Molecular machine learning with conformer ensembles

Simon Axelrod\textsuperscript{1,2} and Rafael Gómez-Bombarelli\textsuperscript{2,*}

\textsuperscript{1} Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138, United States of America
\textsuperscript{2} Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America

\textsuperscript{*} Author to whom any correspondence should be addressed.

E-mail: rafagb@mit.edu

Keywords: conformers, machine learning, drug discovery, computational chemistry

Supplementary material for this article is available online

Abstract

Virtual screening can accelerate drug discovery by identifying promising candidates for experimental evaluation. Machine learning is a powerful method for screening, as it can learn complex structure–property relationships from experimental data and make rapid predictions over virtual libraries. Molecules inherently exist as a three-dimensional ensemble and their biological action typically occurs through supramolecular recognition. However, most deep learning approaches to molecular property prediction use a 2D graph representation as input, and in some cases a single 3D conformation. Here we investigate how the 3D information of multiple conformers, traditionally known as 4D information in the cheminformatics community, can improve molecular property prediction in deep learning models. We introduce multiple deep learning models that expand upon key architectures such as ChemProp and SchNet, adding elements such as multiple-conformer inputs and conformer attention. We then benchmark the performance trade-offs of these models on 2D, 3D and 4D representations in the prediction of drug activity using a large training set of geometrically resolved molecules. The new architectures perform significantly better than 2D models, but their performance is often just as strong with a single conformer as with many. We also find that 4D deep learning models learn interpretable attention weights for each conformer.

1. Introduction

Drug development is a long and costly process. Bringing a new drug to market takes an average of seven years [1] and costs $2.9 billion (USD, 2013) [2]. Because so many drugs fail in late stage trials, it is critical to generate a variety of leads to increase the chance of success. Leads are often discovered by screening large chemical libraries, but these libraries are expensive to manage, and their chemistry is both homogeneous and poorly reflective of actual drug chemistry [3]. Computational screening can improve this process by exploring a much larger space of compounds and identifying the top candidates for experimental testing. Such methods range from physics-based simulations, such as computational docking [4, 5] and molecular dynamics (MD) [6], to data-driven regressions, such as machine learning (ML) [7–9]. Combinations of both have also produced fruitful results [10]. While docking accuracy is limited by the scoring function and force field [4], the accuracy of ML methods is mostly limited by the amount of available data, and ML inference is orders of magnitude faster.

Given enough data, neural networks used in ML can in principle learn any function, including the mapping from molecule to property. However, there is typically a scarcity of data for successful drugs, since it is difficult to design ligands that strongly bind the target protein. It is therefore necessary to optimize the neural network architecture to best leverage the limited data. Advances in deep learning have improved ML performance by training networks directly on molecular graphs [11–23]. These message-passing neural networks (MPNNs) use graphs to generate learnable, continuous fingerprints, which are then used as input to a neural network that outputs predicted properties.
The state of the art in deep learning uses 2D molecular graphs or single 3D molecular structures to generate these fingerprints. A popular 2D architecture is ChemProp [24]. A model that combines 2D graph information with 3D structural information is PotentialNet, which shows strong performance for the prediction of protein–ligand binding affinity [21]. One way to further improve the representation is to use 3D ensemble information. A molecule is neither a 2D structure nor a single 3D structure, but rather an ensemble of continuously inter-converting 3D structures. The process of drug binding is a 3D recognition event between the drug and the binding pocket, and so depends critically on the 3D structures accessible to the molecule. Ensembles have been used to generate fixed descriptors in computational drug discovery [25], but they have not been used in MPNNs. Nevertheless, the ensemble information may still be learned implicitly by 2D MPNNs, as the set of conformers is in principle a function of the graph only. The extent to which 2D models can implicitly learn this information, and thus match the performance of single- or multi-conformer 3D models, is not currently understood.

Here we investigate whether 3D information of one or more conformers can improve computational drug discovery with ML (figure 1). As an example we screen molecules that can inhibit the novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which causes COVID-19 [26]. We find that models based on a single conformer can better identify held out CoV-2 inhibitors than models based on 2D graphs. However, using multiple conformers does not further improve the results. We also test a transfer learning strategy to leverage the large amount of data for SARS-CoV, which causes SARS, to better predict SARS-CoV-2 inhibition. We find that transfer learning with 3D and 4D models can be better than training 2D models from scratch, but that the improvement is small.

2. Previous work

We distinguish two broad approaches to computational drug discovery. The first, which we refer to as QSAR or classical QSAR (quantitative structure–activity relationship), has long been applied by the computational chemistry community. The second, which we refer to as deep learning, has seen growing interest in the last years driven by advances in representation learning, such as graph convolutional neural networks. Both approaches predict molecular properties by applying a readout function to a molecule’s features. Readout methods include linear, inherently regularized functions like partial least-squares [27], and nonlinear, flexible functions like k-nearest neighbors [28] and neural networks. However, MPNNs differ from classical QSAR approaches, including those with neural network readouts, in that they learn the features directly from the molecules in the training set.

Both the cheminformatics and deep learning communities have focused on the improvement of so-called featurization or representation learning. Researchers in computational drug discovery have developed features that better reflect the molecule-protein binding process (see below). ML researchers have developed models that better ‘train themselves’ to learn powerful featurizations. MPNNs can generate strong molecular representations that outperform hand-crafted descriptors, but require vast amounts of data to do so. In practice their performance is often improved by incorporating features developed by experts [24]. Here we review various 3D and 4D methods that have been extensively developed by the drug discovery community and successfully applied to many targets [29–48]. We then briefly review different 2D and 3D approaches in deep learning and discuss how the principles of 4D QSAR can be applied to MPNNs.
2.1. 3D QSAR
Classical QSAR methods generate features based on physical principles. One such principle is that binding affinity and specificity are determined by the forces between ligands and receptors. Examples that incorporate this principle include the CoMFA \([49]\) and GRID \([50–52]\) methods. Other methods use the principle that ligands with similar shapes have similar binding properties. This is the idea behind the MSA \([53]\), ROCS \([39, 54]\), and Phase Shape \([55]\) methods.

In all 3D methods one must choose a molecular conformation for each species. Molecules exist as an ensemble of conformers, but the most important is the one with the strongest binding affinity to the target. This is known as the bioactive conformation. Many 3D methods assume that the lowest energy conformer is the bioactive conformation \([27, 56]\), but this is often not the case. Others, such as Compass \([57]\), generate a model and an estimate of the bioactive pose together.

2.2. 4D QSAR
4D methods use multiple conformations for each molecule. As in 3D models, one can identify the bioactive pose of a new species by selecting the conformer with the highest predicted activity. Even though the binding pocket is not used during training, the interactions between the predicted bioactive pose and the binding site tend to be energetically favorable \([25]\).

Ash and Fourches \([58]\) used MD trajectories of the ligand–protein complex to generate an ensemble of WHIM shape descriptors \([59]\), and used the mean and standard deviation of the descriptors as additional features. WHIM features are 3D descriptors generated by diagonalizing the weighted covariance matrix of atomic positions. The weights are atomic mass, van der Waals volume, electronegativity, and polarizability. Thus the concatenation of these features describes the 3D distribution of different atomic properties. WHIM descriptors are invariant to translation and rotation, while grid-based and volume overlap methods depend on the orientation of the molecule. Therefore, the alignment of each molecule to a reference structure, which is often ambiguous and can significantly alter model predictions \([27, 60, 61]\), is avoided. Further, explicitly identifying a bioactive conformation was not necessary because the MD simulations were performed with the full ligand–receptor complex. The MD descriptors were shown to strongly distinguish between the most active and the moderately/weakly active and inactive ERK2 kinase inhibitors. They were also shown to provide information absent in 2D descriptors, and even in 3D descriptors from binding poses generated with computational docking. However, since the approach used the protein-ligand complex in MD simulations, it is not clear whether the results would also apply to ligand-only calculations.

2.3. MPNNs
Many QSAR methods generate features based on physical principles that are relevant to the target property. A conceptually different approach is to generate fingerprints that encode the maximum amount of molecular information and can be applied in any context. The current state of the art in deep learning uses MPNNs \([11]\) to learn molecular representations directly from 2D or 3D molecular graphs \([12–20]\). Below we give an overview of MPNNs, and then discuss two models that were used as starting architectures for the development of new 3D and 4D models.

2.3.1. Structure of MPNNs
Molecules can be represented as graphs. These graphs consist of a set of nodes (atoms) connected to each other by a set of edges. Both the nodes and edges have features. The atoms, for example, can be characterized by their atomic number and partial charge. The edges can be characterized by bond type or by interatomic distance. In the message passing phase, MPNNs aggregate node and edge features to create a learned fingerprint. The readout phase uses the fingerprint as input to a regressor that predicts a property \([24]\).

