Briefings in Bioinformatics, 22(3), 2021, 1–11

doi: 10.1093/bib/bbaa183
Problem solving protocol

DeepAtomicCharge: a new graph convolutional network-based architecture for accurate prediction of atomic charges

Jike Wang, Dongsheng Cao, Cunchen Tang, Lei Xu, Qiaojun He, Bo Yang, Xi Chen, Huiyong Sun and Tingjun Hou

Corresponding authors: Tingjun Hou, Innovation Institute for Artificial Intelligence in Medicine of Zhejiang University, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou 310058, Zhejiang, P. R. China. E-mail: tingjunhou@zju.edu.cn; Huiyong Sun, Department of Medicinal Chemistry, China Pharmaceutical University, Nanjing 210009, Jiangsu, P.R. China. E-mail: huiyongsun@cpu.edu.cn; Xi Chen, Artificial Intelligence Institute, National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan 430072, Hubei, P. R. China. E-mail: robertcx@whu.edu.cn

Abstract

Atomic charges play a very important role in drug-target recognition. However, computation of atomic charges with high-level quantum mechanics (QM) calculations is very time-consuming. A number of machine learning (ML)-based atomic charge prediction methods have been proposed to speed up the calculation of high-accuracy atomic charges in recent years. However, most of them used a set of predefined molecular properties, such as molecular fingerprints, for model construction, which is knowledge-dependent and may lead to biased predictions due to the representation preference of different molecular properties used for training. To solve the problem, we present a new architecture based on graph convolutional network (GCN) and develop a high-accuracy atomic charge prediction model named *DeepAtomicCharge*. The

Jike Wang is currently a PhD student in the School of Computer Science, Wuhan University, China. His research interests lie in the area of artificial intelligence (AI), including developing new algorithms and applications for drug design.
Dongsheng Cao received his PhD degree in 2013 from Central South University, China. He is currently a professor in the Xiangya School of Pharmaceutical Sciences, Central South University, China. His research interests include (i) artificial intelligent systems for drug discovery and disease diagnosis, (ii) the development of software, web service and database in systems biology and drug discovery and (iii) design and discovery of small molecule inhibitors of important protein targets.
Cunchen Tang received his PhD degree from Wuhan University, China. He is currently a professor in the School of Computer Science, Wuhan University, China. His research interests include artificial intelligence and digital media.
Lei Xu received his PhD degree in 2013 from Soochow University, China, and now he is an associate professor in the Institute of Bioinformatics and Medical Engineering, Jiangsu University of Technology, Changzhou, China. His research interests lie on the methodology development and application of computer aided drug design (CADD) and design and discovery of novel drug candidates for important targets.
Qiaojun He received his PhD degree in 2005 from Zhejiang University, China. He is currently a professor in the Department of Pharmacology, School of Pharmaceutical Science, Zhejiang University. His research focuses on oncology pharmacology and drug toxicology.
Bo Yang received her PhD degree in 1998 from Shanghai Institute of Materia & Medica, Chinese Academy of Sciences, China. She is currently a professor in the Department of Pharmacology, School of Pharmaceutical Science, Zhejiang University. Her research interests lie on identifying and verifying the potential novel drug targets based on the molecular understanding of the process of the chronic non-communicate diseases, especially for cancer.
Xi Chen received his PhD degree in 2007 from Wuhan University, China. He is currently a professor in the School of Computer Science, Wuhan University, China. His research interests include (i) artificial intelligence for cognitive science, (ii) the development of software, web service and database in interdisciplinary areas and (iii) image information processing and applications.
Huiyong Sun received his PhD degree in 2015 from Soochow University, China. He is currently an associate professor in the Department of Medicinal Chemistry, China Pharmaceutical University, China. His research interests include (i) free energy calculation based drug discovery and (ii) artificial intelligence based drug design.
Tingjun Hou received his PhD degree in 2002 from Peking University, China. He is currently a professor in the College of Pharmaceutical Sciences, Zhejiang University, China. His research interests include (i) development of structure-based virtual screening methodologies, (ii) prediction of ADMET and drug-likeness and (iii) design and discovery of small molecular inhibitors of important protein targets. More information can be found at the website of his group: http://cadd.zju.edu.cn.

Submitted: 18 June 2020; **Received (in revised form):** 6 July 2020

new GCN architecture is designed with only the atomic properties and the connection information between the atoms in molecules and can dynamically learn and convert molecules into appropriate atomic features without any prior knowledge of the molecules. Using the designed GCN architecture, substantial improvement is achieved for the prediction accuracy of atomic charges. The average root-mean-square error (RMSE) of *DeepAtomicCharge* is 0.0121 e, which is obviously more accurate than that (0.0180 e) reported by the previous benchmark study on the same two external test sets. Moreover, the new GCN architecture needs much lower storage space compared with other methods, and the predicted DDEC atomic charges can be efficiently used in large-scale structure-based drug design, thus opening a new avenue for high-performance atomic charge prediction and application.

Key words:atomic charge; deep learning; graph convolutional network; structure-based virtual screening

Introduction

Atomic charge is one of the most important properties in computational chemistry. It helps to describe the electrostatic state of atoms in different molecular environment [1]. Although atomic charge is a seemingly simple property, the precise estimation of atomic charges typically requires high-level quantum mechanics (QM) calculations since two atoms that belong to the same element and are connected to the same atoms may involve in different chemical environments that cannot be well characterized by simple rules based on atom types and bond types. However, high-level QM-based atomic charge calculation, such as the fitting of restrained electrostatic potential (RESP) [2] charges, is very time-consuming and not suitable to be employed to large chemical datasets for virtual screening. To speed up the calculation of atomic charges, several compromised methods have been developed, such as the semi-empirical method of AM1-BCC charges [3] and the empirical method of Gasteiger–Marsili charges [4], which can significantly reduce computational cost, whereas decrease prediction accuracy as well [5]. Therefore, how to balance the accuracy and efficiency of atomic charge prediction is a long-standing issue in computational chemistry.

In recent years, machine learning (ML) or artificial intelligence (AI) methods have emerged as powerful tools for solving various complicated issues in medicine, chemistry and biology sciences, such as structure-based virtual screening toward specific drug targets [6], medical image classification [7], compound retrosynthesis [8], etc. There is no exception when it comes to the prediction of atomic charges. Up to date, a number of pioneering studies have been conducted to build ML models to fit the high-level QM atomic charges with the purpose of accelerating the calculation speed while not losing accuracy too much. In 2013, Rai and Bakken proposed a ML method to predict the atomic charges for H, C, O, N, F, S and Cl derived by fitting the b3lyp/6-31G$*$ electrostatic potentials (ESP) [9]. Based on a set of 3D descriptors that consist of 126 artificially defined symmetry function elements (77 radial and 49 angular), random forest (RF) [10] regression was used to train the model on a dataset of ∼80 000 molecules randomly selected from the Pfizer library and ZINC database. The prediction on the test set with 5000 molecules reached a mean unsigned error (MUE) of 0.03 e. In 2018, Bleiziffer *et al.* [11] developed another atomic charge prediction model using RF regression with the aim to reproduce the DDEC charges for C, H, N, O, S, P, F, Cl, Br and I [12–14]. The training set consists of 130 247 drug-like molecules, and the test set includes two parts: the one containing 146 molecules taken from Caleman's work [15] and the other containing 1385 drug-like molecules from the ZINC database [16]. The molecules were parameterized by a set of fixed-length descriptors calculated by RDKit [17], and the model achieved extraordinarily accurate predictions on the two test datasets (RMSEs = 0.029 e and 0.016 e, respectively). In the same year, Sifain *et al.* used the hierarchically interacting particle neural network (HIP-NN) algorithm to develop a new charge assignment model to reproduce the molecular dipole moments across a large diverse dataset containing C, H, N and O atoms, which yielded good predictions on the test dataset [18]. Moreover, recently Martin and Heider [19] proposed a RF-based online tool called *ContraDRG* for fast predicting *PRODRG* or *Automated Topology Builder* (ATB) partial charges for small molecules. The model uses 3D or 2D feature encodings as the input and shows high accuracy for the external test set.

