Learning Protein-Ligand Binding Affinity with Atomic Environment Vectors

Rocco Meli,† Andrew Anighoro,‡ Mike J. Bodkin,‡ Garrett M. Morris,∗¶ and Philip C. Biggin∗†

†Department of Biochemistry, University of Oxford, South Parks Rd., Oxford, OX1 3QU, UK.
‡Evotec (UK) Ltd., 114 Innovation Drive, Milton Park, Abingdon, Oxfordshire OX14 4RZ, UK
¶Department of Statistics, University of Oxford, 24-29 St. Giles’, Oxford, OX1 3LB, UK.

E-mail: garrett.morris@dtc.ox.ac.uk; philip.biggin@bioch.ox.ac.uk
Abstract

Scoring functions for the prediction of protein-ligand binding affinity have seen renewed interest in recent years when novel machine learning and deep learning methods started to consistently outperform classical scoring functions. Here we explore the use of atomic environment vectors (AEVs) and feed-forward neural networks, the building blocks of several neural network potentials, for the prediction of protein-ligand binding affinity. The AEV-based scoring function, which we term AEScore, is shown to perform as well or better than other state-of-the-art scoring functions on binding affinity prediction, with an RMSE of 1.22 pK units and a Pearson’s correlation coefficient of 0.83 for the CASF-2016 benchmark. However, AEScore does not perform as well in docking and virtual screening tasks. We therefore show that the model can be combined with the classical scoring function AutoDock Vina in the context of Δ-learning, where corrections to the AutoDock Vina scoring function are learned instead of the protein-ligand binding affinity itself. Combined with AutoDock Vina, Δ-AEScore has an RMSE of 1.32 pK units and a Pearson’s correlation coefficient of 0.80 on the CASF-2016 benchmark, while retaining the good docking and screening power of the underlying classical scoring function.
Introduction

Structure-based drug discovery exploits knowledge of protein structures in order to design novel and potent compounds for a specific target. Protein-ligand docking is one of the main computational tools employed in the early stages of structure-based drug discovery—where more accurate methods, such as free energy calculations, are too time consuming—to predict the binding mode and binding affinity of different ligands in a binding site. The binding mode search is usually guided by a scoring function. Sometimes the scoring function has the dual purposes of finding the binding poses (docking) and predicting the protein-ligand binding affinity (scoring), whilst at other times different scoring functions are used for different purposes (scoring, ranking, docking, or screening).

Scoring functions can be loosely assigned to four classes: physics-based, regression-based, knowledge-based or machine-learning based. Many scoring functions belonging to the first three categories have been developed over the past decades. Despite their successes in reproducing the binding pose, a rapid and accurate prediction of the protein-ligand binding affinity remains a very challenging task. In recent years, machine learning and deep learning scoring functions have consistently improved protein-ligand binding affinity predictions. These improvements build on decades of quantitative structure-activity relationship (QSAR) modelling, where simpler representations and regressors were used. Deep learning architectures—which are outperforming standard algorithms in image recognition and natural language processing—are under active research, as demonstrated by the large number of new scoring functions based on deep learning.

In this work we explore the use of a collection of feed-forward neural networks (NNs), each computing an atomic contribution to the protein-ligand binding affinity. We show this architecture, combined with atom-centred symmetry functions (ACSFs) to capture the local chemical environment of every atom in the protein-ligand binding site, performs as well as or better than current machine learning and deep learning architectures. This particular representation—commonly employed in the development of neural-network potentials
(NNPs)\cite{26,27}—has the advantage of being translationally and rotationally invariant, unlike NN-based or CNN-based scoring functions that use an order-dependent input vector or a grid-based representations as input.

**Methods**

**Atomic Environment Vectors**

In order to predict the binding affinity of a ligand to a target of interest we need a description of the protein-ligand binding site that allows the key protein-ligand interactions to be learned. Ideally, this representation should depend only on the relative positions of the ligand and the protein—the representation should be invariant under translation, rotation and mirror operations. Unfortunately, some machine learning and especially deep learning scoring functions employed in computational drug discovery, do not satisfy such conditions: grid-based methods are not translationally or rotationally invariant and need extensive data augmentation,\cite{20} while vector-based representations are often order-dependent.

Local representations of the atomic environment satisfying the ideal properties outlined above have been employed with success in quantum machine learning.\cite{26,28,30} In particular, the ACSFs originally introduced by Behler and Parrinello and further developed to build the Accurate NeurAl networK engINe for Molecular Energies (ANAKIN-ME or “ANI” for short) family of NNPs have been successful in producing accurate molecular properties.\cite{26,27,31,32}

Here we employ the ACSFs defined for the ANI family of NNPs in order to represent the protein-ligand binding site, where protein residues with at least one atom within a distance, \(d\), from the ligand are considered.

For each atom \(i\) of element \(X\) in the system, its chemical environment can be represented by combining radial \((G^R_{\alpha,m})\) and angular \((G^A_{\alpha,\beta,m})\) ACSFs in a one dimensional vector, \(G^X_i = \{G^R_{\alpha,m_1}, \ldots, G^A_{\alpha_1,\beta_1,m_1}, \ldots\}\)—called the atomic environment vector \(\text{(AEV)}\). \(X\) corresponds to the element of the atom for which the AEV is being computed, while \(\alpha\) and \(\beta\) denote the
elements of the neighbours within a cutoff radius, $R_c$. The ACSFs capture the atom’s radial and angular chemical environment, and their locality is ensured by a cutoff function:

$$f_c(R_{ij}) = \begin{cases} \frac{1}{2} \left[ \cos \left( \frac{\pi R_{ij}}{R_c} \right) + 1 \right] & R_{ij} \leq R_c \\ 0 & R_{ij} > R_c \end{cases}$$

Radial symmetry functions are given by:

$$G^{R}_{\alpha,m} = \sum_{j \neq i \atop j \in \alpha} e^{-\eta_R (R_{ij} - R_s)^2} f_c(R_{ij})$$

where the index $m$ runs over the set of parameters $\{R_s\}$ and the summation over $j$ runs over all the atoms of element $\alpha$; $\eta_R$ controls the width of the radial Gaussian distributions, while $R_s$ controls their radial shift. The angular symmetry function is defined as:

$$G^{A}_{\alpha,\beta,m} = 2^{1-\zeta} \sum_{j,k \neq i \atop j \in \alpha, k \in \beta} [1 + \cos (\theta_{ijk} - \theta_s)]^{\zeta} e^{-\eta_A \left( \frac{R_{ij} + R_{ik}}{2} - R_s \right)^2} f_c(R_{ij}) f_c(R_{ik})$$

where the index $m$ runs over the set of parameters $\{\{R_s\}, \{\theta_s\}\}$ and the summation runs over pairs of atoms of elements $\alpha$ and $\beta$; $\eta_A$ and $R_s$ have the same role of $\eta_R$ and $R_s$ in the radial symmetry function described above, with $\theta_s$ capturing different regions of the angular environment, while $\zeta$ controls the width of the peaks of the ACSF in the angular environment.

The AEV $G^X_i$ of atom $i$ of element $X$—composed of different ACSFs in a single vector—encodes the neighbour-dependent local atomic environment of atom $i$ of element $X$. This corresponds essentially to a fine-grained and flexible atom typing, in contrast to the static and arbitrary atom types employed in standard scoring functions.

Figure 1 shows schematically the components of an AEV for an atom in a system composed only of the elements H, C and O. By construction, this vector is translationally and
rotationally invariant as well as invariant under the exchange of two atoms of the same element.

Figure 1: AEV constructed using ACSFs\textsuperscript{26,27} (with $R_s = 0$ and $\{\theta_s\} = \{0, \pi\}$ for angular symmetry functions) for an atom in a system composed only of the elements H, C and O. The radial and angular symmetry functions, $G_{R,\alpha,m}^\alpha$ and $G_{A,\alpha,\beta,m}^\alpha$, respectively, are given for the elements $\alpha$ and $\beta$, and iterate over $m$. Loosely adapted from Gao et al.\textsuperscript{33}.

