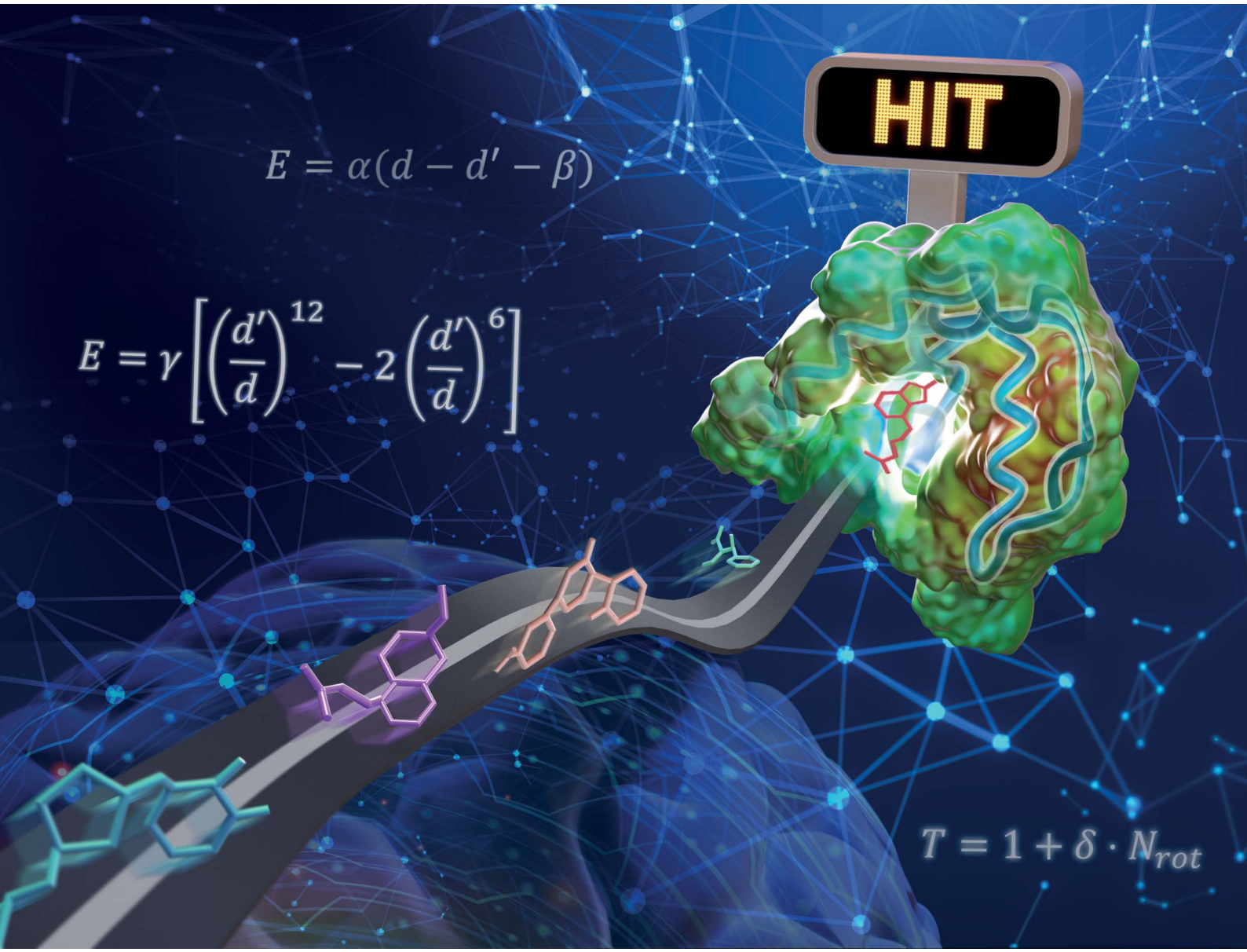
Volume 13
Number 13
7 April 2022
Pages 3613–3904

Chemical Science

rsc.li/chemical-science



ISSN 2041-6539

ROYAL SOCIETY OF CHEMISTRY

EDGE ARTICLE
Woo Youn Kim *et al.*
PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions

Chemical Science

ROYAL SOCIETY OF CHEMISTRY

EDGE ARTICLE

View Article Online
View Journal | View Issue

Cite this: *Chem. Sci.*, 2022, **13**, 3661

All publication charges for this article have been paid for by the Royal Society of Chemistry

Received 13th December 2021
Accepted 6th February 2022

DOI: 10.1039/d1sc06946b

rsc.li/chemical-science

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

Seokhyun Moon, ‡[a] Wonho Zhung, ‡[a] Soojung Yang, ‡§[a] Jaechang Lim [b] and Woo Youn Kim *[abc]

Recently, deep neural network (DNN)-based drug–target interaction (DTI) models were 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 is to predict 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 broader range of binding poses and ligands to training data. We validated our model, PIGNet, in the comparative assessment of scoring functions (CASF) 2016, demonstrating the outperforming docking and screening powers than previous methods. Our physics-informing strategy also enables the interpretation of predicted affinities by visualizing the contribution of ligand substructures, providing insights for further ligand optimization.

1 Introduction

Deep learning is a rapidly growing field of science. The remarkable success of deep learning in various applications such as natural language processing, video games, and computer vision has raised expectations for similar success in other fields, leading to various applications of deep learning algorithms. In particular, biomedical applications have become immediately one of the most active areas because they are not only socially influential but also scientifically challenging.[1–3] Despite the great expectation, however, deep learning has not yet shown its highest potential in this field, due to low generalization issues caused by scarce and heavily imbalanced data.[4,5] Making a reliable model for predicting drug–target interactions (DTIs), which is a key technology in the virtual screening of novel drug candidates, is one such example.[6]

As an ideal DTI prediction method should be reliable yet fast, high prediction accuracy and low computational cost are two essential factors.[7] However, docking methods[8–16] as the most popular conventional approach are fast enough but insufficiently accurate,[17–19] whereas more rigorous ones based on thermodynamic integration is computationally too expensive.[20,21] Such physics-based methods have an inherent limitation that low cost can only be achieved by losing their accuracy as a trade-off. In contrast, a data-driven approach can improve prediction accuracy at no additional inference cost, just by learning with more data. This distinct feature of the data-driven approach has encouraged the active development of deep learning-based drug–target interaction (DTI) models that accomplish both high accuracy and low cost.[22–30]

Among various deep learning-based models, the structure-based approach stands out for its accuracy; the spatial coordination of the protein and ligand is crucial in determining their interactions.[31] Some of the promising studies utilize 3-dimensional convolutional neural networks (3D CNNs),[32–41] graph neural networks (GNNs),[41–44] or feed-forward neural networks based on the atomic environment vectors.[45] These state-of-the-art approaches had significantly improved the accuracy of DTI prediction compared to docking calculations.

Despite the advance of previous structure-based models, their limited generalization ability remains a challenging problem towards better performance. In particular, the deficiency in 3D structural data of the protein–ligand complexes could drive the models to excessively memorize the features in training data. Such models, being over-fitted to the training

[a]*Department of Chemistry, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea. E-mail: wooyoun@kaist.ac.kr*

[b]*HITS Incorporation, 124 Teheran-ro, Gangnam-gu, Seoul 06234, Republic of Korea*

[c]*KI for Artificial Intelligence, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*

† Electronic supplementary information (ESI) available: the algorithm and implementation details of all the models used in the experiments, the details about the benchmark criteria and the physical interpretation result. See DOI: 10.1039/d1sc06946b

‡ These authors contributed equally to this work.

