

Calibrated geometric deep learning improves kinase–drug binding predictions

Received: 20 December 2022

Accepted: 29 September 2023

Published online: 6 November 2023

 Check for updates

Yunan Luo^{1,3}✉, Yang Liu^{2,3} & Jian Peng^{1,2}✉

Protein kinases regulate various cellular functions and hold significant pharmacological promise in cancer and other diseases. Although kinase inhibitors are one of the largest groups of approved drugs, much of the human kinase remains unexplored but potentially druggable. Computational approaches, such as machine learning, offer efficient solutions for exploring kinase–compound interactions and uncovering novel binding activities. Despite the increasing availability of three-dimensional (3D) protein and compound structures, existing methods predominantly focus on exploiting local features from one-dimensional protein sequences and two-dimensional molecular graphs to predict binding affinities, overlooking the 3D nature of the binding process. Here we present KDBNet, a deep learning algorithm that incorporates 3D protein and molecule structure data to predict binding affinities. KDBNet uses graph neural networks to learn structure representations of protein binding pockets and drug molecules, capturing the geometric and spatial characteristics of binding activity. In addition, we introduce an algorithm to quantify and calibrate the uncertainties of KDBNet’s predictions, enhancing its utility in model-guided discovery in chemical or protein space. Experiments demonstrated that KDBNet outperforms existing deep learning models in predicting kinase–drug binding affinities. The uncertainties estimated by KDBNet are informative and well-calibrated with respect to prediction errors. When integrated with a Bayesian optimization framework, KDBNet enables data-efficient active learning and accelerates the exploration and exploitation of diverse high-binding kinase–drug pairs.

Proteins are vital drug targets for therapeutic purposes, but at present only 11% of human proteome can be targeted by drugs or small molecules, leaving a large proportion to be explored for therapeutic opportunities¹. A group of proteins called kinase is of particular interest as drug targets because of their tractability in drug development and diverse pharmacological implications in various diseases^{2,3}. Protein kinases present high evolutionary conservation in sequence and structure. Most of the kinase

inhibitors bind to conserved adenosine triphosphate (ATP)-binding pockets of kinases, leading to extensive target promiscuity⁴. Chemical compounds that inhibit a single kinase are still rare despite significant research efforts devoted to target-based drug discovery⁵. It is therefore crucial to map out target binding profiles of kinase inhibitors to uncover new therapeutic effects and better predict and manage possible adverse effects. Unfortunately, even with automated high-throughput profiling

¹School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA. ²Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA. ³These authors contributed equally: Yunan Luo, Yang Liu. ✉e-mail: yunan@gatech.edu; jianpeng@illinois.edu

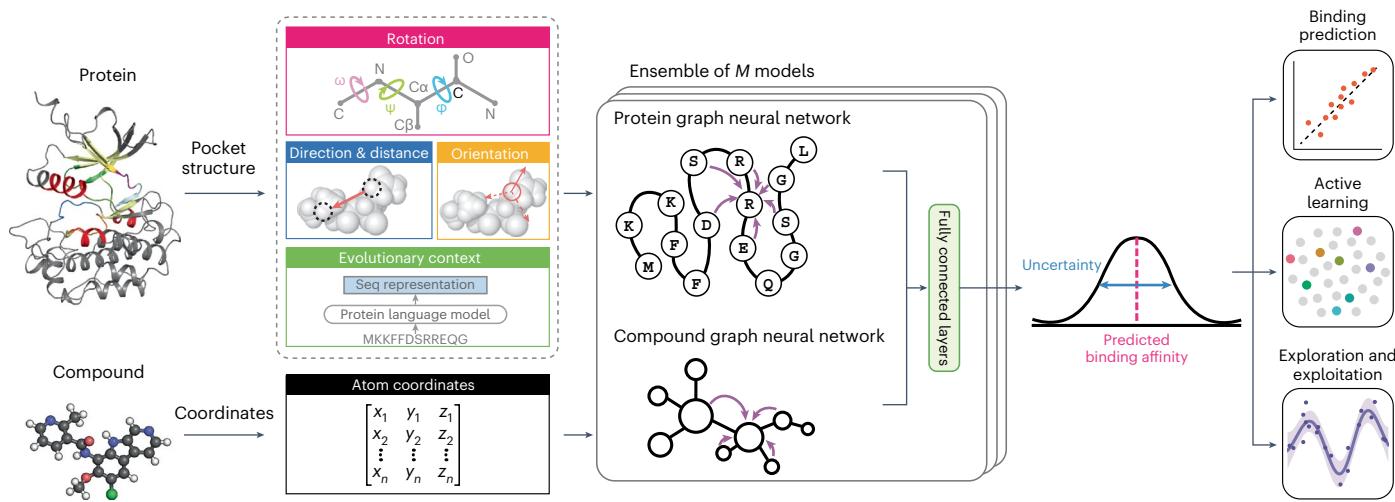


Fig. 1 | Overview of KDBNet. KDBNet is a neural network that integrates protein 3D structure and compound 3D structure to predict compound–protein binding affinity. KDBNet derives a set of features, including sequence (seq), evolutionary representations and 3D-invariant geometric features, on the basis of the input 3D structure and uses a GNN to learn structure-aware representations of a protein.

For the input compound, KDBNet uses a 3D-equivariant GNN to directly learn structure representations from the compound's coordinates in the 3D space. The representations of the input protein are then used to predict the binding affinity as well as the uncertainty of the prediction.

assays, it is still infeasible to exhaustively measure compound-target binding activities because of the vast chemical space.

Machine learning (ML) methods have emerged as alternative solutions to efficiently map compound–protein interaction profiles⁶. Early studies included bipartite graph-based methods that framed the prediction problem as a recommendation system-like task^{7–11}. These methods computed the similarity between compounds or proteins on the basis of simple features like molecule fingerprints or sequence-alignment scores, enabling the prediction of new protein–drug interactions on the basis of known, similar proteins and drugs. With recent advancements in deep learning, a series of studies^{12–16} leveraged deep neural networks to automatically learn features from raw compounds and protein representations in a fully data-driven way, also known as end-to-end learning. Commonly used data representations include one-dimensional (1D) features such as protein sequences and molecule simplified molecular-input line-entry system (SMILES) strings^{12,13}. Recent approaches indicated that incorporating two-dimensional (2D) features, including molecular graphs and protein contact maps, enhanced prediction accuracy^{14–17}. Although the compound–protein binding is, in essence, a physicochemical process in the three-dimensional (3D) space, there remains a paucity of studies that incorporate 3D structure information to enhance protein–drug binding prediction, in part because of the scarcity of protein structure data and the absence of predictive models that effectively use 3D structure data. Fortunately, for kinases, the data bottleneck is less pronounced owing to their biological significance. Kinases are one of the best-represented protein families in the Protein Data Bank (PDB) database¹⁸, with a rapidly growing number of solved kinase structures^{19,20}. In parallel, recent progress in graph deep learning offers promising avenues for effectively modelling 3D protein structures^{21–23}. Jointly, there are great opportunities and pressing needs to develop new methods that integrate 3D structure information to improve predictions of kinase–drug binding affinity.

The primary importance of ML approaches for compound–protein binding prediction is to accelerate the discovery of compounds or targets. With an accurate ML predictive model, virtual screening can be performed by applying the model to generate hypotheses about binding activities, allowing the selection of candidates with the highest predicted activities for further validation. However, these data-driven methods are susceptible to inherent noise and bias in the

training data, rendering them vulnerable to failures when applied to out-of-distribution scenarios. To mitigate this issue, one solution is to quantify the uncertainty of model predictions, providing a confidence assessment to support human decision-making, as higher novelty often comes with a higher risk of failure. Although the importance of uncertainty estimation in ML algorithms has been recognized^{24–26}, most existing methods of compound–protein binding prediction only provide point-estimate predictions without quantifying uncertainty^{12–16}. In the context of compound or target discovery, relying solely on point-estimate predictions to select top candidates for validation may result in false positives. Ref. 17 introduced uncertainty estimation using Gaussian processes (GPs) to prioritize strong-binding compound–protein pairs, but quantifying uncertainty with more expressive deep neural networks has not been explored for kinase–drug binding prediction.

Here, we develop the kinase–drug binding prediction neural network (KDBNet), a deep learning algorithm that integrates 3D structure information to predict the binding affinity of kinase–drug binding while also estimating prediction uncertainties. KDBNet represents the 3D protein and molecule structure data as graphs and uses graph neural networks (GNNs) to learn structure representations from binding pocket structures of proteins and atom coordinates of molecules. We built KDBNet as an ensemble model of several replicates of individual neural networks, which not only improves prediction accuracy and robustness but also allows us to estimate the uncertainty of model predictions. We further applied an uncertainty recalibration technique to refine the uncertainty estimates, enhancing KDBNet's utility in the ML-guided discovery of proteins and targets. Benchmarking on public datasets of kinase–drug binding-affinity measurements, we found that KDBNet achieved more accurate predictions than existing models that used only 1D or 2D representations of proteins and drugs. Our experiments also indicated that KDBNet's uncertainty estimates were largely consistent with respect to prediction errors, meaning predictions with lower uncertainty are often more accurate. Furthermore, we found the uncertainty estimates were also well-calibrated, providing interpretable confidence intervals for individual predictions. Finally, we extended KDBNet into a Bayesian optimization (BO) framework, showcasing its capability for data-efficient active learning and accelerated exploration of strong-binding kinase–drug pairs.

Results

Overview of KDBNet

KDBNet is a deep learning model that integrates 3D structures to predict binding affinities between kinases and small-molecule compounds (Fig. 1). KDBNet receives 3D structures of proteins and compounds and represents them as two graphs: the graphs' nodes are protein residues or molecule atoms, and the edges encode residue contacts or atom distance. A set of features, which collectively describe the structural, evolutionary, biophysical and chemical properties of protein residues or chemical atoms, are also derived for each node and edge in the protein and molecule graphs. Next, KDBNet uses GNNs to learn structure representations of the input kinase and compound, reflecting the spatial organization and topological neighbourhood of the 3D protein and molecule structures. The learned representations are then combined to predict the binding affinity through another fully connected (FC) neural network. In addition to the binding-affinity prediction, KDBNet also associates each of its predictions with an uncertainty estimate, quantifying its confidence about the prediction. KDBNet achieves this by training an ensemble of models and estimating the uncertainty using the variance of individual models' predictions.