In order to keep the size of the AEVs reasonably small, we restrict the parameters of $G_{A,\alpha,\beta,m}^\alpha$ to those of the original Behler-Parrinello formulation: $\{\theta_s\} = \{0, \pi\}$ and $R_s = 0$. All other parameters are the same as those employed in the ANI-1x NNP\textsuperscript{27} which results in an AEV size of 200 (for each atom). AEVs are built using the AEVComputer as implemented in TorchANI 2.1\textsuperscript{33}.

Neural Network

The NN architecture is implemented using PyTorch 1.5,\textsuperscript{34} loosely following the original work of Behler and Parrinello, the ANI family of NNPs, and the TorchANI implementation.\textsuperscript{26,27,33} It consists of $n_e$ atomic neural networks, where $n_e$ is the number of elements in the dataset. The atomic NNs are standard feed-forward NNs with rectified linear unit (ReLU) activation functions and dropout layers. The outputs of the atomic NNs are then summed together in order to obtain the final estimate of the binding affinity.

Figure 2 shows a schematic representation of the model for a hypothetical system com-
posed of two hydrogen atoms, one carbon atom and one oxygen atom. The AEVs $G^X_i$ corresponding to atoms of the same element $X$ are propagated through the same atomic NNs (with the same weights). All atomic contributions are summed together in order to get the final prediction.

Figure 2: Propagation of AEVs, $G^X_i$, through atomic NNs for the four atoms of a hypothetical system composed of two hydrogen atoms, one carbon atom and one oxygen atom. The AEVs, $G^X_i$, are constructed for each atom $i$ of element $X$ as described in the main text and propagated through the atomic NN of the corresponding element. All atomic contributions are finally summed together to obtain the $pK$ prediction. Loosely adapted from Smith et al.27.

The idea behind the decomposition of the binding affinity into atomic contributions is essentially the one that has been proven useful for short-range energy decomposition in NNPs. The negative logarithm of the binding affinity $pK = -\log_{10}(K/c_0)$ is proportional to the Gibbs free energy of binding

$$pK = -\frac{1}{\ln(10)} \frac{\Delta G_{\text{bind}}^0}{RT}$$

and therefore decomposing $pK$ into atomic contributions corresponds to a decomposition of the Gibbs free energy. As for the total energy in NNPs, this decomposition allows the
description of local contributions only,\textsuperscript{29} but it is very effective in practice—as demonstrated by the success of NNPs in fitting high-dimensional potential energy surfaces.\textsuperscript{26,27,30,32,35} This decomposition also appears to be very effective in generalisation and transferability, since it works for systems much larger than the ones included in the training set.\textsuperscript{27}

**Training and Test Datasets**

The PDBbind dataset provides protein-ligand complexes with associated experimentally determined inhibition constants, $K_i$, dissociation constants, $K_d$, and IC$_{50}$ measurements (in decreasing order of preference).\textsuperscript{36,37} This dataset is divided in two parts: the PDBbind Refined set and the PDBbind General superset. The Refined set only contains high-quality structures with associated $K_i$ or $K_d$ values, while the General set also includes structures with associated IC$_{50}$ values. A curated subset of the PDBbind Refined set is provided for comparative assessment of scoring functions (CASF).\textsuperscript{38,39}

In this work the PDBbind 2016 Refined set is used for training and validation while the CASF-2013 and CASF-2016 data sets are used for testing and comparison with other machine learning and deep learning models, as well as classical scoring functions.\textsuperscript{36–39} The PDBbind 2016 Refined set is randomly split into training and validation sets with a 90/10 ratio. Systems present in both PDBbind and CASF datasets are removed from the training and validation sets and used only for testing. This procedure ensures that there is no exact overlap (“hard overlap”) of protein-ligand complexes between the PDBbind (training/validation) and CASF (test) datasets, although some overlap with similar targets and ligands remains.\textsuperscript{38–40} In order to assess this remaining “soft overlap” between training and test sets—arising from similar proteins, similar binding sites, and similar ligands—we use the subset of the PDBbind 2016 dataset proposed by Su et al.\textsuperscript{40}

A detailed analysis of the CASF-2013 and CASF-2016 test sets—including the distribution of the protein-ligand binding constants and of some key properties of the protein-ligand complexes—is reported by Li et al.\textsuperscript{38} (CASF-2013) and Su et al.\textsuperscript{39} (CASF-2016). In partic-
ular, the CASF-2016 dataset is composed of 57 protein classes—with at least 90% sequence similarity—each containing 5 protein-ligand complexes. The CASF-2013 dataset is smaller in size, with 65 protein classes each containing 3 protein-ligand complexes.

Ligand SDF or MOL2 files from the datasets were either converted to PDB files using OpenBabel and parsed using MDAnalysis (for scoring and ranking) or parsed directly with OpenBabel’s Python bindings (docking and screening). Protein PDB files were discarded when the element column was absent or could not be parsed correctly by MDAnalysis (this never occurred for the test set). All water molecules were removed from the dataset. All the systems in the PDBbind and CASF dataset were automatically protonated using OpenBabel, and given the size of the dataset the protonation state was not further assessed.

The complexity of the NN model grows quickly with the number of atomic species present in the dataset since every element requires its own atomic NN. For this reason, we adopted two different strategies: selecting only protein and ligand atoms, or mapping metal centers to a dummy atom. Additionally, we removed the few selenoproteins present in the training or validation sets. When selecting only protein and ligand atoms, the following elements remained (in order of abundance for the ligands, see Fig. S1): H, C, O, N, S, P, F, Cl, Br, I. This resulted in a total 10 atomic NNs, one for each element. When metal centers were kept (see Fig. S2), all atoms outside of the previous list were mapped to a dummy element, X.

When “hard overlaps” with CASF-2016 were removed, the final training set consisted of 3377 complexes while the validation set consisted of 376 complexes. When “hard overlaps” with CASF-2013 were removed, the final training set consisted of 3464 complexes while the validation set consisted of 385 complexes. The CASF test sets are left unchanged.

Protein-ligand complexes 4O3C and 4IGT were removed from the PDBbind Refined Set since they contain lithium, which is not supported by AutoDock Vina, the classical scoring function used as baseline in this work.

The advantage of mapping metal centers to a dummy atom is that metalloproteins,
which are notoriously difficult to treat with docking and classical molecular dynamics, are supported by our method. However, our treatment has the drawback of considering all metal atoms as equivalent, irrespective of their coordination number. As more experimental data on metalloproteins becomes available, more elements could be added into the model (with an increased computational cost).

**Δ-learning**

Δ-learning is a powerful machine learning approach where the model is trained to predict the corrections to a baseline towards the target value, instead of predicting the target value itself. This approach has been applied successfully to the prediction of molecular properties from quantum mechanical calculations as well as for binding affinity predictions. In the context of docking scoring functions, a Δ-learning approach has the advantage of retaining the good docking power of traditional methods while significantly improving the scoring function.

In this work we explored the use of a Δ-learning approach in combination with the AutoDock Vina scoring function. The Δ-AEScore scoring function is therefore given by:

\[ \Delta\text{-AEScore} = S + \Delta \]

where \( S \) is the standard AutoDock Vina score (in \( pK \) units) and \( \Delta \) is the learned correction.

**Consensus Scoring**

In order to compensate for the variability introduced by random weights initialization and the stochastic optimization, we investigated the use of consensus scoring in order to evaluate our models. Consensus scoring has been shown, in some cases, to improve performance across targets in structure-based virtual screening.

During training, a total of five models were randomly initialized and independently
trained. Final predictions were obtained as the average protein-ligand binding affinity of the models. This technique also allows the computation of the standard deviation associated with each prediction. The benefits of consensus scoring are analysed retrospectively below.

Software

Our implementation is based on open source software from the Python ecosystem. This includes: TorchANI 2.1, PyTorch 1.5, MDAnalysis 2.0-dev, OpenBabel 3.1, NumPy 1.19, SciPy 1.5, pandas 1.1, Matplotlib 3.3, seaborn 0.11, scikit-learn 0.23, pytest 6.0 and many other packages.

