PIGNet: A physics-informed deep learning model toward generalized drug-target interaction predictions†

Seokhyun Moon,‡[1] Wonho Zhung,‡[1] Soojung Yang,‡[1] Jaechang Lim[1] and

Woo Youn Kim[*1,2]

[1]*Department of Chemistry, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*

[2]*KI for Artificial Intelligence, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea (E-mail: wooyoun@kaist.ac.kr)*

‡*These authors contributed equally to this work.*

Recently, deep neural network (DNN)-based drug-target interaction (DTI) models are highlighted for their high accuracy with affordable computational costs. Yet, the models' insufficient generalization remains a challenging problem in the practice of *in-silico* drug discovery. We propose two key strategies to enhance generalization in the DTI model. The first one is to integrate physical models into DNN models. Our model, PIGNet, predicts the atom-atom pairwise interactions via physics-informed equations parameterized with neural networks and provides the total binding affinity of a protein-ligand complex as their sum. We further improved the model generalization by augmenting a wider range of binding poses and ligands to training data. PIGNet achieved a significant improvement in docking success rate, screening enhancement factor, and screening success rate by up to 2.01, 10.78, 14.0 times, respectively, compared to the previous DNN models. The physics-informed model also en-

1

ables the interpretation of predicted binding affinities by visualizing the energy contribution of ligand substructures, providing insights for ligand optimization. Finally, we devised the uncertainty estimator of our model's prediction to qualify the outcomes and reduce the false positive rates.

The accurate prediction of drug-target interaction (DTI) is one of the key steps in the early-stage *in-silico* drug discovery[1]. Experimental validation of all possible compounds would be ideal to find novel drug candidates. However, the brute-force approach is practically inefficient to search across a huge chemical space whose size is estimated from 10^{23} to 10^{602}. On this account, various computational methods with affordable computational costs and reliable accuracy have been proposed for DTI predictions[3–7]. Docking programs are one of the most popular computational tools for the purpose and has been successfully adopted in several early-stage drug discovery projects[8–18]. The docking programs provide the reasonable binding poses and binding affinities of protein-ligand complexes based on physical modeling combined with empirical parameters. Despite its practicability, the accuracy is still far insufficient to find a small fraction of hit compounds among millions of candidates[19–21].

Recently, deep neural networks (DNNs) have been widely studied as a potential surrogate for improving the accuracy of DTI predictions[22–28]. The main advantage of DNN-based DTI models is the models' ability to extract the relevant features directly from raw data as inputs without handcrafted feature descriptions[29]. Among various DNN-based DTI models, the structure-based approach stands out for its accuracy; the spatial coordination of the protein and ligand is crucial in

determining their interactions[30]. Some of the promising approaches utilize a 3-dimensional convolutional neural network (3D CNN)[31–35] or a graph neural network (GNN)[36–38]. Both approaches significantly improved the accuracy of DTI prediction compared to docking calculations.

Despite the advance of previous structure-based DNN models, their generalization ability remains a challenging problem limiting better performance. In particular, the deficiency in 3D structural data of the protein-ligand complexes might drive the models to excessively memorize the features in training data. Such models, being over-fitted to the training data, might fail to generalize in a broader context[39]. Several research reported that DNN-based models often learn the data-intrinsic bias instead of the desirable underlying physics of the protein-ligand interaction[40,41]. For instance, Chen *et al.*[40] reported an extremely high similarity in the performance of the receptor-ligand model and the ligand-only model - both trained with the DUD-E dataset - in terms of area under the ROC curve (AUC). Such a similarity implies that the models might have learned to deduce the protein-ligand binding affinity by only looking at the ligand structures, whether or not the protein structures are included as inputs. Moreover, they showed that the memorization of wrong features can cause a severe degradation in the performance for the proteins that vary from those in the training data. In addition, it was also reported that 3D CNN and GNN models trained on the DUD-E dataset considerably underperformed for different datasets such as the ChEMBL and MUV datasets[35,38]. Such an insufficient generalization of the DTI models can cause an increase in false positive rates in virtual screening scenarios, as the models would often fail to make correct predictions for unseen protein-ligand pairs.

In the field of physical applications of deep learning, the incorporation of right physics as an

inductive bias is a promising mean to improve the model generalization ability. Physical laws are universal, meaning that every nature should follow. Thus, if a model is trained to keep underlying physics, its generalization to unseen data dictated by the same physics can be expected. Several studies have indeed shown that the physics-informed models maintain their generality for unseen data[42,43].

In this regard, we propose two key strategies to enhance the generalization ability of DTI models. First, we introduce a novel physics-informed graph neural network, named PIGNet. It provides the binding affinity of a protein-ligand complex as a sum of atom-atom pairwise interactions, which are the combinations of the four energy components - van der Waals (VDW) interaction, hydrogen bond, metal-ligand interaction, and hydrophobic interaction. Each energy component is computed as an output of a physics model parameterized by deep neural networks, which learns the specific pattern of the interaction. This strategy can increase the generalization ability by allowing the model to dissect an unseen protein-ligand pair as combinations of commonly observed interactions between the protein and the ligand. The detailed pattern of local interactions can render the model to learn the universal physics underlying the protein-ligand binding. Moreover, as the model provides predictions for each atom-atom pair and each energy component, it is possible to analyze the contribution of individual molecular substructures to the binding affinity. This information can be used to modify drug candidates to further strengthen the binding affinity.

Second, we introduce a data augmentation strategy that aims to further improve the model generalization. The experimental data on protein-ligand binding structures is very insufficient to cover the structural diversity of all possible binding complexes. The limitation in the dataset would

make the model fall short in practice. In virtual screening scenarios, typical screening libraries include a wide range of chemical moieties, most of them never seen by the model. Also, the models only trained with optimal binding structures provided by the experimental data may fail to distinguish stable and non-stable binding poses in pose predictions[40]. Therefore, we provide a large quantity of various poses and ligands generated from computations to the dataset. Taking one step closer to the practice of virtual screening, we devise an uncertainty estimator for PIGNet to improve the prediction reliability in this data-deficient situation. We show that one can qualify the outcomes to filter out the most uncertain predictions. The first ever introduction of uncertainty quantification in structure-based DTI models would benefit practitioners by lightening their massive experimental burden.

To evaluate the generalization ability of the proposed model, we use the CASF2016 benchmark. Previously, the evaluation of the DTI models had mostly focused on the correlation between the predicted and the experimental binding affinities[34–38]. However, the high correlation does not automatically guarantee the good model generalization[40]. A well-generalized model should successfully identify the true binding pose in minimum-energy or correctly rank the best binding molecule, where the former criterion is related to docking power, and the latter is related to screening power. Examining both tasks are essential to make sure that the model is well-generalized and thus can be utilized in practice such as virtual high-thoughput screening (vHTS). We compared the performance of PIGNet with other docking programs and previous deep learning models, in terms of various metrics including docking power and screening power of the CASF2016 benchmark[44]. PIGNet significantly improved the docking power and concomitantly the screening power,

compared to the previous deep learning models. In addition to the model performance evaluation, we show the possibility of interpretation. Since explaining the underlying chemistry of the DTI prediction is pharmacologically important, a lack of interpretability in deep learning models was a crucial drawback[45,46]. Physics-based deep learning models can offer interpretability by providing intermediate values that carry certain physical meanings[47]. As our model can predict the interaction energy for each atom-atom pair, we can assess the contribution of each ligand substructure in total binding free energy. Taking account of the information, practitioners would be able to substitute the less contributing moieties, opening up the possibility of further ligand optimization.

Methods

Related works

Summary of previous works: 3D CNN and GNN The 3D CNN takes a 3D rectangular grid that represents the position of atoms at the binding site as an input.[31–35] The proposed 3D CNN models outperformed docking programs for the DUD-E and PDBbind dataset in terms of Pearson's correlation coefficient and AUC. Nevertheless, the high dimensionality of 3D rectangular representations and lack of explicit chemical interactions may put a limitation on 3D CNN models.[48] One of the promising alternatives is a GNN, which takes structural information in the form of molecular graphs.[49] In the molecular graph, each atom and chemical interaction corresponds to the node and the edge of the graph, respectively. Such an efficient representation of chemical interactions contribute to better performance of GNN in DTI predictions compared to docking programs and 3D CNN models.[36–38]

6

Physics-informed neural networks Greydanus *et al.*[42] proposed a Hamiltonian neural network to solve problems that follow Hamiltonian mechanics. They used deep neural networks to predict parameters in the Hamiltonian equation and showed better generalization than regular neural networks. Pun *et al.*[43] proposed a physics-informed neural network potential for atomistic modeling. The model predicts the parameters of the interatomic potentials separately instead of directly predicting the total energy of the system, improving the model generalization for simulating out of bonding region. With neural networks, we parameterize the equations that are derived from the physics regarding chemical interactions.

Overview of our model

PIGNet predicts a binding free energy of any given protein-ligand complex. Taking structural information as an input, our model generates a neural representation for each atom-atom pair interaction through the series of GNN layers. The pairwise representation is fed to neural networks to produce parameters for each of the four physics-informed parameterized equations. The outputs of the equations each corresponds to VDW interaction, hydrogen bond interaction, metal-ligand interaction, and hydrophobic interaction. The binding affinity of each atom-atom pair is computed as a sum of four energy components, and the total binding affinity for a protein-ligand complex is obtained by adding up the binding affinities of all atom-atom pairs. The overall model architecture is described in Figure 1.

