Contents lists available at ScienceDirect

Journal of Chromatography A

journal homepage: www.elsevier.com/locate/chroma

# Deep graph convolutional network for small-molecule retention time prediction

Qiyue Kang[a,*,1], Pengfei Fang[b,1], Shuai Zhang[a], Huachuan Qiu[a], Zhenzhong Lan[a,*]

[a] School of Engineering, Westlake University, Hangzhou, Zhejiang, 310024, China
[b] School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, 210096, China

ARTICLE INFO

*Keywords:*
Retention time prediction
Graph neural network
Transfer learning

ABSTRACT

The retention time (RT) is a crucial source of data for liquid chromatography-mass spectrometry (LCMS). A model that can accurately predict the RT for each molecule would empower filtering candidates with similar spectra but differing RT in LCMS-based molecule identification. Recent research shows that graph neural networks (GNNs) outperform traditional machine learning algorithms in RT prediction. However, all of these models use relatively shallow GNNs. This study for the first time investigates how depth affects GNNs' performance on RT prediction. The results demonstrate that a notable improvement can be achieved by pushing the depth of GNNs to 16 layers by the adoption of residual connection. Additionally, we also find that graph convolutional network (GCN) model benefits from the edge information. The developed deep graph convolutional network, DeepGCN-RT, significantly outperforms the previous state-of-the-art method and achieves the lowest mean absolute percentage error (MAPE) of 3.3% and the lowest mean absolute error (MAE) of 26.55 s on the SMRT test set. We also finetune DeepGCN-RT on seven datasets with various chromatographic conditions. The mean MAE of the seven datasets largely decreases 30% compared to previous state-of-the-art method. On the RIKEN-PlaSMA dataset, we also test the effectiveness of DeepGCN-RT in assisting molecular structure identification. By 30% lessening the number of potential structures, DeepGCN-RT is able to improve top-1 accuracy by about 11%.

## 1. Introduction

Liquid Chromatography-Mass Spectrometry (LC-MS) has been used to characterize small molecule structures for many years. However, determining the structures of small molecules appears to be a challenging task in many fields, such as metabolomics and food analysis [1–5]. Although the tandem mass spectrometry (MS/MS) information had been proven useful in characterizing structures, the number of molecules that have MS/MS information is usually limited. For example, MassBank [6], the most popular open-source MS/MS database, only has 15055 unique compounds, while PubChem [7] and ZINC [8] contain 96.5 million and 230 million unique small molecular structures, respectively. The retention time can help to narrow down the number of annotation candidates by providing orthogonal information to MS/MS [9,10]. Small-molecule structural identification is often plagued by in-source ion or false-positive structural identification [11–13]. Retention time can be used to mitigate these problems. [14–18]. Therefore, numerous effects have been made to accurately predict retention time [9,10,19].

Traditional retention time prediction methods rely on handcrafted features such as molecular fingerprints and molecular descriptors [10,19,20]. Molecular fingerprints (e.g., extended-connectivity fingerprints) are a series of binary vectors that each number represents the existence or absence of a particular substructure, and they are often designed for similarity searching, clustering, and virtual screening [21]. Molecular descriptors are real-valued vectors that quantitatively describe the physical and chemical properties of the molecules [22]. While hand-designed methods have achieved promising results [23–26], end-to-end learning methods such as graph neural networks (GNNs) that directly take molecular graph embedding as inputs have greater potential as it learns to describe the molecule automatically. However, training a good GNN requires a large dataset. Luckily, METLIN recently released the small-molecule retention time (SMRT) dataset, which covers more than 80,000 molecules from the METLIN library, analyzed by reverse-phase

[*] Corresponding authors.
*E-mail addresses:* kangqiyue@westlake.edu.cn (Q. Kang), lanzhenzhong@westlake.edu.cn (Z. Lan).
[1] These authors contributed equally.

https://doi.org/10.1016/j.chroma.2023.464439
Received 10 July 2023; Received in revised form 4 October 2023; Accepted 6 October 2023
Available online 13 October 2023

liquid chromatography (RPLC) [27]. Based on this dataset, Yang et al. [28] built a custom GNN model named GNN-RT, which outperformed all other traditional methods including Bayesian ridge regression, random forest, and shallow artificial neural networks using fingerprints or descriptors; Kensert et al. [29] implemented a relatively shallow graph convolutional network (GCN) with 5 hidden layers, and the model is better than the model developed by Yang et al. [28]. However, it is unclear that whether deep GNN can further improve the accuracy of retention time prediction.

Model depth has been proven to be an important factor for convolutional neural network [30,31]. Recent researches also confirm that deep GNNs are indeed beneficial to the right level of task scale and/or complexity [32–34]. Therefore, we explore whether deep GNN can improve the prediction accuracy of small-molecule retention time. Specifically, we design a deep GCN model, named DeepGCN-RT, which shows that a significant improvement could be accomplished by pushing the model depth to 16 GCN layers. DeepGCN-RT outperforms several previous state-of-the-art models, such as GCN [29], DNNpwa [26], and GNN-RT [28] on the SMRT dataset. In addition to thoroughly comparing the performances of different GCN model depths, we confirm that residual connection is critical to train deeper GCN and that the edge information in the message passing will contribute to a increased model performance.

We further investigate how DeepGCN-RT adapts to different chromatographic conditions by fine-tuning it on seven different datasets. We observe a stunning 30% average performance improvement on these datasets. To evaluate the performance of the RT prediction model, we utilize the RIKEN-PlaSMA dataset for molecular identification. We observe that the top-1 accuracy can be improved by 11% by lessening about 30% of the candidates. We thoroughly investigate the use of deep GCN in small-molecule retention time prediction. The results suggest that the deep GCN is a competitive architecture, and the model developed could facilitate further structural identification in LCMS-based molecular structure analysis. To facilitate further research, we released our model weights and source code of DeepGCN-RT at https://github.com/kangqiyue/DeepGCN-RT.

## 2. Method

### 2.1. Dataset

The SMRT dataset contains the retention times of 80,038 molecules from the METLIN library analyzed by RPLC [27]. To compare the prediction accuracy of our model with those reported in the literature, non-retained molecules were excluded, in line with the previously published research [28]. The final set contains 77,980 molecules. We split the dataset into a training set and a test set, containing 70,182 and 7,798 molecules, respectively. It should be noted that we used the same test set of SMRT with Yang et al. [28], Kersert et al. [29], and Ju et al. [26]. Therefore the results in this study could be directly compared with these of them. To train the model, the training set was further divided into a new training set and a validation set, with a split ratio of 9:1. The molecular retention times of the training and validation set ranged from 5.67 to 24.53 min, and their masses ranged from 113.08 to 738.88 (Figure S1). The molecular retention times of the test set ranged from 8.07 to 24.16 min, with masses ranging from 165.04 to 665.30 (Figure S1).

In addition, this study also evaluated DeepGCN-RT on seven datasets with various chromatographic conditions. RPLC datasets, including Eawag_XBridgeC18 [35,36], FEM_lipids [37], FEM_long [38], IPB_Halle [39], LIFE_new [40], LIFE_old [40], and UniToyama_Atlantis [6] were collected from PredRet [41]. These datasets contain different numbers of molecules, ranging from 72 to 420, and the molecules were eluted with different column types and eluent conditions (Table S1). Therefore, these datasets are appropriate for assessing the transfer learning performance of DeepGCN-RT. Detailed information on these datasets, including the liquid conditions, retention time ranges, and the number of compounds in each dataset, can be found in the Supporting Information.