Although significant improvement has been achieved for atomic charge estimation, traditional ML-based models usually employ artificially defined 2D/3D descriptors, which may yield biased predictions due to the preference of different types of molecular representations. Recently, graph neural network (GNN), a deep learning (DL) algorithm, was used to learn atom representations for the prediction of molecular properties. The basic chemical information encoded by molecular graphs is used as the input for GNN, and then the specific molecular feature of each atom toward different prediction tasks is learned by aggregating the information from its neighboring atoms and the connected bonds through message passing across the molecular graph recursively. That means that the GNN-based representation of a molecule is task-dependent and can avoid artificial intervention in model building. A number of GNN-based variants have been proposed for different tasks. For example, Kearnes *et al.* developed a graph convolutional network (GCN) called Weave [20], and Google proposed a GNN framework named Message Passing Neural Network (MPNN) [21]. Both approaches were used to predict molecular properties and offered remarkable improvements compared with traditional methods. Recently, Zhao *et al.* proposed an attention-based GCN to generate molecular representations for drug discovery [22], in which they took atoms and chemical bonds as nodes and edges, respectively, to encode each molecule into an undirected graph. On this basis, they predicted some pharmacological properties of small molecules, such as molecular toxicity, solubility, etc., and the method gained a general improvement compared with traditional molecular descriptor-based approaches. However, most of the existing models work only at the molecular level, such as predicting various global properties of molecules. Thus, whether the GCN-based approaches can work well at the atomic level is of great interest. Here, we extended the use of GCN to predict atomic charges of molecules and proposed a new GCN architecture, *DeepAtomicCharge*, which exhibits better prediction accuracy and higher computational efficiency compared with traditional ML-based methods. More importantly, the generated DDEC charges can be efficiently used in large-scale virtual screening, thus providing an alternative way for structure-based drug discovery.

Materials and methods

Data collection

In this study, we used the benchmark datasets reported by Bleiziffer to train and validate the models [11]. A total of 130 267 molecules (90 248 from the ZINC [16] lead-like database and 40 019 from the ChEMBL [23] database) with two sets of the DDEC charges based on different dielectric constants ($\varepsilon = 4$ for modeling partial charges in protein and $\varepsilon = 78$ for modeling partial charges in solvent) were incorporated in the dataset, in which the DDEC charges computed at high-level QM calculations were set as the target values for the model training. Each dataset was divided into two parts: the test set contains 3000 molecules randomly selected from the dataset (2000 from the ZINC dataset and 1000 from the ChEMBL dataset), and the remaining 127 267 molecules were used as the training set. The distributions of the numbers of all atoms (C, H, N, O, S, P, F, Cl, Br and I) and the numbers of the heavy atoms (C, N, O, S, P, F, Cl, Br and I) in the dataset are shown in Figure S1. The numbers of different elements in the dataset are shown in Figure S2.

Graph neural network

GNN was first proposed by Scarselli *et al.* [24] in 2009 for processing the non-European-structured data and graph-structured data. Since many graph-based data, such as social relationships, recommendation system, etc., needs for understanding in the real life, GNN has achieved rapid development in recent years. For instance, Kipf *et al.* proposed a convolution method named GCN to solve the semi-supervised classification problem based on graph-structured data [25]. Later, other new GCNs have been developed, such as *Graph Sample and Aggregate* (GraphSAGE) [26], which was developed to reduce the computational cost for traversing subgraphs for model training. Although numerous new types of network models have been developed, many of them follow a same propagation mechanism, namely, the MPNN architecture developed by Google [21]. Generally, an MPNN consists of a message function M_t and a vertex update function U_t as shown below:

The message function:

$$m^{t+1} = \sum_{w \in N(v)} M_t \left(h_v^t, h_w^t, e_{vw}\right) \quad (1)$$

The vertex update function:

$$h_v^{t+1} = U_t \left(h_v^t, m_v^{t+1}\right) \quad (2)$$

where v is the target node and $N_{(v)}$ is the neighborhood set connecting to v. h_v^t and h_w^t are the t layers of the hidden state of v and its neighbor w, respectively. e_{vw} represents the edge information between node v and its neighbor w. In the message passing phase, through message function M_t, the information of the neighbors of the target node is aggregated to obtain the neighbor message vector m^{t+1}. After that, in the status update phase, the neighbor message vector m^{t+1} and the state vector h_v^t of the target node v are combined through the vertex update function and update to get the hidden state vector h_v^{t+1} for the next stage of node v.

Table 1. Features and descriptions of atoms and bonds

Atom feature	Description	Size
Atom type	H, C, N, O, S, F, Cl, Br, I, P and ‘others’ (one-hot)	11
Degree	Number of covalent bonds (one-hot)	6
Hydrogen	Number of connected hydrogens (one-hot)	5
Hybridization	sp, sp2, sp3, sp3d and sp3d2 (one-hot)	5
Valence	Implicit valence (one-hot)	6
Aromaticity	Whether the atom is in an aromatic system (one-hot)	2
Formal charge	Atomic formal charge	1
Radical electrons	Number of radical electrons	1
Bond feature	Description	Size
Bond type	Single, double, triple and aromatic (one-hot)	4
Conjugation	Whether the bond is conjugated (one-hot)	1
Ring	Whether the bond is ring (one-hot)	1

Inspired by the MPNN architecture, here, we proposed a new propagation framework for GCN. The encoding of the molecules is represented in the following part.

Atomic featurization

Before putting the graph-structured data into the network, we need to characterize the features of nodes and edges (atoms and bonds) to describe atoms. Based on the chemical nature of the atoms in molecules, we defined eight features for an atom (including atom type, degree, hydrogen, hybridization, valence, aromaticity, formal charge and radical electrons) and three chemical bond features (including bond type, conjugation and ring). All the annotations of the atom and bond features can be found in Table 1. All the features are converted into the one-hot representation [27], where only one bit is set to 1 with all the others in 0, except the formal charge and radical electron, which use integer type. Afterward, all the atom and bond features are merged together as the final features of atoms to input into the network.

Since different molecules have different number of atoms with different target values (a hard situation for operation), we came up with a solution to make a fixed-length molecule for the embedding process. That is, before putting the molecular features into the network, we reshaped each molecule by adding a certain number of fake atoms. By doing this, all the molecules have the same number of atoms. To make the fake atoms uninformative, we set all the values of the feature vectors of the fake atoms to zero and also disconnect them from any other atoms. Of course, all the prediction results of the fake atoms will be removed when the target value (i.e., RMSE) is calculated.