The message passing phase consists of \(T\) steps, or convolutions. In what follows, superscripts denote the convolution number. The node features of the \(t\)th node are \(\bar{x}_v\), and the edge features between nodes \(v\) and \(w\) are \(\bar{e}_{vw}\). The atom features \(\bar{x}\), are first mapped to so-called hidden states \(\bar{h}_v\). A message is \(\bar{m}_{v}^{t+1}\) is created in the \(t\)th convolution, which combines \(\bar{h}_v\) and \(\bar{h}_w\) for each pair of nodes \(v\) and \(w\) with edge features \(\bar{e}_{vw}\) \([24, 62]\):

\[
\bar{m}_{v}^{t+1} = \sum_{w \in N(v)} M_t(\bar{h}_v, \bar{h}_w, \bar{e}_{vw}),
\]  

(1)
where $N(v)$ is the set of neighbors of $v$ in graph $G$, and $M_i$ is a message function. The hidden states are updated using a vertex update function $U_i$:

$$
\tilde{h}_v^{t+1} = U_i(\tilde{h}_v^t, \tilde{m}_v^{t+1}).
$$

(2)

The readout phase then uses a function $R$ to map the final hidden states to a property $y$, through

$$
\hat{y} = R(\tilde{h}_v^T | v \in G).
$$

(3)

A variety of MPNNs have been proposed for learning force fields, which map a set of 3D atomic positions of a molecular entity to an energy. Architectures designed for force fields typically do not incorporate information about the covalent connectivity [17, 18, 63, 64] since these bonds are broken and formed during chemical reactions and may not be clearly defined. This differs from architectures for property prediction, which are typically based on 2D graphs [12, 24, 65] but can also use 3D information [62].

2.3.2. ChemProp

ChemProp is a directed message-passing model that was designed for 2D graphs [24]. It achieves state-of-the-art performance on a wide range of prediction tasks. In this implementation, hidden states $\tilde{h}_v^t$ and messages $\tilde{m}_{vw}^t$ are used, rather than node-based states $h_v^t$ and messages $m_{vw}^t$. Hidden states are initialized with

$$
\tilde{h}_{vw}^0 = \tau([\tilde{x}_v||\tilde{e}_{vw}]），
$$

(4)

where

$$
W_i \in \mathbb{R}^{F \times F_0}
$$

(5)

and

$$
F_0 = \text{dim}(\tilde{x}_v) + \text{dim}(\tilde{e}_{vw}).
$$

(6)

Here $W_i$ is a learned matrix, $F$ is the hidden state dimension, $F_0$ is the dimension of $[\tilde{x}_v||\tilde{e}_{vw}]$, which is the concatenation of the atom features $\tilde{x}_v$ for atom $v$ and the bond features $\tilde{e}_{vw}$ for bond $vw$. The subscript $i$ in $W_i$ stands for initial, indicating that this is the weight matrix used for initialization. $\tau$ is an activation function. The message passing function is simply $M_i(\tilde{x}_v, \tilde{x}_w, \tilde{h}_w^t) = \tilde{h}_{vw}^t$. The edge update function is the same neural network at each step:

$$
U(\tilde{h}_{vw}^t, \tilde{m}_{vw}^{t+1}) = \tau(\tilde{h}_{vw}^0 + W_m \tilde{m}_{vw}^{t+1}),
$$

(7)

here $W_m \in \mathbb{R}^{F \times F}$ is a learned matrix, where $m$ stands for message. Each message-passing phase is then

$$
\tilde{m}_{vw}^{t+1} = \sum_{k \in N(v) \setminus w} \tilde{h}_{kw}^t, \quad \tilde{h}_{vw}^{t+1} = \tau(\tilde{h}_{vw}^0 + W_m \tilde{m}_{vw}^{t+1}),
$$

(8)

for $t = 1, \ldots, T$. After the final convolution, the atom representation of the molecule is recovered through

$$
\tilde{m}_v = \sum_{w \in N(v)} \tilde{h}_{vw}^T, \quad \tilde{h}_v = \tau(W_a[\tilde{x}_v||\tilde{m}_v]).
$$

(9)

The subscript $a$ in $W_a$ stands for atom, indicating that this matrix generates an atom-wise representation. The hidden states are then summed to give a feature vector for the molecule: $\bar{h} = \sum_{v \in G} \tilde{h}_v$. Properties are predicted through $\hat{y}_G = f(\bar{h})$, where $f$ is a feed-forward neural network. In ChemProp the atom features are atom type, number of bonds, formal charge, chirality, number of bonded hydrogen atoms, hybridization, aromaticity, and atomic mass. The bond features are the bond type (single, double, triple, or aromatic), whether the bond is conjugated, whether it is part of a ring, and whether it contains stereochemistry (none, any, E/Z or cis/trans). All features are one-hot encodings. The ChemProp code was accessed through [66].
2.3.3. SchNet
SchNet is a 3D model that was originally designed for ML force fields [17,18]. In this model the feature vector of each atom is first initialized with an embedding function. This embedding generates a random vector that is unique to every atom with a given atomic number, and is also learnable. The edge features at each step \( t \) are generated through a so-called filter network \( V^t \). The filter network generates an edge vector \( \bar{e}_{vw} \) by expanding the distance \( ||\vec{r}_v - \vec{r}_w|| \) in a basis of Gaussian functions. The expansion coefficients are then transformed into \( \bar{e}_{vw} \) through linear and non-linear operations. Because only the distance between two atoms is used to create \( \bar{e}_{vw} \), the features produced are invariant to rotations and translations.

In each convolution \( t + 1 \), the new messages and hidden vectors are given by
\[
\begin{align*}
\bar{m}^{t+1}_v &= \sum_{w \in N(v)} M_t(\bar{h}^t_v, \bar{h}^t_w, \bar{e}_{vw}) = \sum_{w \in N(v)} f'(|\bar{h}^t_v - \bar{h}^t_w|) \circ \bar{e}_{vw}, \\
\bar{h}^{t+1}_v &= U_t(\bar{h}^t_v, \bar{m}^{t+1}_v) = \bar{h}^t_v + f\left( \sum_{w \in N(v)} f'(|\bar{h}^t_v - \bar{h}^t_w|) \circ \bar{e}_{vw} \right),
\end{align*}
\]
where the edge features are
\[
\bar{e}_{vw} = V^t(\vec{r}_{vw}).
\]

Here \( \circ \) denotes element-wise multiplication, and \( f \) and \( f' \) are a series of linear and non-linear operations applied to the atomic features. The neighbors \( N(v) \) of atom \( v \) are those within a pre-set cutoff distance \( r_{cut} \).

2.4. QSAR and MPNNs
Extensive work has been done in the deep learning community to develop and improve MPNNs that act on 2D or single 3D molecular graphs [11–20], but no work has been done on representations with ensembles. Moreover, unlike in QSAR, the issue of bioactive conformation has not been addressed in deep learning, as the lowest energy conformer is the only structure used, and this may not be the bioactive conformation.

Another issue for 3D and 4D models is the generation of conformers. Conformer generation is a difficult task (see section 3.6 below), and the ranking of conformers by energy with classical force fields is highly inaccurate [67]. MD simulations may be able to better sample low-energy regions of phase space than stochastic conformer generators, but without advanced sampling methods can become stuck in local minima. Moreover, the quality of the classical MD conformers is still limited by the accuracy of the force field. This problem affects all models that use single or multiple 3D structures, whether the featurization is hand-crafted or learned.

In this work we combine the physical picture of a bioactive pose, used extensively in QSAR, with tools developed in deep learning. First we develop a number of new 3D MPNN models by combining force field architectures with 2D property predictors. Next we show how the MPNNs can be trained on multiple developed in deep learning. First we develop a number of new 3D MPNN models by combining force field

3. Methods
3.1. Learning with 3D features
Here we combine graph-based models with 3D models in a number of ways. Our first such model is called SchNetFeatures, as it uses the SchNet force field architecture [17,18] (code adapted from [75]), but adds additional graph-based features based on the covalent connectivity. Our second model is called CP3D, as it uses the ChemProp property prediction architecture, but adds additional 3D (distance-based) edge features between atoms within a distance threshold. We have also explored a reduced CP3D model, in which bond states are updated only based on other covalently-bonded neighbors, and not on non-bonded neighbors. In this case the updated bond features are simply concatenated with the original distance features at each step. This model is called CND, for ChemProp3D with no distance updates. A schematic of the architectures is shown in figure 2.
Figure 2. Schematic of the 3D architectures used in this work. (a) SchNetFeatures. The neighbors $w$ of atom $v$ are defined as the atoms within a cutoff distance $r_{\text{cut}}$. This cutoff radius is shown by a dashed orange circle. The features $h_v$ are updated by message-passing that incorporates the neighbors’ features $h_w$, along with distance and bond information for the pair of atoms. (b) CP3D. Each pair of atoms $vw$ separated by $r < r_{\text{cut}}$ has a set of edge features $h_{vw}$. These are updated through the edge features $h_{kw}$, where the atoms $k$ are within $r_{\text{cut}}$ of atom $w$, and exclude atom $v$. The CND model follows the same structure as CP3D, but only the $h_{vw}$ of bonded pairs $vw$ are updated.