Results

AESC score

Hyper Parameters Optimization

The hyper parameters of our model—the number and size of layers in the elemental NNs, dropout probability, batch size, and protein-ligand distance, \( d \)—were optimized with a grid-based method and manually fine-tuned in order to maximize the Pearson’s correlation coefficient between the predicted and experimental binding affinities on the validation set.

We found that a protein-ligand distance \( d = 3.5 \) Å and 256-128-64-1 feed-forward NNs performed best, when combined with a batch size of 64 and a dropout probability of 25%.

Table S1 shows the performance of the model—with consensus scoring—on the validation test for different values of \( d \). Using a distance of \( d = 4.0 \) Å does not change the performance, compared to \( d = 3.5 \) Å. However, the larger number of protein atoms causes the computational time to be increased. Visual inspection of a selection of systems showed that the \( d = 3.5 \) Å selects the important residues in the binding site.
The model’s weights are optimized using the ADAM optimizer with a learning rate of \(1 \times 10^{-4}\) and using PyTorch’s default parameters, \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\).

Dropout layers are usually not employed in NNPs, but our hyperparameter search shows that they increase the performance of our model by decreasing overfitting on the training set, thus improving transferability.

**Scoring Power**

The scoring power of a scoring function measures the linear correlation between predicted and experimental binding affinities and it is usually quantified by the Pearson’s correlation coefficient:

\[
    r = \frac{\sum_i (\hat{y}_i - \langle \hat{y} \rangle)(y_i - \langle y \rangle)}{\sqrt{\sum_i (\hat{y}_i - \langle \hat{y} \rangle)^2} \sqrt{\sum_i (y_i - \langle y \rangle)^2}}
\]

where \(y\) denotes experimental values, \(\hat{y}\) denotes predicted values, and \(\langle \cdot \rangle\) denotes the average over all experimental or predicted values.

Figure 3 shows the predictions of our model versus the experimental values of the binding affinity for the CASF-2013 and CASF-2016 benchmark data sets—when only protein and ligand atoms are considered. Our model achieves an RMSE of 1.30 pK units and a Pearson’s correlation coefficient of 0.80 on the CASF-2016 test set, and an RMSE of 1.46 pK units and a Pearson’s correlation coefficient of 0.76 on the CASF-2013 test set. Error bars show the standard deviation of the predictions obtained with consensus scoring (average over five independently trained models).

Confidence intervals (CIs) for the correlation coefficient can be obtained by bootstrapping (with 10000 bootstrap replicates), as described in the CASF evaluation. The 90% CI for the Pearson’s correlation coefficient for the CASF-2016 test set is \([0.76, 0.83]_{CI \ 90\%}\), while for the CASF-2013 test set it is \([0.68, 0.81]_{CI \ 90\%}\).

Figure 4 shows a breakdown of the Pearson’s correlation coefficient (and the RMSE) for each protein class in the CASF-2016 benchmark data set. We see that the performance of AEScore is class-dependent and there is no clear correlation between the Pearson’s correla-
Figure 3: Predicted versus experimental binding affinity, expressed as pK for AEScore, when only protein and ligand atoms are retained.

The correlation coefficient and the RMSE (by comparing class #1 and class #55, for example). For the majority of targets, the predicted binding affinity is well correlated with the corresponding experimental value. Only a few classes have a low correlation coefficient and two classes show negative correlation. The classes with negative correlation are (refer to the supplementary information of Su et al. for the full list of classes): β-lactoglobulin (class 13) and queuine tRNA-ribosyltransferase (class 40). The average and median Pearson’s correlation coefficients across all target classes are 0.67 and 0.82, respectively.

Consensus Scoring

In the previous section we employed consensus scoring—with five independently trained models—since this has previously been shown to improve performance. A small performance boost is also obtained in our case, as it can be verified retrospectively.

If we consider the CASF-2016 dataset, the average correlation coefficient of the five independent models is 0.77 (minimum 0.77, maximum 0.78) while consensus scoring reaches
Figure 4: Per-class Pearson’s correlation coefficient, with each bar color-coded by the corresponding RMSE in pK units, for the 57 classes of the CASF-2016 dataset.

0.80—better than the best-performing individual model amongst the five. The same observation is true for the RMSE on the same test set. The average RMSE is 1.38 pK units (minimum 1.35, maximum 1.42) while the consensus scoring has a RMSE of 1.30 pK units—which is lower than the best-performing model amongst the five.

Implicit Hydrogen Atoms

In order to assess the impact of automatic protonation using OpenBabel\textsuperscript{153} we also trained AEScore without hydrogen atoms for both the protein and the ligand. This results in the removal of one atomic NN, thus decreasing the number of parameters in the model.

Training the model without hydrogen atoms does not seem to consistently affect the performance of our model: we observe a small decrease in performance with the CASF-2013 test set and a small gain with the CASF-2016 test set. For the CASF-2013 test set, we obtain a Pearson’s correlation coefficient of $0.75 \in [0.69, 0.80]_{90\%}$ and an RMSE of 1.48 pK units while for the CASF-2016 test set we obtain a Pearson’s correlation coefficient of $0.81 \in [0.77, 0.84]_{90\%}$ and an RMSE of 1.28 pK units.

Per-class Pearson’s correlation coefficient (and RMSE) for the CASF-2016 test set for the model trained without hydrogen atoms are shown in Fig. S3. Again, there is no clear
relationship between Pearson’s correlation coefficient and RMSE. In this case, the average Pearson’s correlation coefficient is 0.69 while the median is 0.85.

**Metalloproteins**

When metal centers are included, they are mapped to a dummy element X. As we can see from Fig. S2, Zn is the most abundant metal center in our dataset (545 systems), followed by Mg (142 systems). All other metal centers appear in fewer than 60 systems.

With the metal centers mapped to a dummy element X (when they are not already present in either the protein or the ligand), we obtain a Pearson’s correlation coefficient of $0.80 \in [0.76, 0.83]_{CI \, 90\%}$ and an RMSE of 1.31 pK units on the CASF-2016 benchmark. When hydrogen atoms are removed, we find a Pearson’s correlation coefficient of $0.81 \in [0.77, 0.84]_{CI \, 90\%}$ and a RMSE of 1.31 pK units.

**Similarity between training and test sets**

As mentioned above, we removed the systems appearing in the CASF-2016 and CASF-2013 benchmark datasets from the training sets (removing the so-called “hard overlap”). However, some “soft overlap”—arising from similar proteins, similar binding sites, and similar ligands—between the training and test sets remains and could therefore artificially inflate the results. This is a known problem as shown by Boyles et al.⁴ and, more recently, by Su et al.⁴⁰ who both proposed non-redundant subsets of the PDBbind refined set with decreasing similarity with respect to the CASF-2016 test set. Such non-redundant datasets allow to assess how scoring functions behave when the “soft overlap” between the training and test sets is incrementally reduced.

In the work of Su et al.⁴⁰ the similarity between the training and test sets is measured by three metrics: similarity between protein sequences, similarity between ligand shapes and similarity between binding pockets. If two protein-ligand complexes—one in the training set, the other in the test set—have all three similarity metrics above a given threshold they
are considered redundant. All redundant complexes are removed from the training set with an iterative procedure until the remaining complexes form a representative, non-redundant training set for the given similarity threshold.

Fig. 5 shows the performance of our model on the CASF-2016 dataset when trained on the non-redundant training sets proposed by Su et al., with different similarity thresholds (“None” indicates that only the “hard overlap” between training and test sets is removed). We see that as the overlap threshold between the training and test sets increases, the performance of our model also increases. Interestingly, a similarity threshold of 95% does not negatively affect our scoring function, in contrast with other machine learning scoring functions. This trend is similar to the RF model of Su et al., which is consistently outperformed by our model. Other machine learning scoring functions evaluated by Su et al. are effectively negatively affected by removing structurally redundant samples already at high thresholds.

