JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

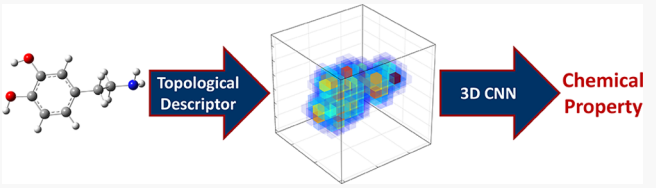
Three-Dimensional Convolutional Neural Networks Utilizing Molecular Topological Features for Accurate Atomization Energy Predictions

Ankur Kumar Gupta* and Krishnan Raghavachari*

Cite This: J. Chem. Theory Comput. 2022, 18, 2132−2143 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Deep learning methods provide a novel way to establish a correlation between two quantities. In this context, computer vision techniques such as three-dimensional (3D)-convolutional neural networks become a natural choice to associate a molecular property with its structure due to the inherent 3D 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 underfitted results. To address this deficiency, in this project, we propose to use quantum-chemically derived molecular topological features, namely, localized orbital locator and electron localization function, as molecular descriptors, which provide a relatively denser input representation in a 3D space. Such topological features provide a detailed picture of the atomic and electronic configuration and interatomic interactions in the molecule and hence are ideal for predicting properties that are highly dependent on the physical or electronic structure of the molecule. Herein, we demonstrate the efficacy of our proposed model by applying it to the task of predicting atomization energies for the QM9-G4MP2 data set, which contains ∼134k molecules. Furthermore, we incorporated the Δ-machine learning approach into our model, which enabled us to reach beyond benchmark accuracy levels (∼1.0 kJ mol^{-1}). As a result, we consistently obtain impressive mean absolute errors of the order 0.1 kcal mol^{-1} (∼0.42 kJ mol^{-1}) versus the G4(MP2) theory using relatively modest models, which could potentially be improved further in a systematic manner 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 two-dimensional (2D) image recognition tasks.[1] These newly developed artificial intelligence techniques have also profoundly impacted other branches of science, and chemistry is no exception. Thus, taking a cue from 2D image representations in computer vision theory, molecules, being intrinsically three-dimensional (3D) in nature, can be imagined as 3D images. Therefore, molecules can be analogously represented in the form of a 3D grid or a multidimensional tensor. However, unlike 2D images, where the input features (or descriptors) 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 grid 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[10−14] to the prediction of material properties[15,16] and NMR chemical shifts.[17] However, a major complication associated with 3D input representations is their high data structure sparsity aggravated due to the grid cell structure; therefore, in this article, we advocate the use of spatially *dense* descriptors, especially the ones based on the electron distribution in a molecule, thus providing an alternative to mitigate the data structure sparsity. Specifically, we propose to use what are known as electron localization functions (ELFs), viz., localized orbital locator (LOL)[18] and electron localization function (ELF),[19] which have found widespread use in elucidating bonding topology and electronic structure in a wide variety of molecules.[20−32]

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)[33] and Gaussian-4 (G4),[34] preclude their use on a routine basis. Therefore, considerable research efforts have been directed toward developing machine learning frameworks that could predict energies (or properties) at high levels of theory.[35−44] Additionally, a variety of noteworthy deep learning

Received: May 20, 2021
Published: February 28, 2022

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

architectures (viz., SchNet,[45] PhysNet,[46] DimeNet,[47] DeepMoleNet,[48] OrbNet,[49] TensorMol,[50] and ANI[51]) have been proposed, which were validated on the DFT level properties.[52,53] In this work, we aim to predict G4(MP2)[54−57] 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 quite a valuable quantity to reproduce.

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.[58] have achieved a highly impressive out-of-sample mean absolute error (MAE) of the order of 0.1 kcal mol^{-1} [vs the G4(MP2)[57] level of theory] on the QM9-G4MP2 data set[53,59] using the SchNet[45] and FCHL[60] models in conjunction with the Δ-machine learning (Δ-ML) approach.[61] 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,[58] thus enabling to reach chemical accuracy (±1.0 kcal mol^{-1}) consistently and also within striking distance of the elusive benchmark accuracy (±1.0 kJ mol^{-1}) with respect to a high level of theory (or the experimental value) through machine learning means. In this context, we note that DFT has emerged as a highly versatile and cost-effective approach to perform quantum computations and remains the go-to method for most chemical problems; however, its results could be *erratic*. Thus, it is imperative to discover new and novel methods to make DFT computations more accurate. Therefore, in light of this, we have also incorporated the Δ-ML scheme in our proposed machine learning protocol to attain chemical accuracy starting from a DFT baseline level of theory. This is likely to lead to a model that we expect will be highly useful for a variety of applications in quantum chemistry.

2. METHODS

2.1. Data. The QM9-G4MP2 data set 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.[53,59] The data set 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.[58] used 130,258 molecules from the QM9-G4MP2 data set, excluding the ones whose bond connectivity was found to be ambiguous. In their study, a random selection of 10% of molecules from the entire data set (13,026 molecules) was chosen as the test set to validate the working of their machine learning models, viz., SchNet[45] and FCHL.[58,60,62] To make a fair comparison with their results, we have also chosen the same training and test split. Therefore, all the results shown in Section 3 were obtained using the full training set (117,232 molecules), except that in Section 3.2, where we discuss the variation in model performance as a function of training set size.

2.1.1. Data Representation. The 3D 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, bond connectivity, hybridization, and so forth, 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, $N \times N \times N \times C$), 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 (CNNs)] 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 underfitted model due to the lack of enough information to learn from, in the input representation. Such performance degradation is caused due to sparse gradients being propagated through the network.