3.1.1. SchNetFeatures
For SchNetFeatures we replace the initial atomic feature embedding with the graph-based atom features described above for ChemProp. We also combine distances features with bond features to create the $\vec{e}_{vw}$ of equation (10), through

$$\vec{e}_{vw} \rightarrow [\vec{e}_{vw}^{\text{dist}} || \vec{e}_{vw}^{\text{bond}}] = [\mathbf{V}(r_{vw}) || \vec{e}_{vw}^{\text{bond}}].$$

Here $\vec{e}_{vw}^{\text{dist}}$ are the edge features generated by the SchNet distance filter network. The $\vec{e}_{vw}^{\text{bond}}$ are hidden bond vectors, obtained from a nonlinear operation applied to the bond features, and are set to 0 for non-bonded atoms pairs. Note that applying $F$ in equation (10) mixes the distance and bond features before they are used to update the $h_v$.

In the original SchNet implementation the readout layer converted each atomic feature vector into a single number, and the numbers were summed to give an energy. Consistent with the notion of property prediction, we here instead convert the node features into a molecular fingerprint by adding the features of each node. The readout function is then applied to this fingerprint.

Here we use 10 Gaussian functions for each distance and a cutoff distance of 5 Å. The SchNet force field uses Gaussians spaced 0.1 Å apart [17, 18], which is necessary to resolve movements that lead to appreciable changes in energy. Here we use a much larger spacing (0.3-0.5 Å) because we are interested in larger-scale features of the molecule. Resolving the structure at a finer scale would not be expected to improve performance, and could more easily lead to overfitting.

3.1.2. CP3D
The usual ChemProp model creates and updates edge embeddings for each bond. For CP3D we create an edge embedding for each pair of atoms separated by less than $r_{\text{cut}}$. In particular, we use one initialization matrix for the distances, $\mathbf{W}_{\text{dist}}$, and one for the bond features, $\mathbf{W}_{\text{bond}}$. Hidden states are then initialized through

$$\vec{h}_{vw} = [\vec{h}_{vw}^{\text{dist}} || \vec{h}_{vw}^{\text{bond}}].$$

The distance features $\vec{e}_{vw}^{\text{dist}}$ are the result of a SchNet filter network applied to the distances $r_{vw}$. The bond features for non-bonded pairs are again set to 0, and the remainder of the ChemProp architecture is unchanged (apart from the neighbors now including all atom pairs within $r_{\text{cut}}$ of each other). We use 10 Gaussian functions for each distance and a cutoff distance of 5 Å. We use two convolutions rather than the three used in SchNetFeatures because of higher memory costs.
3.1.3. CND
The CND model differs from CP3D in that only the edge features of bonded pairs are updated. These updates do not use any distance information, meaning that they are equivalent to the updates in the 2D ChemProp model. Distance information is instead incorporated in the following way. After all $T$ convolutions are complete, a set of distance features $\vec{e}_{vw}^{\text{dist}}$ is generated for all pairs $vw$ within $r_{\text{cut}}$ of each other. The $\vec{e}_{vw}^{\text{dist}}$ of bonded pairs are concatenated with the final bonded edge features $\vec{m}_{vw}$, while those of non-bonded pairs are concatenated with zeros. These concatenated vectors are then summed as in equation (9) to give the node features $\vec{m}_n$, which are in turn summed to give the molecular features $h$. The CND model is similar to PotentialNet [21], in that both contain a separation between distance updates and graph updates. PotentialNet uses two sets of gated graph neural networks (GGNNs) [13], the first of which acts on the 2D graph alone. The output of this network is a set of node features $h^T_n$ that have been updated $T$ times. The second GGNN updates the node features $h^T_n$ through convolutions over the 2D graph and the 3D structure. Using the $h^T_n$ as input to the second network means that graph information is already present in the nodes once 3D information is added. Hence the second network can fine-tune the graph-based features using 3D information, rather than generating all features at once. The CND approach also separates the aggregation of graph and distance information (see Methods above). While it does not include distance-based convolutions, its use of edge states means that distance information is included for all pairs within $r_{\text{cut}}$ of each other. This is roughly equivalent to a single 3D convolution in a node-based model.

3.1.4. Incorporating statistical weights
When using these 3D architectures in a multi-conformer model, one should incorporate the statistical weight of each conformer. However, the fingerprint of the $n$th conformer, $\vec{r}(n)$, does not take into account $p(n)$. Therefore, we need a representation that combines both pieces of information. To generate this representation, we first embed the weight as a vector $\vec{d}(n)$, through

$$\vec{d}(n) = \text{SoftMax}(Dp(n) + \vec{b}), \quad (14)$$

where $D \in \mathbb{R}^{S \times 1}$ is a learned matrix, $S$ is the dimension of $\vec{d}(n)$, and $\vec{b}$ is a learned bias. The softmax activation means that $\vec{d}_k(n)$ can be interpreted as the projection of $p(n)$ onto the $k$th probability bin. Here we choose $S = 10$ so that the probabilities are divided into 10 bins. A linear layer is then applied to the concatenation of $\vec{d}(n)$ with $\vec{r}(n)$, yielding the final conformer fingerprint $\vec{q}(n)$:

$$\vec{q}(n) = H[\vec{r}(n) || \vec{d}(n)] + \vec{b}. \quad (15)$$

Here $H \in \mathbb{R}^{F \times (F + S)}$ is a learned matrix, and $\vec{b}$ is again a learned bias.

3.2. Attention model
It is not immediately clear how to pool the fingerprints of the different conformers. A simple pooling scheme would be to multiply the fingerprint of each conformer by $p(n)$ and add the results. However, this assumes that the resultant property is a statistical average of individual conformer properties, and this is not always the case. For example, the contribution of a conformer to the binding process is not determined by its conformational energy. Rather, it is determined by the affinity of the conformer for the target, but this affinity is not known a priori, and large differences in binding energy may compensate for small differences in conformational energy.

An ideal pooling scheme would be adaptable, learning fingerprint weights that are best suited to the task at hand. For this reason we propose a pooling mechanism based on attention [68–71]. Attention combines a learned vector with two feature vectors to determine the importance of one set of features to another. The resulting coefficients are then normalized to give weights for each feature vector. Similar to other applications of attention [71], we have found that it is useful to include multiple attention heads in the pooling. These different heads can learn different features to focus on in the pooling.

The notion of attention can be applied to conformer fingerprints in one of two ways. The first is to let the attention mechanism learn the importance of one conformer’s features to another, giving a set of attention coefficients $\alpha_{nk}$ for $n, k \in N$, where $N$ is the number of conformers. We call this pair-wise attention. The second, which we call linear attention, is to simply learn the conformer’s overall importance, rather than its importance with respect to another conformer. This gives a set of coefficients $\alpha_n$ for each conformer. The choice of pair-wise attention or linear attention is treated as a hyperparameter to be optimized. The motivation for linear attention is that the model can learn the most likely bioactive conformation and assign it the highest weight. Pair-wise attention does not have a simple physical interpretation, but was included as an option in case it improved performance.
Figure 3. Schematic of the architecture in this work. (a) A neural network uses the 3D structure of the $n$th conformer, along with its graph information, to create a molecular fingerprint $\vec{h}(n)$. (b) Each fingerprint is combined with an embedding of the conformer's statistical weight $p(n)$. The new fingerprints $\vec{q}(n)$ are aggregated through an attention mechanism to yield the final fingerprint $\vec{Q}$.