§ Currently at Computational and Systems Biology, MIT, 77 Massachusetts Ave, Cambridge, MA.

data, might fail to generalize in a broader context.[46] Several studies had suggested that deep learning-based models often learn the data-intrinsic bias instead of the underlying physics of the protein–ligand interaction as desired.[37,47,48] For instance, Chen *et al.*[47] 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 only by looking at the ligand structures, regardless of whether or not the protein structures are included as inputs. Moreover, they showed that such a memorization of wrong features can cause severe degradation in the performance for the proteins that have a high structural variance from those in the training data. The paper also reported that the 3D CNN and GNN models trained on the DUD-E dataset had considerably underperformed when they were tested with the ChEMBL and MUV datasets.[36,44] 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 appropriate physics as an inductive bias is a promising mean to improve the model generalization. If a model is trained to obey certain physical principles, the model is expected to generalize to unseen data that is dictated by the same physics. Several studies have indeed shown that the physics-informed models maintain their generalization ability for unseen data.[49–51]

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 learn 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 leverage a data augmentation strategy. In practice, screening libraries include a variety of compounds where most of them do not appear in the training set. Currently available experimental data on protein–ligand binding structures have very limited coverage on the structural diversity of all possible binding complexes. A model trained with a set of experimental binding structures, which would only include the stable binding poses, may fail to distinguish the stable poses from non-stable poses in the inference set.[47] Therefore, we augmented our training data with computationally generated random binding poses of protein–ligand pairs to improve the model generalization.[48]

To assess the generalization ability of the proposed model, we focused on the docking and the screening power of the CASF-2016 benchmark.[52] Previously, the DTI models had been evaluated in terms of the correlation between the predicted and the experimental binding affinities.[35,36,42–44] However, the high correlation does not automatically guarantee a good model generalization.[47] A well-generalized model should be able to successfully identify the true binding pose that has minimum energy and correctly rank the best binding molecule. The former criterion can be assessed in terms of docking power, while the latter one is related to screening power. Examining a model for both tasks is essential to ensure the model's ability to generalize in real-world settings such as virtual high-throughput screening (vHTS). We compared the benchmark results of PIGNet with traditional docking calculations and previous deep learning models and showed that our model significantly improved both docking power and screening power.

In addition to the improvement in the model performance, we show the interpretability of our model. While interpreting the underlying chemistry of DTI prediction is an essential step of drug discovery, previous deep learning models that take a complete black box approach were not practical in that sense.[53,54] On the contrary, physics-based deep learning models can offer interpretability since several intermediate variables of the models have certain physical meanings.[55] As our model predicts the interaction energy for each atom–atom pair, we can estimate the contribution of each ligand substructure in total binding free energy. Such an interpretation can provide the guidelines for the practitioners regarding ligand optimization – modifying the less contributing moieties into stronger binding moieties can be an example.

2 Method

2.1 Related works

2.1.1 Summary of previous structure-based deep DTI models. The 3D CNN takes a 3D rectangular grid that represents the coordinate of atoms of a protein–ligand complex as an input.[32–41] The proposed 3D CNN models outperformed docking programs for the PDBbind and the DUD-E dataset in terms of Pearson's correlation coefficient and AUC, respectively. Nevertheless, the high dimensionality of 3D rectangular representations and the absence of explicit representation of chemical interactions and bonds may put a limitation on 3D CNN models.[56] One of the promising alternatives is a GNN, which represents structural information as molecular graphs.[57] Each atom and chemical interaction (or bond) in a molecule is represented as a node and an edge in a graph, respectively. Also, molecular graphs can incorporate 3D structural information by regarding an atom–atom pair as neighbors only if its pairwise Euclidean distance is within a certain threshold. Moreover,

graph representations are invariant to translations and rotations, unlike grid representations of 3D CNN. Such advantages of graph representation over grid representation might have contributed to the state-of-the-art performance of GNNs in DTI predictions.[41–44,58]

2.1.2 Physics-informed neural networks. Greydanus *et al.*[49] proposed the Hamiltonian neural network as an effective method to model the systems that follow Hamiltonian mechanics. They used deep neural networks to predict parameters in the Hamiltonian equation and showed better generalization than previous neural networks. Pun *et al.*[50] proposed a physics-informed neural network for atomic potential modeling. The model predicts the parameters of each type of interatomic potential energy, instead of directly predicting the total energy of the system. This strategy had improved the model generalization for simulations performed outside the bonding region. In this work, with neural networks, we parameterize the equations that are derived from the physics of chemical interactions.

2.2 Model architecture

PIGNet is a deep learning model that predicts binding free energy of a given protein–ligand complex structure (Fig. 1). It 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 i^{th} node feature, h_i, is represented by a 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 initial node features and the construction of the two adjacency matrices are explained in the ESI.†

Our model consists of several units of gated graph attention networks (gated GATs) and interaction networks. Gated GATs and interaction networks update each node feature *via* two adjacency matrices that correspond to covalent bonds and intermolecular interactions. During the node feature update, gated GATs and interaction networks 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

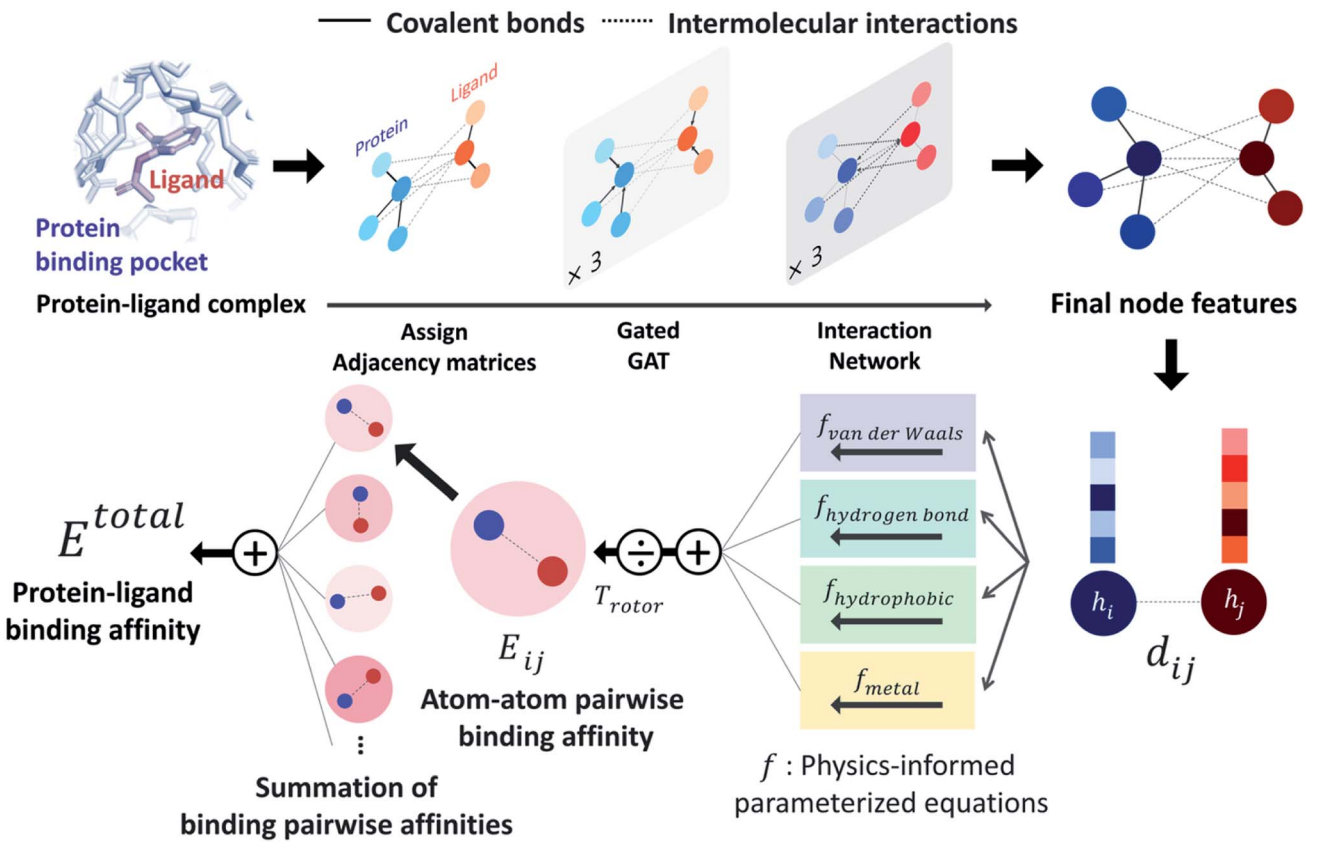


Fig. 1 Our model architecture. A protein–ligand complex is represented in a graph and adjacency matrices are assigned from the binding structure of the complex. Each node feature is updated through neural networks to carry the information of covalent bonds and intermolecular interactions. Given the distance 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.

