out of all the models tested, the best performing (and also the most expensive) model is the one with the most number of dense layers, i.e., a dense block configuration of (32, 32), which provides an MAE of 0.094 ± 0.002 kcal mol^{-1}; however, it should be reiterated that errors could potentially be lowered further by increasing the architecture depth and/or using a finer grid resolution. All the results shown are obtained using the LOL descriptor; however, the general trend is expected to be the same for other topological descriptors as well. The training time for an epoch for different models is shown in Figures 6c and 6e, roughly scaling linearly with respect to the number of dense layers (or convolutional layers) in the model. In summary, the model performance could be systematically improved by increasing the depth of the architecture, which, however, is accompanied by an increase in the computational cost.

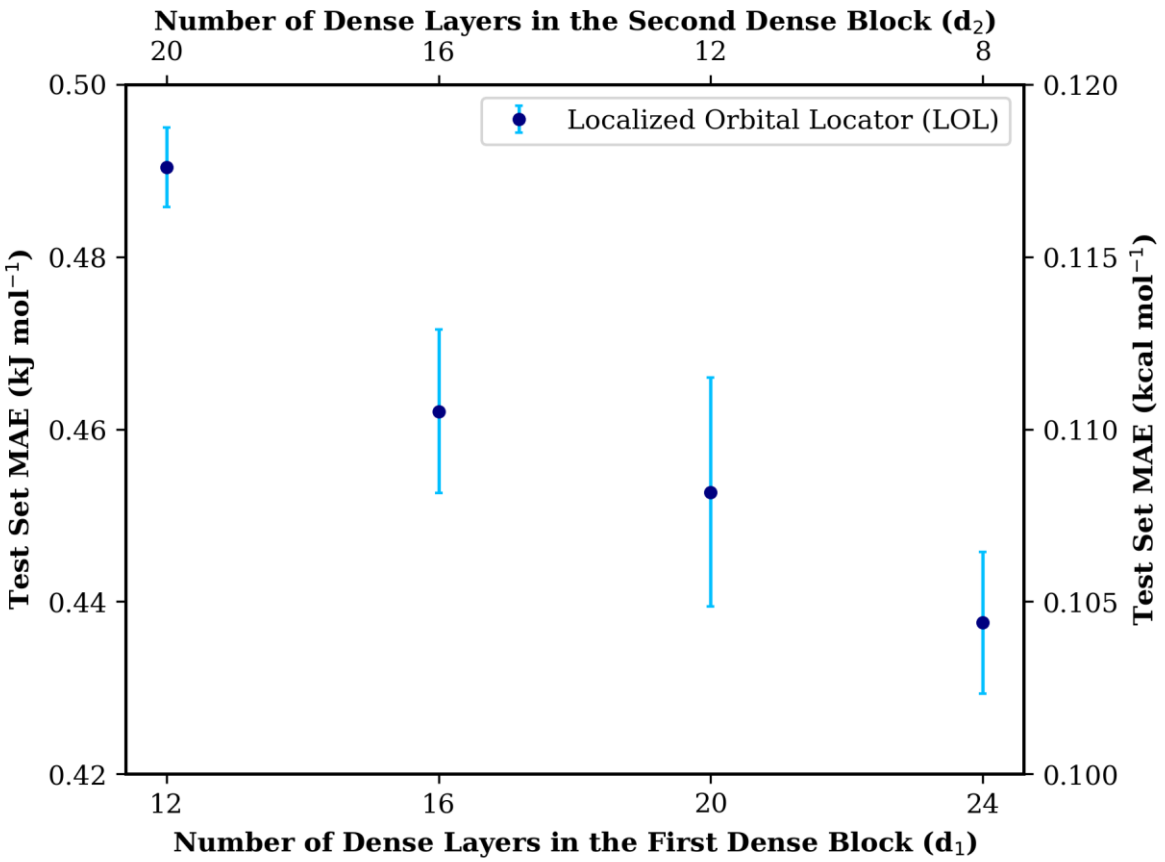

Figure 6a. Effect of varying the depth of the first dense block on the test set MAE while keeping the architecture's total depth constant. All results were obtained using the base DenseNet architecture (Figure 3) with the total number of dense layers fixed at 32.