Model Architecture Our model takes a molecular graph, G, and the distances between the atom pairs, d_{ij}, of a protein-ligand complex as an input. Generally, a graph, G, can be defined as

(H, A), where H is a set of node features and A is an adjacency matrix. In an attributed graph, the ith node feature, h_i, is represented by a vector. An initial node feature includes atom type, degree of the atom, number of the attached hydrogen atoms, number of implicit valence electrons, and aromaticity in an one-hot vector. Notably, our graph representation includes two adjacency matrices to discriminate the covalent bonds in each molecule and the intermolecular interactions between protein and ligand atoms. The details of the explicit node features and the construction of the two adjacency matrices are explained in the Supplementary Information.
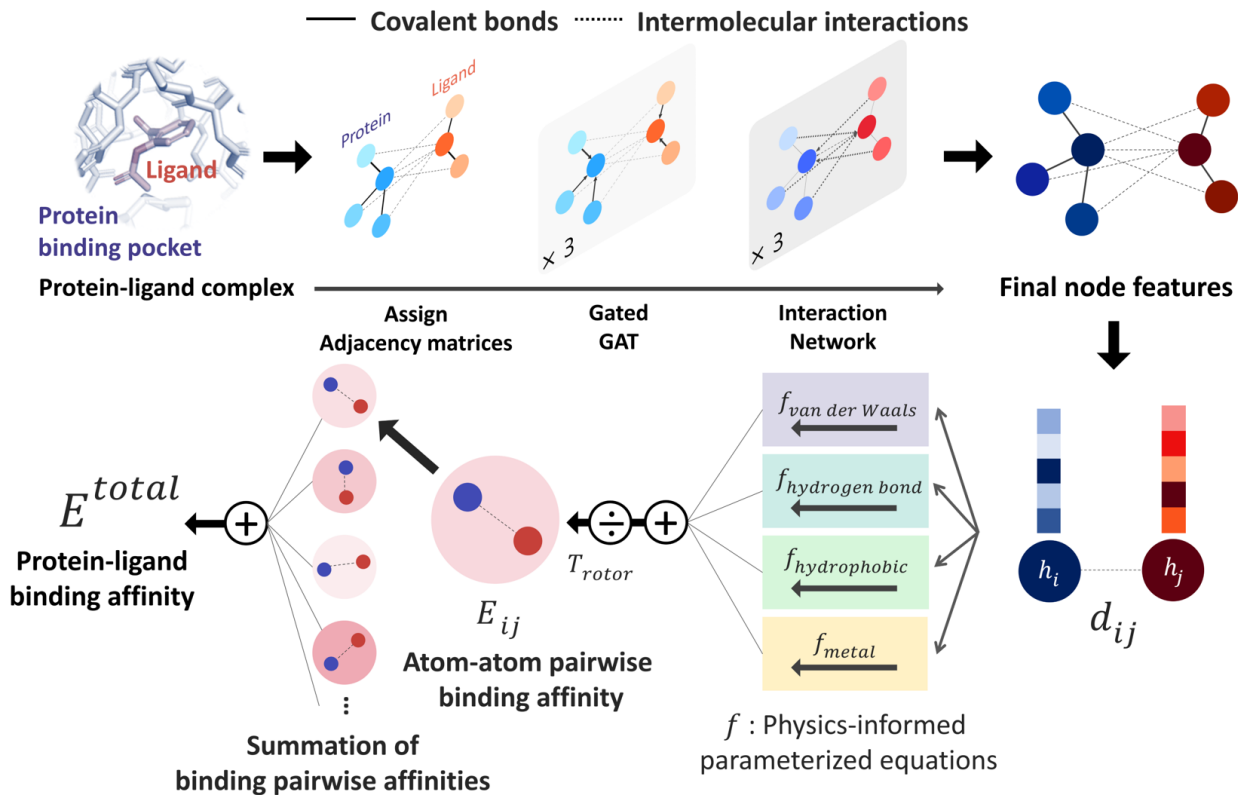

Figure 1: **Our model architecture.** A protein-ligand complex is represented in a graph and adjacency matrices are assigned from the binding structure of the complex. Each node feature is updated through neural networks to carry the information of covalent bonds and intermolecular interactions. Given the distance and final node features of each atom pair, four energy components are calculated from the physics-informed parameterized equations. The total binding affinity is obtained as a sum of pairwise binding affinities, which is a sum of the four energy components divided by an entropy term.

Our model consists of several units of gated graph attention network (gated GAT) and interaction network. The gated GAT and interaction network update each node feature via two adjacency matrices that correspond to covalent bonds and intermolecular interactions. During the node feature update, gated GAT and interaction network learn to convey the information of covalent bonds and intermolecular interactions, respectively. After several node feature updates, we calculate VDW interactions (E^{vdw}), hydrogen bond interactions (E^{hbond}), metal-ligand interactions (E^{metal}), and hydrophobic interactions ($E^{hydrophobic}$), by feeding the final node features into physics-informed parameterized equations. Specifically, for each energy component, the fully connected layers take a set of final node features as input and produce the parametric values of the physics-informed equation. We also consider the entropy loss from the protein-ligand binding by dividing total energy with rotor penalty (T^{rotor}). The total energy can be written as follows:

$$E^{total} = \frac{E^{vdw} + E^{hbond} + E^{metal} + E^{hydrophobic}}{T^{rotor}}. \tag{1}$$

Gated graph attention network (Gated GAT) The gated GAT updates a set of node features with respect to the adjacency matrix for covalent bonds. The attention mechanism aims to put different weights on the neighboring nodes regarding their importance. The attention coefficient, which implies the importance of the node, is calculated from the two nodes that are connected in a covalent bond and then normalized across the neighboring nodes. The purpose of the gate mechanism is to effectively deliver the information from the previous node features to the next node features. The extent of the contribution from the previous nodes is determined by an coefficient, which is obtained from the previous and new node features. We describe the details of gated GAT in the

Supplementary Information.

Interaction network The interaction network takes an updated set of node features from the gated GAT along with the adjacency matrix to generate the next set of node features. Unlike the gated GAT, the interaction network adopts an adjacency matrix featuring intermolecular interactions. The interaction network produces two different sets of embedded node features by multiplying the previous set of node features with two different learnable weights. Next, we apply max pooling to each set of embedded node features, obtaining two set of interaction embedded node features. The interaction embedded node features are then added to the embedded node features to generate the new node features. The final node features are obtained as a linear combination of the new and previous node features, where the linear combination is performed with a gated recurrent unit (GRU)[50]. We describe the details of the interaction network in the Supplementary Information.

Physics-informed parameterized function PIGNet consists of four energy components - VDW interaction, hydrophobic interaction, hydrogen bonding, and metal-ligand interaction - and a rotor penalty. Energy component of an interaction between the ith node and the jth node is computed from two node features, h_i and h_j. Since the node features contain the information of the two atoms and their interaction, the model can reasonably predict DTI.

The energy components and the rotor penalty are motivated from the empirical functions of AutoDock Vina[8]. The total binding affinity is obtained as a weighted sum of energy components, where the weights are introduced to account for the difference between the calculated energies and the true free binding energies.[8] PIGNet employs learnable parameters to find an optimal weight for each

10

component, learning to account for the different types of protein-ligand interactions.

Each energy component is calculated from d_{ij} and d'_{ij}, which are the inter-atomic distance and the corrected sum of the VDW radii of the ith node and the jth node, respectively. d'_{ij} can be represented as follows:

$$d'_{ij} = r_i + r_j + c \cdot b_{ij}, \tag{2}$$

where r_i and r_j are the VDW radii of the ith node and the jth node, respectively, and b_{ij} is a correction term between the two nodes. b_{ij} is resulted from a fully connected layer that accepts two node features h_i and h_j as inputs. We used 0.2 for the constant c to scale the correction term.

van der Waals (VDW) interaction We used 12-6 Lennard-Jones potential to calculate the VDW interaction term, E^{vdw}. We considered all protein and ligand atom pairs except for metal atoms whose VDW radii highly vary depending on the atom type. The total VDW energy is obtained as a sum of all possible atom-atom pairwise VDW energy contribution coefficients. E^{vdw} can be described as follows:

$$E^{vdw} = \sum_{i,j} c_{ij} \left[\left(\frac{d'_{ij}}{d_{ij}} \right)^{12} - 2 \left(\frac{d'_{ij}}{d_{ij}} \right)^6 \right], \tag{3}$$

where c_{ij}, predicted from a fully connected layer, indicates the minimum VDW interaction energy and renders each estimated energy component similar to the true energy component, in order to reflect the physical reality.

Hydrogen bond, Metal-ligand interaction, Hydrophobic interaction The pairwise energy contribution coefficients, e_{ij}, of hydrogen bond (E^{hbond}), metal-ligand interaction (E^{metal}), and hydropho-

bic interaction ($E^{hydrophobic}$) share the same expression as shown in equation (4) with different

coefficients, c_1, c_2, and a learnable scalar variable, w.

$$
e_{ij} = \begin{cases} w & \text{if} \quad d_{ij} - d'_{ij} < c_1 \\ w\left(\frac{d_{ij} - d'_{ij} - c_2}{c_1 - c_2}\right) & \text{if} \quad c_1 < d_{ij} - d'_{ij} < c_2 \\ 0 & \text{if} \quad d_{ij} - d'_{ij} > c_2 \end{cases} \tag{4}
$$

Here, c_1 and c_2 are set as -0.7 and 0.0 for hydrogen bonds and metal-ligand interactions, respec-

tively, while the constants are set as 0.5 and 1.5 for hydrophobic interaction. We chose the same

values of c_1 and c_2 for hydrogen bonds and metal-ligand interactions, since both originate from the

electron donor-acceptor interactions. The total energy component is computed as a summation of

all atom-atom pairwise energy contribution coefficients, as described in equation (5).

$$
E = \sum_{i,j} e_{ij} \tag{5}
$$

We classified atoms into hydrogen bond acceptors, hydrogen bond donors, metal atoms, and hy-

drophobic atoms. Since hydrogen bonds appear between hydrogen bond donors and hydrogen

bond acceptors, each atom that forms hydrogen bonds are selected by substructure matching of the

general SMARTS[51] descriptors, which are summarized in Supplementary Table S2. Metal atoms

include Zn, Mn, Co, Mg, Ni, Fe, Ca, and Cu. Lastly, halogen atoms or carbon centers that are

surrounded only by carbon or hydrogen atoms are classified as hydrophobic atoms[10].

Rotor penalty The rotor penalty term, T^{rotor}, is intended to consider a loss of entropy as the binding

pocket interrupts the free rotation of chemical bonds during protein-ligand binding. We assumed that the entropy loss is proportional to the number of the rotatable bonds of a ligand molecule. T^{rotor} can be described as follows:

$$T^{rotor} = 1 + C_{rotor} \times N_{rotor}, \tag{6}$$

where N_{rotor} is the number of rotatable bonds and C_{rotor} is a positive learnable scalar variable. We used RDKit software[52] to calculate N_{rotor}.

