

# Retrosynthesis Prediction with Local Template Retrieval

Shufang Xie<sup>1</sup>, Rui Yan<sup>1\*</sup>, Junliang Guo<sup>2</sup>, Yingce Xia<sup>3</sup>, Lijun Wu<sup>3</sup>, Tao Qin<sup>3</sup>

<sup>1</sup>Beijing Key Laboratory of Big Data Management and Analysis Methods,  
Gaoling School of Artificial Intelligence (GSAI), Renmin University of China

<sup>2</sup>Microsoft Research Aisa

<sup>3</sup>Microsoft Research AI4Science

{shufangxie,ruiyan}@ruc.edu.cn, {junliangguo,yingce.xia,lijunwu,taoqin}@microsoft.com

## Abstract

Retrosynthesis, which predicts the reactants of a given target molecule, is an essential task for drug discovery. In recent years, the machine learning based retrosynthesis methods have achieved promising results. In this work, we introduce RetroKNN, a local reaction template retrieval method to further boost the performance of template-based systems with non-parametric retrieval. We first build an atom-template store and a bond-template store that contain the local templates in the training data, then retrieve from these templates with a k-nearest-neighbor (KNN) search during inference. The retrieved templates are combined with neural network predictions as the final output. Furthermore, we propose a lightweight adapter to adjust the weights when combining neural network and KNN predictions conditioned on the hidden representation and the retrieved templates. We conduct comprehensive experiments on two widely used benchmarks, the USPTO-50K and USPTO-MIT. Especially for the top-1 accuracy, we improved 7.1% on the USPTO-50K dataset and 12.0% on the USPTO-MIT dataset. These results demonstrate the effectiveness of our method.

## 1 Introduction

Retrosynthesis, which predicts the reactants for a given product molecule, is a fundamental task for drug discovery. The conventional methods heavily rely on the expertise and heuristics of chemists (Corey 1991). Recently, machine learning based approaches have been proposed to assist chemists and have shown promising results (Dong et al. 2021). The typical approaches includes the template-free methods that predict the reactants directly and the template-based methods that first predict reaction templates and then obtain reactants based on templates. For these different approaches, a shared research challenge is effectively modeling this task’s particular property.

As shown in Figure 1, a key property of a chemical reaction is that it is strongly related to modifying the local structure of the target molecule, such as replacing a functional group or breaking a bond. Therefore, much recent research focuses on better modeling the local structure of molecules (Chen and Jung 2021; Somnath et al. 2021). De-

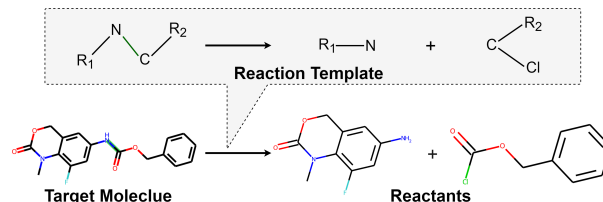


Figure 1: Illustration of retrosynthesis that takes the target molecule on the left side and predicts two reactants on the right side. Inside the callout is its reaction template that breaks the carbon-nitrogen bond into two parts.

spite their promising results, we notice that it is still challenging to learn all reaction patterns only with neural networks, especially for the rare templates.

Therefore, we introduce a non-parametric retrieval-based method to provide concrete guidance in prediction. Specifically, we use a local template retrieval method, the k-nearest-neighbor (KNN) method, to provide additional predictions to improve the prediction accuracy. Following LocalRetro (Chen and Jung 2021), We first take a trained graph-neural network (GNN) for the retrosynthesis task and offline build an atom-template and a bond-template store that contain reaction templates (Section 2.1). During this store construction phase, we iterate all target molecules in the training data and add the templates of each atom and each bond to the corresponding store. The templates are indexed by the hidden representations extracted by the GNN. During inference, for a given new target molecule, we first use the original GNN to extract the hidden representations as well as the original GNN predicted templates. Then, we use the hidden representations to search the two stores to retrieve local templates similar to the query. The GNN predicted templates and the KNN retrieved templates are merged with different weights to build the final output.

Combining the GNN and KNN predictions is one key design factor in the above processes. The conventional way is to use fixed parameters to aggregate these predictions for all reactions, which may be sub-optimal and hurt the model’s generalization (Zheng et al. 2021). Because each prediction may have a different confidence level, it would be beneficial to assign the weights adaptively for each reac-

\*Corresponding author: Rui Yan (ruiyan@ruc.edu.cn).

tion across different instances (Section 4.1). Therefore, we employ a lightweight adapter to predict these values conditioned on the GNN representations and the retrieved results. The adapter network has a simple structure and is trained with a few samples. Although the adapter has a little extra cost, it can help improve the model performance effectively.