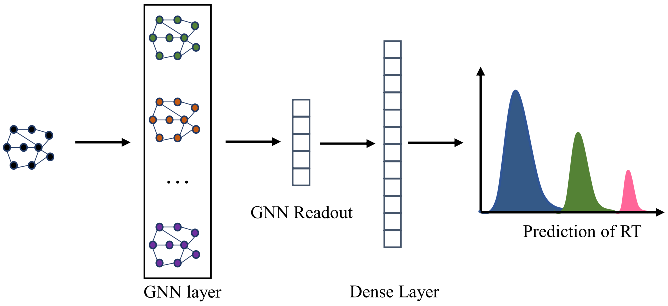

**Fig. 1.** Schematic figure of DeepGCN-RT.

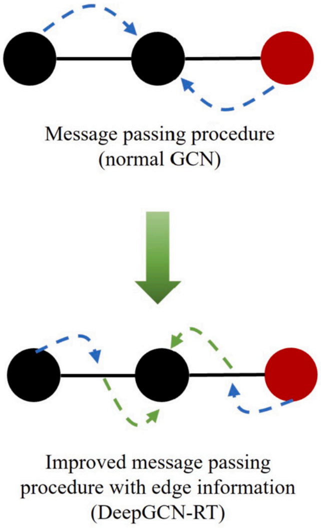

**Fig. 2.** The message passing process in normal GCN, and the improved edge message passing process in DeepGCN-RT.

### 2.2. Graph construction

Molecules can be naturally represented in graph form [42], and therefore it is straightforward to convert molecules to graphs. The graph construction process maps the atoms and bonds of molecules into graph datasets containing nodes and edges. In this study, the Simplified Molecular Input Line Entry System (SMILES) [43] strings of the molecules were converted to graph data using RDKit (version 2020.09) [44]. The atoms and bonds were transformed into nodes and edges of the graph, respectively. The atom properties and bond properties were converted to node features and edge features, respectively, following the procedures proposed by Kensert et al. [29]. The atom features included atom type, chiral center type, chirality, degree of the atom, formal charge, hybridization, aromaticity, hydrogen donor or acceptor, heteroatoms, in a ring of a particular size, number of hydrogens, number of radical electrons, number of valence electrons, Crippen LogP contribution, Crippen molar refractivity contribution, Gasteiger charge, mass, and the accessible surface area contribution (Table S2). The bond features included bond type, conjugated type, whether part of a ring or not, whether rotatable or not, and stereo-information (Table S2).

### 2.3. Model layer for GCN and DeepGCN-RT

The schematic figure of DeepGCN-RT is showed in Fig. 1. We first implemented the normal GCN model, which was used as the baseline in our study, following the procedures proposed by Kipf et al. [45] and Kensert et al. [29], as shown in Equation (1) and Fig. 2.

$$\boldsymbol{h_v}{}^{l+1} = \sigma(b^l + \sum_{u\in\mathcal{N}(v)} \frac{1}{c_{uv}} \boldsymbol{h_v}{}^{l} \boldsymbol{W}^l) + \boldsymbol{h}_{\boldsymbol{v}}^l \tag{1}$$

where $\mathcal{N}(v)$ is the set of neighbors of node $v$, $c_{uv}$ is the product of the square root of node degrees, and $\sigma$ is an activation function. Specifically, for the model without residual connection, we just implemented the model the same as above except removing the term of $\boldsymbol{h}_{\boldsymbol{v}}^l$.

Inspired by Li et al. [46], we implemented our DeepGCN-RT model that considered the edge information and residual connection in the message passing process. Let $\boldsymbol{u} \in \mathbb{R}^m, \boldsymbol{v} \in \mathbb{R}^m$, and $\boldsymbol{e} \in \mathbb{R}^n$ represent the source node, the destination node, and the edge attributes between them, respectively. The message of source node $\boldsymbol{h}_{\boldsymbol{u}}^l$ is first transformed to vectors from the raw features (including one-hot and other features) with a linear layer. The message of the edge information $\boldsymbol{h}_{\boldsymbol{e_{uv}}}^l$ is also transformed to vectors. Then the vector of the source node and that of the edge is summed The message of source node $\boldsymbol{h}_{\boldsymbol{u}}^l$ is summed with the message of the edge $\boldsymbol{h}_{\boldsymbol{e_{uv}}}^l$ (Equation (2)):

$$\boldsymbol{m}_{\boldsymbol{uv}}^l = \boldsymbol{h}_{\boldsymbol{u}}^l + \boldsymbol{h}_{\boldsymbol{e_{uv}}}^l \tag{2}$$

The messages from the source node $\boldsymbol{u}$ and edge $\boldsymbol{e_{uv}}$ are aggregated by softmax aggregation, as shown in Equation (3):

$$\boldsymbol{m}^l = \sum_{u\in\mathcal{N}(v)} \left(\frac{\exp(\boldsymbol{m}_{\boldsymbol{uv}}^l)}{\sum_{u\in\mathcal{N}(v)} \exp(\boldsymbol{m}_{\boldsymbol{uv}}^l)} \cdot \boldsymbol{m}_{\boldsymbol{uv}}^l\right) \tag{3}$$

where $\mathcal{N}(v)$ is the set of neighbors of node $v$, $exp$ is the exponential function.

Then $\boldsymbol{m}^l$ is transformed through a linear function, an activation function. Besides, residual connection is performed by adding the message of $\boldsymbol{h}_{\boldsymbol{v}}^l$, as shown in Equation (4).

$$\boldsymbol{h}_{\boldsymbol{v}}^{l+1} = \sigma(b^l + \boldsymbol{m}^l \boldsymbol{W}^l) + \boldsymbol{h}_{\boldsymbol{v}}^l \tag{4}$$

where $\sigma$ is an rectified linear unit (ReLU) activation function, and $b^l$ is the learnable bias.

### 2.4. Readout module

After the message passing process, the messages from each node are combined by a graph readout module. To get the graph embedding, average pooling use the average of each node's embedding in the graph. In addition, this study imported a readout module based on the graph attention mechanism, as performed by Xiong et al. [47]. In general, one super-virtual node that connects every node in the graph is created, and its embedding is obtained by combining each node's embedding using graph attention. Then, the updated embedding and the embedding obtained by the graph attention mechanism are fed into the gated recurrent unit (GRU) to obtain the final embedding of the molecule, as follows:

$$\boldsymbol{h}^k, \boldsymbol{c}^k = \mathrm{GRU}(\boldsymbol{h}^{k-1}, \boldsymbol{c}^{k-1}) \tag{5}$$

where $\boldsymbol{h}^k$, $\boldsymbol{c}^k$ are the $k$-times updated embedding of the graph and the embedding of the super-virtual node calculated by the graph attention mechanism, respectively. Specifically, $\boldsymbol{h}^0$ is calculated by summing the embedding of all nodes. Finally, the molecular embedding is fed into a dense layer to predict the retention time.

### 2.5. Training details

For the DeepGCN-RT model, the hidden dimension of the GNN layers for all models was 200, and the hidden dimension of the dense layer was 1024. The $k$ for the readout layer was set to 2. The DeepGCN-RT was trained on the training set of SMRT, and the validation set was used for model selection. The final performances were evaluated on the test set to ensure transferability and generalizability. The learning rate was 0.001, with a batch size of 64. The learning rate schedule was cosine annealing, and the dropout rate was 0.1. The Adam optimizer, Huber loss, and early stop strategy were adopted. The training was terminated if the validation loss failed to decrease after 30 consecutive epochs. Additionally, the number of training epochs was limited to 200 to reduce time consumption.