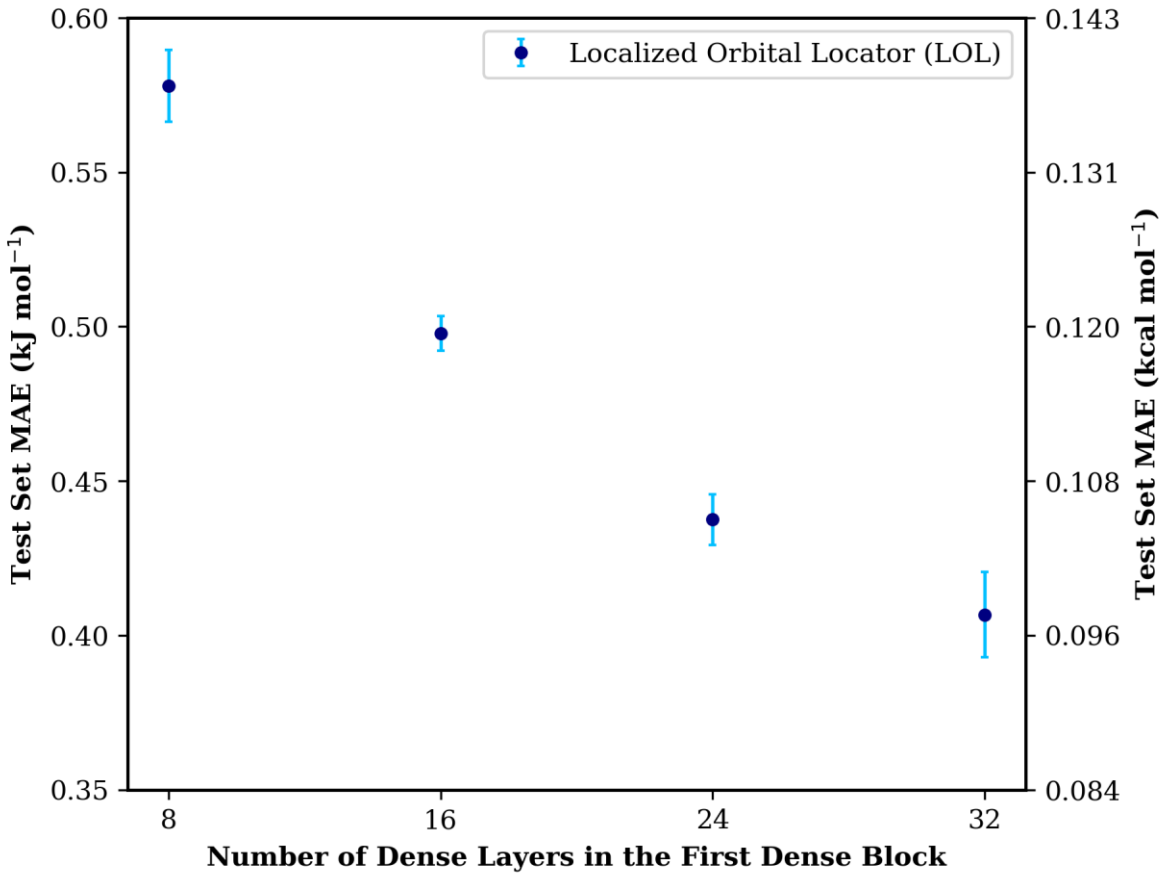

Figure 6b. Effect of varying the depth of the first dense block on the test set MAE. All results were obtained using the base DenseNet architecture (Figure 3) with the number of dense layers of the second dense block fixed at 8.

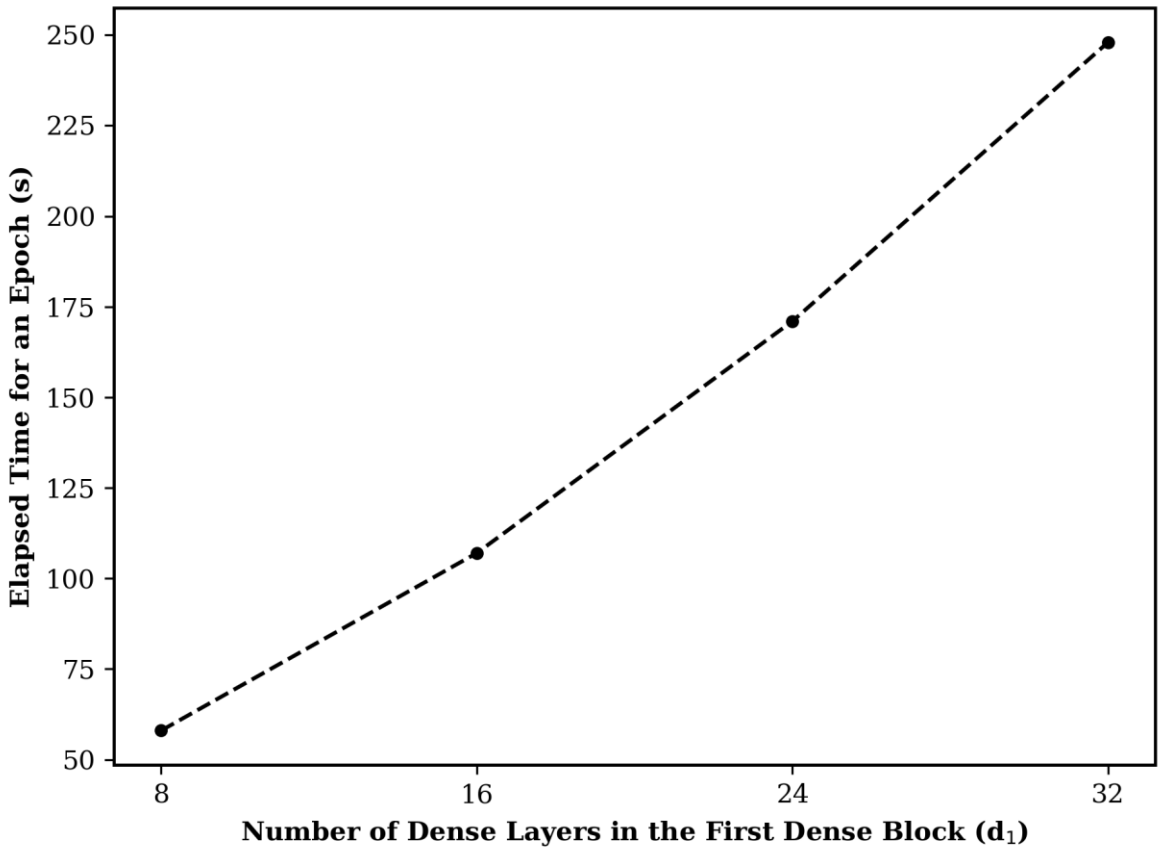

Figure 6c. Change in elapsed time for an epoch with change in depth of the first dense block. All results were obtained using the base DenseNet architecture (Figure 3) with the number of dense layers of the second dense block fixed at 8.