To sum up, our contribution is two fold:

- We propose RetroKNN, a novel method to improve the retrosynthesis prediction performance with local template retrieval by the non-parametric KNN method.
- We propose a lightweight meta-network to adaptively control the weights when combining the GNN and KNN predictions.

We conduct experiments on two widely used benchmarks: the USPTO-50K and USPTO-MIT. These datasets contain organic reactions extracted from the United States Patent and Trademark Office (USPTO) literature. On the USPTO-50K dataset, we improve the top-1 accuracy from 53.4 points to 57.2 points (7.1% relative gain) and achieved new state-of-the-art. Meanwhile, on USPTO-MIT, we improve the top-1 accuracy from 54.1 points to 60.6 points (12.0% relative gain). Moreover, our method shows promising results on the zero-shot and few-shot datasets, which are challenging settings for conventional template-based methods yet essential for this research field. These results demonstrate the effectiveness of our method.

## 2 Method

### 2.1 Preliminaries

We denote a molecule as a graph  $G(\mathcal{V}, \mathcal{E})$  where the  $\mathcal{V}$  is the node set and the  $\mathcal{E}$  is the bond set. Given a target molecule  $M$  as input, the retrosynthesis prediction task is to generate molecules set  $\mathcal{R}$  that are reactants of  $M$ . Instead of directly predicting  $\mathcal{R}$ , we follow LocalRetro (Chen and Jung 2021) that predict a local reaction template  $t$  at reaction center  $c$  and apply  $(t, c)$  to molecule  $M$ . More specifically, the  $t$  is classified into two types: **atom-template**  $t \in \mathcal{T}_a$  and **bond-template**  $t \in \mathcal{T}_b$ , depending whether  $c$  is an atom or a bond.

We also assume that there are a training set  $\mathcal{D}_{\text{train}}$ , a validation set  $\mathcal{D}_{\text{val}}$ , and a test set  $\mathcal{D}_{\text{test}}$  available. Each data split contains the target and corresponding reactants, which is formulated as  $\mathcal{D} = \{(M_i, t_i, c_i, \mathcal{R}_i)\}_{i=1}^{|\mathcal{D}|}$  where  $c_i$  is the reaction center of  $M_i$  to apply the template  $t_i$  and  $|\mathcal{D}|$  is the data size of  $\mathcal{D}$ .

Meanwhile, we assume a GNN model trained on  $\mathcal{D}_{\text{train}}$  exist. Without loss of generality, we split the GNN into two parts: a feature extractor  $\mathbf{f}$  and a prediction head  $\mathbf{h}$ . The feature extractor  $\mathbf{f}$  takes a molecule graph  $G(\mathcal{V}, \mathcal{E})$  as input and output hidden representations  $h_v$  for each node  $v \in \mathcal{V}$  and  $h_e$  for each edge  $e \in \mathcal{E}$ . The  $h_v$  and  $h_e$  are processed by prediction head  $\mathbf{h}$  to predict the probability distribution over the template set  $\mathcal{T}_a$  and  $\mathcal{T}_b$ , respectively.

### 2.2 Store Construction

Our method uses two data store  $\mathcal{S}_A$  and  $\mathcal{S}_B$  that contain the information of atoms and bonds. Both of the store are constructed offline before inference. Inside the store are key-

---

#### Algorithm 1: store construction algorithm

---

**Input:** Training data  $\mathcal{D}_{\text{train}}$ .  
**Input:** Feature extractor  $\mathbf{f}$ .  
**Output:** Atom store  $\mathcal{S}_A$  and bond store  $\mathcal{S}_B$ .

```

1 Let  $\mathcal{S}_A := \emptyset, \mathcal{S}_B := \emptyset$ ; // Initialize.
2 for  $(M, t, c, \mathcal{R}) \in \mathcal{D}_{\text{train}}$  do
3   Let  $\mathcal{V}$  denotes the node set of  $M$ ;
4   Let  $\mathcal{E}$  denotes the edge set of  $M$ ;
5   for  $v \in \mathcal{V}$ ; // Loop each node.
6   do
7     Let  $h_v := \mathbf{f}(v|M)$ ;
8     if  $v == c$  then
9       Let  $\mathcal{S}_A := \mathcal{S}_A \cup \{(h_v, t)\}$ ;
10    else
11      Let  $\mathcal{S}_A := \mathcal{S}_A \cup \{(h_v, \mathbf{0})\}$ ;
12    end
13  end
14  for  $e \in \mathcal{E}$ ; // Loop each edge.
15  do
16    Let  $h_e := \mathbf{f}(e|M)$ ;
17    if  $e == c$  then
18      Let  $\mathcal{S}_B := \mathcal{S}_B \cup \{(h_e, t)\}$ ;
19    else
20      Let  $\mathcal{S}_B := \mathcal{S}_B \cup \{(h_e, \mathbf{0})\}$ ;
21    end
22  end
23 end
24 return  $\mathcal{S}_A, \mathcal{S}_B$ 