Loss functions The loss function of PIGNet consists of three components, L_{energy}, $L_{derivative}$, and $L_{augmentation}$ as in equation (7). Figure 2 explains the overall training scheme of PIGNet based on the three loss functions.

$$L_{total} = L_{energy} + L_{derivative} + L_{augmentation} \tag{7}$$

L_{energy} is the mean squared error (MSE) loss between the predicted value from the model, y_{pred}, and the corresponding experimental binding free energy, y_{true},

$$L_{energy} = \frac{1}{N_{train}} \sum_i \left(y_{pred,i} - y_{true,i}\right)^2, \tag{8}$$

where N_{train} is a number of training data. Minimizing L_{energy} avails the model to correctly predict
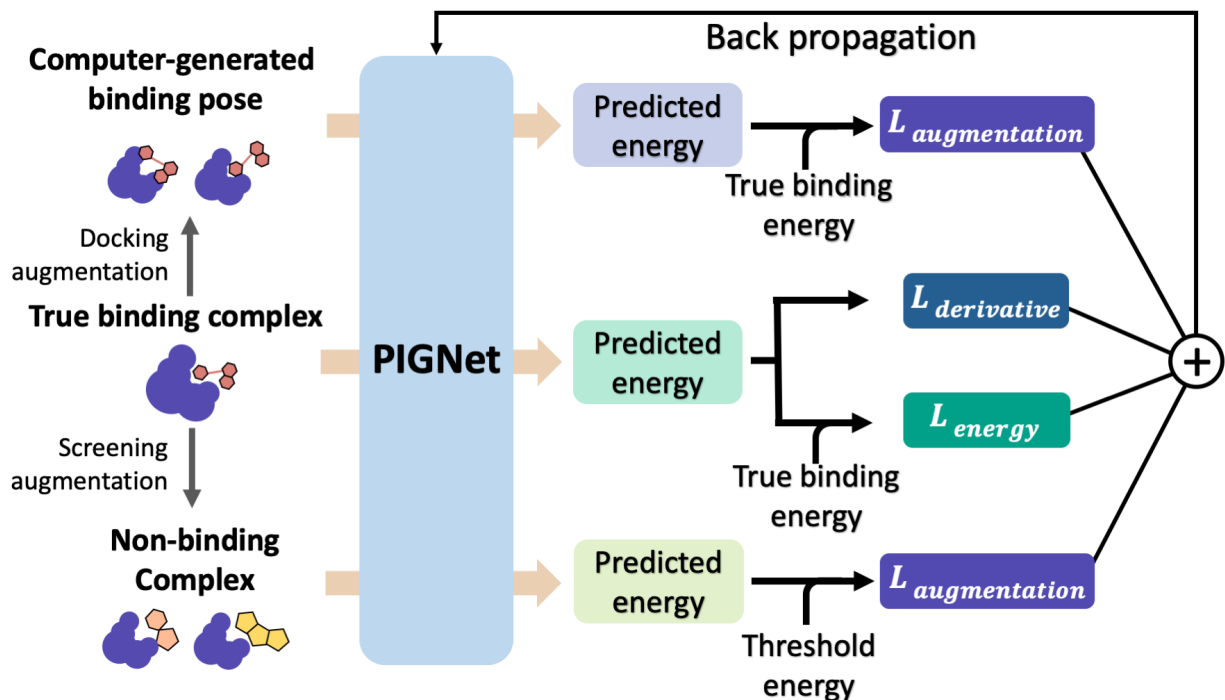
Figure 2: **The training scheme of PIGNet.** We use three types of data in model training - true binding complex, true binder ligand-protein pair in computer-generated binding pose, and non-binding decoy complex. PIGNet predicts binding free energy for each input. For a true binding complex, the model learns to predict its true binding energy. The model also learns to predict the energy of a computer-generated binding pose complex or a non-binding decoy complex in higher value than the true binding energy and threshold energy, respectively. Finally, PIGNet learns the proper correlation of ligand atom position and binding affinity by minimizing the derivative loss.

the binding affinity of experimental 3D structures. $L_{derivative}$ is composed of the first and the second derivative of the energy with respect to the atomic position. Minimizing $L_{derivative}$ intends the model to sensitively find relatively stable poses. $L_{augmentation}$ is the loss related to the data augmentation.

Derivative loss The shape of the potential energy curve between the protein and ligand atoms has a huge impact on distinguishing the stable binding poses. The ligand atoms are located at the local minimum of the potential curve when the ligand binding is stable. Also, a potential energy

curve in proper sharpness makes it easier to distinguish stable conformers from the others, as a small change in atomic positions would induce a large amount of energy deviation. Since a model trained with respect to L_{energy} alone does not control the shape of the potential energy curve, it would be hard to distinguish whether or not a ligand is at a stable position. Accordingly, we guide the model with the derivative loss, $L_{derivative}$, to learn the proper shape of the pairwise potential energy curve, the width and the minimum energy position in particular.

We can assume that the ligand atoms are located at the local minimum of the potential for the experimentally validated binding structures. Thus, we make the experimental structures as a local minimum by forcing the first derivative of the potential energy with respect to position to become zero. The sharpness of the potential energy curve was induced by increasing the second derivative. The derivative loss, $L_{derivative}$, is given as follows:

$$L_{derivative} = \sum_i \left(\frac{\partial E^{total}}{\partial q_i} \right)^2 - min \left(\left(\frac{\partial^2 E^{total}}{\partial q_i^2} \right), C_{der2} \right), \tag{9}$$

where q_i is the position of the ith ligand atom. An excessively sharp potential energy curve may cause a problem in energy prediction by the immense deviation of energy from a small change in ligand atom positions. Therefore, we set its maximum value as C_{der2}, which is 20.0 in our model.

Data augmentation loss Here, we constructed three different data augmentations and the corresponding loss functions; docking augmentation, random screening augmentation, and cross screening augmentation. Smina (scoring and minimization with AutoDock Vina)[53] was used to prepare computer-generated decoy structures.

- Docking augmentation

  The purpose of docking augmentation is to improve the model to distinguish the most stable binding poses from the others. We assume experimental binding structures from the PDB-bind dataset[54] as the most stable binding poses. Thus, the energy of experimental structures should be lower than the predicted energy of decoy structures that have different poses from true binding poses. The loss for docking augmentation, $L_{docking}$, can be written as follows:

  $$L_{docking} = \sum_i \max\left(y_{exp,i} - y_{decoy,i},\ 0\right),\qquad(10)$$

  where y_{exp} is the energy of an experimental structure and y_{decoy} is the predicted energy of a decoy structure. By minimizing $L_{docking}$, the model can predict y_{decoy} as larger than y_{exp}.

- Random screening augmentation

  In general, only a small fraction of molecules in a huge chemical space can bind to a specific target protein. Most molecules would have low binding affinity and high dissociation constant, k_d, with the target. From this nature, we assume that the dissociation constant of an arbitrary protein-ligand pair from the virtual screening library would be higher than $10^{-5}M$, as a criterion for hit identification is conventionally in micromolar ($10^{-6}M$) scale[55]. Referring to the relationship between the binding free energy ΔG and the binding constant, k_a, which is reciprocal to K_d, we can set a threshold for ΔG of a protein-ligand pair as follows:

  $$\Delta G \geq -1.36 \log K_a = -6.8 \text{ kcal/mol}.\qquad(11)$$

A model trained with random screening loss, $L_{random\_screening}$, and a non-binding random molecule-protein pair can sufficiently learn the chemical diversity. The model would predict the binding free energy of a random molecule with the target to a value higher than the threshold energy, -6.8. Thus, the loss for the random screening augmentation, $L_{random\_screening}$, can be written as follows:

$$L_{random\_screening} = \sum_i \max\left(-y_{random,i} - 6.8, 0\right), \tag{12}$$

where y_{random} is the prediction energy of synthetic compounds from the IBS molecule library.[56] The inaccuracy of a docking program is not problematic for the augmentation, as the binding energies of wrong binding poses are typically higher than the true binding energy.

- Cross screening augmentation

  Another nature of protein-ligand binding is that if a ligand strongly binds to a specific target, the ligand is less likely to bind to other targets, because the different types of proteins have different binding pockets. We assumed that the true binders of the PDBbind dataset do not bind to the other proteins in the PDBbind dataset.

  As in the random screening augmentation, training with non-binding ligands and protein pairs affect a model to learn chemical diversity. The loss for the cross screening augmenta-

17

tion, $L_{cross\_screening}$, can be written as follows:

$$L_{cross\_screening} = \sum_i \max\left(-y_{cross,i} - 6.8, 0\right), \tag{13}$$

where y_{cross} is the prediction energy of the cross binder. The same threshold for the binding

free energy as in random screening augmentation is also used here.

Total loss function The total loss, L_{total}, is the weighted sum of all the loss terms: L_{energy},

$L_{derivative}$, $L_{docking}$, $L_{random\_screening}$, and $L_{cross\_screening}$. The total loss can be written as follows:

$$
\begin{aligned}
L_{total} = & L_{energy} \\
& + c_{derivative} L_{derivative} \\
& + c_{docking} L_{docking} \\
& + c_{random\_screening} L_{random\_screening} \\
& + c_{cross\_screening} L_{cross\_screening},
\end{aligned}
\tag{14}
$$

where $c_{derivative}$, $c_{docking}$, $c_{random\_screening}$, and $c_{cross\_screening}$ are hyper parameters. We set

$c_{derivative}$, $c_{docking}$, $c_{random\_screening}$, and $c_{cross\_screening}$ as 10.0, 10.0, 5.0, and 5.0, respectively.

Dataset Our primary training set is the PDBbind v.2019 refined set which provides experimental

protein-ligand x-ray crystal binding structures and the corresponding binding affinities. We used

4,212 samples for the training set and the 259 samples for the test set. For the docking augmen-

tation, we generated 202,035 decoy structures using the PDBbind v.2016 dataset. For the random

18

screening augmentation and the cross screening augmentation, we generated 773,623 complexes

using the IBS molecules and 386,876 complexes based on the random cross binding, respectively.

Any complexes in the augmentation do not include proteins and ligands from the test set. See our

github for more information about data preprocessing codes.

Benchmark method *Benchmark datasets* Our primary test set is the CASF2016 benchmark dataset[44],

which provides four metrics to evaluate a DTI prediction model: scoring power, ranking power,

docking power, and screening power.

The scoring power is defined as a linear correlation of predicted binding affinity and experimental

binding data, measured in terms of the Pearson's correlation coefficient, R.

The ranking power is the ability of a model to correctly rank the binding affinities of true binders

for a certain target protein, given the binders' precise binding poses. A metric for the ranking

power is the Spearman's rank correlation coefficient, ρ.

The docking power is the ability of a model to find the native ligand binding pose among decoys

with computer-generated poses. We report the overall success rate of the docking benchmark.

The screening power is the ability of a model to identify the true binding ligands for a given

target protein among a set of random molecules. We measure the screening power in terms of

enhancement factor (EF) and success rate.

The details of CASF2016 benchmark metrics are further explained in the Supplementary Infor-

mation. From the four metrics, the CASF2016 benchmark dataset enables a detailed evaluation

19

of the DTI prediction model. Along with the CASF2016, we computed the Pearson's correlation coefficient R for CSAR NRC-HiQ (2010) 1 and 2[57] to evaluate the scoring power of the model with respect to external datasets.

