

Development of an AI-Driven Organic Synthesis Planning  
Approach with Retrosynthesis Knowledge

2 0 2 0

Shoichi Ishida



# General Introduction

Organic synthesis is the “art” of building organic molecules[1]. To plan the chemical syntheses of given molecules, chemists use retrosynthetic analysis, which is based on an imaginary process of breaking molecules down into simple building blocks[2]. E. J. Corey formalized the concept of retrosynthetic analysis (retrosynthesis knowledge) and synthetic problem-solving strategies (e.g., transform-based strategy) and stated that the concurrent use of as many different independent strategies as possible is key for efficient retrosynthetic analysis[3, 4]. For the selection of optimal strategies, chemists’ knowledge of chemistry and their experiments are essential; the optimal strategies depend on molecules, persons, and situations involved (e.g., lead optimization)[5], which is one of the reasons why organic synthesis is regarded as an “art”.

Since the 1960s, various computer-aided synthesis planning (CASP) applications have been developed to emulate chemists’ thinking and help organic synthesis chemists in their work[6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. CASP applications have played an important role in definable parts of synthesis (e.g., considering the characteristics of chemical structures and retrosynthetic tree size), while the indefinable parts of synthesis (e.g., chemists’ intuition) and opportunities to contribute creativity in retrosynthetic analysis have been left to chemists[6]. CASP approaches are generally classified into two types: rule-based[9, 15] and data-driven approaches[13, 16]. Rule-based approaches employ manually encoded (human-coded) transformations (reaction rules) considering such as stereo- and regioselectivity and electronic and steric effects[15, 20]. On the other hand, data-driven approaches aim to automatically extract knowledge related to transformations from numerous reaction records in order to discover synthetic routes[17]. Along with developments of these CASP tools, various computer-readable molecular representations (e.g., descriptors[21], fingerprints[22], strings[23], and graphs[24]) have also been developed (Fig. 0.1).

Recent advancements in machine learning (ML), including deep learning (DL), have dramatically improved the quality of CASP, triggering a revival of interest in CASP methods,

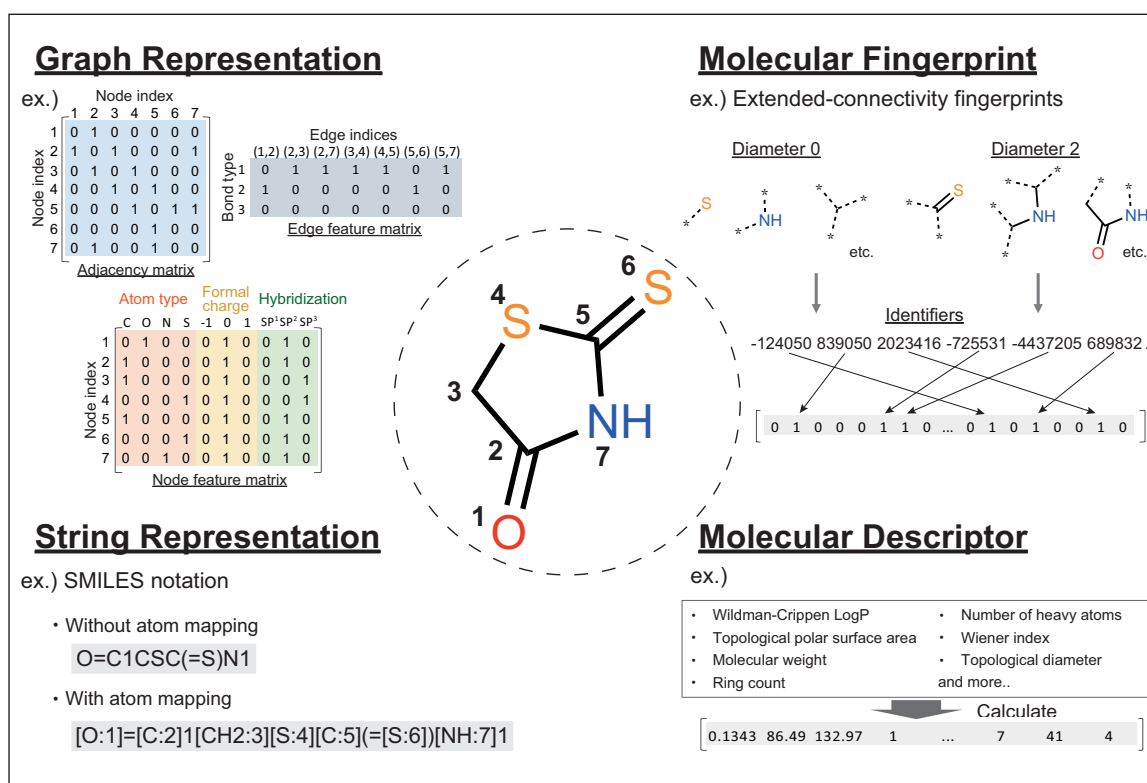


Figure 0.1: Computer-readable molecular representations.

especially data-driven methods[25, 26, 27]. Furthermore, several remarkable data-driven CASP applications are at the stage of practical uses in laboratories[16, 28, 29]. However, most data-driven CASP applications cannot explain how they make decisions due to the black-box problem of DL and still have an insufficient ability to reflect or apply chemists' "artistic" thinking into their systems flexibly.

This thesis describes the development of a data-driven CASP approach with retrosynthesis knowledge (Fig. 0.2). Based on the results obtained through the development of the proposed approach, this thesis also describes how computers recognize and learn organic molecules and how retrosynthesis knowledge influences a search algorithm for CASP.

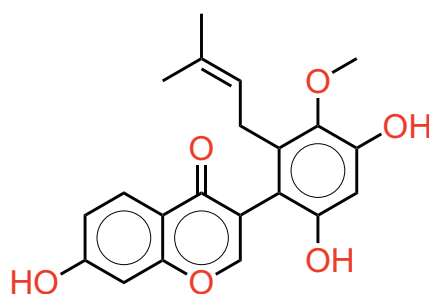
In this thesis, I divide the development of the proposed data-driven CASP approach into three parts, namely, the developments of (1) a graph-based DL platform, (2) an interpretable retrosynthetic reaction prediction model, and (3) a search algorithm integrating retrosynthesis knowledge for a multistep synthetic route search. Chapter 1 focuses on the development of the graph-based DL framework, in which a molecule is represented as a graph structure, for various prediction tasks in the life sciences. This includes graph convolutional networks (GCN) that show outstanding prediction performances for tasks related to molecules as well

---

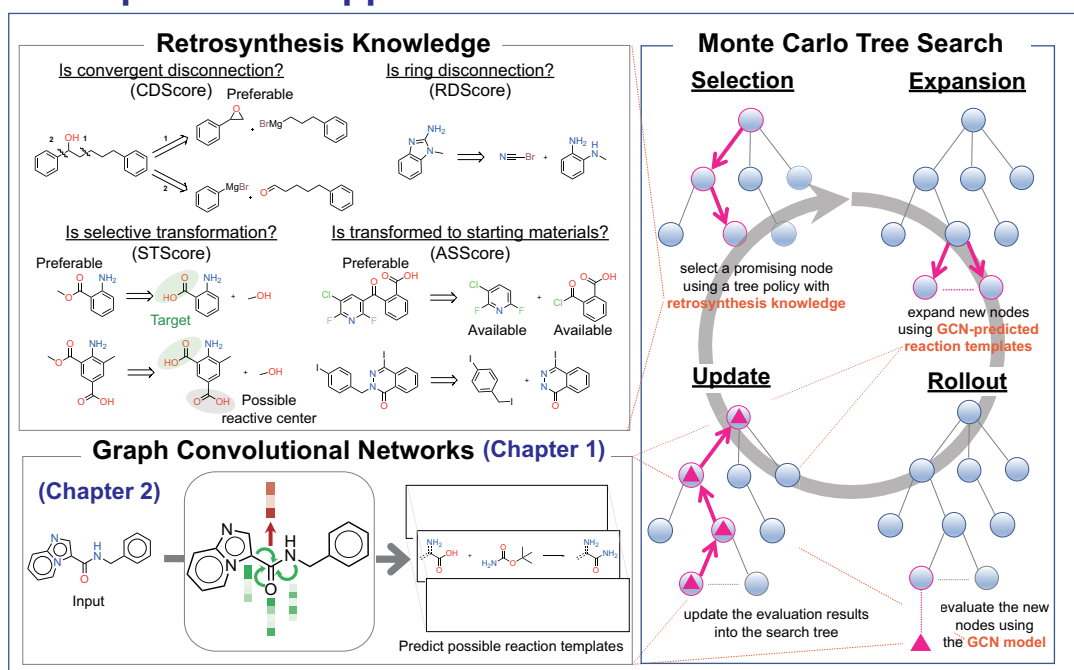
as a visualization technique, integrated gradients (IG), which can be used to calculate the relationships between input features (e.g., atoms) and model predictions. In the following two chapters, the developed framework is employed for a 1-step retrosynthetic reaction prediction task and incorporated into a data-driven CASP application. Chapter 2 focuses on evaluating the performance of a 1-step retrosynthetic reaction prediction model and its interpretability using GCN and IG. Chapter 3 focuses on the development of a hybrid CASP application combining ML techniques with various types of retrosynthesis knowledge and the assessment of the application's performance with and without the incorporation of retrosynthesis knowledge.

This thesis provides insight that how computers recognize and learn chemistry knowledge. This thesis is expected to contribute further developments and improvements in data-driven CASP applications to build a bridge between chemists and applications.

## Input molecule



## Developed CASP application in this thesis (Chapter 3)



## Designed synthetic route

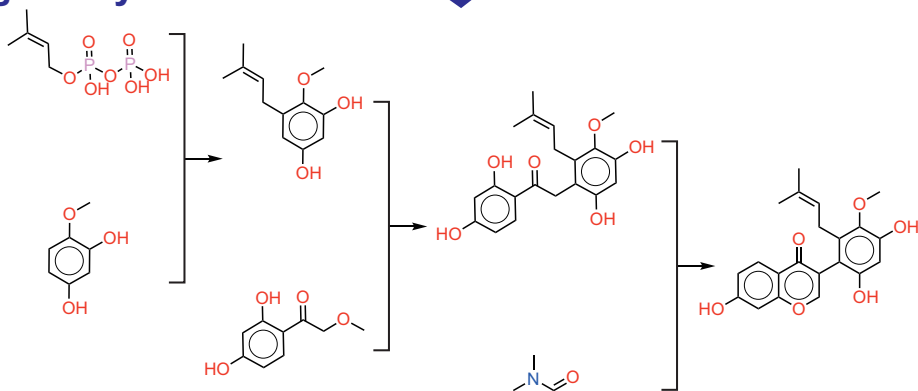


Figure 0.2: Graphical abstract of this thesis.

---

## Acknowledgments

To begin with, I would like to express my gratitude to my Ph.D. advisors, Prof. Yasushi Okuno and Prof. Kiyosei Takasu, for their continuous support of my Ph.D. studies and related research. Their immense knowledge, motivation, and patience have given me the ability and motivation to carry out my research and write research papers. Without their guidance and support, this thesis would not have been possible.

My deep appreciation goes out to Associate Prof. Kei Terayama at Yokohama City University for giving me his invaluable support and opportunities for discussion. It would not have been possible to conduct my research without his support and guidance.

Many thanks also to Assistant Prof. Ryosuke Kojima at Kyoto University and Dr. Nobuo Cho at the Institute of Physical and Chemical Research (RIKEN) who gave me constructive comments and discussed my research. I have learned a lot related to research from them, which has helped to facilitate my research.

I am also very grateful to Dr. Teruki Honma and Dr. Masateru Ohta at RIKEN, who gave me insightful comments and warm encouragement. Every time I had a discussion with them, their insightful and experienced comments inspired and motivated me.

I am indebted to all members of our laboratories for their helpful discussions and fun times in my Ph.D. life. Special thanks to Associate Prof. Mayumi Kamada, Prof. Masahiko Nakatsui (currently at Yamaguchi University), Dr. Shigeyuki Matsumoto, Assistant Prof. Hiroaki Iwata, Assistant Prof. Eiichiro Uchino, Dr. Yuta Isaka, Kei Taneishi, Dr. Daichi Kohmoto, Hiroshi Koshimizu, Kazuki Nakamura, Fumie Ono, Ayako Shimamura, Asuka Niwa, Maina Hayashi, Yukie Matsushita, Kanade Kokubo, Yukiko Neff, Yoshihisa Tanaka, and Dr. Yuji Okamoto.

Finally, I deeply appreciate my family, my father Mitsuhiro and my mother Mari, for their continuous support of my life so far. My sincere gratitude also goes to my fiancée Yuki Takahashi for her dedicated support. Without their support and encouragement, I could have not continued my Ph.D. journey.

---



# Contents

General Introduction . . . . .	i
Acknowledgments . . . . .	v
<b>1 Development of a Graph Neural Network Platform</b>	<b>1</b>
1 Introduction . . . . .	1
2 Implementation . . . . .	4
2.1 Graph representation of molecules for GCN . . . . .	4
2.2 GCN . . . . .	5
2.3 Visualization of GCN . . . . .	9
2.4 Hyperparameter optimization . . . . .	10
2.5 Interfaces . . . . .	10
3 Results . . . . .	16
4 Conclusion . . . . .	18
<b>2 Interpretable Retrosynthetic Reaction Prediction Using Graph Convolutional Networks</b>	<b>21</b>
1 Introduction . . . . .	21
2 Methods . . . . .	23
2.1 Dataset . . . . .	23
2.2 Retrosynthetic reaction prediction . . . . .	27
2.3 GCN and ECFP Models . . . . .	27
2.4 Visualization . . . . .	28
3 Results . . . . .	29
3.1 Retrosynthetic reaction prediction . . . . .	29
3.2 Visualization . . . . .	36

4	Discussion . . . . .	40
5	Conclusion . . . . .	42
<b>3</b>	<b>AI-Driven Synthetic Route Design with Retrosynthesis Knowledge</b>	<b>43</b>
1	Introduction . . . . .	43
2	Methods . . . . .	45
2.1	Construction of ReTReK . . . . .	45
2.2	Datasets . . . . .	46
2.3	MCTS for retrosynthesis . . . . .	48
2.4	Evaluating the effects of expansion sizes and retrosynthesis knowl- edge on MCTS solution performance . . . . .	52
2.5	Evaluating the effects of retrosynthesis knowledge on the search directions in MCTS . . . . .	52
3	Results and Discussion . . . . .	53
3.1	Top- <i>n</i> accuracies of the GCN-based policy network . . . . .	53
3.2	Effects of expansion sizes and retrosynthesis knowledge on the performance of solving for target molecules . . . . .	54
3.3	Effects of retrosynthesis knowledge on the search directions in MCTS	55
3.4	Demonstrations of ReTReK for drug-like molecules . . . . .	57
4	Conclusion . . . . .	65
	Conclusion . . . . .	66
	List of Publications . . . . .	69
	References . . . . .	70

# List of Figures

0.1	Computer-readable molecular representations. . . . .	ii
0.2	Graphical abstract of this thesis. . . . .	iv
1.1	Architecture of kGCN. . . . .	2
1.2	Example of molecular graph representation . . . . .	5
1.3	Visual images of graph convolution, graph dense, and graph gather operations. . . . .	7
1.4	GCN for a prediction task with a compound as input. . . . .	9
1.5	Multitask GCN with a compound as input. . . . .	9
1.6	Multimodal GCN with a compound and a sequence as inputs. . . . .	9
1.7	Single-task workflow for the hold-out procedure using the KNIME interface (top). Multitask workflow for the hold-out procedure (bottom). . . . .	13
1.8	Multimodal workflow for the hold-out procedure. . . . .	14
1.9	ROC-AUCs obtained from five-fold cross-validation. . . . .	17
1.10	(a) Chemical structure. (b) Atomic contributions to the predicted MMP-9 activity. Red shading indicates positive contributions to the prediction results (MMP-9 active in this case). Blue shading indicates negative contributions (not active). . . . .	18
2.1	Workflow of the framework based on the GCN and IG in a retrosynthetic reaction prediction. . . . .	23

---

2.2	Example of a reaction template extracted from the original reaction. . . . .	24
2.3	Overview of the GCN and ECFP models. . . . .	28
2.4	Comparison of the balanced accuracies of the GCN (blue) and ECFP (orange) models. . . . .	30
2.5	Comparison between the distributions of the top-10 accuracies for each template achieved by the GCN and ECFP models. . . . .	32
2.6	Comparison between the distributions of the top-1 and top-30 accuracies for each reaction template achieved by the GCN and ECFP models. (a) Histograms of the top-1 accuracies for each reaction template achieved by the GCN (blue) and ECFP (orange) models. (b) Scatter plot of the top-1 accuracies, with the color bar on the right representing the logarithm of the number of molecules in which the reaction template appears. (c) Histograms of the top-30 accuracies for each reaction template achieved by the GCN (blue) and ECFP (orange) models. (d) Scatter plot of the top-30 accuracies, with the color bar on the right representing the logarithm of the number of molecules in which the reaction template appears. . . . .	34
2.7	Histogram of the number of compounds per reaction template. The number of reaction templates is 1,752, and the number of compounds per template ranges from 50 to 8,000. . . . .	35
2.8	Distributions of the top-10 accuracies achieved by the GCN and ECFP models for the top 100 and bottom 100 reaction templates, where the reaction templates are ranked by the number of compounds per reaction template. (a) Scatter plot of the top-10 accuracies achieved by the GCN and ECFP models for the top 100 reaction templates. (b) Scatter plot of the top-10 accuracies achieved by the GCN and ECFP models for the bottom 100 reaction templates. . . . .	35
2.9	Visualization of the contributions of the atomic features in a molecule to retrosynthetic reaction prediction. The atoms marked in light green in a molecule correspond to the reaction center in the correct reaction template. The IG values are represented by shading in various colors, as shown by the color bar. Below the template drawings, the reaction templates are also expressed in the SMARTS format. . . . .	38

---

2.10	Histograms of the standardized average IG values in reaction centers (blue) and the standardized IG values of all atoms in a molecule (orange). . . . .	39
2.11	Examples of the top-1 predictions of my model for four natural products with different structural complexities. . . . .	41
3.1	Whole workflow of ReTReK. ReTReK combines MCTS and GCN techniques, and retrosynthesis knowledge is incorporated into the selection step of the MCTS procedure. The retrosynthesis knowledge is represented by four scores: the CDScore, STScore, RDScore, and ASScore. . . . .	45
3.2	Workflow of reaction template extraction. The reaction template extraction procedure consists of four steps: (1) Reaction records were standardized by removing explicit hydrogen, aromatizing, and keeping the largest fragments; (2) Reaction records were narrowed down under the condition that a reaction is a single-step reaction that has a product and 1–3 reactants; (3) Reaction templates were extracted from the filtered reaction records; (4) Sets of a product and the corresponding reaction template were filtered by the condition that the reaction template can reversibly be applied to the product and the derived reactants. . . . .	47
3.3	(a) Model architecture of the GCN-based policy network. (b) Top- $n$ accuracies of the model for $n$ values ranging from 1 to 1000. Specifically, the top-1, top-50, top-100, top-300, and top-500 accuracies are 0.361, 0.906, 0.938, 0.968, and 0.976, respectively. . . . .	53
3.4	Comparison of the numbers of solved molecules with different expansion sizes and retrosynthesis knowledge patterns. The green, orange, blue, and pink bars correspond to expansion sizes of 50, 100, 300, and 500, respectively. The results of retrosynthetic analyses with five retrosynthesis knowledge patterns (CDScore, ASScore, RDScore, STScore, and all knowledge) and without any retrosynthesis knowledge (no knowledge) are shown. . . . .	54

- 3.5 Comparison of the times necessary to solve compounds for different expansion sizes and retrosynthesis knowledge patterns. The green, orange, blue, and pink bars correspond to expansion sizes of 50, 100, 300, and 500, respectively. The results of retrosynthetic analyses with five retrosynthesis knowledge patterns (CDScore, ASScore, RDScore, STScore, and all knowledge) and without any retrosynthesis knowledge (no knowledge) are shown. The maximum reach of the whiskers in each boxplot is defined as  $1.5IQR$ , where  $IQR$  represents the interquartile range. Outliers, defined as data points beyond the whiskers, are not shown in the boxplots. . . . . 55
- 3.6 Evaluation of the effects of retrosynthesis knowledge on the search directions in MCTS. Synthetic routes solved with an expansion size of 500 were used for this evaluation. Each route score was standardized based on the corresponding mean and standard deviation for the no-knowledge pattern. The black, blue, orange, green, red, and gray plots represent the standardized route scores for the all-knowledge, STScore, CDScore, ASScore, RDScore, and no-knowledge patterns, respectively. The rhombuses represent the mean values for each case, and the confidence intervals at the 95% confidence level are also shown. . . . . 56
- 3.7 Comparison of (a) the synthetic route for a target compound (hepatitis B virus capsid inhibitor)[30] found by ReTReK with retrosynthesis knowledge and (b) a corresponding route found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. Molecule **1** is the target. . . . . 58
- 3.8 Comparison of (a) the synthetic route for a target compound (kwakhurin)[31] found by ReTReK with retrosynthesis knowledge and (b) a corresponding route found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. . . . . 59
- 3.9 Comparison of (a) the synthetic route for a target compound ( $\alpha 7$  nicotinic acetylcholine receptor silent agonist)[32] found by ReTReK with retrosynthesis knowledge and (b) a corresponding route found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. . . . . 60

- 
- 3.10 For the target compound (MtbTMPK inhibitor)[33] considered here, (a) a synthetic route was found using ReTReK with retrosynthesis knowledge, whereas (b) no synthetic route was found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. Molecule **19** is the target. . . . . 62
- 3.11 For the target compound (Propolone)[34] considered here, (a) a synthetic route was found by ReTReK with retrosynthesis knowledge, whereas (b) ReTReK without retrosynthesis knowledge did not find any synthetic route. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. . . . . 63
- 3.12 For the target compound (EGFR kinase inhibitor)[35] considered here, (a) a synthetic route was found by ReTReK with retrosynthesis knowledge, whereas (b) ReTReK without retrosynthesis knowledge did not find any synthetic route. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. . . . . 64





# List of Tables

1.1	Numbers of compounds in MMP inhibition assay dataset . . . . .	16
2.1	List of removed salts . . . . .	25
2.2	List of atomic features . . . . .	26
2.3	Top- $n$ balanced accuracies of the GCN and ECFP models . . . . .	29



# Acronyms

**ADMET** Absorption, Distribution, Metabolism Elimination, and Toxicology.

**AI** Artificial Intelligence.

**ASScore** Available Substances Score.

**Bash** Bourne again shell.

**BN** Batch Normalization.

**CASP** Computer-Aided Synthesis Planning.

**CDScore** Convergent Disconnection Score.

**CPI** Compound-Protein Interactions.

**CSV** Comma Separated Values.

**DL** Deep Learning.

**ECFP** Extended-Connectivity Fingerprint.

**GCN** Graph Convolutional Network.

**GNN** Graph Neural Network.

**GPU** Graphics Processing Unit.

**GUI** Graphical User Interface.

**IC50** half-maximal Inhibitory Concentration.

**IG** Integrated Gradients.

**kGCN** kyoto-university Graph Convolutional Network framework.

**KNIME** Konstanz Information Miner.

**MAE** Mean Absolute Error.

**MCTS** Monte-Carlo Tree Search.

**ML** Machine Learning.

**MMP** Matrix Metalloproteinase Inhibition.

**PROFEAT** Protein Features.

**QSAR** Quantative Structure-Activity Relationship.

**RDScore** Ring Disconnection Score.

**ReLU** Rectified Linear Unit.

**ROC-AUC** Area Under the Curve in Receiver Operator Characteristic curve.

**SAR** Structure-Activity Relationship.

**SDF** structure-data file.

**SMARTS** SMILES Arbitrary Target Specification.

**SMILES** Simplified Molecular Input Line Entry System.

**STScore** Selective Transformation Score.

**USAN** United States Adopted Names.

**USPTO** United States Patent and Trademark Office.

# Chapter 1

## Development of a Graph Neural Network Platform

### 1 Introduction

Deep learning (DL) is emerging as an important technology for performing various tasks in various fields, such as medicine[36, 37], language translation systems[38], agriculture[39], and drug discovery[40, 41, 42]. Focusing on the field of drug discovery, the application of DL approaches has been practically demonstrated for various prediction tasks, such as virtual screening[43]; quantitative structure-activity relationship (QSAR) studies[44]; retrosynthetic reaction prediction[45]; and absorption, distribution, metabolism, excretion, and toxicity (ADMET) prediction [46, 47]. In particular, with the democratization of artificial intelligence (AI), it is becoming expected that these prediction tools should be readily usable by non-experts. The accessibility of DL to non-experts is an important issue in the field of cheminformatics. For example, because DL can be applied in a wide range of research areas related to drug discovery, such as ADMET prediction for lead optimization or virtual screening for lead identification, chemists should be able to solve such research problems by using the latest technologies and analyzing the results, thus availing themselves of the benefits of DL. However, since chemists are typically not proficient in DL techniques, the development of easy-to-use, multifunctional DL software is necessary.