We also found that the model with a similarity threshold of 95% (denoted AEScore$_{95}$ hereafter) seems to perform slightly better than the model trained by only removing the “hard overlap”. This could be attributed to the removal of some inconsistencies in the training set, introduced by experimental errors, or simply by the variability of the training procedure (minibatches, dropouts, etc.). The AEScore$_{95}$ model is our best performing model on the CASF-2016 test set (Pearson’s correlation coefficient of 0.83 ∈ $[0.79, 0.86]$ CI 90%, RMSE of 1.22 pK$_{a}$ units) and it performs very well compared to other state-of-the-art scoring functions (see discussion of Figure 10)—although differences with other top-performing methods might not be statistically significant.

**Ranking Power**

The ranking power of a scoring function measures its ability to rank different ligands—in a given binding pose—according to their binding affinity against a particular target. The ranking power is usually measured by three quantities: Spearman’s (rank-)correlation...
Figure 5: Scoring power of AEScore (with and without hydrogen atoms) as a function of the similarity threshold between the training and test sets, as defined by Su et al. The raw data for the RF and DT scoring functions was kindly provided by Su et al. upon request. RF and DT are respectively the best and worst performing models (at the 95% similarity threshold) presented in Su et al. and are consistently outperformed by AEScore.
coefficient, Kendall’s (rank-)correlation coefficient and the predictive index (PI).

Our scoring function AEScore has an average Spearman’s correlation coefficient of $0.64 \in [0.54, 0.71]_{CI \, 90\%}$. This is similar to the best classical scoring function evaluated in the CASF-2016, although it is within the 90% confidence interval. The same observation remains true for the average Kendall’s correlation coefficient of $0.55 \in [0.47, 0.62]_{CI \, 90\%}$ and for the PI of $0.67 \in [0.58, 0.73]_{CI \, 90\%}$.

Interestingly, if hydrogen atoms are removed the ranking power does not change. When hydrogen atoms are ignored, the Spearman’s correlation coefficient becomes $0.63 \in [0.54, 0.71]_{CI \, 90\%}$, the Kendall’s correlation coefficient becomes $0.56 \in [0.48, 0.63]_{CI \, 90\%}$, and the PI becomes $0.66 \in [0.57, 0.74]_{CI \, 90\%}$.

Figure 6 shows the per-class Spearman’s rank-correlation coefficient, while the per-class Kendall’s correlation coefficient is reported in Figure S4. For the Spearman’s correlation coefficient we now have four classes with negative correlation. Classes 13 and 40 ($\beta$-lactoglobulin and queuine tRNA-ribosyltransferase, respectively) also had a negative Pearson’s correlation coefficient, while classes 5 (alpha-L-fucosidase) and 51 (transporter) did not. For the Kendall’s correlation coefficient we have only three classes with negative correlation: classes 13, 40, and 51. A few other classes have no correlation.

Figure 6: Per-class Spearman’s correlation coefficient, with each bar color-coded by the corresponding RMSE in p$K$ units, for the 57 classes of the CASF-2016 dataset.
Docking Power

AEScore has been developed with the intent of predicting the binding affinity of a given protein-ligand complex. However, scoring functions can also be used to determine correct binding poses. Therefore we evaluate the docking power of AEScore using the docking decoys provided in CASF-2016 dataset. If we consider a correct binding pose as one with a root mean squared deviation (RMSD) from the crystallographic binding mode that is smaller than 2 Å, we can define the docking success rate as the percentage of targets with a good pose ranked amongst the top one, top two or top three poses.

AEScore has a success rate of $35.8\% \in [30.9\%, 40.4\%]_{90\%\, CI}$ for the top one pose, a success rate of $54.4\% \in [48.8\%, 58.6\%]_{90\%\, CI}$ for the top two poses and a success rate of $60.4\% \in [54.7\%, 64.2\%]_{90\%\, CI}$ for the top three poses. Such low success rates are comparable with the worst classical scoring functions evaluated on the CASF-2016 benchmark. This low success rate is also observed with other deep learning scoring functions: a recent preprint study presenting a CNN-based scoring function, AK-score, reports a top one success rate of $34.9\%$ (single) or $36.0\%$ (ensemble).

These results are not surprising, since AEScore has been trained in order to predict the experimental binding affinity given a protein-ligand complex and has therefore never been exposed to high-RMSD binding poses (decoys). In order to use the scoring function to determine low-RMSD poses one has to train for such task. One way to train a scoring function for docking is to train a pose classifier (distinguishing low RMSD poses from high RMSD poses), but this requires a change in the model architecture. Another way to tailor a machine learning scoring function for docking is to train on docking scores as done for AGL-Score, but this is not particularly interesting. A third way to improve binding affinity predictions while retaining the good docking and screening power of some classical scoring functions is to use Δ-learning. In this work we explore the latter approach.
**Δ-AEScore**

**Δ-learning with AutoDock Vina**

The use of AEVs combined with a collection of feed-forward NNs has proven successful to predict protein-ligand binding affinities on the CASF-2013 and CASF-2016 benchmark datasets using exclusively elements and atomic coordinates, as demonstrated above. Unfortunately, the results of the docking power test were unexpectedly deceiving. However, it has been previously demonstrated that a Δ-learning approach can retain the good screening power of a scoring function while improving the performance in the docking and screening power tests.

In the Δ-learning approach, a classical scoring function is used to obtain a crude prediction of the binding affinity, which is subsequently corrected with a machine learning or deep learning scoring function. If corrections to the AutoDock Vina scoring function can be learned by our model, combining such corrections with the docking power of AutoDock Vina would provide a scoring function with both good scoring and docking powers.

In order to combine AutoDock Vina and the experimental data of PDBbind, AutoDock Vina scores, $S$, are converted to $pK$ values using

$$pK = -\log_{10}(e^{\frac{S}{RT}}),$$

where $T = 295\, \text{K}$ and $R$ is the ideal gas constant.

**Scoring Power**

Figure 7 shows the predictions of our model versus the experimental values of the binding affinity for the CASF-2013 and CASF-2016 benchmark data sets. Δ-AEScore achieves an RMSE of 1.53 $pK$ units and a Pearson’s correlation coefficient of $0.74 \in [0.67, 0.78]_{CI\, 90\%}$ on the CASF-2013 test set and an RMSE of 1.34 $pK$ units and a Pearson’s correlation coefficient of $0.79 \in [0.75, 0.82]_{CI\, 90\%}$ on the CASF-2016 test set. The performance is slightly worse than
that of AEScore, indicating that corrections to the AutoDock Vina native scoring function are more difficult to learn than the protein-ligand binding affinity itself. This is probably caused by the approximate nature of classical scoring functions.

![Graphs showing predicted vs experimental binding affinity for CASF-2013 and CASF-2016](image)

Figure 7: Predicted versus experimental binding affinity using the Δ-learning approach with Δ-AEScore, when only protein and ligand atoms are retained.

