GNN-Retro: Retrosynthetic Planning with Graph Neural Networks

Peng Han[*1,2,3], **Peilin Zhao**[*4], **Chan Lu**[4], **Junzhou Huang**[4], **Jiaxiang Wu**[4,], **Shuo Shang**[†1], **Bin Yao**[5], **Xiangliang Zhang**[†6,2]

[1] University of Electronic Science and Technology of China
[2] King Abdullah University of Science and Technology
[3] Aalborg University
[4] Tencent AI Lab
[5] Shanghai Jiao Tong University
[6] University of Notre Dame

peng.han@kaust.edu.sa, masonzhao@tencent.com, sherryllu@tencent.com, joehhuang@tencent.com, jonathanwu@tencent.com, jedi.shang@gmail.com, yaobin@cs.sjtu.edu.cn, xzhang33@nd.edu

Abstract

Retrosynthetic planning plays an important role in the field of organic chemistry, which could generate a synthetic route for the target product. The synthetic route is a series of reactions which are started from the available molecules. The most challenging problem in the generation of the synthetic route is the large search space of the candidate reactions. Estimating the cost of candidate reactions has been proved effectively to prune the search space, which could achieve a higher accuracy with the same search iteration. And the estimation of one reaction is comprised of the estimations of all its reactants. So, how to estimate the cost of these reactants will directly influence the quality of results. To get a better performance, we propose a new framework, named GNN-Retro, for retrosynthetic planning problem by combining graph neural networks (GNN) and the latest search algorithm. The structure of GNN in our framework could incorporate the information of neighboring molecules, which will improve the estimation accuracy of our framework. The experiments on the USPTO dataset show that our framework could outperform the state-of-the-art methods with a large margin under the same settings.

Introduction

Retrosynthetic planning (Schreck, Coley, and Bishop 2019; Segler and Waller 2017; Zheng et al. 2020; Li et al. 2020; Schwaller et al. 2020; Mao et al. 2021; Yan et al. 2020) plays an important role in the filed of chemical applications (Wang et al. 2021), which could generate a synthetic route for the target molecule. The synthetic route is a series of reactions which are based on available molecules. The example in Fig. 1 shows the components of one synthetic route, where one target molecule or intermediate molecule is obtained by one reaction and one reaction contains multiple molecules. The most challenging problem in the generation of the synthetic route is the huge numbers of candidate reactions for the target product and intermediate products. Because larger
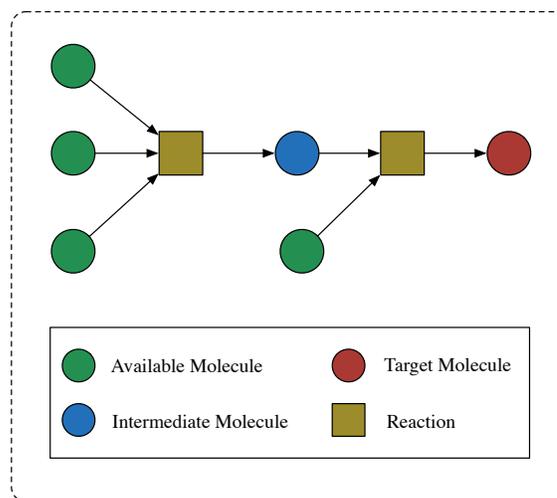

Figure 1: Schematic diagram of one synthetic route

numbers of the candidate reactions will bring exponential growth for the search space.

Estimating the cost of candidate reactions has been proved effectively (Chen et al. 2020) to prune the search space, which could achieve a higher success rate with the same search iteration. And the estimation of one reaction is comprised of the estimations of all its reactants. So, how to estimate the cost of these reactants will directly influence the quality of results. The difficulty of the estimation is that the data space of all molecules is extremely large and the existing training dataset can only cover a small part of it.

Graph neural network (GNN) (Kipf and Welling 2017; Yang et al. 2021) is popular recently, for its high performance in the task of semi-supervision. The main difference between the computation of GNN and the computation of traditional neural networks is that, GNN will directly and explicitly incorporate the information of neighbors in the training and testing process. The computation of GNN could reduce the data sparsity problem when the connection between samples is relevant to their labels.

In order to make GNN applicable in our task to reduce the sparsity problem of training dataset, we have to solve two problems firstly. How to construct the graph for the GNN under the environment of our task is the most important problem. Because the graph in GNN, which determines the relationship between samples, will influence the performance directly. The other problem is that, we cannot get all molecules to construct the global graph before the training process. And all intermediate molecules in the evaluation and test process are unavailable before we expand the search route.