Accurate prediction of kinase–drug binding affinity

We first assessed KDBNet's performance in predicting kinase–drug binding affinity using two public datasets of experimental measurements of kinase–compound binding affinity, Davis²⁷ and KIBA²⁸, which were widely used to benchmark previous methods^{10,12,15–17}. We created three evaluation settings to simulate out-of-distribution scenarios in which the training and test sets do not share any drugs or proteins (Fig. 2a and Supplementary Notes 1.1 and 1.2). We compared KDBNet with several state-of-the-art methods for predicting kinase–drug binding affinity, including three deep-learning-based methods^{12,15,16}, a GP-based method¹⁷ and a kernel-based method²⁹. These baseline methods rely solely on 1D and 2D representations or pairwise similarities of compounds and proteins, without incorporating 3D structural information (Supplementary Note 1.3).

The evaluation results (Fig. 2 and Supplementary Fig. 1) indicated that KDBNet consistently outperformed other methods across several metrics, including Pearson correlation, Spearman correlation and mean squared error (MSE; one-sided rank test $P < 10^{-3}$). These improvements held across various split settings. The enhancements achieved by KDBNet also underscored the efficacy of end-to-end feature learning in comparison to methods (for example, GP) that rely on precomputed, fixed feature embeddings. Similar observations were made when applying KDBNet to the larger KIBA dataset, where it surpassed the baseline methods and even outperformed two recently developed methods that used protein language model embeddings³⁰ or contrastive learning strategies³¹ (Extended Data Fig. 1).

The improvements of KDBNet primarily stem from its direct modelling of 3D structures of proteins and molecules in the neural network. This was confirmed by our ablation study, in which 3D structure data of either the input protein or drug were dropped (Fig. 2c). Compared to baselines that consider only 1D or 2D representations of proteins and compounds, the 3D structure data and structure-derived geometric features in KDBNet (Supplementary Fig. 2) provided more explicit information related to the binding activity, which better respects the 3D physical symmetries of binding activities that might not be fully reflected by the 1D or 2D features. Even compared to recent methods that use the 3D protein–compound binding complex structure as input (CNN3D^{32,33}, GNN3D³⁴ and SIGN³⁵) on the PDBbind database³⁶, KDBNet achieved performance comparable to these complex-based baselines (Fig. 2d and Supplementary Fig. 3) and substantially higher than baselines that used non-3D input (DeepDTA and GraphDTA; one-sided rank test $P < 10^{-3}$). It is noteworthy that complex-based methods had an advantage in this comparison, as they can capture the interaction features from the complex structure. Although slightly superior in

prediction performance, those complex-based methods^{32–35,37–41} are constrained by the availability of binding complex structures. In contrast, KDBNet achieved comparable prediction performance using separate 3D structures, which are more readily accessible, making it suitable for numerous tasks such as virtual drug screening for which complex structures between novel targets and compounds are rarely available.

Overall, these results demonstrated that by incorporating 3D structure data and leveraging geometry-aware deep learning, KDBNet made clear performance improvements in kinase–drug binding prediction compared to several existing methods and was able to generalize to predictions for unseen proteins, unseen drugs or both.

Informative and calibrated uncertainty estimation

One immediate application of an accurate ML model for protein–compound binding-affinity prediction is using it to generate new hypotheses, such as prioritizing promising compounds, to assist drug discovery and drug repurposing. From a practical perspective, in addition to predicting affinity, it is also desirable that the model can provide associated uncertainty estimates, allowing researchers to assess the likelihood of hypothesis success and allocate experimental efforts more effectively. Unlike many previous deep learning methods that only predict a point estimate of binding affinity while overlooking uncertainties in the data or model^{12,15,16}, KDBNet goes a step further by providing an uncertainty estimate for each affinity prediction (Methods).

First we aimed to investigate whether KDBNet's uncertainty estimate is indicative of prediction accuracy. Ideally, the model's uncertainty would be correlated with its prediction error, and predictions with lower uncertainty would have lower prediction errors. We assessed KDBNet's uncertainty quantification on the Davis dataset. We ranked all of KDBNet's predictions by their associated uncertainty estimates (Supplementary Note 1.4) and observed that there was a consistent trend for KDBNet's predictions with lower uncertainty to exhibit lower prediction errors across various split settings (Fig. 3a; average Spearman's correlation $\rho = 0.98$). Compared to the two GP-based methods, GP and GP-multilayer perceptron (GP-MLP)¹⁷, KDBNet achieved much lower mean absolute errors (MAEs) across different uncertainty percentiles (Fig. 3a) and higher correlations between the estimated uncertainty and prediction errors (Fig. 3b and Supplementary Fig. 4a). These indicated that KDBNet's uncertainty estimates were correctly ranked with respect to prediction errors, and its predictions were highly accurate when it had a low level of uncertainty.

The previous evaluation confirmed that KDBNet's uncertainty estimates provided indicative ranking. We now examined whether the magnitude of these uncertainty estimates was statistically meaningful. Models that are over-confident or under-confident usually produce uncertainty estimates that are either too small or large, rendering them challenging to interpret as credible intervals with statistical meaning. This issue is known as miscalibration in uncertainty quantification⁴². Ideally, we desire well-calibrated uncertainty estimation from the model, meaning, for instance, that if the model predicts a 95% confidence interval, we anticipate the true values to fall within the interval 95% of the time. We computed the miscalibration area^{26,42,43} to quantify the degree of uncertainty calibration, which is defined as the area between the model's calibration curve (Supplementary Note 1.5) and the diagonal line representing a perfectly calibrated model (Fig. 3c). A lower miscalibration area signifies superior calibration. We noted that KDBNet's calibration curves closely resembled the ideal diagonal curve (Fig. 3d), resulting in substantially lower miscalibration areas than those observed with GP-based methods (Fig. 3e and Supplementary Fig. 4b). Additionally, KDBNet's recalibration algorithm (Methods) effectively pushed the calibration curves closer to the diagonal and reduced the miscalibration area (Fig. 3d,e and Supplementary Fig. 4b; one-sided rank test $P < 10^{-3}$). These results indicated that KDBNet's uncertainty estimates were calibrated and scaled with errors.

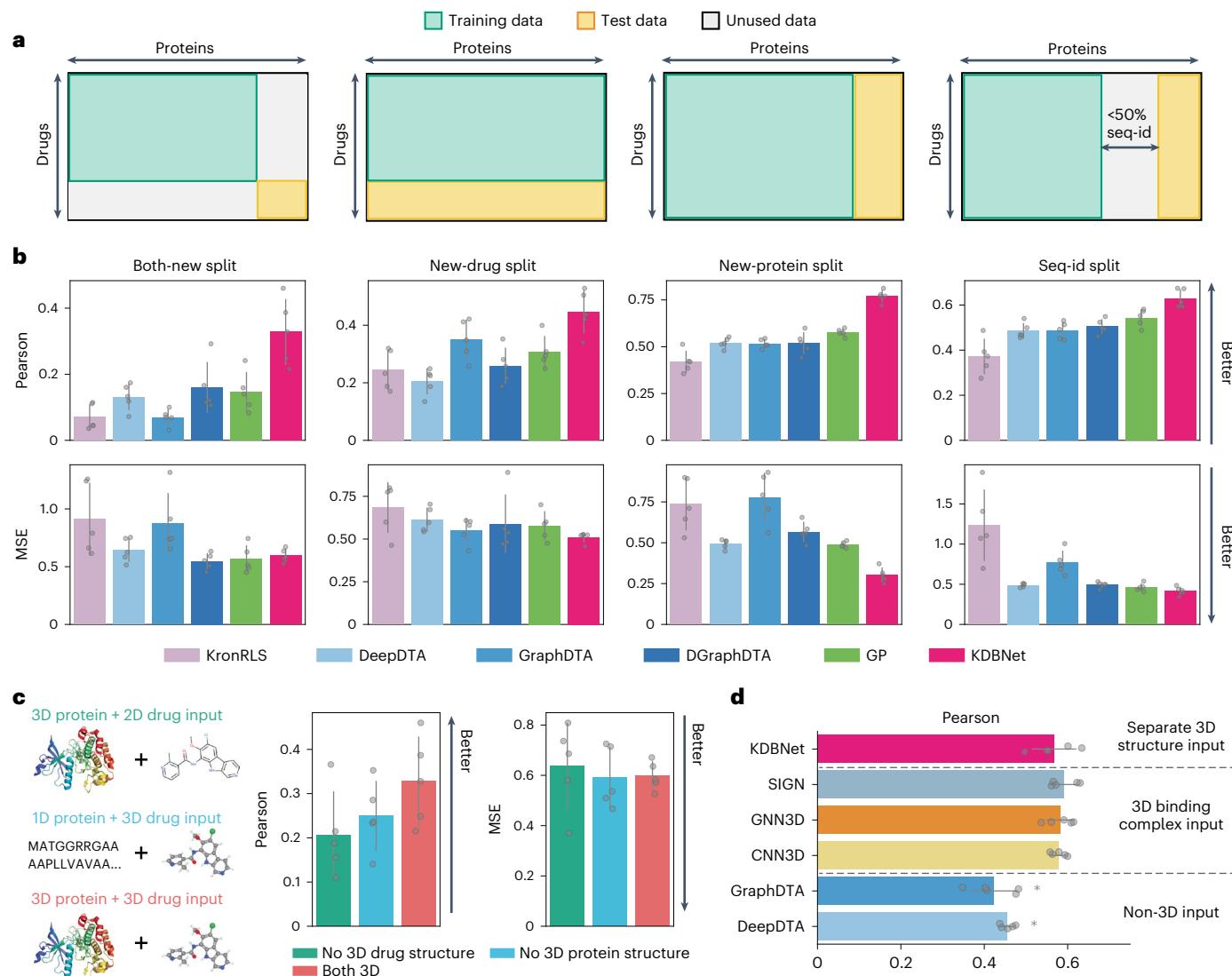


Fig. 2 | KDBNet achieves accurate prediction of kinase–drug binding affinity. **a**, Four train–test split evaluation settings in which the model is evaluated on data of unseen drugs ('new-drug split'), unseen proteins ('new-protein split') or both ('both-new split') and unseen proteins with low (<50%) sequence identity ('seq-id split'). **b**, Comparison of KDBNet prediction performance with KronRLS, DeepDTA, GraphDTA, DGraphDTA and GP on the four train–test split settings. **c**, Comparisons between KDBNet variants that use or do not use 3D structure data on the both-new split. When 3D drug structure is not used, the 2D molecule graph parsed from a SMILES string is used as the representation of the input drug, and no 3D geometric features are used in the molecule GNN. When 3D protein structure is not used, the sequence is used as the representation of the input protein, and the protein GNN is replaced by a convolutional neural network. The full model use both 3D drug and protein structures. **d**, Comparisons