For prediction tasks based on molecular structures, graph neural network (GNN), in which a chemical structure is represented in the form of a graph, have been reported to perform well [48, 49]. In particular, graph convolutional network (GCN), as a type of GNN, exhibit excellent performance in many applications [50, 51]. Nevertheless, the appropriate application of GCN to real-world research problems requires practical programming skills

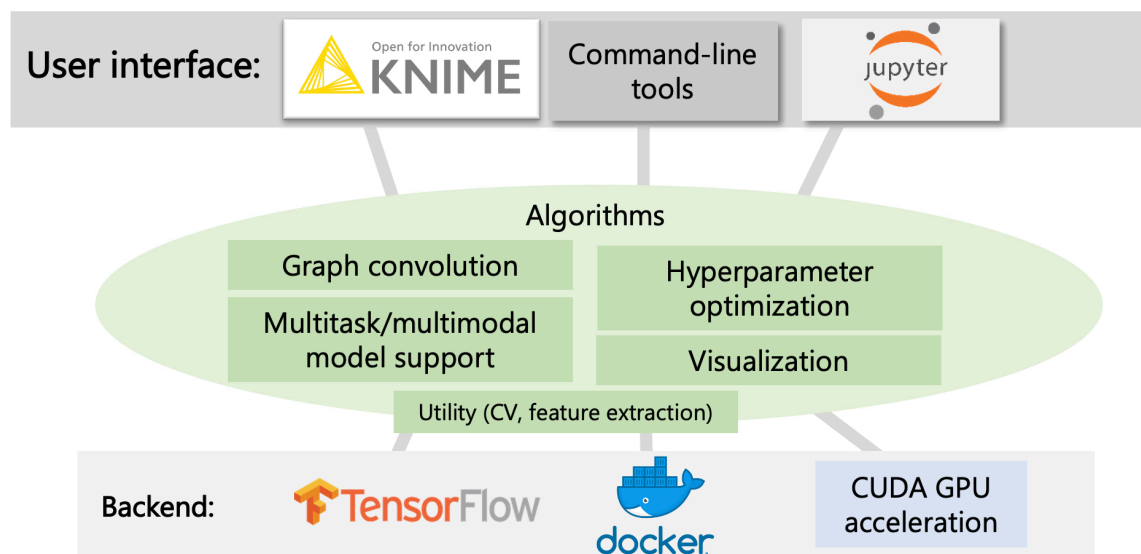


Figure 1.1: Architecture of kGCN.

and a comprehensive understanding of DL and GCN.

To address this issue, a new open-source software tool, Kyoto University Graph Convolutional Network Framework (kGCN), has been introduced to allow diverse users to take advantage of DL technology, including GCN. kGCN is designed to provide the following functions:

- Interfaces suitable for users of various levels, including users with limited programming skills
- Handling of different types of data for cheminformatics tasks
- Easy, intuitive, and convincing interpretation of results
- Hyperparameter optimization

As mentioned above, one function of kGCN is to provide interfaces to assist various users, such as chemists, cheminformaticians, and data scientists. Considering the diverse levels of expertise of these users, a software tool should provide multiple interfaces suitable for different types of users. To satisfy these requirements, kGCN provides three types of user interfaces. Figure 1.1 shows the architecture of the kGCN system. The kGCN system supports both a graphical user interface (GUI) and a command-line interface. To provide intuitive access to machine learning (ML) procedures, the kGCN system provides a GUI on the GUI platform known as the Konstanz Information Miner (KNIME)[52]. The command-line interface supports typical ML procedures such as training, evaluation, and cross-validation. Additionally, the kGCN modules can be used in the form of a Python library to allow flexibility and processing through programming languages. Users can

use the Python library in web applications such as Jupyter Notebook, and kGCN provides example files showing how to construct DL model using the Python library in Jupyter Notebook.

The second function is to support different types of data. In cheminformatics, various types of data, including chemical structures represented by graphs, should be considered. For example, protein sequence data are often represented as symbol sequences or vector descriptors. In the context of DL, various architectures for neural networks have been proposed[53]. The simplest GCN is based on a single-graph-input single-label-output architecture. The kGCN system supports (1) multi-input (multimodal GCN) and (2) multi-output (multitask GCN) architectures. A multimodal GCN is a neural network that can accept inputs of multiple modalities[54, 55]. kGCN can accommodate a neural network with two inputs: a chemical structure represented as a graph and a protein sequence represented as a series of characters. This type of neural network can be used to predict interactions between compounds and proteins for virtual screening and/or drug repurposing[43, 56]. In addition, multiple related tasks often need to be simultaneously handled in cheminformatics[57], for example, tasks for predicting multiple different properties of a compound. To address these tasks, a multitask neural network affords better results than individual predictions[58, 59].

The third function is the interpretation and understanding of the underlying causes of the prediction results via DL by visualizing the contributions of the input data to the prediction process. Such a visualization is important because the validity of a prediction model can be examined through a visual inspection of the good and bad features. The prediction model can be refined or reconstructed if the causes driving the prediction do not appear to be reasonable or are contrary to common sense. In particular, designing new molecules with improved properties is possible if the reasons for good and/or bad predictions can be identified through visualization. In recent years, several methods of calculating the different contributions to the prediction results obtained through DL have been proposed [60, 61]. The kGCN system uses the method of Integrated Gradients[62], which can be applied to any type of neural network architecture, including multitask and multimodal neural networks.

The last function is hyperparameter optimization. In analysis using deep neural networks, the hyperparameters for DL, such as the number of network layers, the number of nodes per layer, the learning rate, and the batch size, should be appropriately set. However, setting these parameters is not easy for users without knowledge and experience in DL. To assist such users by automatically determining the optimal hyperparameters, the kGCN system employs Bayesian optimization and metaheuristics for hyperparameter optimization [63].

In addition to these functions, the kGCN system also provides tools for improving usability. The back-end implementation of kGCN uses Tensorflow[64] and supports execution on a graphics processing unit (GPU). For setting up the execution conditions, kGCN-installed Docker images are also provided at <https://hub.docker.com/r/clinfo/kgcn>. Additional unique tools for enhancing the usability of kGCN are provided for each interface (see later description in the Implementation section).

Similar types of software have been reported in previous studies, e.g., DeepChem[65], Chainer Chemistry[66], and OpenChem[67]. DeepChem is a Python library for neural networks, including GCN. A notable feature of DeepChem is that it supports various ML methods as well as DL methods. Because DL usually requires large amounts of data, this feature can help users handle relatively small amounts of data. Chainer Chemistry provides support for GCN through an extended Python library of Chainer[68]. Both libraries can be used with Python and were developed for use by professional programmers familiar with ML and Python. Although OpenChem supports both command-line and Python interfaces, strong programming skills are still required to use it. By contrast, the kGCN system is a framework containing a GUI, a command-line interface, and a Python interface. The GUI of kGCN is expected to support users with limited programming skills related to GCN and DL. To my knowledge, kGCN is the first open-source and multifunctional GCN software to support all three types of interfaces.

## 2 Implementation

Before the details of the kGCN system are described, the basic implementation techniques for the graph representation of molecules and graph convolution are discussed.

### 2.1 Graph representation of molecules for GCN

This section first describes the formalization of a molecule for application in a GCN. Figure 1.2 shows an illustrative example of molecular graph representation. A molecule is formalized as a tuple  $\mathcal{M} \equiv (V, E, F)$ , where  $V$  is a set of nodes. Each node represents an atom in the molecule. Each node has features  $\mathbf{f}_i \in F$  ( $i \in V$ ), where  $F$  is a set of feature vectors representing the atomic properties such as atom type, formal charge, and hybridization. These features should be appropriately designed by the users.  $E$  is a set of edges, with each edge  $e \in E$  representing a bond between atoms, i.e.,  $e \in V \times V \times T$ , where



$T$  is a set of bond types. An adjacency matrix  $\mathbf{A}^{(t)}$  is used, which is defined as follows:

$$(\mathbf{A}^{(t)})_{i,j} = \begin{cases} 1 & (v_i, v_j, t) \in E \\ 0 & (v_i, v_j, t) \notin E \end{cases}, \quad (1.1)$$

where  $(\cdot)_{i,j}$  represents the  $j$ -th element in the  $i$ -th row. Similarly, the feature matrix is defined as follows:

$$(\mathbf{F})_{j,k} = (\mathbf{f}_j)_k, \quad (1.2)$$

where  $(\cdot)_k$  represents the  $k$ -th element of a vector.

Using these matrices, the molecule is represented by  $\mathcal{M}' = (\mathbf{A}, \mathbf{F})$ , where  $\mathbf{A} = \{\mathbf{A}^{(t)} | t \in T\}$ . In the present system, the RDKit library[69] is used to create the adjacency and feature matrices, and  $\mathcal{M}'$  is employed as the input to a GCN.

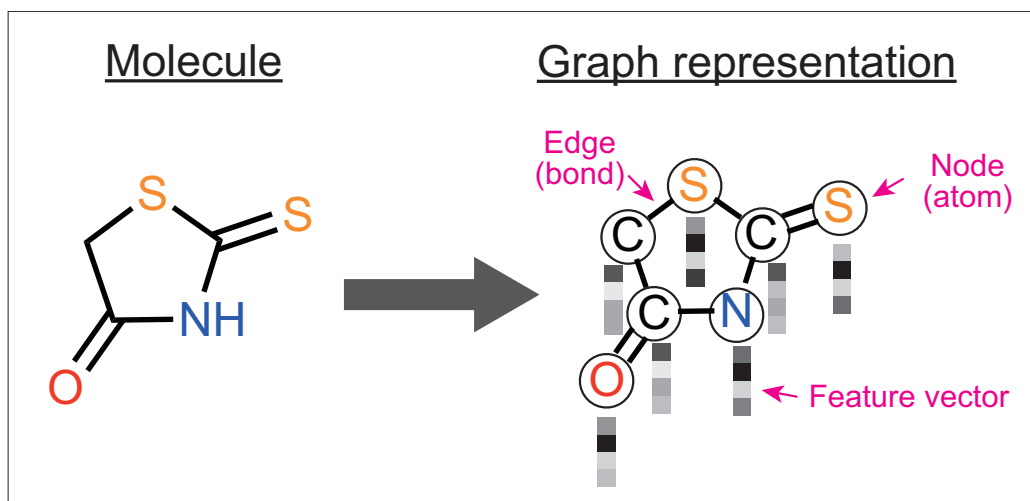


Figure 1.2: Example of molecular graph representation

## 2.2 GCN

kGCN supports GCN in addition to standard feedforward neural networks. Therefore, GCN for molecules are described first. Graph convolution layers, graph dense layers, and graph gather layers are defined as described below, and visual images of these layers are shown in Fig. 1.3.

### Graph convolution layer

The graph convolution operation is applied to the input  $\mathbf{X}^{(\ell)}$  to the  $\ell$ -th layer as

follows:

$$\mathbf{X}^{(\ell+1)} = \sigma \left( \sum_t \tilde{\mathbf{A}}^{(t)} \mathbf{X}^{(\ell)} \mathbf{W}_t^{(\ell)} \right), \quad (1.3)$$

where  $\mathbf{X}^{(\ell)}$  is an  $N \times D^{(\ell)}$  matrix,  $\mathbf{W}_t^{(\ell)}$  is the parameter matrix ( $D^{(\ell)} \times D^{(\ell+1)}$ ) for bond type  $t$ ,  $\sigma$  is the activation function, and  $\tilde{\mathbf{A}}^{(t)}$  is the normalized adjacency matrix ( $N \times N$ ). This layer normalization and implementation follows Kipf’s model [70] by default. There are various possible choices for implementing the settings of graph convolution layers. In the kGCN system, the operation on the first-layer input can be easily switched by changing the initial settings file for building the model.

A GCN is fundamentally based on this graph convolution operation. The input to the first layer,  $\mathbf{X}^{(1)}$ , often corresponds to the feature matrix,  $\mathbf{F}$ .

### Graph dense layer

When  $\mathbf{X}^{(\ell)}$  is the input to a graph dense layer,  $\mathbf{X}^{(\ell+1)}$  is calculated as follows:

$$\mathbf{X}^{\ell+1} = \mathbf{X}^{(\ell)} \mathbf{W}^{(\ell)}, \quad (1.4)$$

where  $\mathbf{X}^{(\ell)}$  is an  $N \times D^{(\ell)}$  matrix and  $\mathbf{W}^{(\ell)}$  is a parameter matrix ( $D^{(\ell)} \times D^{(\ell+1)}$ ).

### Graph gather layer

This type of layer converts a graph into a vector[71], i.e., if the input  $\mathbf{X}^{(\ell)}$  is an  $N \times D^{(\ell)}$  matrix,

$$(\mathbf{X}^{(\ell+1)})_j = \sum_j (\mathbf{X}^{(\ell)})_{ij}, \quad (1.5)$$

where  $(\cdot)_i$  represents the  $i$ -th element of a vector. This operation converts a matrix into a vector.

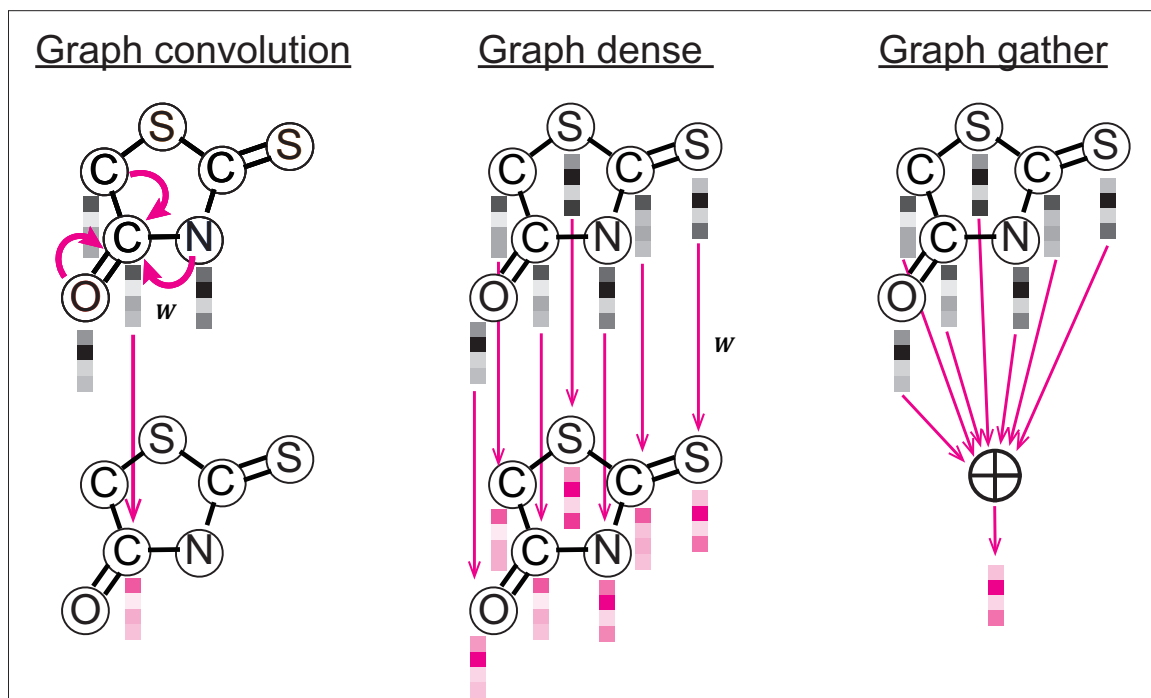


Figure 1.3: Visual images of graph convolution, graph dense, and graph gather operations.

Figure 1.4 shows an example of a GCN for a prediction task. The GCN model is a neural network consisting of a graph convolutional layer (GraphConv) with batch normalization (BN) [72] and rectified linear unit (ReLU) activation, a graph dense layer with ReLU activation, a graph gather layer, and a dense layer with softmax activation. Because it assigns labels that are suitable for each task to compounds, this type of model can be applied for many types of tasks, e.g., ADMET prediction based on chemical structures.

Figure 1.5 shows an example of a multitask GCN for a prediction task. The only difference is that multiple labels for each compound are predicted as the output. In this type of neural network, multiple labels associated with a molecule, such as several types of ADMET properties, can be predicted simultaneously. It is well known that multitask prediction affords more improvement in performance than individual single-task prediction [73].

Figure 1.6 shows an example of a multimodal neural network taking a graph representing a compound and a sequence representing a protein as inputs. In addition to the information derived from a molecular structure, information from other modalities can also be used as input. An example of the prediction of activity using compound- and protein-related information is described in detail in the experiment section.

The kGCN system supports the operations described above and other additional operations for building neural networks. These operations are implemented using TensorFlow[74] and are compatible with Keras[75], allowing users to construct neural networks such as convolutional neural networks and recurrent neural networks [53] with Keras operations.

These neural networks are characterized by hyperparameters such as the number of layers in the model and the number of dimensions of each layer. To determine these hyperparameters, the kGCN system includes a Bayesian optimization functionality.

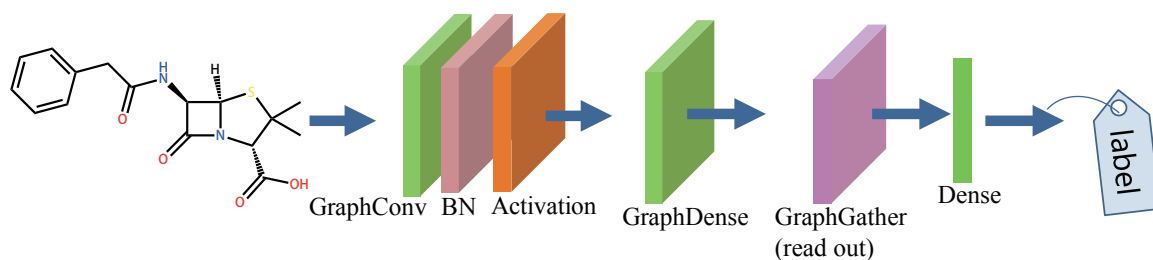


Figure 1.4: GCN for a prediction task with a compound as input.

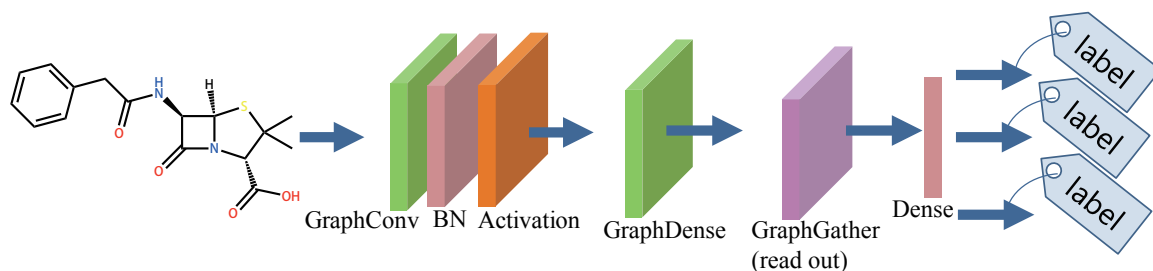


Figure 1.5: Multitask GCN with a compound as input.

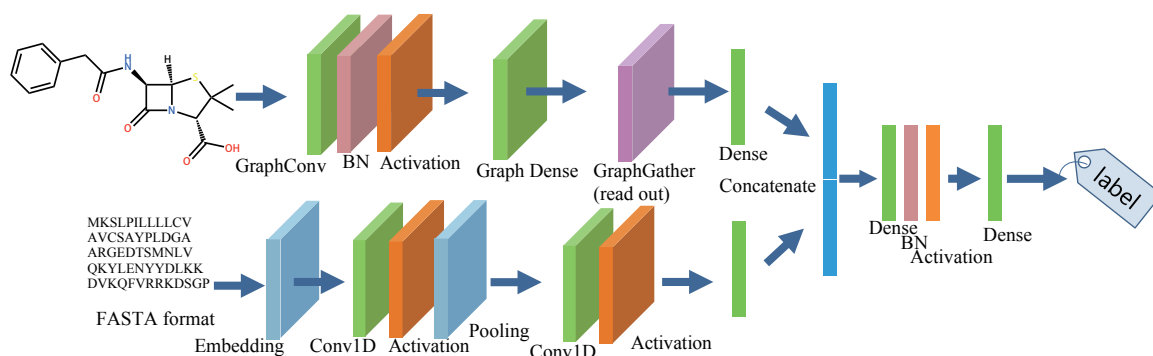


Figure 1.6: Multimodal GCN with a compound and a sequence as inputs.

### 2.3 Visualization of GCN

To confirm the features of molecules that influence the prediction results, a visualization system using the IG method[62] has been developed. After the construction of the prediction model, it is possible to visualize the importance of each atom in a molecular structure based on the IG value  $\mathcal{I}(x)$  derived from the prediction model.

The IG value  $\mathcal{I}(x)$  is defined as follows:

$$\mathcal{I}(x) = \frac{x}{M} \sum_{k=1}^M \nabla S\left(\frac{k}{M}x\right), \quad (1.6)$$

where  $x$  is the input, which is an atom of a molecule;  $M$  is the number of divisions of the input;  $S(x)$  is the prediction score, i.e., the output of the neural network with input  $x$ ; and  $\nabla S(x)$  is the gradient of  $S(x)$  with respect to the input  $x$ . In the default settings,  $M$  is set to 100. The importance of an atom is defined as the sum of the IG values of the features of that atom. The calculation of atom importance is performed on a compound-by-compound basis.

The evaluation of the visualization results depends on each case. Although methods for the visualization of DL results are still under development, their effectiveness in solving common problems has not been reported. Hence, I address a quantitative evaluation of the IG values in the context of retrosynthetic reaction prediction in Chapter 2.

## 2.4 Hyperparameter optimization

For the optimization of a neural network model, hyperparameters such as the number of graph convolution layers, the number of dense layers, the dropout rate, and the learning rate should be determined. As it is difficult to manually determine all these hyperparameters, kGCN enables automatic hyperparameter optimization by means of Gaussian-process-based Bayesian optimization using a Python library, GPyOpt [76].

## 2.5 Interfaces

This section describes the three interfaces of the kGCN system.

### Command-line interface

The kGCN system provides a command-line interface suitable for batch execution. Data processing is designed according to specific aims, but there is a standard procedure common to many data processing designs, e.g., a series of processes for cross-validation. The kGCN commands include these common processes, i.e., the kGCN system allows preprocessing, learning, prediction, cross-validation, and Bayesian optimization to be performed using the following commands:

**kgen-chem command** allows the preprocessing of molecular data, e.g., in the structure-data file (SDF) or Simplified Molecular Input Line Entry System (SMILES) format.

**kgcn command** allows batch execution related to prediction tasks: supervised training, prediction, cross-validation, and visualization.

**kgcn-opt command** allows batch execution related to hyperparameter optimization.

These commands can be used with Linux commands and enable users to construct automatic scripts, e.g., Bourne again shell (Bash) scripts. Because such batch execution is suitable for large-scale experiments using workstations and reproducible experiments, this interface is useful for the evaluation of neural network models.

### **KNIME interface**

The kGCN system supports KNIME modules as a GUI. KNIME is a platform for preparing workflows consisting of KNIME nodes for data processing and is particularly useful in the field of data science. The kGCN KNIME nodes described below are useful for the execution of various kGCN functions in combination with existing KNIME nodes. The command-line interface allows batch execution, whereas the KNIME interface is suitable for early steps of the ML process, such as prototyping and data preparation.

For model training and evaluation, kGCN provides the following two nodes.

**GCNLearner** trains a model on a given dataset. This node receives the training dataset and produces the trained model as its output. Detailed settings such as the batch size and learning rate can be set as the node properties.

**GCNPredictor** predicts labels from a given trained model and a new dataset.

Using the kGCN nodes mentioned above, Fig. 1.7 shows two examples of workflows. Each data flow can be separated into the flows before and after GCNLearner. The purpose of the former part is data preparation, for which kGCN includes the following KNIME nodes:

**CSVLabelExtractor** reads labels from a comma-separated values (CSV) file for training and evaluation.

**SDFReader** reads the molecular information from an SDF.

**GraphExtractor** extracts the graph for each molecule.

**AtomFeatureExtractor** extracts the features of each molecule.

**GCNDataSetBuilder** constructs the complete dataset by combining the input and label data.

**GCNDataSetSplitter** splits the dataset into training and test datasets.

The test dataset is used for the evaluation and interpretation of the results. kGCN also provides modules for displaying the output results.

**GCNScore** provides the scores of the prediction model, such as accuracy.

**GCNScoreViewer** displays a graph of the receiver operating characteristic (ROC) scores as an image file.

**GCNVisualizer** computes the IG values and atom importance.

**GCNGraphViewer** displays the atom importance as an image file.

Another example of a workflow, this one involving multimodal neural networks, is shown in Fig. 1.8. For the design of multi-modal neural networks, the kGCN system provides the following modules:

**AdditionalModalityPreprocessor** reads data of an additional modality from a given file.

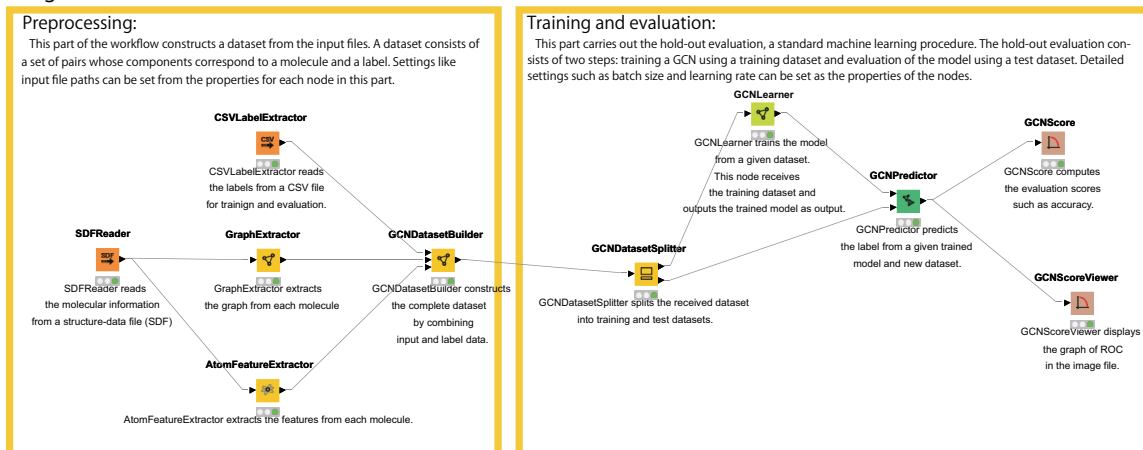
**AddModality** adds data of an additional modality to the dataset.