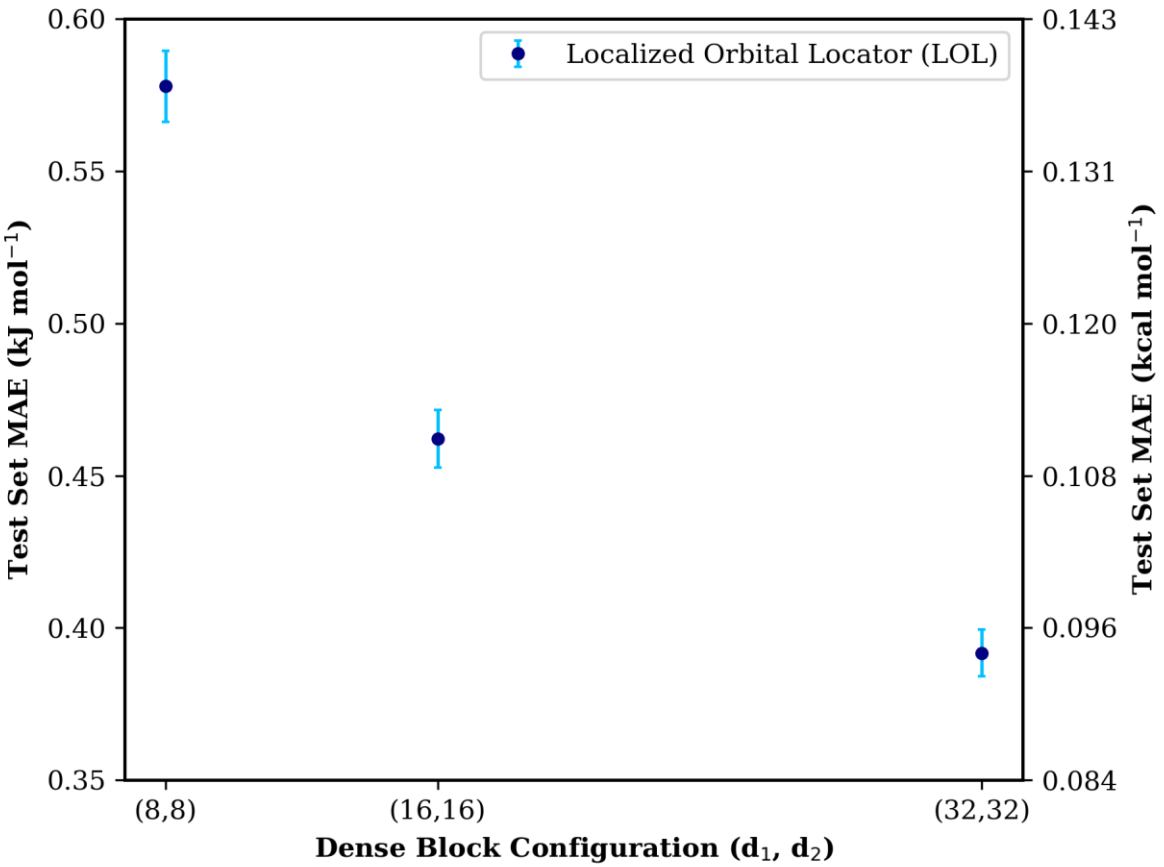

Figure 6d. Effect of doubling the dense block configuration on the test set MAE. All results were obtained using the base DenseNet architecture (Figure 3).