Atomic representation network architecture

In this part, we introduced a new GNN architecture, called *AtomicRepresentation*, to extract the structural features of the atoms in the molecule for atom representation. Following the basic propagation update rule of the MPNN architecture, we designed the network structure according to the characteristics of atoms. With the principle of simplicity and efficiency, the architecture of *AtomicRepresentation* is simply designed but very efficient in the following tests. The handling processes are as follows:

(1) First, we calculated the atom features and bond features for each atom by using RDKit [23]. Taking atoms as nodes, the features of all atoms are input into *AtomicRepresentation*. Then a number of fake atoms with the features of zeros are added to each molecule to keep the molecules in the same length (consisting of the same number of atoms). As can be seen in Table 1, each atom (including the fake atoms) is represented by a 37 bit vector. Thus, each molecule can be finally represented by a matrix of $n \times 37$ (n is the number of atoms in a molecule). The n depends on the maximum number of atoms in the molecules in the dataset (here n is set to 65).

(2) With the proper representation of atom features, the next step is to sample and transfer the information of neighbor nodes of the central atoms. This step includes the sampling and aggregation operations of the neighbor nodes. In most graph-based data, the number of nodes in the subgraph increases exponentially, which may lead to a very high computational complexity. To solve this problem, numerous algorithms, such as *GraphSAGE* [26], set a hyper-parameter to control the number of samples. However, in this work, the connections of atoms in each molecule are not too complicated. Thus, we can sample all the neighbor nodes in the molecule. In the sampling process, we first get h_w^t (the state vector of the neighbor node of the central atom in this layer) and e_{vw} (the edge or chemical bond feature vector between this neighbor node and the central atom) and then concatenate the node state h_w^t and e_{vw} to get a new vector. After obtaining the information of all the neighbor nodes, we aggregate them by simply summing them up to keep all the message vectors with the same length. The formula shows as follows:

$$Agg(v) = \sigma\left(\sum_{w\in N(v)} W\left(h_w \middle\| e_{vw}\right) + b\right) \tag{3}$$

where W and b are the learnable parameters for aggregation operations, respectively, and σ is the nonlinear activation function. Therefore, the message function can be expressed by the following formula:

$$m_v^{t-1} = Agg^{t-1}(v) \tag{4}$$

(3) After obtaining the message information through step 2, the last step is to iteratively update the message for each layer, where we aggregate the state vector of the central atom and its neighbor message information by using the concatenate operation as the following formula:

$$h_v^t = \sigma\left(W^{t-1}\left[h_v^{t-1}\middle\| m_v^{t-1}\right]\right) \tag{5}$$

DeepAtomicCharge

The architecture of the *DeepAtomicCharge* model is shown in Figure 1. Here, we use benzoic acid amide molecule as an example to describe the calculation process. The molecule (with the SMILES/sdf/mol2 format) is input into the model with all H atoms added. The first step is to extract the features of the molecule and add the fake atoms. In the second step, the initialization state vector h_v^0 of the target atom v and its neighbor nodes are obtained and input into layer 0 of *AtomicRepresentation*. The message vector of the neighbor nodes m_v^0 can be obtained through the message function in the *AtomicRepresentation* layer by aggregating all the state vectors of the neighbor nodes. After concatenating the initial state vectors h_v^0 and m_v^0, the concatenated vector will be input into a fully connected layer to get the state vector h_v^1. The state vector h_v^1 has two usages: one is to be used as the input for the next *AtomicRepresentation* layer, where all the subsequent *AtomicRepresentation* layers can be calculated (the final one is h_v^t); the other is to be used in the final aggregation stage, where the state vectors of all intermediate layers and the final state vector are input into a fully connected layer and aggregate again. This aggregation function used here is the same as that used in the *AtomicRepresentation* layer. The final step is to put the obtained aggregated vector into a fully connected layer and output the predicted atomic charges.

Model training and hyper-parameter optimization

The *DeepAtomicCharge* model was constructed with PyTorch [28] and PyTorch Geometric [29] framework, and the gradient descent optimization in the Adam [30] optimizer was employed. Here, MSELoss (measure mean-squared error) was used as the loss function for the charge prediction task. The early stop [31] strategy was used in the training phase to prevent overfitting and reduce the training time.

In addition to the hyper-parameter searching, random search [32] was used to find the best combination since too many hyper-parameters need to be determined. Here, 10-fold cross validation was used for the model construction. Finally, three main model hyper-parameters were determined, in which the best *AtomicRepresentation* layers is 6, the best output atom embedding number is 128 and the best learning rate is 0.005.

Atomic charge correction

Since the model predicts the atomic charges for individual atoms, there is no rule to ensure that the sum of the predicted atomic charges in the same molecule is strictly equal to the formal charge of the molecule. However, in practical applications, such as molecular docking, it requires that the sum of the atomic charges in the entire molecule is an integral number, namely, the formal charge of the molecule. Thus, here we proposed a simple correction strategy to correct the predicted atomic charges. First, we summed the absolute value of the predicted atomic charges in the molecule Q_{abs}^{pre}.

$$Q_{abs}^{pre} = \sum_{i=1}^{N_{atoms}} |q_i| \tag{6}$$

Then, we calculated the charge difference between the sum of the predicted values and the formal chargeΔQ.

$$\Delta Q = \sum_{i=1}^{N_{atoms}} q_i - Q^{formal} \tag{7}$$

Finally, we calculated the corrected charge q_i^{corr} with the following:

$$q_i^{corr} = q_i - \frac{|q_i|\,(\Delta Q)}{Q_{abs}^{pre}} \tag{8}$$

As discussed below, the correlation/deviation between the DDEC charges and the corrected atomic charges will not change a lot since the total correction ΔQ for each molecule is very tiny.