The construction of graph for GNN should be highly related to the aim of the application. The goal of the estimation is to predict the synthetic cost of the given molecules. So connected molecules in the constructed graph are better to have similar synthetic cost. The synthetic cost of one molecule is mainly determined by its composed atoms and structure. To describe the atoms and structures of given molecule, Morgan fingerprint (Rogers and Hahn 2010) is proposed as the representation of the molecule. So, we assume that molecules with similar fingerprints have similar synthetic cost. Under our assumption, cosine similarity of the fingerprints is used as the metric to describe the relationship between molecules. However, these similarities cannot be directly used as the weights of the graph. Because this will make the graph too dense, which will incur much more noise than meaningful information. To fix this, we construct the graph with a hand-crafted threshold, which is tuneable and can filter most of noise. Then, only molecules with similarities larger than the threshold could be connected in the graph and the weight is set as the corresponding similarity. However, the mentioned graph only considers the information of fingerprint which cannot directly reflect the relationship of synthetic cost between molecules. To make the graph able to directly reflect the relationship of synthetic cost and contain the information of fingerprint, we propose an alternative way to construct it by embedding method.

The other challenge tampers the utilization of GNN is that we cannot build the graph for all molecules. The main reason is that the data space of all molecules is extremely large, so we cannot build a global graph before the training process. Moreover, even we only want to construct graphs for all molecules used in the training, evaluation and testing process, we cannot achieve it. Because the search process is dynamic and is related to the estimation function itself. So we cannot get all intermediate molecules which are needed to be estimated in the evaluation and testing process before we expand the search route in the corresponding process. To overcome this, we provide a semi-dynamic way of graph construction for the task of retrosynthetic planning. Before the training process, only molecules in the training set are connected to construct the graph for training the GNN parameters. For every intermediate molecule in the evaluation and testing process, we only connect it with molecules in the training set. With this strategy, we could apply GNN for our task in the training, evaluation and testing process. Moreover, only connecting molecules with labeled samples could incur less noise.

To evaluate the estimation performance of our method, we

compare it with many state-of-the-art baselines. From the results on the USPTO dataset with same settings, we can observe that our method outperform others with a large margin, which could verify our assumption and the performance of our framework. Moreover, we give the ablation experiments and parameter analysis with different values, which could show the effect of different components in our method.

The contributions of our work could be summarized as follows:

- We propose a new method by GNN to estimate the cost of molecules in the task of retrosynthetic planning.

- We propose two ways to define the similarities between molecules based on their fingerprints and the predicted targets.

- We propose a semi-dynamic way for constructing graph, which makes GNN applicable in our task.

- Our framework outperforms all other state-of-the-art methods with a large margin on the benchmark USPTO.

Related Work

The problem of retrosynthesis could be classified into two categories. One type is that it only predicts the reactants of the target molecule within one step where some of the reactants may be unavailable molecules. The other will predict the reactants of the target molecule for multiple steps where all basic reactants are available molecules.

One-step Retrosynthesis

One-step retrosynthesis is aimed to predict the reactants of a target molecule within one step. In (Coley et al. 2017), they use a similarity model to predict the category of the reaction, where three types of similarities are applied in their method. Rule-based method is applied in (Segler and Waller 2017) and neural networks are used in their framework as a classifier to choose the reaction rule. Sequence-to-sequence model is exploited in (Liu et al. 2017; Karpov, Godin, and Tetko 2019), which could directly generate the reactants of the target molecule with this end-to-end model. In (Dai et al. 2019), they propose a maximum log-likelihood estimation and use GNN to generate the molecule embedding. The way they utilize GNN only considers the inner relationship between atoms in one molecule, which cannot reduce the sparsity problem like us.

However, for many target molecules, the reactants of one-step retrosynthesis may be unavailable, which should be synthesized by other available molecules.

Retrosynthetic Planning

Instead of just predicting one step reaction, retrosynthetic planning is proposed to predict the whole reaction route, where all molecules used to synthesize the target molecule are available. In DFPN-E(Kishimoto et al. 2019), authors solve this problem by deep first proof number search, and the heuristic edge initialization is used in their method. Monte Carlo Tree Search (MCTS) is exploited in (Segler, Preuss, and Waller 2018) to solve this task, in which they apply three

neural networks to tackle the expanding, selection and estimation tasks respectively. To make the success rate higher, Retro*(Chen et al. 2020) is proposed to tackle the task of retrosynthetic plan. Same as the A* search method, they use a neural network to estimate the cost from current state to the goal which is equal to the cost of synthesizing the corresponding intermediate molecules in the expanding route.

From the results of Retro*, we could see that the cost estimation of intermediate molecules influences the performance significantly. To make a more accurate estimation of molecule cost, we propose a new method by considering the relationships between molecules.

Preliminary

In this section, we will give the definition of basic elements and the aim of retrosynthetic planning.