between KDBNet and three baseline methods that receive 3D binding complex structure as input (GNN3D, CNN3D and SIGN) on the PDBbind dataset. KDBNet differs from them in that it only uses separate 3D drug and protein structures as input: the baseline methods thus have an advantage as they are aware of the protein–compound docking structure through the input complex. Results of methods that receive non-3D input (GraphDTA and DeepDTA) are also shown for comparison. Asterisks indicate the statistical significance (one-sided Mann–Whitney U rank test $P = 0.00397$ for both GraphDTA and DeepDTA) that KDBNet's performance is higher than the baseline's performance over $n = 5$ random train/test splits. Bar plots in **b–d** represent the mean \pm s.d. of the evaluation results on five random train/test splits. Pearson correlation and MSE were computed using the predicted and true p_d values.

Together, these two sets of experiments demonstrated that the uncertainty estimates of KDBNet were both accurate with respect to prediction errors and well-calibrated. The accurate quantification of uncertainty holds important implications for iterative ML-guided experiment design for which the uncertainty estimates can guide data acquisition and candidate prioritization, as we illustrate in the next section.

Uncertainty-guided, data-efficient active learning

Having validated that KDBNet provides informative and calibrated uncertainty estimation, we set out to assess the utility of uncertainty

in ML-guided discovery. The first application is active learning, for which the objective is to strategically select training samples to achieve improved prediction performance with fewer training data. Analogous to human experts who rely on intuitive confidence to acquire and test new samples, KDBNet used its estimated uncertainty for iterative training and selection (Fig. 4a). We initiated the training of KDBNet using a random 1% subset of the KIBA training data. In each subsequent round, KDBNet predicted binding affinities and uncertainties for the remaining training data and then ranked these samples (drug–protein pairs) on the basis of the predicted uncertainty from highest to lowest (referred to as the 'explorative' strategy). Two other ranking strategies (Methods)

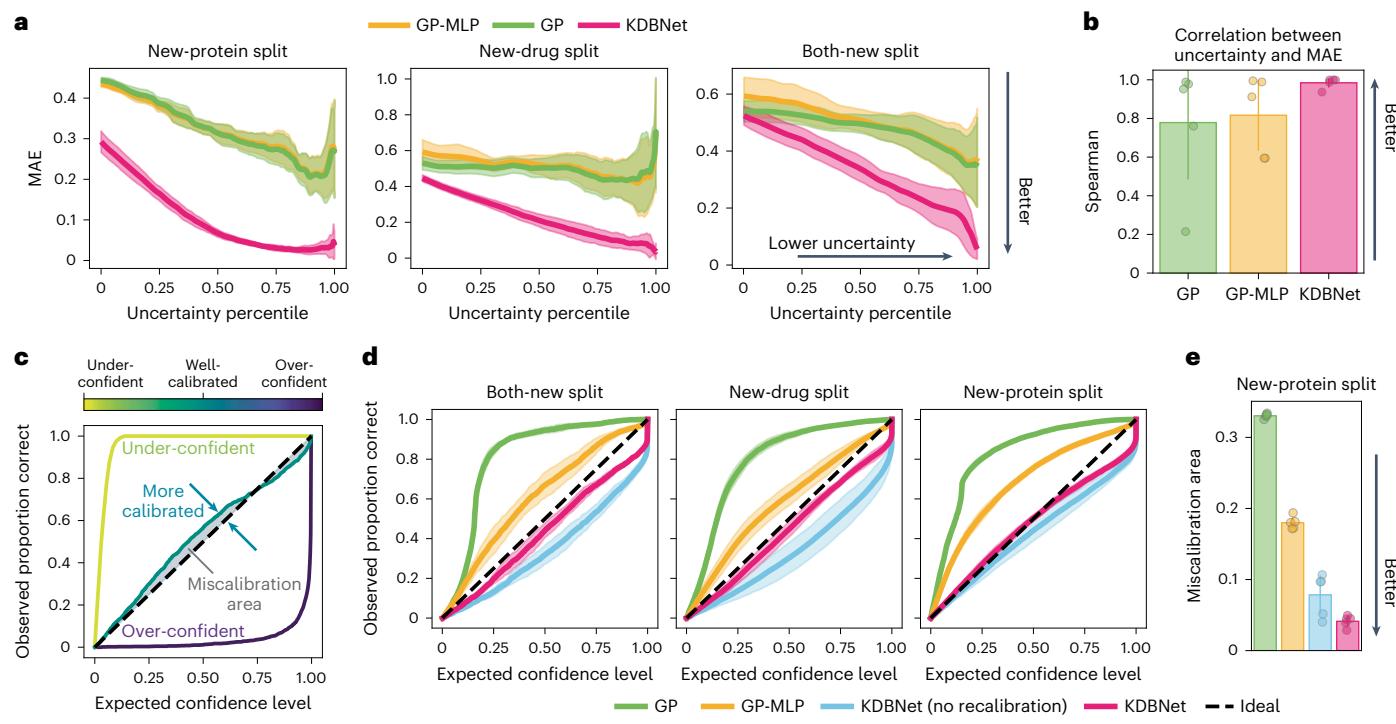


Fig. 3 | KDBNet provides accurate and calibrated uncertainty estimation.

a, Prediction errors of KDBNet, GP and GP-MLP, measured as MAE, at different cutoffs of uncertainty percentiles. The x axis represents the sorted uncertainty such that the 100% percentile is the lowest uncertainty (highest confidence). **b**, Spearman correlation between the estimated uncertainty and the prediction error measured in MAE on the both-new test set. **c**, Calibration curve. For a confidence interval of confidence level e ($0 \leq e \leq 1$), the curve shows the expected fraction and observed fraction of test points that fall within that interval. The diagonal line corresponds to the calibration curve of a perfectly calibrated model. The miscalibration area, defined as the area between a curve and the

diagonal line, is used to quantify the uncertainty calibration, and lower values indicate better calibration. **d**, Calibration curves of KDBNet, KDBNet without recalibration, GP and GP-MLP on test sets. **e**, Miscalibration area of KDBNet, GP and GP-MLP on the new-protein test set. Solid lines in curve plots represent the mean value of five independent trials, and error bands indicate the s.d. MAE values were measured in pK_d values. Bar plots in **b** and **e** represent the mean \pm s.d. of the evaluation results on five random train/test splits. Error bands in **a** and **c** depict mean \pm s.d. calculated over five random train/test splits. Recalib., recalibration.

were considered for comparison: (1) ‘greedy’, which prioritizes samples with higher predicted affinity, and (2) ‘random’, which ranks all samples uniformly at random.

We found that KDBNet achieved efficient active learning by using its estimated uncertainty to acquire new training samples, reaching performance on par with full data training by using only 50% of the data (Fig. 4b). Noticeably, KDBNet’s performance was improved by a large margin in the initial rounds compared to the random strategy, highlighting the efficiency of uncertainty-based active learning compared to brute-force random searches. Furthermore, in contrast to the greedy strategy that continually seeks samples with the highest affinities, KDBNet’s explorative strategy focused on samples that could diversify the training set and best address the model’s uncertainties, thereby exhibiting faster rates of performance improvement and higher efficiency gains (performance improvement over random selection) across all active learning stages (Fig. 4b,c). These indicated that KDBNet, enabled by uncertainty quantification, achieved sample-efficient active learning for data acquisition and model training, a valuable capability in model-guided experimental design where an exhaustive search is costly or infeasible.

Bayesian optimization for rapid exploration and exploitation

As another application of uncertainty estimation, we integrated KDBNet with BO for the exploration and exploitation of strong-binding kinase–drug pairs. Although the previous active learning experiments acquired new samples solely on the basis of uncertainty for diversifying the training set, BO provides a principled framework to

combine both predicted values and estimated uncertainties to guide data acquisition more effectively, enabling us to prioritize candidates in high-confidence, high-desirability regions (‘exploitation’) or probe potentially high-desirability regions, although with less confidence (‘exploration’), as illustrated in Fig. 4d. In BO, a common way to combine predicted scores and uncertainties is through an acquisition function called the upper confidence bound (UCB) with the form $UCB(x) = score(x) + \beta \times uncertainty(x)$, where x represents a kinase–compound pair and constant β controls the trade-off between exploitation and exploration.