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:

$$E^{\text{total}} = \frac{E^{\text{vdW}} + E^{\text{hbond}} + E^{\text{metal}} + E^{\text{hydrophobic}}}{T^{\text{rotor}}}. \quad (1)$$

2.2.1 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.[59] The attention coefficient, which implies the importance of the node, is calculated from the two nodes that are connected with 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 a coefficient, which is obtained from the previous and new node features. We describe the details of gated GAT in the ESI.†

2.2.2 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 sets 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).[60] We describe the details of the interaction network in the ESI.†

2.3 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 i^{th} node and the j^{th} 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 by 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. PIGNet employs learnable parameters to find an optimal weight for each 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 i^{th} node and the j^{th} node, respectively. $d_{ij}^{'}$ can be represented as follows:

$$d_{ij}^{'} = r_i + r_j + c \times b_{ij}, \quad (2)$$

where r is the vdW radius of each node, which are taken from X-Score parameters.[10] b_{ij} is a correction term between the two nodes which is resulted from a fully connected layer that accepts two node features h_i and h_j as inputs. We used 0.2 for a constant c that scales the correction term.

2.3.1 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:

$$E^{\text{vdW}} = \sum_{i,j} c_{ij} \left[\left(\frac{d_{ij}^{'}}{d_{ij}} \right)^{12} - 2 \left(\frac{d_{ij}^{'}}{d_{ij}} \right)^{6} \right], \quad (3)$$

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.

2.3.2 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 hydrophobic interaction ($E^{\text{hydrophobic}}$) share the same expression as shown in eqn (4) with different coefficients, c_1, c_2, and a learnable scalar variable, w.

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

Here, c_1 and c_2 are set as -0.7 and 0.0 for hydrogen bonds and metal–ligand interactions, respectively, 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. Each energy component is computed as a summation of corresponding atom–atom pairwise energy contribution coefficients, as described in eqn (5):

$$E = \sum_{i,j} e_{ij}. \quad (5)$$

We classified atoms into hydrogen bond acceptors, hydrogen bond donors, metal atoms, and hydrophobic atoms. Since hydrogen bonds appear between hydrogen bond donors and hydrogen bond acceptors, each atom that forms hydrogen bonds is selected by substructure matching of the general SMARTS[61] descriptors, which are summarized in the Table S2.† Metal atoms include Mg, Ca, Mn, Fe, Co, Ni, Cu, and Zn. Lastly,

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

2.3.3 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:

$$T^{\text{rotor}} = 1 + C_{\text{rotor}} \times N_{\text{rotor}}, \quad (6)$$

where N_{rotor} is the number of rotatable bonds and C_{rotor} is a positive learnable scalar variable. We used RDKit software[62] to calculate N_{rotor}.

2.4 Monte Carlo dropout (MCDO) and epistemic uncertainty

A total of 30 models is ensembled during the test phase, with the same dropout ratio, 0.1, as the training phase. We obtained the predicted values by averaging individual predictions and interpreted the variances as epistemic uncertainties. Here, we define PIGNet with and without MCDO as PIGNet (ensemble) and PIGNet (single), respectively.

2.5 Loss functions

The loss function of PIGNet consists of three components, L_{energy}, $L_{\text{derivative}}$, and $L_{\text{augmentation}}$ as in eqn (7):

$$L_{\text{total}} = L_{\text{energy}} + L_{\text{derivative}} + L_{\text{augmentation}}. \quad (7)$$

Fig. 2 explains the overall training scheme of PIGNet based on the three loss functions. 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},

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

where N_{train} is a number of training data. Minimizing L_{energy} enables the model to correctly predict the binding affinity of experimental 3D structures. $L_{\text{derivative}}$ is composed of the first and the second derivative of the energy with respect to the atomic position. Minimizing $L_{\text{derivative}}$ intends the model to sensitively find relatively stable poses. $L_{\text{augmentation}}$ is the loss related to the data augmentation.

2.5.1 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

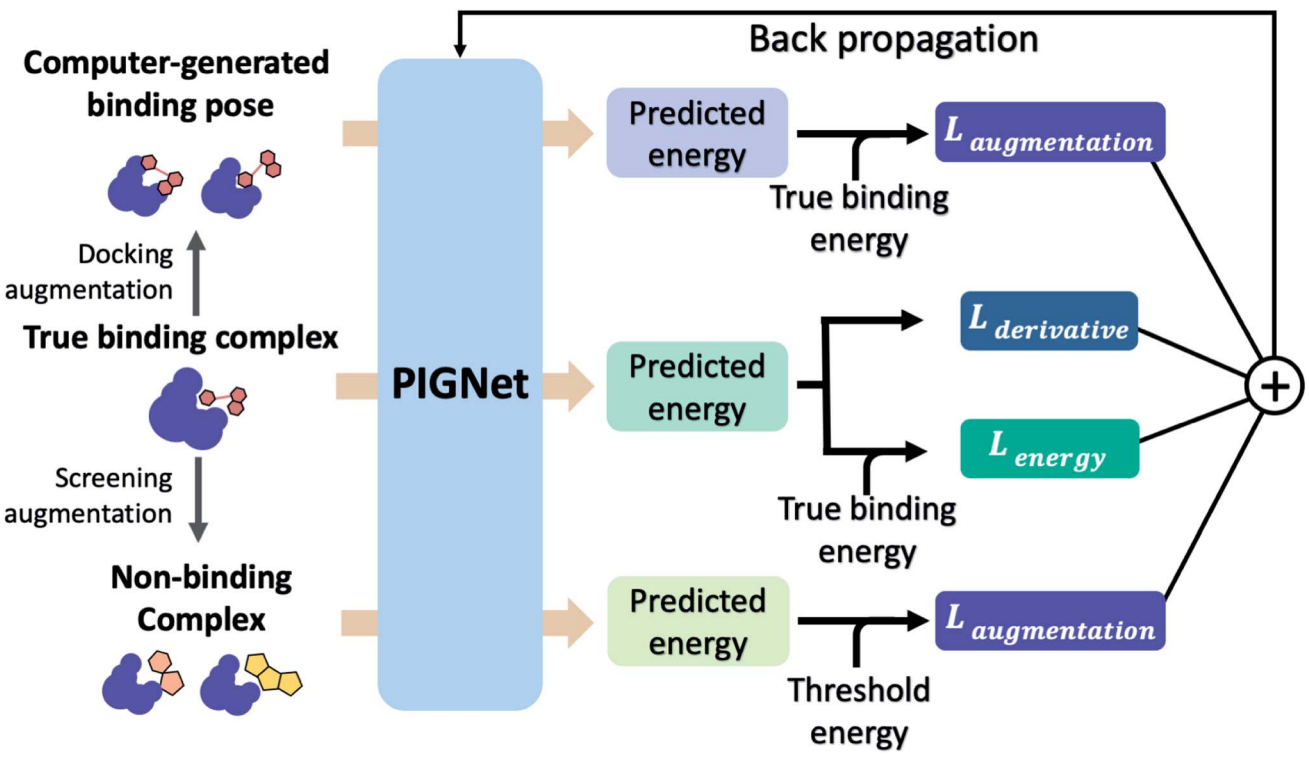


Fig. 2 The training scheme of PIGNet. We use three types of data in model training – true binding complex, true binder ligand–protein pair in a computer-generated binding pose, and non-binding decoy complex. PIGNet predicts binding free energy for each input. For a true binding complex, the model learns 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.

distinguish whether or not a ligand is at a stable position. Accordingly, we guide the model with the derivative loss, $L_{\text{derivative}}$, 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, $L_{\text{derivative}}$, is given as follows:

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

where q_i is the position of the i^{th} 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 the maximum value of the second derivative as C_{der2}, which is 20.0 in our model.