Notification

We use **Tu** to denote the set of all unavailable molecules and **Ta** to denote the set of all available molecules. In the process of retrosynthetic planning, for an unavailable molecule $m \in \mathbf{Tu}$, we use $B(m) = \{R_i, \mathbf{S}_i, c(R_i)\}_{i=1}^k$ to denote the set of k possible one-step reactions which could synthesize m, where R_i is the denotation of i-th reaction, $\mathbf{S}_i$ is the set of all reactants in R_i and $c(R_i)$ is the cost of R_i. If R_i is in the route with minimal cost, for unavailable molecules in $\mathbf{S}_i$, retrosynthetic planning will continue to expand them with corresponding one-step reactions. Given a target molecule, the aim of retrosynthetic planning is to find a series of reactions, where all needed reactants are available and it has a minimal cost.

Motivation

The motivation of our work is to have a better cost estimation of the intermediate molecules. Existing works (Chen et al. 2020) have proved the effectiveness of the cost estimation, which could improve the success rate in the task of retrosynthetic planning. The exact answer of what and how influences the cost of synthesizing one molecule is uncertain. So using learning methods with labeled data to estimate the cost is a feasible way. However, the main challenge of applying learning method is that the data space of all molecules is extremely large, where labeled molecules only are a small part of it.

Graph for Retrosynthetic Planning

Retrosynthetic planning is usually modeled as a searching problem, where how to generate possible states and how to select the next action are the main challenges for it. How to generate possible states in retrosynthetic planning is similar as one-step retrosynthesis, and could be solved by the method of one-step retrosynthesis. How to select the next action is the specific problem for retrosynthetic planning, where the key point is to estimate the cost for the intermediate molecules in the searching process.

GNN for Sparsity

To conquer the data sparsity problem, we propose a new method for the cost estimation of molecules by utilizing graph neural networks. GNN method is firstly proposed to solve the semi-supervised problem, and now is popular in many types of applications (Han et al. 2019; Hu et al. 2020; Gidaris and Komodakis 2019; Zhou et al. 2019; Xu et al. 2019; Lamb et al. 2020; Wang et al. 2019). The main difference between the computation of GNN and the computation of traditional neural networks is that, GNN will directly and explicitly incorporate the information of neighbours in the training and testing process. The computation of GNN could reduce the data sparsity problem when the relationship between samples is relevant to their labels. Given the representation X^n of all samples in n-th layer, the $(n + 1)$-th representation X^{n+1} for all samples are computed as follows:

$$X^{n+1} = Relu(AX^nW^n + B^n) \qquad (1)$$

where A is the Laplacian matrix of the graph, W^n and B^n are the learnable weights for the n-th layer in GNN.

To make GNN applicable for our task to reduce the sparsity problem of training dataset, we need to solve the most important problem firstly, which is how to construct the graph of the GNN for our task. Because the graph in GNN, which determines the relationship between samples, will influence the performance directly. In this section, we propose two ways to define the similarities between molecules.

Graph with Threshold

The construction of graph for GNN should be highly related to the aim of the application. The goal of the estimation is to predict the synthetic cost of the given molecules. So it's better to connect molecules with similar synthetic cost to constructed graph. The synthetic cost of one molecule is mainly determined by its composed atoms and structure. To describe the atoms and structures of given molecule, Morgan fingerprint (Rogers and Hahn 2010) is proposed as the representation of the molecule. So, we assume that molecules with similar fingerprints have similar synthetic cost. Under our assumption, cosine similarity of the fingerprints is used as the metric to describe the relationship between molecules. Given the FingerPrints f_i and f_j of molecules m_i and m_j, we define the similarity $S_{i,j}$ between them as

$$S_{i,j} = \cos(f_i, f_j) = \frac{f_i f_j}{\|f_i\| \|f_j\|}. \qquad (2)$$

However, these similarities cannot be directly used as the weights of the graph. Because this will make the graph too dense, which will incur noise rather than meaningful information. To fix this, we construct the graph with a handcrafted threshold, which can filter most of noise. Then, only the molecules with similarities larger than the threshold could be connected with similarities as weights. For molecules m_i and m_j, we define the connection weight $\mathbb{G}(m_i, m_j)$ between them as:

$$\mathbb{G}(m_i, m_j) = \begin{cases} S_{i,j}, & \text{if } S_{i,j} > \tau \text{ and } i \neq j \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

where τ is the threshold to filter the connections with small similarities.

Learned-based Graph

However, the graph in Eq. (3) only considers the information of fingerprint, that cannot directly reflect the relationship of synthetic cost between molecules. To make the graph able to reflect the relationship of synthetic cost and contain the information of fingerprint, we propose an alternative way to construct it, where embedding representation is learned to exploit the mentioned information. Given the fingerprint f_i of molecule m_i, we could generate the corresponding embedding e_i with an embedding network as follows:

$$e_i = f_i W_e + B_e. \tag{4}$$

Under this way, the embeddings of molecules could contain the information of fingerprint. Before we show how to exploit the information of synthetic cost, we firstly construct the similarities between molecules with their corresponding embeddings. Cosine distance is utilized in our method to generate the similarity, which will make the molecules with similar embeddings have high similarity. Given the embeddings e_i and e_j of molecules m_i and m_j, we generate the similarity $S_{i,j}$ between them as:

