HHS Public Access

Author manuscript

J Chem Inf Model. Author manuscript; available in PMC 2018 April 24.

Published in final edited form as:

J Chem Inf Model. 2017 April 24; 57(4): 942–957. doi:10.1021/acs.jcim.6b00740.

Protein-Ligand Scoring with Convolutional Neural Networks

Matthew Ragoza[†,‡], **Joshua Hochuli**[‡,¶], **Elisa Idrobo**[§], **Jocelyn Sunseri**[‖], and **David Ryan Koes**[‖]

[†]Department of Neuroscience, University of Pittsburgh, Pittsburgh, PA 15260

[‡]Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260

[¶]Department of Biological Sciences, University of Pittsburgh, Pittsburgh, PA 15260

[§]Department of Computer Science, The College of New Jersey, Ewing, NJ 08628

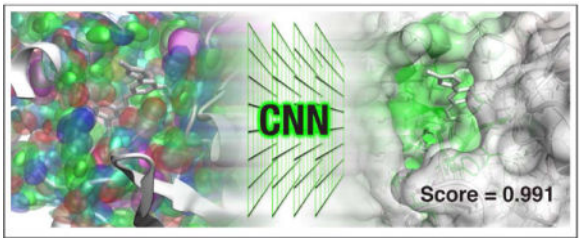
[‖]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 15260

Abstract

Computational approaches to drug discovery can reduce the time and cost associated with experimental assays and enable the screening of novel chemotypes. Structure-based drug design methods rely on scoring functions to rank and predict binding affinities and poses. The ever-expanding amount of protein-ligand binding and structural data enables the use of deep machine learning techniques for protein-ligand scoring.

We describe convolutional neural network (CNN) scoring functions that take as input a comprehensive 3D representation of a protein-ligand interaction. A CNN scoring function automatically learns the key features of protein-ligand interactions that correlate with binding. We train and optimize our CNN scoring functions to discriminate between correct and incorrect binding poses and known binders and non-binders. We find that our CNN scoring function outperforms the AutoDock Vina scoring function when ranking poses both for pose prediction and virtual screening.

Graphical abstract



Correspondence to: David Ryan Koes.

Supporting Information **Available**

Supporting Figures S1–S6 and Tables S1–S17.

Introduction

Protein-ligand scoring is a keystone of structure-based drug design. Scoring functions rank and score protein-ligand structures with the intertwined goals of accurately predicting the binding affinity of the complex, selecting the correct binding mode (pose prediction), and distinguishing between binders and non-binders (virtual screening).

Existing empirical[1–7] and knowledge-based[8–13] scoring functions parameterize a predetermined function, which is usually physically inspired, to fit data, such as binding affinity values. Scoring functions that use machine learning[1,13–25] provide greater flexibility and expressiveness as they learn both parameters and the model structure from data. However, the resulting model often lacks interpretability, and the increased expressiveness increases the probability of overfitting the model to the data, in which case the scoring function will not generalize to protein targets or ligand chemotypes not in the training data. The risk of overfitting increases the importance of rigorous validation,[26,27] but the inherent increase in flexibility allows machine learning methods to outperform more constrained methods when trained on an identical input set.[28] The choice of input features can limit the expressiveness of a machine learning method. Features such as atom interaction counts,[22] pairwise atom distance descriptors,[13] interaction fingerprints,[21] or “neural fingerprints” generated by learned atom convolutions[24] necessarily eliminate or approximate the information inherent in a protein-ligand structure, such as precise spatial relationships.

Neural networks[29] are a neurologically inspired supervised machine learning technique that is routinely and successfully applied to problems such as speech recognition and image recognition. A basic network consists of an input layer, one or more hidden layers, and an output layer of interconnected nodes. Each hidden node computes a feature that is a function of the weighted input it receives from the nodes of the previous layer. The outputs are propagated to each successive layer until the output layer generates a classification. The network architecture and choice of activation function for each layer determine the design of the network. The weights that parameterize the model are typically optimized to fit a given training set of data to minimize the error of the network.

Deep learning[30] refers to neural networks with many layers, which are capable of learning highly complex functions and have been made practical largely by the increase in computational power provided by modern graphics cards. The expressiveness of a neural network model can be controlled by the network architecture, which defines the number and type of layers that process the input to ultimately yield a classification. The network architecture can be manually or automatically tuned with respect to validation sets to be as expressive as needed to accurately model the data and reduce overfitting.[31,32] Structure-based scoring functions that use neural networks[20–25] were recently shown to be competitive with empirical scoring in retrospective virtual screening exercises while also being effective in a prospective screen of estrogen receptor ligands.[33] Neural networks have also been successfully applied in the cheminformatics domain through creative manipulations of 2D chemical structure and construction of the network architecture,[34–37] and as alternatives to computationally intensive quantum chemical calculations.[38–40]

Convolutional neural networks (CNNs)[30] are a type of neural network commonly used in image recognition. CNNs hierarchically decompose an image so that each layer of the network learns to recognize higher-level features while maintaining their spatial relationships as illustrated in Figure 1. For example, the first layer may learn to identify lines and corners in an image, the next may assemble these features to learn different shapes, and so on until the final layer can recognize something as high-level and complex as a dog breed. CNNs are the best performing method for image recognition,[41] as epitomized by the GoogLeNet winning entry to the ImageNet Large Scale Visual Recognition Challenge of 2014[32] and the Microsoft ResNet entry of 2015,[42] both of which perform better at classifying images than most humans.[43]

The impressive performance of CNNs at the image recognition task suggests that they are well-suited for learning from other types of spatial data, such as protein-ligand structures. Unlike previous machine learning methods, a CNN scoring method does not require the extraction of high-level features from the structure. Instead, the method automatically identifies the most informative features required for successful scoring. This allows for the extraction of features that are not readily encoded in simplified potentials, such as hydrophobic enclosure[44] or surface area dependent terms,[45] as well as features that have not yet been identified as relevant by any existing scoring function.

Here we describe the development of a CNN model for protein-ligand scoring that is trained to classify compound poses as binders or non-binders using a 3D grid representation of protein-ligand structures generated through docking. We show that our CNN scoring method outperforms the AutoDock Vina' scoring function that is used to generate the poses both when selecting poses for pose prediction and for virtual screening tasks. Our method outperforms other machine learning approaches in our virtual screening evaluation even when it is also trained to perform well at pose-sensitive pose prediction. Finally, we illustrate how our CNN score can be decomposed into individual atomic contributions to generate informative visualizations.

Methods

In order to create our CNN scoring models we utilize two training sets, one focused on pose prediction and the other on virtual screening. The structural information in these sets is translated into a custom input format appropriate for CNN processing. We systematically optimize the network topology and parameters using clustered cross-validation. The optimized network is then trained on the full training set and evaluated with respect to independent test sets. The predictions from the resulting models are decomposed into atomic contributions to provide informative visualizations.

Training Sets

We utilize two training sets focused on two different goals: pose prediction and virtual screening. In all cases we generate ligand poses for actives and decoys using docking with smina[1] and the AutoDock Vina scoring function.[7] We note that methods using the AutoDock Vina scoring function performed well in blind evaluations of docking performance.[46,47] We use docked poses, even for active compounds with a known crystal structure, because (1)

these are the types of poses the model will ultimately have to score and (2) to avoid the model simply learning to distinguish between docked poses and crystal structures (which were likely optimized with different force fields).

Ligands are docked against a reference receptor within a box centered around a reference ligand with 8Å of padding. If 3D coordinates are not available for the ligand, a single 3D conformer of the ligand is generated using RDKit[48] to provide the initial coordinates (using rdconf.py from https://github.com/dkoes/rdkit-scripts). A single conformer is sufficient since the docking algorithm will sample the degrees of freedom of the ligand. All docking is done against a rigid receptor that is stripped of water but not metal ions. Protonation states for both the ligand and receptor are determined using OpenBabel.[49]

Pose Prediction: CSAR—Our pose prediction training set is based on the CSAR-NRC HiQ dataset, with the addition of the CSAR HiQ Update.[50] This set consists of 466 ligand-bound co-crystals of distinct targets. To generate the training set, we re-docked these ligands with the settings –seed 0 –exhaustiveness 50 –num_modes 20 to thoroughly and reproducibly sample up to 20 distinct poses. We exclude targets where the ligand is annotated with a binding affinity of less than 5 pK units (a value provided as part of the CSAR dataset). This results in 337 co-crystals where the ligand has a reported binding affinity better than 10μM (where the affinity may come from a variety of sources, including IC50 measurements). For the purposes of training, poses with a heavy-atom RMSD less than 2Å from the crystal pose were labeled as positive (correct pose) examples and those with an RMSD greater than 4Å RMSD were labeled as negative examples. Poses with RMSDs between 2Å and 4Å were omitted. The final training set consists of 745 positive examples from 327 distinct targets and 3251 negative examples from 300 distinct targets (some targets produce only low or high RMSD poses).

Virtual Screening: DUD-E—Our virtual screening training set is based off the Database of Useful Decoys: Enhanced (DUD-E)[51] dataset. DUD-E consists of 102 targets, more than 20,000 active molecules, and over one million decoy molecules. Unlike the CSAR set, crystal poses of these ligands are not provided, although a single reference complex is made available. To generate poses for training, we dock against this reference receptor using smina's default arguments for exhaustiveness and sampling and select the pose that is top-ranked by the AutoDock Vina scoring function. Top-ranked poses are used both for the active and decoy compounds. The result is an extremely noisy and unbalanced training set. The noisiness stems from cross-docking ligands into a non-cognate receptor, which substantially reduces the retrieval rate of low-RMSD poses in a highly target-dependent manner,[1] as well as the use of randomly chosen decoys in DUD-E (the dataset may contain false negatives). The unbalance is due to the much larger number of decoy molecules. The final training set contains 22,645 positive examples and 1,407,145 negative examples.

Input Format

Traditionally, CNNs take images as inputs, where a scene is discretized into pixels with red, green, and blue values (RGB). To handle our 3D structural data, we discretize a protein-ligand structure into a grid. The grid is 24Å on each side and centered around the binding

site with a default resolution of 0.5Å, although we evaluate alternative resolutions. Each grid point stores information about the types of heavy atoms at that point. Ligand and protein atoms have distinct atom types and each atom type is represented in a different channel (analogous to RGB channels in images) of the 3D grid. Our default is to use smina[1] atom types for a total of 34 distinct types with 16 receptor types and 18 ligand types as shown in Table S1. Only smina atom types that were present in the ligands and proteins of the training set were retained. For example, halogens are not included as receptor atom types and metals are not included as ligand atom types. Hydrogen atoms are ignored except to determine acceptor/donor atom types. We also evaluate alternative atom typing schemes. Atom type information is represented as a density distribution around the atom center. We represent each atom as a function A(d, r) where d is the distance from the atom center and r is the van der Waals radius:

$$A(d,r) = \begin{cases} e^{-\frac{2d^2}{r^2}} & 0 \leq d < r \\ \frac{4}{e^2 r^2} d^2 - \frac{12}{e^2 r} d + \frac{9}{e^2} & r \leq d < 1.5r \\ 0 & d \geq 1.5r \end{cases} \quad (1)$$

A is a continuous piecewise combination of a Gaussian (from the center to the van der Waals radius) and a quadratic (which goes to zero at 1.5 times the radius). This provides a continuous representation of the input. We also evaluate a ‘hard’ discrete boolean representation.

We generate these grids of atom density using a custom, GPU-accelerated layer, MolGridDataLayer, of the Caffe[52] deep learning framework. This layer can process either standard molecular data files, which are read using OpenBabel,[49] or a compact, custom binary gninatypes file that contains only the atomic coordinates and pre-processed atom type information.

A visualization of our atom type volumetric representation is shown in Figure 2 with density data rendering using isosurfaces. This input format fully represents the spatial and chemical features of the protein-ligand complex; the sole approximations are the choice of grid resolution and the atom typing scheme.

Training

Our CNN models were defined and trained using the Caffe deep learning framework.[52] Training minimized the multinomial logistic loss of the network using a variant of stochastic gradient descent (SGD) and backpropagation. The order of training data was shuffled and classes were balanced by sampling the same number of positive examples as negative examples per batch. Additionally, our `MolGridDataLayer` has the ability to randomly rotate and translate the input structures on-the-fly. This feature is controlled via data augmentation parameters specifying whether to randomly rotate structures and the maximum distance to randomly translate them. Enabling this data augmentation significantly improved training, as shown in Figure 3.

The values for training hyperparameters were initially evaluated in ranges common for neural network training, and these values were verified to behave reasonably for our data. In general, training parameters within conventional ranges converged to similar loss values, with the main difference being the number of iterations needed to converge. The same parameters for the SGD solver (batch_size=10, base_lr=0.01, momentum=0.9), for learning rate decay (lr_policy = inverse, power=1, gamma=0.001), and for regularization (weight_decay=0.001, dropout_ratio=0.5) were used to train all models. In all cases we manually verified that model training had qualitatively converged after 10,000 iterations.

Model Evaluation

The performance of trained CNN models were evaluated by 3-fold cross-validation for both the pose prediction and virtual screening tasks. To avoid evaluating models on targets similar to those in the training set, training and test folds were constructed by clustering data based on target families rather than individual targets. For the CSAR pose prediction training set, clusters were created using the 90% sequence identity families provided by CSAR (i.e., protein targets with greater than 90% sequence identity are always retained in the same fold to avoid testing on a target highly similar to one in the training set). For the DUD-E virtual screening dataset, we created our own clusters of proteins using the hierarchical clustering module of scipy and ensured that proteins with greater than 80% sequence identity were retained in the same fold.