Benchmark criteria In vHTS schemes, a model should be able to identify the true binding poses among others and correctly ranks the complexes by their binding affinities. The former relates to the docking power and the latter relates to the ranking and screening power. Indeed, the docking, ranking, and screening powers depend on the model's ability to predict the correct binding affinity of a given complex, which is typically estimated in the scoring power. However, an experimental analysis on CASF2016 benchmark shows that the screening and docking powers are not necessarily correlated with the scoring power[44]. We attribute this inconsistency to a limitation in the CASF2016 scoring power benchmark - the test dataset only contains experimentally obtained protein-ligand complex structures and binding affinities, where the docking and screening power benchmarks include a much wider range of complexes. In this regard, the docking and screening powers can be an indicator of model over-fitting, and the scoring power itself cannot be a single criterion of a DTI model performance evaluation. Accordingly, in contrast to previous studies that mostly focus on the scoring power alone, we highlight the models' screening and docking powers in the discussion.

Compared models We compare our model with some of the most popular docking programs for DTI predictions: X-Score[10], AutoDock Vina[8], ChemPLP@GOLD[15], and GlideScore[13]. We constructed two DNN models based on 3D CNN and GNN architectures to compare with PIGNet. The 3D CNN and GNN based models are the two major approaches for the structure-based DTI

prediction. The 3D CNN models are characterized by the voxelized 3D grid representation of a protein-ligand complex structure and the 3D convolutional networks, while the GNN models take graph representation - often including atom-atom pairwise distances - and utilize graph convolutional networks. For the 3D CNN-based model, we reimplemented the K_{DEEP} model from Jiménez *et al.*[33] Our rebuilt 3D CNN-based model is identical to K_{DEEP}, except for the atom feature construction. We replaced atom features with those of PIGNet. We also constructed a GNN-based model identical to PIGNet except for the physics modeling part; the model produces final outputs via FC layers instead of the parametric equations used in PIGNet.

Results and Discussions

Assessment of the model performance and generalization ability Table 1 summarizes the performance of our model and the benchmark models for the CASF2016 and the CSAR benchmark. In the CASF2016 benchmark, our model outperformed all the others including deep learning models and conventional scoring functions such as X-Score, AutoDock Vina, ChemPLP@GOLD, and GlideScore. In terms of scoring R, ranking ρ, docking success rate, screening EF and success rate, our model's performance was respectively, 1.67, 1.09, 2.01, 10.78, and 14.03 times of the 3D CNN-based and GNN-based models', at most. Our model also shows competitive performance on the CSAR NRC-HiQ 1 and 2 benchmarks.

As expected, the 3D CNN-based and GNN-based models show high accuracy in predicting binding affinities for the CASF2016 scoring power, ranking power, CSAR 1, and 2 - the benchmark sets consisted of experimentally obtained samples. However, those models fail to achieve high

docking power and screening power. We attribute such low docking power to model over-fitting on the true-binding complex structures and binding affinities. The models have produced inaccurate binding affinities for the computer-generated decoy structures, which are primarily queried for the docking and screening power test. The low docking power then leads to the low screening power, as the most stable binding conformer needs to be identified in order to find the true binder. From these observations, we suspect that the performance reports of the previously introduced deep DTI models have been overoptimistic. In contrast, PIGNet consistently shows high performance across the four CASF2016 metrics and the external CSAR benchmarks. Such results imply that our model is properly fitted to the training data, and also has learned the proper features - the underlying physics of protein-ligand binding patterns. Moreover, the results remind us that the scoring power cannot be a single criterion measuring the model performance. In the following section, we analyze how much each of our strategy had contributed to the result through ablation studies.

Ablation study on the generalization ability of the model We attribute the improvement of our model in generalization to both two major strategies we have utilized; the physics-informed inductive bias on the model, and the data augmentation strategy designed for a DTI model. In this section, we carried out an ablation study to decouple the effects of the two strategies. Table 2 summarizes the results. We did not include a physics-informed version of the 3D CNN-based model, as it is not straightforward to impose the parameterized equations on the 3D CNN architecture. Specifically, 3D CNN models do not produce explicit atom-atom pairwise representations, which are required for the parameterized equations.

Effect of the physics-informed inductive bias on the models We can observe the effect of the physics-informed model by comparing the performances of GNN-based models and PIGNet, as a GNN-based model is identical to PIGNet except the parametric equations. As expected, the effect was not critical for the CASF2016 scoring and ranking powers. However, the employment of the physics-informed model has resulted in a significant increase in docking and screening powers. We conclude that the incorporation of the parametric equations has clearly contributed to enhanced model generalization. The incorporation of the equations may impose an excessive inductive bias on the model, increasing the possibility of under-fitting. However, the comparable scoring powers of PIGNet and GNN-based models render the possibility unlikely. In addition, we believe advances in physical modeling and finer categorization of energy components will further improve PIGNet without under-fitting. Interestingly, PIGNet without data augmentation shows better docking power than GNN with data augmentation. While adding several hundred thousand training data clearly improves the models, the augmentation alone cannot entirely replace the generalization effect given by the physics-informed model. Instead, the data augmentation and physical modeling improve the model in a complementary manner, as we can see from the improvement of PIGNet through the data augmentation.

Effect of the DTI-adapted data augmentation strategy on the models The PDBbind dataset is one of the most representative training datasets for deep DTI models, providing both 3D binding structures and the binding affinities of the protein-ligand complexes[54]. However, the PDBbind dataset is suspected to hold an intrinsic bias[40]; its ligands have insufficient chemical diversity and only the binding structures in minimum energy poses are given. To overcome the current limitation of the

PDBbind dataset, we additionally included total 1,362,534 computer-generated random protein-ligand complexes in training. Taking ligands from the IBS molecular library[56] and generating decoys enrich the chemical diversity of ligands and 3D structural diversity of binding poses. Table 2 reveals the effect of the DTI-adapted data augmentation strategy on the model generalization ability. The augmentation improved the docking and screening power regardless of the model architectures. It shows the applicability of our data augmentation strategy for a variety of DNN-based DTI models. For benchmarks only containing the true binding complexes - scoring power, ranking power, the CSAR 1 and 2 - it was an expected result that the data augmentation did not improve the scores, because the model learns to accurately distinguish the decoy and true binding complexes from the augmented data and the corresponding losses.

Interpretation of the physically modeled outputs One important advantage of our approach is the possibility of the atom-atom pairwise interpretation of DTI. To rationally design a drug for a specific target, knowing the dominant interaction of ligand binding is helpful. Since PIGNet computes atom-atom pairwise energy components, we can calculate the energy contribution of the substructures within a ligand. Here, we conduct a case study for two target proteins retrieved from the PDBbind dataset; protein-tyrosine phosphatase non-receptor type 1 (PTPN1) and platelet activating factor acetylhydrolase (PAF-AH). The result is illustrated in Figure 3a, where two ligands for each protein are compared regarding the predicted substructural energy contributions and the inhibitory constant, K_i. Note that the lower the K_i, the stronger the protein-ligand binding affinity. For PTPN1, the two ligands differ only by the red-circled substructure. The sulfonyl piperidine moiety is inserted in the second ligand instead of the benzyl moiety in the first ligand. The model

predicts greater energy contribution for the sulfonyl piperidine moiety, and such a result is coherent to the experimental K_i values. For PAF-AH, the ligand with the phenyl group has a lower K_i value than that of the ligand with the methyl group. The model predicts greater energy contribution of the phenyl group compared to the methyl group. For both proteins, the common substructure of the two ligands, highlighted in the figure with blue circles, are predicted to have similar energy contributions. This implies that the predicted outcomes provide physically meaningful interpretations of the ligand binding, which can be further used for ligand optimization to strengthen the binding affinity with the target protein.

Most docking programs manually assign different scoring functions to atom-atom pairs according to the predefined categories. This manual assignment would fall short when the binding pattern of the pair does not fit in the existing category. Instead of the handcrafted categorization, our model exploits neural networks to automatically differentiate the atom-atom pairs; the information of each pair's interaction is updated through the graph attention networks. We illustrate the deviation and its physical interpretation in Figure 3b and 3c.

Figure 3b shows a distance-energy distribution plot of VDW component for carbon-carbon pairs within the test set. When trained with learnable parameters, predicted VDW interactions naturally deviate within the carbon-carbon pair, while without the learnable parameters the distance-energy plot follows a single solid line. With the aid of learnable parameters, our model might have learned a wider range of pairwise interactions in a data-driven manner. We also show the deviations in hydrophobic, hydrogen bond, and metal energy components in the Supplementary Figure S1.

Figure 3c shows that the naturally occurring deviations within the atom-atom pairs in our model
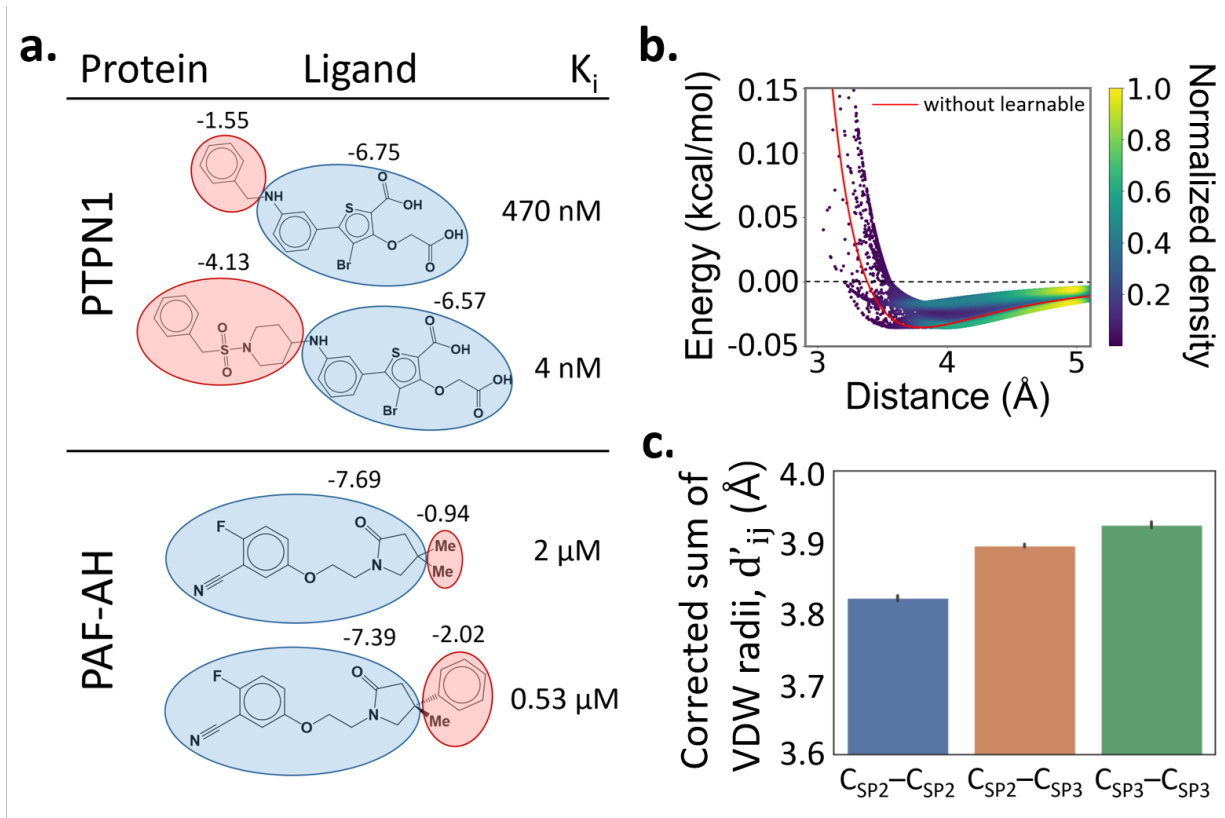
Figure 3: **Interpretation of the predicted outcomes. a.** Substructural analysis of ligands for two target proteins. Protein-tyrosine phosphatase non-receptor type 1 (PTPN1) and platelet activating factor acetylhydrolase (PAF-AH). The blue and red circles indicate common and different substructures, respectively, and the predicted energy contribution (unit: kcal/mol) of each substructure is annotated. The inhibitory constant, K_i, indicates how potent the ligand binds to the target protein. **b.** A distance-energy plot of carbon-carbon pairwise van der Waals (VDW) energy components in the test set. The red solid line illustrates the original distance-energy relation without any deviation induced by learnable parameters. The closer the color of a data point to yellow, the larger the number of corresponding carbon-carbon pairs. **c.** The average value of the corrected sum of VDW radii, d'_{ij}, corresponding to different carbon-carbon pair types. $C_{sp^2} - C_{sp^2}$, $C_{sp^2} - C_{sp^3}$, and $C_{sp^3} - C_{sp^3}$ pairs are compared.