**Table 1**
Overall performance.

| | MAE (s) ↓ | MedAE (s) ↓ | MAPE ↓ | $R^2$ ↑ |
|---|---|---|---|---|
| DeepGCN-RT | **26.55** | **12.38** | **0.03** | **0.89** |
| GCN[a] | 29.4 | 15.02[b] | 0.04 | **0.89** |
| DNNpwa[a] | 39.62 | 25.08 | 0.05 | 0.85 |
| GNN-RT[a] | 39.87 | 25.24 | 0.05 | 0.85 |
| 1D CNN[c] | 34.7 | 18.7 | 0.04 | - |

[a] The results of the GCN, DNNpwa, and GNN-RT models were obtained from Kensert et al. [29], Ju et al. [26], and Yang et al. [28], respectively.

[b] To get the MedAE of GCN, We replicated the experiments of Kensert et al. [29]. We provided the replicated results in the Supporting Information.

[c] The results were abstracted from the paper of Fedorova et al. [51].

The model performances were evaluated through metrics including mean absolute error (MAE), median absolute error (MedAE), mean absolute percentage error (MAPE), mean square error (MSE), and $R^2$. Detailed descriptions of the calculation of MAE, MedAE, MAPE, MSE, and $R^2$ can be found in the Supporting Information. Each experiment was repeated three times with three different seeds, and the mean values and standard deviation values of the three runs were recorded. We chose the learning ratenumber, number of k as hyperparameters, and we determined them by grid-search. All GCN models were implemented in Python 3.6 with PyTorch (version 1.10.1 with CUDA 11.3) [48], DGL (version 0.8.0) [49], and DGL-LifeSci (version 0.2.9) [50].

### 2.6. Transfer learning

To test the transfer ability of our DeepGCN-RT model, we tested it on seven datasets from PredRet [41]: Eawag_XBridgeC18, FEM_lipids, FEM_long, IPB_Halle, LIFE_new, LIFE_old, and UniToyama_Atlantis. Ten rounds of 10-fold cross-validation were performed with 10 different seeds, and the results averaged over the 10 rounds were recorded. During the transfer learning process, all parameters were optimized. The transfer learning process was conducted by fine-tuning DeepGCN-RT with a batch size of 8. Adam optimizer and Huber loss were used with a learning rate of 0.001. Again, the maximum number of training epochs was 200, and the early-stop tolerance was 30 epochs. As a baseline, we also performed the 10 rounds of 10-fold cross-validation by training from scratch, which meant that we did not use the parameters of DeepGCN-RT.

## 3. Results and discussion

### 3.1. Overall performance

The performance of DeepGCN-RT was compared with those from several state-of-the-art models, including GCN [29], DNNpwa [26], GNN-RT [28], 1D-CNN [51]. The MAE of DeepGCN-RT was 26.55 s, while the MAEs of GCN, DNNpwa, GNN-RT, and 1D CNN were 29.4, 39.62, 39.87 and 34.7 s, respectively. The prediction retention times had a relatively low prediction errors (Fig. 3). The MedAE of DeepGCN-RT was 12.38 s, while the MedAEs for DNNpwa, GNN-RT, and 1D CNN were 25.08, 25.24 and 18.7 s, respectively. Overall, DeepGCN-RT achieved an MAE decrease of 9.7% compared to GCN, and 33%, and 33.4% decreases compared to DNNpwa, and GNN-RT, respectively. The MAE, MedAE, and MAPE values of DeepGCN-RT were the lowest among all methods (Table 1), demonstrating the competitive performance of the DeepGCN-RT model.

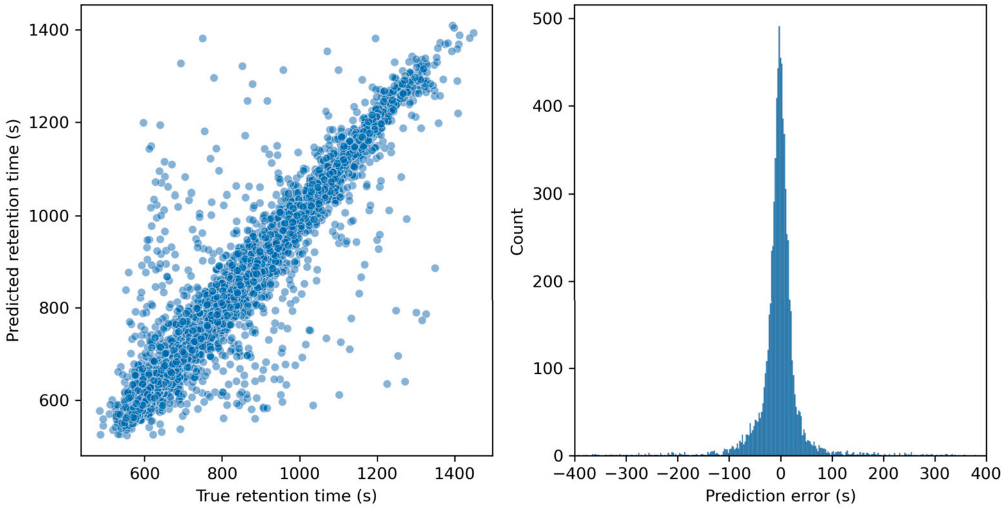

**Fig. 3.** Prediction results of DeepGCN-RT on the SMRT test set.

**Table 2**
The performance for normal GCN without edge information and GCN with edge information in the GCN message passing process.[a]

| Model | Model depth | MAE (s) ↓ | | MedAE (s) ↓ | | MAPE ↓ | | $R^2$ ↑ | | MSE ↓ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| normal_GCN | 3 | 35.30 | 0.30 | 18.45 | 0.22 | 0.044 | 0.000 | 0.861 | 0.001 | 4,238 | 28 |
| | 5 | 31.77 | 0.19 | 15.97 | 0.28 | 0.040 | 0.000 | 0.874 | 0.001 | 3,844 | 23 |
| GCN_edge | 3 | 30.97 | 0.07 | 16.14 | 0.21 | 0.039 | 0.000 | 0.882 | 0.001 | 3,583 | 18 |
| | 5 | **30.12** | **0.26** | **15.51** | **0.38** | **0.038** | **0.000** | **0.885** | **0.000** | **3,493** | **11** |

[a] To calculate the mean and standard deviation values, each experiment was repeated three times with three different seeds.

### 3.2. *Effect of edge information*

We introduced the edge information to the developed model, and to compare the model performance, we also implemented the normal GCN model. The difference was that whether the edge information was conducted in the message passing process (Fig. 2). Kensert et al. showed that the GCN without edge information outperformed the RGCN with edge information, they did not conduct whether the edge information could improve the GCN model's performance [29]. Our study comprehensively compared the GCN model with edge information and that without information. As shown in Table 2, the mean MAE decreased from 35.30 s to 30.97, for the 3-layers model, and decreased from 31.77 s to 30.12 s for the 5-layers model, respectively. The GCN model benefit from the inclusion of edge information.