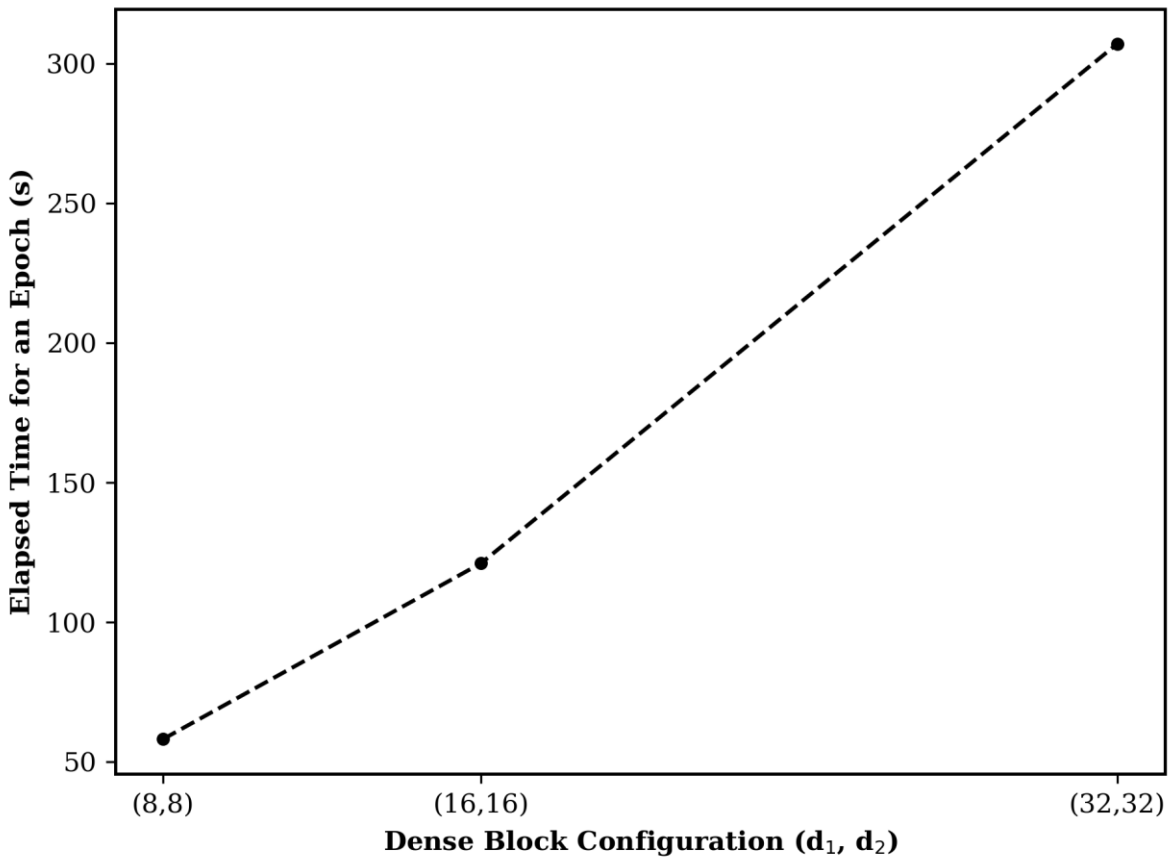

Figure 6e. Change in elapsed time for an epoch with the doubling of the dense block configuration. All results were obtained using the base DenseNet architecture (Figure 3).

3.4 Effect of Varying the Level of Theory to Generate the Localization Functions

Electronic-wave function-dependent topological functions (viz., LOL, and ELF) are obtained through an electronic structure computation and, as such, depend quantitatively on the level of theory used. To test whether the level of theory affects the model performance or not, the localization functions (viz., LOL and ELF) were generated using two different levels of theory (or basis sets), viz., B3LYP/6-31G, and B3LYP/6-31G(*2df,p*), which were then used as inputs to train the standard (16,16)-DenseNet architecture. From the results shown in Figure 7, it is apparent that the results vary negligibly between the two levels of theory. Moreover, as noted before, LOL outperforms ELF at both levels of theory. Thus, the model performance does not rely heavily on the quantitative nature of the localization functions but rather on its

qualitative aspects, further reinforcing the idea of network learning from broad topological features.