High-recall exploration. We first evaluated KDBNet’s exploration capability using the Davis dataset. The objective was to identify kinase–compound pairs with the strongest binding affinity by observing the ground-truth binding affinities of only a small subset of pairs. We started the data acquisition by training KDBNet on 1% of the kinase–compound pairs (~100 pairs). Subsequent steps followed the active learning framework but incorporated UCB as the score function, defined as $UCB(x) = \mu(x) + \beta\sigma(x)$, where $\beta = 1$, and $\mu(x)$ and $\sigma(x)$ are the binding affinity and associated uncertainty predicted by KDBNet, respectively (Methods). Intuitively, this score function promotes samples with high binding affinity and high uncertainties. Because our goal was to identify strong-binding pairs as comprehensively as possible, we aimed to explore ‘good’ regions that had some variability (uncertainty), as this increased the chances of discovering even better samples. We observed that KDBNet yielded clear improvements compared to the random exploration and GP baselines, as quantified by the recall of

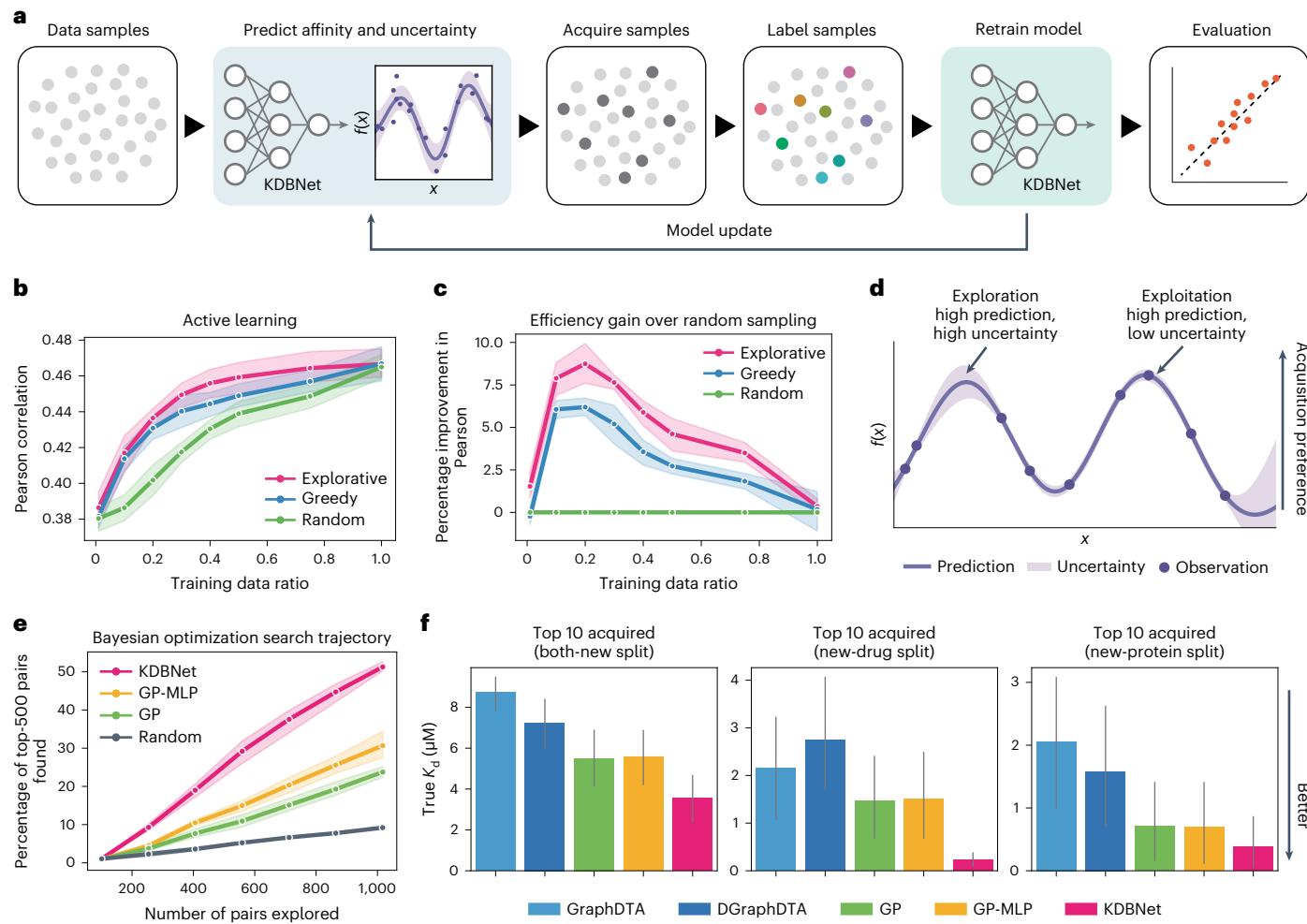


Fig. 4 | Leveraging uncertainty for active learning, exploration and exploitation. **a**, Schematic visualization of the active learning process, which consists of several rounds of model training, data acquisition and model evaluation. **b**, Active learning performance in Pearson correlation on the KIBA both-new test set at different rounds. The explorative sampling is compared to the greedy and random sampling strategies. **c**, Efficiency gain of the explorative and greedy samplings over the random sampling, defined as the relative improvement in Pearson correlation. **d**, Schematic illustration of data acquisition on the basis of KDBNet's prediction and uncertainty. One can exploit regions with high-confidence, high-desirability samples or explore potentially

high-desirability regions with less model confidence. **e**, Exploration using KDBNet and UCB acquisition function with a BO framework. Curves represent the performance trajectory, measured by the percentage of top-500 binding affinities found as a function of the number of kinase–compound pairs explored in the Davis dataset. **f**, Exploitation using KDBNet and LCB acquisition function with BO. True K_d values of the top 10 kinase–drug pairs prioritized by each model are shown. A lower K_d value means a stronger binding affinity. Curve plots in **b**, **c** and **e** depict the mean values over five independent trials in solid lines with s.d. in error bands. Bar plots in **f** represent mean \pm s.d. of the results for five independent trials of top-10 acquisition ($n=50$).

top-500 (about the top 1%) kinase–compound pairs as a function of the number of pairs explored. Specifically, KDBNet retrieved 50% of the top-500 pairs from the pool of 10,000 pairs after exploring only 1,000 pairs (Fig. 4e). This experiment highlighted KDBNet's effectiveness in accelerating the exploration and discovery of strong-binding kinase–compound pairs in the BO framework.

High-confidence exploitation. Next we performed an analysis to evaluate KDBNet's exploitation capability: that is, how effectively it prioritized top kinase–compound pairs with strong binding affinities. This mirrored real-world biological discovery processes in which researchers typically focus on only a handful of top compounds or proteins for further validation instead of testing the entire unexplored space. We simulated an experiment on the Davis dataset where the model was tasked with prioritizing kinase–compound pairs with the strongest binding affinity from the test data. For KDBNet, we defined the score function as the lower confidence bound (LCB): $\text{LCB}(x) = \mu(x) - \beta\sigma(x)$

with $\beta = 1$. Compared to UCB, LCB introduces a negation sign before the uncertainty term, prompting KDBNet to prioritize pairs with strong binding affinity and low uncertainty. Figure 4f presents the binding affinities, measured in K_d (dissociation constant) values, of the top 10 kinase–drug pairs acquired by different methods, where lower K_d values indicate stronger binding affinities. We found that, on average, KDBNet retrieved pairs with stronger binding affinity, outperforming other baseline methods across all three split settings. KDBNet successfully prioritized kinase–drug pairs with a mean K_d value lower than $0.5 \mu\text{M}$ for both the new-protein and new-drug split settings and a mean K_d of $3.5 \mu\text{M}$ for the most challenging both-new split setting. For reference, a K_d value lower than $3 \mu\text{M}$ was considered a very strong binding by the original study of this dataset²⁷. As the binding-affinity datasets often contain inherent noise or technical errors, ML models can be greatly affected and generate uncertain predictions. Consequently, top predictions from uncertainty-agnostic models often include false positives. In contrast, KDBNet's estimated uncertainty offers a further dimension of

information, complementing mean-affinity predictions and facilitating the prioritization of lead candidates with high confidence.

Discussion

We have presented KDBNet, a geometric deep learning algorithm for predicting kinase–drug binding affinity. KDBNet integrates the 3D structure data of both kinases and drugs and models them using structure-aware GNNs to predict binding affinity. Although we focused on the binding activities of kinases in this work primarily because of their important pharmacological implications^{1,44}, KDBNet’s principles can be extended to other proteins as well. Our experiments have demonstrated that KDBNet outperformed several existing deep learning methods in predicting kinase–drug binding affinity. Additionally, KDBNet provides well-calibrated uncertainties that scale with prediction errors, offering statistically indicative confidence intervals. We further showcased KDBNet’s practical utility in both active learning and BO frameworks for prioritizing kinase–drug pairs with strong binding affinities. In future work, KDBNet’s performance for binding prediction and uncertainty quantification can be enhanced by integrating recent techniques such as contrastive learning on low-coverage data³¹ (Supplementary Note 1.6), computation-efficient uncertainty-quantification algorithms²⁶ (for example, conformal predictions^{45,46}) and new calibration techniques^{42,45,47,48}.

We note that the availability of kinase and compound structures is not a critical limitation of KDBNet, as the compound structures are largely available in PubChem⁴⁹, the number of kinase structures is increasing in PDB^{20,50}, and high-quality predicted structures from AlphaFold²¹ can be used as reasonable proxies for understudied kinases with no solved structures (Supplementary Note 1.7). We also found that replacing the PDB structures input to KDBNet with AlphaFold-predicted structures resulted in comparable prediction accuracy (Supplementary Fig. 5). Another line of recent studies examined binding-affinity prediction on the basis of 3D protein–compound binding complex^{32–35,37–41}. The problem setup considered in this work, which was also used in several concurrent studies^{51,52}, is less restricted by data availability than those works, as KDBNet only requires separate structures as input rather than the binding complex.

KDBNet’s integration of uncertainty estimation is particularly valuable for biological discovery processes when data are limited and uncertainty is prevalent. ML-guided discovery can be affected by biases in the ML model stemming from data noise, small sample size and the model’s intrinsic uncertainty. KDBNet’s uncertainty quantification and recalibration serve as crucial safeguards against biased or over-confident model predictions, which are particularly useful for guiding both exploitation and exploration in virtual drug screening, allowing the prioritization of lead predictions for further validation and proposing data samples to explore previously uncharted regions or address model uncertainties. KDBNet creates an interactive cycle between computation and experiment to improve the ML model’s sample efficiency and success rate in drug screening⁵³. We anticipate that KDBNet can facilitate the reliable and robust deployment of ML-guided drug discovery and lead to the identification of promising therapeutic candidates with higher precision and efficiency.