Table 1 compares our Δ-learning results on the CASF-2013 and CASF-2016 data sets with the Δ_{vina}RF scoring function, arguably the most successful implementation of this approach.[18] Our model performs significantly better than Δ_{vina}RF on the CASF-2013 dataset and comparably on the CASF-2016. It is worth noting that Δ_{vina}RF is the best scoring function on the scoring and ranks power tests for the CASF-2016 benchmark, and is ranking consistently amongst the top scoring functions for the docking and screening power tests. However, Δ_{vina}RF is calibrated on protein-ligand complexes from the PDBbind, which overlaps with ~50% of the CASF-2016 test set and its performance might therefore have been artificially enhanced by a large overlap between the training and test sets.[39]

Both Δ_{vina}RF and Δ-AEScore outperform the classical scoring function AutoDock Vina in the scoring power test, by a large margin.[39]
Table 1: Performance of $\Delta$-AEScore compared to the $\Delta_{\text{vina}}$RF for affinity prediction on the CASF-2013 and CASF-2016 benchmarks. For $\Delta$-AEScore the “hard overlap” between the training and both test sets is removed while for $\Delta_{\text{vina}}$RF only the “hard overlap” between the training set and CASF-2013 is removed. The best performance for each test set is underlined. RMSE values are given in pK units.

| Model | Training Set | Test Set   | RMSE | Pearson’s $r$ |
|-------|--------------|------------|------|---------------|
| $\Delta$-AEScore$^\dagger$ | Refined 2013 | CASF-2013  | 1.53 | 0.74          |
| $\Delta$-AEScore$^\dagger$(no H) | Refined 2013 | CASF-2013  | 1.52 | 0.74          |
| $\Delta_{\text{vina}}$RF$^{[18]}$ | Refined 2013 | CASF-2013  | —    | 0.69          |
| Vina (optim) | —          | CASF-2013  | 1.82 | 0.61          |

| —— | —— | —— | —— | —— |
| $\Delta$-AEScore$^\dagger$ | Refined 2016 | CASF-2016 | 1.34 | 0.79 |
| $\Delta$-AEScore$^\dagger$(no H) | Refined 2016 | CASF-2016 | 1.32 | 0.80 |
| $\Delta_{\text{vina}}$RF$^{[9][18]}$ | —         | CASF-2016  | —    | 0.81 |
| Vina (optim) | —         | CASF-2016  | 1.75 | 0.59 |

$^\dagger$ This work.

**Ranking Power**

In terms of ranking power $\Delta$-AEScore has a Spearman’s correlation coefficient of $0.59 \in [0.47, 0.68]_{90\%}$ CI, a Kendall’s correlation coefficient of $0.52 \in [0.42, 0.60]_{90\%}$ CI and a PI of $0.61 \in [0.49, 0.69]_{90\%}$ CI on the CASF-2016 benchmark. For the CASF-2013 benchmark, $\Delta$-AEScore has a Spearman’s correlation coefficient of $0.61 \in [0.47, 0.71]_{90\%}$ CI, a Kendall’s correlation coefficient of $0.58 \in [0.44, 0.67]_{90\%}$ CI and a PI of $0.63 \in [0.49, 0.73]_{90\%}$ CI.

The performance of $\Delta$-AEScore in the ranking power test is lower than the performance of AEScore. This is to be attributed to the poor performance of AutoDock Vina on this benchmark, with a Spearman’s correlation coefficient of $0.53 \in [0.43, 0.61]_{90\%}$ CI on the CASF-2016 benchmark. However, the use of AEScore on top of AutoDock Vina allows us to improve the performance of the latter in both scoring and ranking.

**Docking Power**

We next wanted to see if the corrections to the AutoDock Vina scoring function can be applied in the context of docking. Using the docking decoys of the CASF-2016 benchmark dataset
we obtain a top one success rate of $85.6\% \in [81.1\%, 88.1\%]_{90\% \text{ CI}}$, a top two success rate of $94.4\% \in [90.9\%, 95.8\%]_{90\% \text{ CI}}$ and a top three success rate of $95.8\% \in [92.6\%, 96.8\%]_{90\% \text{ CI}}$. This is a very significant improvement on the previous results obtained with AEScore.

The top one performance is lower than Autodock Vina itself, which performs extremely well in this benchmark with a top 1 success rate of $90.2\% \in [86.7\%, 92.6\%]_{90\% \text{ CI}}$ (when the native ligand binding pose is included), and compared to the performance of $\Delta_{\text{vina}}\text{RF}$, the second-best performing scoring function in CASF-2016 with a top 1 success rate of $89.1\% \in [85.6\%, 91.6\%]_{90\% \text{ CI}}$. However, the much higher performance compared to AEScore indicates that the protein-ligand binding site representation and the model architecture used for AEScore are amenable to $\Delta$-learning. We thus have good scoring power—significantly better than AutoDock Vina alone—while retaining the excellent docking power of Autodock Vina.

**Screening Power**

Given the good success rate of $\Delta$-AEScore in the docking power test, we wanted to evaluate $\Delta$-AEScore in the context of virtual screening as well. The screening power test assesses the ability of a scoring function to identify true binders among a large pool of decoys. There are two types of screening power tests provided in the CASF-2016 benchmark: in forward screening, the goal is to identify the true binders for a given target, while in reverse screening, the goal is to identify a potential target for a given active compound.

For the forward screening power test, $\Delta$-AEScore ranks the best ligand among the top 1% of candidates with a success rate of $19.3\% \in [10.5\%, 26.3\%]_{90\% \text{ CI}}$. The top 5% success rate and the 10% success rates are $49.1\% \in [36.8\%, 57.9\%]_{90\% \text{ CI}}$ and $54.4\% \in [42.1\%, 63.2\%]_{90\% \text{ CI}}$, respectively. The top 1% success rate is rather low compared to Autodock Vina ($29.8\% \in [19.3\%, 38.6\%]_{90\% \text{ CI}}$) and $\Delta_{\text{vina}}\text{RF}$ ($42.1\% \in [29.8\%, 50.9\%]_{90\% \text{ CI}}$), but top 5% and top 10% performances are in line with $\Delta_{\text{vina}}\text{RF}$ and better than AutoDock Vina itself. Again, it is worth re-iterating that the reported performance of $\Delta_{\text{vina}}\text{RF}$ on CASF-2016 might be
artificially inflated by the overlap between training and test sets.\textsuperscript{39}

Another quantitative metric of the screening power is the enrichment factor (EF), defined by:

$$ EF_\alpha = \frac{TB_\alpha}{\alpha TB_{tot}} $$

where $TB_\alpha$ denotes the number of true binders amongst the top $\alpha$% candidates and $TB_{tot}$ is the total number of true binders. $\Delta$-AEScore has an average EF\textsubscript{1}% of 6.16 ∈ [4.14, 8.75]\textsubscript{90% CI}, an average EF\textsubscript{5}% of 3.76 ∈ [2.94, 4.63]\textsubscript{90% CI} and an average EF\textsubscript{10}% of 2.48 ∈ [2.02, 3.00]\textsubscript{90% CI}. The EF are not too far from AutoDock Vina’s EF on CASF-2016, with an EF\textsubscript{1}% of 7.7 ∈ [5.37, 10.97]\textsubscript{90% CI}. $\Delta_{\text{vina}}$RF is again amongst the top performing scoring functions on CASF-2016, not withstanding the training/testing caveats discussed above; $\Delta_{\text{vina}}$RF EF\textsubscript{1}% is 11.73 ∈ [8.84, 15.41]\textsubscript{90% CI}.\textsuperscript{39}

For reverse screening on the CASF-2016 benchmark, we obtain a top 1% success rate of 11.9% ∈ [8.8%, 15.1%]\textsubscript{90% CI}, a top 5% success rate of 19.3% ∈ [15.4%, 23.2%]\textsubscript{90% CI} and a top 10% success rate of 27.0% ∈ [22.5%, 30.9%]\textsubscript{90% CI}. Again, the results are similar to AutoDock Vina (13.7% ∈ [10.5%, 16.8%]\textsubscript{90% CI}) and slightly worse than the optimistic values reported for $\Delta_{\text{vina}}$RF (15.1% ∈ [11.6%, 18.6%]\textsubscript{90% CI}).\textsuperscript{39}

**Ligand-Only Affinity Prediction**

In order to test the effect of protein information in the binding affinity prediction and to elucidate possible biases in the dataset,\textsuperscript{86} we also trained a model with only the ligand atoms ($d = 0$ Å). The AEVs’ parameters used to describe ligand atoms are left unchanged.

For the CASF-2013 dataset we obtained an RMSE of 1.65 pK units and a Pearson’s correlation of 0.70, while for the CASF-2016 dataset we obtained an RMSE of 1.49 pK units and a Pearson’s correlation of 0.74 (when only protein and ligand atoms are kept and systems are automatically protonated). Fig. \textsuperscript{8} also reports the results when hydrogen atoms are removed and when the model is trained on a dataset with a protein/ligand/pocket
similarity threshold of 95% similarity with the training set.