```

---

value pairs that are computed from  $\mathcal{D}_{\text{train}}$  and the construction procedure details are in Algorithm 1.

In this algorithm, the first step is to initialize the atom store  $\mathcal{S}_A$  and bond store  $\mathcal{S}_B$  as an empty set. Next, for each reaction in the training data  $\mathcal{D}_{\text{train}}$ , we iterate all nodes  $v \in \mathcal{V}$  and all edges  $e \in \mathcal{E}$  of the target molecule  $M$  in line 5 to 13 and line 14 to 22, respectively. For each node  $v$ , if it is the reaction center, we add template  $t$  that indexed by the hidden representation  $h_v$  to the  $\mathcal{S}_A$ . Otherwise, we add a special token  $\mathbf{0}$  to indicate that template is not applied here. Similarly, for each edge  $e$ , we add either  $(h_e, t)$  or  $(h_e, \mathbf{0})$  to the bond store  $\mathcal{S}_B$ . Finally, we get the atom store  $\mathcal{S}_A$  and the bond store  $\mathcal{S}_B$  used during inference.

### 2.3 Inference Method

The overview of inference procedure is available in Figure 2. At inference time, given a new target molecule  $M$ , we first compute the hidden representation  $h_v, h_e$  and template probability  $P_{\text{GNN}}(t_a|M, a), P_{\text{GNN}}(t_b|M, b)$  for each atom  $a$  and bond  $b$ , respectively<sup>1</sup>. Next, we retrieve the templates for each node and edge, which can be written as

---

<sup>1</sup>Whenever possible, we omit the subscript of node and edge id to simplify the notations.