### 3.3. *Increasing the model depth*

As described before, recent researches also confirm that deep GNNs are indeed beneficial to the right level of task scale and/or complexity [32–34]. Therefore, we then explored whether deep GCN can improve the prediction accuracy of small-molecule retention time.

When increasing the model depth, GCN often suffers from oversmoothing issues [46]. One straightforward way to alleviate the oversmoothing problem for GCN is residual connection [31,52]. Therefore, we implemented two versions of models, GCN_edge (model without residual connection) and GCN_edge_residual (model with residual connection), respectively. As shown in Table 3 and Table S4, without residual connection implemented, the MAEs of models with 3, 5, 8, and 16 hidden GCN layers were 30.97, 30.12, 31.73, and 41.26 s, respectively. For the models with residual connection, the MAEs of models with 3, 5, 8, 16 hidden GCN layers were 28.70, 28.03, 27.44, and 27.51 s, respectively. On one hand, the residual connection significantly improved the model performance when the models have the same depth. On the other hand, with the help of the residual connection, the model performance became better when increasing the model depth from 3 to 8, while the performance of 16-layers model was a little worse that the of 8-layers model. The previously published state-of-the-art mode, GCN model [29], did not implement the residual connection and it has a relatively shallow model depth (5 GCN layers). This study showed that, except for the edge information, the residual connection and model depth contribute to the increased model performance. Overall, GCN benefits from residual connection.

### 3.4. *Different readout module*

To explore the influences of the readout process, this study further performed an analysis on different readout modules. Average pooling was evaluated. In addition, inspired by Xiong et al. [47], we adopted a readout module in DeepGCN-RT, which includes the attention-based readout and the GRU recurrent network unit. This readout module performed well in terms of information retention and filtering [53].

The results are listed in the Table 3. For the models with average pooling (GCN_edge_residual), the MAEs of 3, 5, 8, and 16 hidden GCN layers were 28.70, 28.03, 27.44, and 27.51 s, respectively, while DeepGCN-RT, which used the attention-based pooling, have MAEs of 27.97, 27.00, 26.61, 26.55 s, respectively. In general, the performance of graph attention-based readout is better than average pooling.

### 3.5. *Ablation study*

To illustrate the effectiveness of residual connection, edge information, and graph attention mechanism-based readout of DeepGCN-RT in the retention time prediction, we perform an ablation study for the DeepGCN-RT model. The results are listed in Fig. 4 and Table S5. The residual connection has a critical effect on the model performance. Residual connection is critical to build deeper models, especially for pushing the model depth to 16 layers.

**Table 3**
Performance comparison for different model architectures.[a]

| Model | Layers | MAE (s) ↓ | | MedAE (s) ↓ | | MAPE ↓ | | $R^2$ ↑ | | MSE ↓ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| GCN_edge[b] | 3 | 30.97 | 0.07 | 16.14 | 0.21 | 0.039 | 0.000 | 0.882 | 0.001 | 3,583 | 18 |
| | 5 | 30.12 | 0.26 | 15.51 | 0.38 | 0.038 | 0.000 | 0.885 | 0.000 | 3,493 | 11 |
| | 8 | 31.73 | 0.92 | 17.20 | 1.32 | 0.039 | 0.001 | 0.882 | 0.001 | 3,595 | 43 |
| | 16 | 41.26 | 1.78 | 27.43 | 2.22 | 0.052 | 0.003 | 0.858 | 0.005 | 4,342 | 168 |
| GCN_edge_residual(average pooling)[c] | 3 | 28.70 | 0.16 | 14.20 | 0.24 | 0.036 | 0.000 | 0.888 | 0.001 | 3,421 | 29 |
| | 5 | 28.03 | 0.10 | 13.60 | 0.06 | 0.035 | 0.000 | 0.889 | 0.001 | 3,391 | 23 |
| | 8 | 27.44 | 0.07 | 13.05 | 0.03 | 0.034 | 0.000 | 0.890 | 0.001 | 3,340 | 17 |
| | 16 | 27.51 | 0.17 | 12.86 | 0.07 | 0.035 | 0.000 | 0.887 | 0.001 | 3,434 | 44 |
| DeepGCN-RT[d] | 3 | 27.97 | 0.20 | 14.01 | 0.07 | 0.035 | 0.000 | **0.892** | **0.002** | 3,303 | 55 |
| | 5 | 27.00 | 0.19 | 12.91 | 0.18 | 0.034 | 0.000 | **0.892** | **0.001** | 3,288 | 33 |
| | 8 | 26.61 | 0.09 | 12.44 | 0.05 | 0.034 | 0.000 | **0.892** | **0.001** | 3,286 | 31 |
| | 16 | **26.55** | **0.17** | **12.38** | **0.12** | **0.033** | **0.000** | **0.892** | **0.001** | **3,299** | **45** |

[a] To calculate the mean and standard deviation values, each experiment was repeated three times with three different seeds.
[b] Edge information and average pooling are used for GCN_edge.
[c] Edge information and residual connection, average pooling are used for GCN_edge_residual.
[d] Edge information, residual connection, and attention-based readout module are used for DeepGCN-RT, and the details could be found in the Method Section.

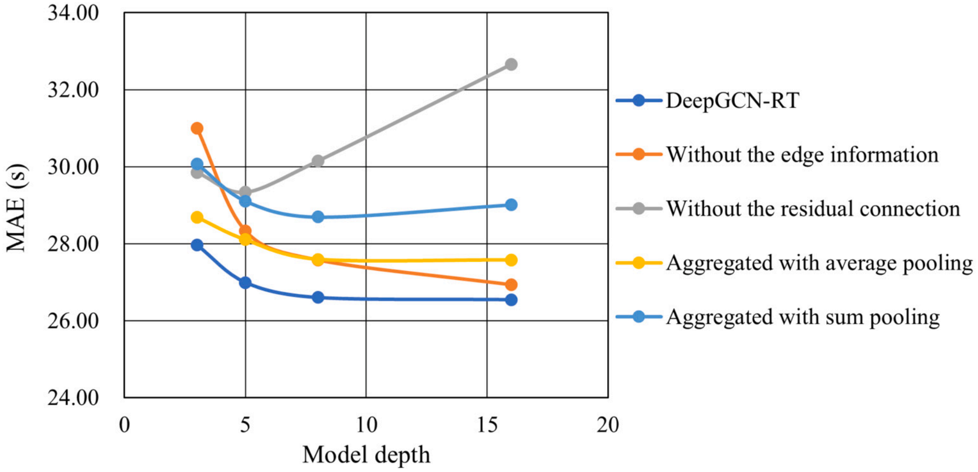

**Fig. 4.** Ablation study for DeepGCN-RT.

The graph-attention based readout has a second important effect on the model performance. The readout module in DeepGCN-RT used a supernode that connects all nodes in the atom, and it performed the readout process using the GRU recurrent network unit, which performed well in terms of information retention and filtering [47]. The performance of average pooling was better than sum pooling, while both were worse than the graph attention mechanism-based readout.

The performance of the model without edge information was a little worse than that of the model with edge information (Fig. 4). The edge information has a very important effect at the model depth of 3, and when we push the model depth to 16 layers, the effects of edge information seem to be decreasing. Overall, the improvement effects for the edge information are still present for even deeper networks, and the effects of edge information seem to be decreasing for deeper networks. In general, more information is conducive to the model's performance.