Methods

Structure and sequence data

Most protein kinases share a common structural fold with two lobes connected by a flexible region that forms the adenosine triphosphate and substrate binding site. The activation loop in this region, typically in a length of 20–30 residues, is crucial for binding activity. We use the pocket structure, rather than the entire structure of the kinase, as the structure input of a kinase to KDBNet because (1) residues in the pocket directly interact with the drug molecule, largely determining the binding activity; and (2) structure elements outside the pocket—that is, the N- and C-terminal lobes—are relatively conserved across

different kinases and may not directly coordinate the binding as they are relatively far from the binding sites. Several structure conformations may exist for the same protein kinase in the RCSB PDB database⁵⁴ because the loop can fold into catalytically active and inactive states; we thus first selected the representative PDB structure for a kinase following a recently developed nomenclature for the active and inactive states of protein kinases^{50,55}. To extract the binding pocket, we used the KLIFS database²⁰ that defines a pocket formed by 85 residues that cover the binding sites in a wide range of kinase inhibitors by analysing around 1,200 kinase–ligand binding crystal structures (Supplementary Note 1.1). In total, we obtained the pocket structure of 283 kinases.

The sequence of amino acids (AAs) of the 85 residues in the binding pocket of a kinase was obtained from its reference protein sequence in UniProt⁵⁶. We did not use the associated sequence in the PDB file because it may contain missing or inaccurate residues. To do this, we mapped the PDB pocket sequence to the full UniProt sequence using pairwise local alignment (score matrix: BLOSUM62, gap open penalty: 10, gap extend penalty: 0.5). We successfully mapped 281 of 283 structures to UniProt sequences.

The 3D structure data in the structure-data format (SDF) of compounds in the kinase–drug binding datasets (described below) were downloaded from the PubChem database⁵⁷.

Kinase–drug binding datasets

For evaluation, we used two public datasets of experimental measurements of binding affinity, Davis²⁷ and KIBA²⁸ (Supplementary Note 1.2), which were widely used to benchmark previous methods^{10,12,15–17}. The Davis study contains binding-affinity measurements of kinase–compound pairs, represented by K_d values ranging from 0.1 to 10,000 nM, where a lower K_d value indicates stronger binding affinity. The KIBA dataset derived a score to integrate three bioactivity values: K_i (inhibitory constant), K_d and IC_{50} (half maximal inhibitory concentration). We removed from both datasets compounds and kinases that do not have 3D structures in the PDB and PubChem databases. We successfully retrieved the 3D structures for nearly all compounds and 50%–70% of proteins in Davis and KIBA (Supplementary Tables 1,2). We expect the availability of kinase structure data will keep increasing (see discussions in Supplementary Note 1.7). Raw K_d values in the Davis dataset were transformed to pK_d (binding affinity) values, defined as $pK_d = -\log_{10}(K_d/10^6)$, to facilitate numerical stability during model training^{12,41}. We created four train–test split settings (Supplementary Note 1.2) to evaluate prediction performance on data of unseen drugs (‘new-drug split’), unseen proteins (‘new-protein split’) or both (‘both-new split’) and unseen proteins with low (<50%) sequence identity (‘seq-id split’). To compare KDBNet with baseline methods that predict binding affinity from binding complex data, we additionally created another benchmark task using the PDBBind dataset³⁶. Further details about the creation of benchmark datasets are provided in Supplementary Note 1.2.

Representation of protein structure

The 3D PDB structure of a protein is given as 3D coordinates of the backbone $\mathcal{C} = \{\mathbf{c}_i \in \mathbb{R}^3\}_{i=1}^N$, where N is the number of residues, \mathbf{c}_i is the coordinate of the C_α atom of the i th residue, and \mathbb{R} is the set of real numbers. We represent the protein structure as a graph $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p)$, where nodes \mathcal{V}_p are residues and edges \mathcal{E}_p indicate residue contacts. In this work, we define a pair of residues as being in contact if the Euclidean distance between their C_α atoms is within 8 Å (ref. 58).

To make the structure graph representation more informative, we associate every node or edge in the graph with a feature vector. Intuitively, we want our node and edge features to be (1) invariant to rotation and translation so the features not depend on the placement, orientation and centring of the PDB structure inputs; and (2) informative about the local structure, as unique structural motifs

may lead to distinct binding affinities. Here, we derive a set of invariant spatial features following a previous study⁵⁹. We further extend their approach to include other features that encode sequence and evolutionary properties of residues. The constructions of node and edge features are described below.

Node features: For every residue, we build three types of features: (1) sequence feature, (2) geometric feature and (3) evolutionary feature. The sequence feature is a one-hot representation to indicate the AA type (of the total 20 possible AAs) of the residue. For geometric features, we compute the three dihedral angles (ϕ_i, ψ_i, ω_i) on the basis of the backbone coordinates of residue i . These angles are encoded as a vector of cosine and sine values: $\mathbf{v}_i = (\sin \phi_i, \sin \psi_i, \sin \omega_i, \cos \phi_i, \cos \psi_i, \cos \omega_i)$. Lastly, for evolutionary features, we ran ESM⁶⁰, a recent protein language model trained on 250 million sequences, to generate the embedding for each residue. The ESM embeddings have been shown to encode structural, functional and evolutionary properties of the protein and can improve a wide range of protein-related prediction tasks, such as function and structure prediction^{60,61}. Those three features are concatenated together as the node feature for a residue.

Edge features: To characterize the local structure surrounding residue i , we create edge features that describe the spatial relationships between residue i and its neighbours (residues j 's). In particular, we compute an orientation matrix O_i that defines the local coordinate frame for residue i ⁵⁹:

$$\begin{aligned} O_i &= [\mathbf{b}_i, \mathbf{n}_i, \mathbf{b}_i \times \mathbf{n}_i], \quad \text{where } \mathbf{u}_i = \frac{\mathbf{c}_i - \mathbf{c}_{i-1}}{\|\mathbf{c}_i - \mathbf{c}_{i-1}\|}, \\ \mathbf{b}_i &= \frac{\mathbf{u}_i - \mathbf{u}_{i+1}}{\|\mathbf{u}_i - \mathbf{u}_{i+1}\|}, \quad \mathbf{n}_i = \frac{\mathbf{u}_i \times \mathbf{u}_{i+1}}{\|\mathbf{u}_i \times \mathbf{u}_{i+1}\|}, \end{aligned} \quad (1)$$

where $\mathbf{c}_i \in \mathbb{R}^3$ is the coordinates of residue i . For an edge (i, j) , we consider an edge representation that reflects the local distance, direction, orientation and relative positions⁵⁹:

$$\mathbf{e}_{ij} = \left(\text{RBF}(\|\mathbf{c}_j - \mathbf{c}_i\|), \quad O_i^T \frac{\mathbf{c}_j - \mathbf{c}_i}{\|\mathbf{c}_j - \mathbf{c}_i\|}, \quad \mathbf{q}(O_i^T O_j), \quad E_{\text{pos}}(\mathbf{c}_j - \mathbf{c}_i) \right), \quad (2)$$

The edge feature \mathbf{e}_{ij} has four components: (1) The first part, $\text{RBF}(\|\mathbf{c}_j - \mathbf{c}_i\|)$, is the distance encoding embedded into radial basis functions (RBFs). We use 16 RBFs with centres evenly spaced between 0 and 8 Å. (2) The second term is the direction encoding that corresponds to the relative direction of \mathbf{c}_j in the local frame of residue i . (3) The third term is the orientation encoding of the quaternion representation $\mathbf{q}(\cdot)$ of the spatial rotation matrix $O_i^T O_j$. (4) The last term, $E_{\text{pos}}(\mathbf{c}_j - \mathbf{c}_i)$, encodes the relative distance and direction between residues i and j . We used the relative positional encoding⁶², an extension of the positional encoding introduced in the Transformer model⁶³. The relative positional embedding represents the vector pointing to \mathbf{c}_j from \mathbf{c}_i through a sinusoidal function. We keep the sign of the distance vector $\mathbf{c}_j - \mathbf{c}_i$ because protein sequence structures are generally asymmetric.

Representation of molecule structure

KDBNet also incorporates the 3D molecular structure of compounds to predict binding affinities. Similarly, given the 3D coordinates of atoms in the molecule, we represent the molecule structure as a graph $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ where nodes \mathcal{V}_d are atoms of the molecule and edges \mathcal{E}_d are defined for a pair of atoms if their distance is less than 4.5 Å, following ref. 34. As molecules do not have a natural backbone as in proteins, we do not derive the angle, orientation and direction features for atoms as we did in the protein graph. Instead, we directly use the 3D coordinates of atoms or edge vectors as node features and edge features, allowing the GNN in KDBNet to learn meaningful geometric representations of the molecule in a data-driven way. The node and edge features of the molecule structure are detailed below.

Node features: For every atom, we include a vector-valued feature and a scalar-valued feature as its node feature. The vector feature is the atom coordinates $\mathbf{c}_i \in \mathbb{R}^3$. The scalar feature is a list of 66 descriptors of chemical properties^{15,16,41}, including the atom type, bond degree, number of hydrogen bonds, number of implicit hydrogen bonds and whether the atom is aromatic (Supplementary Table 3).

Edge features: For an edge between atoms i and j , we also create a vector feature and a scalar feature. The vector feature is the unit vector in the direction of $\mathbf{c}_j - \mathbf{c}_i$, and the scalar feature RBF($\|\mathbf{c}_j - \mathbf{c}_i\|$) is the pairwise distance embedded into 16 Gaussian RBFs with centres evenly spaced between 0 and 4.5 Å.

KDBNet model architecture

The primary components of KDBNet are two GNNs to learn structure representations from the input protein and compound, respectively. The representations produced by the two GNNs are then passed to a FC neural network to predict the binding affinity between the input protein and compound.

Protein GNN. For the protein GNN, we use Graph Transformer⁶⁴, an effective GNN architecture adapted from the vanilla Transformer model for text data⁶³, to model the kinase structure. Given the protein structure graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathcal{E}_p\}$, a Graph Transformer model builds L graph convolution layers. The i th layer is a non-linear transformation function that maps node i 's embedding $\mathbf{h}_i^{(\ell-1)} \in \mathbb{R}^{d_{\ell-1}}$ to $\mathbf{h}_i^{(\ell)} \in \mathbb{R}^{d_\ell}$ for $i \in [N]$, $\ell \in [L]$, where d_ℓ is the embedding's dimension at layer ℓ , N is the number of nodes in \mathcal{G}_p and L is the total number of layers in the GNN. In particular, when $\ell = 0$, the embedding $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d_0}$ is just the node feature of residue i . In addition, we have edge features of each edge (i, j) denoted as $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$, where d_e in the dimension of input edge features.