are the consequences of learning sufficient physics information. The corrected sum of VDW radii,

d'_{ij}, which contains a learnable parameter assigned to each atom-atom pair, deviated according to

the carbon-carbon pair types. Since the interaction between the two carbon atoms would not be

significantly affected by their hybridization, we speculate that the corrected sum of VDW radii of

26

the pair would be dependent on the atom radii. The result shows an increasing tendency from the $C_{sp^2} - C_{sp^2}$ pair to the $C_{sp^3} - C_{sp^3}$ pair. Resonating with the speculation, larger the s-character of the carbon atoms, shorter was the corrected sum of VDW radii.

Uncertainty quantification of PIGNet For the reliable virtual screening, it is important to screen out the false positive binders and secure the true positives[58]. Unfortunately, even though a large number of false positives cost an immense experimental burden, most positive returns of docking programs are actually false positives[59]. The DNN models can also experience the same problem. The data-deficient nature of training DTI models might render the DNN models less fit to out-of-domain complexes[40], producing the false positive results. One promising way of reducing false positives is quantifying the uncertainty of the predictions and sorting out the unreliable positive predictions. However, the scope of the related works is limited to the ligand-based models[60–63]. In this regard, we hereby demonstrate the first uncertainty quantification on a structure-based DTI model.

We employed the MC-dropout, a practical Bayesian inference method utilizing dropout regularization, to estimate epistemic uncertainties[64]. For aleatoric uncertainty quantification, since the PIGNet architecture is not compliant to a direct implementation of the previous methods, we devised a new strategy. Kendall *et al.*[65] shows that a model can be trained to predict both the output and its aleatoric variance, given the convergence of the negative log likelihood loss. Since our model adds up the predictions from each atom-atom pair to obtain an output for a protein-ligand complex, we aggregate the pairwise aleatoric uncertainties by multiplication over all pairs, as atom-atom pairwise interactions are not mutually independent. When the aggregation was proceeded by

summation or taking an average, the model failed to converge. The further details of the imple-

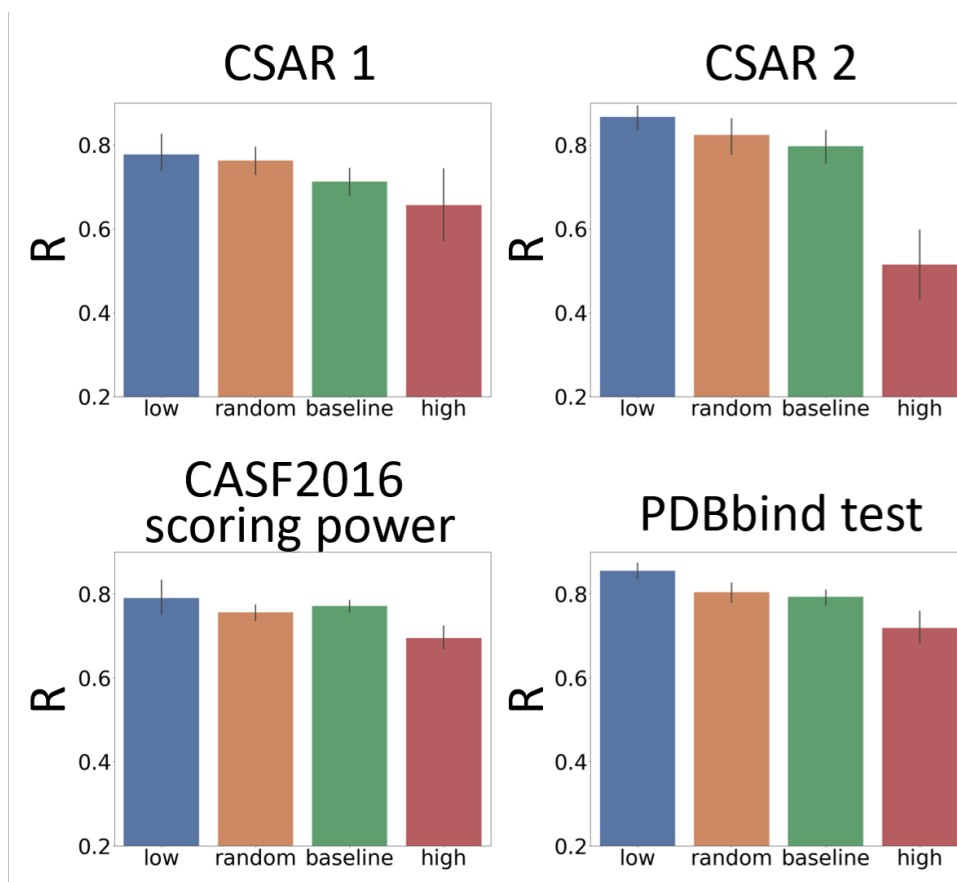mentation are explained in the Supplementary Information.

Figure 4: **Plot of the average Pearson's correlation coefficients, R, of the 5-fold PIGNet model, with or without the uncertainty estimator, on the datasets classified according to the total uncertainty.** PIGNet with the uncertainty estimator - **low**: the lowest third, **random**: the randomly selected one third, **high**: the highest third of the uncertainty distribution. PIGNet without the uncertainty estimator - **baseline**: the randomly selected one third. An error bar represents one standard deviation. PIGNet with and without the uncertainty estimator were tested at the 500th and $1,000^{th}$ training epoch, respectively. For each model, the random selection was performed five times and the R value was obtained by a 5-fold average.

We quantified uncertainties for the samples in four datasets - CSAR NRC-HiQ 1 and 2, the

CASF2016 scoring power, the PDBbind test set. In Figure 4, the 'low', 'random', and 'high'

batches are in descending order in terms of Pearson's correlation, R, in all four datasets. Such a

result resonates with the expectation; the lower the uncertainty, the more probable the model would have correctly predicted the result. The consistent results across the four test datasets show that the uncertainties can be properly measured for our model. Furthermore, a previous study shows an evidence of the correlation between good generalization ability and robust uncertainty quantification[61]. Thus, it might be possible to relate the high generalization ability of our model to the success in the uncertainty quantification.

For PIGNet without the uncertainty estimator, the one third of the predictions was randomly selected, namely the 'baseline' subset. By comparing the R values of the 'random' and 'baseline' batches, we can evaluate the general performance of PIGNet with and without the uncertainty estimator. The result shown in Figure 4 confirms that the addition of uncertainty estimator does not harm the model performance.

The results show that PIGNet is eligible to filter out the false positives by uncertainty quantification. By sorting out the predictions in high uncertainty, we could obtain a high correlation between the prediction and the true affinity value - which means we have excluded the complexes that are highly likely to be the false positives.

Conclusion

In this work, we integrated physical models into neural networks to build a generalized drug-target interaction (DTI) prediction model, named PIGNet. By achieving a high model generalization, our model has substantial pragmatic merits in virtual high-throughput screening schemes. Such an achievement is substantiated with the CASF2016 and CSAR benchmark results. A DTI model for

29

virtual screening should make accurate energy predictions on true binding complexes and correctly screen out invalid, non-binding complexes. The former requirement is associated with scoring and ranking powers, and the latter condition is related to docking and screening powers. Yet the DNN-based benchmark models scored similarly competent scoring and ranking powers, no model was marked comparable to PIGNet regarding the docking and screening powers.

We attribute the success in our model to the two key strategies proposed in this work. The first one is to employ the physics-informed parameterized equations. The physics modeling acts as a proper inductive bias for the neural model, guiding the model to learn the underlying physics of the chemical interactions. We further improved the model performance by augmenting training data with protein-ligand complexes from the wider chemical and structural diversity. We analyzed the effects of the physics-informed model and the data augmentation through the ablation study and found that both contribute to the model generalization.

Our model can enjoy further practical advantages such as the physical interpretation of predicted DTI values and the reduction in false positives via uncertainty quantification. Obtaining binding free energy for every atom-atom pair opens up a possibility of further interpretation. This useful information can later be used to optimize drug candidates to attain better binding affinity. Also, we introduced an uncertainty estimator for DTI prediction models and evaluated the quality of estimation for PIGNet. As predictions in high uncertainty can possibly be false positives, the uncertainty quantification has practical benefits in virtual screening scenarios.

Still, our model has a room for improvement regarding the representation of solvation energy. In reality, proteins and ligands interact while surrounded by numerous water molecules, and the water

molecules engender thermodynamic and structural effects on ligand binding[66]. As our model does not explicitly include a solvent energy component, we think introducing a solvent energy component to PIGNet will further improve the model. Nevertheless, as our results suggest, we believe that incorporating the physical models in a deep DTI prediction model can be a new practical way of improving the quality of DTI predictions.

Data availability

Datasets that we have used for training are shown at github: `https://github.com/jaechanglim/DTI_PDBbind.git`.

Code availability

Our source codes for data preprocessing and training are available at github: `https://github.com/jaechanglim/DTI_PDBbind.git`.