In addition, to further analyse the effects of model depth, we conduct a complete analysis between the model performance and the number of hidden layers. As shown in Fig. 5 and Table S6, the training loss is continually decreasing when increasing the model depth. The validation loss is decreasing fast from 3 hidden layers to 9 hidden layers, then the loss is slightly decreasing after 10 hidden layers. We also conducted the performance comparison with newer GNN models, such as Graph Attention Network (GAT) [54,55], and Graph Isomorphism Network (GIN) [56]. The results are shown in Table S7. The best model is GIN accompanied with the attention-based readout, which has validation MAE of 27.8 s, and its test MAE is 27.4 s. The GIN model has a competitive performance compared to our DeepGCN-RT (with a test MAE of 26.55 s).

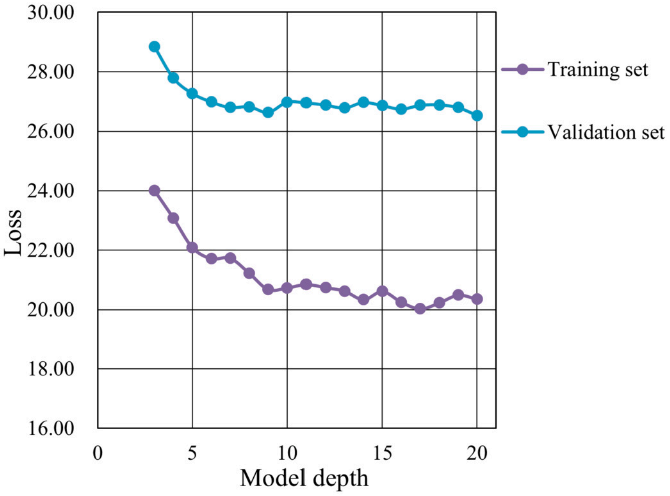

**Fig. 5.** The performance of model with different depth.

Furthermore, we tried to analyze the importance of different features and their contribution to the model performance using explainable

**Table 4**
Performance of DeepGCN-RT on different chromatographic datasets using transfer learning.

| Data set | MAE (s) ↓ | | | | MedAE (s) ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| | DeepGNN-RT-TL | DNNpwa-TL[a] | GNN-RT-TL[a] | Reduction (%) | DeepGNN-RT-TL | DNNpwa-TL[a] | GNN-RT-TL[a] | Reduction (%) |
| Eawag_XBridgeC18 | **45.97** | 89.26 | 112.78 | 49% | **29.59** | 68.56 | 83.79 | 57% |
| FEM_lipids | **74.48** | 89.32 | 128.25 | 17% | **32.64** | 63.79 | 79.49 | 49% |
| FEM_long | **110.89** | 161.58 | 235.01 | 31% | **55.35** | 72.43 | 94.66 | 24% |
| IPB_Halle | **21.20** | 28.85 | 29.76 | 27% | **14.13** | 15.98 | 19.25 | 12% |
| LIFE_new | **18.13** | 27.53 | 29.38 | 34% | **9.89** | 15 | 15.16 | 34% |
| LIFE_old | **12.07** | 15.11 | 17.1 | 20% | **7.28** | 10.67 | 12.88 | 32% |
| UniToyama_Atlantis | **50.03** | 72.3 | 127.52 | 31% | **27.52** | 39.19 | 99.61 | 30% |
| Mean | - | - | - | 30% | - | - | - | 34% |

[a] The results of DNNpwa-TL and GNN-RT-TL were adopted from Ju et al. [26] and Yang et al. [28], respectively.

artificial intelligence methods. We performed the ablation study for all kinds of features (including 20 kinds of node features and 5 kinds of edge features). We excluded one kind of feature one time, and summarized the model performances in Figure S4 and Table S11. It seems that the most important atom feature is the Cahn-Ingold-Prelog(CIP) code (R or S) of the atom, which affects the model performance critically. The second and third important atom features are the formal charge, and the tspa contrib (the contribution of each atom to the TPSA). For the bond features, the is_in_ring, the edge_bond_type, and the is_conjugated are the most important three kinds of features.

### 3.6. Different chromatographic systems

In real-world applications, the chromatographic conditions typically differ between different groups or studies. The retention time prediction model built on one dataset can not be directly applied to another dataset obtained under a different chromatographic condition. Therefore, transfer learning was adopted to utilize the prior knowledge that DeepGCN-RT had learned from the SMRT dataset. In this study, transfer learning was applied to nine datasets obtained from PedRet [41] to predict the retention times of small molecules. There were seven RPLC datasets, namely Eawag_XBridgeC18, FEM_lipids, FEM_long, IPB_Halle, LIFE_new, LIFE_old, and UniToyama_Atlantis. To evaluate the transfer learning ability of DeepGCN-RT, 10 rounds of 10-fold cross-validation were performed. Each fold of fitting was performed on a training set consisting of 90% of the total training set selected at random, with the remaining 10% being used as a hold-out set for validation. Ten different seeds were used for 10 rounds, and the results are summarized in Table 4 and Fig. 6. DeepGCN-RT outperformed the other methods on all datasets, with the lowest MAEs of 45.97, 74.48, 110.89, 77.70, 21.20, 18.13, 12.07, and 50.03 s, respectively, on the seven datasets listed above. The MAEs decreased by 49%, 17%, 31%, 27%, 34%, 20%, and 31%, respectively, in comparison to the previous best model DNNpwa.

In addition, to evaluate the performance of transfer learning, we compared the results from transfer learning with those for trained from scratch. It means that we trained model from scratch for the seven datasets obtained from PedRet, and we did not use the model weights of DeepGCN-RT obtained from SMRT dataset. The results are showed in Table 5. The MAEs from the transfer learning models were 45.97, 74.48, 110.89, 21.20, 18.13, 12.07, and 50.03 s, for Eawag_XBridgeC18, FEM_lipids, FEM_long, IPB_Halle, LIFE_new, LIFE_old, and UniToyama_Atlantis, respectively, while these from models trained from scratch were 59.60, 79.81, 152.32, 24.09, 18.11, 12.74, and 87.34 s, respectively. The results from the transfer learning outperformed those trained from scratch, while these two methods have close results for one dataset (LIFE_new). For all the datasets, the results from the transfer learning matched or exceeded those trained from scratch. We also performed de-duplication between SMRT and seven datasets for transfer learning. We compared the Canonical SMILES obtained by RDKit, and it turned out that the seven transfer learning datasets nearly have no duplicate molecules (Table S12).

**Table 5**
Performance of transfer learning and these of training from stretch.

| | MAE (s)↓ | MedAE (s)↓ | MAPE ↓ | R2 ↑ |
|---|---|---|---|---|
| Transfer learning | | | | |
| Eawag_XBridgeC18 | 45.97 | 29.59 | 0.15 | 0.90 |
| FEM_lipids | 74.48 | 32.64 | 0.30 | 0.82 |
| FEM_long | 110.89 | 55.35 | 0.27 | 0.95 |
| IPB_Halle | 21.20 | 14.13 | 0.28 | 0.91 |
| LIFE_new | 18.13 | 9.89 | 0.28 | 0.87 |
| LIFE_old | 12.07 | 7.28 | 0.17 | 0.89 |
| UniToyama_Atlantis | 50.03 | 27.52 | 0.07 | 0.90 |
| Trained from scratch | | | | |
| Eawag_XBridgeC18 | 59.60 | 39.06 | 0.17 | 0.86 |
| FEM_lipids | 79.81 | 46.24 | 0.22 | 0.81 |
| FEM_long | 152.32 | 73.17 | 0.28 | 0.92 |
| IPB_Halle | 24.09 | 16.42 | 0.19 | 0.91 |
| LIFE_new | 18.11 | 9.12 | 0.23 | 0.86 |
| LIFE_old | 12.74 | 7.62 | 0.17 | 0.88 |
| UniToyama_Atlantis | 87.34 | 48.83 | 0.12 | 0.65 |

### 3.7. DeepGCN-RT in small-molecule identification

To evaluate the molecule identification performance of DeepGCN-RT, we annotated the small molecules in the RIKEN-PlaSMA dataset from MoNA [57] using MSFinder [58] and DeepGCN-RT. The corresponding transfer learning model, denoted as DeepGCN-RT-TL, was built by fine-tuning the DeepGCN-RT using the retention times from the RIKEN-PlaSMA dataset. The RIKEN-PlaSMA dataset contained 434 molecules, and 100 of them were reserved as the test set for molecule identification. Totally 343 molecules were used in the fine-tuning process.