3.2.1. Computing attention coefficients

For the pairwise attention method we then compute the attention coefficients $c_{nk}$ between conformers $n$ and $k$ as

$$c_{nk} = \vec{a} \cdot [\mathbf{A} \vec{q}(n)] [\mathbf{A} \vec{q}(k)],$$

where $\vec{a} \in \mathbb{R}^{2F}$ is a learned vector, $\mathbf{A} \in \mathbb{R}^{F \times F}$ is a learned matrix, and $\cdot$ is the dot product. For ease of notation we have suppressed the dependence on the attention head index. The attention paid by conformer $n$ to conformer $k$ is then

$$\alpha_{nk} = \frac{\exp(\text{LeakyReLU}(c_{nk}))}{\sum_{l} \exp(\text{LeakyReLU}(c_{nl}))}.$$  \hspace{1cm} (17)

The pooled fingerprint $\vec{Q}$ is a weighted sum over each fingerprint followed by an activation function $\tau$, such that

$$\vec{Q} = \tau \left( \frac{1}{N} \sum_{nk} \alpha_{nk} \mathbf{A} \vec{q}(k) \right).$$  \hspace{1cm} (18)

Here $N$ is the number of conformers, and the division by $N$ ensures that $\sum_{nk} \alpha_{nk}/N = 1$. For the linear attention mechanism only one conformer is used to determine each coefficient, through $c_{n} = \vec{a} \cdot (\mathbf{A} \vec{q}(n))$, where $\vec{a} \in \mathbb{R}^{F}$. The coefficients $\alpha_{n}$ are normalized by summing over one index only, and the pooled conformers are

$$\vec{Q}_{\text{linear}} = \tau \left( \sum_{n} \alpha_{n} \mathbf{A} \vec{q}(n) \right).$$  \hspace{1cm} (19)

Adding a superscript to denote fingerprints from different heads, the final fingerprint resulting from $K$ linear or pair-wise attention heads is then

$$\vec{Q} = [\vec{Q}^{(1)} || \vec{Q}^{(2)} || \ldots || \vec{Q}^{(K)}].$$  \hspace{1cm} (20)

A schematic of the fingerprinting architecture is shown in figure 3.
3.3. Average distance model

As an alternative to the attention-based model we also propose an average distance model. In this approach, message-passing occurs over a single effective conformer, with interatomic distances given by the distances averaged over all conformers. That is, the effective distance \( d_{ij} \) between atoms \( i \) and \( j \) is given by

\[
\bar{d}_{ij} = \sum_n p_n d_{ij}^{(n)},
\]

(21)

where the sum is over all conformers, \( p_n \) is the statistical weight of the \( n \)th conformer, and \( d_{ij}^{(n)} \) is the distance between atoms \( i \) and \( j \) in the \( n \)th conformer. The benefit of this approach is the enormous reduction in computational cost compared to using multiple conformers. Since only one effective graph is needed per species, the computational cost is reduced by a factor of \( p_{\text{conf}} \), the average number of conformers per species. The drawback, however, is that we are forced to use non-learnable weights, as learnable weights would require each conformer to be stored in the computation graph. Here we have used statistical weights, but it would also be possible to use any other weights that are determined before training. For example, one could first train an attention model on non-learnable fingerprints, and then use the resulting weights in equation (21).

It is important to notice that equation (21) is not equivalent to a single geometry whose atomic positions are averaged over all conformers. The reason is that \( d_{ij} \) is a nonlinear function of the positions \( r_i \) and \( r_j \), since \( d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \). Hence the average distance does not equal the distance between average positions. Equation (21) then contains information of the position distribution up to at least its second order moment. Note that equation (21) is invariant to translations or rotations of any one conformer, since each \( d_{ij}^{(n)} \) is itself invariant. The average distance models are also invariant to atom re-indexing. This is because they are built upon permutation-invariant 3D models; the only change is that they replace the distances \( d_{ij} \) with distances \( d_{ij} \) that are averaged over all conformers. Invariance to atom-reindexing is independent of the numerical values of the edge features, and so invariance is maintained when using average distances.

3.4. WHIM model

To connect with recent 4D approaches in the drug discovery literature, we have also followed [58] and used the mean and standard deviation of conformer WHIM vectors for training. For each species we computed the WHIM vector of each conformer using RDKit [76], and computed the mean and standard deviation using the conformers’ statistical weights. These features were then combined with learnable fingerprints generated by ChemProp. As is the default for external features supplied to ChemProp, each of the WHIM features was scaled to have zero mean and variance. Our approach differs somewhat from [58] in that our conformers are generated in vacuum rather than sampled from ligand-receptor MD, and also that we concatenate the WHIM features with those generated from message-passing.

3.5. Baseline models

We compared the 3D models to the 2D-based ChemProp and random forest models, with the random forest trained on Morgan fingerprints [77]. We experimented with using atom-pair fingerprints [78] for the random forest training, but did not find a significant difference in performance. We also experimented with a variant of the compass approach to finding bioactive conformations [57], training feed-forward networks on WHIM vectors instead of on ligand-query volume differences. We then used the predicted bioactive conformers as input to single conformer models. However, we discarded this approach because, for some targets, the model performance decreased as the predicted poses were updated. This indicated that the quality of the poses became worse, and hence that the iterative procedure was non-convergent.

3.6. Obtaining conformers

To train a model on 3D ensemble information, one must first reduce the infinite set of 3D structures to a finite set of conformers, the set of thermally accessible structures at local minima on the potential energy surface. However, generating accurate conformers is challenging. A variety of exhaustive, stochastic, and Bayesian methods have been developed to generate conformers [76, 79–85]. The stochastic conformer generation methods used in RDKit are not exhaustive and may miss low-energy structures, and exhaustive sampling has prohibitive exponential scaling with the number of rotatable bonds. Further, the classical force fields used are generally not accurate enough to assign conformer orderings based on energy [67].

Significant progress in conformer generation has been made with the CREST program [73], which uses semi-empirical quantum chemistry and advanced sampling methods to generate reliable geometries. Conformer accuracy is crucial for all models that use 3D geometries. For example, in the QSAR Compass approach one considers only conformers with energy below 2.0 kcal mol\(^{-1}\) as candidates for the bioactive
conformation [57]. However, because classical force fields cannot reliably assign energies in this narrow window [67], it is easy to include high-energy conformers and miss low-energy ones. The CREST conformer orderings are more reliable because the energies are generated with semi-empirical DFT, which is a good approximation to high-accuracy \textit{ab initio} methods [72]. Further, stochastic methods can miss conformers, and MD simulations can become stuck in local minima even with long simulations. Indeed, the latter was suggested as one reason for the poor performance of the per-MD-frame WHIM vectors used in [86]. These issues are avoided in CREST through the use of multiple rounds of metadynamics, high-temperature MD and genetic methods [73].

We recently used CREST to create the GEOM dataset [72, 87, 88], which contains conformers annotated by energy for over 300 000 drug-like molecules with experimental binding data and 130 000 combinatorially generated small molecules [89]. The 300 000 drug-like molecules contain ligands with \textit{in-vitro} data for the inhibition of SARS-CoV 3CL protease, SARS-CoV PL protease, SARS-CoV-2 3CL protease, SARS-CoV-2, \textit{E. coli} and \textit{Pseudomonas aeruginosa} [9, 90–97]. We use this dataset to train the 3D conformer-based models introduced in this work. Figures 4(a)–(c) shows the number of conformers per molecule for each of the three datasets described below. Panels (d)–(f) show the average root-mean-square displacement (RMSD) among all pairs of conformers for a given molecule. The differences in molecular flexibility among the datasets, and the associated impact on model performance, are discussed below.

### 3.7. Training

We trained different models on experimental data for the specific inhibition of the SARS-CoV-2 3CL protease (‘CoV-2 CL’) [92], the general inhibition of SARS-CoV-2, measured \textit{in vitro} in human cells [90, 91], and the specific inhibition of the SARS-CoV 3CL protease (‘CoV CL’) [93]. The CoV 3CL protease is found in the virus causing SARS, and has 96% sequence similarity to CoV-2 CL [98]. Because there is three orders of magnitude more data for CoV 3CL than for CoV-2 3CL (table 1), and because the sequences are so similar, we also explored using transfer learning (TL) from CoV 3CL to CoV-2 3CL.

Prediction of anti-viral activity was treated as a binary classification problem, with each compound labeled as either a hit or a miss. Model performance was measured using the receiver operating characteristic area under the curve (ROC-AUC), the precision-recall area under the curve (PRC-AUC), and the ROC

---

**Figure 4.** Molecular flexibility statistics for each of the three datasets. (a)–(c) Number of conformers per molecule. Note the log scale of the y-axes. (d)–(f) Mean RMSD among conformers. For every molecule we aligned each pair of conformers, computed their RMSD, and then averaged the RMSDs over all pairs. We used all molecules in the CoV-2 3CL dataset, and randomly sampled 1000 molecules from each of the other two datasets.