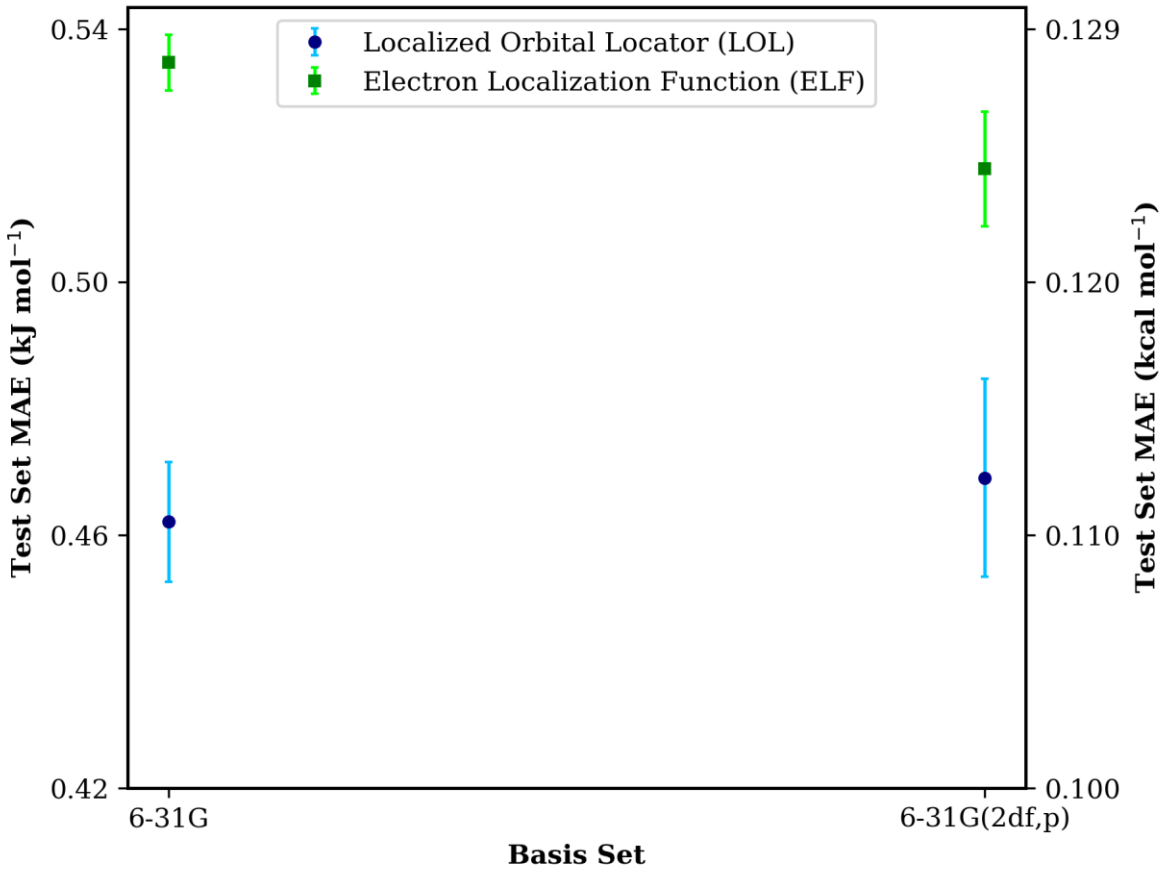

Figure 7. Effect of varying the basis set used to generate the localization functions on model performance. All results were obtained using the base DenseNet architecture (Figure 3) with a dense block configuration of (16,16).

3.5 Effect of Using Multiple Descriptors.

A single molecular descriptor is usually insufficient to capture every molecular detail and thus may lack enough learnable data to provide expected accuracy levels, especially in case of a challenging problem like protein-ligand binding energy predictions. However, for the problem of predicting atomization energies, a single topological feature by itself proved sufficient to provide excellent results. Nonetheless, we tested the model performance for multi-channel inputs as well, which could be obtained by stacking individual input tensors along the channel axis. Specifically, two different combinations from the available topological features were formed, viz., LOL+NEP, and ELF+LOL+NEP, where the '+' sign indicates a

concatenation operation between any two topological tensors. The concatenated inputs were then used to train the base DenseNet network (Figure 3) with a (16,16) dense block configuration. As depicted in Figure 8, the test MAE did not reduce much upon providing more learnable features to the network, potentially due to information overlap between the three topological descriptors, thus leading to redundant features being added to the input upon their concatenation. This observation can also be attributed to the test loss saturation with respect to the network architecture and could mean that a deeper or wider architecture is required to learn the additional features. Due to computational considerations, only three topological descriptors are tested; however, a myriad of other discrete and dense molecular descriptors existing in literature could also be used in different combinations to construct a grid representation of a molecule. In fact, for a set of *n* distinct molecular descriptors, a total of $(2^n - 1)$ input combinations could be obtained, thus making the scaling exponential in the space of molecular descriptors. A thorough benchmark of the architecture's learning capacity with respect to a more extensive set of input features will be pursued in a future publication.