$$S_{i,j} = \cos(e_i, e_j) = \frac{e_i e_j}{\|e_i\|\|e_j\|}. \tag{5}$$

How to design the loss function to train the embeddings network will directly influence the performance of retrosynthetic planning. Because the graph constructed by the similarity with the embedding will be exploited in the GNN to estimate the synthetic cost. The operation of GNN is to incur the representations of neighbors for all samples. Then, we try to make the molecules with high similarities will have similar synthetic cost. Given the training set $\mathbf{Tr}$ and a molecule m_i, we use $\mathbf{S}_{m_i}^+ \subseteq \mathbf{Tr}$ to denote the positive set that contains molecules with similar synthetic cost as molecule m_i. And we use $\mathbf{S}_{m_i}^- = \mathbf{Tr} \setminus \mathbf{S}_{m_i}^+$ to denote the negative set that contains the molecules with different synthetic cost from molecule m_i. To make the molecules with similar synthetic cost have similar embeddings, we use the pair-wise loss function to define our loss function L as follows:

$$L = \sum_{m_i \in \mathbf{Tr}, m_j \in \mathbf{S}_{m_i}^+, m_k \in \mathbf{S}_{m_i}^-} \frac{\mathbb{I}(S_{i,j} > S_{i,k})}{\|\mathbf{Tr}\|\|\mathbf{S}_{m_i}^+\|\|\mathbf{S}_{m_i}^-\|} \tag{6}$$

where $\mathbb{I}$ is the indicator function that equals one if the condition satisfies, otherwise it will be zero. However, the size of training set $\mathbf{Tr}$ is large, which makes the training process time-consuming. To overcome this problem, we use a sampling method to select negative molecules. So, in the training process, for every pair (m_i, m_j) with $m_j \in \mathbf{S}_{m_i}^+$, we randomly select one $m_k \in \mathbf{S}_{m_i}^-$ instead of using all negative molecules for this pair.

Once we have finished the training process of embedding network, we could get embeddings for all molecules and construct the corresponding similarities. However, the similarity matrix will be dense, as every entry is larger than 0.

This will increase the calculation amount of GNN. To solve this, we construct the K-NN graph $\mathbb{G}$ as follows:

$$\mathbb{G}(m_i, m_j) = \begin{cases} S_{i,j}, & \text{if } m_j \in N(m_i) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $N(m_i)$ is the K-NN similarity set of molecule m_i.

Our Framework

In this section, we will introduce the main framework of our work. To apply GNN on our problem, we propose a semi-dynamic strategy to utilize the graph which could solve the problem of graph construction in testing process. And the estimated cost is predicted by the GNN embedding with regression method. The whole process of our framework could be found in Fig. 2.

Semi-Dynamic Graph

Although we have defined the way to connect molecules, we cannot build the graph for all molecules. The main reason is that the data space of all molecules is extremely large, so we cannot build a global graph before the training process. Moreover, even we only want to construct a graph for all molecules used in the training, evaluation and testing process, we cannot achieve it. Because the search process is dynamic and is related to the estimation function itself. So we cannot get all molecules which are needed to be estimated in the evaluation and testing process before we expand the search route in the corresponding process. To overcome this, we provide a semi-dynamic way of graph construction for the task of retrosynthetic planning. With our strategy, only molecules in the training set are connected to construct the graph before the training process. The intermediate molecules in the validation and testing process will be connected to the molecules in the training set dynamically.

Given the training set $\mathbf{Tr}$ with cardinality N_{tr}, we could construct its graph $G^T \in \mathcal{R}^{N_{tr} \times N_{tr}}$ with element $G_{i,j}^T = \mathbb{G}(m_i, m_j)$. As the weight in G^T is constructed with assumption, it can not perfectly reflect the similarity of synthetic cost between corresponding molecule pairs. To make the influence of the neighbours adjustable, we define the weighted Laplacian matrix A^T of graph G^T with a weight α as

$$A^T = I + \alpha Norm(G^T), \tag{8}$$

where $Norm(G) = D^{-\frac{1}{2}} G D^{-\frac{1}{2}}$ is the normalization operation for the matrix G in which D is a dialogue matrix with element $D_{i,i} = \sum_j G_{i,j}$. Then, given fingerprints F of all molecules in training set, we could get the representation X^{Tr} of them under the computation of GNN as

$$X^{Tr} = A^T F W_g + B_g, \tag{9}$$

where W_g and B_g are learnable parameters for GNN. Once we get the representation of all molecules in training dataset, we use a multiple layer perception (MLP) and log function to generate the cost estimation V of them as