Formally, in the ℓ -th Graph Transformer layer of the GNN, the hidden representation $\mathbf{h}_i^{(\ell)}$ is updated by performing a message passing between node i and its neighbours

$$\mathbf{h}_i^{(\ell)} = \mathbf{W}_1^{(\ell)} \mathbf{h}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} (\mathbf{W}_2^{(\ell)} \mathbf{h}_j^{(\ell-1)} + \mathbf{W}_3^{(\ell)} \mathbf{e}_{ij}), \quad (3)$$

where $\mathcal{N}(i)$ is the set of neighbour nodes of node i in the graph, $\mathbf{W}_1^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$, $\mathbf{W}_2^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ and $\mathbf{W}_3^{(\ell)} \in \mathbb{R}^{d_\ell \times d_e}$ are learnable parameters of the GNN, and $\alpha_{i,j}$ is the attention weight used to aggregate messages. The weights $\alpha_{i,j}$ are computed using self-attention:

$$\alpha_{i,j} = \text{softmax} \left(\left[(\mathbf{W}_4^{(\ell)} \mathbf{h}_i^{(\ell-1)})^\top (\mathbf{W}_5^{(\ell)} \mathbf{h}_j^{(\ell-1)} + \mathbf{W}_3^{(\ell)} \mathbf{e}_{ij}) \right] / \sqrt{d_\ell} \right), \quad (4)$$

where $\mathbf{W}_4^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ and $\mathbf{W}_5^{(\ell)} \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ are learnable parameters and d_ℓ is the length of vector $\mathbf{h}_i^{(\ell)}$.

We stack three Graph Transformer layers and use the Leaky ReLU activation function⁶⁵ between two adjacent layers. After the final layer, we use the global add pooling operation as the readout function to aggregate all node representations into a summary representation $\mathbf{h}^p \in \mathbb{R}^{256}$ of the input protein: $\mathbf{h}^p = \text{ADD}(\{\mathbf{h}_i^{(L)} | i = 1, \dots, N\})$.

Molecule graph neural network. Given the molecule structure graph $\mathcal{G}_d = \{\mathcal{V}_d, \mathcal{E}_d\}$, we also use a GNN to learn the representation for the input molecule. Recall that in graph \mathcal{G}_d , we associate each node and edge with both geometric vector features (for example, 3D coordinates) and scalar features (for example, descriptors of chemical properties). We thus use a specialized layer, geometric vector perceptrons (GVPs)²², to build the molecule GNN. The key advantage of GVP is that it has special consideration for 3D data in design (Supplementary Note 1.8) and allows KDBNet to learn structure representations directly from the raw atom coordinates in \mathbb{R}^3 without requiring the construction of features invariant to rotations and translation, such as relative direction embeddings. In the GNN, the GVP layer can be used as a drop-in

replacement for MLPs, such as $\mathbf{W}_1^{(\ell)}, \mathbf{W}_2^{(\ell)}, \mathbf{W}_3^{(\ell)}$ in the protein GNN (equation (3)).

Formally, we use the tuple $\mathbf{v}_i = (\mathbf{v}_i^v, \mathbf{v}_i^s)$ to denote the node feature of atom i (the superscripts v and s stand for vector and scalar, respectively), where $\mathbf{v}_i^v \in \mathbb{R}^{\mu \times 3}$ is a list of vector features in \mathbb{R}^3 and $\mathbf{v}_i^s \in \mathbb{R}^v$ is a list of scalar features (μ and v are the number of vector features and scalar features, respectively). The edge feature $\mathbf{e}_{ij} = (\mathbf{e}_{ij}^v, \mathbf{e}_{ij}^s)$ of edge (i, j) has similar meaning. The molecule GNN transforms the node and edge features through L graph convolution layers to obtain the representation of the input molecule. Specifically, in the i th layer, each node aggregates ‘messages’ (embeddings) from neighbouring nodes and edges and then updates its own representations:

$$\mathbf{h}_i^{(\ell)} = \mathbf{h}_i^{(\ell-1)} + g\left(\mathbf{h}_i^{(\ell-1)} + \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ji}^{(\ell)}\right), \quad (5)$$

where $g(\cdot)$ is a sequence of three GVP layers, $\mathcal{N}(i)$ is the set of neighbour nodes of node i in \mathcal{G}_d , $\mathbf{h}_i^{(\ell)}$ is the embedding of node i in layer ℓ (in particular, $\mathbf{h}_i^{(0)} = \mathbf{v}_i$ is the node feature), and $\mathbf{m}_{ji}^{(\ell)}$ is the ‘message’ passed from node j to node i , computed using another sequence of GVP layers: $\mathbf{m}_{ji}^{(\ell)} = g(\text{concat}(\mathbf{h}_j^{(\ell-1)}, \mathbf{e}_{ji}))$, where $\text{concat}(\cdot)$ is the concatenation operation of two embeddings. Similar to the protein GNN, after the final layer of the molecule GNN, we also apply the global add pooling operation to aggregate all node representations into a scalar representation $\mathbf{h}^d \in \mathbb{R}^{128}$ of the input drug.

In addition to the molecule GNN, the GVP layers can also be used to build the protein GNN. As introduced above, the default architecture of the protein GNN was on the basis of the Graph Transformer layers as we found in our local tests that GVP- and Transformer-based protein GNNs lead to comparable prediction accuracy (Supplementary Fig. 5), but the latter took 50% less training time and GPU memory use. We thus chose Transformer layers as the default building blocks for the protein GNN. Nevertheless, for larger training sets that cover diverse protein families rather than only kinases, we expect GVP layers to be more effective for learning structure representations, as they are able to learn many other implicit geometric features that go beyond the manually defined features used in the Transformer-based GNN.

Prediction module, hyperparameter tuning and model training. We tuned the hyperparameters of KDBNet by performing a small-scale grid search using the training data, such that seven-eighths of the training data were used to train a model with a specific set of hyperparameters, and the remaining one-eighth of the data were used as the validation set to select the hyperparameters. The test split was not used for hyperparameter selection. We tested combinations of GNN layer dimensions from $\{64, 128, 256, 512, 1,024\}$, combinations of FC layer dimensions from $\{64, 128, 256, 512, 1,024\}$, the number of FC layers in $\{1, 2, 3\}$ and the dropout rate in $\{0.1, 0.25, 0.5\}$.

By performing nested cross-validation on the training data, we decided to use three layers with sizes 128, 256 and 256 for the protein GNN and three layers with uniform size 128 for the molecule GNN, which were robust across different settings in our experiments. The two representations of protein and drug structures generated by the GNNs, \mathbf{h}^p and \mathbf{h}^d , are then projected to dimension 128 using two FC layers with sizes 1,024 and 128 and a dropout rate of 0.25. The two projected embeddings are then concatenated and passed to a two-layer FC neural network with sizes 1,024 and 512 and a dropout rate of 0.25, followed by a single scalar output as the predicted binding affinity between the input protein and drug.

The training objective of KDBNet is to minimize the MSE between the predicted binding affinity and the true affinity value. The model is trained using the Adam optimizer with a learning rate of 0.0005. We trained all models for 500 epochs.

Uncertainty quantification

We equipped KDBNet with an uncertainty-quantification module. This was achieved by training an ensemble of M independent model replicates²⁴, which has been widely demonstrated as an effective way to estimate uncertainty⁶⁶. The M model replicates had the same neural network architectures and hyperparameters, but the learnable parameters were initialized with different random seeds. We set $M = 8$ in this work unless otherwise specified. Specifically, let $\hat{y}_k(x_i)$ be a prediction given by the k th individual model, where x_i represents the input kinase-drug pair. KDBNet’s final prediction of binding affinity $\mu(x_i)$ and its estimated uncertainty $\sigma(x_i)$ are given by the mean and standard deviation (s.d.) of the individual model’s predictions:

$$\mu(x_i) = \frac{1}{M} \sum_{k=1}^M \hat{y}_k(x_i), \quad \sigma(x_i)^2 = \frac{1}{M} \sum_{k=1}^M (\hat{y}_k(x_i) - \mu(x_i))^2 \quad (6)$$

The uncertainty $\sigma(x)$ estimated by KDBNet above is known as epistemic uncertainty. In the literature, uncertainties are often categorized into aleatoric uncertainty (data uncertainty due to inherent noise in observations) and epistemic uncertainty (model uncertainty due to uncertainty in parameters or predictions; Supplementary Note 1.9). In this work, we focus on estimating epistemic uncertainty, as many recent studies have demonstrated the utility of epistemic uncertainty for discovery in various domains, including biology¹⁷, chemistry⁶⁷ and healthcare⁶⁸. Nevertheless, KDBNet can be extended to estimate aleatoric uncertainty by modifying the objective function from an MSE minimization to a maximum likelihood estimation^{24,69}.

Active learning

We started training KDBNet on a random 1% subset of KIBA training data. At each subsequent round, KDBNet predicted binding affinities and uncertainties for the remainder of the training data and then ranked them on the basis of the score function $s(x) = \sigma(x)$, where $\sigma(x)$ is the predicted uncertainty for sample x (hereinafter referred to as the ‘explorative’ strategy) and where a sample x represents an input kinase-drug pair. We then added the top T samples with the greatest uncertainties to the training set and retrained KDBNet from scratch with the expanded training set. In our experiments, we performed seven rounds of active learning. The number of samples to acquire (T) for each round was determined such that 10%, 20%, 30%, 40%, 50%, 75% and 100% of the training samples were used to retrain the model in each of the seven rounds, respectively. Two other types of score function $s(x)$ were considered for comparison: (1) ‘greedy’, where samples with higher predicted affinity receive higher scores, $s(x) = \mu(x)$; and (2) ‘random’, where samples receive random scores, $s(x) \sim \mathcal{U}(0, 1)$, that is, the continuous uniform distribution between zero and one. The performance was evaluated on the ‘both-new’ test set.