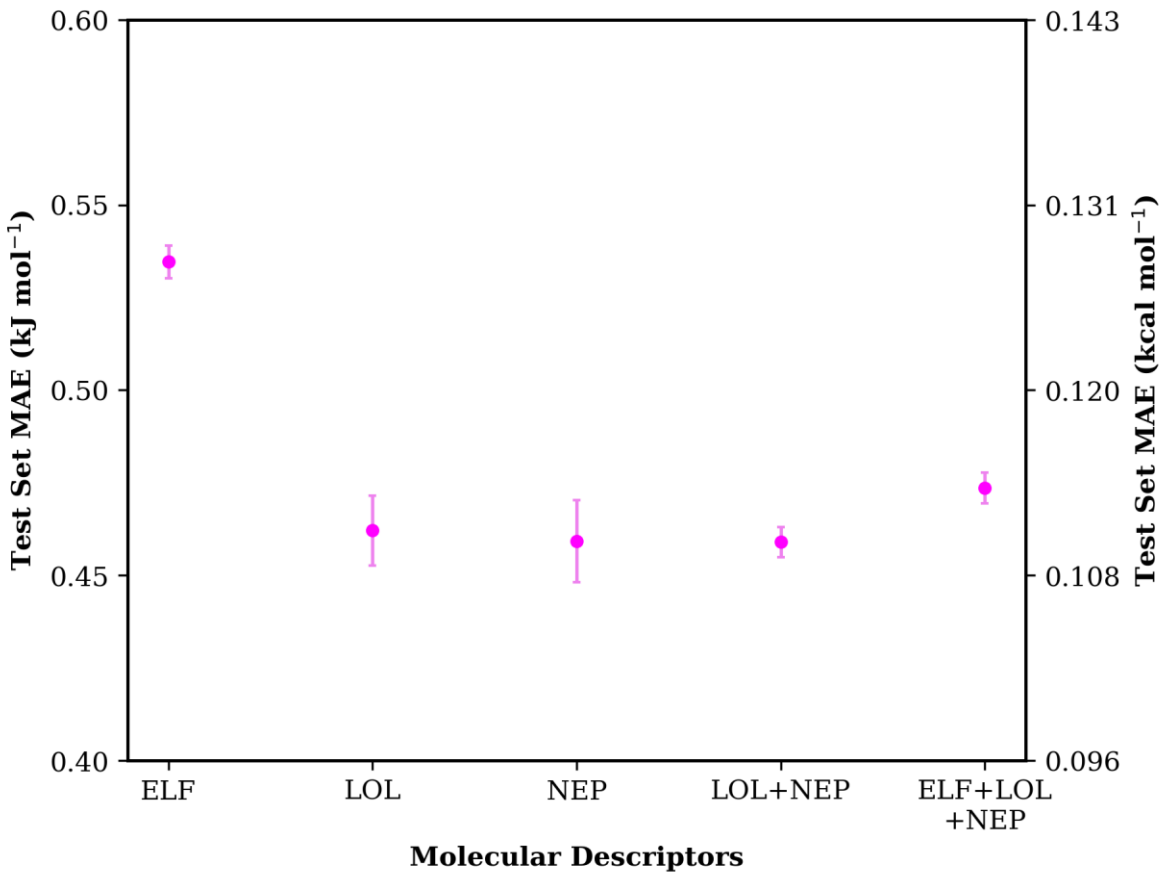

Figure 8. A performance comparison between different molecular descriptor combinations. All results were obtained using the base DenseNet architecture (Figure 3) with a dense block configuration of (16,16).

4 Conclusions

The present article highlights the importance of using non-sparse molecular descriptors for a machine learning task utilizing the 3D-CNN framework. The 3D-DenseNet architecture successfully learned the subtle molecular topological features encoded in the localization functions and correlated them with the Δ-atomization energies. Moreover, the network was also able to learn the structural features through the nuclear electrostatic potential (NEP) of a molecule. Furthermore, we analyzed the proposed model's performance with respect to several key hyperparameters, some of which helped improve model accuracy in a systematic manner. Among the localization functions tested, LOL outperformed ELF in all instances, indicating the former's superiority over the latter in providing a clear topological picture of a molecule,

as noted in several other publications as well.[48-51] Moreover, NEP performed comparably to LOL, potentially due to its relatively denser input representation, and could be a cheaper alternative to LOL as it does not require any additional electronic structure computations. Nevertheless, it is likely that there are cases where NEP will fall short, such as in problems involving open-shell species or electronic transitions, where the electron distribution is known to play a critical role in property determination. Moreover, datasets composed of transition metal species, which often involve multiple energetically accessible spin states, may present a situation where a single *atomic* configuration (but having different *electronic* configurations) has multiple corresponding target values associated with it, differing only due to subtle changes in the electronic structure of the molecule. Fortunately, localization functions provide a novel way to visualize alpha and beta electron topologies (or distributions) separately, thus making it easier to locate regions associated with unpaired electrons, which could help predict properties such as redox potentials and ionization energies.[52, 53] Additionally, localization functions have been widely used to characterize the bonding nature in transition metal complexes and thus could be an indispensable tool for their tensorial representation.[53-60]

The encouraging results in the present article show incredible promise for future endeavors to tackle even more challenging problems. The existing frameworks could be further refined by leveraging recent developments in the field of computer vision. For example, as noted earlier, the 3D-CNNs can be made further computationally efficient by taking advantage of the octree data structure. Additionally, the latest and ever-improving state-of-the-art network architectures could be adopted for learning tasks; however, challenges remain, as the training protocols for chemistry-related problems could be quite different from those for image classification tasks. The model accuracy could be further improved through data augmentation techniques, which could be quite valuable while dealing with small datasets. The future directions concerning these deep learning frameworks should also be directed towards solving problems of practical interest such as predicting ligand-receptor binding affinities, thus providing complementary ways to enhance high throughput virtual screening (HTVS) methods.

5 Acknowledgments

We acknowledge support from the National Science Foundation grant CHE-1665427 at Indiana University. The computations carried out in this work were enabled in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

Supporting Information

All code needed to reproduce this study will be made available on GitHub upon paper acceptance.

Corresponding Authors: Krishnan Raghavachari and Ankur Gupta

*E-mails: kraghava@indiana.edu and ankkgupt@indiana.edu

ORCID

Krishnan Raghavachari: 0000-0003-3275-1426

Notes

The authors declare no competing financial interest.

References

1. https://paperswithcode.com/sota/image-classification-on-imagenet.
2. Jiménez, J.; Doerr, S.; Martínez-Rosell, G.; Rose, A. S.; De Fabritiis, G., DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* **2017,** *33* (19), 3036-3042.
3. Hochuli, J.; Helbling, A.; Skaist, T.; Ragoza, M.; Koes, D. R., Visualizing convolutional neural network protein-ligand scoring. *Journal of Molecular Graphics and Modelling* **2018,** *84*, 96-108.
4. Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R., Protein–Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2017,** *57* (4), 942-957.