2.5.2 Data augmentation loss. Here, we constructed three different data augmentation-related loss functions; docking augmentation, random screening augmentation, and cross screening augmentation losses.

• Docking augmentation loss.

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 PDBbind dataset[63] 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:

$$L_{\text{docking}} = \sum_i \max\left(y_{\text{exp},i} - y_{\text{decoy},i}, \; -1\right), \tag{10}$$

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} larger than $y_{\text{exp}} + 1$.

• Random screening augmentation loss.

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.[64] 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:

$$\Delta G \geq -1.36 \log K_{\text{a}} = -6.8 \text{ kcal mol}^{-1}. \tag{11}$$

A model trained with random screening loss, $L_{\text{random\_screening}}$, 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, $L_{\text{random\_screening}}$, can be written as follows:

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

where y_{random} is the prediction energy of synthetic compounds from the IBS molecule library.[65] 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 loss.

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

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

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.

2.5.3 Total loss function. The total loss, L_{total}, is the weighted sum of all the loss terms: L_{energy}, $L_{\text{derivative}}$, L_{docking}, $L_{\text{random\_screening}}$, and $L_{\text{cross\_screening}}$. The total loss can be written as follows:

$$L_{\text{total}} = L_{\text{energy}} + c_{\text{derivative}} L_{\text{derivative}} + c_{\text{docking}} L_{\text{docking}} + c_{\text{random\_screening}} L_{\text{random\_screening}} + c_{\text{cross\_screening}} L_{\text{cross\_screening}}, \tag{14}$$

where $c_{\text{derivative}}$, c_{docking}, $c_{\text{random\_screening}}$, and $c_{\text{cross\_screening}}$ are hyper-parameters which are set as 10.0, 10.0, 5.0, and 5.0, respectively.

2.6 Baseline models

We constructed two baseline DNN models with the 3D CNN and 3D GNN architecture in comparison to PIGNet, namely a 3D CNN-based model and a 3D GNN-based model. For the 3D CNN-based model, we reimplemented the K_{DEEP} model from Jiménez *et al.*[34] Our rebuilt 3D CNN-based model is identical to K_{DEEP} 's, except we replaced the atom feature with those of PIGNet. We also constructed the 3D GNN-based model from PIGNet, but the model produces final outputs *via* fully connected layers instead of the physically modeled parametric equations.

2.7 Dataset

2.7.1 Training dataset and data augmentation. Our primary training set is the PDBbind 2019 refined set which

provides qualified binding affinity data and corresponding structure of protein–ligand complexes deposited in the protein databank (PDB).[63] We eliminated the redundant samples in the test set – the core set of PDBbind 2016 – from the training set. We used 4514 samples for the training set and 265 samples for the test set, which were remained after the data processing. During the processing, the amino acid residues whose minimum distance between the ligand is greater than 5 Å are cropped to reduce the number of atoms in the protein pocket.

Additionally, we constructed three different data augmentations; docking augmentation, random screening augmentation, and cross screening augmentation. Smina,[66] which is a fork of Autodock Vina, was used for generating decoy structures. For the docking augmentation, we generated 292 518 decoy structures using the PDBbind 2016 dataset. For the random screening augmentation and the cross screening augmentation, we generated 831 885 complexes using the IBS molecules[65] and 527 682 complexes based on the random cross binding, respectively. Any complexes in the test set are excluded during the augmentation.

2.7.2 Benchmark dataset. The CASF-2016 benchmark dataset[52] is originated from the PDBbind 2016 core set. After data processing, we used 283 samples for the scoring and ranking, 22 340 samples for the docking, and 1 612 867 samples for the screening benchmark. We also assessed the model with the CSAR NRC-HiQ (2010) 1 and 2 benchmark sets.[67] We could observe that some protein structures in the CSAR benchmark sets are highly similar to those in our training set. To assure a fair comparison, we further investigated if a bias in model evaluation can arise from this overlap. We built two subsets for each of the CSAR NRC-HiQ benchmark sets. First, we removed the samples if the same key existed in the training set. 48 and 37 samples remained after the exclusion for sets 1 and 2. Next, we excluded the samples that have at least 60% protein sequence similarity with one of the samples in the training set. 37 and 22 samples remained after the exclusion in sets 1 and 2, respectively.

2.7.3 Virtual screening library for human MAPK1 inhibitor. We further constructed the virtual screening library for human mitogen-activated protein kinase 1 (MAPK1) to test the model's ability to discover the true inhibitors from a large number of molecular candidates. We built a new training set excluding the homologs of human MAPK1. For the augmentation of the new training set, we followed the same protocol as in our primary training set. We used CD-HIT[68,69] for target clustering with a 60% sequence similarity cutoff. The targets in the cluster that includes human MAPK1 were considered as human MAPK1's homologs. Actives and inactives of human MAPK1 were obtained from the PubChem database.[70] The compounds with at least 0.7 similarities with one of the ligands either in the training set or the PubChem data itself were excluded. We named the compounds with the negative logarithm of half-maximal inhibitory concentration (pIC_{50}) higher than the specific threshold values as true inhibitors. Overall, 81, 45, and 27 compounds were considered true inhibitors when we used the criteria of 6, 7, and 8, respectively. Note that the higher the criteria are, the more potent the true inhibitors are. 56 413 inactives are combined with the true inhibitors to make up the virtual screening library. Finally, we generated 20 decoy structures per compound with the human MAPK1 protein (PDB id: 3I60) using Smina.

3 Results and discussions

3.1 Assessment of the model performance and the generalization ability

We primarily assessed the model with the CASF-2016 benchmark dataset. The CASF-2016 benchmark provides four different assessment tasks – scoring, ranking, docking, and screening – to evaluate DTI models in several aspects of virtual screening. The scoring power measures a linear correlation of predicted binding affinities and experimental values, calculated by a Pearson's correlation coefficient R. The ranking power measures an ability of a model to correctly rank the binding affinities of true binders of the actual binding pose, calculated by a Spearman's rank–correlation coefficient ρ. These two metrics are designed to assess the model's ability upon the stable-and-precise binding structures. On the other hand, the docking power and the screening power deal with the unnatural structures which are generated computationally. The docking power measures an ability of a model to find out the native binding pose of a ligand among computer-generated decoys, quantified as a success rate within the top N candidates. The screening power measures the ability of a model to identify the specific binding ligand for a given target protein among a set of random molecules, quantified as a success rate and an enhancement factor (EF) within the top α percent of candidates. Detailed equations of each metric are summarized in the ESI.†

In vHTS schemes, a DTI model should identify the most stable binding pose and correctly rank the protein–ligand pairs by their binding affinities at the same time. Indeed, the ranking, docking, and screening powers would be optimal if the model accurately predicts the value of the binding affinity for every given complex, that is, what the scoring power targets to achieve. However, experimental analysis on the CASF-2016 benchmark shows that the high scoring power does not guarantee high screening and docking powers.[52] We attribute this inconsistency to a limitation in the CASF-2016 scoring power benchmark – the scoring power itself cannot be a single criterion of a DTI model performance evaluation. Accordingly, we highlighted the models' docking and screening powers as indicators of model generalization.