As shown in Fig. 8 (and, equivalently, Table S2), the performance of the model in absence of protein atoms (L) is always worse than that obtained when including both ligand and protein atoms (P + L). This indicates that the model is able to exploit the additional information about the binding site provided by the protein atoms in order to improve binding affinity predictions. However, the difference is not as striking as one might expect.

The same observations apply to the Δ-learning approach, although the difference between protein-ligand (P + L) and ligand-only (L) models is less pronounced. This suggests that corrections to the AutoDock Vina scoring function mainly stem from the information about the ligand and that information about the protein target plays a minor role.

Figure 8: Pearson’s correlation coefficient for different models incorporating atoms from the protein and the ligand (P + L, d = 3.5 Å) or atoms of the ligand only (L), for the CASF-2013 and CASF-2016 benchmarks. Each box is color-coded by the corresponding RMSE in pK units.

The fact that AEScore models using only information about the ligand already perform well is in line with recent work from Boyles et al. who showed that ligand features alone are
predictive of the mean protein-ligand binding affinity in PDBbind. Additionally, ligand information plays a significant role in affinity prediction in deep learning models as well. For ligand-only predictions, AEScore is essentially learning a conformation-dependent fingerprint of the active ligand and using such information to predict the mean binding affinity of said ligand; RDKit descriptors alone, combined with a random forest model, can already achieve a Pearson’s correlation coefficient of 0.71 on CASF-2013 and of 0.76 on CASF-2016, as demonstrated by Boyles et al. Our results suggest that the AEScore model presented here can use AEVs as 3D ligand fingerprints and use such information to predict the average binding affinity of a ligand in the same way RDKit descriptors allow.

Visualization

One advantage of working with atomic coordinates directly and using a end-to-end differentiable model, is that the gradient of the loss function can be computed with respect to the atomic coordinates. This technique has been previously used to interpret CNN-based scoring functions: the negative of the gradient of the loss function with respect to the atomic coordinates indicates where the model would like the atoms to “move” in order to minimise the loss function. Leveraging PyTorch’s automatic differentiation tools, such gradients are readily available.

Figure shows the magnitude of the gradients for ligand and protein atoms for the complexes of the CASF-2016 test set with lowest and highest absolute error. For the 3ZT2 complex—the one with lowest absolute error—the maximum gradient magnitude is 1.91 and the gradients are small everywhere, with the exception of a particular functional group of the ligand. For the 2XDL complex—the one with highest absolute error—the maximum gradient magnitude is 21.31.

The gradients of the loss function with respect to the atomic coordinates can also be employed as fictitious forces for a local geometry optimisation: atoms can be displaced along the negative of the gradient with standard optimisation techniques in order to obtain
Figure 9: Visualization of the norm of the gradient of the loss function with respect to atomic coordinates for each atom considered by the model, for (a) HIV type 1 integrase (PDB ID 3ZT2) and (b) chaperone HSP90 (PDB ID 2XDL). Gradient for protein atoms is shown in orange while the gradient for ligand atoms is shown in blue. The colour scale is normalized between 0 and the maximum gradient norm for each system (1.91 and 21.31, respectively); the more saturated the colour, the higher the gradient on that atom.
configurations that minimize the loss.\textsuperscript{10}

**Discussion**

Figure\textsuperscript{10} compares the performance of our model—denoted AEScore—in terms of binding affinity prediction for the CASF-2013 and CASF-2016 benchmark datasets, with other state-of-the-art machine learning and deep learning models. The performance of the other methods is taken directly from the references reported. The same results are also reported in Table S3, together with RMSEs and additional information about models and training datasets.

In the literature there is some confusion about the CASF benchmark and the PDBbind Core set, as indicated on the PDBbind website.\textsuperscript{72} In Table S3 we indicate which dataset has been used for testing. The CASF-2016 benchmark set contains 285 protein-ligand complexes while the PDBbind Core 2016 set contains 290 protein-ligand complexes (complexes 4MRW, 4MRZ, 4MSN, 5C1W, 4MSC, and 3CYX in PDBbind Core 2016 are not included in CASF-2016, while 1G2K is an additional complex not present in the Core set).\textsuperscript{64}

Our results compare favourably with other state-of-the-art deep learning models based on feed-forward NNs or CNNs and machine learning scoring functions based on random forests on both the CASF-2016 and PDBbind Core 2016 test sets. In particular, AEScore performs better than the recently published pair method of Zhu et al.\textsuperscript{72}, which is also based on feed-forward NNs. Additionally, our methods has the lowest RMSE and highest Pearson’s correlation coefficient on the CASF-2016 benchmark (amongst those reported).

However, a quantitative and statistically sound comparison with other methods is somewhat difficult because error bars and confidence intervals are often not reported. If confidence intervals were reported by all methods, one could compute the confidence interval of the difference between two methods with convenient approximations and use this information to test significance against the null hypothesis.\textsuperscript{76}

One of the main advantages of the AEV-based approach is that it is translationally and
Figure 10: Performance of different machine learning and deep learning models for binding affinity prediction on the CASF-2013 and CASF-2016 benchmarks as well as for the Core 2016 set. Our results, shown in orange, include 90% confidence intervals. Numerical values for the Pearson’s correlation coefficient and the RMSE are reported in Table S3, together with references for all the different methods. 4,11,20–24,51,63,64,71–74
rotationally invariant, thus removing an additional source of variability. This is not the case for scoring functions based on standard CNNs, and we expect random translations and rotations of the input protein-ligand systems to give different results, while our results would remain unchanged. Figure S4 shows the variation in CNN-based predictions as a function of the angle of rotation for a particular complex. Data augmentation with random translations and rotations has proved to be essential to prevent overfitting and significantly improve training in CNN-based scoring functions, but this is computationally expensive—another advantage of our approach.

In addition to being translationally and rotationally invariant, our model also requires minimal information about the system. Only elements and atomic coordinates are needed by the model. Other methods often require additional information such as force-field parameters or specific atom types and are therefore limited by these parameters and underlying assumptions.

Compared to “classical” machine learning scoring functions, our method performs similarly to RF Score and other RF-based scoring functions. Despite recent advances in deep learning architectures, which consistently outperform “classical” machine learning algorithms in image recognition and natural language processing, RFs remain very competitive for binding affinity predictions. All top-performing machine learning and deep learning methods considered here achieve similar performance on the CASF benchmarks—as measured by Pearson’s correlation coefficient. This is likely due to the fact that errors in the experimental measurements of the binding affinity and the X-ray crystallographic coordinates of the protein-ligand complex set a theoretical upper limit on the maximal performance of scoring functions trained on such noisy data.

It is instructive to also compare the performance of our model with standard docking scoring functions. Here we used the AutoDock Vina scoring function as implemented in smina as a baseline. We see that our model clearly outperforms the Vina scoring function for protein-ligand affinity predictions, as do other machine learning and deep learning ap-
proaches. This is expected, since previous studies show that standard scoring functions do not perform very well in scoring and ranking power tests.\textsuperscript{37}

The removal of the systems in the CASF test set from the PDBbind Refined set used for training is a common practice with machine learning and deep learning scoring functions and therefore ensures a fair comparison with other methods. However, it has been previously noted that the performance on the CASF set is not necessarily very indicative of the a model’s ability to generalize, since this dataset samples the same regions of the chemical and target spaces as the PDBbind dataset.\textsuperscript{40,51} In order to better evaluate the ability of a model to generalize, we tested its performance when trained on a recently developed non-redundant training set.\textsuperscript{40} We showed in Fig. 5 that the performance of AEScore deteriorates gradually when the similarity between the training set and the test set is reduced, in contrast with many other machine learning scoring functions that are severely inhibited by removing structurally redundant samples from the training set.\textsuperscript{40}