5. Mahmoud, A. H.; Masters, M. R.; Yang, Y.; Lill, M. A., Elucidating the multiple roles of hydration for accurate protein-ligand binding prediction via deep learning. *Communications Chemistry* **2020,** *3* (1), 19.
6. Hassan-Harrirou, H.; Zhang, C.; Lemmin, T., RosENet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2020,** *60* (6), 2791-2802.
7. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G., KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2018,** *58* (2), 287-296.
8. Liu, Q.; Wang, P.-S.; Zhu, C.; Gaines, B. B.; Zhu, T.; Bi, J.; Song, M., OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. *Journal of Molecular Graphics and Modelling* **2021,** *105*, 107865.
9. Wallach, I.; Dzamba, M.; Heifets, A., AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855* **2015**.
10. Pu, L.; Govindaraj, R. G.; Lemoine, J. M.; Wu, H.-C.; Brylinski, M., DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS computational biology* **2019,** *15* (2), e1006718.
11. Torng, W.; Altman, R. B., High precision protein functional site detection using 3D convolutional neural networks. *Bioinformatics* **2018,** *35* (9), 1503-1512.
12. Simonovsky, M.; Meyers, J., DeeplyTough: Learning Structural Comparison of Protein Binding Sites. *Journal of Chemical Information and Modeling* **2020,** *60* (4), 2356-2366.
13. Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P., Improving detection of protein-ligand binding sites with 3D segmentation. *Scientific Reports* **2020,** *10* (1), 5035.
14. Homer, E. R.; Hensley, D. M.; Rosenbrock, C. W.; Nguyen, A. H.; Hart, G. L. W., Machine-Learning Informed Representations for Grain Boundary Structures. *Frontiers in Materials* **2019,** *6* (168).
15. Kajita, S.; Ohba, N.; Jinnouchi, R.; Asahi, R., A Universal 3D Voxel Descriptor for Solid-State Material Informatics with Deep Convolutional Neural Networks. *Scientific Reports* **2017,** *7* (1), 16991.
16. Liu, S.; Li, J.; Bennett, K. C.; Ganoe, B.; Stauch, T.; Head-Gordon, M.; Hexemer, A.; Ushizima, D.; Head-Gordon, T., Multiresolution 3D-DenseNet for Chemical Shift Prediction in NMR Crystallography. *The Journal of Physical Chemistry Letters* **2019,** *10* (16), 4558-4565.

17. Schmider, H. L.; Becke, A. D., Chemical content of the kinetic energy density. *Journal of Molecular Structure: THEOCHEM* **2000,** *527* (1), 51-61.
18. Becke, A. D.; Edgecombe, K. E., A simple measure of electron localization in atomic and molecular systems. *The Journal of Chemical Physics* **1990,** *92* (9), 5397-5403.
19. Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M., A fifth-order perturbation comparison of electron correlation theories. *Chemical Physics Letters* **1989,** *157* (6), 479-483.
20. Curtiss, L. A.; Redfern, P. C.; Raghavachari, K., Gaussian-4 theory. *The Journal of Chemical Physics* **2007,** *126* (8), 084108.
21. Schütt, K. T.; Sauceda, H. E.; Kindermans, P. J.; Tkatchenko, A.; Müller, K. R., SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **2018,** *148* (24), 241722.
22. Unke, O. T.; Meuwly, M., PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *Journal of Chemical Theory and Computation* **2019,** *15* (6), 3678-3693.
23. Klicpera, J.; Groß, J.; Günnemann, S., Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* **2020**.
24. Liu, Z.; Lin, L.; Jia, Q.; Cheng, Z.; Jiang, Y.; Guo, Y.; Ma, J., Transferable Multilevel Attention Neural Network for Accurate Prediction of Quantum Chemistry Properties via Multitask Learning. *Journal of Chemical Information and Modeling* **2021,** *61* (3), 1066-1082.
25. Qiao, Z.; Welborn, M.; Anandkumar, A.; Manby, F. R.; Miller, T. F., OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *The Journal of Chemical Physics* **2020,** *153* (12), 124111.
26. Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L., Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling* **2012,** *52* (11), 2864-2875.
27. Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A., Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014,** *1* (1), 140022.
28. Ward, L.; Blaiszik, B.; Foster, I.; Assary, R. S.; Narayanan, B.; Curtiss, L., Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations. *MRS Communications* **2019,** *9* (3), 891-899.
29. Curtiss, L. A.; Redfern, P. C.; Raghavachari, K., Gaussian-4 theory using reduced order perturbation theory. *The Journal of Chemical Physics* **2007,** *127* (12), 124105.
30. Narayanan, B.; Redfern, P. C.; Assary, R. S.; Curtiss, L. A., Accurate quantum chemical energies for 133 000 organic molecules. *Chemical Science* **2019,** *10* (31), 7449-7455.