**Table 1.** Number of hits (total number of species in parentheses) for each split of each dataset.

| Dataset    | CoV-2 3CL | CoV-2 | CoV 3CL |
|------------|-----------|-------|---------|
| Train      | 50 (485)  | 53 (3294) | 275 (167 255) |
| Validation | 15 (157)  | 17 (1096) | 81 (55 750)   |
| Test       | 11 (162)  | 22 (1086) | 70 (55 756)   |
| Total      | 76 (804)  | 92 (5476) | 426 (278 758) |
enrichment (ROCE) [99] at 0.5%, 1%, 2% and 2%. Further discussion of these metrics can be found in the Supplementary Material (SM). For the specific inhibition of the CoV-2 3CL protease [92], hits were defined as any ligand forming a complex with the target protease, identified using the crystallographic techniques of [100]. For general activity against SARS-CoV-2, compounds were screened for inhibition of viral-induced cytotoxicity in vitro. Hits from [90] were defined as any molecule with half-maximal inhibitory concentration (IC50) below 20 µM. Hits from [91] were those with an inhibition index above 1, where the inhibition index is a measure of cell viability measured with optical density methods. For the CoV 3CL protease, compounds were tested for the inhibition of protease-mediated peptide cleavage [93]. The cleavage event was measured through fluorescence techniques, which were based on the separation of a fluorescent compound and a quencher attached at opposite ends of the peptide. A compound was identified as a hit if it inhibited peptide cleavage.

We used the conformers from the GEOM dataset as inputs to different 3D/4D models (SchNetFeatures, CP3D, CND). The training was performed with one conformer, with up to 200 conformers, and with a single effective conformer. We also trained a ChemProp + WHIM model, which contains non-learnable conformer ensemble information. For comparison we also trained models with only graph information (ChemProp, and random forest over circular fingerprints). These networks were first trained on the CoV 3CL task and the two CoV-2 tasks. The CoV 3CL models were then used to generate fixed features for molecules in the CoV-2 3CL dataset for TL. Details of the data preprocessing, hyperparameter optimization, and uncertainty quantification of the model scores can be found in the SM.

4. Results

ROCE scores of various models are shown in figure 5, and ROC and PRC scores are given in table 2. For visualization purposes we have also included thin horizontal bars for each ROCE. The thin bars of identical ROCE scores are shifted down to avoid overlap. Bars stacked vertically with no spaces between them all have the value of the highest bar in the stack. For example, thin bars at negative y-values are always part of a stack for which the highest value is \( \geq 0 \).

4.1. CoV-2 3CL

We begin by analyzing performance on CoV-2 3CL, the task with the smallest amount of data, and then proceed to datasets of increasing size (CoV-2 followed by CoV 3CL). We use the labels 'single', 'multiple', and 'average' to respectively refer to models that use the lowest energy conformer, the lowest 200 conformers, and the average distances among conformers.

Figure 5(a) shows that the 2D models generally have the best ROCE scores, while the 3D and 4D models are generally no better than the 2D models. The only exception is CND (single) at 2% FPR, where its ROCE of 22.3 is slightly higher than ChemProp’s ROCE of 20.8. The ChemProp model with WHIM features averaged over all conformers (‘CP WHIM’) performs similarly to ChemProp and random forest. All other 3D and 4D models are significantly outperformed by 2D models.

We also see that using MPNNs in the 2D models is largely unnecessary, as a simple random forest has comparable performance to the ChemProp models. For example, the scores and rankings of random forest and ChemProp are, respectively, 20.6/first and 14.4/third; 28.1/first and 27.8/second; 18.9/third and 20.8/second; and 10.5/fourth and 11.4/first. The random forest is the best model at 0.5% and 1% FPR, and is only slightly outperformed by ChemProp at higher FPR.

The ROCE scores are consistent with the ROC and PRC scores in table 2. Here we see that ChemProp, random forest, CND (single) and CP WHIM all have high ROC and PRC scores. In fact, the CND (single) model has the highest ROC score while the CP WHIM model has the highest PRC. The other CND models also perform well, with CND (multiple) and CND (average) achieving better ROC scores than ChemProp, and reasonably high PRC scores.

This task has the smallest amount of data (485 training species), which may explain the results: with so little data, it is difficult for the neural networks to learn meaningful representations. The learned representations are likely no more powerful than the Morgan fingerprints used in random forest; this is often the case in low-data regimes [101]. This is especially true of the 3D models with distance updates, as they haven an abundance of input information with few labels to guide the representation learning. This may make the 3D models especially prone to overfitting.

4.2. CoV-2

The results are quite different for the CoV-2 task. As seen in figure 5(c), the ChemProp and random forest models are outperformed by a number of 3D models. The CND (single) model in particular has the highest ROCE scores at 0.5%, 1%, and 2% FPR, and is essentially tied with CND (average) and CP3D (multiple) for
the highest score at 5% FPR. The CND (single) model outperforms ChemProp, the best 2D model, by a significant margin. At 0.5% FPR, CND (single) has an ROCE of 16.1, nearly 40 times higher than ChemProp’s ROCE of 0.4. At lower FPR values the improvement is smaller but still quite significant, with CND (single) outperforming ChemProp by factors of 26, 11, and 4 at FPRs of 1%, 2%, and 5%, respectively.

Other 3D and 4D models also perform better than ChemProp, though none achieves the success of CND (single). For example, SchNetFeat (single), CND (multiple), and CND (average) all achieve high ROCE scores, improving on the ChemProp scores by factors of 4–20 at various FPR values. WHIM fingerprints also improve the performance of ChemProp, increasing ROCE scores by factors between 3 and 9. This should be contrasted with the CoV-2 3CL task, in which adding WHIM fingerprints did not improve performance. We also note that the random forest model performs quite poorly, achieving zero ROCE for 0.5% and 1% FPR, and an ROCE of 0.22 at 2% FPR (three times lower than ChemProp). Evidently message-passing is far more helpful here than in the CoV-2 3CL task.

These trends are reflected in the PRC scores, and to a lesser extent in the ROC scores. Table 2 shows that SchNetFeat (single), CND (single), CND (multiple), and CND (average) all have some of the highest ROC scores. However, the same is true of CP WHIM, which has a rather low ROC score. Moreover, ChemProp and random forest, which have the lowest ROC scores, have ROC values only slightly below those of the best models (approximately 8%). Clearly the ROC score is insufficient for evaluating the models. By contrast, the PRC scores show that ChemProp and Random Forest are significantly outperformed by SchNetFeat (single), CND (average) and CP WHIM, which in turn are significantly outperformed by CND (single). Indeed, the CND (single) score is four times that of ChemProp and double that of SchNetFeat (single), CND (average) and CP WHIM. The CP3D (average) model also has a very high PRC, though interestingly its ROC is rather low. We also note that while CND (single) performance is quite impressive, the CND (single) model with the best validation PRC also had the best validation ROC, so the uncertainty for each of its scores is zero (see SM). It is therefore possible that the extremely high PRC and ROC scores are somewhat lucky. However, even if we ignored the CND (single) model, the next best 3D models would still be significant improvements over the 2D models.

For this task it is clear that 3D information can significantly improve performance. Two factors distinguish this task from the CoV-2 3CL problem. First, there is substantially more data (see table 1). Second, the hits may have completely different mechanisms of action from each other, as they inhibit the in-vitro growth of SARS-CoV-2 in human cells, while the CoV-2 CL data is for ligands that specifically
inhibit the CoV-2 3CL protease. Indeed, the ROC scores are far lower than the best ROC scores in the CoV-2 3CL task, indicating that this problem is more difficult. Below we show that MPNN-based 3D and 4D models also perform well for the CoV-3CL classification, but that they are only slightly better than 2D models. Therefore, it is possible that 3D information is more important when the mechanisms of action are different, and/or when there is an intermediate amount of training data (not too much or too little). Further investigation is required to test these hypotheses.

Another important observation is that the 4D models are no better than the 3D models. For example, the 4D models with the best ROC scores are CND (multiple) and CND (average), but both are outperformed by CND (single) at FPR values below 5% and matched at 5%. Similarly, SchNetFeat (multiple) is underperformed by SchNetFeat (single) at every FPR. The exception to this trend is CP3D (average), which is more accurate than CP3D (single). However, it is still far worse than the other 3D models. These trends can also be seen in the ROC and PRC scores.

Since all information contained in a single conformer representation is also available with 200 conformers, it seems that the model training is hampered by the extra information. This is particularly intriguing because the attention mechanism learns meaningful information about the conformers (see below), focusing on geometries that are similar to those of other hits. The model may be overfitting to the extra information, despite the hyperparameter optimization to determine reasonable dropout rates.

### 4.3. CoV 3CL

The final target is CoV 3CL. It contains by far the largest training set, and, like CoV-2 3CL, contains experimental data for ligands that bind to a specific protein. According to the ROC scores in figure 5(d), the models that perform best on average are CND (average), CP WHIM, CND (single), CND (multiple), and ChemProp. The CND (average) is best at FPRs of 0.5% and 1% (ROCE = 47.0 and 28.6), CP WHIM is best at 2% (ROCE = 17.1), and ChemProp is best at 5% (ROCE = 8.6).