Receiver-operating characteristic (ROC) curves were generated for each scoring function, plotting the true positive rate against the false positive rate. The performance metric was the area under the ROC curve (AUC), with AUC = 1 representing a perfect classifier and AUC = 0.5 being no better than chance. For early enrichment, we report the ROC enrichment[53,54] at 0.5%, 1%, 2% and 5% false positive rate (FPR) thresholds. The ROC enrichment is the ratio of the true positive rate (TPR) to the FPR at a given FPR threshold; as the maximum TPR is 1.0, the maximum possible ROC enrichment depends on the chosen FPR threshold (e.g., at an FPR of 5%, the best possible ROC enrichment is 20). Random performance has an expected ROC enrichment of 1.0.

In addition to the default Vina scoring function, we also evaluate the ability of two machine learning scoring functions, RF-Score[13] and NNScore,[20] to separate actives from inactives in the virtual screening evaluation. In both cases we used published models;[55,56] no effort was made to compensate for overlap between the training sets used to create these models and our test set. These models were applied to all docked poses of a ligand and the best score was used to classify the ligand. For NNScore we took the average of the three provided models.

Independent Test Sets—To control for any systematic bias in the training sets, we also chose to assess classification accuracy on several completely independent test sets. To evaluate pose prediction performance, we utilized the 2013 PDBbind core set.[57] The PDBbind database consists of high quality protein-ligand complexes with no unusual atomic features, such as uncommon elements. The core set is a representative, non-redundant subset of the database and is composed of 195 protein-ligand complexes in 65 families.

To assess virtual screening performance, we utilized two datasets created from assay results. One was generated from ChEMBL by Riniker and Landrum,[58] following Heikamp and Bajorath.[59] They selected a set of 50 human targets from ChEMBL version 14. They chose actives that had at least 10 μM potency, had a molecular weight under 700 *g/mol*, and did not have metal ions. The actives were down sampled using the RDKit diversity picker to select the 100 most diverse compounds for each target. For each active, two decoys with a Dice similarity greater than 0.5 using a simple atom-count fingerprint (ECFC0) were randomly selected from the ZINC database to yield a total of 10,000 decoys that were shared across all targets. Our other virtual screening dataset is a subset of the maximum unbiased validation (MUV) dataset,[60] which is based on PubChem bioactivity data. MUV consists of assay data from 17 targets, each with 30 actives and 15000 decoys. Actives were selected from confirmatory screens and were chosen to be maximally spread based on simple descriptors and embedded in decoys. The decoys were selected from a primary screen for the same target. The MUV datasets were designed to avoid analog bias and artificial enrichment, which produce overly optimistic predictions of virtual screening performance.

To avoid artificially enhancing our performance on these test sets, we enforced a maximum similarity between targets included in the test sets and targets from DUD-E and CSAR used for training. We performed a global sequence alignment for all targets from the training and proposed test sets and removed any test targets that had more than 80% sequence identity with a training target. We also performed ProBiS[61] structural alignment on the binding sites of all pairs of targets from the training and proposed test sets and rejected those for which a significant alignment was found using the default ProBiS parameters. Finally, since structural data were necessary for scoring, assay targets were only included if a crystal structure of a ligand-bound complex containing the target was available in the Protein Data Bank. If multiple structures of active-site drug-like inhibitors were available, we selected the lexicographically first structure that had minimal mutations. This structure was used to generate docked poses at a known binding site. After these constraints were applied, the independent test sets consisted of a 54 complex subset of the 2013 PDBbind core set, a 13 target subset of the Riniker and Landrum ChEMBL set, and a 9 target subset of the MUV set.

For the pose prediction task, we re-docked ligands from the PDBbind core set with the settings –seed 0 –exhaustiveness 50 –num_modes 20 (the same settings used to generate poses for the CSAR training set). The resulting PDBbind core subset had 98 low RMSD (< 2Å) out of 897 total poses. For the virtual screening task, the active and decoy sets were docked against an appropriate reference receptor using smina's default arguments for exhaustiveness and sampling. All generated poses were scored and the best score for each ligand was used to assess virtual screening performance. The resulting ChEMBL subset had 11,406 poses associated with 1,300 active compounds and 663,671 poses associated with 10,000 decoys. The resulting MUV subset had 1,913 poses associated with 270 active compounds and 1,177,989 poses associated with 135,000 decoys.

The ChEMBL and MUV test sets provide collections of actives and decoys associated with a target protein, but they do not provide crystal structures for the target. We only included targets with bound crystal structures available, and we used the bound ligand to identify the

pocket into which to dock the assay's actives and decoys. Table S2 shows the PDB accession code for the crystal structure we used for each target, the bound ligand associated with that structure, that ligand's experimental affinity for the target (if available), and the type of assay used to identify the actives and decoys.

Optimization

An initial CNN architecture was constructed using simple guidelines in order to limit parameterization and serve as a starting point for optimization. The preliminary model architecture consisted of five 3×3×3 convolutional layers with rectified linear activation units alternating with max pooling layers. The number of filters in each convolutional layer was doubled from the previous one so that the width of the network increased as the spatial dimensionality decreased. Following the alternating convolution and pooling layers was a single fully connected layer with two outputs and a softmax layer for binary classification.

The various parameters of the neural network model were tuned to train the most accurate model with respect to the CSAR pose prediction test set. The CSAR set was chosen as its smaller size made iterative model optimization more practical. Model optimization was performed by systematically modifying a reference model. A single parameter was varied and the resulting training times and accuracies computed. After all parameters were tested, the changes resulting in the best gain of accuracy and similar or reduced training time were combined to create a new reference model. This process was repeated until the model's accuracy no longer increased. Several model parameters were explored.

Atom Types—In addition to the default smina atom types, we evaluated two simpler atom typing schemes: element-only and ligand/receptor only. Unlike smina atom types, which include aromaticity and protonation state information, element-only types only record the element, although we still provide distinct types for receptor and ligand atoms. With ligand/receptor only types, there are only two types (corresponding to two "channels" in the input 3D image): ligand atoms and receptor atoms.

Occupancy Type—In addition to a smoothed Gaussian distribution of atom density, we also evaluated a Boolean representation, where grid point values are one if they overlap an atom and zero otherwise. Unlike with the Gaussian scheme, in the Boolean representation individual grid point values provide no indication of the distance of the grid point from the atom center.

Atomic Radius Multiplier—By default, we extend atom densities beyond the van der Waals radius by a multiple of 1.5 (e.g., if the atomic radius is 1.0, the atom density decays to zero at 1.5). Additionally, we evaluated multiples of 1.0, 1.25, 1.75, and 2.0. With larger multiples, a single grid point contains more information about the local neighborhood.

Resolution—The default grid resolution is 0.5Å resulting in 48^3 grid points. We also evaluated higher (0.25Å) and lower (0.75, 1.0, and 1.5Å) resolution grids.

Layer Width—In our initial reference model, the first convolutional layer generates 128 feature maps, and each successive layer doubles the number of feature maps after halving

the dimensions of the maps with a pooling layer. We also evaluate models that double, half, and quarter the width of these layers. Wider layers allow for a more expressive model, but at the cost of more computation.

Model Depth—Our initial model contained 5 convolution layers. We also evaluate models with more (up to 8) and fewer (as little as 1) convolution layers. More layers allow for a more expressive model, but take longer to process and increase the risk of suffering from vanishing gradients, which inhibit convergence.[62]

Pooling Type—Pooling layers reduce the size of their inputs by propagating a single value for each window (or kernel) of the input. The propagated value can either be the maximum value or the average value of the kernel and the kernel size can be varied. In our initial model we use max pooling with a kernel size of 2. We additionally evaluate average pooling and kernels of size 4.

Fully Connected Layer—After a series of convolution and pooling layers, a traditional fully connected layer reduces the final feature maps to two outputs. Our initial model contains a single fully connected layer with no hidden nodes. Additionally, we evaluate alternative models with a single hidden layer with anywhere from 6 to 50 nodes. More expressive fully connected layers allow the model to arbitrarily combine the spatial features generated by the convolution layers to generate the final prediction.

Visualization

In order to better understand the features that the neural network learns, we implemented a visualization algorithm based on masking.[63] In image recognition masking, pixels are systematically masked out and the image is reclassified in order to get a “heat map” of important areas. The visualization algorithm is illustrated in Figure 4. Atoms are colored by relative contribution to the total neural network score as determined by removing the atom and rescoring the complex.

Atoms are removed either one at a time, or as part of larger fragments. The individual and fragment removals of atoms differ significantly enough that an average of both scores is computed. The individual removals produce sharper contrasts between “good” and “bad”, compared to a more gradual effect in the fragment removals. The combination of the two methods provides a broader representation of how the model interprets functional groups, while maintaining any significant individual atom scores.

In order to reduce computational load, removals were carried out on whole residues of the protein at a time. This provided enough information to assess spatial relationships between protein and ligand, which is a key goal of visualization.

Results

Our systematic optimization of network and training parameters successfully improved the performance of the CNN models in clustered cross-validation while revealing the importance, or lack thereof, of various choices of parameters. We evaluated the optimized

network architecture for performance in pose prediction, virtual screening, and affinity prediction, while also considering the importance of the training set used to create the model.

Optimization

Two rounds of model optimization were performed. In each round, parameters of a reference model were individually varied. For each parameter type, the best parameter was used to define the reference model of the next iteration. Each iteration both increased the cross-validation AUC and decreased the training time of the model. The results obtained in the first two iterations are shown in Figure 5. A third iteration did not result in further improvements (data not shown). The initial reference model had an AUC of 0.78 and a training time of 580ms per an iteration, and the final model increased to an AUC of 0.82 with a training time of 120ms per an iteration. The ROC curves for all three models are shown in Figure S1.

Based on the first iteration of parameter sensitivity analysis, the second reference model reduced the depth from five to four convolutional layers and quartered the widths of these layers. After another round of optimization, the final reference model further reduced the depth to three convolutional layers. The final optimized network architecture is shown in Figure 6. Since parameters were varied individually in each optimization iteration, we can assess the relative importance of each parameter class on the overall model performance.

Atom Types—The best AUCs are achieved using smina atom types. However, simpler atom types are remarkably competitive, with at most a 0.05 reduction in AUC for the binary protein/ligand atom typing in the second iteration of optimization. Although this is consistent with previous findings with empirical scoring functions where purely steric terms were found to be the dominant terms of the scoring function,[1,64] it is likely that the model is inferring atom types from the atomic radii. When a single radius is used for all elements, the binary protein/ligand atom typing AUC drops by an additional 0.11. We also note that, although the overall AUCs were similar, smina and element-only atom types result in better early enrichment (the initial slope of the ROC curve is steeper).

Occupancy Type—Interestingly, changing the atom density representation from the more informative Gaussian to a simple Boolean did not reduce the AUC. The models do not seem to need the additional distance information provided by a Gaussian atomic density.

Atomic Radius Multiplier—The default radius multiplier of 1.5 provided the best AUC, although other multipliers were nearly equivalent with all but the 2.0 multiplier within 0.01 of the reference AUC.

Resolution—Predictive performance correlates with resolution, with the highest resolution (0.25Å) achieving an AUC more than 0.1 greater than the lowest (1.5Å). However, we decided against using higher resolution grids since the small increase in AUC (0.02) in increasing the resolution from 0.5Å to 0.25Å was accompanied by a more than 4X increase in per-iteration training time, which directly correlates with the evaluation time of the final

model (i.e., the final model would have 4X less throughput when performing a virtual screen).

Layer Width—We found that increasing the width of the layers resulted in significant increases in training time, but slight decreases in predictive performance, possibly due to overfitting. Reducing the width improved both the AUC and training time up to a limit. In our final model, the first convolutional layer generates 32 feature maps; reducing this number further hurts predictive performance.

Model Depth—Model depth behaved similarly to the layer width parameter. Our initial model topology was needlessly expressive, and by reducing the depth (ultimately to only three convolutional layers), we improved both training time and predictive performance, likely by reducing the amount of overfitting.

Pooling Type—Somewhat surprisingly, the use of average pooling instead of max pooling obliterated predictive performance and prevented the model from learning. Alternative kernel sizes did not improve the AUC.

Fully Connected Layer—Modifications to the final fully connected layer had no discernible effects on predictive performance or training time, suggesting most of the learning is taking place in the convolutional layers.

The final optimized model architecture was used to train and evaluate pose prediction, virtual screening, and affinity prediction performance. It is available at https://github.com/gnina/models.

Pose Prediction

Pose prediction assesses the ability of a scoring function to distinguish between low RMSD and high RMSD poses of the same compound. We assess pose prediction performance both in terms of inter-target ranking and intra-target ranking. With inter-target ranking, which is most similar to the training protocol, all poses across all targets are ranked to generate a ROC curve. Intra-target ranking better represents the typical docking scenario, and the goal is to select the the lowest RMSD pose among poses generated for each individual target. A scoring function can do well in intra-target ranking even if the low RMSD pose has a poor score as long as all other poses for that target have worse scores.

The CNN model performed substantially better than the Autodock Vina scoring function in its ability to perform inter-target ranking of CSAR poses as shown by the cross-validation results in Figure 7. The CNN model achieves an AUC of 0.815 while the Vina scoring function has an AUC of 0.645.