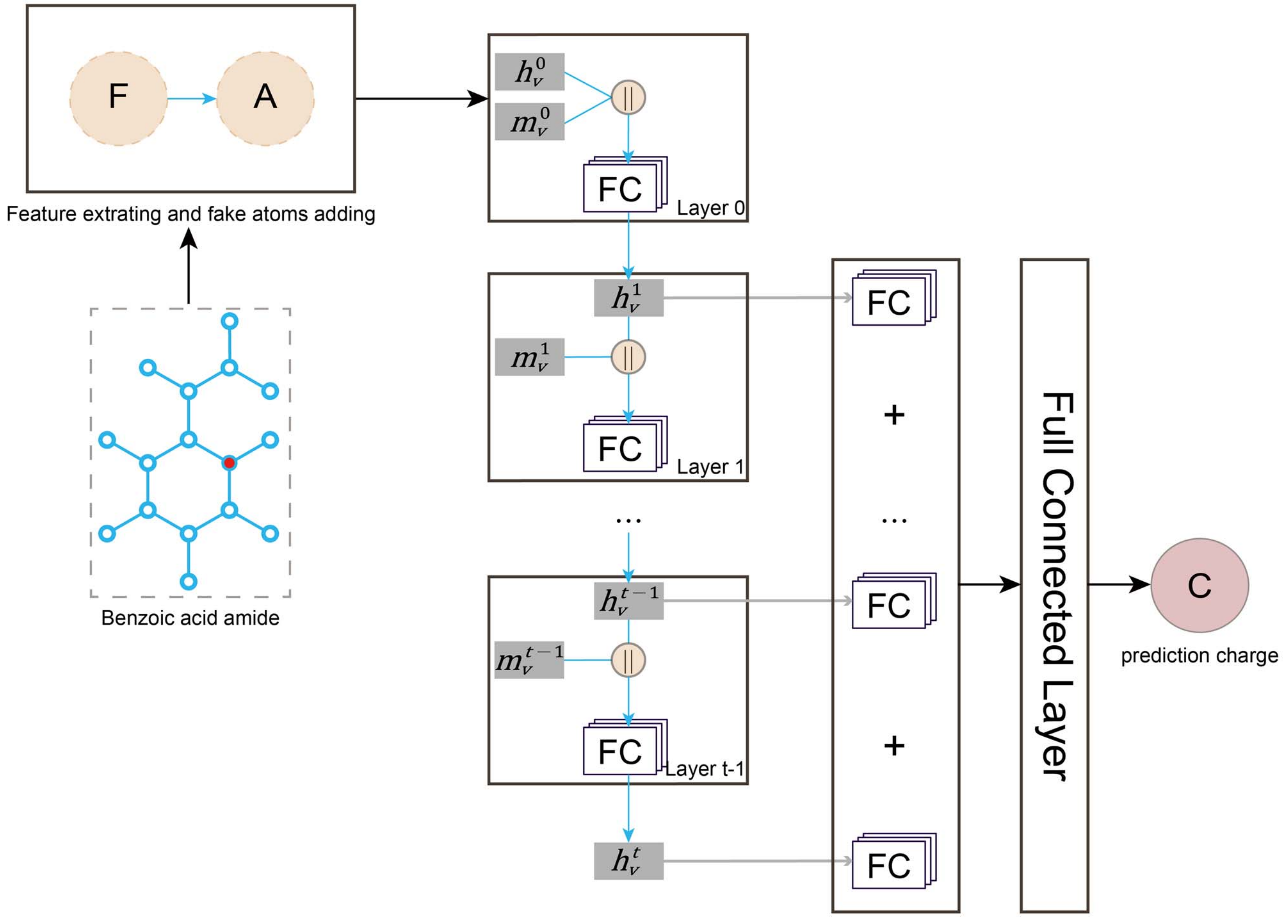

Figure 1. Architecture of the *DeepAtomicCharge* model. The F module is the feature extraction process, and the A module is the process of filling up the fake atoms. || and + are the concatenate operator and the bitwise addition operator, respectively. Layer 0 to layer t-1 are AtomicRepresentation layers. FC is the fully connected layer. C is the atomic charge predicted by the model.

Evaluation metrics

In order to quantitatively evaluate the accuracy between *DeepAtomicCharge* and the benchmark model reported by Bleiziffer *et al*., mean-absolute-error (MAE), root-mean-square error (RMSE), coefficient of determination (R^2) and cumulative distribution curve were used as the evaluation metrics.

Application in structure-based virtual screening

Since *DeepAtomicCharge* is quite computationally efficient, it can be used to calculate the partial charges of a large number of molecules in chemical libraries for virtual screening. Here, the performance of the DDEC charges predicted by *DeepAtomicCharge* on docking-based virtual screening was assessed, and androgen receptor (AR), a target that has been extensively investigated by our group [33, 34], was employed as an example system.

A total of 311 actives (agonists/antagonists) targeting AR with $K_i < 10$ μM were collected from BindingDB [35]. To mimic the unbalanced nature of inactives versus actives toward a specific target, we chose to set the ratio of actives–inactives to 1:100. Then, a total of 31 100 decoy molecules with similar distribution of molecular weight (MW) to the actives (Figure S3) were randomly selected from the ChemDiv library. All the ligands were prepared with the *LigPrep* module in Schrödinger using pH = 7.0 and saved as mol2 format for further molecular docking.

The crystal structure of AR (PDB code: 1Z95 [36]) was used as the initial structure for molecular docking, where the docking position was set at the ligand binding pocket (LBP) with the box size of 20 × 20 × 20 Å centered on the centroid of the ligand. The protein was prepared with the standard procedure of the *Protein Preparation Wizard* in Schrödinger, which includes adding the missing hydrogen atoms and repairing the imperfect crystallized side chains of the protein residues. The protonation states of the protein were determined by PROPKA (version 3.1) [37]. The *Glide* [38] module with the standard precision (SP) scoring mode in Schrödinger was employed as the docking engine since it is the most widely used docking approach in drug design campaign [39–42]. To give a comparison, all the compounds were docked with two types of partial charges, namely, the default OPLS3e charge and the DDEC charge.

The screening power proposed by Wang [43], enrichment factor and the area under curve (AUC) value of receiver operating characteristic (ROC) curve were employed as the evaluation metrics for docking-based virtual screening.

Results and discussion

It is well known that different molecular descriptors are suitable for different tasks. However, how to select the optimal descriptors for a specific task is quite challenging, and in a large extent,

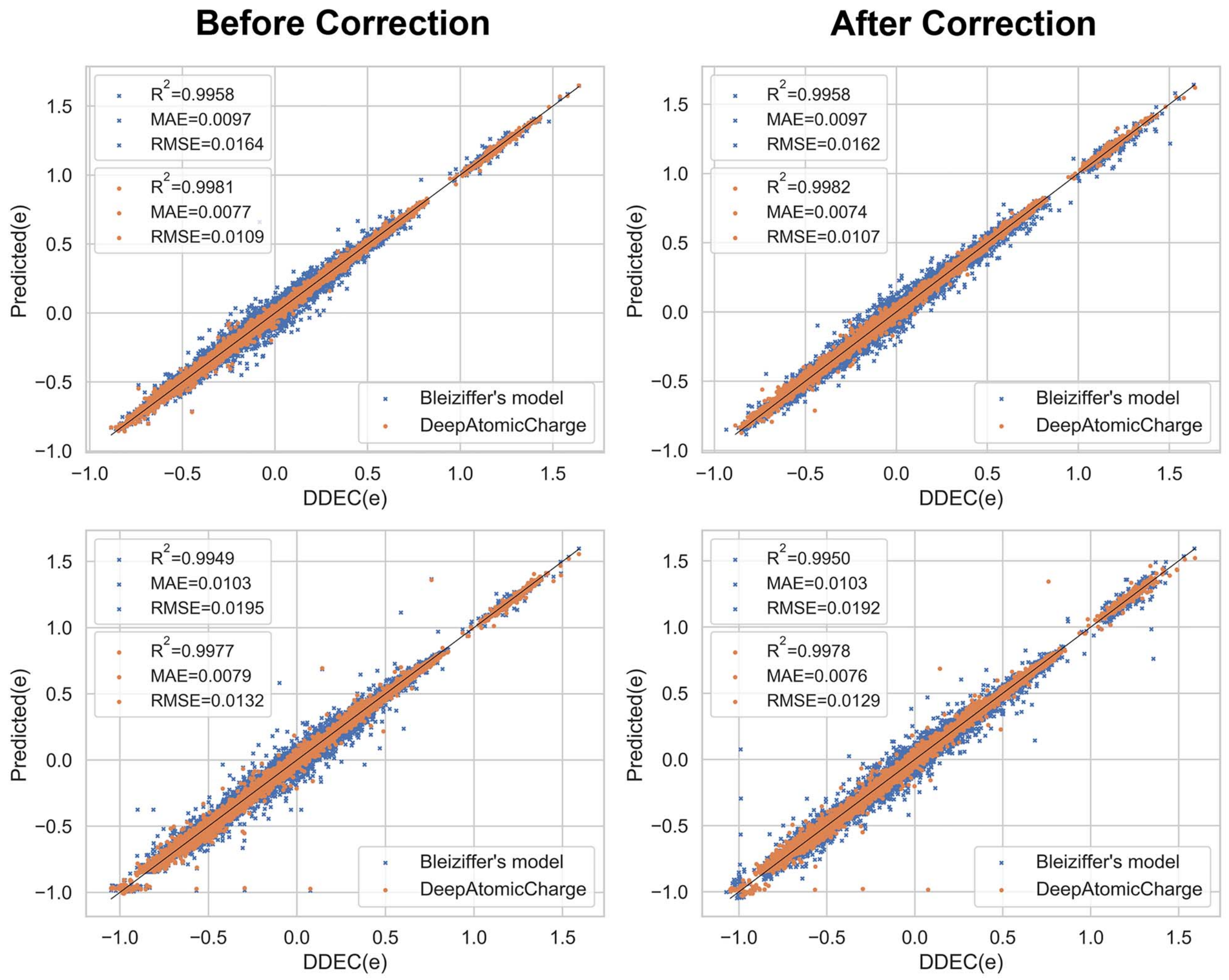

Figure 2. Performance of the *DeepAtomicCharge* and Bleiziffer's models with the 10-fold cross validation on the two training sets. (**A**) Distribution of the predicted atomic charges versus the DDEC atomic charges on the training dataset with $\varepsilon = 4$. (**B**) Distribution of the predicted atomic charges versus the DDEC atomic charges on the training dataset with $\varepsilon = 78$. (**C**) Cumulative distribution curve of the predicted atomic charges of the two training sets.