The 3D/4D models are generally better than the 2D models, but only by a small margin. For example, the ratios of top 3D/4D score to top 2D score are 1.18, 1.05, 1.09, and 0.95 at FPRs of 0.5%, 1%, 2% and 5%, respectively. Interestingly, this is the only task in which 4D models consistently outperform 3D models, but

| Table 2. Performance of various models on the three classification tasks, evaluated with the ROC and PRC scores. Uncertainty on the last digit is denoted by a subscript, and is given here by the standard deviation of test scores. Uncertainties less than $10^{-3}$ are recorded as zero. Top scores are bolded. Due to computational constraints we did not train CP3D (multiple) on the CoV 3CL task. |
|----------------------------------|-----------------|-----------------|-----------------|
|                                  | CoV-2 3CL       | CoV-2           | CoV 3CL         |
|----------------------------------|-----------------|-----------------|-----------------|
| Random Forest                    | 0.92_{(1)}      | 0.61_{(1)}      | 0.69_{(1)}      |
| ChemProp                         | 0.89_{(1)}      | 0.62_{(4)}      | 0.75_{(7)}      |
| CND (single)                     | 0.929_{(3)}     | 0.67_{(0)}      | 0.72_{(3)}      |
| CP3D (single)                    | 0.78_{(5)}      | 0.50_{(1)}      | 0.65_{(2)}      |
| SchNetFeatures (single)          | 0.907_{(3)}     | 0.676_{(1)}     | 0.718_{(7)}     |
| CND (multiple)                   | 0.901_{(4)}     | 0.663_{(8)}     | 0.759_{(2)}     |
| CP3D (multiple)                  | 0.66_{(7)}      | 0.53_{(0)}      | 0.70_{(3)}      |
| SchNetFeatures (multiple)        | 0.86_{(2)}      | 0.63_{(2)}      | 0.72_{(1)}      |
| CND (average)                    | 0.916_{(3)}     | 0.647_{(3)}     | 0.71_{(0)}      |
| CP3D (average)                   | 0.73_{(0)}      | 0.56_{(2)}      | 0.70_{(3)}      |
| SchNetFeatures (average)         | 0.84_{(3)}      | 0.61_{(1)}      | 0.74_{(2)}      |
| ChemProp + WHIM                  | 0.89_{(2)}      | 0.66_{(5)}      | 0.77_{(7)}      |

|----------------------------------|-----------------|-----------------|-----------------|
|                                  | CoV-2 3CL       | CoV-2           | CoV 3CL         |
|----------------------------------|-----------------|-----------------|-----------------|
| Random Forest                    | 0.55_{(3)}      | 0.028_{(2)}     | 0.079_{(2)}     |
| ChemProp                         | 0.56_{(6)}      | 0.028_{(3)}     | 0.05_{(1)}      |
| CND (single)                     | 0.533_{(1)}     | 0.124_{(0)}     | 0.07_{(2)}      |
| CP3D (single)                    | 0.33_{(6)}      | 0.021_{(0)}     | 0.04_{(3)}      |
| SchNetFeatures (single)          | 0.333_{(0)}     | 0.051_{(3)}     | 0.021_{(6)}     |
| CND (multiple)                   | 0.413_{(3)}     | 0.06_{(1)}      | 0.04_{(3)}      |
| CP3D (multiple)                  | 0.20_{(1)}      | 0.032_{(0)}     | —               |
| SchNetFeatures (multiple)        | 0.26_{(5)}      | 0.037_{(3)}     | 0.03_{(2)}      |
| CND (average)                    | 0.467_{(5)}     | 0.058_{(4)}     | 0.063_{(1)}     |
| CP3D (average)                   | 0.31_{(0)}      | 0.10_{(6)}      | 0.07_{(1)}      |
| SchNetFeatures (average)         | 0.29_{(6)}      | 0.027_{(0)}     | 0.07_{(3)}      |
| ChemProp + WHIM                  | 0.57_{(6)}      | 0.04_{(1)}      | 0.07_{(2)}      |
the margin is again rather small. Specifically, the ratios of top 4D score to top 3D score are 1.84, 1.11 1.20, and 1.14 at FPRs of 0.5%, 1%, 2% and 5%, respectively.

We also see that the random forest is outperformed by MPNN models, though to a lesser extent than in the CoV-2 task. The superiority of MPNNs is to be expected for this task because the dataset is quite large. This provides ample information from which to learn an ideal molecular representation. The size of the dataset also likely explains the fact that the ROCE scores are higher than in any other task.

The ROCE trends are only somewhat similar to the PRC and ROC trends. The random forest model has the highest PRC score, and the CND (single), SchNetFeat (average), CP3D (average) and CP WHIM models are close behind. The ChemProp PVC is noticeably lower than each of these models, while CND (multiple) score is lower still. While the CND (single), CND (average) and CP WHIM models also have high ROCEs, the CP3D (average) and random forest models do not. In fact, according to the ROCE metric, the random forest is one of the worst models. Moreover, ChemProp and CND (multiple) have high ROCE scores but low PRCs. These results again emphasize the importance of going beyond the ROC and PRC scores when comparing models.

4.4. Transfer learning

We also analyzed how TL could improve CoV-2 3CL scores. To do so we used the pre-trained CoV 3CL models to generate fingerprints, and used the fingerprints as input to a standard feed-forward neural network. We did this with and without additional message passing from ChemProp. When additional message passing was used, the fixed fingerprints were concatenated with the learned ChemProp fingerprints. For each architecture we used two models, one with the best CoV 3CL AUC score and one with the best CoV 3CL PRC score. Each of the two pre-trained models was then used to produce molecular fingerprints, which were then used to train an ensemble of ten different CoV-2 3CL models. The final scores are thus an average over 20 different models. Hyperparameters were again optimized separately for each model.

Transfer learning ROC and PRC score are shown in table 3, and can be compared with the ChemProp and random forest baselines in table 2. The ROCE scores are shown in figure 5(b), and correspond to averages over 40 models (20 with additional message-passing and 20 without). The ROC and PRC scores show no evidence that TL improves performance, as the best TL scores are lower than those of ChemProp and Random Forest without pre-training. However, the ROCE scores show that TL helps at low FPR. In particular, SchNetFeat (multiple) TL at 0.5% FPR has an ROCE of 25.7, which is 1.25 times greater than that of random forest (ROCE = 20.6), and 1.78 times greater than that of ChemProp (ROCE = 14.4). CND (single) TL also shows strong performance, as its ROCE of 22.3 is 1.08 times larger than random forest and 1.54 larger than ChemProp. The strong performance of these models is consistent with their high ROC and

| CoV 3CL model for fingerprinting | CoV-2 3CL message passing (Y/N) | ROC   | PRC   |
|----------------------------------|---------------------------------|-------|-------|
| ChemProp (2D)                    | Y                               | 0.78(3) | 0.36(8) |
| ChemProp (2D)                    | N                               | 0.71(6) | 0.35(7) |
| SchNetFeatures (single)          | Y                               | 0.76(8) | 0.3(1)   |
| SchNetFeatures (single)          | N                               | 0.77(7) | 0.3(2)   |
| CP3D (single)                    | Y                               | 0.75(5) | 0.2(1)   |
| CP3D (single)                    | N                               | 0.69(6) | 0.2(1)   |
| CND (single)                     | Y                               | 0.85(3) | 0.43(9)  |
| CND (single)                     | N                               | 0.81(4) | 0.47(9)  |
| SchNetFeatures (multiple)        | Y                               | 0.86(6) | 0.42(9)  |
| SchNetFeatures (multiple)        | N                               | 0.87(5) | 0.50(9)  |
| CND (multiple)                   | Y                               | 0.88(3) | 0.47(9)  |
| CND (multiple)                   | N                               | 0.87(5) | 0.4(1)   |
| SchNetFeatures (average)         | Y                               | 0.79(6) | 0.3(1)   |
| SchNetFeatures (average)         | N                               | 0.83(6) | 0.4(1)   |
| CP3D (average)                   | Y                               | 0.6(2)  | 0.2(1)   |
| CP3D (average)                   | N                               | 0.74(6) | 0.29(9)  |
| CND (average)                    | Y                               | 0.88(2) | 0.48(9)  |
| CND (average)                    | N                               | 0.87(3) | 0.41(5)  |
| ChemProp + WHIM                  | Y                               | 0.77(3) | 0.42(4)  |
| ChemProp + WHIM                  | N                               | 0.81(2) | 0.44(3)  |
In the table we have reported the uncertainty as the standard error of the mean (SEM). The standard deviation would indicate the expected range of scores for any single pair of species, and thus the likelihood of the attention model being the random model for any one pair. By contrast, the SEM measures our confidence that the average difference between the models is statistically significant. The SEM values in table 4 suggest that the difference between the models is not random.

| Species | CoV | CoV-2 | CoV-2 3CL |
|---------|-----|-------|-----------|
| Hit/hit | Hit/miss | ∆     | Hit/hit | Hit/miss | ∆     |
| Attention | 0.325 (1) | 0.3074 (1) | **0.018** (1) | 0.340 (1) | 0.291 (1) | **0.049** (1) | 0.378 (1) | 0.276 (1) | **0.102** (1) |
| Random   | 0.321 (1) | 0.3054 (1) | 0.015 (1) | 0.323 (1) | 0.293 (1) | 0.031 (2) | 0.344 (1) | 0.273 (1) | 0.071 (2) |

Table 4. Average cosine similarity between E3FP fingerprints of different species. A single conformer was selected for each species, either randomly or based on highest attention weight from a SchNetFeatures model with the lowest validation loss. ‘hit/hit’ means similarity among hits, ‘hit/miss’ means similarity between hits and misses, and ∆ = (hit/hit) − (hit/miss). The highest ∆ scores for each task are shown in bold; 5000 comparisons were made for each category. Uncertainty on the last digit is denoted by a subscript and given by the standard error of the mean.