In intra-target ranking, the CNN model performed substantially worse than Autodock Vina, as shown in Figure 8. The Autodock Vina scoring function is parameterized to excel at redocking[1,7] and correctly identifies a low RMSD pose as the top ranked pose for the given target for 84% of the targets compared to 64% with the CNN model. When the top 5 poses are considered, the difference between Vina and the CNN model shrinks with Vina

exhibiting a success rate of 93% and the CNN model 92%. As pose selection performance is dependent on the range of poses that are selected from (e.g., some targets have highly rigid ligands in tightly constrained pockets resulting in nearly all low RMSD poses), we also show the results of random selection in Figure 8. Both methods are substantially better than random.

The correlations between pose RMSD and scores are shown in Figure S3. The CNN scores weakly correlate with RMSD, with higher RMSD poses exhibiting lower scores as expected (a more positive CNN score is more favorable). Vina scores do not correlate with RMSD, although there is a noticeable "funnel" shape due to the best scoring poses having very low RMSDs. Interestingly, there is no correlation between CNN scores and Vina scores, indicating that they use different criteria to rank poses.

Virtual Screening

Structure-based virtual screening assesses the ability of a scoring function to distinguish between active and inactive compounds using docked structures. In assessing virtual screening, we consider both the case where the CNN model ranks only the top-ranked (by Vina) docked pose of each ligand (single-pose prediction) and the case where the CNN model selects from all available docked poses of the ligand (multi-pose prediction).

Overall cross-validation results for the entire DUD-E benchmark are shown in Figures 9 and S4 and Tables 1 and S3. Early enrichment performance is shown in Figure 10 and Tables S4–S7. Even using the exact same poses (single-pose scoring), CNN scoring substantially outperforms Vina with an overall AUC of 0.85 versus 0.68. Multi-pose scoring does slightly better with an AUC of 0.86. In terms of early enrichment, the CNN model is 2–4 times better than Vina on average (Table 1 and Figure 10). On a per-target basis, CNN scoring outperforms Vina scoring for 90% of the DUD-E targets, as shown in Figure 11. As shown in Figures 9 and 10, the CNN models and Vina both outperform the alternative machine learning scoring functions evaluated. It is possible these methods are at a disadvantage due to having been trained on structures created with a different pose generation protocol.

Combined Training

CNN models trained on one kind of data do not generalize particularly well to another. For example, as shown in Figure 12, a CNN model trained exclusively on DUD-E data achieves a cross-validation AUC of 0.56 in CSAR pose prediction. This is not unexpected as the DUD-E training data consists of noisy, likely inaccurate, docked poses. A CNN model trained on this data will be less sensitive to changes in ligand pose. In the other direction, training on CSAR data resulted in a cross-validation AUC of 0.66 at the virtual screening task. However, as shown in Figure 12, combining CSAR and DUD-E training data results in models that perform nearly as well as single-task trained models. At a ratio 2:1 DUD-E to CSAR (for every two virtual screening training examples from DUD-E, one pose prediction example from CSAR is included during training), the resulting CNN model exhibits an AUC of 0.79 at pose prediction and an AUC of 0.83 at virtual screening. The inclusion of pose prediction training data accentuates the difference between single-pose and multi-pose

DUDE evaluation (e.g., 0.79 vs 0.83 at a 2:1 ratio), suggesting that such data allows the CNN model to select more accurate poses.

Although a combined training set results in a minimal reduction in overall AUC for DUD-E, on a per-target basis, shown in Figure 11, there is a more significant reduction in performance, with only 81% of targets performing better than Vina, compared with 90% with a DUD-E-only training set. Early enrichment (Figure 10) is also reduced, although still significantly better than Vina on average (Table 1). In a few cases, there is a dramatic performance difference, such as with the fpps (farnesyl diphosphate synthase) target which goes from a 0.98 AUC with the DUD-E-only training set to a 0.10 AUC with the combined 2:1 training set. This target is also a challenge for the Vina scoring function, which also achieves a worse-than-random 0.29 AUC, suggesting that the generated poses may be highly inaccurate.

An example ligand from the fpps target is CHEMBL457424, which the DUD-E model scores as 0.99 but the combined model scores as 0.01. The DUD-E model is completely pose insensitive - all poses of this ligand score similarly despite large differences in RMSD. The pose selected with Vina is shown visualized with the DUD-E model in Figure 13. This pose is most likely incorrect; based on the 3Z0U crystal structure, the bisphosphonate group should chelate with the magnesium ions. The DUD-E model highlights the polar and aromatic parts of the molecule and disfavors the apolar parts. It also highlights the polar residues of the binding site. It is possible that the DUD-E-only model is simply ranking polar molecules highly, having recognized the highly polar binding site. Furthermore, all the actives associated with this target in the DUD-E benchmark contain a bisphosphonate group, whereas fewer than 1% of the decoy compounds even contain phosphorous. A scoring function that favors this group regardless of the 3D interaction structure will do exceptionally well in scoring these actives. When pose quality is incorporated into the training of the model, as with the combined model, erroneous poses are penalized and non-structural properties, such as polarity, play a less dominant role. Similar trade-offs between learning non-structural cheminformatic information and enforcing structural constraints likely explain the difference in performance between the DUD-E and combined models.

Independent Test Sets

To evaluate CNN scoring performance on our independent test sets, we trained three models using all folds of the available training data: a pose prediction model trained only on CSAR data, a virtual screening model trained only on DUD-E data, and a combined model trained on DUD-E and CSAR data at a 2:1 ratio.

Pose Prediction—Summary results for the PDBbind core set are shown in Figures 14, 15, 16 and S2. As with the cross-validation results, the CNN models outperform Vina in an inter-target assessment of pose ranking (Figure 14) with an improvement of about 0.1 AUC. Also consistent with the cross-validation results is the Ending that, on average, Vina's top-ranked pose has a lower RMSD than the top-ranked poses of any of the CNN methods, but by the second ranked pose the CSAR and DUD-E/CSAR combined CNN models, both of which were trained on pose prediction data, have improved on Vina (Figure S2). As

expected, the model trained on DUD-E data, which consisted of inaccurate docked poses, does poorly at pose prediction.

The distribution of best RMSD values at different ranks is shown in Figure 15. Even for the poorly performing DUD-E-only model there is a significant cluster of low RMSD poses. The percentage of complexes where a low RMSD pose (< 2Å) was found in the top N ranked poses for each method is shown in Figure 16. The DUD-E trained model had similar performance to random pose selection, providing further evidence for the conclusion that models trained on this kind of data lack pose sensitivity. The models trained with pose prediction data did significantly better than random, with the CSAR-trained model correctly identifying a low RMSD pose as the top ranked pose in 46% of the complexes, compared to 57% for Vina. As with the cross-validation results, accuracy improved significantly as the number of top ranked poses considered increased. The combined DUD-E/CSAR model outperformed Vina at identifying a low RMSD pose within the first three ranked poses.

Examples of PDBbind poses visualized with the CSAR model are shown in Figures 17 and 18. Figure 17 shows human protein kinase CK2 (PDB 3PE2). For this complex, Vina correctly top-ranks a low RMSD pose while the CNN model prefers to flip the compound in the binding site. The visualization illustrates why. The CNN model correctly favors the binding of the low RMSD pose to the hinge region of the kinase (as indicated by the green highlighting on both the ligand and protein in this region), but it disfavors the position of the alkynyl. Although flipping the compound results in less favorable interactions with the hinge region, it results in what the model considers to be a better pose of the alkynyl. Figure 18 shows an Aurora A kinase (PDB 3MYG). In this case, the CNN model correctly top-ranks a low RMSD pose while Vina prefers a pose that is flipped and more buried in the binding site. Again, the model highlights the interactions with the hinge region of the kinase. While the model slightly disfavors the solvent exposed portion of the compound, flipping the compound and burying this portion of the compound in the interior of the kinase is more strongly disfavored (as indicated by the red highlighting).

Virtual Screening—Virtual screening results for the ChEMBL and MUV independent test sets are shown in Figures 19–24, S5, and S6 and Tables 2, 3, and S8–S17. The ChEMBL and MUV tests sets are more challenging than the DUD-E benchmark for all methods. Averaging across targets, the AUCs for the ChEMBL benchmark are 0.67, 0.64, and 0.78 for the Vina, 2:1 DUD-E/CSAR CNN, and DUD-E CNN methods, which is consistently lower than the corresponding average cross-validation AUCs on DUD-E: 0.71, 0.80, and 0.86. Average values for ROC enrichment are shown in Table 2 and track the AUC means. Per-target values are shown in Table S5. The distributions of AUCS and ROC enrichments across targets are shown in Figures 19 and 20. Overall ROC curves (where the test set is evaluated as a whole) are shown in Figure 24. Consistent with previously reported results,[58,65] the MUV set is even more challenging, with average AUCs of 0.55, 0.50, and 0.52 for Vina, 2:1 DUD-E/CSAR CNN, and DUD-E CNN. Unlike the cross-validation results (Figure 12), the CSAR-trained CNN has close to random performance at virtual screening for most targets (Figures 21, 22, 24, and Tables 3 and S6).

Consistent with the cross-validation results, the DUD-E-trained CNN model generally outperforms the DUD-E/CSAR combined model and the alternative machine learning models, although RF-Score does better in this evaluation than when targeting DUD-E. Since the ChEMBL and MUV sets were constructed using a methodology that differs from the DUD-E benchmark, this suggests that the DUD-E CNN model is learning genuinely useful information about features of the ligand and protein binding site that are relevant to binding, despite a lack of pose sensitivity, and is not learning solely an artifact of the construction of the DUD-E set. Interestingly, as shown in Figure 23, the targets with the biggest drop in performance between the DUD-E model and the pose sensitive DUD-E/CSAR model are also some of the targets with the lowest Vina performance. This would be the expected effect if docking is failing to sample accurate poses, as in this case a more cheminformatic-oriented, pose insensitive model would perform better.

The MUV benchmark is particularly challenging, with no method achieving an AUC greater than 0.6 on more than two targets. The overall performance across the benchmark is essentially random for all methods, as shown in Figure 24. Unlike with the ChEMBL set (Figure S5), in MUV the few individual targets where methods do appreciably better than random the improvement in AUC is not driven by early enrichment (Figure S6). The use of cell-based assays and the lack of structures bound to ligands of known affinity (Table S2) may make MUV a poor choice for a structure-based virtual screening assessment. Alternatively, the observed poor performance may be due to the method MUV uses to construct the active and decoy sets, which attempts to avoid analog bias and artificial enrichment by ensuring that actives are well embedded in the chemical space of the decoys. The MUV target 466, a lipid G protein-coupled receptor, is identical to ChEMBL target 11631, and we used the same structure, PDB 3V2Y, to generate poses. This allows us to compare the effect of the different decoy construction approaches between the two benchmarks. As shown in Table 4, for all methods, the highest performance is achieved with the ChEMBL actives. This suggests, for this target at least, that the method used to construct the decoys is not the cause of the observed poor performance and that the performance observed on the ChEMBL set is not due to artificial enrichment.

Visualization

Visualization is intended to provide a qualitative and easy to interpret indication of the atomic features that are driving the CNN model's output. In order to more quantitatively assess the utility of our visualization approach, we considered single-residue protein mutation data and partially aligned poses.

Mutation Analysis—The Platinum[66] database provides measured differences in protein-ligand binding affinity upon mutation of single receptor residues. This experimental technique is a close analogue of the visualization algorithm, where whole residues are removed and the complex re-scored. For our assessment, we filtered the database to consider only experiments with single mutations to alanine or glycine in proteins that are not present in our training data and evaluated those with the largest changes in binding affinity.

The CNN was able to identify critical residues in many of the examples that were tested. The three protein-ligand pairs with the highest changes in binding affinity are shown in Figure 25. In all three cases, many residues had heavy green coloring, and the mutant residue is always colored green. Other highlighted residues may also be critical, but were not present in the Platinum database. It is worth emphasizing that the CNN model was not trained on protein mutational data. The fact that critical residues are highlighted suggests that the model is learning some general underlying model of the key features of protein-ligand interactions.

Partially Aligned Poses—We identified the high RMSD (> 4Å) docked poses in the core PDBbind dataset that had the highest percentage of aligned atoms (< 0.1Å distant to the corresponding crystal atoms). These are poses that are partially correct; part of the molecule matches the crystal and part does not.

The five poses with the highest percentage of congruent atoms are shown visualized using the 2:1 DUD-E/CSAR model in Figure 26. For all five poses, the CNN model ranks the crystal pose higher than the docked pose. Our visualization shows why these poses are scored lower. In all cases, the part of the docked pose that is aligned to the crystal pose is predominantly or entirely green (indicating positive contributions), but the divergent part of the ligand is entirely or partially red (indicating negative contributions).

Discussion

We have provided the first detailed description and evaluation of applying deep learning and convolutional neural networks to score protein-ligand interactions using a direct, comprehensive 3D depiction of the complex structure as input. By several metrics, our CNN models outperform alternative approaches, in particular the Autodock Vina empirical scoring function and the RF-Score and NNScore machine learning scoring functions. In inter-target evaluations of pose prediction, both using cross-validation and an independent test set, CNN models can perform substantially better (e.g., Figures 7 and 14). Likewise, CNN models can do well in virtual screening evaluations (e.g., Figures 11 and 23). However, our results also point to weaknesses in the current method and opportunities for improvement.