Table 1 summarizes the performance of PIGNets, baseline models, and other published works for the CASF-2016 and the CSAR NRC-HiQ benchmarks. The reference scores of docking methods – AutoDock Vina,[8] GlideScore-SP,[13] and ChemPLP@GOLD[15] – were taken from Su *et al.*,[52] which ranks the first in a docking success rate, screening success rate, and screening EF, respectively. The performance of other deep learning approaches except K_{DEEP}[34] was directly taken from their references. The scores of K_{DEEP} were taken from Kwon *et al.*[39] since the docking power was not included in its original work.

PIGNet, both single and ensemble models, outperformed all other previous works in the CASF-2016 docking and screening

powers. Our best model achieves a top 1 docking success rate of 87%, a top 1% screening success rate of 55.4%, and a top 1% average EF of 19.6. The scoring and ranking power outperformed the docking methods while competitive with other deep learning-based approaches.

The Pearson's correlation coefficient for the CSAR NRC-HiQ benchmark sets showed consistent results with the CASF-2016 scoring power benchmark. Interestingly, the performance was not affected by the removal of the samples with a similarity threshold of 60% from the CSAR benchmark sets. Instead, we could observe consistent improvements in the performance. This slight improvement could be attributed to the smaller test set size, where a few data points can affect much of the result. Such a result suggests that our model has not merely been overfitted to the training set targets or homologous structures, as the performance would have been deteriorated if the overfitting was the case.

For baseline models, the 3D GNN-based model showed better performance on ranking, docking, and screening powers than the 3D CNN-based model. The difference might lead from the lack of the chemical interaction information in the 3D CNN-based model, where the 3D GNN-based model implicitly has. However, the 3D CNN-based model and the 3D GNN-based model fail to achieve high docking power and screening power. We attribute such low docking power to model overfitting 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 CASF-2016 metrics and the CSAR NRC-HiQ 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 strategies had contributed to the result through ablation studies.

3.2 Ablation study of two main strategies

We attribute the improvement of the model performance to two major strategies that have been utilized; the physics-informed parameterized functions introduced in the previous section and the data augmentation. In this section, we carried out an ablation study to decouple the effects of the two strategies and summarized the results in Table 2.

3.2.1 Effect of the physics-informed parametrized functions. We can observe the effect of the physics-informed model by comparing the performances of the 3D GNN-based model and PIGNet since a 3D GNN-based model is identical to PIGNet except for the parametric equations. As expected, the effect was not critical for the CASF-2016 scoring and ranking powers. However, the employment of the physics-informed model has resulted in a significant increase in docking and screening powers. We can infer that the incorporation of the parametric equations has contributed to enhancing the model generalization. Incorporating a certain form of equations may impose an excessive inductive bias on the model, which can lead to the model under-fitting. However, it turns out to be unlikely from the comparable scoring powers of PIGNet and the 3D GNN-based model. Especially, PIGNet without data augmentation still shows better docking power than the 3D GNN-based model

Table 1 Benchmark results on the CASF-2016 and the CSAR NRC-HiQ dataset. R, ρ indicate the Pearson correlation coefficient and Spearman's rank correlation coefficient, respectively. The top 1 score was used for a docking success rate, and the top 1% rate was used for an average EF and a screening success rate. $\Delta_{Vina}RF_{20}$[71] was excluded from the comparison, as it was fine-tuned on the PDBbind 2017 data, which in fact includes ~ 50% of data in the CASF-2016 test set. Numbers in the parenthesis of CSAR NRC-HiQ benchmarks are for the test sets that have excluded the targets with protein sequence similarity higher than 60% with the training set. The results of the 3D CNN-based, the 3D GNN-based model, and PIGNets were averaged from 4-fold models. The highest values of each column are shown in bold

	CASF-2016					CSAR NRC-HiQ	
	Docking	Screening		Scoring	Ranking	Set 1	Set 2
Model	Success rate	Average EF	Success rate	R	ρ	R	R
AutoDock Vina[8]	84.6%	7.7	29.8%	0.604	0.528	—	—
GlideScore-SP[13]	84.6%	11.4	36.8%	0.513	0.419	—	—
ChemPLP@GOLD[15]	83.2%	11.9	35.1%	0.614	0.633	—	—
K_{DEEP}[39]	29.1%	—	—	0.701	0.528	—	—
AK-Score (single)[39]	34.9%	—	—	0.719	0.572	—	—
AK-Score (ensemble)[39]	36.0%	—	—	**0.812**	0.67	—	—
AEScore[45]	35.8%	—	—	0.800	0.640	—	—
Δ-AEScore[45]	85.6%	6.16	19.3%	0.790	0.590	—	—
3D CNN-based model	48.2%	3.9	10.1%	0.687	0.580	0.738(0.756)	**0.804**(0.837)
3D GNN-based model	67.7%	10.2	28.5%	0.667	0.604	0.514(0.566)	0.627(0.723)
PIGNet (single)	85.8%	18.5	50.0%	0.749	0.668	**0.774**(**0.798**)	0.799(**0.863**)
PIGNet (ensemble)	**87.0%**	**19.6**	**55.4%**	0.761	**0.682**	0.768(**0.798**)	0.800(0.857)

Table 2 The CASF-2016 benchmark results for the 3D GNN-based model and PIGNet (single) with and without using data augmentation. The top 1 score was used for a docking success rate, and the top 1% rate was used for an average EF and a screening success rate. The highest values within the same model are shown in bold

		CASF-2016				
		Docking	Screening		Scoring	Ranking
Model	Use Data augmentation?	Success rate	Average EF	Success rate	R	ρ
3D GNN-based model	No	29.9%	1.4	4.9%	**0.772**	0.604
	Yes	**66.6%**	**10.2**	**28.5%**	0.689	**0.629**
PIGNet (single)	No	77.4%	6.6	24.6%	**0.792**	**0.672**
	Yes	**85.8%**	**18.5**	**50.0%**	0.749	0.668

trained with the augmented data. Although adding a large number of augmented data improves the performance, the data augmentation strategy itself 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 following section.

3.2.2 Effect of the DTI-adapted data augmentation strategy. The PDBbind dataset is one of the most representative training datasets for the data-driven DTI models, providing both 3D binding structures and the binding affinities of the protein–ligand complexes.[63] However, the PDBbind dataset is suspected to hold an intrinsic bias;[47] its ligands have insufficient chemical diversity and only the binding structures in minimum energy poses are given. To expand the chemical space which the model learns, we additionally included 1 652 085 augmented samples in the training set. In particular, computationally generated structures, which happen to be more unstable than the actual structures, are used for learning. Table 2 clearly shows the effect of the DTI-adapted data augmentation strategy on the generalization ability. The augmentation apparently improved the docking and screening power of both the 3D GNN-based model and PIGNet. 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, and the CSAR NRC-HiQ – 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.

3.3 Virtual screening of human MAPK1 inhibitor

To be practically used for the virtual screening, DTI models should generalize well, even for unseen targets and ligands. PIGNet showed state-of-the-art performances among other docking methods and deep learning methods for the conventionally used docking and screening benchmarks. In an attempt towards an even fairer comparison, we noticed that common protein homologs exist in both training and benchmark sets. In practice, the target and its true inhibitors might be distant from the training data distribution. Hence, to ascertain the applicability of our model, we conducted a virtual screening case study in more realistic settings where we excluded from the training set the targets and ligands similar to the given target and its true inhibitors.

We selected the human MAPK1 protein, an essential therapeutic target[72–74] for our case study. The top 1% average EF of Smina, 3D CNN, and 3D GNN-based models were compared with that of PIGNet.