To change from a single-task workflow to a multimodal workflow, the AddModality node should be added next to the GCNDataSetBuilder node.

The visualization process shown on the bottom right in Fig. 1.8 requires a specific computation time depending on the number of molecules to be visualized, as the computation time of the IG method for each molecule is 1-5 sec during GPU execution. To reduce the size of the dataset, GCNDataSetSplitter can be used to select only part of the dataset.



## Single-task workflow



## Multi-task workflow

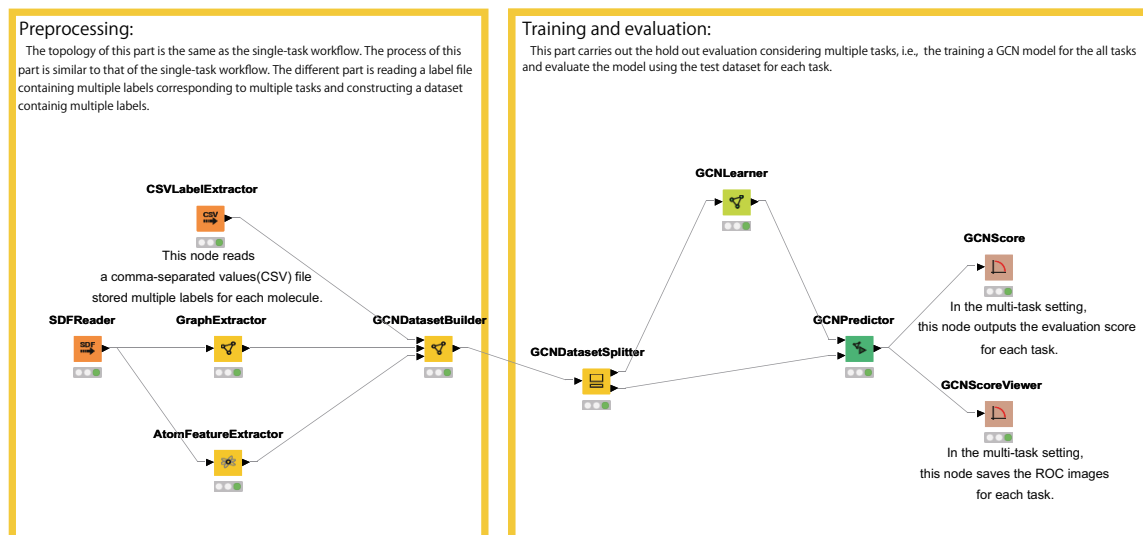


Figure 1.7: Single-task workflow for the hold-out procedure using the KNIME interface (top). Multitask workflow for the hold-out procedure (bottom).

## Multi-modal workflow

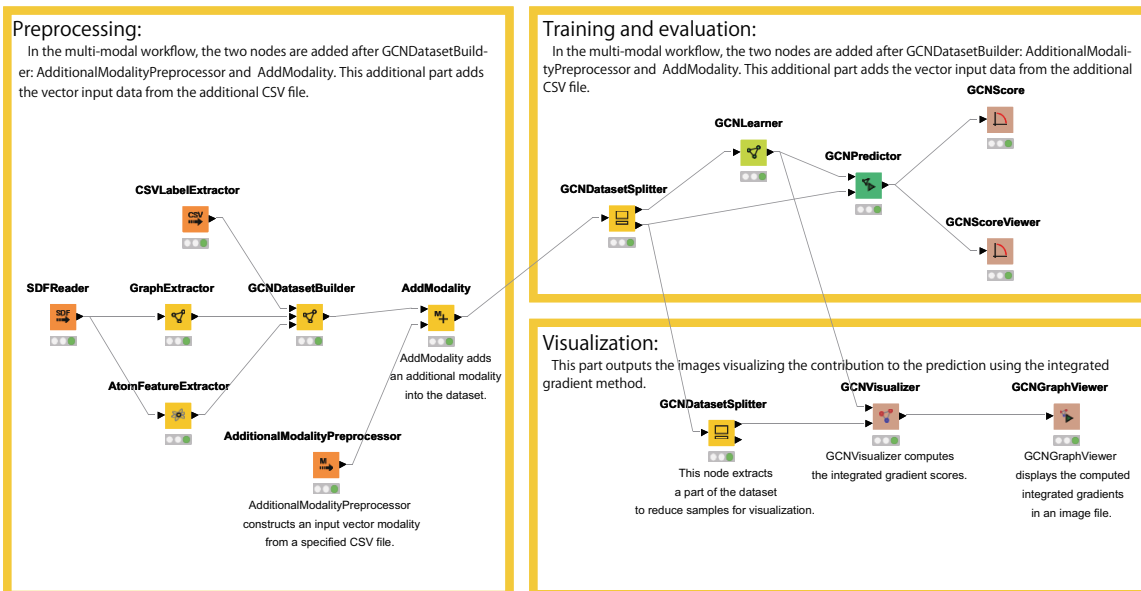


Figure 1.8: Multimodal workflow for the hold-out procedure.

## Python interface

The kGCN system also provides a Python library to allow programmers to more precisely tune the settings for analysis. The kGCN system can be used in a manner similar to any standard library and supports pip, a standard Python package manager. Furthermore, the kGCN system can be used in the Jupyter notebook, which is an interactive interface. Therefore, users can easily explore this library using Google Collaboratory, a cloud environment for the execution of Python programs.

The kGCN system adopts an interface similar to that of scikit-learn, a de facto standard ML library in Python. Accordingly, the procedure for employing the kGCN library includes preprocessing, training by *fit* methods, and evaluation with *pred* method, in this order. Users can easily access the kGCN library in a manner similar to that of scikit-learn. Furthermore, designing a neural network, which is necessary for using kGCN, is easy if users are familiar with Keras because kGCN is fully compatible with the Keras library, and users can easily design a neural network in the same way as with Keras.

To demonstrate the broad applicability of the presented framework, three sample programs comprising datasets and scripts using the standard functions of kGCN are available on the web site for the framework.

## Flexible user interfaces

As described in the introduction and implementation sections, kGCN provides a KNIME GUI, a command-line interface, and a programming interface to support various types of users with various skill levels. For example, the easy-to-use high-layer GUI can assist chemists with limited programming knowledge in using kGCN and understanding structure-activity relationships at the molecular level. By contrast, it is expected that ML professionals with good programming skills will focus on the improvement of algorithms using the low-layer Python interface. By using the Python interface, users can make ML procedures more flexible and incorporate kGCN functions into user-specific programs such as web services. Users with good programming skills can also use the command-line interface to automate data analysis procedures using kGCN functions because this interface makes it easy to construct a pipeline in combination with other commands, such as Linux commands.

### 3 Results

Regarding the applications of kGCN, this section describes the prediction of assay results for a protein based on molecular structure. The prediction of compound-protein interaction (CPI) has played an important role in drug discovery [77], and CPI prediction methods using DL have achieved excellent results [43, 54, 55, 56]. In this study, the applicability of kGCN for CPI prediction is demonstrated as examples of single-task, multitask, and multimodal GCNs using matrix metalloproteinase (MMP) inhibition assay data. A single-task GCN predicts the activity against a protein based on a chemical structure represented as a graph. A multitask GCN predicts the activities against multiple proteins based on a chemical structure. Whereas single-task and multitask GCNs do not use information related to the proteins themselves, multimodal neural networks predict activity based on information on both protein sequences and chemical structures.

For this examination, a dataset was prepared from the ChEMBL ver. 20 database. The threshold for classification as active or inactive was defined as  $30 \mu\text{M}$ . This dataset consists of results from four types of MMP inhibition assays: MMP-3, MMP-9, MMP-12, and MMP-13. The number of compounds for each assay is listed in Table 1.1. These MMPs were selected because relatively large amounts of data are available for them in the ChEMBL dataset [78].

Table 1.1: Numbers of compounds in MMP inhibition assay dataset

Assay type	Number of compounds
MMP-3	2095
MMP-9	2829
MMP-12	533
MMP-13	2607

kGCN provides many types of descriptors for compounds and proteins. For example, kGCN allows graph representations for GCNs and vector representations, such as extended-connectivity fingerprints (ECFP) [22] and DRAGON [79], for standard neural networks. Additionally, to represent proteins, kGCN uses amino acid sequences and vector representations such as PROFEAT descriptors [80]. This application uses a graph representation for a compound and a sequence representation for a protein by default.

To simplify the experiment, molecules with more than 50 atoms were excluded. Because the dataset was unbalanced, negative data corresponding to inactivity were selected in the

same manner [54]. Negative data were then generated to equalize the numbers of negative and positive data entries for each assay. Such preprocessing can be realized using the *kgc-chem* command introduced in the section describing the command-line interface.

Figure 1.9 shows the areas under the ROC curves (ROC-AUCs) for five-fold cross-validation. The results show that the multimodal approach outperforms the other approaches. The reason why prediction with the multimodal approach yields a better ROC-AUC is speculated to be the use of the sequence-related information of the target proteins in addition to the graph representations of the compounds. These results suggested that the multimodal model learned more general features associated with the CPI by extracting common information between molecules and proteins, compared with the single-task and multitask models that can use only molecule information. These results are consistent with previously reported results indicating that the inclusion of sequence descriptors contribute to improving accuracy [43, 54, 55, 56].

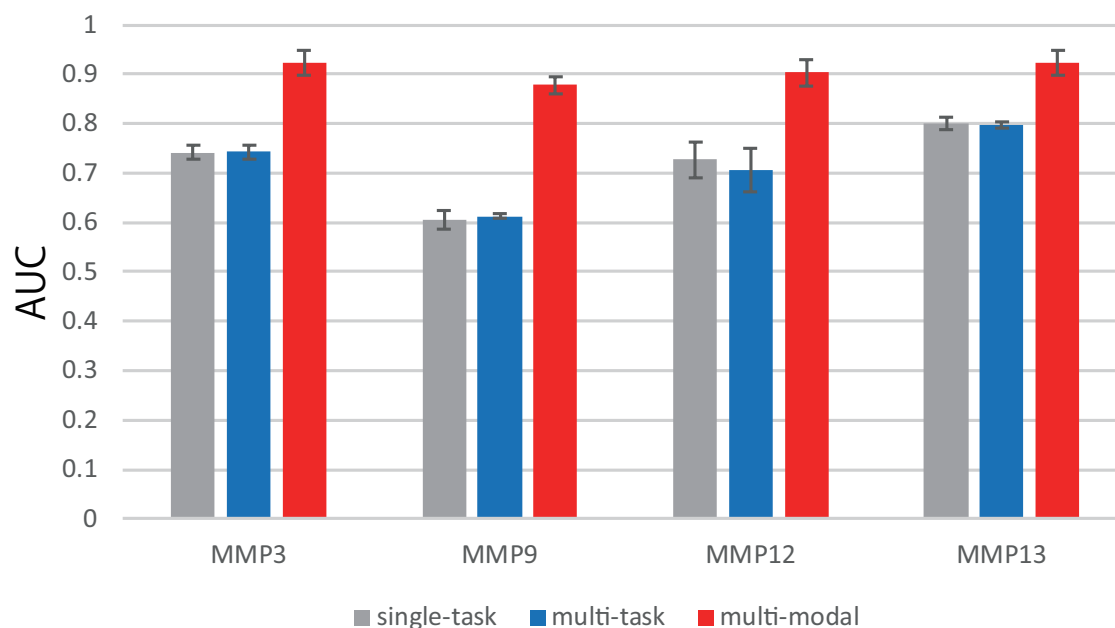


Figure 1.9: ROC-AUCs obtained from five-fold cross-validation.

kGCN enables the visualization of the atomic contributions to the prediction results, as shown in Fig. 1.10b. The compound N-hydroxy-2-[N-(propan-2-yloxy)[1,1'-biphenyl]-4-sulfonamido]acetamide (Fig. 1.10a) was used for this prediction, and its reported activity against MMP-9 is 200 nM (IC<sub>50</sub>) [81]. The label of this compound for MMP-9 in the dataset is active, and the activity predicted for this compound in single-task mode is correct (the probability of the active label is 0.964). This compound possesses a hydroxamic acid

group ( $-\text{C}(=\text{O})\text{NHOH}$ ), and it is well known that many MMP inhibitors have a hydroxamic group. The crystallographic structure of a complex of MMP-9 and this compound has been previously reported [82]. MMP-9 is a zinc protease, and the hydroxamic acid group of the above compound is coordinated to the zinc ion of MMP-9. The positive contributions of the OH, NH, and C=O of the hydroxamic acid group shown in Fig. 1.10b are consistent with the interaction of the hydroxamic group with the zinc of MMP-9.

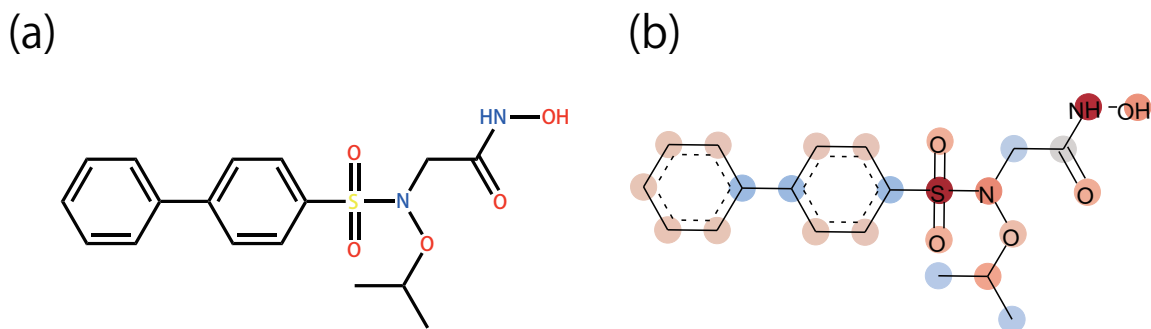


Figure 1.10: (a) Chemical structure. (b) Atomic contributions to the predicted MMP-9 activity. Red shading indicates positive contributions to the prediction results (MMP-9 active in this case). Blue shading indicates negative contributions (not active).

Such a visualization can be used to confirm the validity of a prediction by comparing the atomic contributions to the prediction with the structure-activity and/or structure-property relationships. Additionally, such visualizations can be useful in drug design for improving the activity, physicochemical properties and/or ADMET properties of a drug candidate by modifying the chemical moieties that contribute negatively to the corresponding prediction.

## 4 Conclusion

An open-source GCN tool named kGCN that is designed to assist various types of users, including chemists and cheminformaticians, is described. To support users with various levels of programming skill, kGCN provides three interfaces: a GUI using the KNIME platform for users with limited programming skills, such as chemists, and command-line and Python library interfaces for advanced users, such as cheminformaticians and data scientists. A three-step procedure consisting of preprocessing, model tuning, and interpretation of results is required for the construction of a prediction model and the utilization of its prediction results. kGCN supports all three steps by means of functions such as the automatic preparation of graph representations based on chemical structures for preprocessing, Bayesian optimization for the automatic optimization of the hyperparameters

of neural networks for model tuning, and the use of the IG method to visualize the atomic contributions to the prediction results for interpretation. Regarding the approaches used for prediction, kGCN supports single-task, multitask, and multimodal predictions. CPI prediction based on inhibition assays for four MMPs, namely, MMP-3, MMP-9, MMP-12, and MMP-13, is presented as a representative case study using kGCN. In this case study, multimodal prediction shows higher accuracy than single-task or multitask prediction. Additionally, a visualization of the atomic contributions to the prediction results indicates that the hydroxamate group of the compound exhibits a positive contribution to the activity, which is consistent with the known structure-activity relationships. Such visualizations are useful for the validation of models and the design of new molecules based on a given model. This visualization capability also allows the realization of “explainable AI” for understanding the factors influencing the results of AI prediction tools, which are typically black boxes.

kGCN is available at <https://github.com/clinfo/kGCN>. Various examples, such as Jupyter notebooks, are also provided. Future work will address support for new methods based on GNNs because GNNs are a popular focus of related research at present, and hence, new related methods, such as graph attention and pooling, are actively being developed. I will proactively adopt the latest methods and continue to develop kGCN to enable diverse users to easily apply such methods to appropriately analyze their data and understand the reasons underlying the corresponding predictions. Additionally, I will gather user feedback to improve kGCN in terms of usability.

In the next chapter, I describe the application of kGCN for the task of retrosynthetic reaction prediction.





## Chapter 2

# Interpretable Retrosynthetic Reaction Prediction Using Graph Convolutional Networks

## 1 Introduction

One-step retrosynthetic reaction prediction is an essential step of implementing retrosynthetic analysis in data-driven approaches. Retrosynthetic reaction prediction in data-driven approaches corresponds to the reaction rules in rule-based approaches and must be performed repeatedly for retrosynthesis. Hence, the quality of retrosynthetic reaction prediction strongly influences the quality of a synthetic route designed via data-driven approaches because the prediction errors associated with multiple predictions will accumulate throughout the analysis process. To date, investigations of DL-based approaches to retrosynthetic reaction prediction have shown that such approaches can achieve excellent performance[45, 83, 84, 85, 86, 87, 88, 89].

However, there is a gap between the process of retrosynthetic reaction prediction based on DL and chemist's basic knowledge of reaction mechanisms. When chemists design a proposed synthetic route, they consider not only the local structures of a molecule but also its overall structure because chemical reactions may be affected by atoms and functional groups that appear to be irrelevant to the reactive center of the molecule[4]. Therefore, due to the lack of complete chemical structural information, molecular fingerprints such as extended-connectivity fingerprint (ECFP)[22, 90] are considered inadequate as input features for reaction prediction. However, for handling molecular structures in machine learning (ML) or DL, ECFP is still used because it is handy and effective[45, 83, 91].

In addition, the black-box problem often arises[92] with DL methods, which tend to

achieved higher prediction accuracy than conventional ML methods. The black-box problem causes difficulty in the interpretability of the underlying drivers of predictions. Hence, the black-box problem can make predictions generated by means of DL less acceptable to chemists, whereas it is easy to explain why a certain reaction is selected via the rule-based approach. Despite these problems, data-driven approaches have shown comparable performance to rule-based approaches[16, 18]. Thus, to make data-driven approaches more accessible to chemists, it is essential to solving the above problems. To this end, this chapter aims to address two issues: (1) improving the performance of retrosynthetic reaction prediction and (2) developing an interpretable visualization system to resolve the black-box problem.

In this chapter, I propose a new framework for retrosynthetic reaction prediction based on graph convolutional network (GCN)[70], using the Integrated Gradients (IG) method[93] for visualization, to address the above two problems. It has been reported that GCN is a state-of-the-art method that treats a molecule as a graph structure in various tasks[71, 94]. A GCN can offer a solution to the problems that arise with fingerprints because a GCN uses information on the whole molecular structure for prediction. To overcome the black-box problem of DL prediction, methods of visualizing the feature contributions for each sample can be applied[93, 95, 96, 97]. IG is an architecture-free visualization method, i.e., this method can be applied to any differentiable neural network, including a GCN, without affecting the neural network performance. However, IG has been proposed and evaluated mainly in the context of image recognition [93], and no study has yet been conducted on the use of IG to quantitatively analyze chemical properties. In addition, in studies on retrosynthetic reaction prediction[45, 83, 91], no comparison of the performance of GCN and ECFP for retrosynthetic reaction prediction, i.e., which molecular representation is better suited for retrosynthetic reaction prediction, has yet been reported.

In this study, I demonstrated the effectiveness of my framework combining a GCN and IG using a United States patent dataset[98] that has been employed in many studies on data-driven approaches[83, 91, 99]. Following previous studies[45, 83], I extracted reaction templates from the patent dataset. I trained a GCN model and an ECFP model to predict the reaction template for a given molecule. I found that the GCN model showed better performance in retrosynthetic reaction prediction toward retrosynthetic analysis than the ECFP model did. I also demonstrated that IG-based visualizations of the GCN predictions successfully highlighted reaction-related atoms. Furthermore, the visualizations of the GCN predictions quantitatively showed the contributions of the reaction-related atoms to the prediction results. My implementations are available on GitHub at <https://github.com/clinfo/kGCN> and [https://github.com/clinfo/extract\\_reaction\\_](https://github.com/clinfo/extract_reaction_)

template. My method will contribute to retrosynthetic analysis based on data-driven approaches.

## 2 Methods

Figure 2.1 shows the workflow of the developed framework based on the GCN and IG. Detailed procedures for constructing the framework are described in the following sections.

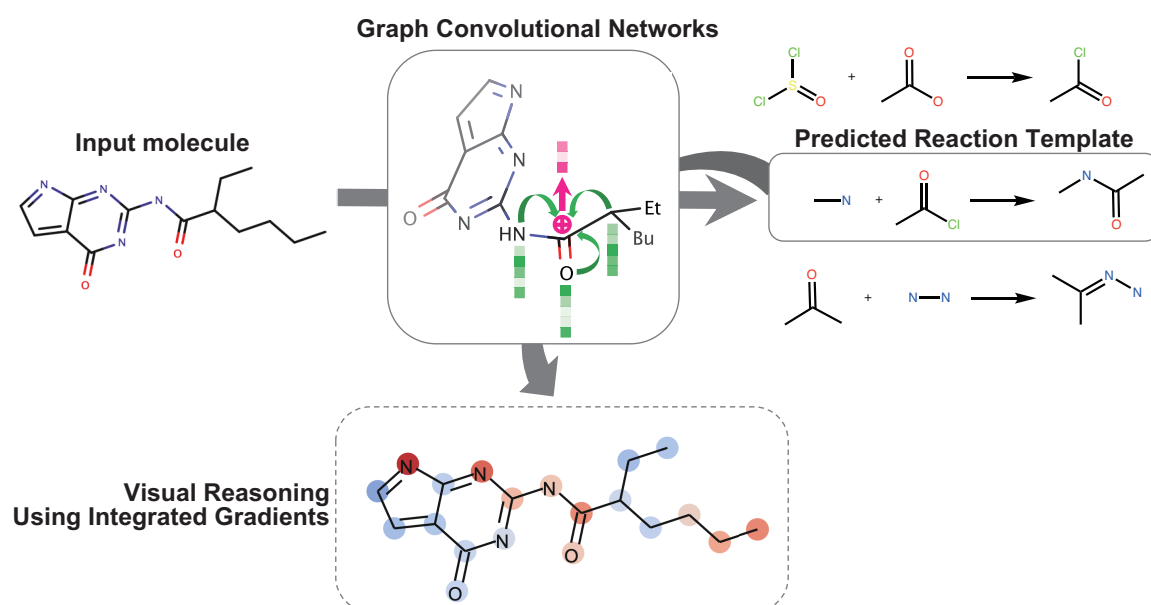


Figure 2.1: Workflow of the framework based on the GCN and IG in a retrosynthetic reaction prediction.

### 2.1 Dataset

I define a reaction template as consisting of a reactive center, orphan atoms, and their first-degree neighbors in a reaction, as shown in Fig. 2.2. An orphan atom is one that appears only on one side of the reaction arrow in ChemAxon[100]. I refer to a product in a reaction template as a reaction center. To create a dataset of reaction templates, I employed a set of 1,808,937 reactions from United States Patent and Trademark Office (USPTO) published between 1976 and September 2016, prepared by Lowe[98]. The reactions are stored in the form of reaction SMILES: reactants'>'agents'>'products, where the reactants, agents, and

products are separated by '>', and the agents include solvents, reagents, and catalysts. This reaction set contains many duplicate reactions, and the solvents and chemical agents are inconsistently recorded. Therefore, I used only the reactants and products in the reactions for simplicity. As part of my procedure, I removed duplicates (resulting in 1,105,130 reactions), reduced all reactions to their reactants and products, and retained only reactions with a product. In detail, I filtered the reactions by performing two steps. First, I removed agents defined by the reaction SMILES syntax. Then, I removed salts from the reactions because the presence of salts would influence the condition of retaining only reactions with a product. I defined salts in reference to ChemAxon's default salts (Table 2.1). After the refinement of the reaction set, a total of 1,072,175 reactions remained, and I extracted reaction templates from those reactions using Automapper in the ChemAxon API[100]. Following previous studies[16, 83], I used 1,752 unique templates occurring at least 50 times among the 1,072,175 extracted reaction templates and 371,003 molecules with a corresponding correct template among the unique templates as the input dataset.

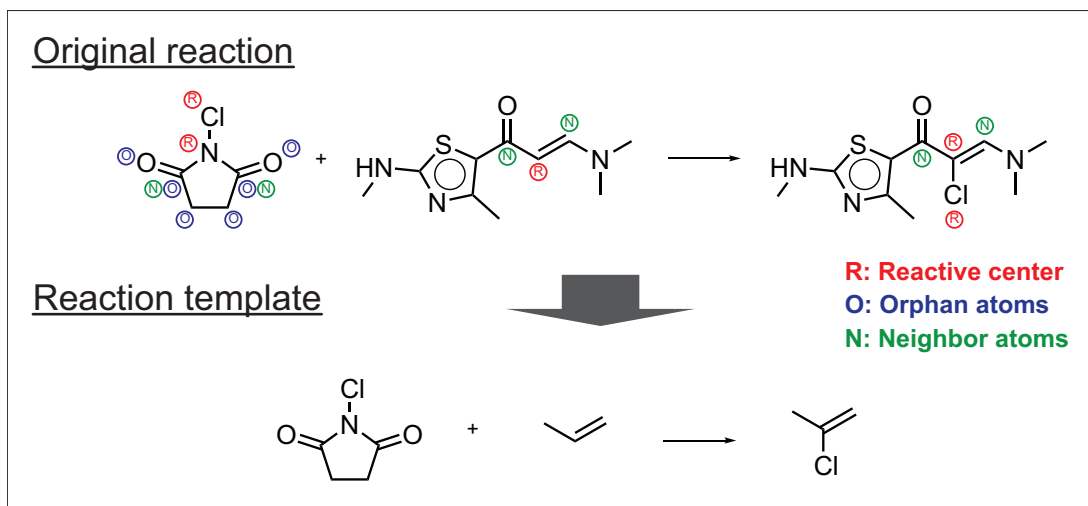


Figure 2.2: Example of a reaction template extracted from the original reaction.