Although the CNN models performed well in an inter-target pose prediction evaluation, they performed worse at intra-target pose ranking (e.g., Figures 8, 15, and S2), which is more relevant to molecular docking. It is likely that intra-target ranking could be improved by changing the training protocol to more faithfully represent this task. For example, currently ligands are treated identically regardless of their affinity, as long as they fall below a threshold (10μM). It is conceivable that a high RMSD pose of a high affinity ligand should legitimately be scored better than a low RMSD pose of a low affinity ligand, a distinction the current training protocol cannot make. Incorporating the binding affinity as a component of training, or performing relation classification,[67] which assesses the ability of the network to *rank* rather than score poses, may significantly improve intra-target performance of CNN models.

Our models perform well in a clustered cross-validation evaluation of virtual screening on the DUD-E benchmark. However, this benchmark may be susceptible to artificial enrichment,[37] resulting in overly optimistic predictions of virtual screening performance. We believe that our use of clustered cross-validation, which not only avoids training on ligands of the same target but also all similar targets, should mitigate some of the artificial enrichment issues inherent in DUD-E. Furthermore, our independent test sets both used an entirely different method of dataset construction than the DUD-E set.

Ideally the CNN models learn a generalizable model of protein-ligand binding from the training data. However, our models' ability to generalize beyond the task inherent in the training data, while present, is limited (e.g. Figure 12). This is further highlighted by that fact that our CNN scores do not correlate ($|R| < 0.1$) with binding affinity data when evaluating the CSAR crystal poses. In contrast, Vina exhibits a modest correlation ($R = 0.37$) on the same benchmark. That is, training to classify poses and active/inactive compounds does not generalize to the regression problem of binding affinity prediction. We expect that CNN models trained on binding affinity data would provide substantially improved results on this task. Furthermore, our experience training combined pose prediction and virtual screening models indicates that multiple data types can be integrated to generate effective multi-task models. Unfortunately, we have not yet observed instances where including multi-task training data resulted in a synergistic effect, improving the performance of all tasks, although such an effect has been observed in other domains.[37]

In total, we believe that the current work demonstrates the potential of convolutional neural network models of protein-ligand binding to outperform current methods. There remain many possible avenues for improving CNN models, such as training with larger datasets spanning a range of objectives (e.g. pose ranking, affinity prediction, virtual screening, etc.) related to ligand binding. In order to aid in the development of more robust and higher performance CNN models, all of our code and models are available under an open source license as part of our gnina molecular docking software at https://github.com/gnina.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Justin Spiriti and Alec Helbling for their input on the manuscript. This work is supported by R01GM108340 from the National Institute of General Medical Sciences, by the National Institutes of Health training grant T32 EB009403 as part of the Howard Hughes Medical Institute—National Institute of Biomedical Imaging and Bioengineering Interfaces Initiative, by a GPU donation from the NVIDIA corporation, and by the TECBio REU@Pitt program which is supported by the National Science Foundation under Grant DBI-1263020 and is co-funded by the Department of Defense.

References

1. Koes DR, Baumgartner MP, Camacho CJ. Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. J Chem Inf Model. 2013; 53:1893–1904. [PubMed: 23379370]

2. Eldridge MD, Murray CW, Auton TR, Paolini GV, Mee RP. Empirical Scoring Functions: I. The Development of a Fast Empirical Scoring Function to Esti mate the Binding Affinity of Ligands in Receptor Complexes. J Comput-Aided Mol Des. 1997; 11:425–45. [PubMed: 9385547]

3. Böhm HJ. The Development of a Simple Empirical Scoring Function to Estimate the Binding Constant for a Protein-Ligand Complex of Known Three-Dimensional Structure. J Comput-Aided Mol Des. 1994; 8:243–256. [PubMed: 7964925]

4. Wang R, Lai L, Wang S. Further Development and Validation of Empirical Scoring Functions for Structure-Based Binding Affinity Prediction. J Comput-Aided Mol Des. 2002; 16:11–26. [PubMed: 12197663]

5. Korb O, Stützle T, Exner TE. Empirical Scoring Functions for Advanced Protein-Ligand Docking with PLANTS. J Chem Inf Model. 2009; 49:84–96. [PubMed: 19125657]

6. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, et al. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. J Med Chem. 2004; 47:1739–49. [PubMed: 15027865]

7. Trott O, Olson AJ. AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. J Comput Chem. 2010; 31:455–461. [PubMed: 19499576]

8. Huang SY, Zou X. Mean-Force Scoring Functions for Protein-Ligand Binding. Annu Rep Comp Chem. 2010; 6:280–296.

9. Muegge I, Martin YC. A General and Fast Scoring Function for Protein-Ligand Interactions: A Simplified Potential Approach. J Med Chem. 1999; 42:791–804. [PubMed: 10072678]

10. Gohlke H, Hendlich M, Klebe G. Knowledge-Based Scoring Function to Predict Protein-Ligand Interactions. J Mol Biol. 2000; 295:337–356. [PubMed: 10623530]

11. Zhou H, Skolnick J. GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction. Biophys J. 2011:2043–52.

12. Mooij WT, Verdonk ML. General and Targeted Statistical Potentials for Protein-Ligand Interactions. Proteins. 2005; 61:272–87. [PubMed: 16106379]

13. Ballester PJ, Mitchell JBO. A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking. Bioinformatics. 2010; 26:1169. [PubMed: 20236947]

14. Ashtawy HM, Mahapatra NR. Machine-Learning Scoring Functions for Identifying Native Poses of Ligands Docked to Known and Novel Proteins. BMC Bioinf. 2015; 16:1–17.

15. Sato T, Honma T, Yokoyama S. Combining Machine Learning and Pharmacophore-Based Interaction Fingerprint for In Silico Screening. J Chem Inf Model. 2009; 50:170–185.

16. Zilian D, Sotriffer CA. SFCscore RF: A Random Forest-Based Scoring Function for Improved Affinity Prediction of Protein-Ligand Complexes. J Chem Inf Model. 2013; 53:1923–1933. [PubMed: 23705795]

17. Jorissen RN, Gilson MK. Virtual Screening of Molecular Databases Using a Support Vector Machine. J Chem Inf Model. 2005; 45:549–561. [PubMed: 15921445]

18. Schietgat L, Fannes T, Ramon J. Predicting Protein Function and Protein-Ligand Interaction with the 3D Neighborhood Kernel. Discovery Science. 2015:221–235.

19. Deng W, Breneman C, Embrechts MJ. Predicting Protein-Ligand Binding Affinities using Novel Geometrical Descriptors and Machine-Learning Methods. J Chem Inf Comput Sci. 2004; 44:699–703. [PubMed: 15032552]

20. Durrant JD, McCammon JA. NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein- Ligand Complexes. J Chem Inf Model. 2010; 50:1865–1871. [PubMed: 20845954]

21. Chupakhin V, Marcou G, Baskin I, Varnek A, Rognan D. Predicting Ligand Binding Modes from Neural Networks Trained on Protein-Ligand Interaction Fingerprints. J Chem Inf Model. 2013; 53:763–772. [PubMed: 23480697]

22. Durrant JD, McCammon JA. NNScore 2.0: A Neural-Network Receptor-Ligand Scoring Function. J Chem Inf Model. 2011; 51:2897–2903. [PubMed: 22017367]

23. Durrant JD, Amaro RE. Machine-Learning Techniques Applied to Antibacterial Drug Discovery. Chem Biol Drug Des. 2015; 85:14–21. [PubMed: 25521642]
24. Gonczarek A, Tomczak JM, Zaręba S, Kaczmar J, Dąbrowski P, Walczak MJ. Learning Deep Architectures for Interaction Prediction in Structure-based Virtual Screening. arXiv preprint: 1610.07187. 2016
25. Wallach I, Dzamba M, Heifets A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery. arXiv preprint:1510.02855. 2015
26. Kramer C, Gedeck P. Leave-Cluster-Out Cross-Validation Is Appropriate for Scoring Functions Derived from Diverse Protein Data Sets. J Chem Inf Model. 2010; 50:1961–1969. [PubMed: 20936880]
27. Gabel J, Desaphy J, Rognan D. Beware of Machine Learning-Based Scoring Functions - On the Danger of Developing Black Boxes. J Chem Inf Model. 2014; 54:2807–2815. [PubMed: 25207678]
28. Li H, Leung KS, Wong MH, Ballester PJ. The Importance of the Regression Model in the Structure-Based Prediction of Protein-Ligand Binding. International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics. 2014:219–230.
29. Rojas, R. Neural Networks: A Systematic Introduction. Springer Science & Business Media; Berlin: 2013.
30. LeCun Y, Bengio Y, Hinton G. Deep Learning. Nature. 2015; 521:436–444. [PubMed: 26017442]
31. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res. 2014; 15:1929–1958.
32. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Van-houcke V, Rabinovich A. Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015:1–9.
33. Durrant JD, Carlson KE, Martin TA, Offutt TL, Mayne CG, Katzenel-lenbogen JA, Amaro RE. Neural-Network Scoring Functions Identify Structurally Novel Estrogen-Receptor Ligands. J Chem Inf Model. 2015; 55:1953–1961. [PubMed: 26286148]
34. Xu Y, Dai Z, Chen F, Gao S, Pei J, Lai L. Deep Learning for Drug-Induced Liver Injury. J Chem Inf Model. 2015; 55:2085–2093. [PubMed: 26437739]
35. Lusci A, Pollastri G, Baldi P. Deep Architectures and Deep Learning in Chemoin-formatics: The Prediction of Aqueous Solubility for Drug-Like Molecules. J Chem Inf Model. 2013; 53:1563–1575. [PubMed: 23795551]
36. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP. Convolutional Networks on Graphs for Learning Molecular Fingerprints. Advances in Neural Information Processing Systems. 2015:2224–2232.
37. Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V. Massively Multitask Networks for Drug Discovery. arXiv preprint:1502.02072. 2015. http://arxiv.org/abs/1502.02072v1
38. Smith JS, Isayev O, Roitberg AE. ANI-1 An Extensible Neural Network Potential with DFT Accuracy at Force Field Computational Cost. Chem Sci. 2017
39. Mills K, Spanner M, Tamblyn I. Deep Learning and the Schrödinger Equation. arXiv preprint: 1702.01361. 2017
40. Carleo G, Troyer M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. Science. 2017; 355:602–606. [PubMed: 28183973]
41. Krizhevsky A, Sutskever I, Hinton GE. Imagenet Classification with Deep Convo-lutional Neural Networks. Advances in Neural Information Processing Systems. 2012:1097–1105.
42. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:770–778.
43. Karpathy, A. What I Learned from Competing Against a Con-vNet on ImageNet. http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/, accessed Feb 20, 2017
44. Friesner RA, Murphy RB, Repasky MP, Frye LL, Greenwood JR, Halgren TA, Sanschagrin PC, Mainz DT. Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for Protein-Ligand Complexes. J Med Chem. 2006; 49:6177–6196. [PubMed: 17034125]

45. Jain AN. Scoring Noncovalent Protein-Ligand Interactions: A Continuous Differ entiable Function Tuned to Compute Binding Affinities. J Comput-Aided Mol Des. 1996; 10:427–40. [PubMed: 8951652]
46. Gathiaka S, Liu S, Chiu M, Yang H, Stuckey JA, Kang YN, Delproposto J, Kubish G, Dunbar JB, Carlson HA, et al. D3R Grand Challenge 2015: Evaluation of Protein-Ligand Pose and Affinity Predictions. J Comput-Aided Mol Des. 2016; 30:651–668. [PubMed: 27696240]
47. Damm-Ganamet KL, Smith RD, Dunbar JB, Stuckey JA, Carlson HA. CSAR Benchmark Exercise 2011–2012: Evaluation of Results from Docking and Relative Ranking of Blinded Congeneric Series. J Chem Inf Model. 2013; 53:1853–1870. [PubMed: 23548044]
48. RDKit: Open-Source Cheminformatics. http://www.rdkit.org, accessed September 4, 2015
49. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An Open Chemical Toolbox. J Cheminf. 2011; 3:33.
50. Dunbar JB, Smith RD, Yang C-Y, Ung PM-U, Lexa KW, Khazanov NA, Stuckey JA, Wang S, Carlson HA. CSAR Benchmark Exercise of 2010: Selection of the Protein-Ligand Complexes. J Chem Inf Model. 2011; 51:2036–2046. [PubMed: 21728306]
51. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. J Med Chem. 2012; 55:6582–94. [PubMed: 22716043]
52. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093. 2014
53. Jain AN, Nicholls A. Recommendations for Evaluation of Computational Methods. J Comput-Aided Mol Des. 2008; 22:133–139. [PubMed: 18338228]
54. Nicholls A. What Do We Know and When Do We Know It? J Comput-Aided Mol Des. 2008; 22:239–255. [PubMed: 18253702]
55. Lee, H. RF-Score. https://github.com/HongjianLi/RF-Score, accessed Feb 2, 2017
56. Durrant, J. NNScore 1.0. 2011. https://sourceforge.net/projects/nnscore/accessed Feb 2, 2017
57. Wang R, Fang X, Lu Y, Yang C-Y, Wang S. The PDBbind Database: Methodologies and Updates. J Med Chem. 2005; 48:4111–4119. [PubMed: 15943484]
58. Riniker S, Landrum GA. Open-Source Platform to Benchmark Fingerprints for Ligand-Based Virtual Screening. J Cheminf. 2013; 5:1–17.
59. Heikamp K, Bajorath J. Large-Scale Similarity Search Profiling of ChEMBL Compound Data Sets. J Chem Inf Model. 2011; 51:1831–1839. [PubMed: 21728295]
60. Rohrer SG, Baumann K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. J Chem Inf Model. 2009; 49:169–84. [PubMed: 19434821]
61. Konc J, Janežič D. ProBiS Algorithm for Detection of Structurally Similar Protein Binding Sites by Local Structural Alignment. Bioinformatics. 2010; 26:1160–1168. [PubMed: 20305268]
62. Bengio Y, Simard P, Frasconi P. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks. 1994; 5:157–166. [PubMed: 18267787]
63. Szegedy C, Toshev A, Erhan D. Deep Neural Networks for Object Detection Advances in Neural Information Processing Systems. 2013:2553–2561.
64. Novikov FN, Zeifman AA, Stroganov OV, Stroylov VS, Kulkov V, Chilov GG. CSAR Scoring Challenge Reveals the Need for New Concepts in Estimating Protein-Ligand Binding Affinity. J Chem Inf Model. 2011; 51:2090–2096. [PubMed: 21612285]
65. Tiikkainen P, Markt P, Wolber G, Kirchmair J, Distinto S, Poso A, Kallioniemi O. Critical Comparison of Virtual Screening Methods against the MUV Data Set. J Chem Inf Model. 2009; 49:2168–2178. [PubMed: 19799417]
66. Pires DE, Blundell TL, Ascher DB. Platinum: A Database of Experimentally Measured Effects of Mutations on Structurally Defined Protein-Ligand Complexes. Nucleic Acids Res. 2015; 43:D387–D391. [PubMed: 25324307]
67. Santos, CNd, Xiang, B., Zhou, B. Classifying Relations by Ranking with Convolutional Neural Networks. arXiv preprint:1504.06580. 2015