The virtual screening results shown in Table 3 are consistent with the CASF-2016 screening benchmark results, where PIGNets outperform the baseline models and a docking method. The trend was consistent regardless of the ratio of the true inhibitors, which is determined by a pIC_{50} criteria. While this single-case study does not give us statistical significance, we can inspect the potential applicability of our model on virtual screening for unseen targets.

3.4 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-

Table 3 The virtual screening results of the human MAPK1 inhibitors. The results of the 3D CNN-based, the 3D GNN-based model, and PIGNets were averaged from 4-fold models. Top 1% rate was used for an average EF. The highest values of each column are shown in bold

	pIC_{50} Criteria for true inhibitor		
	6	7	8
Model	Average EF		
Smina[66]	9.9	13.3	18.5
3D CNN-based model	7.7	6.7	9.3
3D GNN-based model	9.3	11.1	12.0
PIGNet (single)	20.1	21.1	24.1
PIGNet (ensemble)	**21.3**	**21.7**	**26.9**

receptor type 1 (PTPN1) and platelet activating factor acetylhydrolase (PAF-AH). The result is illustrated in Fig. 3a, where two ligands for each protein are compared regarding the predicted substructural energy contributions and the inhibitory constant, K_i. Each ligand pair has high structural similarity, and only differs in red-circled moieties. For PTPN1, the model predicts a greater energy contribution for the tetramethyl cyclohexyl moiety than the cyclohexyl moiety. 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 a greater energy contribution of the phenyl group compared to the methyl group. For both proteins, the blue-circled common substructures are predicted to have similar energy contributions. This implies the predicted energy contribution of each substructure provides a physically meaningful interpretation, which can take further advantages to strengthen the total binding affinity towards the target protein during the ligand optimization.

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 Fig. 3b and c. Fig. 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–ligand energy components in the Fig. S1.†

Fig. 3c shows that the naturally occurring deviations within the atom–atom pairs in our model are the consequences of learning sufficient physics information. The corrected sum of

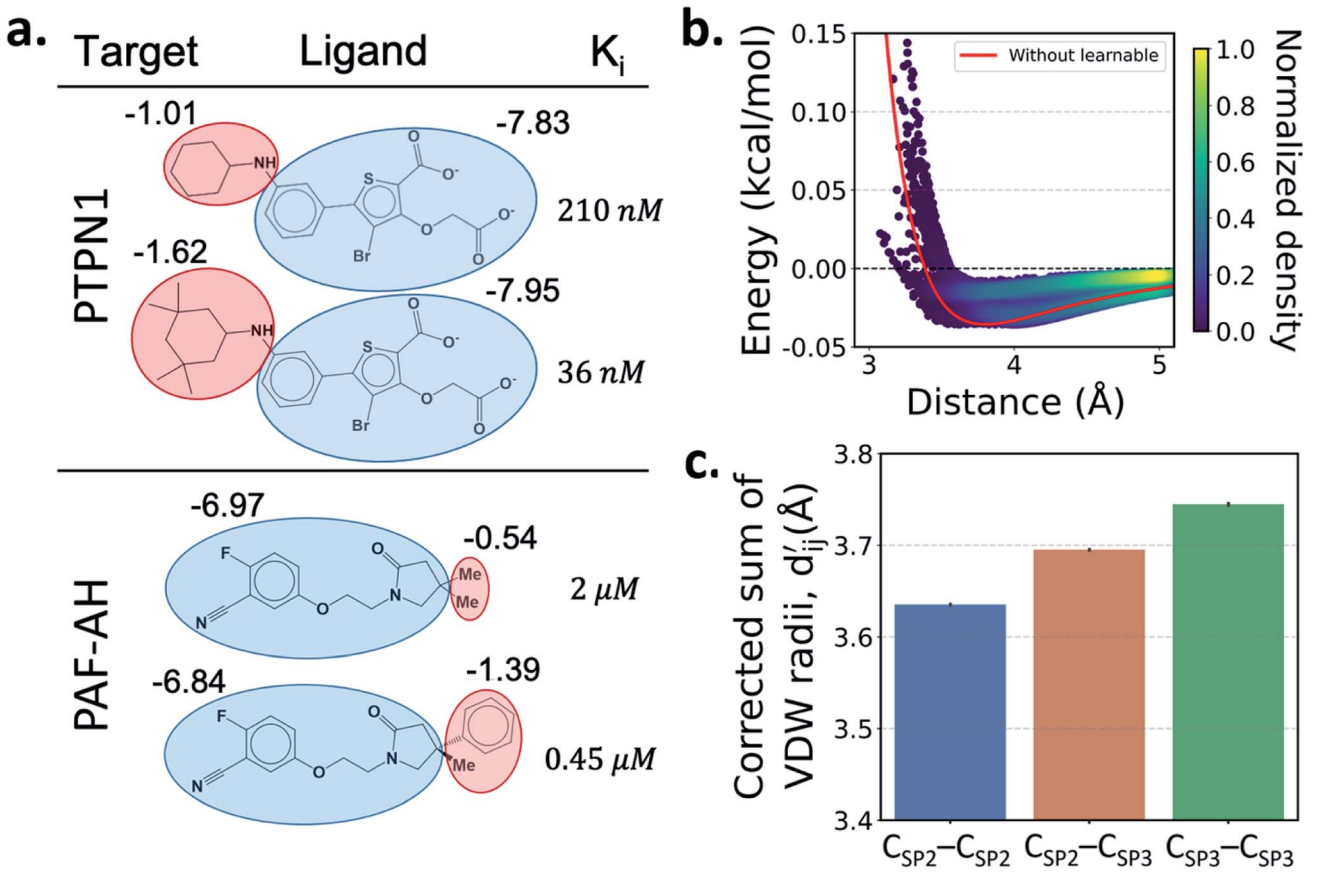


Fig. 3 Interpretation of the predicted outcomes. (a) Substructural analysis of ligands for two target proteins. Protein-tyrosine phosphatase non-receptor type 1 (PTPN1) and platelet activating factor acetylhydrolase (PAF-AH). The blue and red circles indicate common and different substructures, respectively, and the predicted energy contribution (unit: kcal mol^{-1}) 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. The results include 95% confidence intervals.

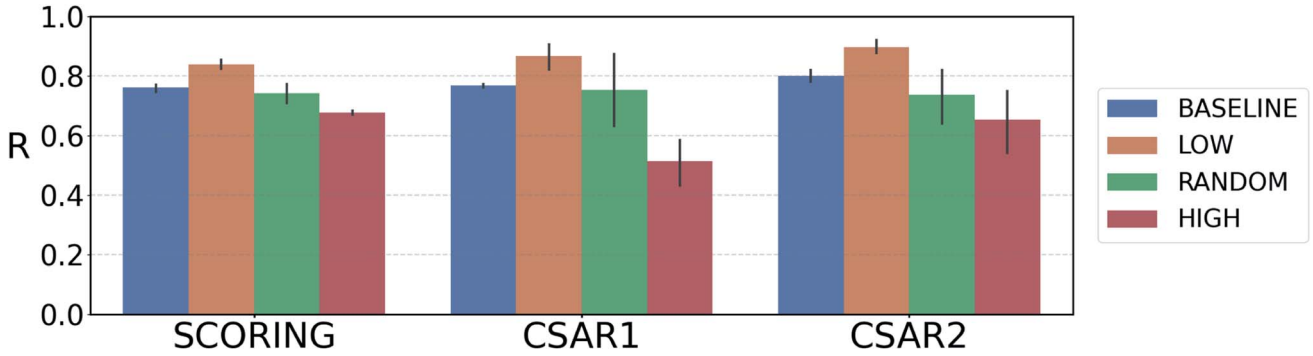


Fig. 4 Plot of the average Pearson's correlation coefficients, *R*, of the 4-fold PIGNet model, with or without the uncertainty estimator, on the datasets classified according to the total uncertainty. PIGNet with the uncertainty estimator – low: the lowest third, random: the randomly selected one third, high: the highest third of the uncertainty distribution. PIGNet without Monte Carlo dropout – baseline: the scores of a single PIGNet model shown in the Table 1. The lower the uncertainty, the more probable the model would have correctly predicted the result. Error bars represent 95% confidence intervals. PIGNet was tested at the 2 300^{th} training epoch with and without Monte Carlo dropout.