When we tested AEScore for docking power we obtained poor results. This is not surprising since the model was trained to predict binding affinities given the correct binding pose and it was not trained explicitly to distinguish low- from high-RMSD poses. However, we showed that by combining AEScore with the classical scoring function AutoDock Vina using a $\Delta$-learning approach improves the performance in terms of docking and screening, while maintaining good scoring and ranking performance. As already demonstrated by $\Delta_{\text{vina}}\text{RF}$, this is a good approach for developing a scoring function that works well on all four tasks: scoring, ranking, docking and screening. Usually, machine learning and deep learning scoring functions work very well for scoring but not as well for docking and virtual screening, while classical scoring functions have the opposite behaviour. Figure 11 collects most of the results of AEScore and $\Delta$-AEScore on the CASF-2016 benchmark, together with the results for $\Delta_{\text{vina}}\text{RF}$ and AutoDock Vina (our baseline) as reported by Su et al.\textsuperscript{39}. We also added the best- and worst-performing scoring functions for each of the CASF-2016 benchmarks reported in Su et al.\textsuperscript{39}, whenever these scoring function were different from $\Delta_{\text{vina}}\text{RF}$.\textsuperscript{31}
or AutoDock Vina. We see that both AEScore and ∆-AEScore perform well in scoring and ranking power tests, but AEScore performance for docking is low. However, the ∆-learning approach is able to recover a good docking power (similar to the AutoDock Vina baseline) while retaining a good performance in scoring and ranking. The performance of ∆-AEScore in forward screening is rather poor as measured by EF 1% or top 1% success rate, but greatly improves for EF 5% and the top 5% success rate.

Given the good performance of our ligand-only model—which was nonetheless consistently worse than that of the protein-ligand model—it is clear that the model is extracting a lot of information from the ligand. Finding strategies to force the model to rely more on protein information could further improve the model and make it more transferable. This is a known problem and strategies to force the model to rely more on the protein structure are an active area of research.

The advantage of using an end-to-end differentiable model is that the gradient of the scoring function with respect to the input parameters can be readily obtained by backpropagation. Since the TorchANI AEVComputer is fully differentiable and its inputs are atomic coordinates, the gradient of the scoring function with respect to atomic coordinates can be computed. This can be used for visualization, which could help to understand the behaviour of the scoring function. In future iterations of the model, such gradients could be employed in the context of a local geometry optimization of the binding pose.

Finally, it is worth noting that we exploited the representation and architecture commonly used to develop NNP to predict a different endpoint, namely the protein-ligand binding affinity, and corrections to classical scoring functions. However, given the success of NNPs, one could use them in a MM/PBSA- or MM/GBSA-style approach to directly compute the free energy of binding on more physical grounds. In fact, approaches to combine NNP with molecular mechanics for drug discovery applications are already starting to appear.
Figure 11: Performance of AEScore, Δ-AEScore, Δ\textsubscript{vina}RF, and AutoDock Vina. The best- and worst-performing scoring function on CASF-2016 (as reported by Su et al.\textsuperscript{39}) is also added for comparison. The results include 90% confidence intervals (where they were available).
Conclusions

We demonstrated that AEVs are a promising representation of the protein-ligand binding site (and of the ligand alone, for ligand-based model) amenable to machine learning-based predictions of the protein-ligand binding affinity, and of corrections to classical scoring functions. This representation is rotationally and translationally invariant and, in contrast to CNN-based scoring functions, does not require data augmentation. The results reported here for AEScore show similar or better performance than other state-of-the-art machine learning and deep learning methods on the CASF-2013 and CASF-2016 benchmarks (as well as the Core 2016 set) in binding affinity prediction.

However, AEScore still presents shortcomings also found in other machine learning and deep learning scoring functions: when only AEVs for the ligand are used, the performance is still surprisingly high. This suggests that information about the ligand provides a significant contribution to the protein-ligand binding affinity prediction, as discussed in the literature. This is particularly evident in ∆-AEScore corrections, where the difference between ligand-only and protein-ligand models is small and corrections seems to stem mainly from ligand information.

Using training sets with decreasing similarity to the test set, first introduced by Boyles et al. and later by Su et al., we showed that our model is not completely hindered by the removal of systems with high similarity, but that AEScore’s performance deteriorates only gradually. This is in contrast with other machine learning and deep learning scoring functions, where a performance drop is observed as soon as a similarity threshold is introduced. This property could be useful in real drug discovery applications, where data on similar or related systems (such as a congenic series of ligands) is acquired gradually.

In this work we did not optimise the ANI parameters for radial symmetry functions, and we did not explore the full flexibility of the angular symmetry functions. A Bayesian optimisation of the symmetry functions’ hyper parameter space could lead to further improvements of the scoring function.
We also showed that the AEScore model presented here can be exploited in tandem with standard docking scoring functions using a $\Delta$-learning approach, in order to improve the performance in docking and virtual screening (in which AEScore does not perform well). $\Delta$-AEScore outperforms the $\Delta_{\text{vina}}$RF scoring function by a good margin on the CASF-2013 test set and performs similarly on the CASF-2016 test set (notwithstanding the training/test set overlap in $\Delta_{\text{vina}}$RF reported performance). $\Delta$-learning has the advantage of partially retaining the good docking and screening power of standard scoring functions while improving affinity predictions using machine-learning corrections, allowing the development of a scoring function that works reasonably well on all four tasks of early-stage structure-based drug discovery applications.
Acknowledgement

This work was supported by funding from the Biotechnology and Biological Sciences Research Council (BBSRC) [BB/MO11224/1] National Productivity Investment Fund (NPIF) [BB/S50760X/1] and Evotec (UK) via the Interdisciplinary Biosciences DTP at the University of Oxford.

RM thanks Dr. Irfan Alibay for fruitful discussions, especially about Python and MD-Analysis.

Supporting Information

The code for training and inference is available on GitHub at [https://github.com/bigginlab/aescore](https://github.com/bigginlab/aescore) while data and scripts for the numerical experiments are available on Zenodo (https://doi.org/10.5281/zenodo.4155365).
References

(1) Aldeghi, M.; Heifetz, A.; Bodkin, M. J.; Knapp, S.; Biggin, P. C. Accurate calculation of the absolute free energy of binding for drug molecules. *Chem. Sci.* **2016**, *7*, 207–218.

(2) Aldeghi, M.; Bluck, J. P.; Biggin, P. C. In *Methods in Molecular Biology*; Gore, M., Jagtap, U. B., Eds.; Springer New York: New York, NY, 2018; pp 199–232.

(3) Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nat. Rev. Drug Discov* **2004**, *3*, 935–949.

(4) Boyles, F.; Deane, C. M.; Morris, G. M. Learning from the ligand: Using ligand-based features to improve binding affinity prediction. *Bioinformatics* **2019**, *36*, 758–764.

(5) Liu, J.; Wang, R. Classification of Current Scoring Functions. *J. Chem. Inf. Model.* **2015**, *55*, 475–482.

(6) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking 1 1Edited by F. E. Cohen. *J. Mol. Biol.* **1997**, *267*, 727–748.

(7) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening. *J. Med. Chem.* **2004**, *47*, 1750–1759.

(8) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785–2791.

(9) Ravindranath, P. A.; Forli, S.; Goodsell, D. S.; Olson, A. J.; Sanner, M. F. AutoDockFR: Advances in Protein-Ligand Docking with Explicitly Specified Binding Site Flexibility. *PLoS Comput. Biol.* **2015**, *11*, e1004586.
(10) Trott, O.; Olson, A. J. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* 2009, *31*, 455–461.

(11) Ballester, P. J.; Mitchell, J. B. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* 2010, *26*, 1169–1175.

(12) Ain, Q. U.; Aleksandrova, A.; Roessler, F. D.; Ballester, P. J. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *WIREs Comput. Mol. Sci.* 2015, *5*, 405–424.

(13) Cherkasov, A. et al. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* 2014, *57*, 4977–5010.

(14) Muratov, E. N. et al. QSAR without borders. *Chem. Soc. Rev.* 2020, *49*, 3525–3564.

(15) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc., 2012; pp 1097–1105.

(16) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 2017, *60*, 84–90.

(17) Graves, A.; Mohamed, A.-R.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. *arXiv:1303.5778 [cs]* 2013.

(18) Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* 2012, *29*, 82–97.

(19) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, *521*, 436–444.

38
(20) 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.

(21) 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.