### Graph representation of molecules for GCN

A molecule is formalized as a tuple  $\mathcal{M} \equiv (V, E, F)$ , where  $V$  is a set of nodes. Each node represents an atom in the molecule. Each node has features  $f_i \in F$  ( $i \in V$ ), where  $F$  is a set of feature vectors representing the properties of an atom. Here, I employed the features used in DeepChem[65] (Table 2.2).  $E$  is a set of edges. Each edge  $e \in E$  represents a bond between atoms, i.e.,  $e \in V \times V \times T$ , where  $T$  is a set of bond types. I used an adjacency matrix  $\mathbf{A}^{(t)}$  of the form defined in Eq. (1.1). With this matrix, a molecule is represented

by  $\mathcal{M}' = (\mathbf{A}, F)$ , where  $\mathbf{A} = \{\mathbf{A}^{(t)} | t \in T\}$ . Specifically, I used RDKit[69] to create the adjacency matrix and the feature matrix and employed  $\mathcal{M}'$  as the input to a GCN.

Table 2.1: List of removed salts

<b>List of removed salts</b>		
(2E)-but-2-enedioic acid	butanedioic acid	methanesulfonic acid
(2Z)-but-2-enedioic acid	cesium(1+) ion	nitric acid
2,3-dihydroxy-2,3-bis (4-methylbenzoyl) butanedioic acid	chloride	oxalate
2,3-dihydroxybutanedioic acid	cyclohexanamine	oxalic acid
2-(dimethylamino)ethan-1-ol	fluoride	perchlorate
2-hydroxypropane-1,2,3-tricarboxylic acid	formic acid	phosphoric acid
4-methylbenzene-1-sulfonic acid	hydrogen bromide	potassium(1+) ion
acetic acid	hydrogen chloride	sodium(1+) ion
ammonia	hydrogen fluoride	sulfuric acid
aluminium(1+) ion	hydrogen iodide	triethylamine
aluminium(3+) ion	iodide	trifluoroacetate
barium(2+) ion	lithium(1+) ion	trifluoroacetic acid
bromide	magnesium(1+) ion	zinc(1+) ion
butanedioic acid	magnesium(2+) ion	

Table 2.2: List of atomic features

<b>RDKit atom class method</b>	<b>Description</b>	<b>Possible values</b>	<b>Dimension</b>
GetSymbol()	Returns the atomic symbol.	C,N,O,S,F,Si,P,Cl,Br,Mg,Na,Ca,Fe,As,Al,I,B, V,K,Tl,Yb,Sb,Sn,Ag,Pd,Co,Se,Ti,Zn,H,Li,Ge, Cu,Au,Ni,Cd,In,Mn,Zr,Cr,Pt,Hg,Pb,Unknown	44
GetDegree()	Returns the degree of the atom, which is defined to be its number of directly-bonded neighbors.	0,1,2,3,4,5,6,7,8,9,10	11
GetImplicitValence()	Returns the number of implicit hydrogens on the atom.	0,1,2,3,4,5,6	7
GetFormalCharge()	Returns the atom's formal charge.	Formal charge value	1
GetNumRadicalElectrons()	Returns the number of radical electrons on the atom.	Number of radical electrons	1
GetHybridization()	Returns the atom's hybridization. Returns RDKit's aromatic flag.	SP,SP2,SP3,SP3D,SP3D2	5
GetIsAromatic()	'1' indicates the atom is aromatic and '0' indicates not.	0,1	1
GetTotalNumHs()	Returns the total number of hydrogens on the atom.	0,1,2,3,4	5

## Representation of molecules in terms of ECFP

ECFP is a circular topological fingerprint for molecular characterization[22] and is commonly used in a wide variety of studies. In this study, I set the maximum diameter to four and prepared ECFPs with different bit lengths of 2,048, 4,096, and 8,192 bits. I used the ChemAxon API for calculating ECFPs.

## 2.2 Retrosynthetic reaction prediction

I aimed to predict the reaction templates from corresponding molecules (products) correctly. In this study, the retrosynthetic reaction prediction task was defined as a multiclass classification problem. I built two models for comparison, namely, a model taking graph representations of molecules as input (GCN model) and a model taking ECFPs as input (ECFP model); see Fig. 2.3. To evaluate the prediction performance, a balanced accuracy[101] and an accuracy for each reaction template were used as evaluation metrics, and a five-fold cross-validation was performed. The balanced accuracy is the average of recall scores per class.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.1)$$

$$\text{balanced accuracy} = \frac{1}{C} \sum_{i=1}^C \text{recall}_i \quad (2.2)$$

Here, TP and FN represent the number of true positives and false negatives, respectively,  $C$  denotes the number of classes, and  $\text{recall}_i$  denotes a recall score of a class. Accuracy for each reaction template is the ratio of the number of correct predictions to total predictions per reaction template. In the cross-validation, the dataset was split into three sets in each fold: 65% of the dataset was used for training, 15% for validation, and 20% for testing.

## 2.3 GCN and ECFP Models

### GCN model

In this study, I used graph convolution layers (Eq. (1.3)), graph dense layer (Eq. (1.4)), and a graph gather layer (Eq. (1.5)) to construct my GCN model. The GCN model was a neural network consisting of three graph convolution layers with batch normalization[72] and rectified linear unit (ReLU) activation, a graph dense layer with ReLU activation, a

graph gather layer and a dense layer with softmax activation. Each layer contained 128 units. I set the hyperparameters as follows: epochs = 100, batch size = 128, and learning rate = 0.0001. In addition, I adopted early stopping with a patience of three. To implement this model, TensorFlow[74] was used.

### ECFP model

The ECFP model was a neural network designed in following previous studies[16, 45], consisting of a dense layer with exponential linear unit (ELU) activation and five highway network layers with ELU activation. The dense layer contained 512 units with a dropout ratio of 0.3. The highway network layers each contained 512 units with a dropout ratio of 0.1. I set the hyperparameters as follows: epochs = 1,000, batch size = 128, and learning rate = 0.001. In addition, I adopted early stopping with a patience of three. To implement this model, Keras[75] was employed.

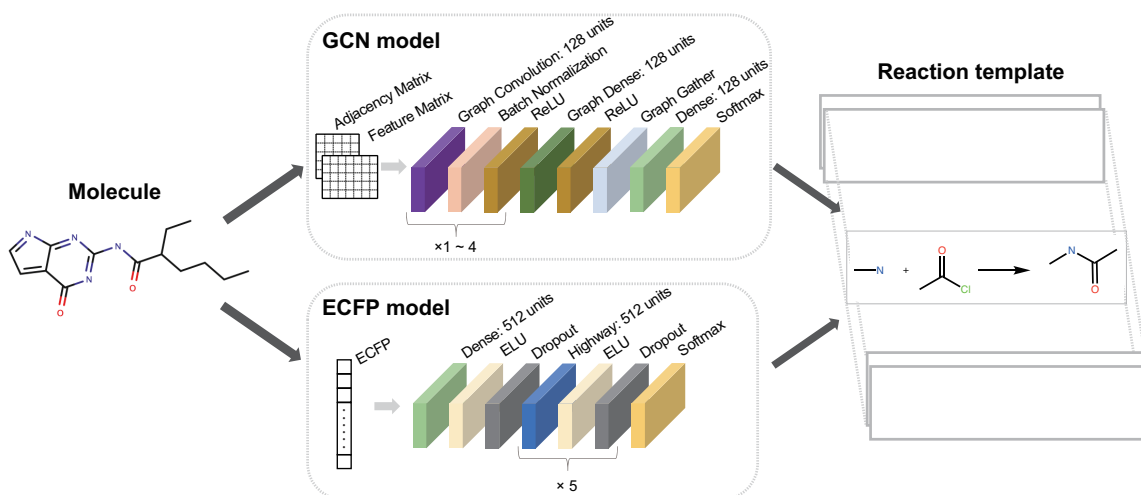


Figure 2.3: Overview of the GCN and ECFP models.

## 2.4 Visualization

To confirm which features of the molecules influenced the prediction results, I developed a visualization system using the IG method[93]. Once I had trained a model for retrosynthetic reaction prediction, this system allowed us to visualize the attributes of the prediction results with respect to the molecular structures. I also quantitatively evaluated the IG values of 10,000 molecules that were correctly predicted.



## Integrated Gradients (IG)

I used the IG values  $I$  as defined in Eq. (1.6). I defined the importance of an atom as the sum of the IG values of its atomic features and calculated the IG values for each reaction template individually.

### Quantitative evaluation of IG-based visualization

To quantify the visualization results obtained using the IG method, I calculated the average of IG value of the atomic features in a reaction center. If multiple reaction centers existed in a molecule, I chose the reaction center with the highest average IG value among them. I then compared the average IG value of the reaction center to the IG values of all atoms in the molecule by means of a histogram. To facilitate these comparison, I standardized the IG values in each molecule and plotted Gaussian kernel density estimates of each histogram.

## 3 Results

### 3.1 Retrosynthetic reaction prediction

The GCN model showed better performance than the ECFP model in terms of balanced accuracy, as shown in Fig. 2.4. For determining the top- $n$  balanced accuracy, we regard a prediction that contains the correct reaction template among the top- $n$  reaction templates in terms of the softmax probability as a correct prediction. The best GCN model was a model with three convolutional layers; its top-one balanced accuracy was 0.249, its top-10 balanced accuracy was 0.510, and its top-30 balanced accuracy was 0.662. The best ECFP model was a model with 2,048 dimensions; its top-one balanced accuracy was 0.217, its top-10 balanced accuracy was 0.473, and its top-30 balanced accuracy was 0.642. Table 2.3 shows detailed results of the retrosynthetic reaction prediction with the GCN and ECFP models.

Table 2.3: Top- $n$  balanced accuracies of the GCN and ECFP models

Descriptor	ECFP			GCN			
	2048 dim	4096 dim	8192 dim	1 conv layer	2 conv layers	3 conv layers	4 conv layers
Top-1 balanced accuracy	0.217	0.205	0.192	0.192	0.237	0.249	0.128
Top-10 balanced accuracy	0.473	0.464	0.451	0.421	0.489	0.510	0.347
Top-30 balanced accuracy	0.642	0.634	0.623	0.569	0.641	0.662	0.496

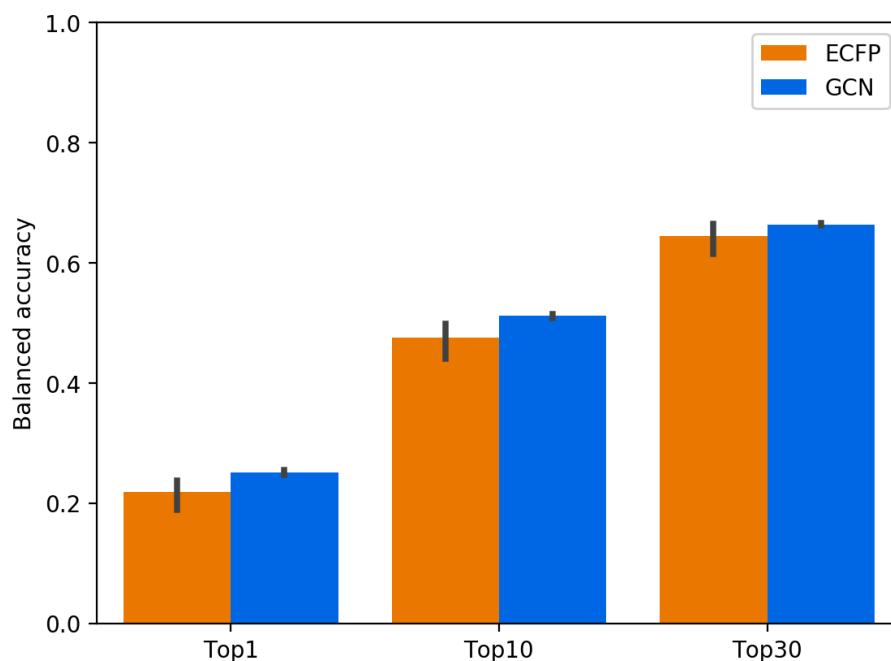
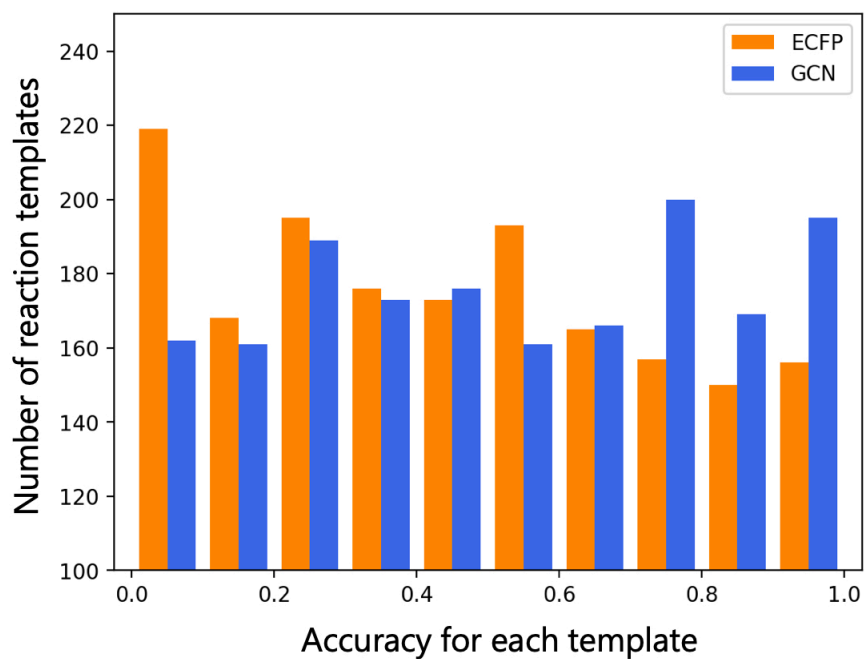
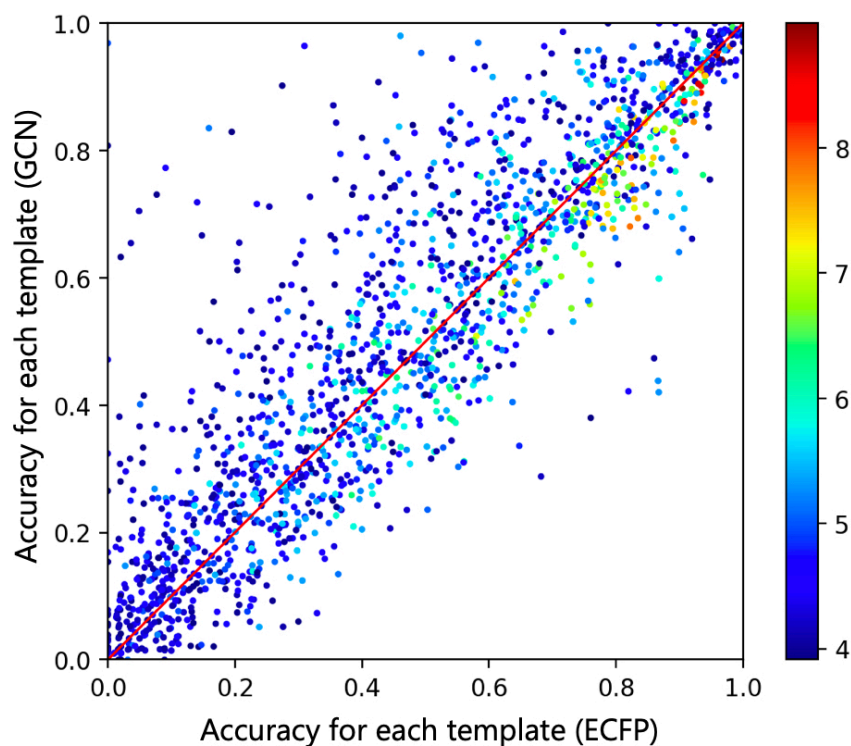


Figure 2.4: Comparison of the balanced accuracies of the GCN (blue) and ECFP (orange) models.

To elucidate the differences between the GCN and ECFP models, we show the detailed prediction results in Fig. 2.5. We compare the top-10 accuracies for each reaction template achieved with the best GCN and ECFP models. Figure 2.5 presents the differences in accuracy between the GCN model and the ECFP model. We also show corresponding results for the top-one and top-30 cases in Fig. 2.6.



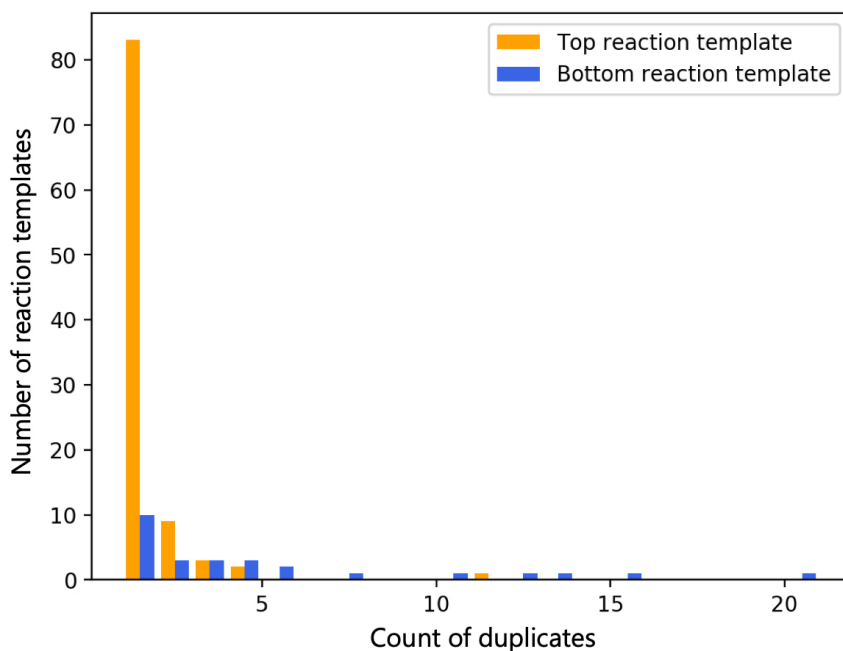
(a): Histograms of the accuracies for each template achieved by the GCN (blue) and ECFP (orange) models.



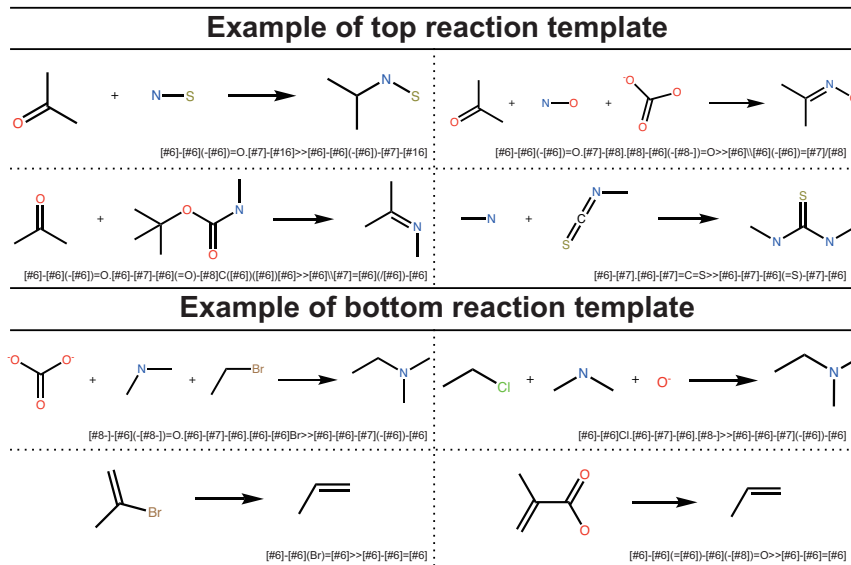
(b): Scatter plot of the accuracies, where the color corresponds to the logarithm of the number of molecules in which the template appears, as indicated by the color bar on the right.

Figure 2.5: Comparison between the distributions of the top-10 accuracies for each template achieved by the GCN and ECFP models.

### 3. RESULTS



(c): Histograms of the counts of duplicated reaction templates appearing as the top reaction template (orange) and the bottom reaction template (blue).



(d): Examples of top and bottom reaction templates. Below the template drawings, the reaction templates are also expressed in the SMARTS format.

Figure 2.5: Comparison between the distributions of the top-10 accuracies for each template achieved by the GCN and ECFP models.

Figure 2.5a shows that the GCN model yielded more accurate predictions than the ECFP model, with an accuracy ranging from 0.7 to 1.0 for a larger number of templates and lower accuracy for fewer templates. Figure 2.5b shows a scatter plot of the accuracy achieved for each template by the GCN model versus that achieved by the ECFP model. To clarify the effect of the number of molecules in which a reaction template appears on the prediction results, we have also added color information reflecting the number of molecules on a logarithmic scale. We can see from this figure that both the GCN and ECFP models tended to generate more accurate predictions for templates associated with a large number of molecules, whereas the prediction performance tended to be poorer for a small number of molecules per reaction template. In the dataset we used, the frequencies of occurrence of different reaction templates are quite different, as shown in Fig.2.7. When a dataset is biased, conventional ML methods will tend to mainly learn the characteristics of classes containing many training data. We can see in Fig. 2.5b that the ECFP model also showed this tendency, achieving more accurate predictions for reaction templates associated with many training data, more so than the GCN model. In addition, Figure 2.5b shows that the GCN model offered more accurate predictions for reaction templates that were not predicted accurately by the ECFP model. To clearly show the differences between the GCN and ECFP model predictions, we present the distributions of the top-10 accuracies for the top-100 and bottom-100 reaction templates (ranked by the frequency of occurrence of each reaction template) achieved by the GCN and ECFP models in Fig. 2.8.

Additionally, to clarify which reaction templates the GCN and ECFP models could and could not predict with high accuracy, we selected the top and bottom reaction templates and counted duplicated product structures. We defined the top reaction templates as the 129 templates predicted with more than 90% accuracy and the bottom reaction templates as the 125 templates predicted with less than 10% accuracy. Figure 2.5c shows that various unique product structures appear among the top reaction templates, i.e., the corresponding prediction tasks are easy, whereas there are many duplicated product structures among the bottom reaction templates. Examples of top and bottom reaction templates are shown in Fig. 2.5d. We can see that reaction templates with the same product structure have significant negative effects on the performance of both the GCN and ECFP models.

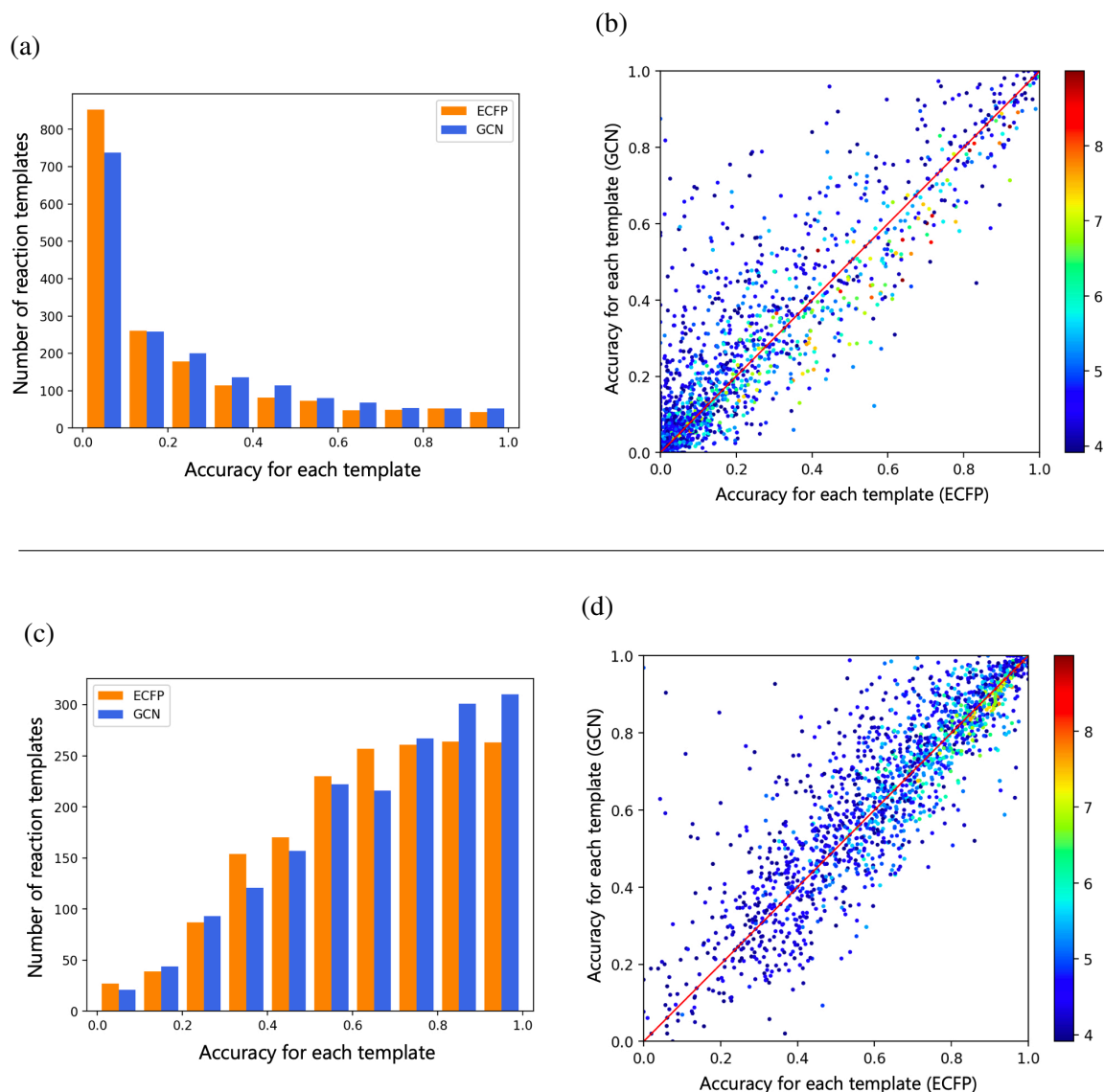


Figure 2.6: Comparison between the distributions of the top-1 and top-30 accuracies for each reaction template achieved by the GCN and ECFP models. (a) Histograms of the top-1 accuracies for each reaction template achieved by the GCN (blue) and ECFP (orange) models. (b) Scatter plot of the top-1 accuracies, with the color bar on the right representing the logarithm of the number of molecules in which the reaction template appears. (c) Histograms of the top-30 accuracies for each reaction template achieved by the GCN (blue) and ECFP (orange) models. (d) Scatter plot of the top-30 accuracies, with the color bar on the right representing the logarithm of the number of molecules in which the reaction template appears.