$$V = \log(1 + e^{X^{Tr} W_m + B_m}), \tag{10}$$
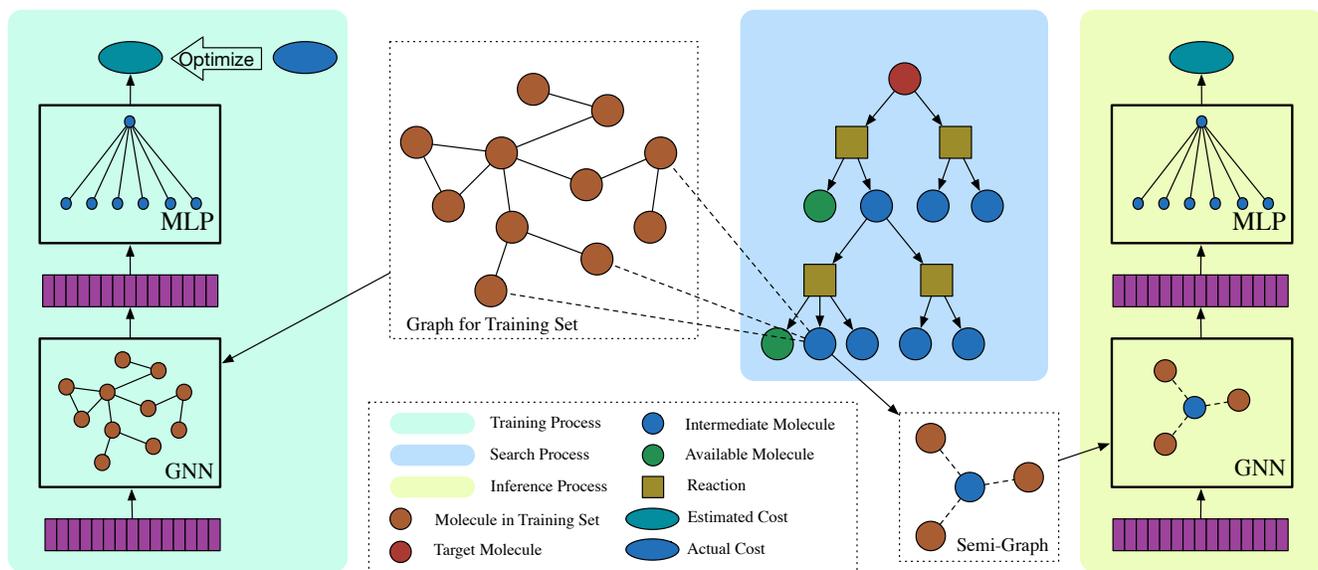
Figure 2: Framework of our method

where W_m and B_m are learnable parameters for MLP. And we use V_{m_i} to denote the cost estimation of molecule m_i.

For molecules in the evaluation and testing process, we only connect them with molecules in the training set **Tr**. With this strategy, we could apply GNN for our task in the evaluation and testing process. Moreover, only connecting molecules with labeled samples could incur less noise under our assumption. Then, once we have an intermediate molecule m_i which needs the cost estimation, we firstly get the set of its connected neighbors $N^t(m_i)$ as follows

$$N^t(m_i) = \{m_j | m_j \in \textbf{Tr} \text{ and } \mathbb{G}(m_i, m_j) > 0\}. \quad (11)$$

Cost Estimation

Once we have the set of connected molecules of molecule m_i, we could get its GNN embedding x_i as

$$x_i = (f_i + \alpha \frac{\sum_{m_j \in N^t(m_i)} (\mathbb{G}(m_i, m_j) f_j)}{Z}) W_g + B_g, \quad (12)$$

where $Z = \sum_{m_j \in N^t(m_i)} \mathbb{G}(m_i, m_j)^2$ and α is the weight to control the influence of the neighbors with the same value as Equation 8. Under this way, the information of molecules in the training set could be imposed to refine the process of estimation by reducing the underfitting problem.

Regression is used to construct the relationship between the GNN embedding and estimated cost for every molecule. Then we could get the cost estimation V_{m_i} for the intermediate molecule m_i as:

$$V_{m_i} = log(1 + e^{x_i W_m + B_m}), \quad (13)$$

where W_m and B_m are learnable parameters for the cost estimation.

Optimization and Search

In this section, we will introduce the optimization of our model and search process for retrosynthetic planning.

Loss Function

The loss function we use to optimize our model is composed of two main components. The first part is the regression loss L^R which is designed as

$$L^R = \frac{1}{N_{Tr}} \sum_{m_i \in \textbf{Tr}} \|V_{m_i} - V'_{m_i}\|^2, \quad (14)$$

where V'_{m_i} is the actual synthetic cost for molecule m_i. By minimizing this loss function, we could make sure that the estimated synthetic cost will be approximately equal to the actual one.