References

1. Hopkins, A. L. Drug discovery: Predicting promiscuity. *Nature* **462,** 167–168 (2009).

2. Polishchuk, P. G., Madzhidov, T. I. & Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design* **27,** 675–679 (2013).

3. Bender, A. & Glen, R. C. Molecular similarity: A key technique in molecular informatics. *Organic and Biomolecular Chemistry* **2,** 3204–3218 (2004).

4. Koes, D. R. & Camacho, C. J. Pharmer: Efficient and exact pharmacophore search. *Journal of Chemical Information and Modeling* **51,** 1307–1314 (2011).

5. Zoete, V., Daina, A., Bovigny, C. & Michielin, O. SwissSimilarity: A Web Tool for Low to Ultra High Throughput Ligand-Based Virtual Screening. *Journal of Chemical Information and Modeling* **56,** 1399–1404 (2016).

6. Lo, Y. C., Rensi, S. E., Torng, W. & Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today* **23,** 1538–1546 (2018).

7. Moumbock, A. F., Li, J., Mishra, P., Gao, M. & Günther, S. Current computational methods for predicting protein interactions of natural products. *Computational and Structural Biotechnology Journal* **17,** 1367–1376 (2019).

8. Allouche, A.-r. Gabedit–A Graphical User Interface for Computational Chemistry Softwares. *Journal of computational chemistry* **31,** 174–182 (2012).

9. Ruiz-Carmona, S. *et al.* rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Computational Biology* **10,** 1–7 (2014).

10. Wang, R., Lai, L. & Wang, S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design* **16,** 11–26 (2002).

11. Jain, A. N. Surflex: Fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of Medicinal Chemistry* **46,** 499–511 (2003).

12. Jones, G. Development and Validation of a Genetic Algorithm for Flexible Docking. *J. Mol. Biol.* **267,** 727–748 (1997).

13. Friesner, R. A. *et al.* Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of medicinal chemistry* **47,** 1739–1749 (2004).

14. Venkatachalam, C. M., Jiang, X., Oldfield, T. & Waldman, M. LigandFit: A novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular Graphics and Modelling* **21,** 289–307 (2003).

15. Korb, O., Stützle, T. & Exner, T. E. Empirical scoring functions for advanced Protein-Ligand docking with PLANTS. *Journal of Chemical Information and Modeling* **49,** 84–96 (2009).

16. Allen, W. J. *et al.* DOCK 6: Impact of new features and current docking performance. *Journal of Computational Chemistry* **36,** 1132–1156 (2015).

17. Zhao, H. & Caflisch, A. Discovery of ZAP70 inhibitors by high-throughput docking into a conformation of its kinase domain generated by molecular dynamics. *Bioorganic and Medicinal Chemistry Letters* **23,** 5721–5726 (2013).

18. Yang, Q. *et al.* In vitro and in silico evaluation of stereoselective effect of ginsenoside isomers on platelet P2Y12 receptor. *Phytomedicine* **64,** 152899 (2019).

19. Waszkowycz, B., Clark, D. E. & Gancia, E. Outstanding challenges in protein-ligand docking and structure-based virtual screening. *WIREs Computational Molecular Science* **1,** 229–259 (2011).

20. Leach, A. R., Shoichet, B. K. & Peishoff, C. E. Prediction of protein-ligand interactions. Docking and scoring: Successes and gaps. *Journal of Medicinal Chemistry* **49,** 5851–5855 (2006).

21. Chen, Y. C. Beware of docking! *Trends in Pharmacological Sciences* **36,** 78–95 (2015).

22. Fitriawan, A., Wasito, I., Syafiandini, A. F., Amien, M. & Yanuar, A. *Multi-label classification using deep belief networks for virtual screening of multi-target drug* in *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)* (2016), 102–107.

23. "Oztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: Deep drug-target binding affinity prediction. *Bioinformatics* **34,** 821–829 (2018).

24. Thafar, M., Raies, A. B., Albaradei, S., Essack, M. & Bajic, V. B. Comparison Study of Computational Prediction Tools for Drug-Target Binding Affinities. *Frontiers in Chemistry* **7,** 782 (2019).

25. Lipinski, C. F., Maltarollo, V. G., Oliveira, P. R., da Silva, A. B. F. & Honorio, K. M. Advances and Perspectives in Applying Deep Learning for Drug Design and Discovery. *Frontiers in Robotics and AI* **6,** 108 (2019).

26. Tsubaki, M., Tomii, K. & Sese, J. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* **35,** 309–318 (2019).

27. Lee, I., Keum, J. & Nam, H. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Computational Biology* **15,** 1–21 (2019).

28. Zheng, S., Li, Y., Chen, S., Xu, J. & Yang, Y. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence* **2,** 134–140 (2020).

29. Gawehn, E., Hiss, J. A. & Schneider, G. Deep Learning in Drug Discovery. *Molecular Informatics* **35,** 3–14 (2016).

30. Panday, S. K. & Ghosh, I. in *Structural Bioinformatics: Applications in Preclinical Drug Discovery Process* 109–175 (Springer, 2019).

31. Imrie, F., Bradley, A. R., Van Der Schaar, M. & Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *Journal of Chemical Information and Modeling* **58,** 2319–2330 (2018).

32. Stepniewska-Dziubinska, M. M., Zielenkiewicz, P. & Siedlecki, P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics* **34,** 3666–3674 (2018).

33. Jiménez, J., Škalič, M., Martínez-Rosell, G. & De Fabritiis, G. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **58,** 287–296 (2018).

34. Wallach, I., Dzamba, M. & Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. *Preprint at arXiv:1510.02855* (2015).

35. Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J. & Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **57,** 942–957 (2017).

36. Feinberg, E. N. *et al.* PotentialNet for Molecular Property Prediction. *ACS Central Science* **4,** 1520–1530 (2018).

37. Torng, W. & Altman, R. B. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *Journal of Chemical Information and Modeling* **59,** 4131–4149 (2019).

38. Lim, J. *et al.* Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *Journal of Chemical Information and Modeling* **59** (2019).

39. Hawkins, D. M. The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences* **44,** 1–12 (2004).

40. Chen, L. *et al.* Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLoS one* **14,** e0220113 (2019).

41. Morrone, J. A., Weber, J. K., Huynh, T., Luo, H. & Cornell, W. D. Combining Docking Pose Rank and Structure with Deep Learning Improves Protein–Ligand Binding Mode Prediction over a Baseline Docking Approach. *Journal of Chemical Information and Modeling* (2020).

42. Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian Neural Networks. *Preprint at arXiv:1906.05163* (2019).

43. Pun, G. P., Batra, R., Ramprasad, R. & Mishin, Y. Physically informed artificial neural networks for atomistic modeling of materials. *Nature Communications* **10,** 2339 (2019).

44. Su, M. *et al.* Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling* **59,** 895–913 (2019).

45. Pahikkala, T. *et al.* Toward more realistic drug-target interaction predictions. *Briefings in bioinformatics* **16,** 325–337 (2015).

46. Gao, K. Y. *et al. Interpretable Drug Target Prediction Using Deep Neural Representation.* in *IJCAI* **2018** (2018), 3371–3377.

47. Zubatyuk, T. *et al.* Machine Learned Hückel Theory: Interfacing Physics and Deep Neural Networks. *Preprint at arXiv:1909.12963* (2019).

48. Karlov, D. S., Sosnin, S., Fedorov, M. V. & Popov, P. GraphDelta: MPNN Scoring Function for the Affinity Prediction of Protein-Ligand Complexes. *ACS Omega* **5,** 5150–5159 (2020).

49. Zhou, J. *et al.* Graph Neural Networks: A Review of Methods and Applications. *Preprint at arXiv:1812.08434* (2018).

50. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Preprint at arXiv:1412.3555* (2014).

51. Wunderlich, R. E., Wenisch, T. F., Falsafi, B. & Hoe, J. C. SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling. *Conference Proceedings - Annual International Symposium on Computer Architecture, ISCA,* 84–95 (2003).

52. Landrum, G. *RDKit: Open-source cheminformatics* <http://www.rdkit.org>.

53. Koes, D. R., Baumgartner, M. P. & Camacho, C. J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *Journal of Chemical Information and Modeling* **53,** 1893–1904 (2013).

54. Liu, Z. *et al.* Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Accounts of Chemical Research* **50,** 302–309 (2017).

55. Principles of early drug discovery. *British Journal of Pharmacology* **162,** 1239–1249 (2011).

56. *InterBioScreen Ltd* <http://www.ibscreen.com>.

57. Jr., J. B. D. *et al.* CSAR benchmark exercise of 2010: Selection of the protein-ligand complexes. *Journal of Chemical Information and Modeling* **51,** 2036–2046 (2011).

58. Maia, E. H. B., Assis, L. C., de Oliveira, T. A., da Silva, A. M. & Taranto, A. G. Structure-Based Virtual Screening: From Classical to Artificial Intelligence. *Frontiers in Chemistry* **8** (2020).

59. Sink, R., Gobec, S., Pecar, S. & Zega, A. False Positives in the Early Stages of Drug Discovery. *Current Medicinal Chemistry* **17,** 4231–4255 (2010).

60. Ryu, S., Kwon, Y. & Kim, W. Y. A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chemical Science* **10,** 8438–8446 (2019).

61. Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P. & Green, W. H. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *Journal of Chemical Information and Modeling* (2020).

62. Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R. & Coley, C. W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *Preprint at arXiv:2005.10036* (2020).

63. Yang, S., Lee, K. H. & Ryu, S. A comprehensive study on the prediction reliability of graph neural networks for virtual screening. *Preprint at arXiv:2003.07611* (2020).

64. Gal, Y. & Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33 rd International Conference on Machine Learning* **48** (2016).

65. Kendall, A. & Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Preprint at arXiv:1703.04977* (2017).

66. Geschwindner, S. & Ulander, J. The current impact of water thermodynamics for small-molecule drug discovery. *Expert Opinion on Drug Discovery* **14,** 1221–1225 (2019).

67. Wang, C. & Zhang, Y. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *Journal of Computational Chemistry* **38,** 169–177 (2017).

Competing Interests   There are no competing interests to declare.

Author contributions   Conceptualization: J.L. and W.Y.K.; Methodology: S.M., W.Z., S.Y., and J.L.; Software, Investigation and Formal Analysis: S.M., W.Z., S.Y., and J.L.; Writing – Original Draft: S.M., W.Z., S.Y., and J.L.; Writing – Review & Editing: S.M., W.Z., S.Y., J.L., and W.Y.K.; Supervision: W.Y.K.