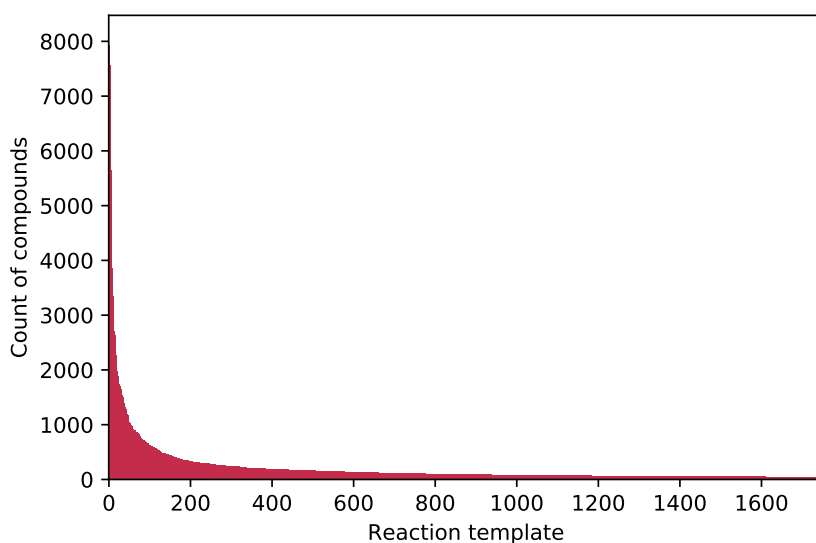


Figure 2.7: Histogram of the number of compounds per reaction template. The number of reaction templates is 1,752, and the number of compounds per template ranges from 50 to 8,000.

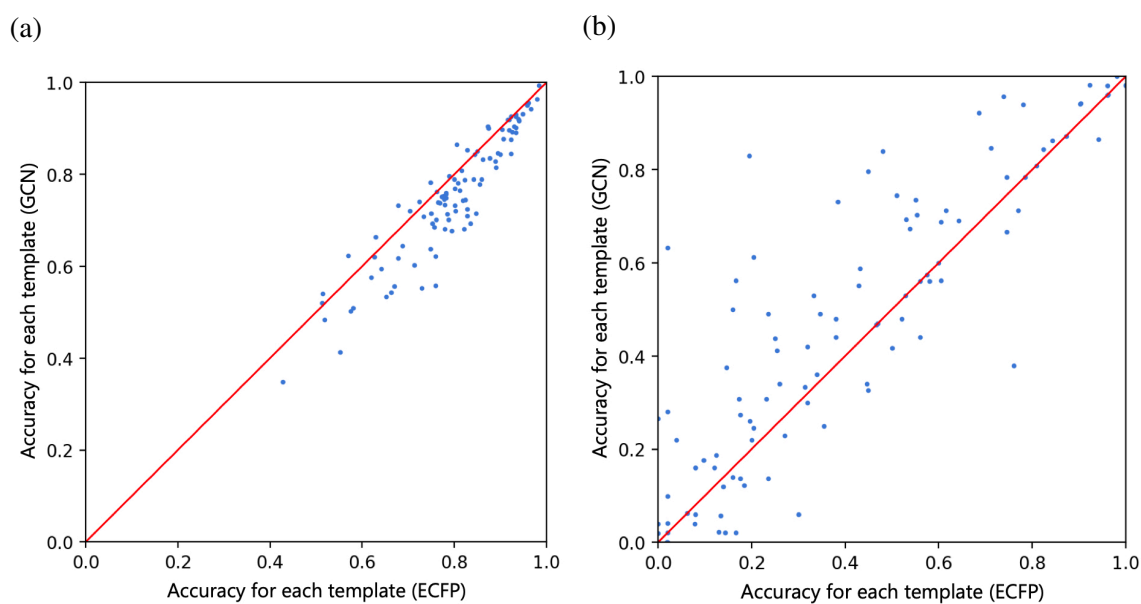
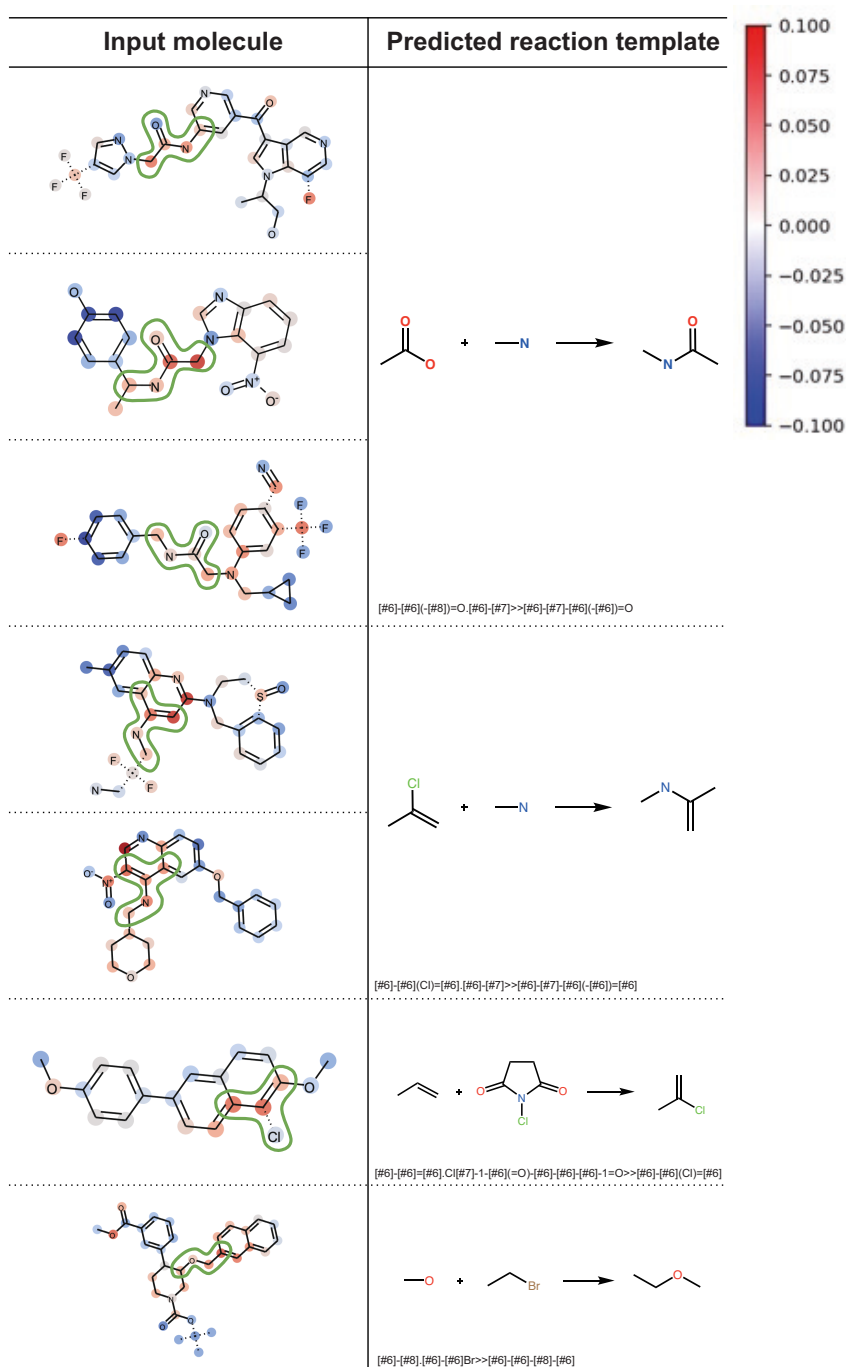


Figure 2.8: Distributions of the top-10 accuracies achieved by the GCN and ECFP models for the top 100 and bottom 100 reaction templates, where the reaction templates are ranked by the number of compounds per reaction template. (a) Scatter plot of the top-10 accuracies achieved by the GCN and ECFP models for the top 100 reaction templates. (b) Scatter plot of the top-10 accuracies achieved by the GCN and ECFP models for the bottom 100 reaction templates.

## 3.2 Visualization

We visualized the contributions of the atomic features in a molecule, which is a product in a reaction, to the results of retrosynthetic reaction prediction (Fig.2.9). We selected examples for which the GCN model correctly predicted the reaction template, as shown in Fig. 2.9a and 2.9b. Figure 2.9a shows examples in which the reaction center and the atomic contributions match, and Fig. 2.9b shows examples in which the reaction center and the atomic contributions do not match. Figure 2.9c shows examples of incorrect prediction; here, both the correct reaction template and the predicted reaction template are shown in the reaction template column. Red shading indicates positive contributions to the prediction results, and blue shading indicates negative contributions to the prediction results. The product substructures corresponding to correctly predicted reaction templates are indicated in light green, and the product substructures corresponding to incorrectly predicted reaction templates are indicated in light purple. To determine the atoms belonging to these light-green and light-purple substructures, we performed substructure matching for each molecule (as shown in, e.g., the left column in Fig. 2.9a) using the corresponding reaction center, as defined in the Dataset section (as shown in, e.g., the right column in Fig. 2.9a). Then, we marked the matching part of the molecule with the appropriate color.

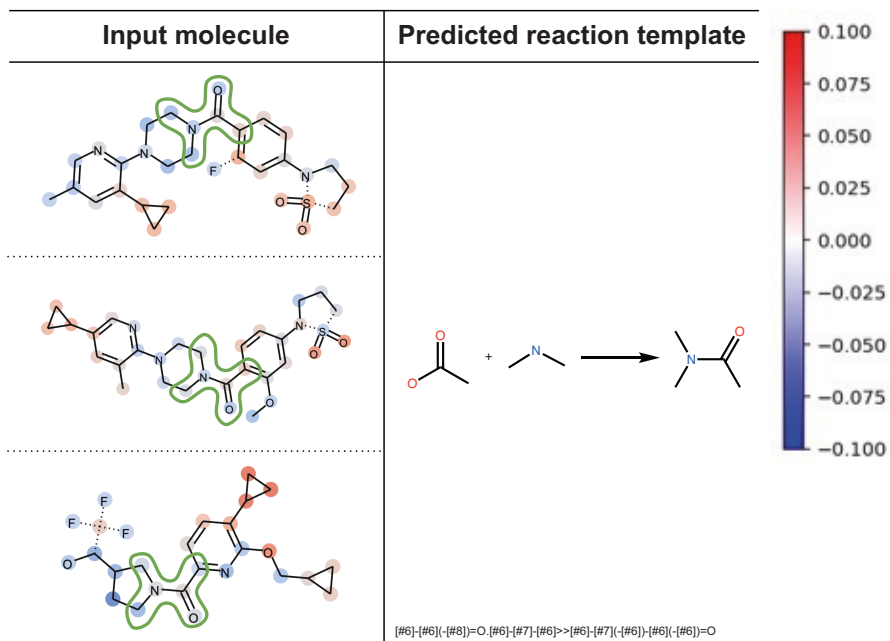




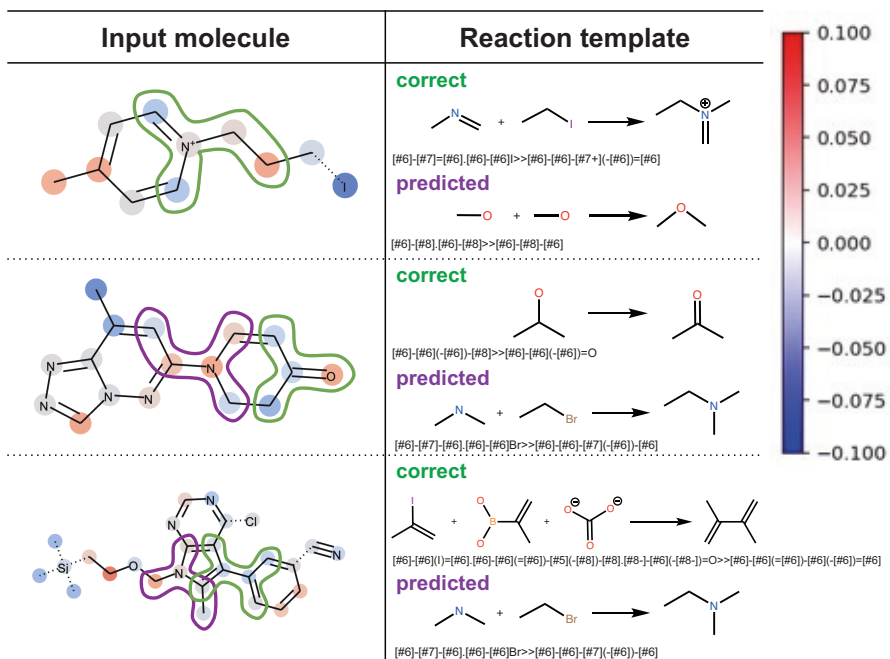
(a): Examples of correct predictions for which the atomic contributions and the reaction center match.

Figure 2.9: Visualization of the contributions of the atomic features in a molecule to retrosynthetic reaction prediction. The atoms marked in light green in a molecule correspond to the reaction center in the correct reaction template. The IG values are represented by shading in various colors, as shown by the color bar. Below the template drawings, the reaction templates are also expressed in the SMARTS format.

### 3. RESULTS



(b): Examples of correct predictions for which the atomic contributions and the reaction center do not match.



(c): Examples of incorrect predictions. The substructures marked in light green and light purple correspond to the correct and predicted reaction centers, respectively.

Figure 2.9: Visualization of the contributions of the atomic features in a molecule to retrosynthetic reaction prediction. The atoms marked in light green in a molecule correspond to the reaction center in the correct reaction template. The IG values are represented by shading in various colors, as shown by the color bar. Below the template drawings, the reaction templates are also expressed in the SMARTS format.

### Quantitative evaluation of IG-based visualization

To quantitatively evaluate the visualization performance, we calculated the average IG values of the atomic features in the reaction centers. Figure 2.10 shows a histogram of the standardized IG values of all atoms in a molecule (orange) and a histogram of the standardized average IG values in reaction centers (blue). Here, the standardized average IG value in a reaction center is defined as the average of the standardized IG values of the atoms in the reaction center. The orange and blue lines show Gaussian kernel density estimates of the distributions of the standardized IG values of all atoms in a molecule and the standardized average IG values in reaction centers, respectively. The average IG value of a reaction center is 3.71, whereas that of all atoms in a molecule is 0.0 because the IG values for each molecule are standardized. Thus, the distribution of the standardized averages in reaction centers is positively shifted. This result suggests that my system successfully recognizes reaction centers. The reason why the two distributions are not completely separated is that not all atoms in a reaction center will necessarily have a positive IG value (see Fig. 2.9).

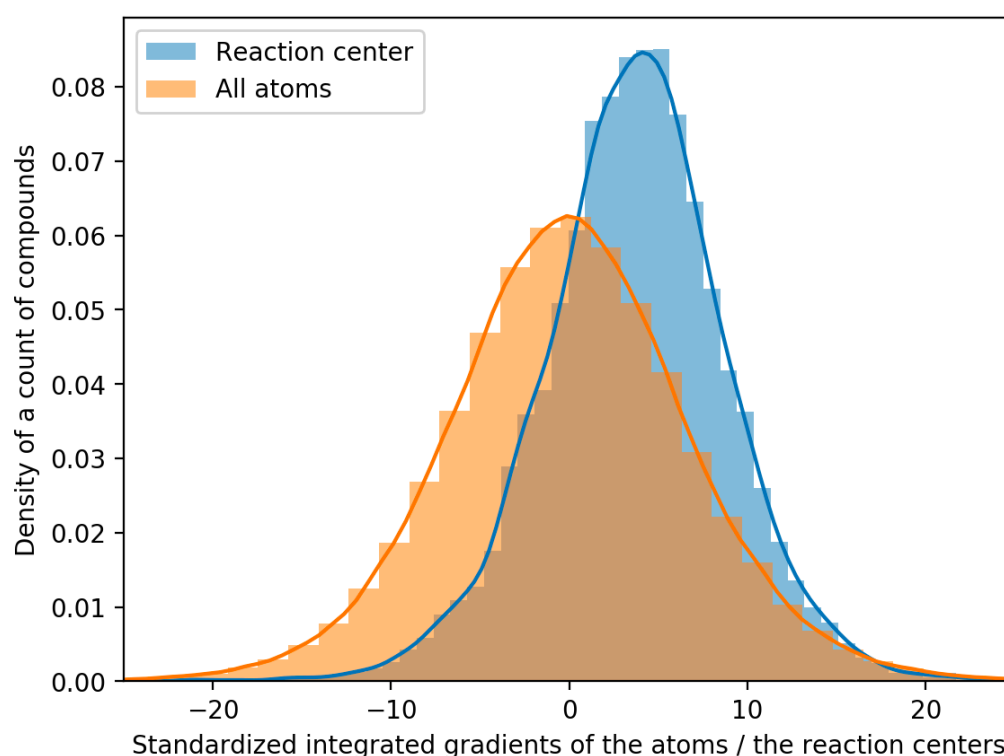


Figure 2.10: Histograms of the standardized average IG values in reaction centers (blue) and the standardized IG values of all atoms in a molecule (orange).

## 4 Discussion

As seen from the results shown in Fig. 2.4 and 2.5, the GCN model showed higher performance than the ECFP model in retrosynthetic reaction prediction. Many previous studies have shown that in tasks involving the prediction of molecular properties, graph-based approaches outperform conventional ML approaches[94]. In addition, graph-based approaches have performed well in multitask learning for problems involving as many as hundreds of classes[94]. In this study, it was found that compared to a conventional neural network method based on ECFP, the graph-based approach was also effective for retrosynthetic reaction prediction and showed better performance for prediction involving almost 1,800 classes. Moreover, although the dataset was biased, the GCN model tended to predict a wide range of reaction templates correctly. This may be attributable to the generally low susceptibility of graph-based methods to overfitting on a dataset[102]. In general, this tendency is important in retrosynthetic analysis because important reactions do not always appear frequently among the reaction templates.

Using the IG method, I have shown that the proposed system successfully recognizes reaction centers for retrosynthetic reaction prediction, as illustrated in Fig. 2.9 and 2.10. Although data-driven retrosynthetic analysis has not previously shown sufficient interpretability for retrosynthetic reaction prediction, my IG-based system offers a basic approach for investigating the rationale for each step of a proposed synthetic route. Even if the contributions to a prediction generated based on ECFP can be visualized, this can be done only by the substructure unit. When contributions are visualized by the substructure unit, I cannot consider the influences of neighboring atoms on the reaction center because the substructures in the fingerprint are not related to each other. Conversely, using my model, I can visualize the contributions by the atomic unit, allowing the influences of neighboring atoms on the reaction center to be considered. I believe that this improvement in interpretability is essential in making data-driven approaches more accessible to chemists.

Figure 2.9 suggests that the existence of common substructures in molecules associated with the same reaction template contributes positively to retrosynthetic reaction prediction. If various molecules are associated with the same reaction template, as shown in Fig. 2.9a, the IG method can reflect the common reaction center for visualization. However, if similar molecules are associated with the same reaction template, the GCN model tends to predict a reaction template by recognizing a characteristic substructure (e.g., the cyclopropyl group in Fig. 2.9b) other than the reaction center, as shown in Fig. 2.9b. One possible solution to the above problem could be to use larger-scale chemical reaction databases such as Reaxys

and SciFinder. Larger databases would ensure higher diversity of molecules associated with the same reaction template, and the GCN model should be more likely to predict the correct class by recognizing a common reaction center.

To confirm the GCN model's performance for natural products, I performed retrosynthetic reaction prediction for four natural products with different structural complexities: benzylpenicillin, erythromycin A, morphine, and prostaglandin E1 (Fig.2.11), which were not included in the training dataset. The prediction for benzylpenicillin is thought to be reasonable. However, the other prediction results are considered to be unreasonable. The reason why these predictions were unsuccessful is that the model could not effectively learn important features for natural products because the USPTO reaction dataset contains few natural products.

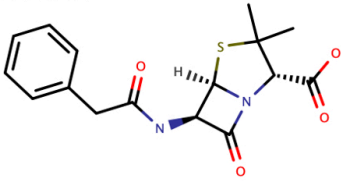
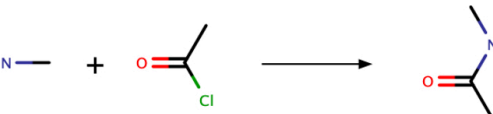
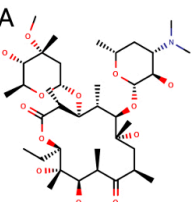
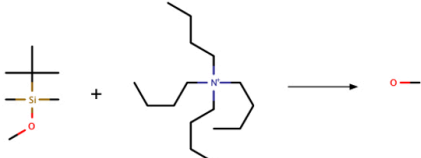
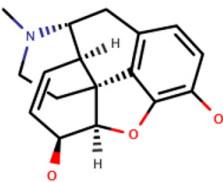

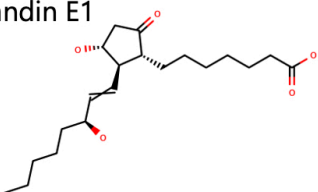
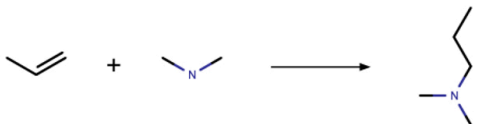
Natural Product	Predicted reaction template
Benzylpenicillin 	 <chem>[#6]-[#7].[#6]-[#6](Cl)=O&gt;&gt;[#6]-[#7]-[#6](-[#6])=O</chem>
Erythromycin A 	 <chem>[#6]-[#8][Si]([#6])([#6])C([#6])([#6])[#6]-[#6]-[#6]-[#6][N+]([#6]-[#6]-[#6])([#6]-[#6]-[#6])[#6]-[#6]-[#6]&gt;&gt;[#6]-[#8]</chem>
Morphine 	 <chem>[#6]-[#7]-[#6].[#6]-[#6](F)=[#6]&gt;&gt;[#6]-[#7](-[#6])-[#6](-[#6])=[#6]</chem>
Prostaglandin E1 	 <chem>[#6]-[#6]=[#6].[#6]-[#7]-[#6]&gt;&gt;[#6]-[#6]-[#6]-[#7](-[#6])-[#6]</chem>

Figure 2.11: Examples of the top-1 predictions of my model for four natural products with different structural complexities.

In future work, I will focus on improving the performance of my model in the following three ways. The first is by oversampling the lower reaction templates. The second is by setting a postfiltering parameter in combination with the IG method to rerank the predicted reaction templates. The last is by developing a molecular representation that considers precise local charge and chemical structure information, such as rotamers, topoisomers, and steric hindrance. These methods are expected to improve the top- $n$  balanced accuracy and make the improved model more suitable for chemist-friendly retrosynthetic analysis. I also plan to compare the improved GCN model with other advanced DL approaches, including Transformer models[88, 89]. Additionally, I will address the tasks of predicting reaction conditions, yields, and multistep routes using my system.

## 5 Conclusion

I succeeded in developing a GCN-based interpretable retrosynthetic reaction prediction system using IG for visualization. The prediction performance of my GCN-based model was compared with that of a traditional ECFP model. The results showed that the prediction accuracy of the GCN model was higher than that of the ECFP model and that the GCN predictions were less influenced by dataset bias. Additionally, visualizations of the GCN predictions using IG successfully showed the atomic contributions to the results of retrosynthetic reaction prediction. Through such visualization, I can investigate the underlying drivers of retrosynthetic reaction predictions, which is expected to help chemists better understand the outcomes of retrosynthetic reaction prediction based on a data-driven approach. My model is expected to serve as a cornerstone for the construction of a high-quality model for retrosynthetic reaction prediction, which will be important in facilitating searches for retrosynthetic routes.

In the next chapter, I develop a data-driven CASP approach using this graph-based retrosynthetic reaction prediction model.

## Chapter 3

# AI-Driven Synthetic Route Design with Retrosynthesis Knowledge

### 1 Introduction

As briefly described in the general introduction, CASP approaches are generally classified into two types: rule-based[15, 9] and data-driven approaches[13, 16]. One excellent rule-based CASP application, Chematica[15] (now rebranded as Synthia™), provides considerable discretion for chemists to perform retrosynthetic analysis based on their own ways of thinking and has come into global use[20, 103]. However, rule-based approaches require the extreme efforts of many experts and cannot keep up with the exponential growth in knowledge related to chemistry[104].