Uncertainty recalibration

There are two widely used definitions of regression calibration in the literature: confidence-interval-based calibration⁴² and error-based calibration⁴⁷. Under confidence-based calibration, a model is said to be well-calibrated if $e\%$ of its predictions fall in the $e\%$ predicted confidence interval ($0 \leq e \leq 1$)⁴², whereas error-based calibration defines a well-calibrated model as one for which the uncertainty estimate of a prediction, in expectation, equals the prediction errors⁴⁷. Several approaches have been proposed to recalibrate regression models^{42,47,48}. The general idea is to learn a post hoc transformation function, which receives the model’s predicted uncertainties as input and outputs the transformed uncertainty estimates that would be better calibrated. In our method, we use a simple yet effective scaling approach^{47,70} to recalibrate the uncertainty estimates. Specifically, we transform the model’s output $(\mu(x_i), \sigma(x_i))$ to $(\mu(x_i), r\sigma(x_i))$, where r is the scaling factor to be learned. Note that the model’s prediction of binding affinity $\mu(x_i)$ does not change. To learn the scaling factor r , we introduce an optimization

problem in which the objective is to minimize the miscalibration area (Supplementary Note 1.5). The recalibration is a post hoc process, meaning the model's predicted uncertainties are fixed and only r is optimized. As indicated previously⁴², the recalibration is performed on a held-out validation set that has not been used for model training. We use Brent's method⁷¹ implemented in the SciPy package⁷² to solve this single-variable optimization.

Data availability

The two kinase–drug binding-affinity datasets, Davis²⁷ and KIBA²⁸, were curated by and available in the Therapeutics Data Commons benchmark⁷³. The PDBbind dataset (v.2020) was downloaded from <http://www.pdbbind.org.cn/>. The PDB codes of representative structures of kinases were obtained from the Kincore database⁵⁰ (<http://dunbrack.fccc.edu/kincore/home>). The binding pocket structure of each kinase was downloaded from the KLIFS database²⁰ (<https://klifs.net/>). Full AA sequences of kinases were obtained from UniProt⁵⁶ (<https://www.uniprot.org/>). The 3D molecular structures were downloaded from PubChem⁵⁷ (<https://pubchem.ncbi.nlm.nih.gov/>). Our processed version of the binding-affinity datasets and the identifier list of kinases and drugs are available on our GitHub repository (<https://github.com/luoyunan/KDBNet>).

Code availability

The source code of KDBNet is available at <https://github.com/luoyunan/KDBNet> and has been deposited to Zenodo⁷⁴ at <https://doi.org/10.5281/zenodo.7959829>. KDBNet was developed using Python v.3.9, PyTorch v.1.16, PyTorch Geometric v.2.2, RDKit (v.2022.03.2), NumPy v.1.23.4 and SciPy v.1.9.3.

References

1. Oprea, T. I. et al. Unexplored therapeutic opportunities in the human genome. *Nat. Rev. Drug Discov.* **17**, 317–332 (2018).
2. Attwood, M. M., Fabbro, D., Sokolov, A. V., Knapp, S. & Schiöth, H. B. Trends in kinase drug discovery: targets, indications and inhibitor design. *Nat. Rev. Drug Discov.* **20**, 839–861 (2021).
3. Cohen, P., Cross, D. & Jänne, P. A. Kinase drug discovery 20 years after imatinib: progress and future directions. *Nat. Rev. Drug Discov.* **20**, 551–569 (2021).
4. Hanson, S. M. et al. What makes a kinase promiscuous for inhibitors? *Cell Chem. Biol.* **26**, 390–399 (2019).
5. Arrowsmith, C. H. et al. The promise and peril of chemical probes. *Nat. Chem. Biol.* **11**, 536–541 (2015).
6. Cichońska, A. et al. Crowdsourced mapping of unexplored target space of kinase inhibitors. *Nat. Commun.* **12**, 3307 (2021).
7. Bleakley, K. & Yamanishi, Y. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics* **25**, 2397–2403 (2009).
8. Cobanoglu, M. C., Liu, C., Hu, F., Oltvai, Z. N. & Bahar, I. Predicting drug–target interactions using probabilistic matrix factorization. *J. Chem. Inf. Model.* **53**, 3399–3409 (2013).
9. Zheng, X., Ding, H., Mamitsuka, H. & Zhu, S. Collaborative matrix factorization with multiple similarities for predicting drug–target interactions. In *Proc. 19th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (eds Ghani, R. et al.) 1025–1033 (ACM, 2013).
10. Cichonska, A. et al. Computational-experimental approach to drug–target interaction mapping: a case study on kinase inhibitors. *PLoS Comput. Biol.* **13**, e1005678 (2017).
11. Luo, Y. et al. A network integration approach for drug–target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.* **8**, 573 (2017).
12. Öztürk, H., Özgür, A. & Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics* **34**, i821–i829 (2018).
13. Karimi, M., Wu, D., Wang, Z. & Shen, Y. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* **35**, 3329–3338 (2019).
14. 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).
15. Jiang, M. et al. Drug–target affinity prediction using graph neural network and contact maps. *RSC Adv.* **10**, 20701–20712 (2020).
16. Nguyen, T. et al. Graphdta: predicting drug–target binding affinity with graph neural networks. *Bioinformatics* **37**, 1140–1147 (2021).
17. Hie, B., Bryson, B. D. & Berger, B. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Syst.* **11**, 461–477 (2020).
18. Rose, P. W. et al. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* **45**, gkw1000 (2016).
19. Van Linden, O. P., Kooistra, A. J., Leurs, R., De Esch, I. J. & De Graaf, C. KLIFS: a knowledge-based structural database to navigate kinase–ligand interaction space. *J. Med. Chem.* **57**, 249–277 (2014).
20. Kanev, G. K., de Graaf, C., Westerman, B. A., de Esch, I. J. & Kooistra, A. J. KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Res.* **49**, D562–D569 (2021).
21. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
22. Jing, B., Eismann, S., Suriana, P., Townshend, R. J. & Dror, R. Learning from protein structure with geometric vector perceptrons. Paper presented at the International Conference on Learning Representations (ICLR). (eds Oh, A., Murray, N. & Titov, I.) (2021); <https://openreview.net/forum?id=1YLJDvSx6J4>
23. Gainza, P. et al. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat. Methods* **17**, 184–192 (2020).
24. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)* Vol. 30 (eds Guyon, I. et al.) 6402–6413 (Curran Associates, Inc., 2017).
25. Zeng, H. & Gifford, D. K. Quantification of uncertainty in peptide–mhc binding prediction improves high-affinity peptide selection for therapeutic design. *Cell Syst.* **9**, 159–166 (2019).
26. Soleimany, A. P. et al. Evidential deep learning for guided molecular property prediction and discovery. *ACS Cent. Sci.* **7**, 1356–1367 (2021).
27. Davis, M. I. et al. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **29**, 1046–1051 (2011).
28. Tang, J. et al. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J. Chem. Inf. Model.* **54**, 735–743 (2014).
29. Pahikkala, T. et al. Toward more realistic drug–target interaction predictions. *Brief. Bioinform.* **16**, 325–337 (2015).
30. Goldman, S., Das, R., Yang, K. K. & Coley, C. W. Machine learning modeling of family wide enzyme–substrate specificity screens. *PLoS Comput. Biol.* **18**, e1009853 (2022).
31. Singh, R., Sledzieski, S., Bryson, B., Cowen, L. & Berger, B. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proc. Natl Acad. Sci.* **120**, e2220778120 (2023).
32. Jiménez, J., Skalic, M., Martínez-Rosell, G. & De Fabritiis, G. K deep: protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. *J. Chem. Inf. Model.* **58**, 287–296 (2018).

33. Townshend, R., Bedi, R., Suriana, P. & Dror, R. End-to-end learning on 3D protein structure for interface prediction. In *Adv. Neural. Inf. Process. Syst.* Vol 32 (eds Wallach, H. et al.) 15616–15625 (Curran Associate, Inc., 2019).

34. Townshend, R. J. et al. Atom3d: tasks on molecules in three dimensions. Preprint at <https://arXiv.org/2012.04035> (2020).

35. Li, S. et al. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proc. 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (eds Zhu, F., Ooi, B. C. & Miao, C.) 975–985 (ACM, 2021).

36. Liu, Z. et al. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* **31**, 405–412 (2015).

37. Lim, J. et al. Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation. *J. Chem. Inf. Model.* **59**, 3981–3988 (2019).

38. Zheng, L., Fan, J. & Mu, Y. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein-ligand binding affinity prediction. *ACS Omega* **4**, 15956–15965 (2019).

39. Zhou, J. et al. Distance-aware molecule graph attention network for drug-target binding affinity prediction. Preprint at <https://arXiv.org/2012.09624> (2020).

40. Hassan-Harrirou, H., Zhang, C. & Lemmin, T. Rosenet: improving binding affinity prediction by leveraging molecular mechanics energies with an ensemble of 3D convolutional neural networks. *J. Chem. Inf. Model.* **60**, 2791–2802 (2020).

41. Li, S. et al. Monn: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Syst.* **10**, 308–322 (2020).

42. Kuleshov, V., Fenner, N. & Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *Proc. International Conference on Machine Learning (PMLR)* (eds Dy, J. & Krause, A.) 2796–2804 (ACM, 2018).

43. Tran, K. et al. Methods for comparing uncertainty quantifications for material property predictions. *Mach. Learn. Sci. Technol.* **1**, 025006 (2020).

44. Ali, K. et al. Inactivation of PI3K p110 δ breaks regulatory t-cell-mediated immune tolerance to cancer. *Nature* **510**, 407–411 (2014).

45. Angelopoulos, A. N. & Bates, S. Conformal prediction: a gentle introduction. *Found. Trends Mach. Learn.* **16**, 494–591 (2023).

46. Bosc, N. et al. Large scale comparison of qsar and conformal prediction methods and their applications in drug discovery. *J. Cheminform.* **11**, 4 (2019).

47. Levi, D., Gispan, L., Giladi, N. & Fetaya, E. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors* **22**, 5540 (2023).

48. Song, H., Diethé, T., Kull, M. & Flach, P. Distribution calibration for regression. In *Proc. International Conference on Machine Learning (PMLR)* (eds Chaudhuri, K. & Salakhutdinov, R.) 5897–5906 (ACM, 2019).