Then 100 molecules reserved for structural identification were annotated by MSFinder and DeepGCN-RT-TL-assisted MSFinder, separately. The difference between these two methods is whether we use the fine-tuned model to filter the candidate structures. For the MSFinder identification, the search methods were formula prediction and structure elucidation accompanied by in silico fragmentation. The mass tolerance for $MS^1$ is ±0.001 Da, and LEWIS and SENIOR checks [59] were performed for the calculated formulas. The formulas were searched through the structure databases to get possible candidates. The databases included HMDB, DrugBank, PubChem, and others (the full list was shown in the Supporting Information), and the maximum of candidate structures was 100. We used RDKit to calculate the canonical SMILES, and strictly compare them to the “true canonical SMILES” labeled in the RIKEN-PlaSMA dataset. It should be noted that, that we only used the formula prediction and structure elucidation accompanied by in silico fragmentation, and we did not search any online MS/MS database. More automated MS/MS searching tool could be developed, and well-structured databases are necessary in further research.

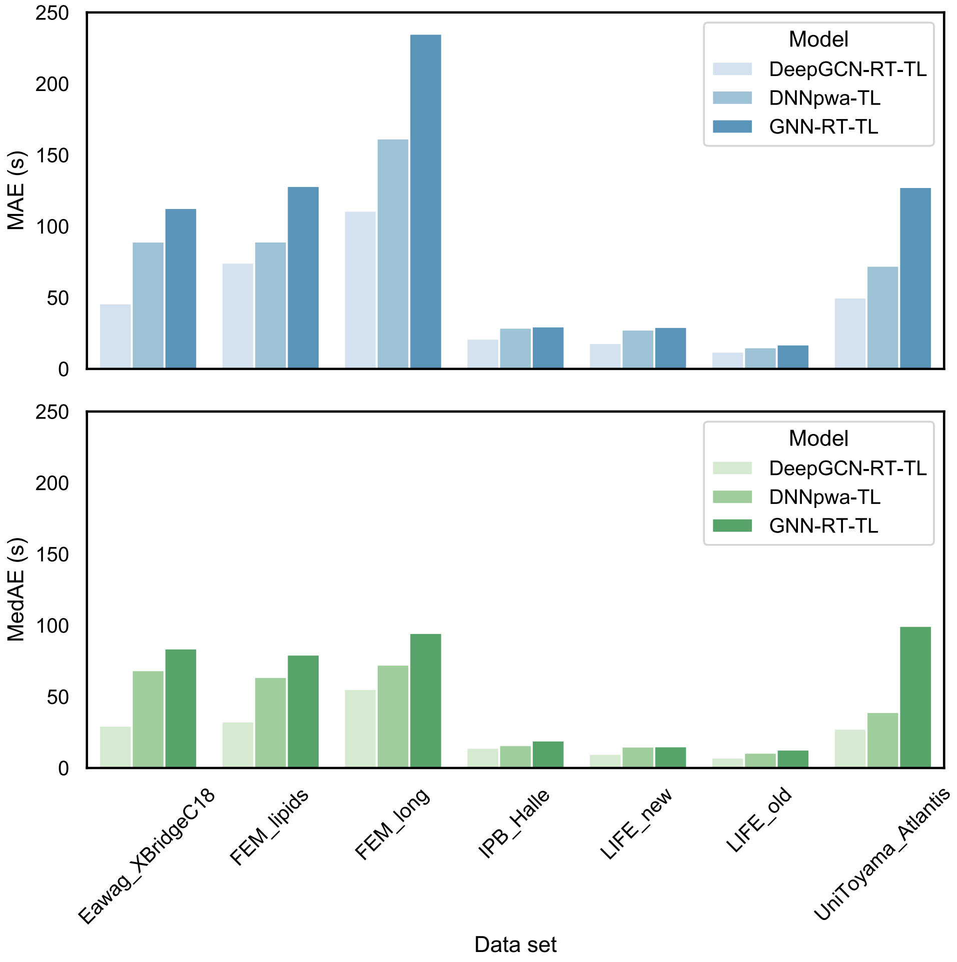

**Fig. 6.** Results of 10 rounds of 10-fold cross validation of DeepGCN-RT, DNNpwa, and GNN-RT (lower is better, and the results of DNNpwa-TL and GNN-RT-TL were abstracted from Ju et al. [26] and Yang et al. [28], respectively).

For the DeepGCN-RT-assisted method, after fine-tuning, the MAEs of DeepGCN-RT-TL for the training and validation sets were 9.04 s and 22.23 s, respectively (Table S4). Therefore, 66.69 s, a filtering threshold of 3 times the validating MAE, was used to filter the candidate structures. Overall, the mean number of the candidate structures in the annotation process for the test set reduced from 50 to 35, which decreased by 30% (Fig. 7), after we used the retention time information predicted by DeepGCN-RT-TL. In all cases, the application of DeepGCN-RT-TL largely reduced the number of possible candidate structures, demonstrating the effectiveness and efficiency of molecular identification based on retention time. In addition to the decreased number of candidates, the addition of retention time also improved the prediction accuracy of the top-*k* (Fig. 7). For MSFinder identification, the top-1, top-2, top-5, top-10, and top-20 accuracies were 23%, 28%, 30%, 31%, 33%, and 33%, respectively, while these were 26%, 29%, 31%, 32%, 33%, and 33% for DeepGCN-RT-TL assisted identification (Table S5). These results showed the effectiveness of DeepGCN-RT in structural identification.

In the meanwhile, we must point out that we did not adjust different hyper-parameters during the identification process. These hyper-parameters should be regarded as very important in real-world applications. Therefore, we have open-sourced all the model weights, and we hope these models could facilitate further structural identification in MS-based analysis. We also perform de-duplication between SMRT and RIKEN-PlaSMA dataset. For the RIKEN-PlaSMA dataset, the train split has 14 duplicate molecules, and the test split has 1 duplicate molecule. The training split and test split have 334 and 100 molecules, respectively. After de-duplication, the training split and test split have 320 and 99 molecules, respectively. The test MAE of the origin dataset is 21.45 s, while that of the dataset with de-duplication is 24.43 s. The model performances are summarized in Table S13. If the transferring datasets, especially the training set, have been used in pre-training, it appears that this could result in overly optimistic outcomes. Additional investigation is necessary to analyze the duplication of molecules in pre-training and transfer learning.