On the other hand, recent breakthroughs in deep learning (DL)[37, 105], along with the widespread availability of reaction records[98, 106] and open-source codes[69, 107, 108, 109], have improved the core techniques of data-driven CASP, such as 1-step (retro)synthetic reaction prediction[45, 88, 110] and multistep synthetic route searches[16, 29, 18, 111, 112]. In reaction prediction models, various representations of molecules (e.g., fingerprints[45], SMILES strings[88, 110, 113], and graphs[86]) and corresponding suitable DL techniques have been used, showing promising performances. Regarding search algorithms, Monte Carlo tree search (MCTS)[16, 109, 114, 115], depth-first proof number search[111, 116], and graph-based exploration methods[29, 112] have been used to obtain optimal or possible synthetic routes. Several outstanding data-driven CASP applications have been on the stage of practical use in industry and in the laboratory[16, 28, 29]; these applications have led to a remarkable revival of interest in CASP research[25, 26, 27].

However, in regard to practical retrosynthesis, most data-driven CASP applications are lacking in their ability to reflect or support flexible adaptation to individual chemists' ways of thinking. The search algorithms used in such applications depend on naive scoring functions for evaluating whether one synthetic position found during a search is preferable to another[16, 114]; this implies that there are few opportunities to learn diverse strategies for retrosynthetic analysis. Moreover, the data-driven CASP approaches incorporating generally used retrosynthesis knowledge have not been developed, and the effects of knowledge on search performance have yet to be investigated.

In this study, I developed a hybrid CASP application combining data-driven and rule-based techniques called "ReTrosynthesis planning application using Retrosynthesis Knowledge (ReTReK)," which integrates the knowledge into the evaluation of promising search directions. ReTReK takes the knowledge as input in the form of user-adjustable parameters, thus allowing users to easily decide which retrosynthesis knowledge to use and how much emphasis to place on it. To represent retrosynthesis knowledge in ReTReK, I formulated four scores: a convergent disconnection score (CDScore), an available substances score (ASScore), a ring disconnection score (RDScore), and a selective transformation score (STScore). To construct the ReTReK model with high generalization capability, the Reaxys reaction database[106], one of the largest reaction databases in the world, was used. I evaluated and demonstrated the performance of ReTReK using molecules from the ChEMBL database[117] and drug-like molecules[30, 33, 31, 32, 34, 35]. I successfully demonstrated that synthetic routes designed using ReTReK with retrosynthesis knowledge were preferable to those designed without retrosynthesis knowledge. Furthermore, I demonstrated that retrosynthesis knowledge improves the performance when solving for certain target molecules, and I successfully guided the search direction in MCTS. The proposed concept of integrating retrosynthesis knowledge, in the form of adjustable parameters, into a data-driven CASP application is expected to enhance the performance of both existing data-driven CASP applications and those under development. My implementation is available on GitHub at <https://github.com/clininfo/ReTReK>.



## 2 Methods

### 2.1 Construction of ReTReK

To implement a data-driven CASP application that can reflect the retrosynthesis knowledge, ReTReK was constructed using MCTS and graph convolutional network (GCN) techniques in addition to the four retrosynthesis knowledge scores introduced above (Fig. 3.1).

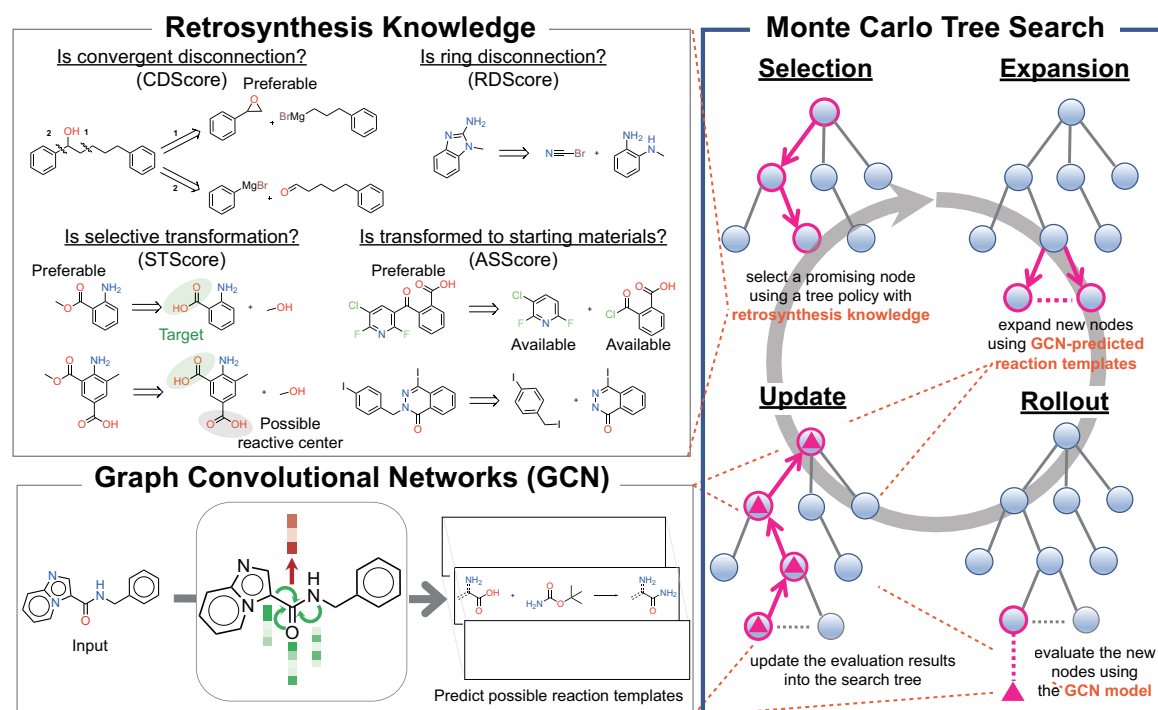


Figure 3.1: Whole workflow of ReTReK. ReTReK combines MCTS and GCN techniques, and retrosynthesis knowledge is incorporated into the selection step of the MCTS procedure. The retrosynthesis knowledge is represented by four scores: the CDScore, STScore, RDScore, and ASScore.

The basic MCTS algorithm comprises four steps: selection, expansion, rollout, and update. For the selection step, a tree policy is used to select a promising retrosynthetic tree position; this policy considers the retrosynthesis knowledge scores. A GCN-based model is used for 1-step retrosynthetic reaction prediction as a policy network in the expansion and rollout steps. Reaxys reaction records[106] were used to train the model and prepare the starting materials; compounds obtained from the ZINC database[118] were used as starting materials. By iterating through the four steps listed above, a retrosynthetic tree is expanded, thus attempting to identify a promising synthetic route.

## 2.2 Datasets

To create the ReTReK model, compounds obtained from Reaxys reaction records[106] and the ZINC 15 database[118] were used. To evaluate the performance of ReTReK, compounds obtained from the ChEMBL 27 database[117] and the literature[30, 33, 31, 32, 34, 35].

### Reaction template extraction

A set of approximately 50 million reaction records from Reaxys[106] (from 1795–2019) was used to construct a 1-step retrosynthetic reaction prediction model. The model was designed to take a target or intermediate molecule as input and was trained to predict a suitable reaction template for the input molecule. The purpose of a reaction template is to represent a generalized chemical reaction, and for this study, a reaction template was defined as consisting of a reactive center, orphan atoms, and their first-degree neighbors. An orphan atom is one that appears on only one side of the reaction arrow in ChemAxon[100]. The reaction template extraction procedure comprised four steps (see Fig. 3.2 for the workflow of reaction template extraction).

In the first step, the reaction records were standardized by removing explicit hydrogen, aromatizing, and retaining the largest fragments.

In the second step, the reaction records were filtered based on three conditions: (1) the reaction was required to consist of a single step, (2) the reaction was required to have a product and up to three reactants, and (3) the number of heavy atoms in the product was limited to 50 or fewer. After this step, the number of remaining reaction records was 22,337,137.

In the third step, reaction templates were extracted from the reaction records, and sets consisting of a product and the corresponding reaction template were retained if the reaction template occurred at least 50 times. To prevent the occurrence of two or more fragments, a reaction template was retained only in the case in which all atoms on the product side of the template were connected.

In the final step, the sets consisting of a product and the corresponding reaction template were filtered by the condition that the reaction template could be reversibly applied to the product and the derived reactants. With this requirement, 7,589,744 product–template sets remained, and the number of unique reaction templates was 19,633. Referring to a

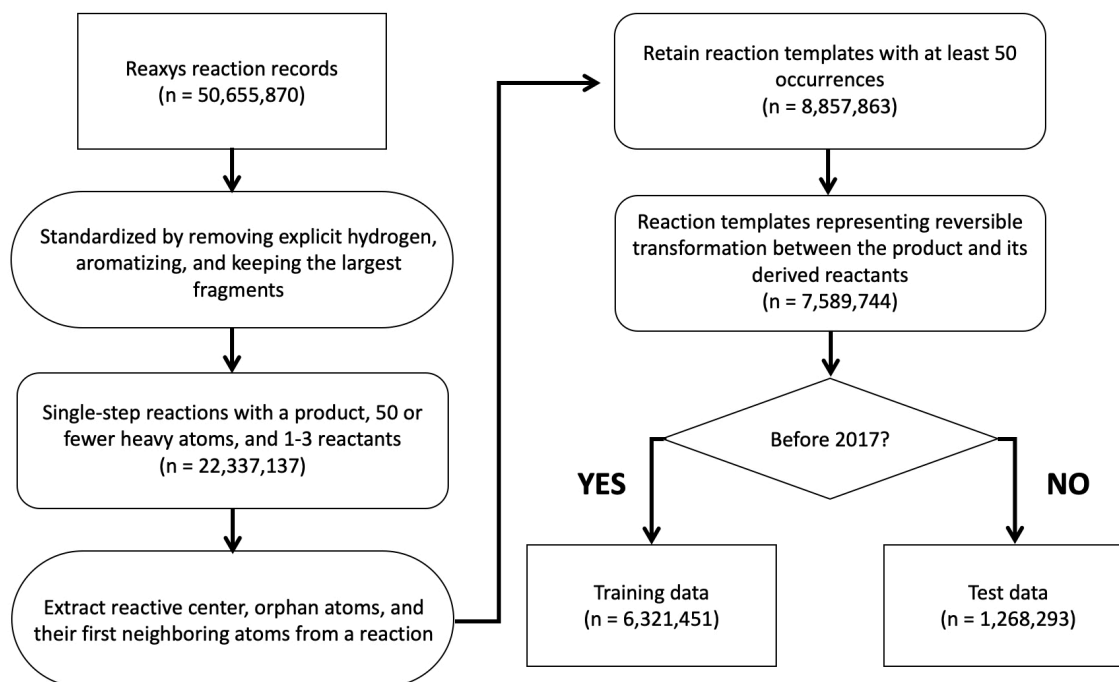


Figure 3.2: Workflow of reaction template extraction. The reaction template extraction procedure consists of four steps: (1) Reaction records were standardized by removing explicit hydrogen, aromatizing, and keeping the largest fragments; (2) Reaction records were narrowed down under the condition that a reaction is a single-step reaction that has a product and 1–3 reactants; (3) Reaction templates were extracted from the filtered reaction records; (4) Sets of a product and the corresponding reaction template were filtered by the condition that the reaction template can reversibly be applied to the product and the derived reactants.

previous study[119], a time-splitting strategy was employed to evaluate the neural network model performance. I assigned 6,321,451 sets published before 2017 to training data and 1,268,293 published in 2017 and later to test data.

### Preparation of molecules for ReTReK evaluations and demonstrations

The molecules used for the ReTReK evaluations were obtained from ChEMBL 27[117] and preprocessed via the following procedures. First, molecules whose USAN years ranged from 2017 to 2019 and for which chemical structure records were available were selected, resulting in a total of 219 compounds. Then, the compounds were preprocessed using the following steps: removing explicit hydrogen, aromatizing, retaining the largest fragments, removing compounds with more than 50 atoms, and removing duplicates. The remaining 161 compounds were used for the evaluations (ChEMBL dataset). For further evaluation

of ReTReK, six drug-like compounds[30, 33, 31, 32, 34, 35] were also used for synthetic route search demonstrations.

### Starting materials

A set of compounds obtained from the ZINC database and Reaxys reaction records were used as starting materials. A subset of 100,023 building blocks from major suppliers (Sigma-Aldrich, Alfa Aesar, and Acros) was obtained from the ZINC database. From the Reaxys reaction records, 649,130 compounds recorded as reactants with at least five occurrences before 2017 were used. All compounds were stored in the canonical SMILES format calculated by RDKit[69].

## 2.3 MCTS for retrosynthesis

MCTS has been implemented in various CASP studies based on the achievements of Segler *et al.*[16]. MCTS is a search algorithm for exploring optimal solutions and comprises four steps: selection, expansion, rollout, and update[120]. Following Segler’s implementation[16], a state contains a set of molecules and is solved (the optimal solution) if all molecules in the state are starting materials. In this study, retrosynthesis knowledge scores were incorporated into the evaluation term used in the selection step. The same policy network was used for both the expansion and rollout steps, similar to previous research[114].

### Retrosynthesis knowledge used in ReTReK

I have defined four scores representing four types of retrosynthesis knowledge, namely, the CDScore, ASScore, RDScore, and STScore, inspired by previous work[10, 14, 15].

**Convergent disconnection score** The CDScore is designed to favor convergent synthesis, which is known to be an efficient strategy in multistep chemical synthesis. The CDScore is calculated by evaluating how equally a product is divided among the reactants of a reaction  $\{R_1 + R_2 + \dots + R_n \rightarrow P\}$ , where  $R_i$  is a reactant and  $P$  denotes the product.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{a(P)}{n} - a(R_i) \right| \quad (3.1)$$

$$\text{CDScore} = \frac{1}{1 + \text{MAE}} \quad (3.2)$$

Here,  $a(\text{P})$  and  $a(\text{R}_i)$  represent the numbers of atoms in the product and a reactant, respectively, and MAE is the mean absolute error.

**Available substances score** To serve a similar purpose as the CDScore, the ASScore is defined to reflect the number of available substances generated in a reaction step and is calculated as

$$\text{ASScore} = \frac{b(\text{S})}{b(\text{R})} \quad (3.3)$$

Here,  $b(\text{S})$  and  $b(\text{R})$  represent the numbers of available substances (starting materials) and reactants, respectively.

**Ring disconnection score** A ring construction strategy is preferred if the target compound has complex ring structures because the construction of ring structures in a synthetic route tends to result in simple and easily available starting materials. The RDScore is calculated by checking whether ring construction occurs in a reaction step, as follows:

$$\text{RDScore} = \begin{cases} 1 & d(\text{P}) > \sum_{i=1}^n d(\text{R}_i) \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

Here,  $d(\text{P})$  and  $d(\text{R}_i)$  represent the numbers of rings in the product and a reactant, respectively.

**Selective transformation score** A synthetic reaction with few byproducts is preferred in terms of yield. To reflect the number of possible products from a reaction, the STScore is calculated by focusing on the number of reactive centers in the reactants, as follows:

$$\text{STScore} = \frac{1}{e(\sum_{i=1}^n \text{R}_i)} \quad (3.5)$$

Here,  $e(\sum_{i=1}^n \text{R}_i)$  represents the applicable number of patterns of products enumerated using the reactants and a certain reaction template.

## Policy network

In this study, the term ‘policy network’ (Fig. 3.3a) refers to a template-based retrosynthetic reaction prediction model, and the same model was used in both the expansion and rollout steps. I employed a GCN model, which was found to be a promising model for retrosynthesis in Chapter 2, as the retrosynthetic reaction prediction model. The model was trained using the dataset prepared as described in the reaction template extraction section and comprised three graph convolutional layers with leaky rectified linear unit (Leaky ReLU) activation and a dropout ratio of 0.3, a graph dense layer with Leaky ReLU activation, a graph gather layer with hyperbolic tangent activation, and a dense layer with softmax activation. To confirm the effectiveness of the expansion sizes on the MCTS performance, the top- $n$  accuracies were calculated for  $n$  values in the range from 1 to 1000. To implement this model, the graph-based deep learning framework kGCN (Chapter 1) was used.

## Selection

Starting from the root node, a tree policy is recursively applied to select the next action, which implies that the simulation descends through the search tree step by step until an unvisited node with a nonterminal state is reached. The tree policy is based on the upper confidence bound (UCB) score, and retrosynthesis knowledge is incorporated into the policy as follows:

$$K = \frac{1}{n}(w_1\text{CDScore} + w_2\text{ASScore} + w_3\text{RDScore} + w_4\text{STScore}), \quad (3.6)$$

where the  $w_i$  represent weights, with values of  $w_1 = 5.0$ ,  $w_2 = 0.5$ ,  $w_3 = 2.0$ , and  $w_4 = 2.0$ , and  $n$  denotes the number of retrosynthesis knowledge scores used in a search (e.g.,  $n$  is four if all four types of retrosynthesis knowledge are used).

$$\text{action} = \frac{Q}{N} + cP \frac{\sqrt{N_{-1}}}{1+N} + K \quad (3.7)$$

Here,  $Q$  denotes an action value calculated in the update step;  $N$  and  $N_{-1}$  are the visit counts of the child and parent nodes, respectively;  $c$  denotes a constant value that is set to 10;  $P$  denotes the softmax probability obtained from the policy network; and  $K$  represents the mean of the retrosynthesis knowledge scores.

## Expansion

Child nodes whose states are selected by the policy network are added to the node selected by the tree policy. Based on the top- $n$  accuracies in the policy network, in independent trials, the top 50, 100, 300, and 500 reaction templates were selected, and they were filtered by the condition that the reaction templates could be successfully applied to the molecules in the states and were then added to the node selected by the tree policy.

## Rollout

A simulation is implemented in the policy network if the state of a node is not proven or terminal[16]. During the simulation, the following steps are recursively implemented for a maximum of five times: an unresolved molecule (not included among the starting materials) of the state is randomly sampled, the top 10 reaction templates of the molecule are obtained by the policy network, and a randomly sampled reaction template is applied to the molecule. At the end of each step, it is checked whether the state is proven or not.

A reward function  $r$  returns one of three values as a reward  $z$ , depending on the simulation result. Before the simulation is started, the reward is 10 if the state is proven and -1 if the state is terminated. After the simulation, the reward is equal to the ratio of the number of resolved molecules in the state to the total number of molecules.

## Update

The reward obtained from the rollout step is backpropagated through the selected nodes to update their action values  $Q$ . Based on previous research[16], the value of  $Q$  is defined as

$$W = \max\left(0, \frac{L_{max} - L + \sum_{i=1}^n P_i}{L_{max}}\right), \quad (3.8)$$

$$Q = zW, \quad (3.9)$$

where  $L_{max}$  denotes the maximal branch length and is set to 10,  $L$  denotes the current branch length, and  $\sum_{i=1}^n P_i$  denotes the sum of the softmax probabilities of the reaction templates in the selected nodes.

## 2.4 Evaluating the effects of expansion sizes and retrosynthesis knowledge on MCTS solution performance

To investigate the effect of the expansion size on the MCTS performance in solving for target molecules, both the number of solved molecules in the ChEMBL dataset and the times needed to solve the molecules were compared for different expansion sizes and six retrosynthesis knowledge patterns. The expansion sizes were 50, 100, 300, and 500, and were determined by the policy network's top- $n$  accuracies. The six knowledge patterns were as follows: no retrosynthesis knowledge (no knowledge), the CDScore, the ASScore, the RDScore, the STScore, and all four retrosynthesis knowledge scores (all knowledge). In these experiments, the maximum number of iterations was set to 500 and the score weights for the CDScore, ASScore, RDScore, and STScore were fixed to 5.0, 2.0, 0.5, and 2.0, respectively.

## 2.5 Evaluating the effects of retrosynthesis knowledge on the search directions in MCTS

To quantify the effects of retrosynthesis knowledge on the search directions in MCTS, I defined a route score as the average value of the corresponding retrosynthesis knowledge score in each step of a solved synthetic route. I calculated four types of route scores (rCDScore, rASScore, rRDScore, and rSTScore) for the solved synthetic routes under the corresponding retrosynthesis knowledge patterns. To facilitate comparison, each route score for the five retrosynthesis knowledge patterns was standardized based on the corresponding mean and standard deviation for the no-knowledge patterns. In these experiments, synthetic routes solved under the condition of an expansion size of 500 were used. The maximum number of iterations was set to 500, and the score weights for the CDScore, ASScore, RDScore, and STScore were fixed to 5.0, 2.0, 0.5, and 2.0, respectively.



## 3 Results and Discussion

### 3.1 Top- $n$ accuracies of the GCN-based policy network

To determine the effective size for the expansion step in the MCTS procedure, the top- $n$  accuracies (for  $n$  up to 1,000) of the GCN-based policy network were calculated, as shown in Fig. 3.3b. The 1-step retrosynthetic reaction prediction model aimed to prioritize 19,633 reaction templates for application to an input molecule. In this study, I used reaction templates consisting of a reactive center, first-degree neighbors, and protecting groups because a previous study proposed the use of this type of reaction template to maintain chemical integrity[114]. Accordingly, the top-1, top-50, top-100, top-300, and top-500 accuracies were found to be 0.361, 0.906, 0.938, 0.968, and 0.976, respectively. Beyond the top-500 accuracies, the increase in the prediction performance was not insignificant. Considering the results of Chapter 2 for reaction templates of different sizes, the prediction performance was assumed to be equivalent to or better than that of the previous template-based 1-step retrosynthetic reaction prediction model[16]. Based on the results for the top- $n$  accuracies, I evaluated the effect of the MCTS expansion sizes and retrosynthesis knowledge on the performance of solving for target molecules using the top 50, 100, 300, and 500 predicted templates.

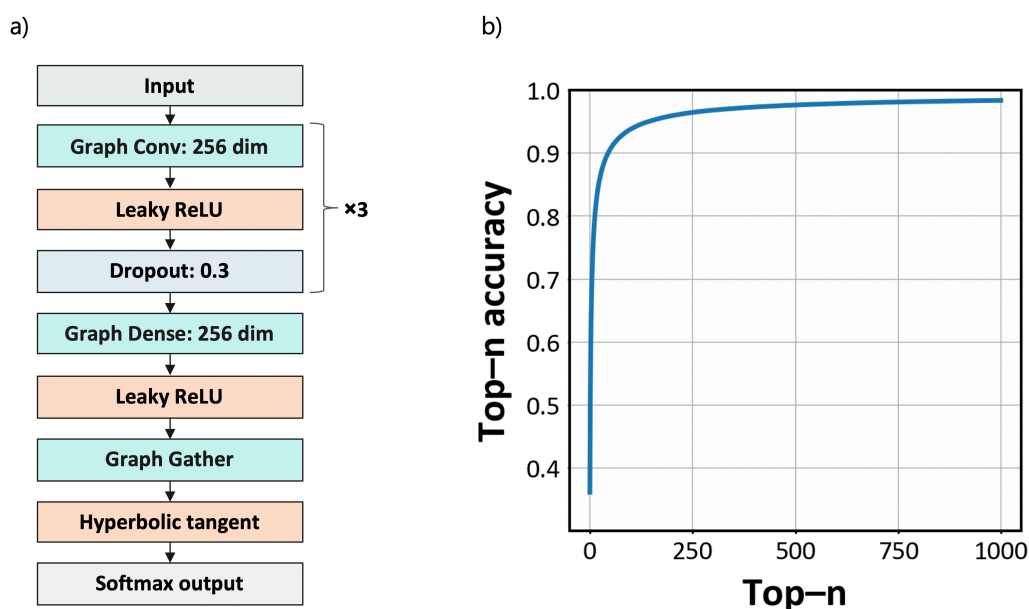


Figure 3.3: (a) Model architecture of the GCN-based policy network. (b) Top- $n$  accuracies of the model for  $n$  values ranging from 1 to 1000. Specifically, the top-1, top-50, top-100, top-300, and top-500 accuracies are 0.361, 0.906, 0.938, 0.968, and 0.976, respectively.

### 3.2 Effects of expansion sizes and retrosynthesis knowledge on the performance of solving for target molecules

Figure 3.4 shows how the expansion sizes and retrosynthesis knowledge influenced the performance of solving for target molecules. The 161 molecules from the preprocessed ChEMBL dataset were used as the target molecules. The searches were performed with different expansion sizes (50, 100, 300, and 500) and six retrosynthesis knowledge patterns: no retrosynthesis knowledge (no knowledge), the CDScore, the ASScore, the RDScore, the STScore, and all four retrosynthesis knowledge scores (all knowledge). In most cases, the solution performance was improved in proportion to the expansion size. However, the case of the STScore pattern and an expansion size of 100 resulted in a lower number of solved molecules than in the case of the same pattern and an expansion size of 50. This result is attributed to a relative lack of MCTS iterations because of the increase in the expansion size. Regarding retrosynthesis knowledge, all knowledge patterns except the STScore pattern resulting in an increase in the number of solved molecules compared to the no-knowledge pattern. The CDScore pattern with an expansion size of 500 showed the best solution performance, yielding 90 solved molecules, whereas the no-knowledge pattern with the same expansion size resulted in 59 solved molecules. Although the STScore pattern resulted in fewer solved molecules than the no-knowledge pattern, this result is considered reasonable because the STScore focuses on reactions with few byproducts, which often leads to strict conditions for retrosynthesis.