The other component is the partial ordering loss which is proposed to keep the partial ordering relation between the optimal one-step reaction and the others. Given the molecule m_i and its possible reactions $B(m_i) = \{R_j, \textbf{S}_j, c(R_j)\}_{j=1}^k$, we assume that the optimal reaction is R_i^*. For any $R_j \neq R_i^*$, we define its partial loss as:

$$L(m_i, R_j) = \max\{0, V_{m_i} + \epsilon - c(R_j) - \sum_{m' \in S_j} V_{m'}\}, \quad (15)$$

where ϵ is the slack variable to make estimated cost of the optimal reaction smaller than the estimated cost of others with a large margin. After this, we could get the partial ordering loss L^P of all molecules by averaging as follows:

$$L^P = \frac{1}{N_{Tr}} \sum_{m_i \in \textbf{Tr}} \frac{1}{\|B(m_i)\| - 1} \sum_{R_j \in B(m_i) \setminus \{R_i^*\}} L(m_i, R_j). \quad (16)$$

By minimizing this loss function, we could keep the partial order between the optimal reactions and others, which will help us find the optimal route within less steps in the process of retrosynthetic planning.

By combining these two losses, we could get the final loss function L as:

$$L = L^R + \lambda L^P, \quad (17)$$

where λ is the weight to control the influence of partial ordering loss. Adam (Kingma and Ba 2015) is utilized as the optimizer to minimize the loss L with learning rate 0.001.

Route Search

To make a fair comparison, we apply our estimation method with the A* search method to generate the final result, which is the same as Retro* (Chen et al. 2020) method. For the target molecule and every intermediate molecule, a MLP template (Segler and Waller 2017) is applied to generate k possible one-step reactions, where $k = 50$ in our framework. The search process is initialized from the target molecule. For every step, we will expand one intermediate molecule which has the minimal route cost. The route cost is the sum of current cost and estimated cost. After every expansion, we will update the current cost of all relevant molecules which are in the same route as the expanded molecule. These processes will run iteratively until there is one route for which all basic molecules are available or that reach the maximal iteration.

Experiment

In this section, we will give the experiment settings and results with the comparison of state-of-the-art methods. Moreover, we will give the parameter analysis and ablation experiments which could show the effectiveness of different components in our framework.

Dataset

The public reaction dataset United States Patent Office (USPTO) is used in our method with the same preprocessing as (Chen et al. 2020). There are about 1.3 million reactions after the deduplication and filtration, which are randomly separated into training/validation/testing sets with proportion 80%/10%/10% respectively. Synthesis routes are generated from these reactions for test set. After all process, there are 299,202 and 65,274 one-step reactions for training and validating the estimation method, and 190 retrosynthetic routes in the testing set for the evaluation of whole framework.

Experiment Setting

For every target molecule, we at most run the one-step reactions 500 times, which is the same as (Chen et al. 2020). The embedding of the molecule is fixed as 128. We set the weight λ of partial ordering loss as 1. The slack variable ϵ is set as 7. For the threshold τ, we select it from the range $[0 : 0.1 : 1.0]$. The weight α is also selected from the range $[0 : 0.1 : 1.0]$.

Baselines

We use **GNN-Retro (Threshold)** to denote our method with graph in Eq. (3) and **GNN-Retro (Embedding)** to denote our method with graph in Eq. (7). And We compare our method with 5 baselines, and the details of these methods are summarized as follows:

Method	Success Rate	#Route
Greedy DFS	22.64%	43
MCTS	33.68%	64
DFPN-E	55.26%	105
Retro*-0	79.47%	151
Retro*	86.84%	165
GNN-Retro (Threshold)	91.05%	173
GNN-Retro (Embedding)	87.37%	166

Table 1: Results

- Retro* (Chen et al. 2020): Based on the A* search method, they use a neural network to estimate the cost from current state to the goal.

- Retro*-0: This is a simplified version of Retro*, which only use the current cost to select the next molecule.

- DPFN-E (Kishimoto et al. 2019): Deep first proof number search is applied in this method with the heuristic edge initialization.

- MCTS (Segler, Preuss, and Waller 2018): Monte Carlo Tree Search (MCTS) is exploited here to solve this task.

- Greedy DFS: A naive baseline which is implemented by greedy depth first search algorithm.

Results

The results of all methods could be found in the Table 1. As all experiment settings are same as (Chen et al. 2020), we use their reported results. From our experiment results, we could make some observations as follows:

- Our method outperforms Greedy DFS, MCTS, DFPN-E, Retro*-0 and Retro* with a ratio 302.2%, 170.3%, 64.8%, 14.6% and 4.8% respectively, which could verify the effectiveness of our framework in improving performance of retrosynthetic planning.

- From the comparison between GNN-Retro and Retro*-0, we could see that using the cost estimation of molecules will improve the success rate significantly. This conclusion could also be obtained by the comparison between Retro* and Retro*-0.

- From the comparison between GNN-Retro and Retro*, we could verify that the way we estimate the synthetic cost of molecules is better than that in Retro*. Because we utilize GNN to conquer the problem of data sparsity by incurring the information of labeled neighbors.