49. PubChem3D release notes. PubChem <https://pubchemdocs.ncbi.nlm.nih.gov/pubchem3d> (2019).

50. Modi, V. & Dunbrack, R. Kincore: a web resource for structural classification of protein kinases and their inhibitors. *Nucleic Acids Res.* **50**, D654–D664 (2022).

51. Zhou, G. et al. Uni-mol: a universal 3d molecular representation learning framework. In *Proc. of the 11th International Conference on Learning Representations* (eds Nickel, M. et al.) (OpenReview, 2023).

52. Lu, W. et al. Tankbind: trigonometry-aware neural networks for drug-protein binding structure prediction. In *Advances in Neural Information Processing Systems* Vol 35 (eds Koyejo, S. et al.) 7236–7249 (Curran Associates, Inc., 2022).

53. Luo, Y., Peng, J. & Ma, J. Next decade's AI-based drug development features tight integration of data and computation. *Health Data Sci.* **2022**, 9816939 (2022).

54. Burley, S. K. et al. RCSB protein data bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.* **49**D437–D451 (2021).

55. Modi, V. & Dunbrack, R. L. Defining a new nomenclature for the structures of active and inactive kinases. *Proc. Natl Acad. Sci.* **116**, 6818–6827 (2019).

56. Consortium, T. U. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**, D480–D489 (2021).

57. Kim, S. et al. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **49**, D1388–D1395 (2021).

58. Liu, Y., Palmedo, P., Ye, Q., Berger, B. & Peng, J. Enhancing evolutionary couplings with deep convolutional neural networks. *Cell Syst.* **6**, 65–74 (2018).

59. Ingraham, J., Garg, V., Barzilay, R. & Jaakkola, T. Generative models for graph-based protein design. In *Proc. Advances in Neural Information Processing Systems 32* (eds Wallach, H. et al.) 15820–15831 (Curran, 2019).

60. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci.* **118**, e2016239118 (2021).

61. Luo, Y. et al. ECNet is an evolutionary context-integrated deep learning framework for protein engineering. *Nat. Commun.* **12**, 5743 (2021).

62. Shaw, P., Uszkoreit, J. & Vaswani, A. Self-attention with relative position representations. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (eds Walker, M., Ji, H. & Stent, A.) 464–468 (Association for Computational Linguistics, 2018).

63. Vaswani, A. et al. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems 30* (eds Guyon, I. et al.) 5998–6008 (Curran, 2017).

64. Shi, Y. et al. Masked label prediction: unified message passing model for semi-supervised classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)* (2021).

65. Maas, A. L., Hannun, A. Y. & Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. 30th International Conference on Machine Learning (ICML)* (eds Dasgupta, S. & McAllester, D.) 3–8 (JMLR, 2013).

66. Ashukha, A., Lyzhov, A., Molchanov, D. & Vetrov, D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. Paper presented at the 8th International Conference on Learning Representations (ICLR) (eds Song, D., Cho, K. & White, M.) (2020).

67. Eyle, N. S., Green, W. H. & Jensen, K. F. Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *React. Chem. Eng.* **5**, 1963–1972 (2020).

68. Roy, A. G. et al. Does your dermatology classifier know what it doesn't know? Detecting the long-tail of unseen conditions. *Med. Image Anal.* **75**, 102274 (2021).

69. Busk, J. et al. Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks. *Mach. Learn. Sci. Technol.* **3**, 015012 (2021).

70. Chung, Y., Char, I., Guo, H., Schneider, J. & Neiswanger, W. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification. Preprint at <https://arXiv.org/2109.10254> (2021).

71. Brent, R. P. An algorithm with guaranteed convergence for finding a zero of a function. *Comput. J.* **14**, 422–425 (1971).
72. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
73. Huang, K. et al. Therapeutics data commons: machine learning datasets and tasks for drug discovery and development. In *Proc. Neural Information Processing Systems Track on Datasets and Benchmarks* (eds Vanschoren, J. & Yeung, S.) (Conference on Neural Information Processing Systems, 2021).
74. Luo, Y. KDBNet: release v.0.1. Zenodo <https://zenodo.org/record/7959829> (2023).

Acknowledgements

Y. Luo is supported in part by the National Institute of General Medical Sciences of the National Institutes of Health under award R35GM150890, the 2022 Amazon Research Award and the Seed Grant Program from the NSF AI Institute: Molecule Maker Lab Institute (grant no. 2019897) at the University of Illinois Urbana-Champaign. This work used the Delta GPU Supercomputer at NCSA of the University of Illinois Urbana-Champaign through allocation CIS230097 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) programme, which is supported by NSF grant nos. 2138259, 2138286, 2138307, 2137603 and 2138296. The authors acknowledge the computational resources provided by Microsoft Azure through the Cloud Hub program at GaTech IDEaS and the Microsoft Accelerate Foundation Models Research (AFMR) program.

Author contributions

Y. Luo and J.P. conceived the research project. J.P. and Y. Luo supervised the project. Y. Luo developed the computational method and implemented the software. Y. Luo and Y. Liu performed the evaluation analyses. All authors analysed the results and participated in the interpretation. Y. Luo wrote the manuscript with support from all other authors.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at <https://doi.org/10.1038/s42256-023-00751-0>.

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42256-023-00751-0>.

Correspondence and requests for materials should be addressed to Yunan Luo or Jian Peng.

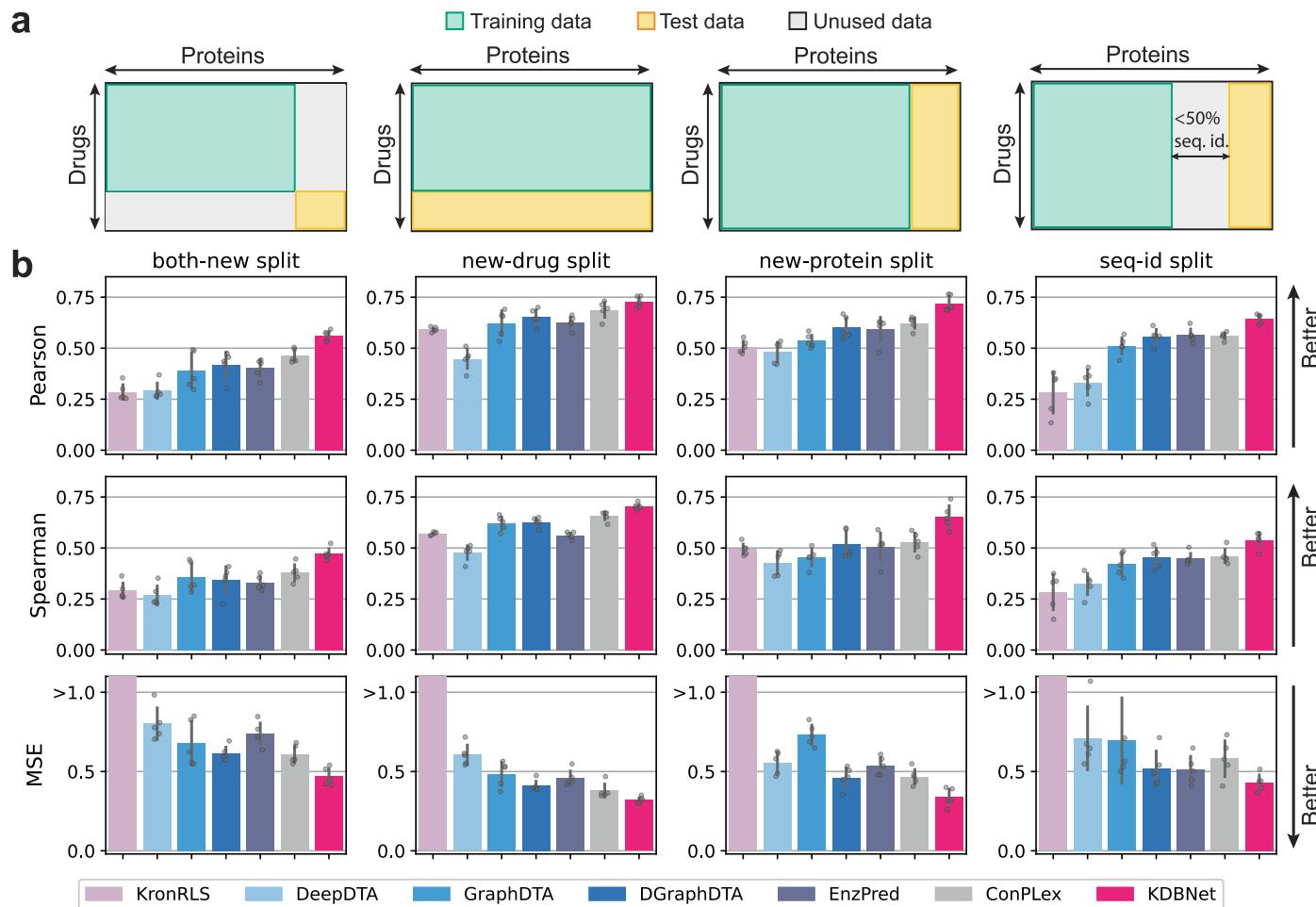
Peer review information *Nature Machine Intelligence* thanks Tero Aittokallio and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Jacob Huth, in collaboration with the *Nature Machine Intelligence* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature Limited 2023



Extended Data Fig. 1 | Prediction performance evaluation on KIBA dataset.
(a) Four train-test split settings of evaluation, where the model is evaluated on data of unseen drugs ('new-drug split'), unseen proteins ('new-protein split') or both ('both-new split'), and unseen proteins with low (<50%) sequence identity ('seq-id split'). **(b)** Comparisons of prediction performance of KDBNet with KronRLS, DeepDTA, GraphDTA, DGraphDTA, EnzPred, and ConPLex on the KIBA dataset using four train-test split settings. The performances of GP were not

shown as it was not evaluated in the original study¹⁷ and it is computationally costly to run GP at the scale of KIBA dataset because of the high memory footprint of kernel computation. Performances were evaluated using three metrics, including Pearson correlation, Spearman correlation, and mean squared error (MSE) between predicted and true KIBA scores²⁸. All bar plots represented the mean \pm SD of evaluation results on five random train/test splits. Abbreviations: seq. id.: sequence identity.