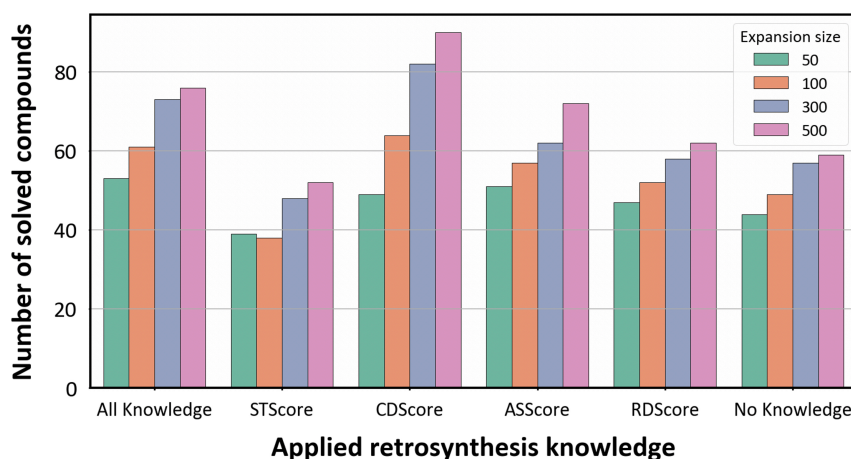


Figure 3.4: Comparison of the numbers of solved molecules with different expansion sizes and retrosynthesis knowledge patterns. The green, orange, blue, and pink bars correspond to expansion sizes of 50, 100, 300, and 500, respectively. The results of retrosynthetic analyses with five retrosynthesis knowledge patterns (CDScore, ASScore, RDScore, STScore, and all knowledge) and without any retrosynthesis knowledge (no knowledge) are shown.

Moreover, the search times necessary for solution were compared for the different expansion sizes and the six retrosynthesis knowledge patterns, as shown in Fig. 3.5. The search time increased in proportion to the expansion sizes because an increase in the expansion size expands the search space for MCTS. The median search times for expansion sizes of 50, 100, 300, and 500 were 32, 45, 133, and 294 seconds, respectively. The STScore pattern required shorter search times than the no-knowledge pattern, although all other knowledge patterns except STScore resulted in longer search times. These results suggest that synthetic routes can be more efficiently identified under the STScore pattern than the other patterns, although the STScore pattern results in lower solution performance.

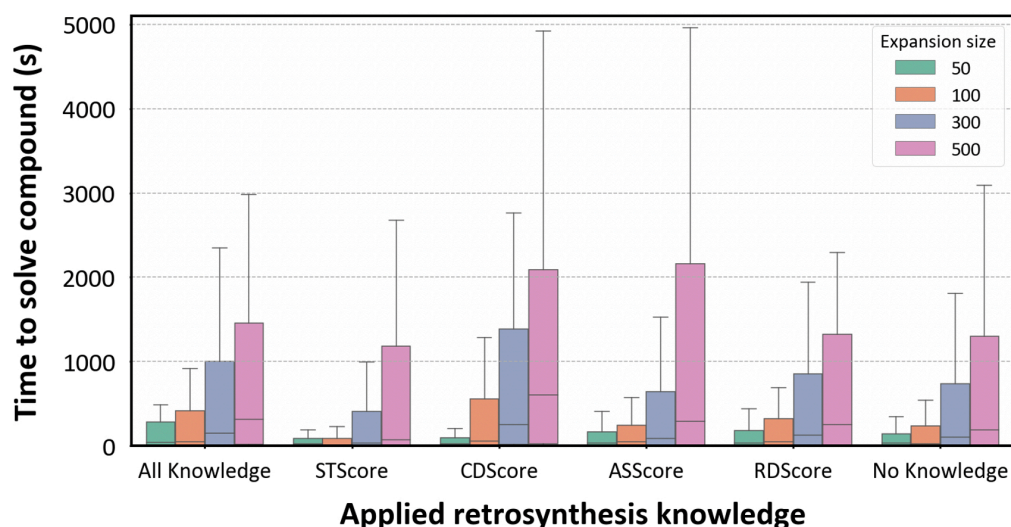


Figure 3.5: Comparison of the times necessary to solve compounds for different expansion sizes and retrosynthesis knowledge patterns. The green, orange, blue, and pink bars correspond to expansion sizes of 50, 100, 300, and 500, respectively. The results of retrosynthetic analyses with five retrosynthesis knowledge patterns (CDScore, ASScore, RDScore, STScore, and all knowledge) and without any retrosynthesis knowledge (no knowledge) are shown. The maximum reach of the whiskers in each boxplot is defined as  $1.5IQR$ , where  $IQR$  represents the interquartile range. Outliers, defined as data points beyond the whiskers, are not shown in the boxplots.

### 3.3 Effects of retrosynthesis knowledge on the search directions in MCTS

Figure 3.6 shows how the six retrosynthesis knowledge patterns influenced the characteristics of the searched synthetic routes in terms of four route scores (rSTScore, rCDScore,

rASScore, and rRDScore). Each route score was defined as the average corresponding retrosynthesis knowledge score in each step of the searched synthetic route. For ease of comparison, each route score for each of the five knowledge patterns was standardized with respect to the corresponding score for the no-knowledge pattern. The standardized mean values of the rSTScore for the STScore pattern, the rCDScore for the CDScore pattern, the rASScore for the ASScore pattern, and rRDScore for the RDScore were 0.178, 0.555, 0.130, and 0.309, respectively. All values were positively shifted compared to the values for the no-knowledge pattern, indicating that all four retrosynthesis knowledge scores successfully guided the search directions in MCTS according to the characteristics of each type of knowledge. Intriguingly, the CDScore pattern led MCTS to selective transformation-oriented searches compared to the STScore pattern (the mean values of the CDScore and STScore were 0.299 and 0.178, respectively). A convergent-disconnection-oriented search is assumed to have more chances to split reactive centers into divided molecules because the CDScore attempts to minimize the sizes of each divided molecule simultaneously.

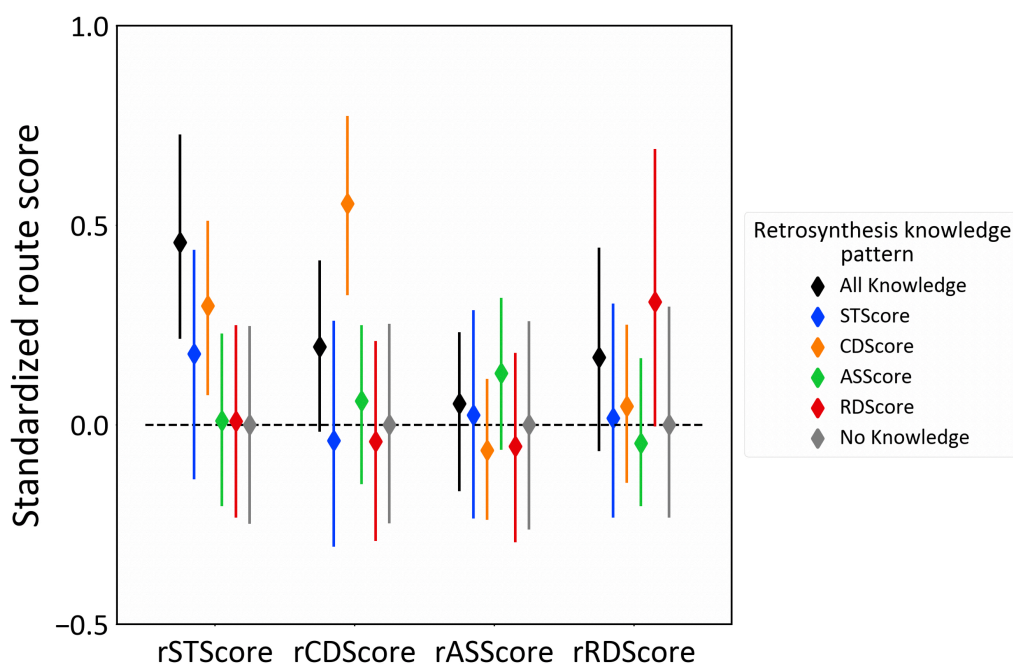


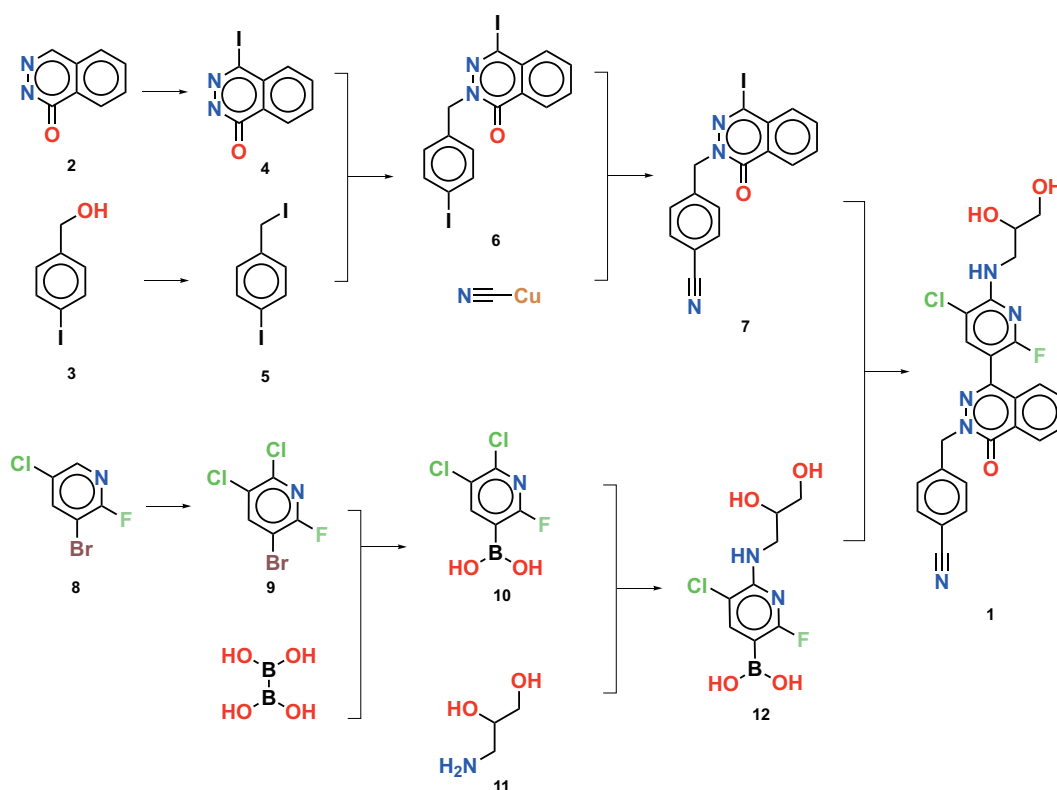
Figure 3.6: Evaluation of the effects of retrosynthesis knowledge on the search directions in MCTS. Synthetic routes solved with an expansion size of 500 were used for this evaluation. Each route score was standardized based on the corresponding mean and standard deviation for the no-knowledge pattern. The black, blue, orange, green, red, and gray plots represent the standardized route scores for the all-knowledge, STScore, CDScore, ASScore, RDScore, and no-knowledge patterns, respectively. The rhombuses represent the mean values for each case, and the confidence intervals at the 95% confidence level are also shown.

Considering that the all-knowledge pattern showed a higher rSTScore than the CD-Score and STScore patterns, these results suggest the existence of synergistic effects of retrosynthesis knowledge; however, this hypothesis needs further analysis.

### 3.4 Demonstrations of ReTReK for drug-like molecules

To demonstrate retrosynthesis planning using ReTReK, I applied ReTReK to two drug-like molecules in the cases of the all-knowledge and no-knowledge patterns, and the results are presented in this section. Figure 3.7 illustrates exemplary retrosynthetic routes to molecule **1**, known as a hepatitis B virus capsid inhibitor[30], found by ReTReK with and without retrosynthesis knowledge. Exploration with retrosynthesis knowledge suggested a convergent route, successfully reflecting the specified knowledge scores (Fig. 3.7a). In this route, the target molecule **1** is disconnected into two main segments, iodophthalazine **7** and pyridylboronic acid **12**, which can be converted into **1** by the Suzuki coupling reaction. The key intermediates **7** and **12** could be retrosynthetically divided into three representative materials: hydroxyphthalazine **2**, benzyl alcohol **3**, and tri-halogenated pyridine **8**. Iodination of **2** and subsequent N-benylation with *p*-iodobenzyl iodide **5**, which can be obtained from **3**, would give 2-(iodobenzyl)phthalazin-1-one **6**. A reaction of **6** with copper cyanide would provide the intermediate **7**. The other intermediate **12** would be prepared from **8** via a 3-step sequence (i.e., chlorination, incorporation of a boronic acid moiety, and amination with aminoalcohol **11**). In contrast, a straightforward route was presented through exploration without the retrosynthesis knowledge (Fig. 3.7b). Friedel-Crafts acylation of trihalopyridine **14** with 2-(chlorocarbonyl)benzoic acid (**13**) would give 2-(pyridinecarbonyl)benzoic acid **15**, which is further reacted with aminoalcohol **11** to afford tricyclic ketone **16**. Construction of the phthalazine ring could be performed by the reaction of **16** with *p*-bromobenzylhydrazine (**17**) to yield the precursor **18**. Finally, the introduction of a nitrile group into the benzyl group of **18** would provide the target molecule **1**. Additional demonstrations for two other drug-like molecules[31, 32] are shown in Fig. 3.8 and 3.9.

## a) With retrosynthesis knowledge



## b) Without retrosynthesis knowledge

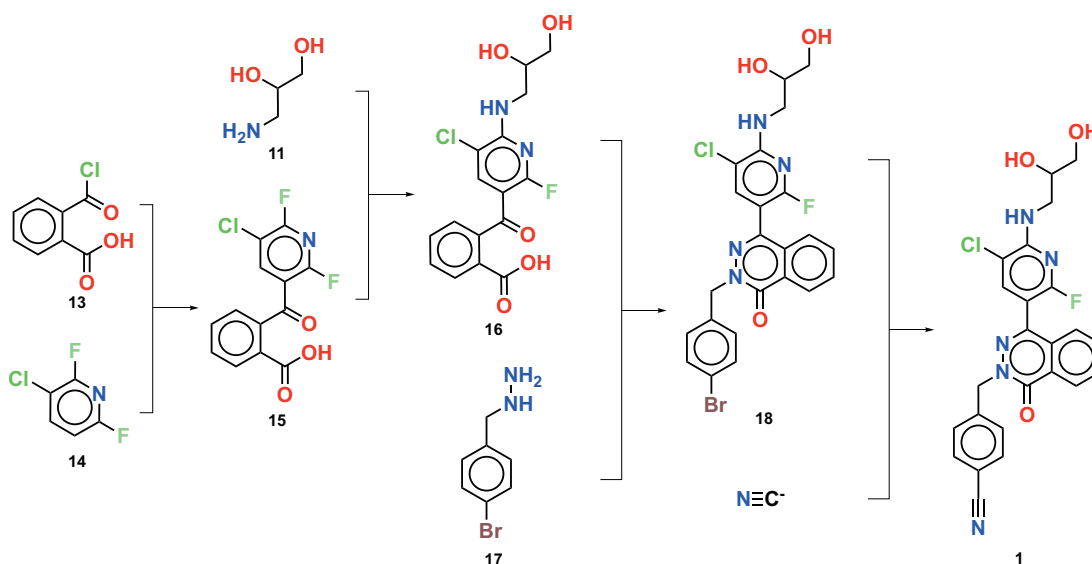


Figure 3.7: Comparison of (a) the synthetic route for a target compound (hepatitis B virus capsid inhibitor)[30] found by ReTReK with retrosynthesis knowledge and (b) a corresponding route found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. Molecule **1** is the target.







Furthermore, the effectiveness of retrosynthesis knowledge was confirmed in several cases of retrosynthetic analyses. In these cases, ReTReK with retrosynthesis knowledge succeeded in finding retrosynthetic routes to the target molecules, whereas no routes were found using ReTReK without retrosynthesis knowledge (Fig. 3.10, 3.11, and 3.12). Figure 3.10 shows the retrosynthetic route to a molecule **19**, known as a *Mycobacterium tuberculosis* thymidylate kinase (MtbTMPK) inhibitor[33], as a representative example. In the suggested route, **19** is disconnected in the center of the molecule, giving imidazo[1,2-a]pyridine-3-carboxamide **25** and 1-(piperidin-4-yl)pyrimidine-2,4-dione **29**. Compound **25** would be obtained from dibromide **24** via a selective S<sub>N</sub>Ar reaction with an organometallic reagent, such as ethylmagnesium bromide, which is prepared from ethyl bromide. Dibromide **24** can be obtained from benzylamide **22** via stepwise S<sub>E</sub>Ar bromination. Amide **22** would be provided from imidazopyridine-3-carboxylic acid (**20**) and N-Boc-benzylamine (**21**). Another intermediate **29** would be synthesized from 4-iodopiperidine (**27**) with pyrimidine-2,4-dione **28** by S<sub>N</sub>2 reaction. **27** would be easily prepared from N-Boc **26**. From the viewpoint of the practical synthesis, deprotection of N-Boc group of the piperidine ring should be performed after N-alkylation of **28** with **26** to avoid oligomerization of **27** by self N-alkylation. Additional demonstrations of other two drug-like molecules[34, 35] are shown in Fig. 3.11, and 3.12.

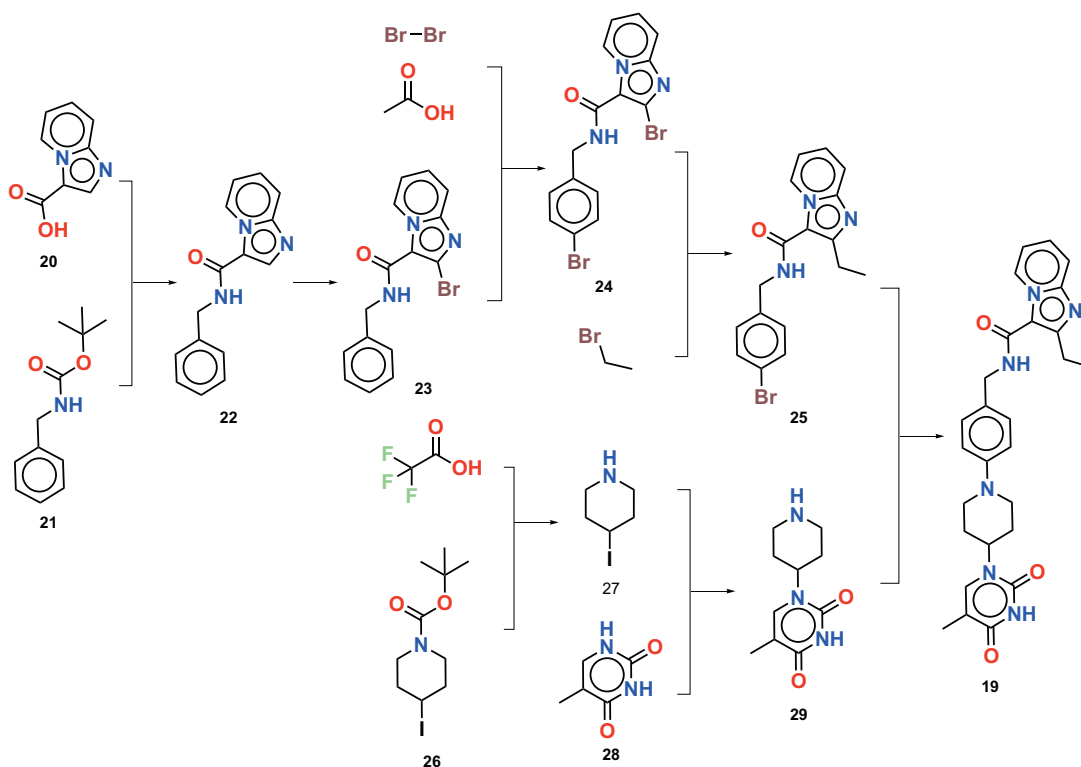
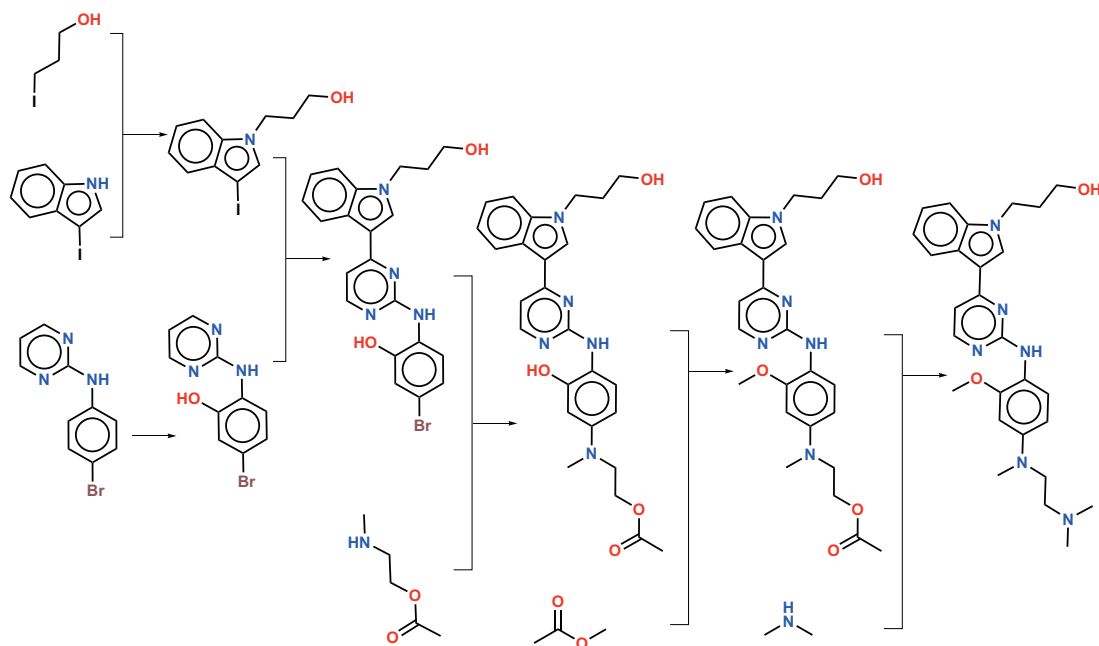
**a) With retrosynthesis knowledge****b) Without retrosynthesis knowledge****Not Found**

Figure 3.10: For the target compound (MtbTMPK inhibitor)[33] considered here, (a) a synthetic route was found using ReTReK with retrosynthesis knowledge, whereas (b) no synthetic route was found without retrosynthesis knowledge. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively. Molecule **19** is the target.



### a) With retrosynthesis knowledge



### b) Without retrosynthesis knowledge

**Not Found**

Figure 3.12: For the target compound (EGFR kinase inhibitor)[35] considered here, (a) a synthetic route was found by ReTReK with retrosynthesis knowledge, whereas (b) ReTReK without retrosynthesis knowledge did not find any synthetic route. The weight parameters of the retrosynthesis knowledge scores,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , were set to 5.0, 2.0, 0.5, and 2.0, respectively.

Overall, the results demonstrate that retrosynthesis knowledge effectively contributes to retrosynthetic analyses using ReTReK. Considering these demonstrations, the ReTReK framework with integrated the retrosynthesis knowledge has the potential to further improve the performance of data-driven CASP applications. In future work, I plan to conduct an AB test with chemists to show the effectiveness of the retrosynthesis knowledge in a statistical way using a large variety of molecules.

## 4 Conclusion

I developed ReTReK, a hybrid CASP application combining data-driven and rule-based techniques that can flexibly reflect and apply retrosynthesis knowledge. Through an evaluation of ReTReK with and without retrosynthesis knowledge, I showed that the integration of such knowledge into data-driven CASP applications helps improve their performance and enhance the quality of the explored synthetic routes. I expect the concept of ReTReK to contribute to further developments and improvements in data-driven CASP applications; furthermore, it is expected to bridge the current gap between applications and chemists by allowing chemists' ways of thinking to be interactively introduced into CASP systems.

To allow more realistic and preferable synthetic routes to be obtained in the future, I will address the further development of automatic reaction template extraction methods while maintaining chemical integrity. In this study, orphan atoms (atoms appearing on only one side of the reaction arrow) were included in the reaction templates in order to automatically retain protecting and leaving groups in the templates because these groups are often not recorded as reactants or products. Considering that such groups have been manually defined in a previous study[107], this template definition is expected to contribute to the further development of the template extraction methods. In future work, I will also focus on integrating chemical knowledge, such as electronic and quantum-mechanical aspects of chemical reactions, into the template. As a solution for describing such information in reaction templates, Chemical Terms—one of the functions of ChemAxon—may be considered. In addition, I may define additional retrosynthesis knowledge scores to allow ReTReK to represent chemists' ways of thinking more extensively than in the current model. Furthermore, to ensure the ease of use of ReTReK, I will prepare a user-friendly interactive interface with functions such as range sliders for adjusting each retrosynthesis knowledge score and other tools for displaying explored synthetic routes.



# Conclusion

In this thesis, I achieved (1) the application of GCN to the retrosynthetic reaction prediction task, (2) showing that the GCN model for the task recognized the reaction-related atoms of a molecule using IG through the quantitative evaluation, and (3) the development of a data-driven CASP application considering retrosynthesis knowledge using MCTS and GCN techniques as well as four retrosynthesis knowledge scores.

In Chapter 1, I developed a graph-based deep learning platform, kGCN, to enable users with various levels of programming skill to use GNN for various prediction tasks related to chemistry. Additionally, I implemented the explainable AI technique IG, which allows us to calculate the relationships between model predictions and the features of organic molecules. In Chapter 2, I successfully developed a GCN-based interpretable retrosynthetic reaction prediction model using IG. The prediction accuracy of my GCN model was higher than that of a conventional (ECFP) model, and the GCN predictions were less influenced by dataset bias, which is a beneficial characteristic for a data-driven CASP approach. Visualizations of the GCN predictions using IG successfully showed the contributions of individual atoms to the results of retrosynthetic reaction prediction. Based on such visualizations, my system is expected to aid chemist's understanding of data-driven retrosynthetic reaction prediction models. In Chapter 3, I developed ReTReK, a data-driven CASP application integrating retrosynthesis knowledge. Experimental results showed that the consideration of retrosynthesis knowledge successfully increases the solution performance and can guide the search directions in MCTS according to the characteristics of each type of knowledge. Through an evaluation of ReTReK with and without retrosynthesis knowledge, I showed that the concept of integrating such knowledge into data-driven CASP applications can improve their performance and enhance the quality of the explored synthetic routes. Furthermore, I clearly demonstrated that synthetic routes found with retrosynthesis knowledge were preferable to those found without such knowledge.