- From the comparison between GNN-Retro (Threshold) and GNN-Retro (Embedding), we could find that the way we construct graph will influence the final performance. However, both of them outperform all other state-of-the-art methods, which could verify the superiority of our method.

Ablation Experiments

In this section, we will give the ablation experiments for GNN-Retro by deleting some components in GNN-Retro. The way we construct the ablation experiments are summarized as follows:

Method	Success Rate	#Route
GNN-Retro/GNN	86.84%	165
GNN-Retro/Partial	80.53%	153
GNN-Retro	91.05%	173

Table 2: Ablation Experiments

- GNN-Retro/GNN: In this method, we delete the component of GNN, in which we only use the original embedding of molecule as the input for the regression component.

- GNN-Retro/Partial: To show the effectiveness of the partial ordering loss (Equ. 16), we only keep the regression loss (Equ. 14) as the objective function to train our model.

The results of the ablation experiments could be found in Table. 2. And we give the analysis of these results as:

- The result of GNN-Retro/GNN gives the clear image that applying GNN could improve the success rate significantly. The structure of GNN could incur the information of neighbors in the training and inference process. And the construction of graph for GNN is the key point to make the incurred neighbors contain the similar synthetic cost as the target one. Our strategy of graph construction for GNN could reduce the data sparsity problem and achieve a higher success rate.

- The comparison between GNN-Retro/Partial and GNN-Retro shows that the partial order plays an important role in the task of retrosynthetic planning. Because in the search process, how to choose the next expanded molecule depends on the partial order instead of the absolute synthetic cost. So, making the synthetic cost of optimal reaction smaller than the synthetic costs of others will influence the success rate of retrosynthetic plaining explicitly and directly.

Parameter Analysis

To evaluate our strategy of constructing graph for GNN, we give the experiment results for the threshold τ and weight α in Fig. 3(a) and Fig. 3(b).

From Fig. 3(a), we could find that we could achieve the best performance when $\tau = 0.9$. When $\tau = 1.0$, the molecule will only be connected to itself, which is the same as MLP. Comparing the results between $\tau = 0.9$ and $\tau = 1.0$, we could verify that applying GNN by incurring information of labeled neighbors will increase the success rate of retrosynthetic planning. It also prove the correctness of our assumption that molecules with similar fingerprint will have similar synthetic cost. Comparing the results between $\tau = 0.9$ and other τs with value less than 0.9, it could prove that our strategy of filtering molecules for the graph will improve the performance. Because the similarity is hand-crafted, and it may incur many noises where the relationship between molecule with low similarities is uncertain. Filtering molecules with low similarities will reduce the percentages of irrelevant connected molecules.
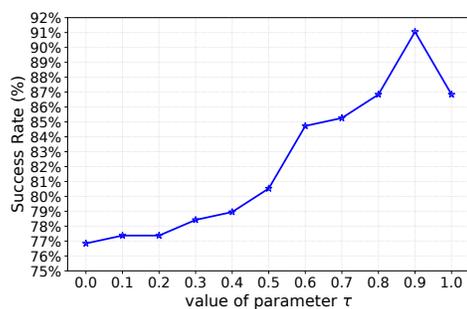
(a) Results of parameter τ
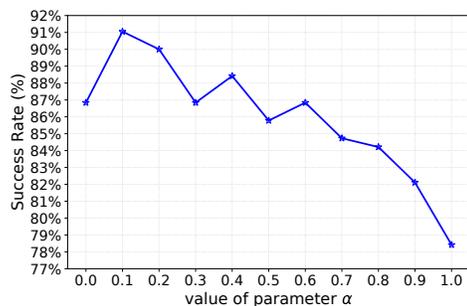
(b) Results of parameter α

Figure 3: Results of different parameters

The results in Fig. 3(b) gives a clear image that different α will lead to various performances. This phenomenon verifies that the construction of graph influences the success rate of retrosynthetic planning significantly. When $\alpha = 1.0$, we could see that the performance of GNN-Retro is even lower than Retro*-0, which means applying cost estimation is worse than the version without cost estimation. The reason why this scenario happens is that the similarity cannot reflect the relationship of synthetic cost perfectly. So using a tunable parameter α to control the influence of neighbors is indispensable for GNN with hand-crafted graph. Moreover, comparing the results between $\alpha = 0.1$ and $\alpha = 0.0$, we could once again verify that applying GNN by incurring information of labeled neighbors will improve the success rate of retrosynthetic planning.

Conclusion

In this work, we propose a new method for the synthetic cost estimation of molecules in the task of retrosynthetic planning. To overcome the data sparsity problem, we apply GNN to incur the information of neighbors. To construct the graph of GNN, we assume that molecules with similar fingerprint will have similar synthetic cost. A graph with threshold is constructed to filter the incurred noise. Moreover, a semi-dynamic graph is proposed to apply the GNN on molecules in the test process. A* search method is applied as the search algorithm for the route generation. The results on the benchmark USPTO show that our algorithm outperforms all state-of-the-art methods with a large margin.