the selection of molecular descriptors is knowledge-dependent. The emergence of GCN helps to solve this problem by automatically learning the appropriate features for each specific task. *DeepAtomicCharge* is a GCN-based model that can automatically extract features from molecular structures to describe atomic charges according to the chemical environments of atoms. In order to verify the effectiveness of *DeepAtomicCharge*, we used the two datasets derived from Bleiziffer's study for the model building and testing, respectively, which contain the same molecules with different atomic charges calculated in different solute environments (ε = 4 and 78).

In order to give a fair comparison between the *DeepAtomicCharge* model and Bleiziffer's model, we used the scripts provided by Bleiziffer *et al.* to construct the RF-based model with their best parameters on the same dataset. It should be noted that, unlike Bleiziffer's work, where they trained 10 models for different types of atoms, in the framework of *DeepAtomicCharge*, all types of atoms were trained together to build a single model that is applicable to all types of atoms. Therefore, to make the results comparable, we integrated all the 10 models in Bleiziffer's work into a single model as well (hereinafter referred to as Bleiziffer's model), where a classifier was added to classify different atom types to input into their respective models. Then, in this way, the model reported by this study and that reported by Bleiziffer *et al.* can be compared directly.

Performance of 10-fold cross validation on training datasets

In this part, we performed the 10-fold cross validation on the two datasets (ε = 4 and 78). Figure 2A shows the distribution of the predicted atomic changes versus the DDEC charges on the training set based on ε = 4. As shown in Figure 2A, the MAE and RMSE of the *DeepAtomicCharge* model are approximately 23.71 and 36.90% lower than those of Bleiziffer's model (0.0074 e versus 0.0097 e and 0.0106 e versus 0.0168 e for MAE and RMSE, respectively). Similar case can be observed for the training set with ε = 78 (Figure 2B), where the results of *DeepAtomicCharge* are more convergent than Bleiziffer's model with the MAE and RMSE approximately 20.39 and 32.64% lower than the corresponding results of Bleiziffer's model (0.0082 e versus 0.0103 e and 0.0130 e versus 0.0193 e for MAE and RMSE, respectively).

In order to quantify the results better, Figure 2C and Table 2 illustrate the cumulative distribution curves and the

Table 2. Ratio of the samples with the absolute error within 0.01, 0.02, 0.03, 0.05 and 0.1 e in the 10-fold cross validation of the two training sets (ε = 4 and 78) between the *DeepAtomicCharge* model and Bleiziffer's model

ε = 4	0.01 e	0.02 e	0.03 e	0.05 e	0.10 e
Bleiziffer's model	69.71%	88.24%	94.25%	98.12%	99.71%
DeepAtomicCharge	72.46%	93.21%	98.05%	99.72%	99.98%
ε = 78	0.01 e	0.02 e	0.03 e	0.05 e	0.10 e
Bleiziffer's model	69.87%	86.58%	92.81%	97.36%	99.55%
DeepAtomicCharge	72.78%	91.50%	96.80%	99.29%	99.93%

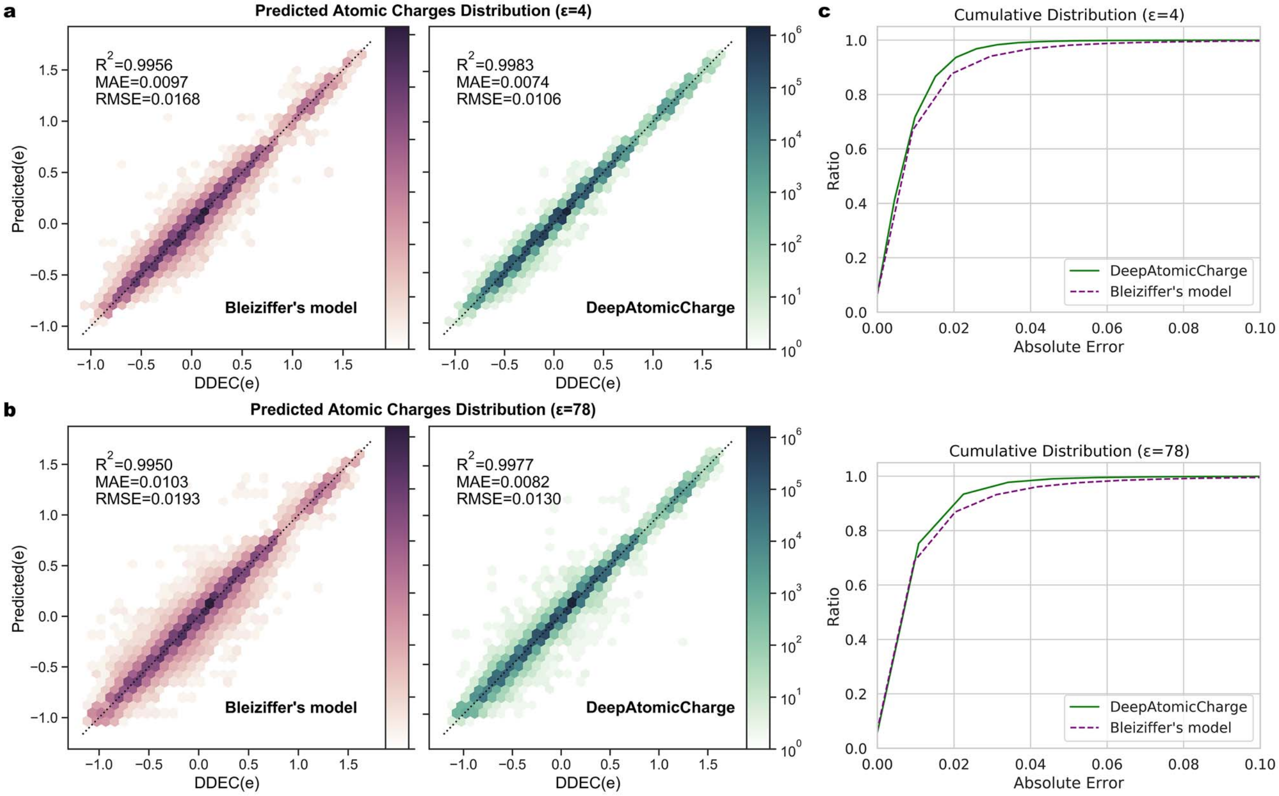

Figure 3. Distribution of the predicted atomic charges versus the DDEC atomic charges of *DeepAtomicCharge* and Bleiziffer's model on the test datasets with ε = 4 (top) and 78 (bottom) before (left) and after (right) correction.

corresponding values of the predicted results for the two training sets. As shown in Figure 2C, the green solid lines are both above the corresponding purple dotted lines at any time, suggesting that the graph-based *DeepAtomicCharge* model always outperforms the traditional descriptor-based ML model. Table 2 shows more detailed difference in each fraction of the data for the two methods, where no matter in which dataset, the absolute error of the *DeepAtomicCharge* model is within 0.03 e for >95% data. This is a very low prediction error for atomic charge prediction.