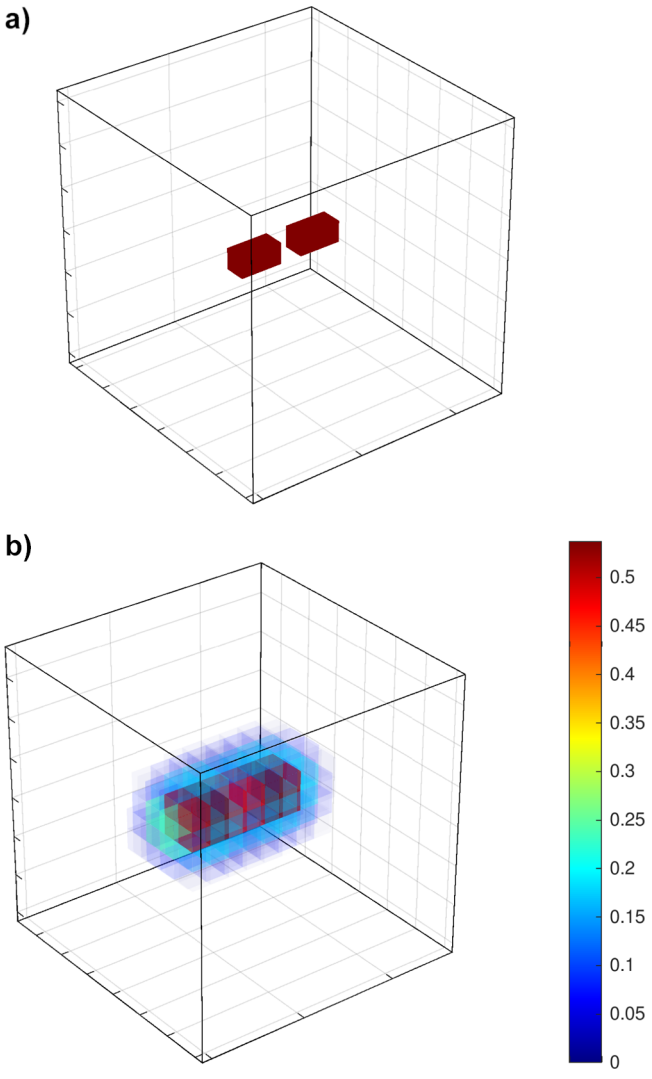


Figure 1. (a) Discrete voxelized representation of C_2N_2. The colored voxels represent the atomic positions. (b) Voxelized LOL profile of C_2N_2. Larger values represent electron-localized regions, while smaller values represent electron-diffuse regions.

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

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 interatomic interactions, while also providing a continuous feature representation.[63,64] Alternatively, 3D sparse data could be efficiently represented through an octree data structure, where only the nonsparse regions of a cubic volume are recursively partitioned into octants. Following this algorithm, a uniformly spaced voxelized data structure can be converted into one with numerically dense regions represented at fine resolutions and sparse spaces at low resolutions. The octree-based CNN[65,66] 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 3D spatial coordinates would be an obvious choice for such a descriptor. In this context, we note that molecular descriptors based on the electron probability distribution (or electronic structure) of a molecule provide a *less* sparse way of encoding its geometrical and electronic features into the spatial grid and hence forms 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 ELFs (or localization functions, for brevity), viz., LOL,[18] and ELF,[19] which are known to provide comprehensive topological information of a molecule. Localization functions have found wide applications in deciphering bonding and electronic structure of problematic systems such as radicals[20] and transition metal complexes.[24−32] In addition, they have also been used to better understand aromaticity[21−23] and electron density shifts during a chemical reaction.[67−69]

ELF and LOL, developed by Becke, 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 noninteracting kinetic energy density using the Taylor series expansion.[70,71] 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, it 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. In addition, because localization functions are based on pair probability density, they also help capture some degree of electron correlation in the input representation. For the sake of visual comparison, a discrete voxel-based representation of a simple molecule (C_2N_2) is shown in Figure 1a, and a voxelized LOL profile for the same molecule is shown in Figure 1b. 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 are not included in most curated data sets, potentially due to huge memory requirements. Therefore, to obtain the requisite descriptors, an additional electronic structure calculation on the full data set 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.[72] 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 (Section 3.4). Additionally, we have also analyzed the performance of nuclear electrostatic potential (NEP) (as a molecular descriptor) due to its dense nature and being independent of any electronic structure computation.

2.1.2. Data Preparation. A molecule can attain multiple orientations and positions in a 3D space; therefore, multiple grid representations for a molecule could be obtained upon translating or rotating it in 3D space. However, it is quite apparent that molecular properties are invariant to such transformations. Therefore, various techniques with varying degrees of rigor have been proposed to minimize or eliminate variance in CNNs with respect to spatial transformations. Due to the intricacy of the subject matter, we have dedicated a separate section (Section 3.6) on translational and rotational invariance (and equivariance), where we discuss these topics in more detail in the context of CNNs.

However, to analyze the effect of varying various hyperparameters on model performance, we used a standard orientation for each molecule so as to remove any ambiguity in the orientations between different molecules in the data set.[63] Such a unique orientation for every molecule was obtained using the principal component analysis algorithm, which provided a

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

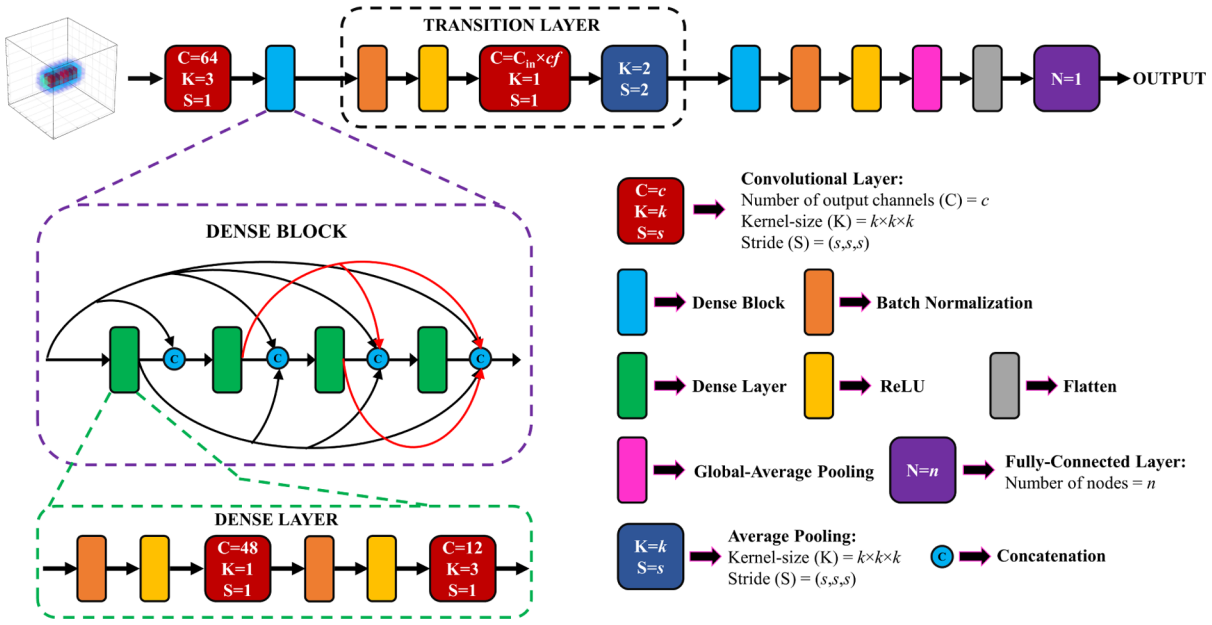


Figure 2. Schematic diagram of the DenseNet architecture. C_{in} represents the number of *incoming* channels from the first dense block. cf stands for compression factor and is taken to be 0.5 in all cases.

new set of molecular coordinates at which the variance in the (heavy atom) x-coordinates is maximum. In simple terms, the molecule is oriented along the first principal component (or the x-axis in this case). The reoriented molecules were then used to obtain the dimensions of an imaginary cubic box, large enough to encompass all the heavy atoms of any molecule in the data set. The dimension of such a cube turns out to be 10.4 Å for the QM9 data set. Although, as discussed in more detail in Section 2.2.1, because our network architecture is a *pure* CNN, we could choose a custom box dimension for every molecule depending on its spatial span. Nevertheless, for the sake of simplicity, we chose a constant box size to construct the input representation for every molecule in the data set.