I believe that the results obtained through the work presented in this thesis will contribute to the development of more practical data-driven CASP approaches and furthermore provide

#### 4. CONCLUSION

---

important concepts, such as the integration of retrosynthesis knowledge in the form of adjustable parameters into a CASP system, to bridge the gap between chemists and data-driven approaches. I also believe that pure data-driven AI will be able to discuss with chemists how to synthesize molecules in their daily work within several decades. To realize such AI, I think there is a need for a better understanding of how AI learns and recognizes chemistry and the development of interfaces between chemists and AI to communicate with each other, as well as we should improve AI performances of chemical synthesis planning more and more. In the hope of developing such AI, I will continue to address and contribute these attractive themes.



---

## List of Publications

### Journal articles related to this thesis

- S. Ishida, K. Terayama, R. Kojima, K. Takasu, and Y. Okuno, “Prediction and Interpretable Visualization of Retrosynthetic Reactions Using Graph Convolutional Networks,” *Journal of Chemical Information and Modeling*, vol. 59, no. 12, pp. 5026–5033, Nov. 2019.
- R. Kojima, S. Ishida, M. Ohta, H. Iwata, T. Honma, and Y. Okuno, “kGCN: a graph-based deep learning framework for chemical structures,” *Journal of Cheminformatics*, vol. 12, pp. 32, May. 2020.

### Preprint related to this thesis

- S. Ishida, K. Terayama, R. Kojima, K. Takasu, and Y. Okuno, “AI-Driven Synthetic Route Design with Retrosynthesis Knowledge,” *ChemRxiv*, Dec. 2020.



## References

- [1] K. C. Nicolaou, D. Vourloumis, N. Winssinger, and P. S. Baran, "The art and science of total synthesis at the dawn of the twenty-first century," *Angewandte Chemie International Edition*, vol. 39, pp. 44–122, Jan. 2000.
- [2] E. J. Corey, M. Ohno, R. B. Mitra, and P. A. Vatakencherry, "Total synthesis of longifolene," *Journal of the American Chemical Society*, vol. 86, pp. 478–485, Feb. 1964.
- [3] E. J. Corey, "The logic of chemical synthesis: Multistep synthesis of complex carbogenic molecules (nobel lecture)," *Angewandte Chemie International Edition in English*, vol. 30, pp. 455–465, 5 1991.
- [4] E. J. Corey and X. M. Cheng, *The Logic of Chemical Synthesis*. New York: Wiley, 1995.
- [5] S. D. Satyanarayanajois and R. A. Hill, "Medicinal chemistry for 2020," *Future Medicinal Chemistry*, vol. 3, pp. 1765–1786, Oct. 2011.
- [6] E. J. Corey and W. T. Wipke, "Computer-assisted design of complex organic syntheses," *Science*, vol. 166, pp. 178–192, 10 1969.
- [7] E. J. Corey, R. D. Cramer, and W. J. Howe, "Computer-assisted Synthetic Analysis for Complex Molecules. Methods and Procedures for Machine Generation of Synthetic Intermediates," *Journal of the American Chemical Society*, vol. 94, pp. 440–459, 1972.
- [8] W. Wipke, G. I. Ouchi, and S. Krishnan, "Simulation and evaluation of chemical synthesis-SECS: An application of artificial intelligence techniques," *Artificial Intelligence*, vol. 11, pp. 173–193, 1978.
- [9] E. Corey, A. Long, and S. Rubenstein, "Computer-assisted analysis in organic synthesis," *Science*, vol. 228, pp. 408–418, Apr. 1985.

- [10] J. B. Hendrickson and A. G. Toczko, "SYNGEN program for synthesis design: basic computing techniques," *Journal of Chemical Information and Modeling*, vol. 29, pp. 137–145, Aug. 1989.
- [11] W.-D. Ihlenfeldt and J. Gasteiger, "Computer-assisted planning of organic syntheses: The second generation of programs," *Angewandte Chemie International Edition in English*, vol. 34, pp. 2613–2633, Jan. 1996.
- [12] K. Funatsu and S.-I. Sasaki, "Computer-assisted organic synthesis design and reaction prediction system, "AIPHOS"," *Tetrahedron Computer Methodology*, vol. 1, pp. 27–37, 1988.
- [13] K. Satoh and K. Funatsu, "A novel approach to retrosynthetic analysis using knowledge bases derived from reaction databases," *Journal of Chemical Information and Computer Sciences*, vol. 39, pp. 316–325, Mar. 1999.
- [14] J. Law, Z. Zsoldos, A. Simon, D. Reid, Y. Liu, S. Y. Khew, A. P. Johnson, S. Major, R. A. Wade, and H. Y. Ando, "Route designer: A retrosynthetic analysis tool utilizing automated retrosynthetic rule generation," *Journal of Chemical Information and Modeling*, vol. 49, pp. 593–602, Feb. 2009.
- [15] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, and B. A. Grzybowski, "Computer-Assisted Synthetic Planning: The End of the Beginning," *Angewandte Chemie, International Edition*, vol. 55, pp. 5904–5937, 2016.
- [16] M. H. S. Segler, M. Preuss, and M. P. Waller, "Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI," *Nature*, vol. 555, pp. 604–610, 2018.
- [17] C. W. Coley, W. H. Green, and K. F. Jensen, "Machine Learning in Computer-Aided Synthesis Planning," *Accounts of Chemical Research*, vol. 51, pp. 1281–1289, 2018.
- [18] J. S. Schreck, C. W. Coley, and K. J. M. Bishop, "Learning Retrosynthetic Planning through Simulated Experience," *ACS Central Science*, vol. 5, no. 6, pp. 970–981, 2019.
- [19] P. Judson, *Knowledge-based Expert Systems in Chemistry*. Theoretical and Computational Chemistry Series, The Royal Society of Chemistry, 2019.
- [20] T. Klucznik, B. Mikulak-Klucznik, M. P. McCormack, H. Lima, S. Szymkuć, M. Bhowmick, K. Molga, Y. Zhou, L. Rickershauser, and B. A. Grzybowski, "Efficient Syntheses of Diverse, Medicinally Relevant Targets Planned by Computer and Executed in the Laboratory," *Chem*, vol. 4, pp. 522–532, 2018.

- [21] H. Moriwaki, Y.-S. Tian, N. Kawashita, and T. Takagi, "Mordred: a molecular descriptor calculator," *Journal of Cheminformatics*, vol. 10, Feb. 2018.
- [22] D. Rogers and M. Hahn, "Extended-Connectivity Fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, pp. 742–754, 2010.
- [23] D. Weininger, "SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of Chemical Information and Modeling*, vol. 28, pp. 31–36, Feb. 1988.
- [24] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *Journal of Computer-Aided Molecular Design*, vol. 30, pp. 595–608, Aug. 2016.
- [25] A. F. de Almeida, R. Moreira, and T. Rodrigues, "Synthetic organic chemistry driven by artificial intelligence," *Nature Reviews Chemistry*, vol. 3, pp. 589–604, Aug. 2019.
- [26] F. Strieth-Kalthoff, F. Sandfort, M. H. S. Segler, and F. Glorius, "Machine learning the ropes: principles, applications and directions in synthetic chemistry," *Chemical Society Reviews*, vol. 49, no. 17, pp. 6154–6168, 2020.
- [27] P. M. Pflüger and F. Glorius, "Molecular machine learning: The future of synthetic chemistry?," *Angewandte Chemie International Edition*, vol. 59, pp. 18860–18865, Sept. 2020.
- [28] C. W. Coley, D. A. Thomas, J. A. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, R. W. Hicklin, P. P. Plehiers, J. Byington, J. S. Piotti, W. H. Green, A. J. Hart, T. F. Jamison, and K. F. Jensen, "A robotic platform for flow synthesis of organic compounds informed by ai planning," *Science*, vol. 365, p. 6453, 8 2019.
- [29] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano, and T. Laino, "Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy," *Chemical Science*, vol. 11, no. 12, pp. 3316–3325, 2020.
- [30] W. Chen, F. Liu, Q. Zhao, X. Ma, D. Lu, H. Li, Y. Zeng, X. Tong, L. Zeng, J. Liu, L. Yang, J. Zuo, and Y. Hu, "Discovery of phthalazinone derivatives as novel hepatitis b virus capsid inhibitors," *Journal of Medicinal Chemistry*, vol. 63, pp. 8134–8145, July 2020.

- [31] G. Tsuji, M. Yusa, S. Masada, H. Yokoo, J. Hosoe, T. Hakamatsuka, Y. Demizu, and N. Uchiyama, "Facile synthesis of kwakhurin, a marker compound of *pueraria mirifica* and its quantitative NMR analysis for standardization as a reagent," *Chemical and Pharmaceutical Bulletin*, vol. 68, pp. 797–801, Aug. 2020.
- [32] M. C. Pismataro, N. A. Horenstein, C. Stokes, M. Quadri, M. D. Amici, R. L. Papke, and C. Dallanocce, "Design, synthesis, and electrophysiological evaluation of NS6740 derivatives: Exploration of the structure-activity relationship for  $\alpha 7$  nicotinic acetylcholine receptor silent activation," *European Journal of Medicinal Chemistry*, vol. 205, p. 112669, Nov. 2020.
- [33] Y. Jian, R. Merceron, S. D. Munck, H. E. Forbes, F. Hulpia, M. D. Risseeuw, K. V. Hecke, S. N. Savvides, H. Munier-Lehmann, H. Boshoff, and S. V. Calenbergh, "Endeavors towards transformation of m. tuberculosis thymidylate kinase (MtbTMPK) inhibitors into potential antimycobacterial agents," *European Journal of Medicinal Chemistry*, vol. 206, p. 112659, Nov. 2020.
- [34] T. P. Banzato, J. R. Gubiani, D. I. Bernardi, C. R. Nogueira, A. F. Monteiro, F. F. Juliano, S. M. de Alencar, R. A. Pilli, C. A. de Lima, G. B. Longato, A. G. Ferreira, M. A. Foglio, J. E. de Carvalho, D. B. Vendramini-Costa, and R. G. S. Berlinck, "Antiproliferative flavanoid dimers isolated from brazilian red propolis," *Journal of Natural Products*, vol. 83, pp. 1784–1793, June 2020.
- [35] Z. Su, T. Yang, J. Wang, M. Lai, L. Tong, G. Wumaier, Z. Chen, S. Li, H. Li, H. Xie, and Z. Zhao, "Design, synthesis and biological evaluation of potent EGFR kinase inhibitors against 19d/t790m/c797s mutation," *Bioorganic & Medicinal Chemistry Letters*, vol. 30, p. 127327, Aug. 2020.
- [36] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. v. d. Laak, B. v. Ginneken, and C. I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
- [37] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, and C. S. Greene, "Opportunities and Obstacles for Deep Learning in Biology and Medicine," *Journal of the Royal Society, Interface*, vol. 15, 2018. 20170387.
- [38] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *Computing Research Repository (CoRR)*, vol. abs/1609.08144, 2016.

- [39] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep Learning in Agriculture: A Survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [40] E. Gawehn, J. A. Hiss, and G. Schneider, “Deep learning in drug discovery,” *Molecular Informatics*, vol. 35, no. 1, pp. 3–14, 2016.
- [41] G. B. Goh, N. O. Hodas, and A. Vishnu, “Deep learning for computational chemistry,” *Journal of computational chemistry*, vol. 38, no. 16, pp. 1291–1307, 2017.
- [42] D. C. Elton, Z. Boukouvalas, M. D. Fuge, and P. W. Chung, “Deep learning for molecular design - a review of the state of the art,” *Molecular Systems Design and Engineering*, vol. 4, no. 4, pp. 828–849, 2019.
- [43] W. Torng and R. B. Altman, “Graph convolutional neural networks for predicting drug-target interactions,” *Journal of Chemical Information and Modeling*, vol. 59, no. 10, pp. 4131–4149, 2019.
- [44] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, “Deep neural nets as a method for quantitative structure–activity relationships,” *Journal of Chemical Information and Modeling*, vol. 55, no. 2, pp. 263–274, 2015.
- [45] M. H. S. Segler and M. P. Waller, “Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction,” *Chemistry - A European Journal*, vol. 23, pp. 5966–5971, 2017.
- [46] S. Schneckener, S. Grimbs, J. Hey, S. Menz, M. Osmers, S. Schaper, A. Hillisch, and A. H. Göller, “Prediction of oral bioavailability in rats: Transferring insights from in vitro correlations to (deep) machine learning models using in silico model outputs and chemical structure parameters,” *Journal of Chemical Information and Modeling*, vol. 59, no. 11, pp. 4893–4905, 2019.
- [47] J. K. Wegner, A. Sterling, R. Guha, A. Bender, J.-L. Faulon, J. Hastings, N. O’Boyle, J. Overington, H. Van Vlijmen, and E. Willighagen, “Cheminformatics,” *Communications of the ACM*, vol. 55, no. 11, pp. 65–75, 2012.
- [48] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, “Molecular graph convolutions: moving beyond fingerprints,” *Journal of computer-aided molecular design*, vol. 30, no. 8, pp. 595–608, 2016.
- [49] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1263–1272, 2017.

- [50] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems 28*, pp. 2224–2232, 2015.
- [51] W. Jin, C. Coley, W. R. Barzilay, and T. Jaakkola, "Predicting organic reaction outcomes with weisfeiler-lehman network," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2604–2613, 2017.
- [52] M. R. Berthold, N. Cebon, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "Knime - the konstanz information miner: Version 2.0 and beyond," *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 26–31, Nov. 2009.
- [53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [54] M. Hamanaka, K. Taneishi, H. Iwata, J. Ye, J. Pei, J. Hou, and Y. Okuno, "CGBVS-DNN: Prediction of Compound-protein Interactions Based on Deep Learning," *Molecular informatics*, vol. 36, no. 1-2, p. 1600045, 2017.
- [55] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "GraphDTA: Predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, Oct. 2020.
- [56] M. Tsubaki, K. Tomii, and J. Sese, "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences," *Bioinformatics*, vol. 35, no. 2, pp. 309–318, 2019.
- [57] B. Ramsundar, B. Liu, Z. Wu, A. Verras, M. Tudor, R. P. Sheridan, and V. Pande, "Is multitask deep learning practical for pharma?," *Journal of Chemical Information and Modeling*, vol. 57, no. 8, pp. 2068–2076, 2017.
- [58] S. Sanyal, J. Balachandran, N. Yadati, A. Kumar, P. Rajagopalan, S. Sanyal, and P. Talukdar, "MT-CGCNN: Integrating crystal graph convolutional neural network with multitask learning for material property prediction," *arXiv preprint arXiv:1811.05660*, 2018.
- [59] K. Liu, X. Sun, L. Jia, J. Ma, H. Xing, J. Wu, H. Gao, Y. Sun, F. Boulnois, and J. Fan, "Chemi-net: A molecular graph convolutional network for accurate drug property prediction," *International Journal of Molecular Sciences*, vol. 20, no. 14, p. 3389, 2019.



- [60] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.
- [61] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” *arXiv preprint arXiv:1706.03825*, 2017.
- [62] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3319–3328, JMLR. org, 2017.
- [63] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2951–2959, 2012.
- [64] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 265–283, 2016.
- [65] B. Ramsundar, P. Eastman, P. Walters, and V. Pande, *Deep Learning for the Life Sciences*. Sebastopol: O’Reilly Media inc., 2019.
- [66] Preferred Networks, “Chainer chemistry: A library for deep learning in biology and chemistry.” <https://github.com/chainer/chainer-chemistry> (Accessed January 19, 2021).
- [67] M. Popova, “Openchem: Deep learning toolkit for computational chemistry and drug design,” <https://github.com/Mariawelt/OpenChem> (Accessed January 19, 2021).
- [68] S. Tokui, K. Oono, S. Hido, and J. Clayton, “Chainer: a next-generation open source framework for deep learning,” in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, vol. 5, pp. 1–6, 2015.
- [69] G. Landrum, “RDKit: Open-source Cheminformatics,” 2018. <http://www.rdkit.org> (Accessed January 19, 2021).
- [70] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

- [71] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS Central Science*, vol. 3, pp. 283–293, 2017.
- [72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, pp. 448–456, 2015.
- [73] F. Montanari, L. Kuhnke, A. Laak, Ter, and D.-A. Clevert, "Modeling physico-chemical admet endpoints with multitask graph convolutional networks," *Molecules*, vol. 25, no. 1, p. 44, 2020.
- [74] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. <https://www.tensorflow.org/> (Accessed January 19, 2021).
- [75] F. Chollet *et al.*, "Keras." <https://keras.io>, 2015. (Accessed January 19, 2021).
- [76] The GPyOpt authors, "Gpyopt: A bayesian optimization framework in python," 2016. <http://github.com/SheffieldML/GPyOpt> (Accessed January 19, 2021).
- [77] M. J. Keiser, V. Setola, J. J. Irwin, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, *et al.*, "Predicting new molecular targets for known drugs," *Nature*, vol. 462, no. 7270, pp. 175–181, 2009.
- [78] A. Gimeno, R. Beltrán-Debón, M. Mulero, G. Pujadas, and S. Garcia-Vallvé, "Understanding the variability of the S1' pocket to improve matrix metalloproteinase inhibitor selectivity profiles," *Drug Discovery Today*, vol. 25, no. 1, pp. 38–57, 2020.
- [79] A. Mauri, V. Consonni, M. Pavan, and R. Todeschini, "Dragon software: An easy approach to molecular descriptor calculations," *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 56, no. 2, pp. 237–248, 2006.
- [80] P. Zhang, L. Tao, X. Zeng, C. Qin, S. Chen, F. Zhu, Z. Li, Y. Jiang, W. Chen, and Y.-Z. Chen, "A protein network descriptor server and its use in studying protein, disease, metabolic and drug targeted networks," *Briefings in Bioinformatics*, vol. 18, no. 6, pp. 1057–1070, 2016.
- [81] A. Rossello, E. Nuti, P. Carelli, E. Orlandini, M. Macchia, S. Nencetti, M. Zandomenighi, F. Balzano, G. U. Barretta, A. Albini, R. Benelli, G. Cercignani, G. Murphy, and A. Balsamo, "Ni-propoxy-n-biphenylsulfonaminobutylhydroxamic acids

- as potent and selective inhibitors of mmp-2 and mt1-mmp,” *Bioorganic & Medicinal Chemistry Letters*, vol. 15, no. 5, pp. 1321–1326, 2005.
- [82] C. Antoni, L. Vera, L. Devel, M. P. Catalani, B. Czarny, E. Cassar-Lajeunesse, E. Nuti, A. Rossello, V. Dive, and E. A. Stura, “Crystallization of bi-functional ligand protein complexes,” *Journal of Structural Biology*, vol. 182, no. 3, pp. 246–254, 2013.
- [83] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen, “Prediction of Organic Reaction Outcomes Using Machine Learning,” *ACS Central Science*, vol. 3, pp. 434–443, 2017.
- [84] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen, “Computer-Assisted Retrosynthesis Based on Molecular Similarity,” *ACS Central Science*, vol. 3, pp. 1237–1245, 2017.
- [85] B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. L. Nguyen, S. Ho, J. Sloane, P. Wender, and V. Pande, “Retrosynthetic Reaction Prediction using Neural Sequence-to-Sequence Models,” *ACS Central Science*, vol. 3, pp. 1103–1113, 2017.
- [86] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay, and K. F. Jensen, “A graph-convolutional neural network model for the prediction of chemical reactivity,” *Chemical Science*, vol. 10, pp. 370–377, 2018.
- [87] K. Lin, Y. Xu, J. Pei, and L. Lai, “Automatic retrosynthetic route planning using template-free models,” *Chemical Science*, vol. 11, no. 12, pp. 3355–3364, 2020.
- [88] P. Karpov, G. Godin, and I. V. Tetko, “A transformer model for retrosynthesis,” in *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*, (Cham), pp. 817–830, Springer International Publishing, 2019.
- [89] S. Zheng, J. Rao, Z. Zhang, J. Xu, and Y. Yang, “Predicting retrosynthetic reactions using self-corrected transformer neural networks,” *Journal of Chemical Information and Modeling*, vol. 60, pp. 47–55, Dec. 2019.
- [90] Y. Hu, E. Loukine, and J. Bajorath, “Improving the Search Performance of Extended Connectivity Fingerprints through Activity-Oriented Feature Filtering and Application of a Bit-Density-Dependent Similarity Function,” *ChemMedChem*, vol. 4, pp. 540–548, 2009.
- [91] J. L. Baylon, N. A. Cilfone, J. R. Gulcher, and T. W. Chittenden, “Enhancing Retrosynthetic Reaction Prediction with Deep Learning using Multiscale Reaction Classification,” *Journal of Chemical Information and Modeling*, vol. 59, pp. 1–16, 2019.

- [92] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in *Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144, 2016.
- [93] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning (ICML)*, 2017.
- [94] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.
- [95] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, pp. 818–833, 2014.
- [96] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Computing Research Repository (CoRR)*, vol. abs/1312.6034, 2013.
- [97] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, Oct. 2017.
- [98] D. Lowe, "Chemical Reactions from US Patents (1976-Sep2016)," 6 2017.
- [99] S. Avramova, N. Kochev, and P. Angelov, "RetroTransformDB: A Dataset of Generic Transforms for Retrosynthetic Analysis," *Data*, vol. 3, 2018. 14.
- [100] "Chemaxon." <http://www.chemaxon.com>, 2021. (Accessed January 19, 2021).
- [101] L. Mosley, *A balanced approach to the multi-class imbalance problem*. PhD thesis, Iowa State University, 2013.
- [102] K. Ishiguro, S. ichi Maeda, and M. Koyama, "Graph warp module: an auxiliary module for boosting the power of graph neural networks," *ArXiv*, vol. abs/1902.01020, 2019.
- [103] B. Mikulak-Klucznik, P. Gołębiowska, A. A. Bayly, O. Popik, T. Klucznik, S. Szymkuć, E. P. Gajewska, P. Dittwald, O. Staszewska-Krajewska, W. Beker, T. Badowski, K. A. Scheidt, K. Molga, J. Młynarski, M. Mrksich, and B. A. Grzybowski, "Computational planning of the synthesis of complex natural products," *Nature*, Oct. 2020.

- [104] M. A. Kayala, C.-A. Azencott, J. H. Chen, and P. Baldi, "Learning to predict chemical reactions," *Journal of Chemical Information and Modeling*, vol. 51, pp. 2209–2222, Sept. 2011.
- [105] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 5 2015.
- [106] "Reaxys," 2021. <https://www.reaxys.com/> (Accessed January 19, 2021).
- [107] C. W. Coley, W. H. Green, and K. F. Jensen, "RDChiral: An RDKit wrapper for handling stereochemistry in retrosynthetic template extraction and application," *Journal of Chemical Information and Modeling*, vol. 59, pp. 2529–2537, June 2019.
- [108] C. Coley, "connorcoley/askcos: First public release of askcos," June 2019.
- [109] S. Genheden, A. Thakkar, V. Chadimova, J.-L. Reymond, O. Engkvist, and E. J. Bjerrum, "AiZynthFinder: A fast robust and flexible open-source software for retrosynthetic planning," *ChemRxiv*, June 2020.
- [110] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, "Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction," *ACS Central Science*, vol. 5, pp. 1572–1583, Aug. 2019.
- [111] R. Shibukawa, S. Ishida, K. Yoshizoe, K. Wasa, K. Takasu, Y. Okuno, K. Terayama, and K. Tsuda, "CompRet: a comprehensive recommendation framework for chemical synthesis planning with algorithmic enumeration," *Journal of Cheminformatics*, vol. 12, p. 52, Sept. 2020.
- [112] J. Bradshaw, B. Paige, M. Kusner, M. Segler, and J. M. Hernández-Lobato, "Barking up the right tree: an approach to search over molecule synthesis dags," in *Advances in Neural Information Processing Systems 33*, Dec. 2020.
- [113] I. V. Tetko, P. Karpov, R. V. Deursen, and G. Godin, "State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis," *Nature Communications*, vol. 11, p. 5575, Nov. 2020.
- [114] A. Thakkar, T. Kogej, J.-L. Reymond, O. Engkvist, and E. J. Bjerrum, "Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain," *Chemical Science*, vol. 11, no. 1, pp. 154–168, 2020.
- [115] X. Wang, Y. Qian, H. Gao, C. W. Coley, Y. Mo, R. Barzilay, and K. F. Jensen, "Towards efficient discovery of green synthetic pathways with monte carlo tree search

- and reinforcement learning,” *Chemical Science*, vol. 11, no. 40, pp. 10959–10972, 2020.
- [116] A. Kishimoto, B. Buesser, B. Chen, and A. Botea, “Depth-first proof-number search with heuristic edge cost and application to chemical synthesis planning,” in *Advances in Neural Information Processing Systems*, pp. 7224–7234, 2019.
- [117] D. Mendez, A. Gaulton, A. P. da Costa Bento, J. Chambers, M. D. Veij, E. Félix, M. a Paula Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, M. a Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. J. Radoux, A. Segura-Cabrera, A. Hersey, and A. R. Leach, “ChEMBL: Towards direct deposition of bioassay data,” *Nucleic Acids Research*, vol. 47, pp. D930–D940, 1 2019.
- [118] T. Sterling and J. J. Irwin, “ZINC 15 – ligand discovery for everyone,” *Journal of Chemical Information and Modeling*, vol. 55, pp. 2324–2337, Nov. 2015.
- [119] R. P. Sheridan, “Time-split cross-validation as a method for estimating the goodness of prospective prediction,” *Journal of Chemical Information and Modeling*, vol. 53, pp. 783–790, Apr. 2013.
- [120] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, “A survey of monte carlo tree search methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, pp. 1–43, 2012.