Performance on test datasets

The left column of Figure 3 shows the distribution of the predicted atomic charges versus the DDEC atomic charges before correction on the two test datasets. The orange and blue dotted lines represent the prediction results of the *DeepAtomicCharge* model and Bleiziffer's model, respectively. For ε = 4 (the top panel in first column of Figure 3), the MAE and RMSE of the *DeepAtomicCharge* model are 0.0077 e and 0.0109 e, respectively, which are 20.62 and 33.54% lower than those of Bleiziffer's model (0.0097 e and 0.0164 e for MAE and RMSE, respectively). Similar results are shown in the bottom panel (ε = 78) of the left column in Figure 3, in which the performance of the *DeepAtomicCharge* model is also much better than that of Bleiziffer's model with the MAE and RMSE decreased by 22.30 and 32.31%, respectively, compared with those of Bleiziffer's work (0.0079 e versus 0.0103 e for MAE and 0.0132 e versus 0.0195 e for RMSE). Since the two test sets (ε = 4 and 78) were randomly selected from the original datasets, the predicted results for the two test sets are a bit different. Nevertheless, it will not affect the conclusion that the graph-based GCN model (*DeepAtomicCharge*) substantially outperforms the descriptor-based ML model (Bleiziffer's model).

Correction of the predicted charges on the test datasets

Since the sum of the predicted atomic charges of a molecule may not be an integer (formal charge), to make the predicted atomic charges applicable for practical applications (such as MD simulation and docking-based virtual screening), we corrected

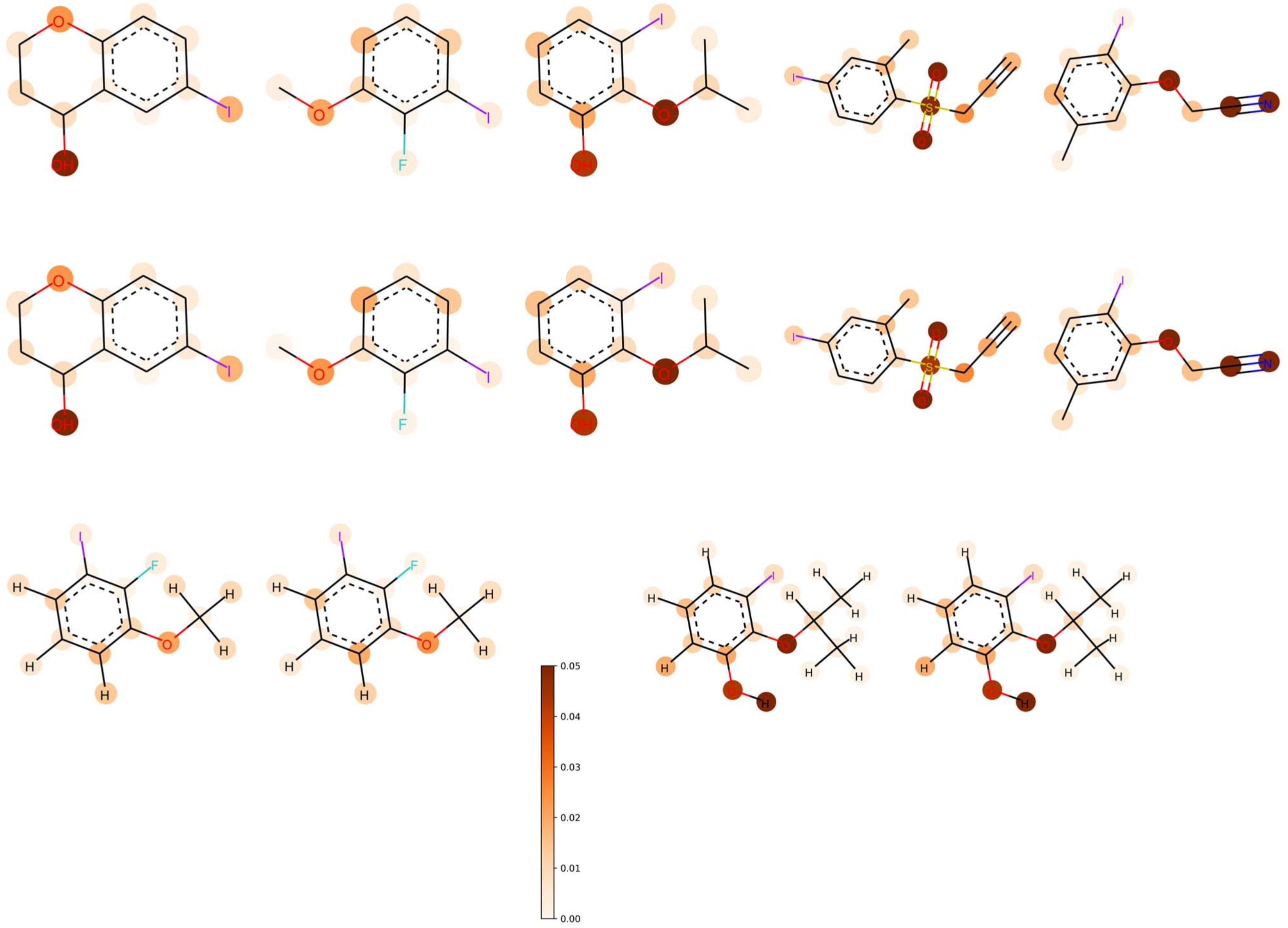

Figure 4. Example of heat maps of absolute error between the atomic charges predicted by *DeepAtomicCharge* and the DDEC calculations before and after correction. (a) Molecules showing without H element on training dataset with ε = 4. (b) Molecule showing includes H element on training dataset with ε = 4.

the predicted data according to Equation (8) for *DeepAtomicCharge* and also corrected Bleiziffer's results with their strategy. Figure 3 shows the distribution of the atomic charges predicted by the *DeepAtomicCharge* or Bleiziffer's model versus the DDEC atomic charges on the two test sets (ε = 4 and 78) before and after correction. It can be observed that the prediction accuracies become even slightly higher after correction for both test sets, while the gap between them remains the same, in which the MAEs of *DeepAtomicCharge* increase 23.71 and 26.21% and the RMSEs increase 33.95 and 32.81% for the two datasets (ε = 4 and 78), respectively, compared with those of Bleiziffer's model.

To show the effect of correction on the atomic charges predicted by *DeepAtomicCharge* clearly, the molecules containing iodine atoms in the test set (ε = 4) were employed as an example for analyzing. There are three reasons for this: (1) the number of I-containing molecules in the dataset is the lowest (43 in the test set), which is more convenient for analyzing; (2) these molecules usually contain <30 atoms and are more clear for visualization; and (3) besides containing large numbers of C and H atoms, these molecules also contain several heteroatoms (such as I and F). Thus we can try to understand how these heteroatoms affect the prediction results.