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 shape and size of the four-dimensional (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 CNN roughly scales as the cubic power of the grid length. Therefore, the grid length (or grid resolution) parameter should be carefully chosen, taking the available computational resources into consideration. We used a grid length of 14 (grid resolution = 0.743 Å) to construct the input tensors, which were used to obtain the majority of results discussed in Section 3; however, we also experimented with multiple grid lengths (or equivalently, grid resolutions) to ascertain their correlation with the model performance (Section 3.1).

The requisite molecular descriptors (viz., LOL, ELF, and NEP) were obtained using the Multiwfn program[73] that takes the wavefunction data from the Gaussian 16[74] generated *wfx* files as the input and computes the value of a desired real space function at each of the Cartesian coordinates of a user-defined 3D grid. The generated data can then be converted into a 4D tensor ($N \times N \times N \times 1$), suitable to be used as an input for a 3D CNN. All the data preparation scripts are available in the paper's GitHub repository (https://github.com/ankur56/ELFNet).

2.2. Model. *2.2.1. Architecture.* The input data structure usually dictates the architecture type of the neural network. Therefore, with the input data structure defined as a 3D grid, a CNN becomes the concomitant architecture. Among the host of CNN architectures available in the literature for image learning tasks, we chose to employ the DenseNet[75] architecture chiefly for its high parameter efficiency and ease of training. Most computer vision architectures, including DenseNets, were developed for 2D image recognition tasks where the input shape is a 3D tensor. Thus, for the task of learning the molecular topology, 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 2. DenseNet introduces what is known as a dense block into a CNN architecture, which is composed of the so-called dense layers (not to be confused with a fully connected layer, which is also called a dense layer quite commonly), which in turn is a stack of $1 \times 1 \times 1$ and $3 \times 3 \times 3$ convolutional layers, reminiscent of the bottleneck-block in ResNets.[76] However, 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. Furthermore, because 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 the DenseNets parameter efficient by design and, hence, less susceptible to overfitting. Although quite simple in concept, the densely connected topology of DenseNets also makes it robust to the vanishing gradient problem and boosts

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

information and gradient flow. For the sake of simplicity, only four dense layers are included in the dense block shown in Figure 2.

Any two adjacent dense blocks in the DenseNet architecture 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), whose domain is $0 < cf \leq 1$, 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 the model training cost and accuracy.[75] All the architectural hyperparameters except the number of dense layers in each of the dense blocks are depicted in Figure 2. In this work, we have experimented with the number of dense layers in the two dense blocks (Section 3.2), which determines the overall depth of the architecture, and is one of the primary factors dictating the model's overall performance and training 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.

The feature maps obtained from the second dense block are then passed through a global average pooling layer that outputs the spatial average of each of the feature maps. Finally, the resultant 4D tensor (of shape, say, $1 \times 1 \times 1 \times C'$) is flattened into a vector (of size C') and then passed through a single node fully connected layer with linear activation to obtain the requisite output (or Δ-atomization energy in this case). Such an architectural design ensures that the network remains independent of the initial input size and is therefore sometimes referred to as a purely convolutional network. Convolutional layers operate on input through a small kernel whose size is independent of the given input size (or shape); hence, convolutional layers are input size-agnostic by design. However, global average pooling is the key layer that makes the given network input size independent, as it enables providing a constant-sized vector irrespective of the starting input size. Therefore, input tensor size need not be the same for every molecule in the data set and could be chosen to be different from each other; however, most applications involving CNNs use only a fixed-sized input for simplicity. The input agnostic nature of the given DenseNet architecture enables us to make predictions for spatially extensive systems using a model trained only on relatively compact systems (or molecules). In addition, we could also build a customized tensor representation for every molecule based on its physical range that could be incorporated in the data processing pipeline to reduce grid data sparsity. However, to make a model robust to variable-sized input, it may also be necessary to include inputs of variable size in the training set, and hence, a systematic analysis is required and is being pursued in our group. A model summary of a (16, 16)-DenseNet architecture is available in Table S1 of the Supporting Information, which shows the output shape generated after every layer for a fixed input size and depicts how inputs are propagated through the network.

2.2.2. Training. The entire machine learning workflow was implemented in PyTorch-Lightning.[77,78] The 3D-DenseNet code was adapted from the publicly available memory-efficient version of DenseNet implemented in PyTorch by Pleiss et al.[79] Each of the models was trained in parallel on four NVIDIA V100 GPUs with a cumulative batch size of 128. The MAE is chosen as the loss function for model training. The model parameters were optimized using the stochastic gradient descent algorithm in conjunction with Nesterov momentum (0.9) using a weight decay parameter (L2 penalty) of 1.0×10^{-4}. 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. During the optimization procedure, the learning rate is controlled through a learning rate schedule (ReduceLROnPlateau) that decreases the learning rate by a factor of 0.75 whenever the training loss plateaus within a certain user-provided threshold (0.005 kcal mol^{-1}). 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 the B3LYP/6-31G(2df,p) level are known. The predicted values for an out-of-sample data set by the trained 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 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/architecture and data-related hyperparameters; therefore, the effect of varying a few seemingly important hyperparameters is reported in this section, thus gleaning insights into different ways that systematically improve the 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 training a convolutional neural network formally scales as the cubic power of the grid length (Figure S1 in the Supporting Information). 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, 0.743, and 0.650 Å, respectively. These input representations are then used to train the base DenseNet architecture (Figure 2) with a fixed dense block configuration of (16, 16). The obtained metrics summarized in Figure 3 clearly

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

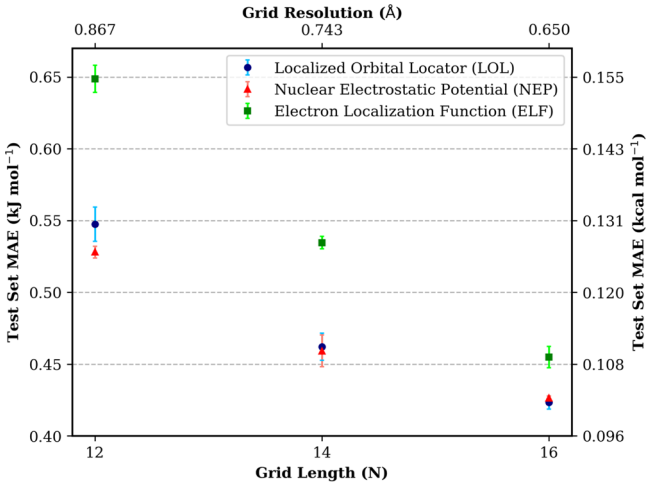


Figure 3. Effect of varying grid length (N) on the MAE of the test set. All results were obtained using the base DenseNet architecture (Figure 2) with a dense block configuration of (16, 16).