31. Faber, F. A.; Christensen, A. S.; Huang, B.; von Lilienfeld, O. A., Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of Chemical Physics* **2018,** *148* (24), 241717.
32. Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A., Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *Journal of Chemical Theory and Computation* **2015,** *11* (5), 2087-2096.
33. Christensen, A. S.; Bratholm, L. A.; Faber, F. A.; Anatole von Lilienfeld, O., FCHL revisited: Faster and more accurate quantum machine learning. *The Journal of Chemical Physics* **2020,** *152* (4), 044107.
34. Kuzminykh, D.; Polykovskiy, D.; Kadurin, A.; Zhebrak, A.; Baskov, I.; Nikolenko, S.; Shayakhmetov, R.; Zhavoronkov, A., 3D Molecular Representations Based on the Wave Transform for Convolutional Neural Networks. *Molecular Pharmaceutics* **2018,** *15* (10), 4378-4385.
35. Torng, W.; Altman, R. B., 3D deep convolutional neural networks for amino acid environment similarity analysis. *BMC Bioinformatics* **2017,** *18* (1), 302.
36. Wang, P.-S.; Liu, Y.; Guo, Y.-X.; Sun, C.-Y.; Tong, X., O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)* **2017,** *36* (4), 1-11.
37. Riegler, G.; Osman Ulusoy, A.; Geiger, A. In *Octnet: Learning deep 3d representations at high resolutions*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp 3577-3586.
38. Becke, A. D., Hartree–Fock exchange energy of an inhomogeneous electron gas. *International Journal of Quantum Chemistry* **1983,** *23* (6), 1915-1922.
39. Becke, A. D., Local exchange-correlation approximations and first-row molecular dissociation energies. *International Journal of Quantum Chemistry* **1985,** *27* (5), 585-594.
40. Fuentealba, P.; Chamorro, E.; Santos, J. C., Chapter 5 Understanding and using the electron localization function. In *Theoretical and Computational Chemistry*, Toro-Labbé, A., Ed. Elsevier: 2007; Vol. 19, pp 57-85.
41. Lu, T.; Chen, F., Multiwfn: A multifunctional wavefunction analyzer. *Journal of Computational Chemistry* **2012,** *33* (5), 580-592.
42. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda,

R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. *Gaussian 16 Rev. C.01*, Wallingford, CT, 2016.

43. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. In *Densely connected convolutional networks*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp 4700-4708.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp 770-778.
45. Falcon, W. a. a., PyTorch Lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* **2019,** *3*.
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L., PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* **2019,** *32*, 8026-8037.
47. Pleiss, G.; Chen, D.; Huang, G.; Li, T.; van der Maaten, L.; Weinberger, K. Q., Memory-efficient implementation of densenets. *arXiv preprint arXiv:1707.06990* **2017**.
48. Jacobsen, H., Localized-orbital locator (LOL) profiles of chemical bonding. *Canadian Journal of Chemistry* **2008,** *86* (7), 695-702.
49. Steinmann, S. N.; Mo, Y.; Corminboeuf, C., How do electron localization functions describe π-electron delocalization? *Physical Chemistry Chemical Physics* **2011,** *13* (46), 20584-20592.
50. Nkungli, N. K.; Ghogomu, J. N., Theoretical analysis of the binding of iron(III) protoporphyrin IX to 4-methoxyacetophenone thiosemicarbazone via DFT-D3, MEP, QTAIM, NCI, ELF, and LOL studies. *Journal of Molecular Modeling* **2017,** *23* (7), 200.
51. B, F. R.; Prasana, J. C.; Muthu, S.; Abraham, C. S., Molecular docking studies, charge transfer excitation and wave function analyses (ESP, ELF, LOL) on valacyclovir : A potential antiviral drug. *Computational Biology and Chemistry* **2019,** *78*, 9-17.
52. Melin, J.; Fuentealba, P., Application of the electron localization function to radical systems. *International Journal of Quantum Chemistry* **2003,** *92* (4), 381-390.
53. Hou, X.-J.; Gopakumar, G.; Lievens, P.; Nguyen, M. T., Chromium-Doped Germanium Clusters CrGen (n = 1−5): Geometry, Electronic Structure, and Topology of Chemical Bonding. *The Journal of Physical Chemistry A* **2007,** *111* (51), 13544-13553.

54. Lepetit, C.; Fau, P.; Fajerwerg, K.; Kahn, M. L.; Silvi, B., Topological analysis of the metal-metal bond: A tutorial review. *Coordination Chemistry Reviews* **2017,** *345*, 150-181.
55. Schweitzer, B.; Daniel, C.; Gourlaouen, C., Metal–metal bonding in 1st, 2nd and 3rd row transition metal complexes: a topological analysis. *Journal of Molecular Modeling* **2017,** *23* (5), 163.
56. Llusar, R.; Beltrán, A.; Andrés, J.; Fuster, F.; Silvi, B., Topological Analysis of Multiple Metal−Metal Bonds in Dimers of the M2(Formamidinate)4 Type with M = Nb, Mo, Tc, Ru, Rh, and Pd. *The Journal of Physical Chemistry A* **2001,** *105* (41), 9460-9466.
57. Kohout, M.; Wagner, F. R.; Grin, Y., Electron localization function for transition-metal compounds. *Theoretical Chemistry Accounts* **2002,** *108* (3), 150-156.
58. Matito, E.; Solà, M., The role of electronic delocalization in transition metal complexes from the electron localization function and the quantum theory of atoms in molecules viewpoints. *Coordination Chemistry Reviews* **2009,** *253* (5), 647-665.
59. Michelini, M. D. C.; Russo, N.; Alikhani, M. E.; Silvi, B., Energetic and topological analyses of the oxidation reaction between Mon (n = 1, 2) and N2O. *Journal of Computational Chemistry* **2005,** *26* (12), 1284-1293.
60. Jacobsen, H., Localized-orbital locator (LOL) profiles of transition-metal hydride and dihydrogen complexes. *Canadian Journal of Chemistry* **2009,** *87* (7), 965-973.