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 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, the larger the s-character of the carbon atoms, the shorter was the corrected sum of vdW radii.

3.5 Epistemic uncertainty quantification of PIGNet

For reliable virtual screening, it is important to screen out the false positive binders and secure the true positives.[75] Unfortunately, most positive returns from docking programs turn out to be false positives.[76] DNN-based models may also have the same problem. In particular, the data-deficient nature of training DTI models might render the DNN models less fit to out-of–domain complexes,[47] producing false positives. One possible way to reduce the false positives is to use the uncertainty of the predictions and to filter unreliable positive predictions. We employed a Monte Carlo dropout (MCDO), a practical Bayesian inference method utilizing dropout regularization, to estimate epistemic uncertainties which are originated from the model uncertainty.[77]

We quantified prediction uncertainties for the samples in three datasets – CSAR NRC-HiQ 1 and 2, the CASF-2016 scoring power. In Fig. 4, the 'low', 'random', and 'high' batches in terms of the prediction uncertainties are in descending order in the value of Pearson's correlation, *R*. Such a result resonates with our expectation; the lower the uncertainty, the more probable the model would have correctly predicted the result. This result shows that the prediction uncertainties of our model can be properly quantified. A previous study reported that the robust uncertainty quantification of model predictions can be evidence of good generalization ability.[78] Thus, it might be possible to relate the high generalization ability of our model to the success in uncertainty quantification.

Comparing the *R* values of the 'random' and 'baseline' batches in Fig. 4 enables to evaluate the general performance of PIGNet with and without the uncertainty estimator. The result confirms that the addition of uncertainty estimator does not harm the model performance. Furthermore, the comparison shows that uncertainty quantification can be used to filter out the false positives.

4 Conclusion

In this work, we studied the inadequate generalization problem of deep learning models that are often encountered in real-world applications where data for training is very scarce and imbalanced. As an important practical example, we focused on drug–target interaction (DTI) models for the fast and reliable virtual screening of drug candidates. The resulting model, named PIGNet, could achieve better generalization as well as higher accuracy compared to other deep learning models. We attribute the success of our model to the following two strategies. 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. These strategies can be readily adopted to other science fields where similar data problems are expected and the related physics is well-established. Such applications would include materials design, structural biology, and particle physics. A similar improvement in the generalization reported in the atomistic modeling of solid materials[50] is one such example.

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 by joining our model with generative models such as Imrie *et al.*[79] 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, uncertainty quantification has practical benefits in virtual screening scenarios.

Still, our model has room for improvements regarding the parametric expression of atom pairwise interactions. To further improve the predictive power of our model, we can test out alternative expressions for the energy components. For example, as an expression for the van der Waals interaction, it would be possible to use the Hamaker formula parameterized with a learnable Hamaker constant.[80,81] Additionally, our model could serve as a reliable oracle for the molecular generative models,[82,83] in case the design objective is focused on the target binding affinity.

Data availability

Training datasets can be preprocessed from the codes available at github: https://github.com/ACE-KAIST/PIGNet.

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.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by Basic Science Research Programs through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017R1E1A1A01078109).

References

1 P. Mamoshina, A. Vieira, E. Putin and A. Zhavoronkov, *Mol. Pharmaceutics*, 2016, **13**, 1445–1454.
2 C. Cao, F. Liu, H. Tan, D. Song, W. Shu, W. Li, Y. Zhou, X. Bo and Z. Xie, *Genomics, Proteomics Bioinf.*, 2018, **16**, 17–32.
3 R. Zemouri, N. Zerhouni and D. Racoceanu, *Appl. Sci.*, 2019, **9**, 1526.
4 M. Wainberg, D. Merico, A. Delong and B. J. Frey, *Nat. Biotechnol.*, 2018, **36**, 829–838.
5 J. G. Greener, S. M. Kandathil, L. Moffat and D. T. Jones, *Nat. Rev. Mol. Cell Biol.*, 2021, 1–16.
6 A. L. Hopkins, *Nature*, 2009, **462**, 167–168.
7 M. Bagherian, E. Sabeti, K. Wang, M. A. Sartor, Z. Nikolovska-Coleska and K. Najarian, *Briefings Bioinf.*, 2021, **22**, 247–269.
8 O. Trott and A. J. Olson, *J. Comput. Chem.*, 2010, **31**, 455–461.
9 S. Ruiz-Carmona, D. Alvarez-Garcia, N. Foloppe, A. B. Garmendia-Doval, S. Juhos, P. Schmidtke, X. Barril, R. E. Hubbard and S. D. Morley, *PLoS Comput. Biol.*, 2014, **10**, 1–7.
10 R. Wang, L. Lai and S. Wang, *J. Comput.-Aided Mol. Des.*, 2002, **16**, 11–26.
11 A. N. Jain, *J. Med. Chem.*, 2003, **46**, 499–511.
12 G. Jones, *J. Mol. Biol.*, 1997, **267**, 727–748.
13 R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, *et al.*, *J. Med. Chem.*, 2004, **47**, 1739–1749.
14 C. M. Venkatachalam, X. Jiang, T. Oldfield and M. Waldman, *J. Mol. Graphics Modell.*, 2003, **21**, 289–307.
15 O. Korb, T. Stützle and T. E. Exner, *J. Chem. Inf. Model.*, 2009, **49**, 84–96.
16 W. J. Allen, T. E. Balius, S. Mukherjee, S. R. Brozell, D. T. Moustakas, P. T. Lang, D. A. Case, I. D. Kuntz and R. C. Rizzo, *J. Comput. Chem.*, 2015, **36**, 1132–1156.
17 B. Waszkowycz, D. E. Clark and E. Gancia, *Wires Comput. Mol. Sci.*, 2011, **1**, 229–259.
18 A. R. Leach, B. K. Shoichet and C. E. Peishoff, *J. Med. Chem.*, 2006, **49**, 5851–5855.
19 Y. C. Chen, *Trends Pharmacol. Sci.*, 2015, **36**, 78–95.
20 M. R. Shirts, D. L. Mobley and J. D. Chodera, *Annu. Rep. Comput. Chem.*, 2007, **3**, 41–59.
21 C. Chipot, X. Rozanska and S. B. Dixit, *J. Comput.-Aided Mol. Des.*, 2005, **19**, 765–770.
22 A. Fitriawan, I. Wasito, A. F. Syafiandini, M. Amien and A. Yanuar, *International Conference on Computer, Control, Informatics and its Applications*, IC3INA, 2016.
23 H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, 821–829.
24 M. Thafar, A. B. Raies, S. Albaradei, M. Essack and V. B. Bajic, *Front. Chem.*, 2019, **7**, 782.
25 C. F. Lipinski, V. G. Maltarollo, P. R. Oliveira, A. B. F. da Silva and K. M. Honorio, *Front. Robot. AI*, 2019, **6**, 108.
26 M. Tsubaki, K. Tomii and J. Sese, *Bioinformatics*, 2019, **35**, 309–318.
27 I. Lee, J. Keum and H. Nam, *PLoS Comput. Biol.*, 2019, **15**, 1–21.
28 S. Zheng, Y. Li, S. Chen, J. Xu and Y. Yang, *Nat. Mach. Intell.*, 2020, **2**, 134–140.
29 K. Huang, T. Fu, L. M. Glass, M. Zitnik, C. Xiao and J. Sun, *Bioinformatics*, 2020, **36**, 5545–5547.
30 Y.-B. Wang, Z.-H. You, S. Yang, H.-C. Yi, Z.-H. Chen and K. Zheng, *BMC Med. Inf. Decis. Making*, 2020, **20**, 1–9.
31 S. K. Panday and I. Ghosh, *Struct. Bioinf.*, 2019, 109–175.
32 F. Imrie, A. R. Bradley, M. van der Schaar and C. M. Deane, *J. Chem. Inf. Model.*, 2018, **58**, 2319–2330.
33 M. M. Stepniewska-Dziubinska, P. Zielenkiewicz and P. Siedlecki, *Bioinformatics*, 2018, **34**, 3666–3674.
34 J. Jiménez, M. Škalič, G. Martínez-Rosell and G. De Fabritiis, *J. Chem. Inf. Model.*, 2018, **58**, 287–296.