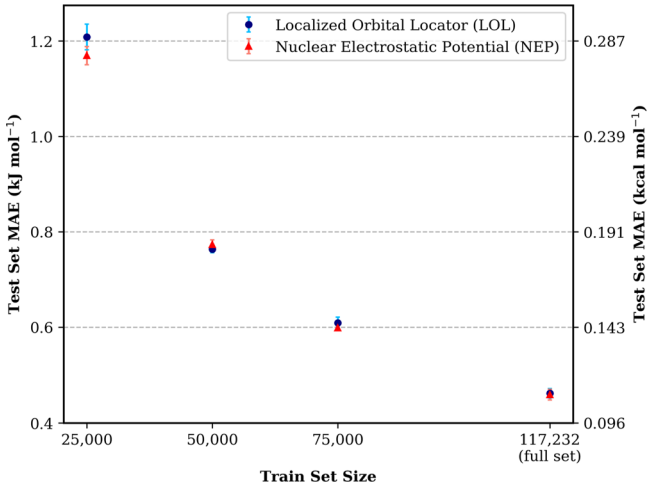


Figure 4. Effect of varying the training set size on the MAE of the test set. All results were obtained using the base DenseNet architecture (Figure 2) with a dense block configuration of (16, 16).

show an improvement in model performance with an increase in grid length. Furthermore, 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. Interestingly, among the two localization functions tested, LOL provides superior results than ELF in all cases. In addition, the performance of 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 MAE (0.423 kJ mol^{-1} or 0.101 kcal mol^{-1} or 4.38 meV) comparable to the best result obtained by Ward et al.[58] (0.434 kJ mol^{-1} or 0.104 kcal mol^{-1} or 4.50 meV).

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. Therefore, we experimented with multiple training set sizes to decipher the extent of correlation between the amount of data and model accuracy, 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 (N) 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 MAE of the test set as a function of the training set size is depicted in Figure 4. 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} (or 0.287 kcal mol^{-1}), which could be useful in situations with limited compute availability. Interestingly, both LOL and NEP provide similar quantitative results with respect to change in the training set size, indicating further similarity between the efficacies of the two descriptors.

3.3. Effect of Varying the Dense Block Configuration. The depth of a dense block refers to the number of dense layers (or convolutional layers) it is composed of. The depth of the *first* dense block (d_1) is a hyperparameter of critical importance because it is one of the primary determining factors of the training 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 training 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, and 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 5a).

Furthermore, we also experimented varying d_1 (*viz.*, 8, 16, 24, 32) while keeping d_2 constant (at 8) (Figure 5b). Finally, we also show results obtained by simultaneously doubling the number of layers in each of the dense blocks (Figure 5c). In both of these cases, the overall depth of the network is being increased, which causes a lowering of the 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, that is, a dense block configuration of (32, 32), which provides an MAE of 0.393 ± 0.008 kJ mol^{-1} (or 0.094 kcal mol^{-1} or 4.08 meV); 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 were obtained using the LOL descriptor; however, the general trend is expected to be the same for other topological descriptors as well. The training times for an epoch for different models are shown in Figures S2 and S3 of the Supporting Information, 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

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

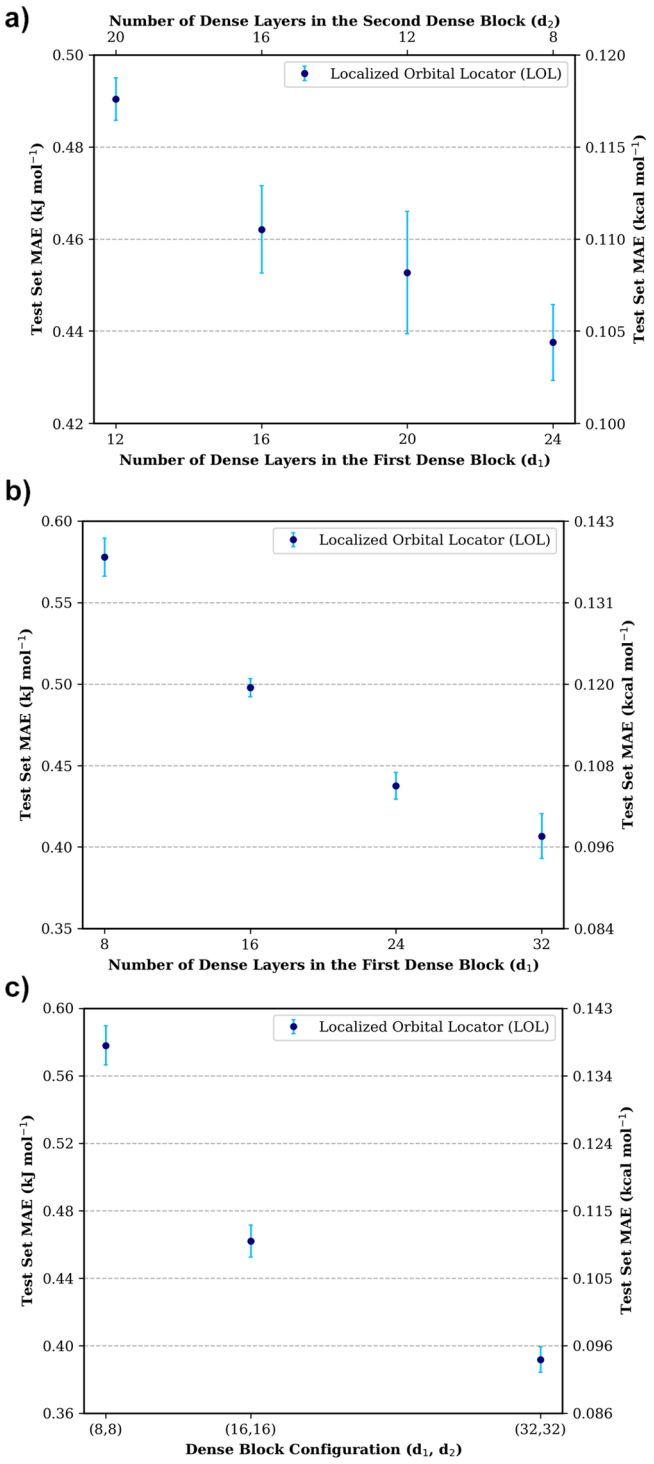


Figure 5. (a) Effect of varying the depth of the first dense block on the test set MAE while keeping the architecture's total depth constant. All results were obtained using the base DenseNet architecture (Figure 2) with the total number of dense layers fixed at 32. (b) Effect of varying the depth of the first dense block on the test set MAE. All results were obtained using the base DenseNet architecture (Figure 2) with the number of dense layers of the second dense block fixed at 8. (c) Effect of doubling the dense block configuration on the test set MAE. All results were obtained using the base DenseNet architecture (Figure 2).

increasing the depth of the architecture, which, however, is accompanied by an increase in the training cost.