PRC scores, though the latter never exceed those of non-TL models. At FPR values above 0.5% there is no advantage to using transfer learning, as the highest scores are always those of non-TL models.

It is interesting that 3D/4D TL models perform better than the 2D TL model. This is reflected in the ROCE, ROC, and PRC scores. It seems that the representations learned by the 3D models are more easily generalized to new targets than the 2D representations. It is also noteworthy that SchNetFeat (multiple), a 4D model, provides the best TL results. However, by all metrics SchNetFeat (multiple) is one of the worst models at predicting CoV 3CL inhibition, while CND (single) is one of the best. It is therefore difficult to explain why they provide some of the best TL results at low FPR.

The ROCE improvements at 0.5% FPR are certainly interesting, but they are not enough to justify pre-training a 3D or 4D model with hundreds of thousands of species. The non-TL ChemProp model performs better at all other FPR values, and a random forest with Morgan fingerprints has competitive accuracy. Further work is needed to evaluate the usefulness of TL in drug discovery.

4.5. Conformer importance

Using 4D models leads to minor improvement over 3D models for CoV 3CL, but hurts performance for other targets. Moreover, at each FPR value the best 4D model is CND (average) or CP WHIM, neither of which uses attention to determine the most important conformation. It is therefore important to understand how conformers are pooled to see if the models are learning meaningful information from them. To this end we analyzed the similarity of high-attention conformers in different species. For each comparison we randomly selected two species, and for each species we selected the conformer with the highest attention weight among all attention heads. We then computed the fingerprints of the two conformers using the E3FP method [102], which is an extension of the extended connectivity fingerprint to 3D structures. The cosine similarity metric was then applied to the fingerprints. This was repeated for 5000 random pairs in the test set to obtain the average similarity. We applied this analysis separately to pairs of species that were both hits, and to pairs of species that contained one hit and one miss. We also repeated the analysis with a random selection of conformers for comparison.

Results of this analysis are shown in table 4. Random selection of conformers yields hits that are more similar to each other than to misses. Choosing conformers based on attention increases the similarity of hits to each other, while leaving the similarity of hits to misses virtually unchanged. The difference between hit-hit and hit/miss scores, denoted by ∆, is on average 40% larger for conformers chosen by attention than conformers chosen randomly. This means that it is easier to distinguish hits from misses using attention conformers than using random conformers. Interestingly, unlike the other models, CND (multiple) did not learn meaningful attention weights for the CoV 3CL task, assigning equal weights to nearly every conformer. This could be because of the high molecular flexibility of this dataset: as shown in figure 4(c), it has the widest distribution of conformer numbers of any dataset, and has the highest mean RMSD among geometries. Despite the lack of meaningful attention weights, the CND (multiple) model trained on CoV 3CL had the strongest performance of any attention-based 4D model. This result, along with the strong performance of the attention-free CP WHIM and CND (average) models, suggest that best results may be achieved without attention. However, the clear evidence that attention weights contain meaningful conformer information indicate that further investigation is required.

4.6. Average distances vs. single conformer

It is of interest to specifically compare the average distance model to the single conformer model. These approaches use the same architecture and hyperparameters, and differ only in the whether multiple

---

3 In the table we have reported the uncertainty as the standard error of the mean (SEM). The standard deviation would indicate the expected range of scores for any single pair of species, and thus the likelihood of the attention model beating the random model for any one pair. By contrast, the SEM measures our confidence that the average difference between the models is statistically significant. The SEM values in table 4 suggest that the difference between the models is not random.
conformers are incorporated into interatomic distances. Hence in some sense, they provide the most direct measure of the importance of multiple conformers.

The results are mixed, and depend on the task, metric, and architecture. Figure 5 shows that average distances hurt the ROCE score for the CND model in the CoV-2 3CL, CoV-2 3CL (TL), and CoV-2 tasks, and have virtually no effect for the CoV 3CL task. Using average distances improves the ROCE for SchNetFeatures in the first two tasks, but hurts performance for the latter two. There is also mixed performance for both architectures according to the ROC and PRC metrics, as shown in table 2. Interestingly, however, average distances show a more positive effect for the ROC and PRC metrics for the TL task. As shown in table 3, the SchNetFeatures results are substantially improved when using average distances, while CND results are improved for three of the four (metric, architecture) combinations.

Figure 6 compares the conformer-averaged interatomic distances to those of the highest-probability conformer. The differences between the two increase from CoV-2 3CL to CoV-2 to CoV 3CL, with the mean value climbing from 0.17 Å to 0.22 Å to 0.27 Å. This is similar to the trends in flexibility statistics in figure 4. It is interesting that the absolute deviations in distances are rather small, but that the effects on performance are both noticeable and inconsistent across architectures and metrics.

5. Discussion

5.1. Accuracy
Adding 3D information can improve prediction accuracy when using the right model. Of all the models introduced, the new CND model shows the strongest and most consistent performance, either matching or beating 2D baselines in most tasks. CND (single) in particular shows the most consistent performance of all models, even on the CoV-2 3CL task with little data.

While the CND model matches or improves upon 2D baselines, many 3D models do not improve performance, and some can actually hurt it. For example, CP3D is outperformed by 2D models in every task, while SchNetFeatures improves results in CoV-2 but hurts them in CoV 3CL. It is also clear that adding extra conformers hurts performance in most tasks (CoV-2 3CL and CoV-2), and only leads to small improvements in others (CoV 3CL). Given the enormous computational cost of training the multiple models (see below), it is a far better use of resources to train single conformer models. Transfer learning can somewhat improve accuracy, but the effect is small and absent in most models. Much of the improvement can be matched with a simple 2D baseline like random forest.

The results of MPNN-based 3D models and 3D models with non-learnable features are also mixed. For CoV-2 3CL it is best to use a 2D model or a 3D model with non-learnable WHIM features. For CoV 3CL it is best to use a 3D or 4D model with learnable features, but using WHIM features gives comparable performance. By contrast, for CoV-2 it is far better to use MPNN-based 3D models. This is another example of a trend that is different for CoV-2 than for the other targets. It therefore reinforces the need for further investigation into tasks without a well-defined protein target.

It is also possible that Bayesian hyperparameter optimization did not converge to the global minimum for each model. Indeed, it is somewhat counter-intuitive that the hyperparameters differ so significantly among datasets for a given model. This might be related to the amount of data, since learning curves can be very noisy for small datasets, which can in turn affect the optimization outcome. It may therefore be of interest to improve hyperparameter optimization in future work. Noise can be reduced by using a moving average of the target metric over several epochs. It can further be reduced by re-training with several random seeds for each hyperparameter combination, and then averaging the results.
5.2. Computational cost

The 4D attention-based models are quite computationally demanding. 3D models are generally more expensive than 2D models, as they incorporate non-bonded edges in addition to bonded ones. More importantly, conformer-based models are \( n_{\text{conf}} \) times more expensive than single conformer models, where \( n_{\text{conf}} \) is the average number of conformers. Here we used a maximum of 200 conformers per species, which corresponds to 76 on average. Hence our models are hundreds of times slower than 2D models, and our training on the CoV 3CL dataset took several days on 32 GPUs.