	CASF2016 Benchmark					CSAR	
	Scoring	Ranking	Docking	Screening		NRC-HiQ set1	NRC-HiQ set2
	R	ρ	Success Rate	Average EF	Success Rate	R	R
X-Score[10]	0.631	0.604	63.5%	2.7%	7.0%	0.6	0.65
AutoDock Vina[8]	0.604	0.528	84.6%	7.7%	29.8%	-	-
GlideScore-SP[13]	0.513	0.419	84.6%	11.4%	36.8%	-	-
GlideScore-XP[13]	0.467	0.257	81.8%	8.8%	26.3%	-	-
ChemPLP@GOLD[15]	0.614	0.633	83.2%	11.9%	35.1%	-	-
KDEEP[33]	-	-	-	-	-	0.72	0.65
3D CNN based model	0.652	0.611	42.5%	1.4%	3.5%	0.692	**0.787**
GNN based model	0.723	0.583	67.7%	7.0%	26.3%	0.635	0.786
PIGNet	**0.761**	**0.64**	**85.6%**	**15.1%**	**49.1%**	**0.736**	0.763

Table 1: Benchmark test results on CASF2016 and CSAR NRC-HiQ. R, ρ indicate Pearson Correlation Coefficient, Spearman's rank Correlation Coefficient, respectively. Top 1% rate used for Average EF and Success Rate. ΔVinaRF20[67] was excluded from the comparison for scoring power, as it was fine-tuned on the PDBbind v.2017 data, which in fact includes 50% of data in the CASF2016 test set. We collected the reference scores of CASF2016 from Su *et al.*[44] We have removed the overlapping PDBbind data from the CSAR NRC-HiQ datasets. The reference scores of CSAR NRC-HiQ were curated from Jiménez *et al.*[33], which also has removed the overlapping data.

	CASF2016 Benchmark					CSAR	
	Scoring	Ranking	Docking	Screening		NRC-HiQ set1	NRC-HiQ set2
	R	ρ	Success Rate	Average EF	Success Rate	R	R
3D CNN based model W/O data augmentation	**0.695**	0.589	20.4%	0.7%	1.8%	**0.786**	**0.785**
3D CNN based model with data augmentation	0.652	**0.611**	**42.5%**	**1.4%**	**3.5%**	0.692	0.787
GNN based model W/O data augmentation	**0.773**	**0.617**	28.1%	1.4%	5.3%	**0.792**	**0.787**
GNN based model with data augmentation	0.723	0.583	**67.7%**	**7.0%**	**26.3%**	0.635	0.786
PIGNet W/O data augmentation	0.703	0.606	77.9%	6.0%	26.3%	0.72	**0.789**
PIGNet with data augmentation	**0.761**	**0.64**	**85.6%**	**15.1%**	**49.1%**	**0.736**	0.763

Table 2: Benchmark test results on the CASF2016 and CSAR NiQ dataset for different models with or without data augmentation.

PIGNet: A physics-informed deep learning model toward generalized drug-target interaction predictions

Seokhyun Moon,[†][1] Wonho Zhung,[†][1] Soojung Yang,[†][1]
Jaechang Lim,[1] and Woo Youn Kim[*][1,2]

[1]*Department of Chemistry, KAIST, Daejeon 34141, Republic of Korea*
[2]*KI for Artificial Intelligence, KAIST, Daejeon 34141, Republic of Korea*
[*] Email: wooyoun@kaist.ac.kr
[†] These authors contributed equally to this work.

Contents

1. Model construction

    (a) Neural model architecture
    (b) Physics-informed parameterized functions

2. Benchmark methods

3. Interpretation of the physically modeled outputs

4. Uncertainty quantification method

List of Figures

List of Tables

1. Model construction

(a) Neural model architecture

Atom features

Initial atom features in our model are summarized in Supplementary Table 1. The X in the atom feature means all atoms except C, N, O, S, F, P, Cl, and Br. The final dimension of our node in the graph representation of molecule is 27.

Feature	list
Atom type	C, N, O, S, F, P, Cl, Br, X (onehot)
Degree of atom	0, 1, 2, 3, 4, 5 (onehot)
Number of hydrogen atoms attached	0, 1, 2, 3, 4 (onehot)
Number of implicit valence electrons	0, 1, 2, 3, 4, 5 (onehot)
Aromatic	0 or 1

Supplementary Table 1: The list of atom features

Adjacency matrices

Our graph representation, $G(H, A)$, contains two adjacency matrices expressed as equations (1) and (2). A^1 and A^2 are constructed to account for the covalent bonds and intermolecular interactions in a protein-ligand complex, respectively.

$$A_{ij}^1 = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected by covalent bonds or } i{=}j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$A_{ij}^2 = \begin{cases} 1 & \text{if } 0.5 \text{ Å} < d_{ij} < 5.0 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For A^2, we neglect the atom-atom pair interactions whose pairwise distance is smaller than 0.5 or larger than 5.0. By placing the upper threshold, we limit the effect of distant atoms and reduce the complexity of the graph representation. By setting the lower threshold, we avoid exceptional atom pairs within extremely short distances.

Gated graph attention network (Gated GAT)

The nth unit of the gated GAT generates a set of the next node features from the set of the current node features, $H^n = \{h_1^n, h_2^n, \cdots, h_N^n\}$, and the adjacency matrix A^1, where $h_i^n \in R^F$. The scalar values N and F are the number of the atoms in a protein-ligand complex and the dimension of the node feature, respectively. The initial step of the gated GAT is the multiplication of a learnable weight, $W_1^n \in R^{F \times F}$ and the node feature, h_i^n to produce an embedded node feature, m_i^n, which has more information about protein-ligand complex. From the embedded node feature, m_i^n, the attention coefficient, e_{ij}^n between the ith and the jth nodes is computed as follows:

$$e_{ij}^n = (m_i^n)^T W_2^n m_j^n + (m_j^n)^T (W_2^n)^T m_i^n, \tag{3}$$

where $W_2^n \in R^{F \times F}$ is also a learnable matrix. e_{ij}^n implies the influence of the ith node to update the features of the jth node. The summation of $(m_i^n)^T W_2^n m_j^n$ and $(m_j^n)^T (W_2^n)^T m_j^n$ forces e_{ij}^n and e_{ji}^n to be equal. We adopted the softmax activation function to normalize the attention coefficient, e_{ij}^n, across neighboring nodes. The normalized attention coefficient, a_{ij}^n, is given by

$$a_{ij}^n = \frac{exp(e_{ij}^n)}{\sum_{j \in N_i} exp(e_{ij}^n)}, \tag{4}$$

where N_i is the set of the neighboring nodes of the ith node. Then, the current node feature, $\tilde{h}_i^n$ is calculated via the linear combination of the neighboring node features weighted by the attention coefficient, a_{ij}, with a ReLU activation function:

$$\tilde{h}_i^n = ReLU(\Sigma_j a_{ij}^n h_j^{n\prime}). \tag{5}$$

We also used the gate mechanism to effectively deliver the previous node features and the current node features to the next node features. The importance of the previous node features, z_i, is computed from h_i^n and $\tilde{h}_i^n$ as follows:

$$z_i = \sigma(W_3^n((h_i^n \| \tilde{h}_i^n))), \tag{6}$$

where σ is a sigmoid activation function which constrains z_i between 0 and 1, $(\cdot\|\cdot)$ is a concatenation operation, and $W_3^n \in R^{2F\times1}$ is a learnable weight vector. Lastly, the next node feature, h_i^{n+1}, is a linearly interpolated value between h_i^n and $\tilde{h}_i^n$:

$$h_i^{n+1} = z_i h_i^n + (1 - z_i)\tilde{h}_i^n. \tag{7}$$

We used three units of the gated GAT to incorporate intramolecular interactions into the node features.

Interaction network

The interaction network takes the previous set of node features, $H^n = \{h_1^n,\ h_2^n,\ \cdots,\ h_N^n\}$, and the adjacency matrix A^2 to generate the next set of node features, $H^{n+1} = \{h_1^{n+1},\ h_2^{n+1},\ \cdots,\ h_N^{n+1}\}$. The interaction network first multiplies h_i^n with a learnable weight, $W_1^n \in R^{F\times F}$ to get the set of embedded node features, $M^1 = \{m_1^1,\ m_2^1,\ \cdots,\ m_N^1\}$ as follows:

$$m_i^1 = W_1^n h_i^n, \tag{8}$$

where i is the index of node features. The interaction network also makes a set of interaction embedded node features, $M^2 = \{m_1^2,\ m_2^2,\ \cdots,\ m_N^2\}$ with each previous node feature, h_i^n, a learnable weight, $W_2^n \in R^{F\times F}$, and the adjacency matrix A^2. The interaction embedded feature of the ith node is represented by:

$$m_i^2 = \max_{j \in N_i}\{W_2^n h_j^n\}, \tag{9}$$

where N_i is the set of nodes which have interactions with the ith node. By maximum aggregation for each node, the set of interaction embedded node features, M^2, becomes the most important node feature element within nodes with intermolecular interactions. From M^1 and M^2, we can get a set of total node features, $H'^n = \{h_1'^n, h_2'^n, \cdots, h_N'^n\}$ through the summation and a ReLU activation function:

$$h_i'^n = ReLU(m_i^1 + m_i^2). \tag{10}$$

The next set of node features, H_i^{n+1}, can be obtained from a gated recurrent unit (GRU)[2] by using a set of total node feature, H'^n, as a hidden state input and the previous set of node features, H^n.

$$h_i^{n+1} = GRU(h_i'^n,\ h_i^n) \tag{11}$$

Total node features, H'^n, updates the previous set of node features, H^n, recursively in the GRU cell. As a result, the next set of node features, H^{n+1}, is more likely to reflect only important features of the given protein-ligand complex when updating node features.

The interaction network makes a significant role in our model by transforming a set of node features of the protein-ligand complex to contain information about intermolecular interactions. To make a set of node features that sufficiently contains intermolecular interaction information, the interaction network consists of three units.