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 6, it is apparent that the results

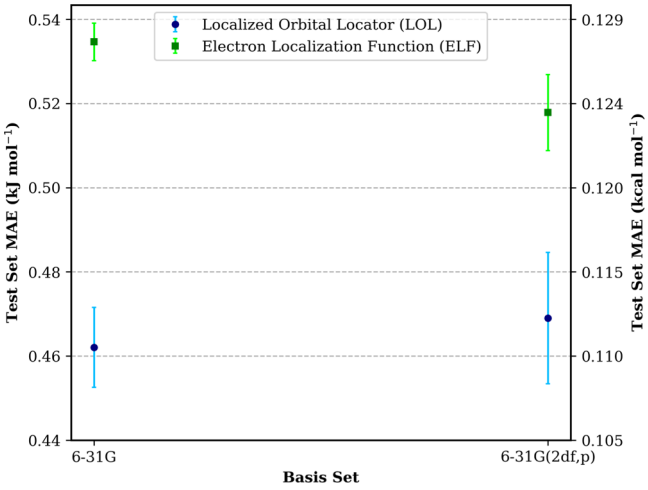


Figure 6. Effect of varying the basis set used to generate the localization functions on model performance. All results were obtained using the base DenseNet architecture (Figure 2) with a dense block configuration of (16, 16).

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.

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 such as 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 also tested the model performance for multichannel inputs, which are 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 2) with a (16, 16) dense block configuration. As depicted in Figure 7, 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. Such an observation can also be attributed to the test loss saturation with respect to the network architecture and could mean that a deeper or wider

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

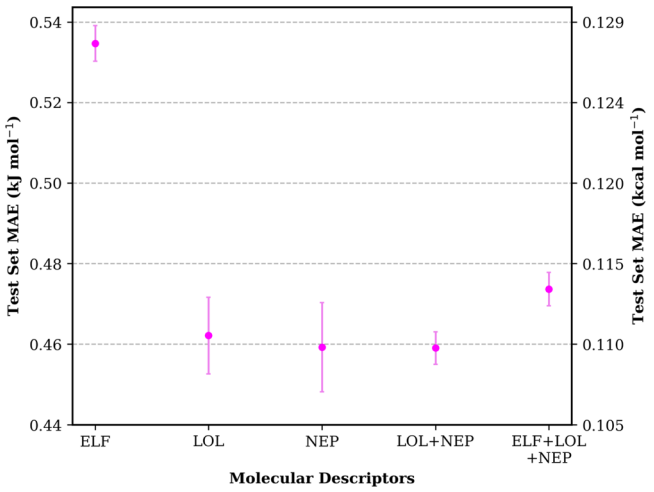


Figure 7. Performance comparison between different molecular descriptor combinations. All results were obtained using the base DenseNet architecture (Figure 2) with a dense block configuration of (16, 16).

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 for a molecule. In fact, for a set of n distinct molecular descriptors, a total of $(2^n - 1)$ input combinations could be obtained, which makes 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.

3.6. Effect of Incorporating Transformation Invariance. The learning capacity of an ML model could be greatly enhanced by imbuing it with invariances in the data.[80] Furthermore, such transformation invariances could be embedded in the network architecture itself, thus significantly improving their data and compute efficiency. In this section, we discuss translational and rotational invariances in the context of CNNs, as they are the most prevalent invariant transformations in chemistry data sets; however, research on other affine transformation invariances (viz., reflection, scaling, and shear) is also being actively pursued in the computer vision community.[81]

In a convolutional layer, only one set of kernel parameters is learned for every location in the grid. Such a parameter-sharing mechanism imparts *translational* equivariance to the convolutional layer (and, by extension, to CNNs).[82] In addition, the pooling layers help make CNNs approximately invariant to small translations of the input object.[82] However, commonly used downsampling or subsampling techniques such as strided convolutional and pooling layers may destroy translational equivariance and invariance in a CNN.[83−86] For an extended discussion on translational equivariance and invariance in CNNs, please refer to sections 9.2 and 9.3 of ref 82. However, standard CNNs are neither equivariant nor invariant to *rotational* transformations of the input object. Therefore, many research activities have been focused on explicitly encoding rotational equivariance (and invariance) directly into CNN architectures, which has led to the development of a range of networks, viz., harmonic networks,[87] 3D steerable CNNs,[88] CFNets,[89] tensor field networks,[90] and so forth,[91−99] though none of them has dominated standard practice. Although such enhanced CNNs provide a mathematically rigorous ML framework to account for invariances with respect to different input transformations, they also add another layer of complexity into the network architecture and thus may require substantial changes or even a complete overhaul to the base architecture and the training protocol. Therefore, a careful study is needed to benchmark different models available on various parameters such as model performance, time complexity, ease of implementation, and so forth. To that end, we are actively working toward exploring different rotationally equivariant models that could integrate well with DenseNets.

Another popular way to incorporate rotation invariance into a CNN-based model is through data augmentation, which involves generating spatially transformed copies of the input grid representation of every sample in the data set. Therefore, augmenting the data set increases the number of training samples, which in turn increases the computational cost of training the network. However, an overwhelming advantage of using data augmentation is that it does not require any modification to the architecture and training protocol and hence is relatively simple to implement. Although not mathematically rigorous and exact, data augmentation is a simple and effective means to make a model (partially) impervious to spatial transformations and has been widely used in the CNN literature. Therefore, in this work, we have used data augmentation to assess the effect of incorporating rotational invariance in our ML model.

Because a cube has 24 rotational symmetries (cube group), we augmented the data set by transforming it through the set of all right-angle rotations, as prescribed in some of the previous works utilizing 3D CNNs.[6,7,100] In other words, the cubic grid representation of the standard orientation (discussed in Section 2.1.2) of each molecule in the data set was transformed through all possible combinations of 90° rotations, which yielded 24 different orientations. As a result, both the training and test data set increased by a factor of 24. We used the augmented data set [grid length (N) = 14] for training a (16, 16)-DenseNet model using a cumulative batch size of 1024 across 16 GPUs while keeping the rest of the training hyperparameters the same as described in Section 2.2.2. The training plots for the unaugmented and data-augmented models are shown in Figure S4. As shown in Figure 8, the results obtained using the data-augmented model are a definite improvement over those from the unaugmented version and demonstrate the usefulness of data augmentation in reducing overfitting, improving generalizability, and increasing model robustness. In addition, the data-augmented model is less susceptible to errors associated with spatial transformations of the molecule, as shown in Figures S5 and S6 of the Supporting Information. Although the results in the previous sections (Sections 3.1−3.5) were obtained using only a single standard representative molecular orientation, the qualitative trends should behave similarly for the data-augmented model.