It is not beneficial to use conformer models in their current state, as they are expensive and can hurt performance. However, given the sheer volume of extra information contained in conformers, it is helpful to think of other ways to use the ensemble. Let us consider how model complexity can be reduced, as this would reduce computational cost and could also improve performance. A simple method is to bin similar conformers together, so that the number of effective conformers is reduced. The statistical weight of each conformer would be the sum of its constituent weights. This would lead to models that act over far fewer conformers for each species, thereby reducing complexity and cost. One might also consider expanding upon the average-distance models introduced here, which come at the cost of single-conformer models but contain ensemble information. Performance could be improved by using pre-set weights that reflect binding affinity rather than statistical weight. These could perhaps be found with computational docking or with attention models acting over non-learnable 3D fingerprints, which are far less expensive. One might also use the docking results to simply select the best single conformer for each species, and train a 3D model on one conformer only for each species.

6. Conclusion

We have introduced a new method for predicting properties from conformer ensembles, and introduced several new 3D-based models. In some cases we found that 3D models with a single conformer achieved significant improvement over 2D models, but in other cases had comparable or even worse performance. We also found that multiple conformers did not improve the predictions. With access to the GEOM dataset and the models introduced in this work, the community will be able to use our models and refine our conformer approach to improve virtual screening.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: doi.org/10.7910/DVN/N4VLQL.

Acknowledgments

The authors thank the XSEDE COVID-19 HPC Consortium, Project CHE200039, for compute time. NASA Advanced Supercomputing (NAS) Division and LBNL National Energy Research Scientific Computing Center (NERSC), MIT Engaging cluster, Harvard Cannon cluster, and MIT Lincoln Lab Supercloud clusters are gratefully acknowledged for computational resources and support. The authors also thank Christopher E Henze (NASA) and Shane Canon and Laurie Stephey (NERSC) for technical discussions and computational support, MIT AI Cures (www.aicures.mit.edu/) for molecular datasets and Wujie Wang, Daniel Schwalbe Koda, Shi Jun Ang (MIT DMSE) for scientific discussions and access to computer code. Financial support from DARPA (Award HR00111920025) and MIT-IBM Watson AI Lab is acknowledged.

ORCID iD

Simon Axelrod  
https://orcid.org/0000-0003-2936-0133

References

[1] Kaitin K I 2010 Deconstructing the drug development process: the new face of innovation Clin. Pharmacol. Therap. 87 356–61
[2] DiMasi J A, Grabowski H G and Hansen R W 2016 Innovation in the pharmaceutical industry: new estimates of R&D costs J. Health Econ. 47 20–33
[3] Brown D G, May-Dracka T L, Gagnon M M and Tommasi R 2014 Trends and exceptions of physical properties on antibacterial activity for Gram-positive and Gram-negative pathogens J. Med. Chem. 57 10144–61
[4] Trott O and Olson A J 2010 AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading J. Comput. Chem. 31 455–61
[5] Honglin L et al 2006 TarFisDock: a web server for identifying drug targets with docking approach Nucleic Acids Res. 34 W219–24
[6] Alonso H, Bliznyuk A A and Gready J E 2006 Combining docking and molecular dynamic simulations in drug design Med. Res. Rev. 26 531–68
[7] Burbidge R, Matthew Trotter B B and Holden S 2001 Drug design by machine learning: support vector machines for pharmaceutical data analysis Comput. Chem. 25 5–14
[8] Vamathevan J et al 2019 Applications of machine learning in drug discovery and development Nat. Rev. Drug Discovery 18 463–77
[9] Stokes J M et al 2020 A deep learning approach to antibiotic discovery Cell 180 688–702
[10] Shen C, Ding J, Wang Z, Cao D, Ding X and Hou T 2020 From machine learning to deep learning: advances in scoring functions for protein–ligand docking Wiley Interdiscip. Rev.: Comput. Mol. Sci. 10 e1429
[11] Bartók A P, Kondor R and Csányi G 2013 On representing chemical environments Phys. Rev. B 87 184115
[12] Duvenaud D K, Maclaurin D, Aguilera-Iparraguirre J, Romero-Paz J, Hirzel T, Aspuru-Guzik A and Adams R P 2015 Convolutional networks on graphs for learning molecular fingerprints Advances in Neural Information Processing Systems pp 2215–23
[13] Yuja L, Tarlov D, Brockschmidt M, and Zemel R 2015 Gated graph sequence neural networks (arXiv:1511.05943)
[14] Battaglia P et al 2016 Interaction networks for learning about objects, relations and physics Advances in Neural Information Processing Systems vol 29 pp 4502–10
[15] Kearnes S, McCloskey K, Berndl M, Pande V and Riley P 2016 Molecular graph convolutions: moving beyond fingerprints J. Comput.-Aid. Mol. Design 30 595–608
[16] Schütt K T, Arbabzadah F, Chmiela S, Müller K R and Tkatchenko A 2017 Quantum-chemical insights from deep tensor neural networks Mach. Learn.: Sci. Technol. 4 41
[17] Gieleciak R and Polanski J 2007 Modeling robust QSAR. 2. Iterative variable elimination schemes for CoMSA: application for a series of cholecystokinin-A receptor antagonists, determined by molecular docking, scoring functions and 3D-QSAR methods J. Comput.-Aid. Mol. Design 21 795–802
[18] machine learning for 3D-QSAR studies of antifungal p450 inhibitors and 3D-pharmacophore comparisons as a function of alignment J. Chem. Inf. Comput. Sci. 43 2170–9
[83] Chemical Computing Group ULC. 2018 CCG: molecular operating environment (MOE) (available at: www.chemcomp.com/)

[84] Hawkins P C D, Geoffrey Skillman A, Warren G L, Ellenson B A and Stahl M T. 2010 Conformer generation with OMEGA: algorithm and validation using high quality structures from the Protein Databank and Cambridge Structural Database. J. Chem. Inf. Model. 50, 572–84

[85] Chan L, Hutchison G R and Morris G M. 2019 Bayesian optimization for conformer generation. J. Cheminf. 11, 1–11

[86] Phyo Kyaw Zin P, Borrel A and Foursches D. 2020 Benchmarking 2D/3D/MD-QSAR models for imatinib derivatives: How far can we predict? J. Chem. Inf. Model. 60, 3542–60

[87] Axelrod S and Gomez-Bombarelli R. 2021 GEOM (available at: https://doi.org/10.7910/DVN/JNGTDF)

[88] Hawkins P C D, Geoffrey Skillman A, Warren G L, Ellenson B A and Stahl M T. 2010 Conformer generation with OMEGA: algorithm and validation using high quality structures from the Protein Databank and Cambridge Structural Database. J. Chem. Inf. Model. 50, 572–84

[89] Ramakrishnan R, Dral P O, Rupp M and von Lilienfeld O A. 2014 Quantum chemistry structures and properties of 134 kilo molecules. Sci. Data 1, 140022

[90] Ellinger B et al. 2020 Identification of inhibitors of SARS-CoV-2 in-vitro cellular toxicity in human (Caco-2) cells using a large scale drug repurposing collection (https://doi.org/10.21203/rs.3.rs-23951/v1)

[91] Touret F, Gilles M, Barral K, Nougarede A, Decroly E, de Lamberlere X and Coutard B. 2020 In vitro screening of a FDA approved chemical library reveals potential inhibitors of SARS-CoV-2 replication. Scientific Reports 10, 13093

[92] Diamond Light Source. 2020 Main protease structure and XChem fragment screen (available at: www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem.html)

[93] Engel D. 2010 qHTS of Yeast-based Assay for SARS-CoV PLP (available at: https://pubchem.ncbi.nlm.nih.gov/bioassay/485353)

[94] Zampieri M, Zimmermann M, Claassen M and Sauer U. 2017 Nontargeted metabolomics reveals the multilevel response to antibiotic perturbations. Cell Rep. 19, 1214–28

[95] AICures 2020 (available at: www.aicures.mit.edu/data)

[96] Zhijian X, Peng C, Shi Y, Zhu Z, Kaijie M, Wang X and Zhu W. 2020 Nelfinavir was predicted to be a potential inhibitor of 2019-nCoV main protease by an integrative approach combining homology modelling, molecular docking and binding free energy calculation (bioRxiv preprint) (posted online 28 January 2020) (https://doi.org/10.1101/2020.01.27.921627)

[97] Jain A N and Nicholls A. 2008 Recommendations for evaluation of computational methods. J. Comput.-Aid. Mol. Design 22, 133–9

[98] Pearce N M et al. 2017 A multi-crystal method for extracting obscured crystallographic states from conventionally uninterpretable electron density. Nat. Commun. 8, 15123

[99] Pappu A and Paige B. 2020 Making graph neural networks worth it for low-data molecular machine learning (arXiv:2011.12203)

[100] Axsen S D, Huang X-P, Cáceres E L, Gendelev L, Roth B L and Keiser M J. 2017 A simple representation of three-dimensional molecular structure. J. Med. Chem. 60, 7393–409