35 I. Wallach, M. Dzamba and A. Heifets, preprint, arXiv:1510.02855, 2015, https://arxiv.org/abs/1510.02855.
36 M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri and D. R. Koes, *J. Chem. Inf. Model.*, 2017, **57**, 942–957.
37 J. A. Morrone, J. K. Weber, T. Huynh, H. Luo and W. D. Cornell, *J. Chem. Inf. Model.*, 2020, **60**, 4170–4179.
38 L. Zheng, J. Fan and Y. Mu, *ACS Omega*, 2019, **4**, 15956–15965.
39 Y. Kwon, W.-H. Shin, J. Ko and J. Lee, *Int. J. Mol. Sci.*, 2020, **21**, 8424.
40 H. Hassan-Harrirou, C. Zhang and T. Lemmin, *J. Chem. Inf. Model.*, 2020, **60**, 2791–2802.
41 D. Jones, H. Kim, X. Zhang, A. Zemla, G. Stevenson, W. F. D. Bennett, D. Kirshner, S. E. Wong, F. C. Lightstone and J. E. Allen, *J. Chem. Inf. Model.*, 2021, **61**, 1583–1592.
42 E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar and V. S. Pande, *ACS Cent. Sci.*, 2018, **4**, 1520–1530.
43 W. Torng and R. B. Altman, *J. Chem. Inf. Model.*, 2019, **59**, 4131–4149.
44 J. Lim, S. Ryu, K. Park, Y. J. Choe, J. Ham and W. Y. Kim, *J. Chem. Inf. Model.*, 2019, **59**, 3981–3988.
45 R. Meli, A. Anighoro, M. J. Bodkin, G. M. Morris and P. C. Biggin, *J. Cheminf.*, 2021, **13**, 1–19.
46 D. M. Hawkins, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 1–12.
47 L. Chen, A. Cruz, S. Ramsey, C. J. Dickson, J. S. Duca, V. Hornak, D. R. Koes and T. Kurtzman, *PLoS one*, 2019, **14**, e0220113.
48 J. Scantlebury, N. Brown, F. Von Delft and C. M. Deane, *J. Chem. Inf. Model.*, 2020, **60**, 3722–3730.
49 S. Greydanus, M. Dzamba and J. Yosinski, *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019, vol. 32, pp. 15379–15389.
50 G. P. Pun, R. Batra, R. Ramprasad and Y. Mishin, *Nat. Commun.*, 2019, **10**, 2339.
51 L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley and K. Burke, *Phys. Rev. Lett.*, 2021, **126**, 036401.
52 M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li and R. Wang, *J. Chem. Inf. Model.*, 2019, **59**, 895–913.
53 T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang and T. Aittokallio, *Briefings Bioinf.*, 2015, **16**, 325–337.
54 K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey and P. Zhang, *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.
55 T. Zubatiuk, B. Nebgen, N. Lubbers, J. S. Smith, R. Zubatyuk, G. Zhou, C. Koh, K. Barros, O. Isayev and S. Tretiak, *J. Chem. Phys.*, 2021, **154**, 244108.
56 D. S. Karlov, S. Sosnin, M. V. Fedorov and P. Popov, *ACS Omega*, 2020, **5**, 5150–5159.
57 J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li and M. Sun, *AI Open*, 2020, **1**, 57–81.
58 M. Jiang, Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan and Z. Wei, *RSC Adv.*, 2020, **10**, 20701–20712.
59 P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, *International Conference on Learning Representations*, 2018.
60 J. Chung, C. Gulcehre, K. Cho and Y. Bengio, preprint, arXiv:1412.3555, 2014, https://arxiv.org/abs/1412.3555.
61 R. E. Wunderlich, T. F. Wenisch, B. Falsafi and J. C. Hoe, *Conference Proceedings – Annual International Symposium on Computer Architecture*, ISCA, 2003, pp. 84–95.
62 *RDKit: Open-source cheminformatics*, http://www.rdkit.org.
63 Z. Liu, M. Su, L. Han, J. Liu, Q. Yang, Y. Li and R. Wang, *Acc. Chem. Res.*, 2017, **50**, 302–309.
64 J. P. Hughes, S. Rees, S. B. Kalindjian and K. L. Philpott, *Br. J. Pharmacol.*, 2011, **162**, 1239–1249.
65 InterBioScreen Ltd, http://www.ibscreen.com.
66 D. R. Koes, M. P. Baumgartner and C. J. Camacho, *J. Chem. Inf. Model.*, 2013, **53**, 1893–1904.
67 J. B. Dunbar Jr, R. D. Smith, C.-Y. Yang, P. M.-U. Ung, K. W. Lexa, N. A. Khazanov, J. A. Stuckey, S. Wang and H. A. Carlson, *J. Chem. Inf. Model.*, 2011, **51**(9), 2036–2046.
68 W. Li and A. Godzik, *Bioinformatics*, 2006, **22**, 1658–1659.
69 L. Fu, B. Niu, Z. Zhu, S. Wu and W. Li, *Bioinformatics*, 2012, **28**, 3150–3152.
70 Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang and S. H. Bryant, *Nucleic Acids Res.*, 2009, **37**, W623–W633.
71 C. Wang and Y. Zhang, *J. Comput. Chem.*, 2017, **38**, 169–177.
72 J. S. Sebolt-Leopold, *Oncogene*, 2000, **19**, 6594–6599.
73 A. S. Dhillon, S. Hagan, O. Rath and W. Kolch, *Oncogene*, 2007, **26**, 3279–3290.
74 L. Miao and H. Tian, *J. Drug Targeting*, 2020, **28**, 154–165.
75 E. H. B. Maia, L. C. Assis, T. A. de Oliveira, A. M. da Silva and A. G. Taranto, *Front. Chem.*, 2020, **8**, 343.
76 R. Sink, S. Gobec, S. Pecar and A. Zega, *Curr. Med. Chem.*, 2010, **17**, 4231–4255.
77 Y. Gal and Z. Ghahramani, *Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, 2016.
78 G. Scalia, C. A. Grambow, B. Pernici, Y.-P. Li and W. H. Green, *J. Chem. Inf. Model.*, 2020, 2697–2717.
79 F. Imrie, A. R. Bradley, M. van der Schaar and C. M. Deane, *J. Chem. Inf. Model.*, 2020, **60**, 1983–1995.
80 C. M. Roth, B. L. Neal and A. M. Lenhoff, *Biophys. J.*, 1996, **70**, 977–987.
81 V. A. Parsegian, *van der Waals forces: a handbook for biologists, chemists, engineers, and physicists*, Cambridge University Press, 2005.
82 A. Grosnit, R. Tutunov, A. M. Maraval, R.-R. Griffiths, A. I. Cowen-Rivers, L. Yang, L. Zhu, W. Lyu, Z. Chen and J. Wang, *et al.*, arXiv preprint arXiv:2106.03609, 2021.
83 S. Yang, D. Hwang, S. Lee, S. Ryu and S. J. Hwang, *Advances in Neural Information Processing Systems*, 2021.