(22) Jiménez, J.; Škalić, 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.

(23) Cang, Z.; Mu, L.; Wei, G.-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Comput Biol* 2018, 14, e1005929.

(24) 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.

(25) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. PotentialNet for Molecular Property Prediction. *ACS Cent. Sci.* 2018, 4, 1520–1530.

(26) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* 2007, 98, 146401.

(27) Smith, J.; Isayev, O.; Roitberg, A. ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* 2017, 8, 3192–3203.

(28) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* 2010, 104, 136403.
(29) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B* **2013**, *87*, 184115.

(30) Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.* **2017**, *3*, e1701816.

(31) Behler, J. Neural network potential-energy surfaces in chemistry: A tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930.

(32) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is more: Sampling chemical space with active learning. *J. Chem. Phys.* **2018**, *148*, 241733.

(33) Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E. TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Network Potentials. *J. Chem. Inf. Model.* **2020**, *60*, 3408–3415.

(34) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc., 2019; pp 8024–8035.

(35) 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.

(36) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014**, *54*, 1700–1716.

(37) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for
Developing Protein–Ligand Interaction Scoring Functions. Acc. Chem. Res. 2017, 50, 302–309.

(38) Li, Y.; Han, L.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. J. Chem. Inf. Model. 2014, 54, 1717–1736, PMID: 24708446.

(39) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. J. Chem. Inf. Model. 2018, 59, 895–913.

(40) Su, M.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Tapping on the Black Box: How Is the Scoring Power of a Machine-Learning Scoring Function Dependent on the Training Set? J. Chem. Inf. Model. 2020, 60, 1122–1136.

(41) O’Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. J. Cheminform. 2011, 3, 33.

(42) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. J. Comput. Chem. 2011, 32, 2319–2327.

(43) Gowers, R.; Linke, M.; Barnoud, J.; Reddy, T.; Melo, M.; Seyler, S.; Domanski, J.; Dotson, D.; Buchoux, S.; Kenney, I.; Beckstein, O. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. Proceedings of the 15th Python in Science Conference. 2016; pp 98 – 105.

(44) O’Boyle, N. M.; Morley, C.; Hutchison, G. R. Pybel: A Python wrapper for the Open-Babel cheminformatics toolkit. Chem. Cent. J. 2008, 2, 5.

(45) Banci, L. Molecular dynamics simulations of metalloproteins. Curr. Opin. Chem. Biol. 2003, 7, 524.
(46) Çınaroğlu, S. S.; Timuçin, E. Comparative Assessment of Seven Docking Programs on a Nonredundant Metalloprotein Subset of the PDBbind Refined. *J. Chem. Inf. Model.* **2019**, *59*, 3846–3859.

(47) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The ∆-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.

(48) Wang, C.; Zhang, Y. Improving scoring-docking-screening powers of protein-ligand scoring functions using random forest. *J. Comput. Chem.* **2016**, *38*, 169–177.

(49) Lu, J.; Hou, X.; Wang, C.; Zhang, Y. Incorporating Explicit Water Molecules and Ligand Conformation Stability in Machine-Learning Scoring Functions. *J. Chem. Inf. Model.* **2019**, *59*, 4540–4549.

(50) Ericksen, S. S.; Wu, H.; Zhang, H.; Michael, L. A.; Newton, M. A.; Hoffmann, F. M.; Wildman, S. A. Machine Learning Consensus Scoring Improves Performance Across Targets in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2017**, *57*, 1579–1590.

(51) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 4200–4215.

(52) Van Rossum, G.; Drake, F. L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, 2009.

(53) O’Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J Cheminform* **2011**, *3*, 33.

(54) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362.
(55) Virtanen, P. et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat Methods* **2020**, *17*, 261–272.

(56) Wes McKinney, Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. 2010; pp 56 – 61.

(57) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95.

(58) Waskom, M.; the seaborn development team, mwaskom/seaborn. 2020; [https://doi.org/10.5281/zenodo.592845](https://doi.org/10.5281/zenodo.592845).

(59) Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; Mueller, A. Scikit-learn. *GetMobile: Mobile Comp. and Comm.* **2015**, *19*, 29–33.

(60) Krekel, H.; Oliveira, B.; Pfannschmidt, R.; Bruynooghe, F.; Laugher, B.; Bruhin, F. pytest 6.0. 2004; [https://github.com/pytest-dev/pytest](https://github.com/pytest-dev/pytest).

(61) Kingma, D. P.; Ba, J. ADAM: A Method for Stochastic Optimization. 2014.

(62) Pearlman, D. A.; Charifson, P. S. Are Free Energy Calculations Useful in Practice? A Comparison with Rapid Scoring Functions for the p38 MAP Kinase Protein System†. *J. Med. Chem.* **2001**, *44*, 3417–3423.

(63) Kwon, Y.; Shin, W.-H.; Ko, J.; Lee, J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using the Ensemble of 3D-Convolutional Neural Network. 2020.

(64) Nguyen, D. D.; Wei, G.-W. AGL-Score: Algebraic Graph Learning Score for Protein–Ligand Binding Scoring, Ranking, Docking, and Screening. *J. Chem. Inf. Model.* **2019**, *59*, 3291–3304.

(65) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on a Diverse Test Set. *J. Chem. Inf. Model.* **2009**, *49*, 1079–1093.
(66) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* 2019, *59*, 947–961.

(67) Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *J. Chem. Inf. Model.* 2018, *58*, 916–932.

(68) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLoS One* 2019, *14*, e0220113.

(69) Hochuli, J.; Helbling, A.; Skaist, T.; Ragoza, M.; Koes, D. R. Visualizing convolutional neural network protein-ligand scoring. *J. Mol. Graphics Model.* 2018, *84*, 96–108.

(70) Ragoza, M.; Turner, L.; Koes, D. R. Ligand Pose Optimization with Atomic Grid-Based Convolutional Neural Networks. 2017.

(71) Durrant, J. D.; McCammon, J. A. NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function. *J. Chem. Inf. Model.* 2011, *51*, 2897–2903.

(72) Zhu, F.; Zhang, X.; Allen, J. E.; Jones, D.; Lightstone, F. C. Binding Affinity Prediction by Pairwise Function Based on Neural Network. *J. Chem. Inf. Model.* 2020, *60*, 2766–2772.

(73) Afifi, K.; Al-Sadek, A. F. Improving classical scoring functions using random forest: The non-additivity of free energy terms’ contributions in binding. *Chem. Biol. Drug Des.* 2018, *92*, 1429–1434.

(74) Li, Y.; Rezaei, M. A.; Li, C.; Li, X.; Wu, D. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. 2019.

(75) PDBbind-CN Database, http://pdbbind-cn.org/.
(76) Nicholls, A. Confidence limits, error bars and method comparison in molecular modeling. Part 2: Comparing methods. *J Comput Aided Mol Des* **2016**, *30*, 103–126.

(77) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. *J. Chem. Inf. Model.* **2013**, *53*, 1893–1904.

(78) Scantlebury, J.; Brown, N.; Von Delft, F.; Deane, C. M. Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalize to Unseen Target Classes and Highlight Important Binding Interactions. *J. Chem. Inf. Model.* **2020**, *60*, 3722–3730.

(79) Genheden, S.; Ryde, U. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opin. Drug Discovery* **2015**, *10*, 449–461.

(80) Cole, D. J.; Mones, L.; Csányi, G. A machine learning based intramolecular potential for a flexible organic molecule. *Faraday Discuss.* **2020**, –.

(81) Lahey, S.-L. J.; Rowley, C. N. Simulating protein–ligand binding with neural network potentials. *Chem. Sci.* **2020**, *11*, 2362–2368.

(82) Rufa, D. A.; Bruce Macdonald, H. E.; Fass, J.; Wieder, M.; Grinaway, P. B.; Roitberg, A. E.; Isayev, O.; Chodera, J. D. Towards chemical accuracy for alchemical free energy calculations with hybrid physics-based machine learning / molecular mechanics potentials. *bioRxiv* **2020**, –.
Graphical TOC