3.7. Model Performance on Heavier and Larger Molecules. To quantify the transferability of the proposed DenseNet model, we assessed its performance on molecules heavier and larger than that in the QM9 data set. To this end, we borrowed the G4MP2-heavy data set from ref 58, a collection of 66 bio-oil-derived molecules containing 10−14 heavy atoms (C, N, O, and F). We performed energy inferences on the G4MP2-heavy data set using the best performing DenseNet model (trained on 117,232 molecules from the G4MP2-QM9 data set) from Section 3.1, that is, (16, 16)-DenseNet (descriptor: LOL)

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

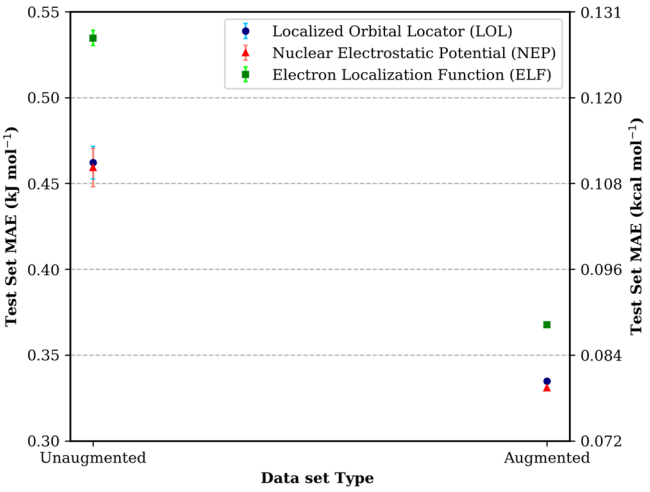


Figure 8. Performance comparison between data-augmented and unaugmented models [grid length (N) = 14]. All results were obtained using the base DenseNet architecture (Figure 2) with a dense block configuration of (16, 16). Data-augmented model results were obtained using only one training run.

with grid length (N) = 16. The corresponding MAE obtained from the DenseNet model is compared with that from the FCHL-Δ and SchNet-Δ models and are summarized in Table 1.

Table 1. Comparison of MAEs [with Respect to G4(MP2) Atomization Energies] on the G4MP2-Heavy Data Set between DenseNet, FCHL-Δ, and SchNet-Δ Models

machine learning model	MAE on the G4MP2-heavy data set (kcal mol^{-1})
FCHL-Δ	0.29
SchNet-Δ	0.91
(16, 16)-DenseNet [grid length (N) = 16; descriptor: LOL]	0.36 (0.21)[a]

[a]The value in parenthesis in the second column corresponding to the DenseNet model represents the associated mean absolute deviation.

As shown in Table 1, the (16, 16)-DenseNet [grid length (N) = 16; descriptor: LOL] model outperforms SchNet-Δ by a respectable margin. Interestingly, the MAE obtained from the DenseNet model is more comparable to that from the FCHL-Δ model. However, unlike FCHL-Δ, the inference time for DenseNets scales more favorably with the training data set size. Furthermore, from the trends discussed in this article, we expect the transferability to improve (or the MAEs to reduce) even further upon training the DenseNet architecture on higher resolution and memory-efficient data structures.

4. CONCLUSIONS

The present article highlights the importance of using dense 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 ELFs and correlated them with the Δ-atomization energies. Moreover, the network could also learn the structural features through the NEP. 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, viz., grid length (or grid resolution) and network depth. 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 previously.[32,101−103] 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 computation. 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, data sets 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.[20,31] Additionally, localization functions have been widely used to characterize the bonding topology in transition metal complexes and thus could be an indispensable tool for their tensorial representation.[24−31]

The encouraging results in the present article show incredible promise for future endeavors to tackle even more challenging problems. The proposed ML model in this work, based on the DenseNet architecture, is a crucial stepping-stone toward our goal to build even more efficient and physically motivated models utilizing 3D descriptors. The existing framework 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 CNN 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. Furthermore, our immediate research efforts would focus on engineering a rotationally equivariant DenseNet architecture. Finally, the future directions concerning these deep learning frameworks would be directed toward solving problems of practical interest, such as predicting ligand−receptor binding affinities, thus providing complementary ways to enhance high-throughput virtual screening tasks.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.1c00504.

All codes needed to reproduce this study can be found at https://github.com/ankur56/ELFNet; change in elapsed time for an epoch with change in grid length, depth, and doubling of the dense block configuration; model summary of a (16, 16)-DenseNet architecture; training plots; and energy inferences for a single molecule (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Ankur Kumar Gupta − *Department of Chemistry, Indiana University, Bloomington, Indiana 47405, United States;*

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

orcid.org/0000-0002-3128-9535; Email: ankkgupt@indiana.edu

Krishnan Raghavachari − *Department of Chemistry, Indiana University, Bloomington, Indiana 47405, United States;* orcid.org/0000-0003-3275-1426; Email: kraghava@indiana.edu

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.1c00504

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We acknowledge support from the National Science Foundation grant CHE-2102583 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.