## 4. Conclusion

Prediction of chromatographic retention time has become an active research field as the retention time provides meaningful information orthogonal to that contained in MS/MS. In this study, we developed a deep GCN model named DeepGCN-RT, which outperformed several previous state-of-the-art models in predicting the retention times of the SMRT dataset. Furthermore, the DeepGCN-RT model was applied to other chromatographic datasets including RPLC and HILIC. It turned out that DeepGCN-RT also greatly reduced the predictive errors of these datasets. This study demonstrated the competitive prediction ability of deep GNN architecture, and the developed model could facilitate further structural identification in MS-based molecular structure analysis.

## 5. Note

The source code of the DeepGCN-RT is publicly available at https://github.com/kangqiyue/DeepGCN-RT.

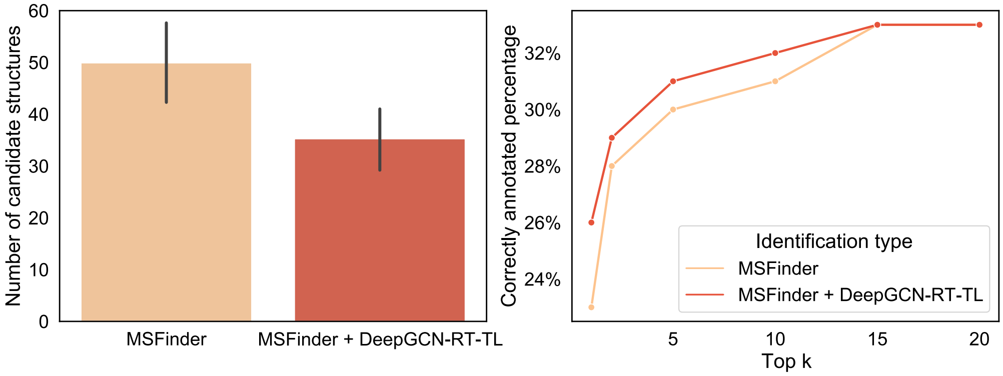

**Fig. 7.** Identification results of RIKEN-Plasma using MSFinder and DeepGCN-RT-TL (Only formula prediction and structure elucidation accompanied by in silico fragmentation were used in the identification process).

## CRediT authorship contribution statement

**Qiyue Kang:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Pengfei Fang:** Formal analysis, Writing – original draft. **Shuai Zhang:** Formal analysis. **Huachuan Qiu:** Formal analysis. **Zhenzhong Lan:** Funding acquisition, Project administration, Supervision, Writing – original draft.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work was supported by the Key Research and Development Program of Zhejiang Province of China under Grant No. 2021-C03139. We appreciated the help from Mufei Li for his kind explanations on DGL-LifeSci.

## Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.chroma.2023.464439.

## References

[1] T. Cajka, O. Fiehn, Toward merging untargeted and targeted methods in mass spectrometry-based metabolomics and lipidomics, Anal. Chem. 88 (2016) 524–545.
[2] T. Cajka, O. Fiehn, Comprehensive analysis of lipids in biological systems by liquid chromatography-mass spectrometry, TrAC, Trends Anal. Chem. 61 (2014) 192–206.
[3] C. Aydoğan, Recent advances and applications in LC-HRMS for food and plant natural products: a critical review, Anal. Bioanal. Chem. 412 (2020) 1973–1991.
[4] P. Lucci, J. Saurina, O. Núñez, Trends in LC-MS and LC-HRMS analysis and characterization of polyphenols in food, TrAC, Trends Anal. Chem. 88 (2017) 1–24.
[5] J. Hollender, E.L. Schymanski, H.P. Singer, P.L. Ferguson, Nontarget screening with high resolution mass spectrometry in the environment: ready to go?, Environ. Sci. Technol. 51 (2017) 11505–11512.
[6] MassBank, https://massbank.eu/MassBank/. (Accessed 23 May 2022).
[7] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B.A. Shoemaker, P.A. Thiessen, B. Yu, et al., PubChem 2019 update: improved access to chemical data, Nucleic Acids Res. 2019 (47) (2019) D1102–D1109.
[8] T. Sterling, J.J. Irwin, ZINC 15–ligand discovery for everyone, J. Chem. Inf. Model. 55 (2015) 2324–2337.
[9] R.I. Amos, P.R. Haddad, R. Szucs, J.W. Dolan, C.A. Pohl, Molecular modeling and prediction accuracy in quantitative structure-retention relationship calculations for chromatography, TrAC, Trends Anal. Chem. 105 (2018) 352–359.
[10] P.R. Haddad, M. Taraji, R. Szücs, Prediction of analyte retention time in liquid chromatography, Anal. Chem. 93 (2020) 228–256.
[11] Q. Kang, Q. Li, L. Wang, Y. Jia, X. Zhang, J. Hu, Comment on “Suspect and nontarget screening of per- and polyfluoroalkyl substances in wastewater from a fluorochemical manufacturing park”, Environ. Sci. Technol. 55 (2021) 5589–5592.
[12] R. Bouwmeester, R. Gabriels, N. Hulstaert, L. Martens, S. Degroeve, DeepLC can predict retention times for peptides that carry as-yet unseen modifications, Nat. Methods 18 (2021) 1363–1369.
[13] S.H. Giese, L.R. Sinn, F. Wegner, J. Rappsilber, Retention time prediction using neural networks increases identifications in crosslinking mass spectrometry, Nat. Commun. 12 (2021) 1–11.
[14] Y.-F. Xu, W. Lu, J.D. Rabinowitz, Avoiding misannotation of in-source fragmentation products as cellular metabolites in liquid chromatography–mass spectrometry-based metabolomics, Anal. Chem. 87 (2015) 2273–2281.
[15] Y. Jia, H. Zhang, W. Hu, L. Wang, Q. Kang, J. Liu, T. Nakanishi, Y. Hiromori, T. Kimura, S. Tao, et al., Discovery of contaminants with antagonistic activity against retinoic acid receptor in house dust, J. Hazard. Mater. (2021) 127847.
[16] J. Guo, S. Shen, S. Xing, H. Yu, T. Huan, ISFrag: de novo recognition of in-source fragments for liquid chromatography–mass spectrometry data, Anal. Chem. 93 (2021) 10243–10250.
[17] Q. Kang, F. Gao, X. Zhang, L. Wang, J. Liu, M. Fu, S. Zhang, Y. Wan, H. Shen, J. Hu, Nontargeted identification of per- and polyfluoroalkyl substances in human follicular fluid and their blood-follicle transfer, Environ. Int. 139 (2020) 105686.
[18] R.M. Gathungu, P. Larrea, M.J. Sniatynski, V.R. Marur, J.A. Bowden, J.P. Koelmel, P. Starke-Reed, V.S. Hubbard, B.S. Kristal, Optimization of electrospray ionization source parameters for lipidomics to reduce misannotation of in-source fragments as precursor ions, Anal. Chem. 90 (2018) 13523–13532.
[19] M. Witting, S. Böcker, Current status of retention time prediction in metabolite identification, J. Sep. Sci. 43 (2020) 1746–1754.
[20] M. Cao, K. Fraser, J. Huege, T. Featonby, S. Rasmussen, C. Jones, Predicting retention time in hydrophilic interaction liquid chromatography mass spectrometry and its use for peak annotation in metabolomics, Metabolomics 11 (2015) 696–706.
[21] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (2010) 742–754.
[22] B. Chandrasekaran, S.N. Abed, O. Al-Attraqchi, K. Kuche, R.K. Tekade, Dosage Form Design Parameters, 2018, pp. 731–755.
[23] R. Bouwmeester, L. Martens, S. Degroeve, Comprehensive and empirical evaluation of machine learning algorithms for small molecule LC retention time prediction, Anal. Chem. 91 (2019) 3694–3703.
[24] P. Bonini, T. Kind, H. Tsugawa, D.K. Barupal, O. Fiehn, Retip: retention time prediction for compound annotation in untargeted metabolomics, Anal. Chem. 92 (2020) 7515–7522.
[25] C. Feng, Q. Xu, X. Qiu, Y. Jin, J. Ji, Y. Lin, S. Le, J. She, D. Lu, G. Wang, Evaluation and application of machine learning-based retention time prediction for suspect screening of pesticides and pesticide transformation products in LC-HRMS, Chemosphere 271 (2021) 129447.
[26] R. Ju, X. Liu, F. Zheng, X. Lu, G. Xu, X. Lin, Deep neural network pretrained by weighted autoencoders and transfer learning for retention time prediction of small molecules, Anal. Chem. 93 (2021) 15651–15658.
[27] X. Domingo-Almenara, C. Guijas, E. Billings, J.R. Montenegro-Burke, W. Uritboonthai, A.E. Aisporna, E. Chen, H.P. Benton, G. Siuzdak, The METLIN small molecule dataset for machine learning-based retention time prediction, Nat. Commun. 10 (2019) 1–9.
[28] Q. Yang, H. Ji, H. Lu, Z. Zhang, Prediction of liquid chromatographic retention time with graph neural networks to assist in small molecule identification, Anal. Chem. 93 (2021) 2200–2206.
[29] A. Kensert, R. Bouwmeester, K. Efthymiadis, P. Van Broeck, G. Desmet, D. Cabooter, Graph convolutional networks for improved prediction and interpretability of chromatographic retention data, Anal. Chem. 93 (2021) 15633–15641.