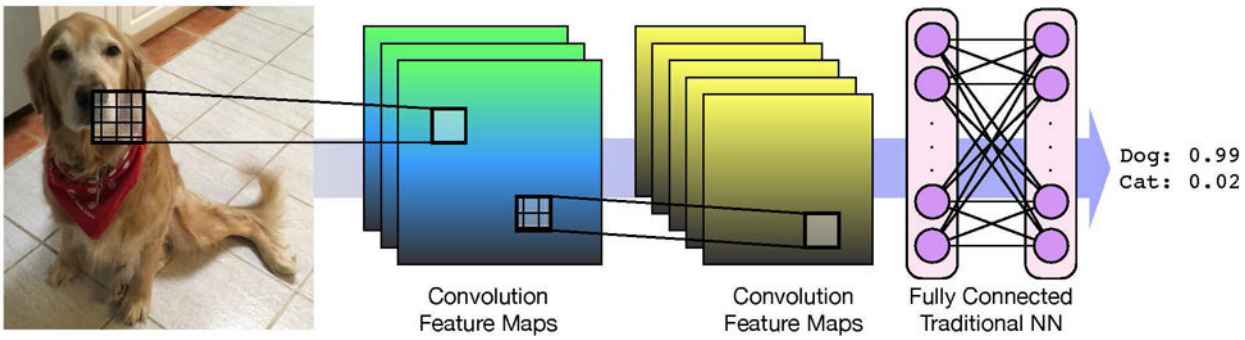


Figure 1.

A classical convolutional neural network for image recognition. The first layer applies three different convolutions to the input image to create three maps of low level features that are the input for another convolutional layer that creates five maps. Feature maps preserve the spatial locality of the features. As a last step, a traditional neural net is applied to generate a classification.

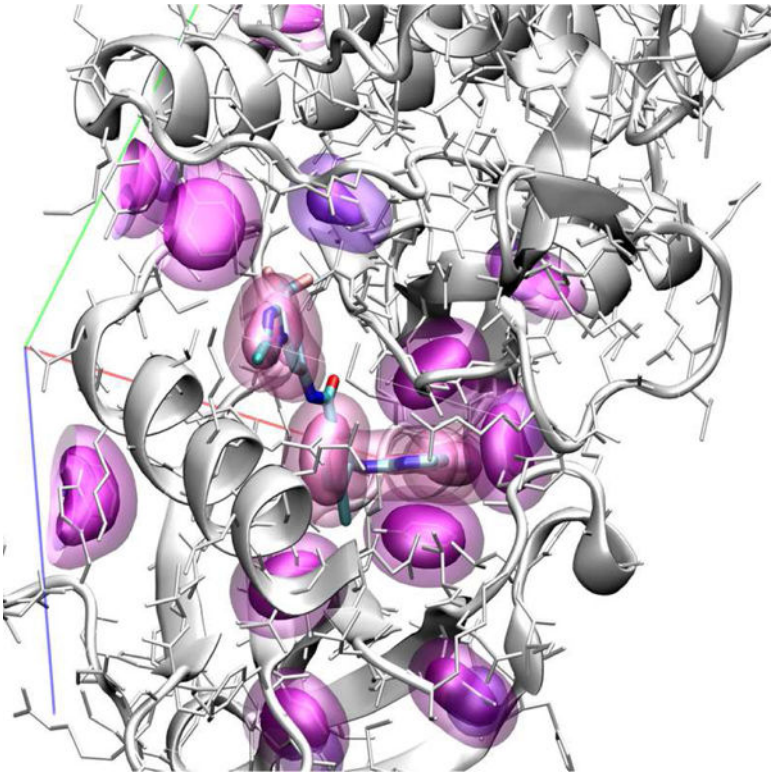


Figure 2.
Visualization of atom densities used as input to CNN scoring. Aromatic carbon atom densities are shown at two isosurface levels (solid and transparent surfaces) for both the receptor (purple) and ligand (lavender).

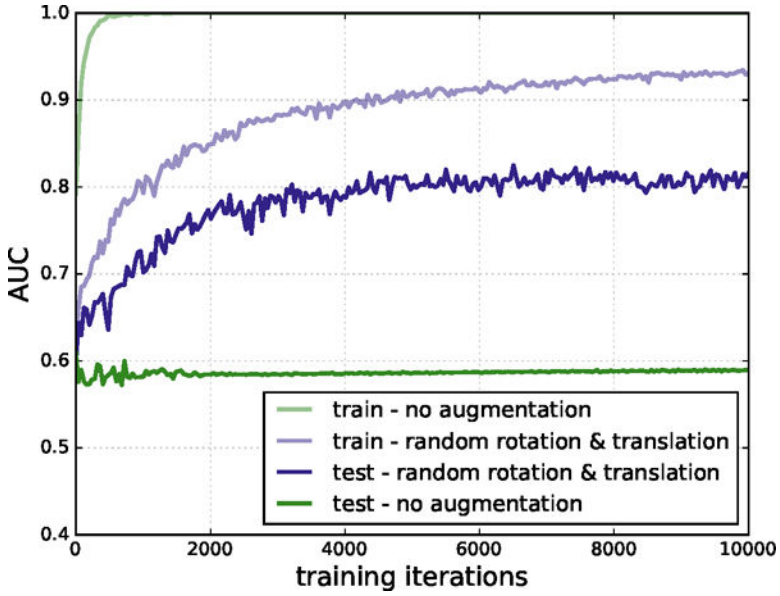


Figure 3.

AUC on training and test sets, with and without data augmentation. Training on CSAR without data augmentation results in classic signs of overfitting: the training set AUC approaches 1.0, but the test AUC plateaus at a much lower value. When additional random rotations and translations are included in the training set, overfitting is reduced.

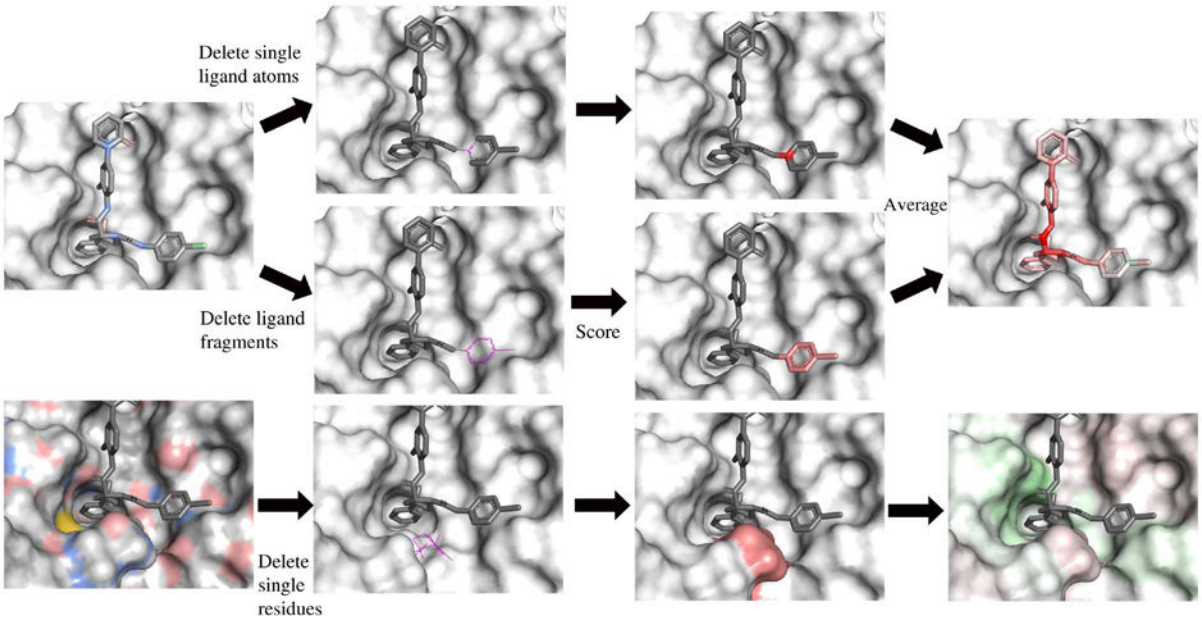


Figure 4.

Visualization algorithm. In the ligand, atoms are removed individually or as fragments and each modified molecule is scored. The assigned color is the difference between the unmodified protein-ligand score and the score with the removed atom. The protein is treated similarly, but whole residues are removed. Positive score differences indicate a positive contribution by the atom to the overall score and are colored green, with the intensity depending on the magnitude of difference. Red represented negative score differences.

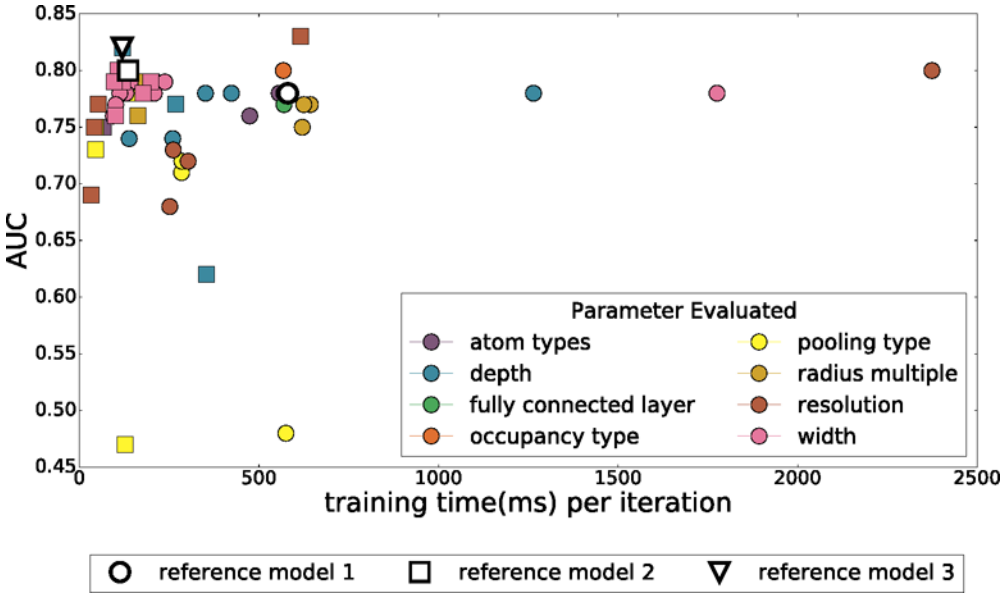


Figure 5.

The training time and average cross-validation AUC of various models created by systematically varying parameters. Marker shape indicates iteration of optimization and the color what parameter was varied.

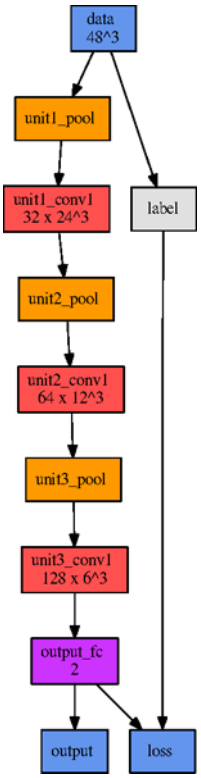


Figure 6.
The network architecture of our final model.