To make the result more clear, the 43 I-containing molecules were clustered into three categories using *AgglomerativeClustering()* in scikit-learn [44] library (version 0.21.2), and five molecules close to the cluster center were used for display. Figure 4 shows the absolute error between the predicted atomic charges and the DDEC atomic charges before and after correction for the five I-containing molecules, in which Figure 4A illustrates the heavy atoms only, while Figure 4B shows all atoms for an example molecule. It can be seen from the figure that the absolute errors of the predicted charges of all atoms only change slightly after correction, implying that the correction operation has tiny effect on the predicted atomic charges of *DeepAtomicCharge* and thus the predicted results are suitable for various molecular modeling applications.

Application of DeepAtomicCharge in structure-based virtual screening

Because the calculation of DDEC charges is too time-consuming, there is no any practical application of this type of charge in large-scale structure-based drug discovery. To assess the applicability of DDEC charges in virtual screening, here we assessed the screening power of molecular docking based on the DDEC charges generated by *DeepAtomicCharge* and the default OPLS3e charges assigned by Schrödinger for the system of AR. As shown in Figure 5, the DDEC charge (panel b) exhibits stronger screening power in discriminating actives from inactives for AR compared with the widely used OPLS3e charge (panel a) based on the *Glide SP* docking mode, where the active/inactive peaks were separated more obviously of the DDEC charge (with the *P*-value

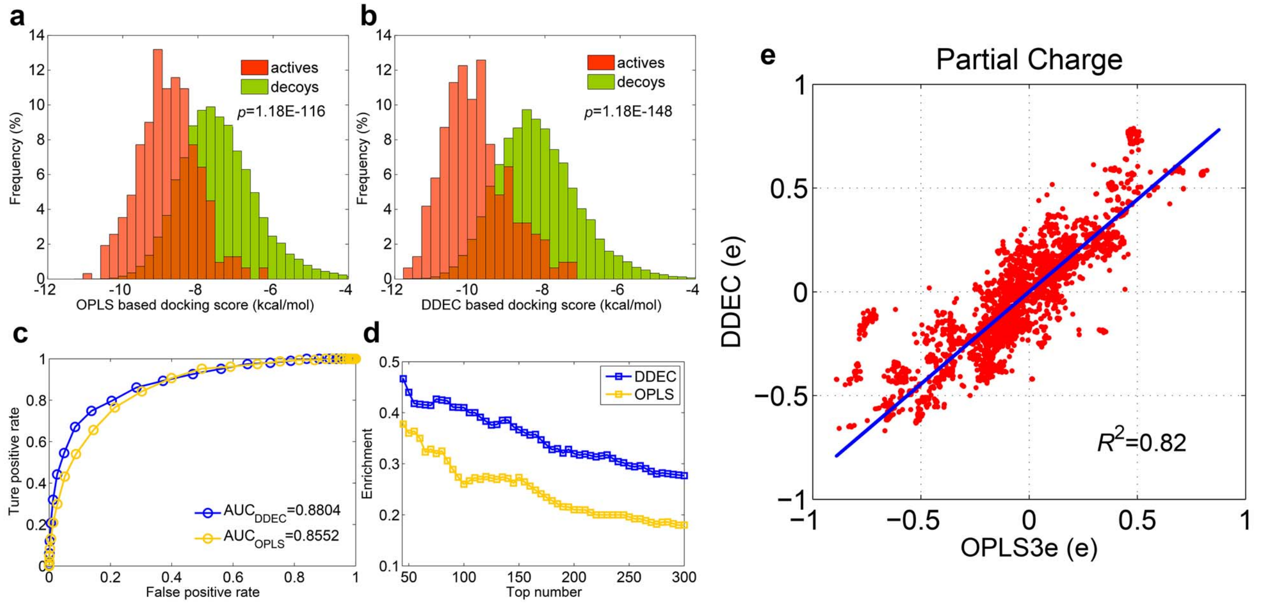

Figure 5. Performance of DDEC and OPLS3e charges on virtual screening. The screening power (**A** and **B**), AUC value under ROC curve (**C**) and the enrichment factor (**D**) are used as the metrics for comparison. The known actives and decoys are colored with red and green, respectively, for OPLS3e charge (**A**) and DDEC charge (**B**). The ROC curves and enrichment factor curves are colored blue and orange for the DDEC and OPLS3e charges, respectively. The correlation between the DDEC and OPLS3e charges for each atom of the actives is plotted in **E**.

given by Student's *t*-test of 1.18×10^{-148} versus 1.18×10^{-116} for the DDEC charges and OPLS3e charges, respectively). Moreover, both the AUC value under ROC curve and the enrichment factor for the DDEC charges are significantly higher (blue lines in panel c and d) than those for the OPLS3e charges (orange lines in panel c and d), indicating that the DDEC charge is a good choice for structure-based drug design. A comparison between the DDEC charges and the OPLS3e charges for the actives of AR illustrates that, although high correlation is shown of the two types of charges ($R^2 = 0.82$), large difference may exist between each atom pair (MAE > 0.07e, Figure 5E), implying that the two types of charges are different in nature and may be suitable for different types of systems which need further investigation. Furthermore, it should be noted that the *DeepAtomicCharge* method is very computationally efficient. The calculation of the DDEC charges for the 311 actives and 31 100 inactives only needs around 32 minutes on a laptop (CPU, Intel Core i7-8750H and GPU, NVIDIA GeForce GTX 1060). That is to say, the *DeepAtomicCharge* method can be used to process very large chemical libraries in a very short time.

Conclusion

In this work, we proposed an atomic representation layer based on GCN and an atomic charge prediction model called *DeepAtomicCharge*. Compared with existing GCN frameworks and charge prediction models, *DeepAtomicCharge* exhibits the following advantages:

(1) Compared with existing GCNs, the *AtomicRepresentation* layer and *DeepAtomicCharge* model are designed for atomic level tasks rather than molecular level ones, which guarantee the better capability of the model to reveal the relationships between atoms.

(2) Compared with the reported charge prediction models, *DeepAtomicCharge*, as an end-to-end model, can dynamically learn the topology features between atoms without the need to predefine any atom descriptors. This makes the algorithm more flexible and avoids introducing artificial influence compared with traditional descriptor-based approaches.

(3) The MAE and RMSE of the predicted atomic charges given by *DeepAtomicCharge* decrease more than 20% and 30%, respectively, compared with those reported by Bleiziffer's benchmark study. Moreover, the average storage size of the trained model of *DeepAtomicCharge* (2.62 MB) is approximately 300 times smaller than that of Bleiziffer's method (771.79 MB) (Table S1). The three advantages make the algorithm more useful and easier to be embedded into websites or python packages.

Furthermore, the DDEC charges generated by *DeepAtomicCharge* also exhibit high accuracy in structure-based virtual screening, giving an alternative way for large-scale structure-based drug design.

In conclusion, all the experimental results support that *DeepAtomicCharge* is a more flexible, convenient and accurate atomic charge prediction model and can be applied in actual structure-based drug discovery.

Key Points

- A new GCN-based architecture was developed and applied for large-scale atomic charge prediction.
- The new algorithm exhibits significant improvement for atomic charge prediction (with higher accuracy, higher computational efficiency and less storage space) compared with traditional methods.
- The high-level DDEC atomic charges predicted by *DeepAtomicCharge* were applied to the large-scale structure-based virtual screening and achieved better performance than the OPLS3e charges.