[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2016, pp. 770–778.

[32] R. Addanki, P.W. Battaglia, D. Budden, A. Deac, J. Godwin, T. Keck, W.L.S. Li, A. Sanchez-Gonzalez, J. Stott, S. Thakoor, et al., Large-scale graph representation learning with very deep GNNs and self-supervision, arXiv preprint, arXiv:2107.09422, 2021.

[33] S. Zhang, L. Liu, S. Gao, D. He, X. Fang, W. Li, Z. Huang, W. Su, W. Wang, LiteGEM: lite geometry enhanced molecular representation learning for quantum property prediction, arXiv preprint, arXiv:2106.14494, 2021.

[34] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, J. Leskovec, OGB-LSC: a large-scale challenge for machine learning on graphs, arXiv preprint, arXiv:2103.09430, 2021.

[35] E.L. Schymanski, H.P. Singer, P. Longrée, M. Loos, M. Ruff, M.A. Stravs, C. Ripollés Vidal, J. Hollender, Strategies to characterize polar organic contamination in wastewater: exploring the capability of high resolution mass spectrometry, Environ. Sci. Technol. 48 (2014) 1811–1818.

[36] M.A. Stravs, E.L. Schymanski, H.P. Singer, J. Hollender, Automatic recalibration and processing of tandem mass spectra using formula annotation, J. Mass Spectrom. 48 (2013) 89–99.

[37] A. Della Corte, G. Chitarrini, I.M. Di Gangi, D. Masuero, E. Soini, F. Mattivi, U. Vrhovsek, A rapid LC–MS/MS method for quantitative profiling of fatty acids, sterols, glycerolipids, glycerophospholipids and sphingolipids in grapes, Talanta 140 (2015) 52–61.

[38] G. Theodoridis, H. Gika, P. Franceschi, L. Caputi, P. Arapitsas, M. Scholz, D. Masuero, R. Wehrens, U. Vrhovsek, F. Mattivi, LC-MS based global metabolite profiling of grapes: solvent extraction protocol optimisation, Metabolomics 8 (2012) 175–185.

[39] J. Stanstrup, M. Gerlich, L.O. Dragsted, S. Neumann, Metabolite profiling and beyond: approaches for the rapid processing and annotation of human blood serum mass spectrometry data, Anal. Bioanal. Chem. 405 (2013) 5037–5048.

[40] T. Barri, J. Holmer-Jensen, K. Hermansen, L.O. Dragsted, Metabolic fingerprinting of high-fat plasma samples processed by centrifugation- and filtration-based protein precipitation delineates significant differences in metabolite information coverage, Anal. Chim. Acta 718 (2012) 47–57.

[41] J. Stanstrup, S. Neumann, U. Vrhovsek, PredRet: prediction of retention time by direct mapping between multiple chromatographic systems, Anal. Chem. 87 (2015) 9421–9428.

[42] L. David, A. Thakkar, R. Mercado, O. Engkvist, Molecular representations in AI-driven drug discovery: a review and practical guide, J. Cheminform. 12 (2020) 1–22.

[43] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1988) 31–36.

[44] G. Landrum, RDKit: a Software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

[45] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint, arXiv:1609.02907, 2016.

[46] G. Li, C. Xiong, A. Thabet, B. Ghanem, DeeperGCN: all you need to train deeper GCNs, arXiv preprint, arXiv:2006.07739, 2020.

[47] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, et al., Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism, J. Med. Chem. 63 (2019) 8749–8760.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019) 8026–8037.

[49] M.Y. Wang, Deep graph library: towards efficient and scalable deep learning on graphs, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[50] M. Li, J. Zhou, J. Hu, W. Fan, Y. Zhang, Y. Gu, G. Karypis, DGL-LifeSci: an open-source toolkit for deep learning on graphs in life science, ACS Omega 6 (2021) 27233–27238.

[51] E.S. Fedorova, D.D. Matyushin, I.V. Plyushchenko, A.N. Stavrianidi, A.K. Buryak, Deep learning for retention time prediction in reversed-phase liquid chromatography, J. Chromatogr. A 1664 (2022) 462792.

[52] J. Chen, W. Liu, Z. Huang, J. Gao, J. Zhang, J. Pu, Universal deep GNNs: rethinking residual connection in GNNs from a path decomposition perspective for preventing the over-smoothing, arXiv preprint, arXiv:2205.15127, 2022.

[53] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, T. Langer, A compact review of molecular property prediction with graph neural networks, Drug Discov. Today Technol. 37 (2020) 1–12.

[54] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint, arXiv:1710.10903, 2017.

[55] V.P. Dwivedi, C.K. Joshi, A.T. Luu, T. Laurent, Y. Bengio, X. Bresson, Benchmarking graph neural networks, arXiv preprint, arXiv:2003.00982, 2020.

[56] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, arXiv preprint, arXiv:1810.00826, 2018.

[57] MassBank of North America (MoNA), https://mona.fiehnlab.ucdavis.edu, 2022. (Accessed 16 May 2022).

[58] H. Tsugawa, T. Kind, R. Nakabayashi, D. Yukihira, W. Tanaka, T. Cajka, K. Saito, O. Fiehn, M. Arita, Hydrogen rearrangement rules: computational MS/MS fragmentation and structure elucidation using MS-FINDER software, Anal. Chem. 88 (2016) 7946–7958.

[59] T. Kind, O. Fiehn, Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry, BMC Bioinform. 8 (2007) 1–20.