Acknowledgments

References

Chen, B.; Li, C.; Dai, H.; and Song, L. 2020. Retro*: Learning Retrosynthetic Planning with Neural Guided A* Search. In *International Conference on Machine Learning*.

Coley, C. W.; Rogers, L.; Green, W.; and Jensen, K. 2017. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Central Science*, 3: 1237 – 1245.

Dai, H.; Li, C.; Coley, C. W.; Dai, B.; and Song, L. 2019. Retrosynthesis Prediction with Conditional Graph Logic Network. In *Annual Conference on Neural Information Processing Systems*.

Gidaris, S.; and Komodakis, N. 2019. Generating Classification Weights With GNN Denoising Autoencoders for Few-Shot Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 21–30.

Han, P.; Yang, P.; Zhao, P.; Shang, S.; Liu, Y.; Zhou, J.; Gao, X.; and Kalnis, P. 2019. GCN-MF: Disease-Gene Association Identification By Graph Convolutional Networks and Matrix Factorization. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; and Sun, Y. 2020. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Karpov, P.; Godin, G.; and Tetko, I. 2019. A Transformer Model for Retrosynthesis. In *ICANN*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kipf, T.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv*, abs/1609.02907.

Kishimoto, A.; Buesser, B.; Chen, B.; and Botea, A. 2019. Depth-First Proof-Number Search with Heuristic Edge Cost and Application to Chemical Synthesis Planning. In *Annual Conference on Neural Information Processing Systems*.

Lamb, L.; Garcez, A.; Gori, M.; Prates, M. O. R.; Avelar, P. H. C.; and Vardi, M. Y. 2020. Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective. In *International Joint Conference on Artificial Intelligence*.

Li, J.; Li, F.; King-Smith, E.; and Renata, H. 2020. Merging Chemoenzymatic and Radical-Based Retrosynthetic Logic For Rapid and Modular Synthesis of Oxidised Meroterpenoids. *Nature chemistry*, 12: 173 – 179.

Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Nguyen, Q. L.; Ho, S.; Sloane, J. L.; Wender, P.; and Pande, V. 2017. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Central Science*, 3: 1103 – 1113.

Mao, K.; Xiao, X.; Xu, T.; Rong, Y.; Huang, J.; and Zhao, P. 2021. Molecular graph enhanced transformer for retrosynthesis prediction. *Neurocomputing*, 457: 193–202.

Rogers, D.; and Hahn, M. 2010. Extended-Connectivity Fingerprints. *Journal of chemical information and modeling*, 50 5: 742–54.

Schreck, J. S.; Coley, C. W.; and Bishop, K. 2019. Learning Retrosynthetic Planning through Simulated Experience. *ACS Central Science*, 5: 970 – 981.

Schwaller, P.; Petraglia, R.; Zullo, V.; Nair, V. H.; Häuselmann, R.; Pisoni, R.; Bekas, C.; Iuliano, A.; and Laino, T. 2020. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical Science*, 11: 3316–3325.

Segler, M. H. S.; Preuss, M.; and Waller, M. P. 2018. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555: 604–610.

Segler, M. H. S.; and Waller, M. 2017. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chemistry*, 23 25: 5966–5971.

Wang, Q.; Wang, B.; Xu, Z.; Wu, J.; Zhao, P.; Li, Z.; Wang, S.; Huang, J.; and Cui, S. 2021. PSSM-Distil: Protein Secondary Structure Prediction (PSSP) on Low-Quality PSSM by Knowledge Distillation with Contrastive Learning. In *AAAI*, volume 35, 617–625.

Wang, Z.; Ren, Z.; He, C.; Zhang, P.; and Hu, Y. 2019. Robust Embedding with Multi-Level Structures for Link Prediction. In *International Joint Conference on Artificial Intelligence*.

Xu, C.; Zhao, P.; Liu, Y.; Sheng, V.; Xu, J.; Zhuang, F.; Fang, J.; and Zhou, X. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *International Joint Conference on Artificial Intelligence*.

Yan, C.; Ding, Q.; Zhao, P.; Zheng, S.; Yang, J.; Yu, Y.; and Huang, J. 2020. RetroXpert: Decompose Retrosynthesis Prediction Like A Chemist. In *NeurIPS*.

Yang, J.; Zhao, P.; Rong, Y.; Yan, C.; Li, C.; Ma, H.; and Huang, J. 2021. Hierarchical Graph Capsule Network. In *AAAI*, 10603–10611.

Zheng, S.; Rao, J.; Zhang, Z.; Xu, J.; and Yang, Y. 2020. Predicting Retrosynthetic Reactions Using Self-Corrected Transformer Neural Networks. *Journal of chemical information and modeling*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *CIKM*, 2357–2360.