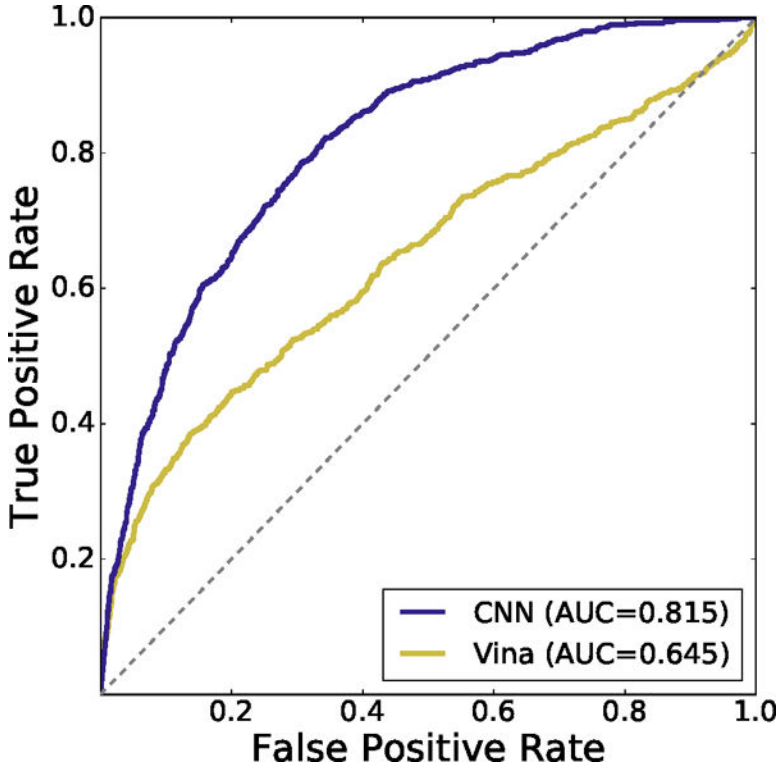


Figure 7.

Inter-target cross-validated ROC curve of CNN scoring method compared to Autodock Vina on the CSAR pose prediction dataset. The CNN performs better at classifying generated poses as low or high RMSD across targets.

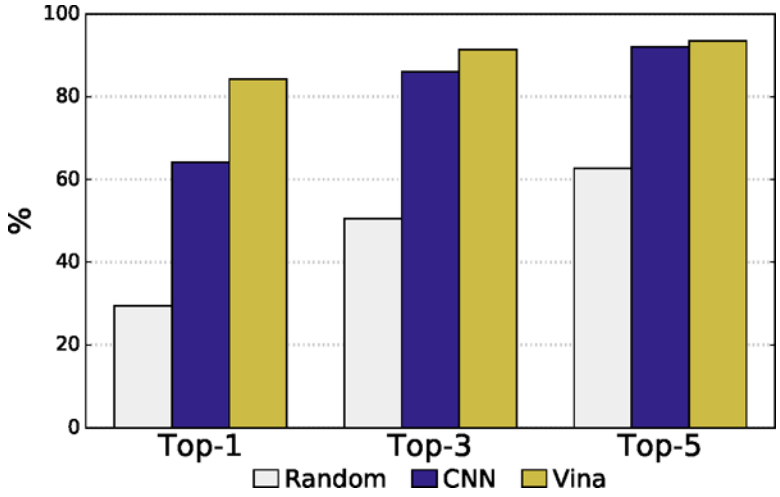


Figure 8.

Intra-target pose ranking. The percent of targets with a low RMSD pose ranked as the top one, three, or five poses is shown. Vina and CNN have similar recovery rates among the top-5 ranked poses, but Vina more often ranks a low RMSD pose as the top-1 ranked pose.

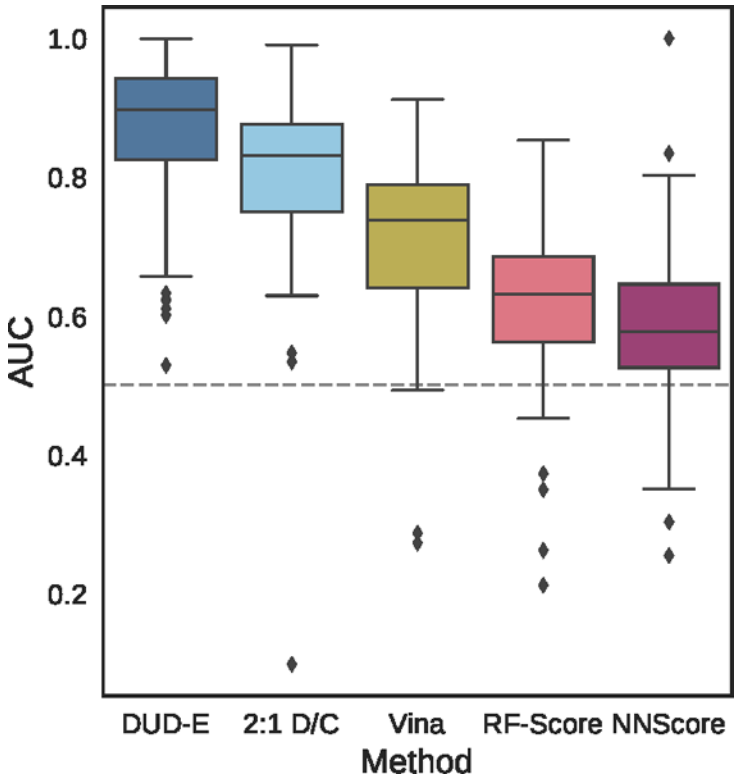


Figure 9.
Distribution of the area under the ROC curve for targets of the DUD-E dataset for the pose-insensitive CNN model trained only on DUD-E, the pose-sensitive DUD-E/CSAR 2:1 model, Vina, RF-Score, and NNScore.

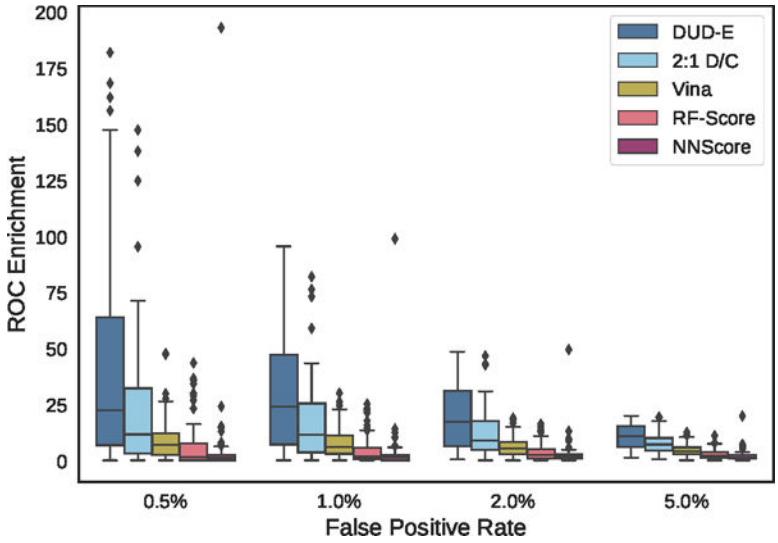


Figure 10.

Distribution of ROC enrichment of at different false positive rates for CNN models compared to Vina, RF-Score, and NNScore scoring functions on the DUD-E dataset.

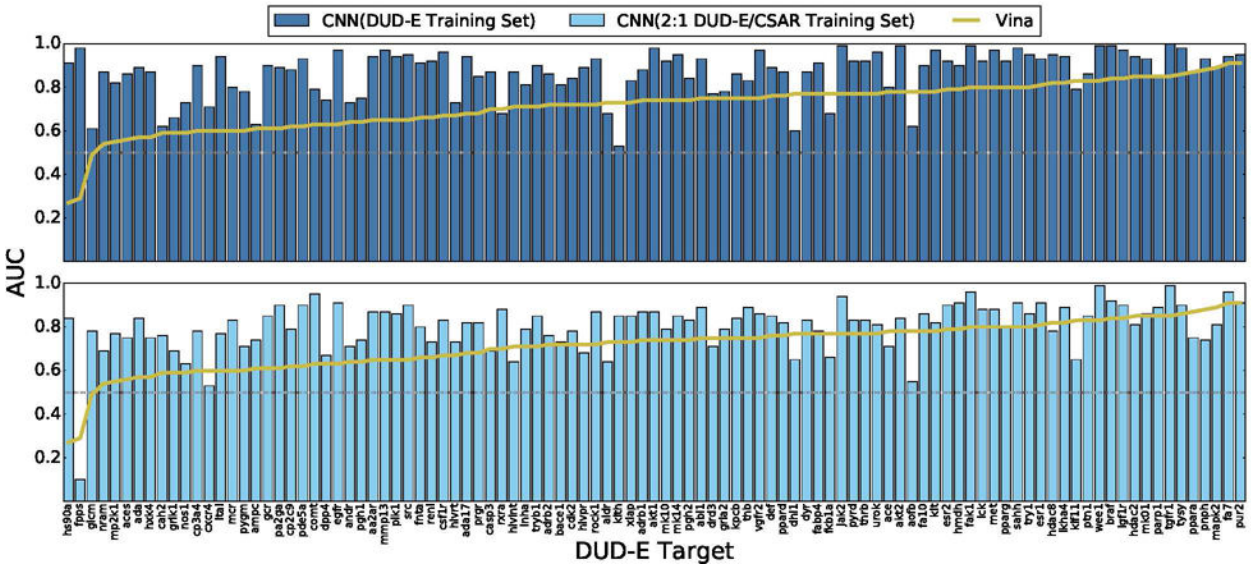


Figure 11.

Cross-validation performance of CNN models on the DUD-E virtual screening benchmark compared to the Vina scoring function. Targets are sorted by performance with Vina. Identical sets of docked poses were ranked. The score of the top ranked pose of each ligand is used to predict activity (multi-pose scoring). CNN models trained only on DUD-E training data perform best, outperforming Vina in 90% of the targets. Models trained using a mix of DUD-E and CSAR data also perform well, achieving better AUCs than Vina in 81% of the targets.

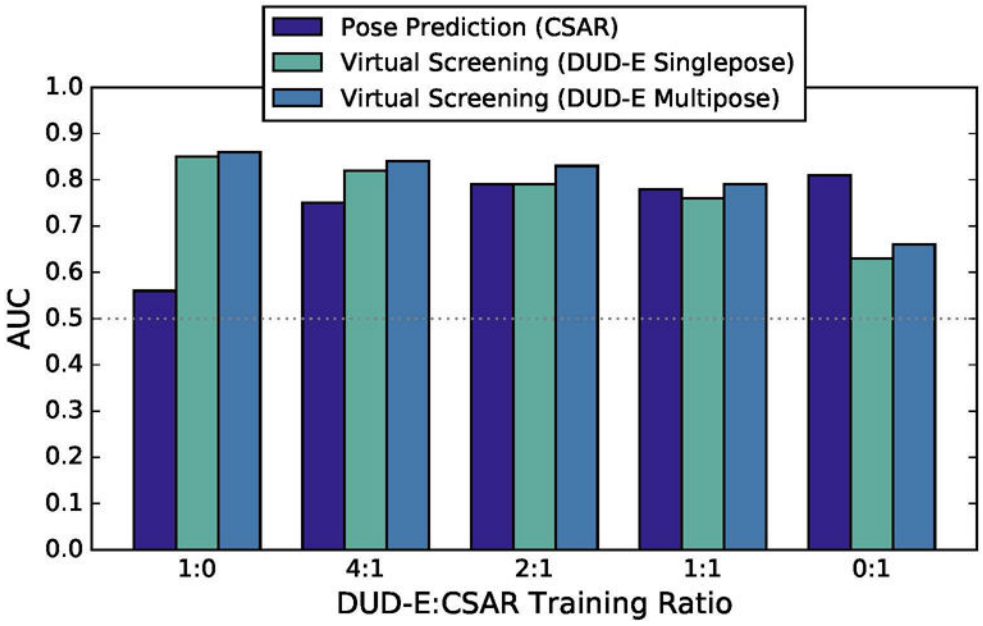


Figure 12.

The cross-validation performance of the CNN model when trained with different ratios of CSAR and DUD-E data and evaluated in terms of pose prediction (CSAR) and virtual screening (DUD-E).

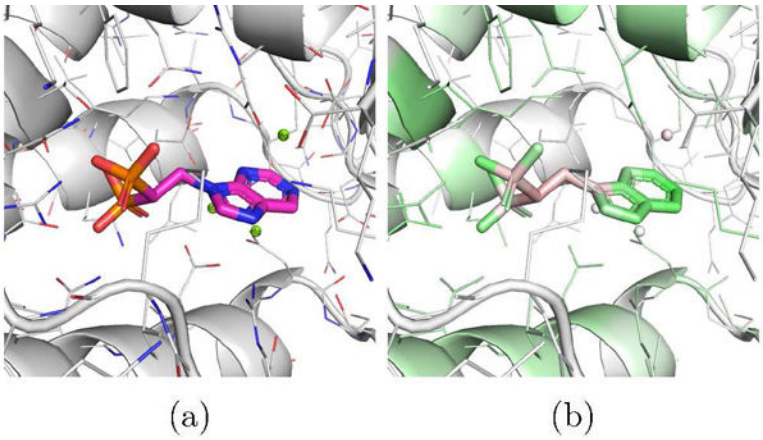


Figure 13.

(a) The top ranked pose by Vina of the CHEMBL457424 ligand of the fpps DUD-E target, (b) Visualization of a CNN model trained using only DUD-E training data. The pose is scored highly due to the polar parts of the structure regardless of the orientation of the ligand.