(b) Physics-informed parameterized function

PIGNet consists of several physics-informed parameterized functions: four energy components and a rotor penalty. Each energy component is computed with a set of pair-wise node features, H^{concat}, which represented as equation (12). Each pair-wise node feature consists of two node features, h_i and h_j.

$$
\begin{aligned}
H^{concat} &= \{h_1^{concat},\ h_2^{concat},\ \cdots,\ h_{N^2}^{concat}\} \\
&= \{(h_1||h_1),\ (h_1||h_2),\ \cdots,\ (h_N||h_{N-1}),\ (h_N||h_N)\}
\end{aligned}
\tag{12}
$$

Each energy component depends on d_{ij}, and d'_{ij}, which are distance, and corrected minimum distance between the ith node and the jth node. d'_{ij} can be represented as follows:

$$
d'_{ij} = r_i + r_j + c \cdot b_{ij},
\tag{13}
$$

where r_i and r_j are van der Waals radii of the ith and the jth nodes respectively, and b_{ij} is corresponding correction between two nodes. For the constant c that scales b_{ij}, we used 0.2. The correction constant, b_{ij}, originates from a set of pair-wise node features, H^{concat}, by using learnable weights, $W^1 \in R^{2F \times F}$ and $W^2 \in R^{F \times 1}$, as the following:

$$
b_{ij} = tanh(W^2(ReLU(W^1(h_{ij}^{concat})))).
\tag{14}
$$

van der Waals interaction

We used the 12-6 Lennard-Jones potential to calculate a van der Waals (VDW) interaction term, e_{ij}^{vdw}, between the ith and j^{th} atoms. Equation (15) summarizes e_{ij}^{vdw}:

$$
e_{ij}^{vdw} = c_{ij}\left[\left(\frac{d'_{ij}}{d_{ij}}\right)^{12} - 2\left(\frac{d'_{ij}}{d_{ij}}\right)^{6}\right],
\tag{15}
$$

where c_{ij} indicates the minimum interaction energy which is also predicted from neural networks. We constrain the minimum and maximum values as 0.0178 and 0.0356, respectively, to render the predicted energy component similar to the true energy component. The maximum value of c_{ij} is referred from the parameter of AutoDock Vina for steric interactions.[1] Equation (16) summarizes the calculation of c_{ij}:

$$
c_{ij} = \sigma(W_2^{vdw}(ReLU(W_1^{vdw}h_{ij}^{concat}))) \times (0.0356 - 0.0178) + 0.0178,
\tag{16}
$$

where $W_1^{vdw} \in R^{2F \times F}$, and $W_2^{vdw} \in R^{F \times 1}$, are weight matrices. We consider all protein and ligand atom pairs except metal atoms whose van der Waals radii have high variance depending on atoms types. We obtain the total van der Waals energy, E^{vdw}, by summing e_{ij}^{vdw} of all possible pairs, as follows.

$$
E^{vdw} = \sum_{i,j} e_{ij}^{vdw}
\tag{17}
$$

The hydrogen bond, metal-ligand interaction, hydrophobic interaction components, and rotor penalty can be computed as described in the main article.

Hydrogen bond, Metal-ligand interaction, Hydrophobic interaction

Supplementary Table 2 shows SMARTS descriptors which are used to select the hydrogen bond donors and hydrogen bond acceptors.

hydrogen bond acceptor	[$([!#6;+0);!$([F,Cl,Br,I]); $([o,s,nX3]);!$([Nv5,Pv5,Sv4,Sv6])]
hydrogen bond donor	[!#6;!H0]

Supplementary Table 2: SMARTS descriptors for hydrogen bond acceptor and donor

2. Benchmark methods

CASF2016 benchmark metrics[6]

In this supplementary section, we explain the benchmark metrics we report as our results.

- **Scoring power**: This is defined as a linear correlation of predicted binding affinity and experimental binding data. The linear correlation is measured in the Pearson's correlation coefficient, R. In fact, the scoring functions in comparison with PIGNet produce log affinity ($logK_a$) values, while PIGNet computes binding free energy values [kJ/mol]. Nevertheless, since a constant multiplication converts $logK_a$ to the binding free energy, such a difference does not make a change in the R values.

- **Ranking power**: This refers to the ability of a model to correctly rank the binding affinities of the true binders for a certain target protein, given the binders' precise binding poses. The ranking power can be measured in terms of the average Spearman's rank correlation coefficient, ρ, the Kendall's rank correlation coefficient, τ, and the Predictive Index (PI). We only report Spearman's rank correlation coefficients for our experiments, as all three metrics are well-correlated.

- **Docking power**: This is the ability of a model to find the native ligand binding pose among decoys with computer-generated poses. The metric for the docking power measurement is the **overall success rate**, which counts a complex identified by a model as a successful case if it has a high conformational similarity (RMSD < 2Å) with the native binding pose.

- **Screening power**: This is the ability of a model to identify the true binding ligands for a given target protein among a set of random molecules. We measure the screening power in terms of the enhancement factor (EF) and success rate, averaged across all 57 target proteins. The success rate is computed as a ratio of highest-affinity binder among the top $\alpha(\%)$ ligands. The enhancement factor for a target protein is defined as follows:

$$EF_\alpha = \frac{NTB_\alpha}{NTB_{total} \times \alpha},$$ (18)

where NTB_α is the number of the true binders among the top $\alpha(\%)$ candidates ranked by a model, and NTB_{total} is the total number of the true binders for the given target protein. We cite our results in the average enhancement factor and success rate measured at $\alpha = 1\%$.

3. Interpretation of the physically modeled outputs

Distribution plot of atom-atom pairwise interaction in each energy component
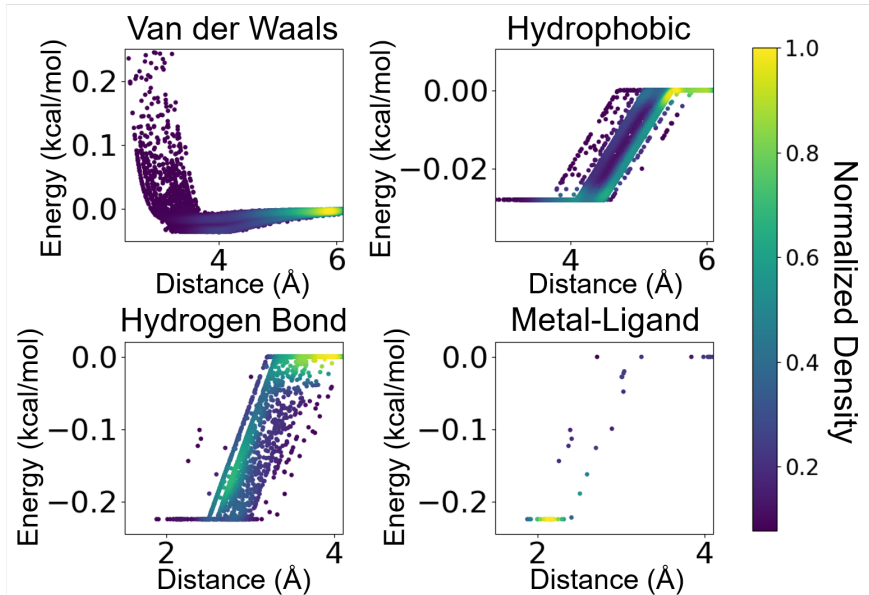
Supplementary Figure 1: The distance-energy plot for each energy component in the test set. The closer to the yellow, the larger the number of pairs corresponding to the data point.

By dissecting the predicted energies into individual energy components, we could observe that the model has learned the deviations within each energy component. Supplementary Figure 1 shows a distance-energy plot of each energy component, where the data points are the atom-atom pairs in the test set. Note that the pairwise energy plots are not a single solid line, but the multifariously deviated distributions. For the van der Waals component, while the plot generally complies with the form of the 12-6 Lennard-Jones potential, the deviations arise from the two learnable parameters, b_{ij} and c_{ij}. b_{ij} also contributes to the deviations in the hydrophobic, hydrogen bond, and metal energy components, as the parameter is used to calculate the corrected sum of van der Waals radii in equation (13).

4. Uncertainty quantification method

The uncertainty of predicted value, can be quantified in two types - aleatoric uncertainty, a noise inherent in data, and epistemic uncertainty, a uncertainty in model parameters. While the epistemic uncertainty can be reduced by increasing the quantity of training data, the aleatoric uncertainty is related to data quality rather than the quantity. The two types of the uncertainty can be added up to plausibly approximate the total uncertainty[3]. To quantify the uncertainties, we employed the following well-known methods:

- **Aleatoric uncertainty**: We adopt the Mean Variance Estimation (MVE) approach, which trains a model to produce both mean and aleatoric variance for a prediction. In the MVE scheme, the model is trained by minimizing the loss function derived as follows:

$$L_{aleatoric}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i)^2}\|y_i - f(x_i)\|^2 + \frac{1}{2}\log\sigma(x_i)^2. \tag{19}$$

  Mean $\hat{y}_i = f(x_i)$ and variance $\sigma(x_i)^2$ of a prediction are the functions of input x_i. In practice, the embedded feature vector of an input can be fed into two different output layers to produce the mean and variance.[5]

- **Epistemic uncertainty**: A popular solution to approximate the uncertainty from the model is Bayesian inference which gives probabilistic model parameters. Among various Bayesian inference methods, the MC-Dropout stands out as a practical method since it only requires the implementation of the widely used dropout regularization.

In fact, the PIGNet architecture is not compliant to a direct implementation of the aleatoric uncertainty inference method introduced in Kendall *et al.*[4]. Thus, we hereby introduce the following strategy to quantify the aleatoric uncertainty for PIGNet.

- **Aggregation of the atom-atom pairwise uncertainties**: Since our model adds up the predictions from each atom-atom pair to obtain an output for a protein-ligand complex, the corresponding pairwise aleatoric uncertainties should be aggregated as well. We chose to aggregate the pairwise uncertainties by multiplication over all pairs, as atom-atom pairwise interactions are not mutually independent.

- **Distance-dependency of uncertainty**: Our model takes atom-atom pair distances as inputs to account for the physical reality where the strength of interactions decay with distance. In an analogous manner, we conjecture that aleatoric uncertainties also decay with distance. We multiply a distance-dependent exponential decay function to pairwise aleatoric uncertainties, as the exponential decay had given better convergence compared to the reciprocal decay.

The whole process is formulated as follows:

$$\sigma^2 = \prod_{i,j}\sigma_{ij}^2 = \prod_{i,j}|W_2^{var}(ReLU(W_1^{var}(h_{ij}^{concat}))) \times a\exp(-bd_{ij})|, \tag{20}$$

where h_{ij}^{concat} is a concatenated embedding of the ligand i^{th} atom and the protein j^{th} atom, and d_{ij} is a distance between the atoms. FC layers and learnable parameters, a and b, were trained to minimize the aleatoric uncertainty loss function (equation (19)). Note that h_{ij}^{concat} and d_{ij} are the inputs to produce both predictive output and uncertainty.

References

[1] Abdul-rahman Allouche. "Gabedit–A Graphical User Interface for Computational Chemistry Softwares". In: *Journal of computational chemistry* 31.2 (2012), pp. 174–182.

[2] Junyoung Chung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *Preprint at arXiv:1412.3555* (2014).

[3] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33 rd International Conference on Machine Learning* 48 (2016).

[4] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Preprint at arXiv:1703.04977* (2017).

[5] Gabriele Scalia et al. "Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction". In: *Journal of Chemical Information and Modeling* (2020).

[6] Minyi Su et al. "Comparative Assessment of Scoring Functions: The CASF-2016 Update". In: *Journal of Chemical Information and Modeling* 59.2 (2019), pp. 895–913.