■ REFERENCES

(1) https://paperswithcode.com/sota/image-classification-on-imagenet (accessed 2021).
(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*, 3036−3042.
(3) Hochuli, J.; Helbling, A.; Skaist, T.; Ragoza, M.; Koes, D. R. Visualizing convolutional neural network protein-ligand scoring. *J. Mol. Graphics Modell.* **2018**, *84*, 96−108.
(4) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein−Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57*, 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. *Commun. Chem.* **2020**, *3*, 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. *J. Chem. Inf. Model.* **2020**, *60*, 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. *J. Chem. Inf. Model.* **2018**, *58*, 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. *J. Mol. Graphics Modell.* **2021**, *105*, 107865.
(9) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. **2015**, arXiv:1510.02855. arXiv preprint.
(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 Comput. Biol.* **2019**, *15*, No. e1006718.
(11) Torng, W.; Altman, R. B. High precision protein functional site detection using 3D convolutional neural networks. *Bioinformatics* **2018**, *35*, 1503−1512.
(12) Simonovsky, M.; Meyers, J. DeeplyTough: Learning Structural Comparison of Protein Binding Sites. *J. Chem. Inf. Model.* **2020**, *60*, 2356−2366.
(13) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Improving detection of protein-ligand binding sites with 3D segmentation. *Sci. Rep.* **2020**, *10*, 5035.
(14) de Jesus, D. R.; Cuevas, J.; Rivera, W.; Crivelli, S. Capsule networks for protein structure classification and prediction. **2018**, arXiv:1808.07475. arXiv preprint.
(15) Homer, E. R.; Hensley, D. M.; Rosenbrock, C. W.; Nguyen, A. H.; Hart, G. L. W. Machine-Learning Informed Representations for Grain Boundary Structures. *Front. Mater.* **2019**, *6*, 168.
(16) Kajita, S.; Ohba, N.; Jinnouchi, R.; Asahi, R. A Universal 3D Voxel Descriptor for Solid-State Material Informatics with Deep Convolutional Neural Networks. *Sci. Rep.* **2017**, *7*, 16991.
(17) Liu, S.; Li, J.; Bennett, K. C.; Ganoe, B.; Stauch, T.; Head-Gordon, M.; Hexemer, A.; Ushizima, D.; Head-Gordon, T. Multi-resolution 3D-DenseNet for Chemical Shift Prediction in NMR Crystallography. *J. Phys. Chem. Lett.* **2019**, *10*, 4558−4565.
(18) Schmider, H. L.; Becke, A. D. Chemical content of the kinetic energy density. *J. Mol. Struct.: THEOCHEM* **2000**, *527*, 51−61.
(19) Becke, A. D.; Edgecombe, K. E. A simple measure of electron localization in atomic and molecular systems. *J. Chem. Phys.* **1990**, *92*, 5397−5403.
(20) Melin, J.; Fuentealba, P. Application of the electron localization function to radical systems. *Int. J. Quantum Chem.* **2003**, *92*, 381−390.
(21) Santos, J. C.; Andres, J.; Aizman, A.; Fuentealba, P. An Aromaticity Scale Based on the Topological Analysis of the Electron Localization Function Including σ and π Contributions. *J. Chem. Theory Comput.* **2005**, *1*, 83−86.
(22) Poater, J.; Duran, M.; Solà, M.; Silvi, B. Theoretical Evaluation of Electron Delocalization in Aromatic Molecules by Means of Atoms in Molecules (AIM) and Electron Localization Function (ELF) Topological Approaches. *Chem. Rev.* **2005**, *105*, 3911−3947.
(23) Santos, J. C.; Tiznado, W.; Contreras, R.; Fuentealba, P. Sigma−pi separation of the electron localization function and aromaticity. *J. Chem. Phys.* **2004**, *120*, 1670−1673.
(24) Lepetit, C.; Fau, P.; Fajerwerg, K.; Kahn, M. L.; Silvi, B. Topological analysis of the metal-metal bond: A tutorial review. *Coord. Chem. Rev.* **2017**, *345*, 150−181.
(25) Schweitzer, B.; Daniel, C.; Gourlaouen, C. Metal−metal bonding in 1st, 2nd and 3rd row transition metal complexes: a topological analysis. *J. Mol. Model.* **2017**, *23*, 163.
(26) 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. *J. Phys. Chem. A* **2001**, *105*, 9460−9466.
(27) Kohout, M.; Wagner, F. R.; Grin, Y. Electron localization function for transition-metal compounds. *Theor. Chem. Acc.* **2002**, *108*, 150−156.
(28) 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. *Coord. Chem. Rev.* **2009**, *253*, 647−665.
(29) 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. *J. Comput. Chem.* **2005**, *26*, 1284−1293.
(30) Jacobsen, H. Localized-orbital locator (LOL) profiles of transition-metal hydride and dihydrogen complexes. *Can. J. Chem.* **2009**, *87*, 965−973.
(31) 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. *J. Phys. Chem. A* **2007**, *111*, 13544−13553.
(32) 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. *J. Mol. Model.* **2017**, *23*, 200.
(33) Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. A fifth-order perturbation comparison of electron correlation theories. *Chem. Phys. Lett.* **1989**, *157*, 479−483.
(34) Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory. *J. Chem. Phys.* **2007**, *126*, 084108.
(35) Käser, S.; Boittier, E. D.; Upadhyay, M.; Meuwly, M. Transfer Learning to CCSD(T): Accurate Anharmonic Frequencies from Machine Learning Models. *J. Chem. Theory Comput.* **2021**, *17*, 3687−3699.

(36) Smith, J. S.; Nebgen, B. T.; Zubatyuk, R.; Lubbers, N.; Devereux, C.; Barros, K.; Tretiak, S.; Isayev, O.; Roitberg, A. E. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **2019**, *10*, 2903.
(37) Käser, S.; Boittier, E.; Upadhyay, M.; Meuwly, M. MP2 is not good enough: Transfer learning ML models for accurate VPT2 frequencies. **2021**, arXiv:2103.05491. arXiv preprint.
(38) Zheng, P.; Zubatyuk, R.; Wu, W.; Isayev, O.; Dral, P. O. Artificial Intelligence-Enhanced Quantum Chemical Method with Broad Applicability. *Nat. Commun.* **2021**, *12*, 7022.
(39) Nandi, A.; Qu, C.; Houston, P. L.; Conte, R.; Bowman, J. M. Δ-machine learning for potential energy surfaces: A PIP approach to bring a DFT-based PES to CCSD(T) level of theory. *J. Chem. Phys.* **2021**, *154*, 051102.
(40) Chen, Y.; Zhang, L.; Wang, H.; E, W. Ground State Energy Functional with Hartree−Fock Efficiency and Chemical Accuracy. *J. Phys. Chem. A* **2020**, *124*, 7155−7165.
(41) Peyton, B. G.; Briggs, C.; D'Cunha, R.; Margraf, J. T.; Crawford, T. D. Machine-Learning Coupled Cluster Properties through a Density Tensor Representation. *J. Phys. Chem. A* **2020**, *124*, 4861−4871.
(42) Schran, C.; Behler, J.; Marx, D. Automated Fitting of Neural Network Potentials at Coupled Cluster Accuracy: Protonated Water Clusters as Testing Ground. *J. Chem. Theory Comput.* **2020**, *16*, 88−99.
(43) Margraf, J. T.; Reuter, K. Pure non-local machine-learned density functional theory for electron correlation. *Nat. Commun.* **2021**, *12*, 344.
(44) Bogojeski, M.; Vogt-Maranto, L.; Tuckerman, M. E.; Müller, K.-R.; Burke, K. Quantum chemical accuracy from density functional approximations via machine learning. *Nat. Commun.* **2020**, *11*, 5223.
(45) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet—A deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722.
(46) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678−3693.
(47) Klicpera, J.; Groß, J.; Günnemann, S. Directional message passing for molecular graphs. **2020**, arXiv:2003.03123. arXiv preprint.
(48) 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. *J. Chem. Inf. Model.* **2021**, *61*, 1066−1082.
(49) Qiao, Z.; Welborn, M.; Anandkumar, A.; Manby, F. R.; Miller, T. F. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **2020**, *153*, 124111.
(50) Yao, K.; Herr, J. E.; Toth, D. W.; McKintyre, R.; Parkhill, J. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chem. Sci.* **2018**, *9*, 2261−2269.
(51) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, *8*, 3192−3203.
(52) 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. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(53) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.
(54) Chan, B.; Deng, J.; Radom, L. G4(MP2)-6X: A Cost-Effective Improvement to G4(MP2). *J. Chem. Theory Comput.* **2011**, *7*, 112−120.
(55) Chan, B.; Coote, M. L.; Radom, L. G4-SP, G4(MP2)-SP, G4-sc, and G4(MP2)-sc: Modifications to G4 and G4(MP2) for the Treatment of Medium-Sized Radicals. *J. Chem. Theory Comput.* **2010**, *6*, 2647−2653.
(56) Chan, B.; Karton, A.; Raghavachari, K.; Radom, L. Restricted-Open-Shell G4(MP2)-Type Procedures. *J. Phys. Chem. A* **2016**, *120*, 9299−9304.
(57) Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **2007**, *127*, 124105.
(58) 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 Commun.* **2019**, *9*, 891−899.
(59) Narayanan, B.; Redfern, P. C.; Assary, R. S.; Curtiss, L. A. Accurate quantum chemical energies for 133000 organic molecules. *Chem. Sci.* **2019**, *10*, 7449−7455.
(60) Faber, F. A.; Christensen, A. S.; Huang, B.; von Lilienfeld, O. A. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.* **2018**, *148*, 241717.
(61) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087−2096.
(62) Christensen, A. S.; Bratholm, L. A.; Faber, F. A.; Anatole von Lilienfeld, O. FCHL revisited: Faster and more accurate quantum machine learning. *J. Chem. Phys.* **2020**, *152*, 044107.
(63) 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. *Mol. Pharm.* **2018**, *15*, 4378−4385.
(64) Torng, W.; Altman, R. B. 3D deep convolutional neural networks for amino acid environment similarity analysis. *BMC Bioinf.* **2017**, *18*, 302.
(65) 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 Trans. Graph.* **2017**, *36*, 1−11.
(66) Riegler, G.; Osman Ulusoy, A.; Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017; pp 3577−3586.
(67) Andrés, J.; Berski, S.; Silvi, B. Curly arrows meet electron density transfers in chemical reaction mechanisms: from electron localization function (ELF) analysis to valence-shell electron-pair repulsion (VSEPR) inspired interpretation. *Chem. Commun.* **2016**, *52*, 8183−8195.
(68) Krokidis, X.; Noury, S.; Silvi, B. Characterization of Elementary Chemical Processes by Catastrophe Theory. *J. Phys. Chem. A* **1997**, *101*, 7277−7282.
(69) Andres, J.; Berski, S.; Domingo, L. R.; Polo, V.; Silvi, B. Describing the Molecular Mechanism of Organic Reactions by Using Topological Analysis of Electronic Localization Function. *Curr. Org. Chem.* **2011**, *15*, 3566−3575.
(70) Becke, A. D. Hartree−Fock exchange energy of an inhomogeneous electron gas. *Int. J. Quantum Chem.* **1983**, *23*, 1915−1922.
(71) Becke, A. D. Local exchange-correlation approximations and first-row molecular dissociation energies. *Int. J. Quantum Chem.* **1985**, *27*, 585−594.
(72) 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.
(73) Lu, T.; Chen, F. Multiwfn: A multifunctional wavefunction analyzer. *J. Comput. Chem.* **2012**, *33*, 580−592.
(74) 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, J. A., Jr.; 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; Gaussian, Inc.: Wallingford, CT, 2016.