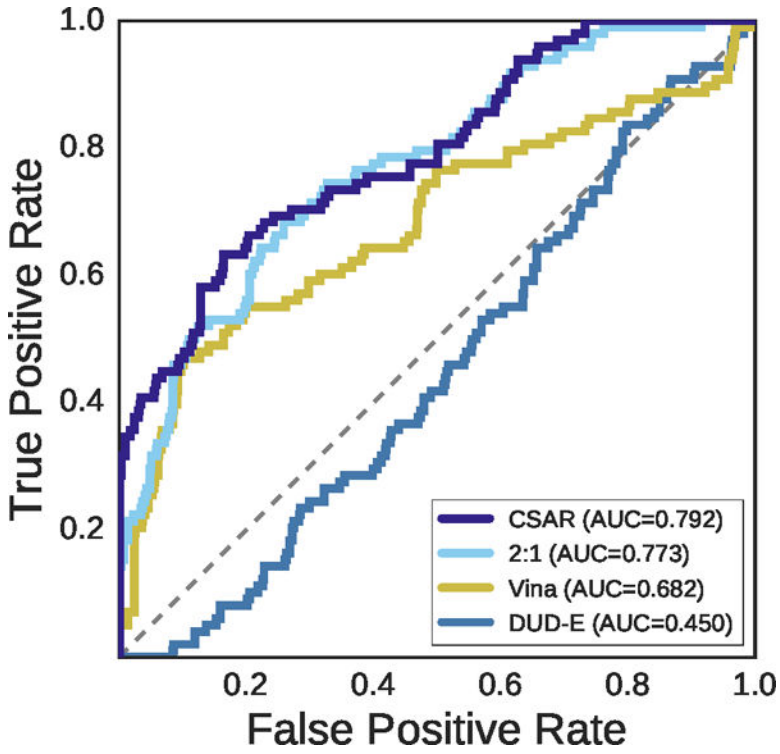


Figure 14.

ROC plot for discriminating low RMSD from high RMSD poses generated from the PDBbind core set. The CSAR-trained CNN performs best at classifying generated poses as low or high RMSD across targets, with a steep initial slope evincing good performance at early recognition.

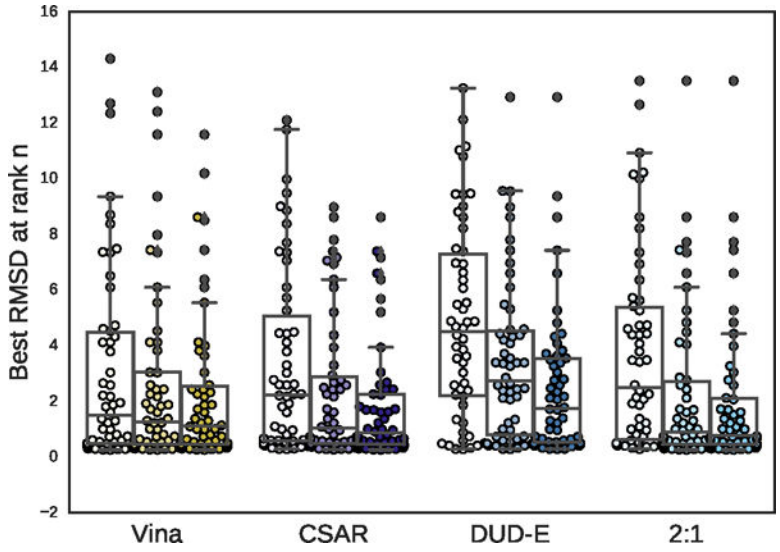


Figure 15.

Boxplots of the best RMSD seen so far at ranks 1, 3, and 5 (shown from left to right) for all targets in the PDBbind core subset.

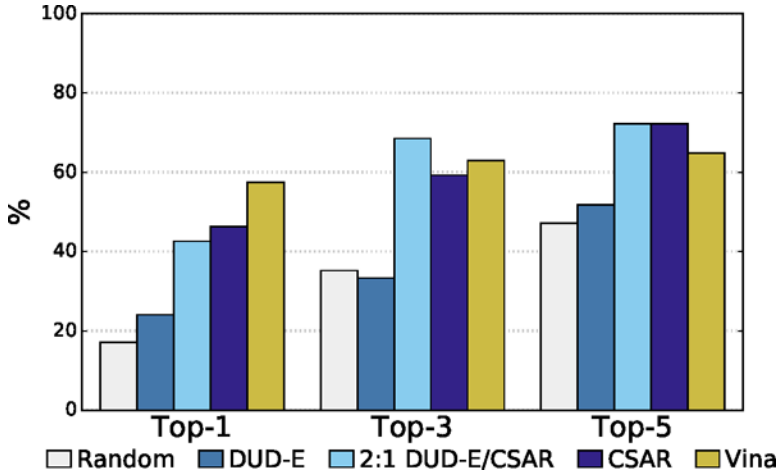


Figure 16.

The percentage of complexes with low RMSD poses identified as the top one, three or five poses for different scoring methods.

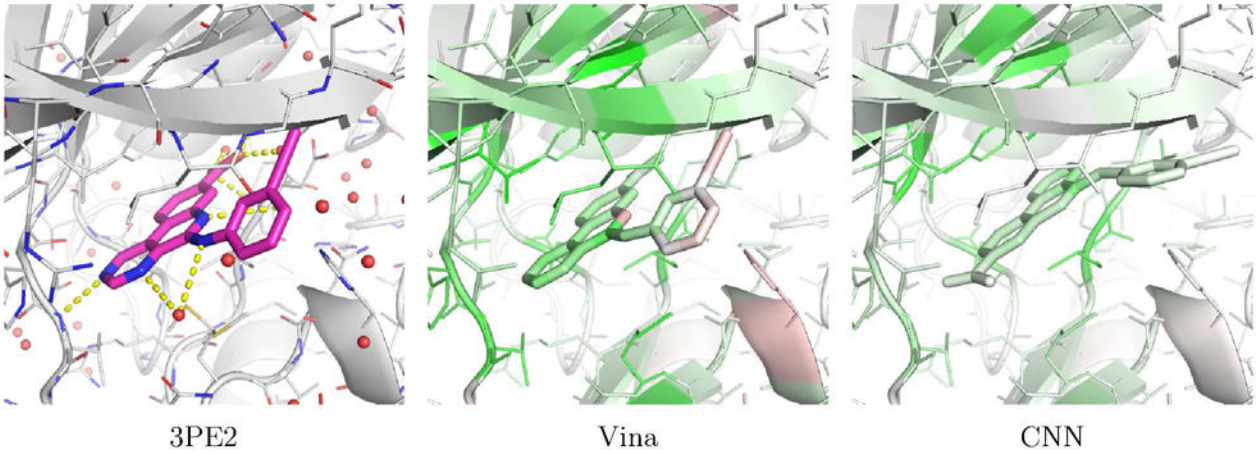


Figure 17.

An example, PDB 3PE2, of a complex from the PI)BI hurl core set where Vina correctly top-ranks a low RMSD pose (0.25A) and the CNN model does not (5.27A). The crystal pose is shown as magenta sticks and the two docked poses are visualized using the CSAR trained CNN model.

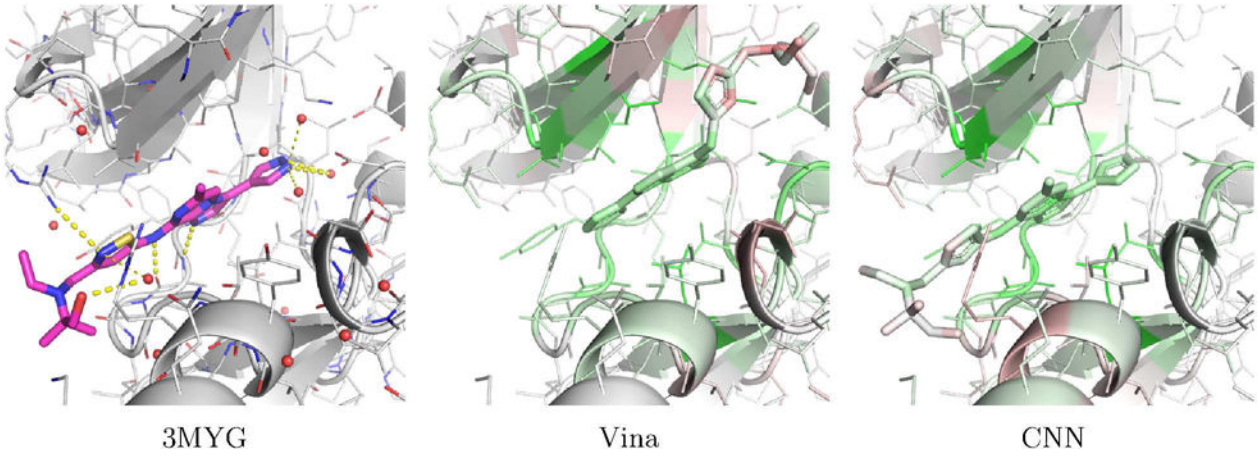


Figure 18.

An example, PDB 3MYG, of a complex from the PDBbind core set where the CNN model correctly top-ranks a low RMSD pose (0.96A) and V ina does not (12.7lA). The crystal pose is shown as magenta sticks and the two docked poses are visualized using the CSAR trained CNN model.

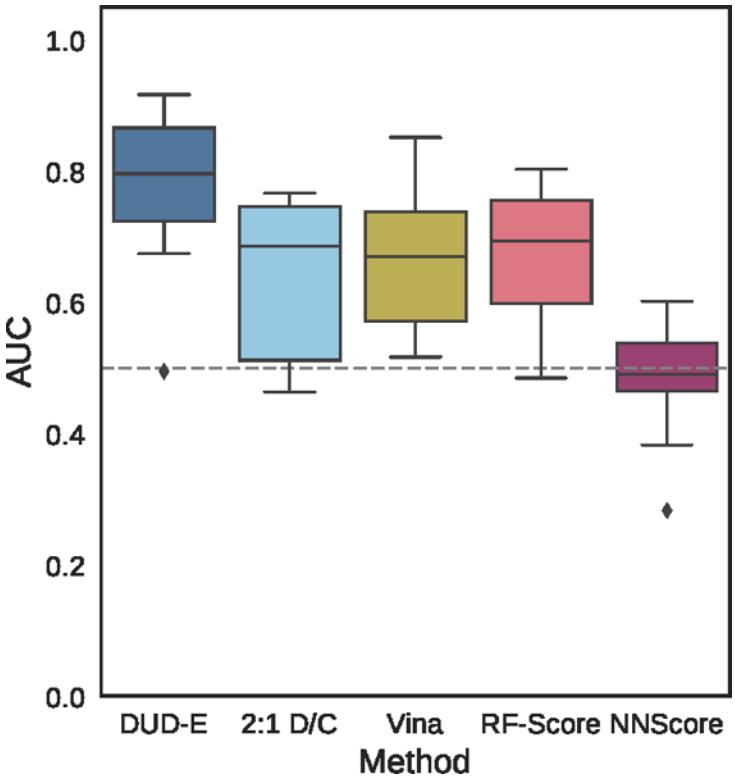


Figure 19.
Distribution of the area under the ROC curve for targets of the ChEMBL dataset for the pose-insensitive CNN model trained only on DUD-E, the pose-sensitive DUD-E/CSAR 2:1 model, Vina, RF-Score, and NNScore.

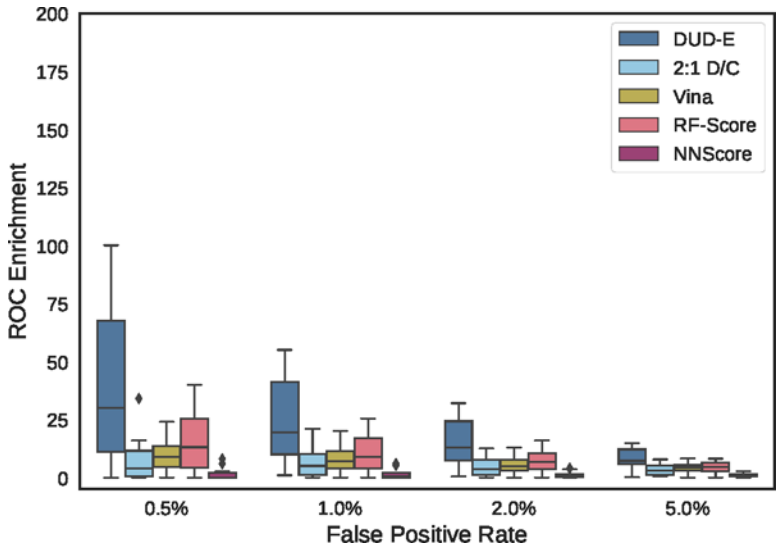


Figure 20.

Distribution of ROC enrichment of ChEMBL targets at different false positive rates for CNN models compared to Vina, RF-Score, and NNScore scoring functions.

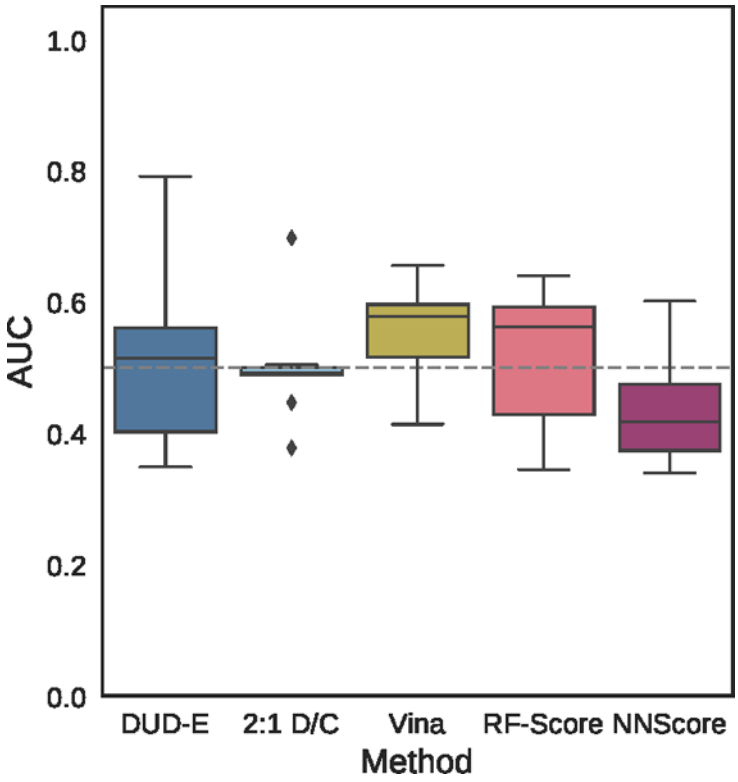


Figure 21.
Distribution of the area under the ROC curve for targets of the MUV dataset for the pose-insensitive CNN model trained only on DUD-E, the pose-sensitive DUD-E/CSAR 2:1 model, Vina, RF-Score, and NNScore.