Supplementary data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Funding

This work was supported by Key R&D Program of Zhejiang Province (2020C03010), National Natural Science Foundation of China (81773632), Zhejiang Provincial Natural Science Foundation of China (LZ19H300001) and the Young Elite Scientists Sponsorship Program by CPU (NO. 131810011).

Conflict of interest

There are no conflicts to declare.

References

1. Tian L, Chen F. Comparison of computational methods for atomic charges. *Acta Physico-Chimica Sinica* 2012;**28**(1):1–18.
2. Bayly CI, Cieplak P, Cornell W, *et al.* A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model. *J Phys Chem* 1993;**97**:10269–80.
3. Jakalian A, Jack DB, Bayly CI. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J Comput Chem* 2002;**23**:1623–41.
4. Gasteiger J, Marsili M. Iterative partial equalization of orbital electronegativity—a rapid access to atomic charges. *Tetrahedron* 1980;**36**:3219–28.
5. Xu L, Sun H, Li Y, *et al.* Assessing the performance of MM/PBSA and MM/GBSA methods. 3. The impact of force fields and ligand charge models. *J Phys Chem B* 2013;**117**: 8408–21.
6. Zhavoronkov A, Ivanenkov YA, Aliper A, *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotech* 2019;**37**:1038–40.
7. Courtiol P, Maussion C, Moarii M, *et al.* Deep learning-based classification of mesothelioma improves prediction of patient outcome. *Nat Med* 2019;**25**:1519–25.
8. Lin K, Youjun X, Pei J, *et al.* Automatic retrosynthetic route planning using template-free models. *Chem Sci* 2020. doi: 10.1039/c1039sc03666k.
9. Rai BK, Bakken GA. Fast and accurate generation of ab initio quality atomic charges using nonparametric statistical regression. *J Comput Chem* 2013;**34**:1661–71.
10. Breiman L. Random forests. *Machine Learning* 2001;**45**:5–32.
11. Bleiziffer P, Schaller K, Riniker S. Machine learning of partial charges derived from high-quality quantum-mechanical calculations. *Journal of Chemical Information & Modeling* 2018;**58**:acs.jcim.7b00663.
12. Manz TA, Limas NG. Introducing DDEC6 atomic population analysis: part 1. Charge partitioning theory and methodology. *RSC Adv* 2016;**6**:47771–801.
13. Manz TA, Sholl DS. Improved atoms-in-molecule charge partitioning functional for simultaneously reproducing the electrostatic potential and chemical states in periodic and nonperiodic materials. *J Chemical Theory & Computation* 2012;**8**:2844.
14. Manz TA, Sholl DS. Chemically meaningful atomic charges that reproduce the electrostatic potential in periodic and nonperiodic materials. *J Chemical Theory & Computation* 2010;**6**:2455.
15. Caleman C, van Maaren PJ, Hong M, *et al.* Force field benchmark of organic liquids: density, enthalpy of vaporization, heat capacities, surface tension, isothermal compressibility, volumetric expansion coefficient, and dielectric constant. *J Chemical Theory & Computation* 2012;**8**:61–74.
16. Sterling T, Irwin JJ. ZINC 15 - ligand discovery for everyone. *Journal of Chemical Information & Modeling* 2015;**55**: 2324–37.
17. RDKit: Open-source cheminformatics. http://www.rdkit.org (accessed July 24, 2019) (24 July 2019, date last accessed).
18. Sifain AE, Lubbers N, Nebgen BT, *et al.* Discovering a transferable charge assignment model using machine learning. *The Journal of Physical Chemistry Letters* 2018;**9**:4495–501.
19. Martin R, Heider D. ContraDRG: automatic partial charge prediction by machine learning. *Front Genet* 2019;**10**(990).
20. Kearnes S, McCloskey K, Berndl M, *et al.* Molecular graph convolutions: moving beyond fingerprints. *J Comput Aided Mol Des* 2016;**30**:595–608.
21. Gilmer J, Schoenholz SS, Riley PF, *et al.* Neural message passing for quantum chemistry. arXiv, 2017, 1704.01212. .
22. Xiong Z, Wang D, Liu X, *et al.* Pushing the boundaries of molecular representation for drug discovery with graph attention mechanism. *J Med Chem* 2019. doi: 10.1021/acs.jmedchem.9b00959.
23. Anna G, Bellis LJ, A Patricia B, *et al.* ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* 2012;**40**:1100–7.
24. Scarselli F, Gori M, Tsoi AC, *et al.* The graph neural network model. *IEEE Trans Neural Netw* 2009;**20**:61–80.
25. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks arXiv e-prints. arXiv, 2016, 1609.02907.
26. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. arXiv, 2017, 1706.02216.
27. Harris S, Harris D. *Digital design and computer architecture*. Chian Machine Press, 2012.
28. El-Kabbani O, Green NC, Lin G, *et al.* Structures of human and porcine aldehyde reductase: an enzyme implicated in diabetic complications. *Acta Crystallogr D Biol Crystallogr* 1994;**50**:859–68.
29. Fey M, Lenssen JE. *Fast Graph Representation Learning with PyTorch Geometric*. arXiv, 2019, 1903.02428.
30. Kingma DP, Ba J. *Adam: A method for stochastic optimization* International Conference on Learning Representations. 2014
31. Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning. *Constructive Approximation* 2007; **26**:289–315.
32. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 2012; **13**:281–305.
33. Tang Q, Fu W, Zhang M, *et al.* Novel androgen receptor antagonist identified by structure-based virtual screening, structural optimization, and biological evaluation. *Eur J Med Chem* 2020;**192**:112156.
34. Zhou W, Duan M, Fu W, *et al.* Discovery of novel androgen receptor ligands by structure-based virtual screening and bioassays. *Genom Proteom Bioinf* 2018;**16**(6):416–427.
35. Liu T, Lin Y, Wen X, *et al.* BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res* 2007;**35**: D198–201.
36. Bohl CE, Gao W, Miller DD, *et al.* Structural basis for antagonism and resistance of bicalutamide in prostate cancer. *Proc Natl Acad Sci U S A* 2005;**102**:6201–6.

37. Søndergaard CR, Olsson MH, Rostkowski M, *et al*. Improved treatment of ligands and coupling effects in empirical calculation and rationalization of pKa values. *J Chem Theory Comput* 2011;**7**:2284–95.
38. Friesner RA, Banks JL, Murphy RB, *et al*. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem* 2004;**47**:1739–49.
39. Wang L, Zhang L, Li L, *et al*. Small-molecule inhibitor targeting the Hsp90-Cdc37 protein-protein interaction in colorectal cancer. *Sci Adv* 2019;**5**:eaax2277.
40. Tang Y, Feng B, Wang Y, *et al*. Structure-based discovery of CZL80, a caspase-1 inhibitor with therapeutic potential for febrile seizures and later enhanced epileptogenic susceptibility. *Brit J Pharmacol* 2020;**117**(15):3519–34.
41. Pan P, Yu H, Liu Q, *et al*. Combating drug-resistant mutants of anaplastic lymphoma kinase with potent and selective type-I1/2 inhibitors by stabilizing unique DFG-shifted loop conformation. *ACS Cent Sci* 2017;**3**: 1208–20.
42. Xu L, Zhang Y, Zheng L, *et al*. Discovery of novel inhibitors targeting the macrophage migration inhibitory factor via structure-based virtual screening and bioassays. *J Med Chem* 2014;**57**:3737–45.
43. Liu Z, Su M, Han L, *et al*. Forging the basis for developing protein–ligand interaction scoring functions. *Acc Chem Res* 2017;**50**:302–9.
44. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.