https://doi.org/10.1021/acs.jctc.1c00504
J. Chem. Theory Comput. 2022, 18, 2132−2143

(75) Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017; pp 4700−4708.
(76) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016; pp 770−778.
(77) Falcon, W. A. A. *PyTorch Lightning*; GitHub. https://github.com/PyTorchLightning/pytorch-lightning, 2019; Vol. 3.
(78) 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. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026−8037.
(79) Pleiss, G.; Chen, D.; Huang, G.; Li, T.; van der Maaten, L.; Weinberger, K. Q. Memory-efficient implementation of densenets. **2017**, arXiv:1707.06990. arXiv preprint.
(80) LeCun, Y. Generalization and network design strategies. *Connectionism Perspect.* **1989**, *19*, 143−155.
(81) Jansson, Y.; Maydanskiy, M.; Finnveden, L.; Lindeberg, T. Inability of spatial transformations of CNN feature maps to support invariant recognition. **2020**, arXiv:2004.14716. arXiv preprint.
(82) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.
(83) Azulay, A.; Weiss, Y. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint* arXiv:1805.12177, **2018** (accessed 2022-02-24).
(84) Kauderer-Abrams, E. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint* arXiv:1801.01450, **2017** (accessed 2022-02-24).
(85) Mouton, C.; Myburgh, J. C.; Davel, M. H. Stride and translation invariance in CNNs. In *Proceedings of the Southern African Conference for Artificial Intelligence Research*; Springer, 2021, pp 267−281.
(86) Zhang, R. Making convolutional networks shift-invariant again. In *Proceedings of the International Conference on Machine Learning (PMLR)*; Vol 97, 2019, pp 7324−7334.
(87) Worrall, D. E.; Garbin, S. J.; Turmukhambetov, D.; Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017; pp 5028−5037.
(88) Weiler, M.; Geiger, M.; Welling, M.; Boomsma, W.; Cohen, T. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *NeurIPS*, 2018.
(89) Chidester, B.; Zhou, T.; Do, M. N.; Ma, J. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics* **2019**, *35*, i530−i537.
(90) Thomas, N.; Smidt, T.; Kearnes, S.; Yang, L.; Li, L.; Kohlhoff, K.; Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. **2018**, arXiv:1802.08219. arXiv preprint.
(91) Worrall, D.; Brostow, G. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018; pp 567−584.
(92) Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2017−2025.
(93) Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017; pp 764−773.
(94) Dieleman, S.; De Fauw, J.; Kavukcuoglu, K. Exploiting cyclic symmetry in convolutional neural networks. In *International Conference on Machine Learning, PMLR*, 2016; pp 1889−1898.
(95) Marcos, D.; Volpi, M.; Komodakis, N.; Tuia, D. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017; pp 5048−5057.
(96) Sabour, S.; Frosst, N.; Hinton, G. E. Dynamic routing between capsules. **2017**, arXiv:1710.09829. arXiv preprint.
(97) Cohen, T.; Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning, PMLR*, 2016; pp 2990−2999.
(98) Henriques, J. F.; Vedaldi, A. Warped convolutions: Efficient invariance to spatial transformations. In *International Conference on Machine Learning, PMLR*, 2017; pp 1461−1469.
(99) Della Libera, L.; Golkov, V.; Zhu, Y.; Mielke, A.; Cremers, D. Deep Learning for 2D and 3D Rotatable Data: An Overview of Methods. **2019**, arXiv:1910.14594, arXiv preprint.
(100) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. *Bioinformatics* **2018**, *34*, 3666−3674.
(101) Jacobsen, H. Localized-orbital locator (LOL) profiles of chemical bonding. *Can. J. Chem.* **2008**, *86*, 695−702.
(102) Steinmann, S. N.; Mo, Y.; Corminboeuf, C. How do electron localization functions describe π-electron delocalization? *Phys. Chem. Chem. Phys.* **2011**, *13*, 20584−20592.
(103) Rizwana B, F.; 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. *Comput. Biol. Chem.* **2019**, *78*, 9−17.