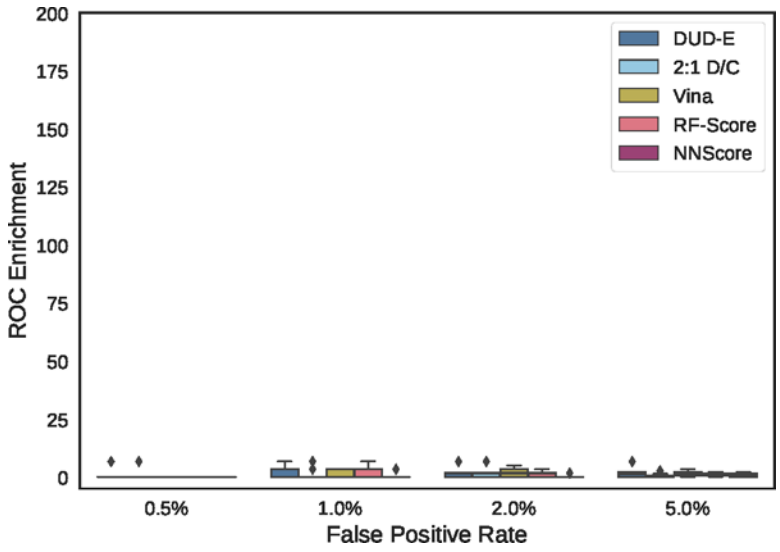


Figure 22.

Distribution of ROC enrichment across MUV targets at different false positive rates for CNN models compared to Vina, RF-Score, and NNScore scoring functions.

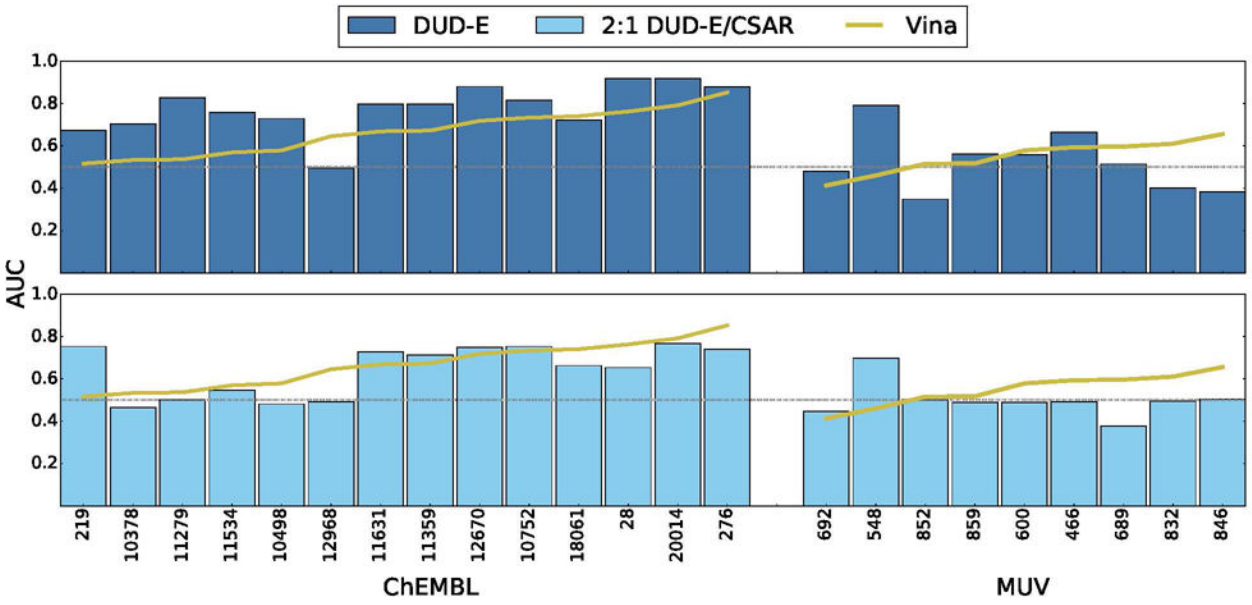


Figure 23.

Performance of CNN models on ChEMBL and MUV screening benchmarks compared to the Vina scoring function. Targets are sorted by performance with Vina. Identical sets of docked poses were ranked. The score of the top ranked pose of each ligand is used to predict activity (multi-pose scoring). Consistent with the cross-validation results (Figure 11), a CNN model trained only on DUD-E training data performs best, outperforming Vina in 86% of the ChEMBL targets and 56% of the MUV targets. Models trained using a mix of DIJD-E and CSAR data performed less well compared to Vina, achieving better AUCs than Vina in 36% of the ChEMBL targets and 22% of the MUV targets.

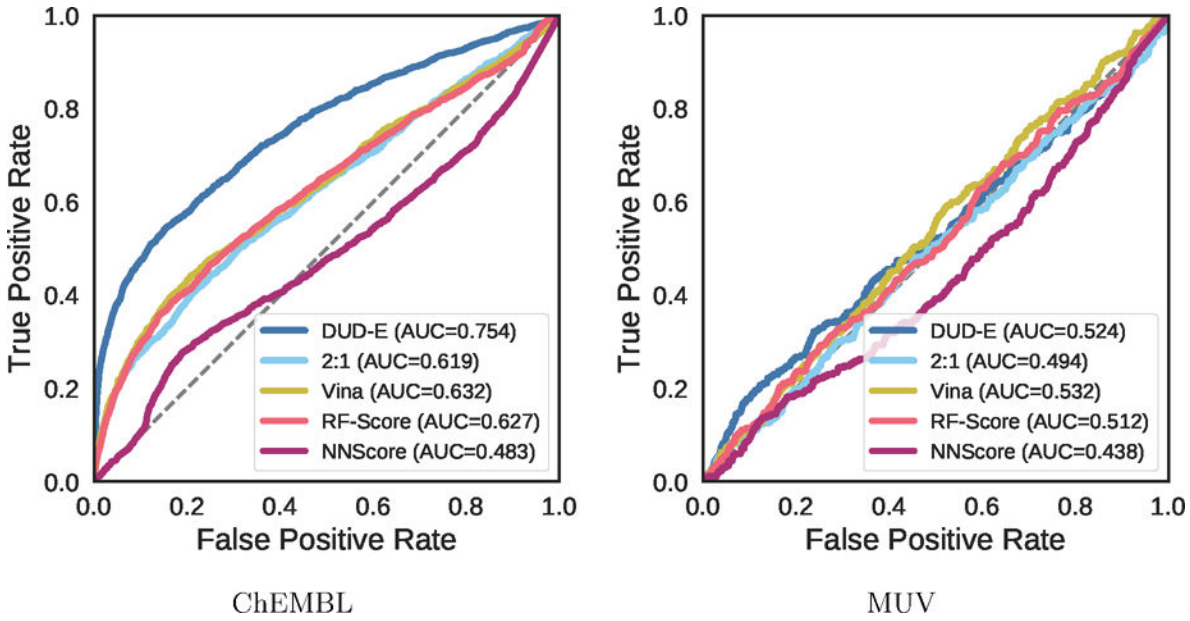


Figure 24.

Overall virtual screening performance represented as a combined ROC curve for two CNN models trained on their full training sets and tested on the ChEMBL and MUV independent test sets and compared to Vina, RF-Score, and NNScore.

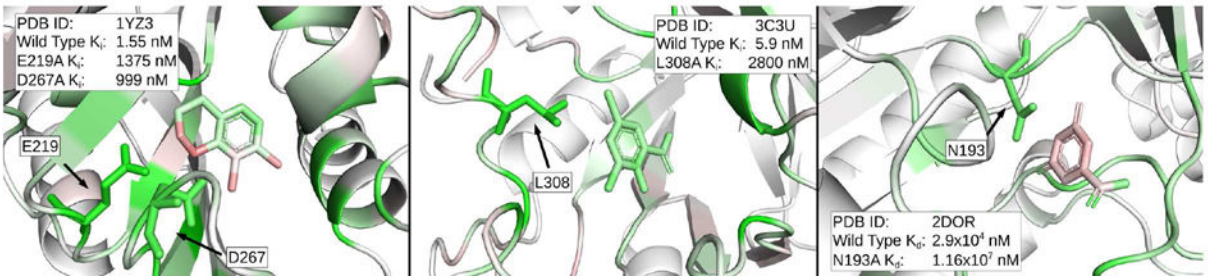


Figure 25.

Visualizations of protein-ligand complexes with binding affinity data for point mutations in the protein. The top three most significant changes in binding affinity from the Platinum database are shown from left to right. Any residue that was mutated experimentally is shown in stick form, while the rest of the protein is shown as a cartoon. In all three cases, the green coloring supports the experimental results that the residues in question are important for ligand binding. Visualization is performed using the 2:1 DUD-E/CSAR model.

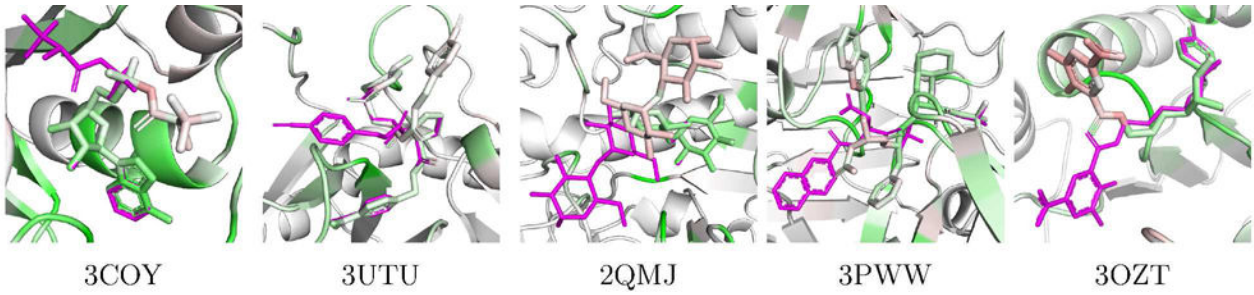


Figure 26.

Visualizations of partially aligned docked poses from the PDBbind core set. The crystal pose is shown as magenta sticks and the docked pose and receptor are colored according to our visualization algorithm and the 2:1 DUD-E/CSAR model. None of these protein targets were included in training. The visualization highlights that the model assesses the part of the pose aligned to the crystal ligand as more favorable than the differing part.

Table 1

Mean AUC and ROC enrichment (RE) across targets in the DUD-E dataset for CNN models, Vina, RF-Score, and NNScore.

Metric	DUD-E	2:1 D/C	Vina	RF-Score	NNScore
AUC	**0.868**	0.804	0.716	0.622	0.584
0.5% RE	**42.559**	22.366	9.139	5.628	4.166
1.0% RE	**29.654**	16.274	7.321	4.274	2.980
2.0% RE	**19.363**	11.888	5.881	3.499	2.460
5.0% RE	**10.710**	7.376	4.444	2.678	1.891

J Chem Inf Model. Author manuscript; available in PMC 2018 April 24.

Table 2

Mean AUC and ROC enrichment (RE) across targets in the ChEMBL dataset for CNN models, Vina, RF-Score, and NNScore.

Metric	DUD-E	2:1 D/C	Vina	RF-Score	NNScore
AUC	**0.779**	0.642	0.665	0.673	0.484
0.5% RE	**40.720**	7.579	9.579	16.005	1.474
1.0% RE	**25.506**	6.291	7.719	10.695	1.733
2.0% RE	**15.575**	4.756	5.503	7.300	1.282
5.0% RE	**8.303**	3.575	4.388	4.380	1.045

Table 3

Mean AUC and ROC enrichment (RE) across targets in the MUV dataset for CNN models, Vina, RF-Score, and NNScore.

Metric	DUD-E	2:1 D/C	Vina	RF-Score	NNScore
AUC	0.522	0.499	**0.549**	0.512	0.441
0.5% RE	**1.481**	0.741	0.000	0.000	0.000
1.0% RE	**1.481**	1.111	1.111	**1.481**	0.370
2.0% RE	1.296	1.111	**1.852**	0.926	0.370
5.0% RE	**1.556**	0.593	1.333	1.053	0.667

Table 4

The virtual screening performance for sphingosine 1-phosphate receptor EDG-1 (PDB 3V2Y) with different choices of active and decoy sets. The active compounds were identified in different screens (biochemical for ChEMBL, cell-based for MUV) and the method used to construct the decoy sets is also different.

Actives	Decoys	Vina	DUD-E	2:1
MUV	MUV	0.593	0.663	0.492
MUV	ChEMBL	0.619	0.682	0.523
ChEMBL	ChEMBL	0.668	0.796	0.727
ChEMBL	MUV	0.667	0.793	0.696