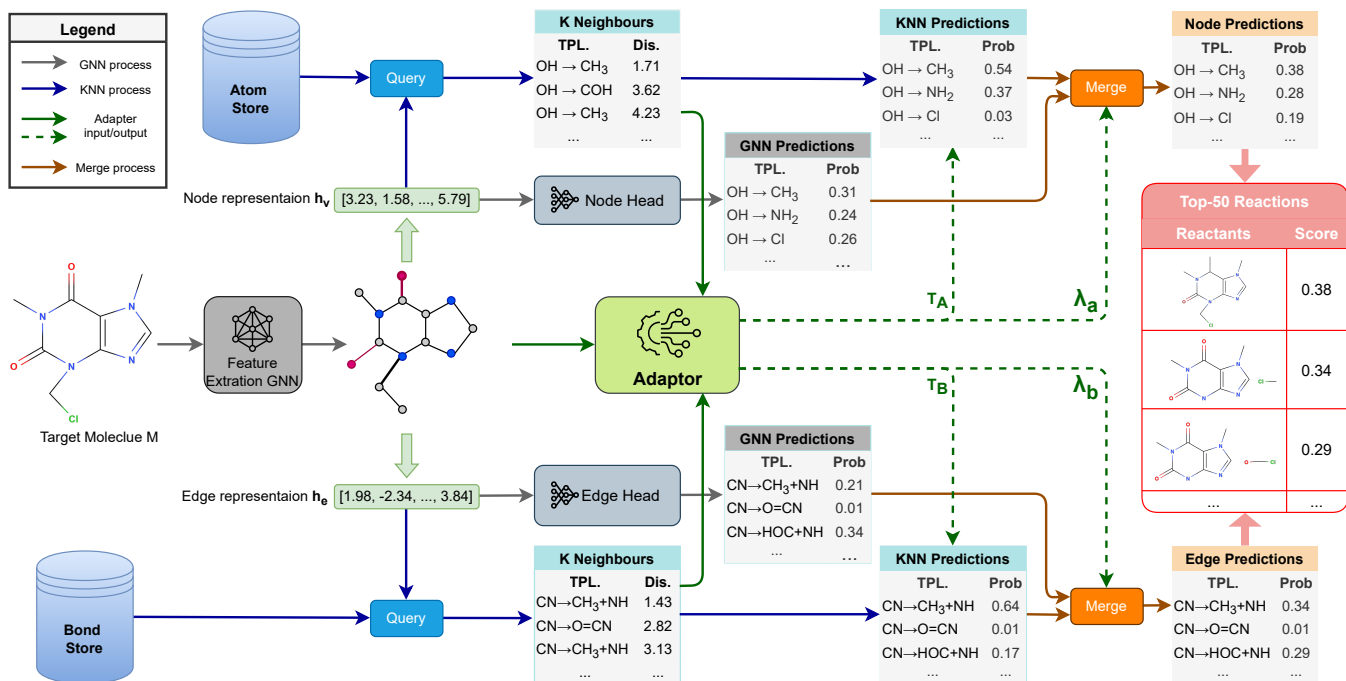


Figure 2: The illustration of RetroKNN for a target molecule in the middle left. The top and bottom half show the examples of one atom and bond retrieval. The gray, blue, green, and brown lines denote the GNN prediction, KNN prediction, adaptor input/output, and merge process, respectively. The pink table denotes the final output from all predictions.

$$P_{\text{KNN}}(t_a|M, a) \propto \sum_{(h_i, t_i) \in \mathcal{N}_a} \mathbb{I}_{t_a=t_i} \exp\left(\frac{-d(h_a, h_i)}{T_A}\right), \quad (1)$$

$$P_{\text{KNN}}(t_b|M, b) \propto \sum_{(h_i, t_i) \in \mathcal{N}_b} \mathbb{I}_{t_b=t_i} \exp\left(\frac{-d(h_b, h_i)}{T_B}\right). \quad (2)$$

In Equations (1, 2), the  $\mathcal{N}_a, \mathcal{N}_b$  are candidates sets that retrieved from  $\mathcal{S}_A, \mathcal{S}_B$ , the  $\mathbb{I}$  is the indicator function that only outputs 1 when the condition (i.e.,  $t_a = t_i$  or  $t_b = t_i$ ) is satisfied, and the  $T_A, T_B$  are the softmax temperate. Meanwhile, the  $d(\cdot, \cdot)$  is the distance function to measure the similarity between  $h_i$  with  $h_v$  or  $h_e$ . In another words, the  $P_{\text{KNN}}(t_a|M, a)$  is proportional to the sum of the weights of the neighbours whose template is  $t_a$ .

Finally, we combine the GNN output and KNN output with interpolation factors  $\lambda$ , which is

$$P(t_a|M, a) = \lambda_a P_{\text{GNN}}(t_a|M, a) + (1 - \lambda_a) P_{\text{KNN}}(t_a|M, a), \quad (3)$$

$$P(t_b|M, b) = \lambda_b P_{\text{GNN}}(t_b|M, b) + (1 - \lambda_b) P_{\text{KNN}}(t_b|M, b). \quad (4)$$

In the Equation (1)-(4), the temperature  $T_A, T_B \in \mathbb{R}^+$  and the interpolation factors  $\lambda_a, \lambda_b \in [0, 1]$  are predicted by the adaptor network and details are introduced in Section 2.4.

In Figure 2, we only illustrate one node and one bond retrieval as examples, but in practice, we conduct such a process for all atoms and bonds. Following LocalRetro (Chen and Jung 2021), after we get the  $P(t_a|M, a)$  and  $P(t_b|M, b)$  for each atom  $a$  and bond  $b$ , we will rank all non-zero predictions by their probability. The atom template and bonds templates are ranked together, and the top 50 predictions are our system’s final output.

## 2.4 Adaptor Network

To adaptively choose the  $T_A, T_B, \lambda_a$ , and  $\lambda_b$  for each atom and bond, we design a lightweight network to predict these values. The input to adaptor are hidden representation  $h_v, h_e$  from GNN side and distance list  $d(h_v, h_i), d(h_e, h_i)$  from the KNN side.

We use a one-layer GNN followed by a few fully connected (FC) layers for the network architecture. We use the the graph isomorphism network (GIN) with edge features (Hu et al. 2019) layer to capture both node feature  $h_v$  and edge feature  $h_e$ , which is formulated as:

$$h_v^{(g)} = W_{\text{vg}}((1 + \epsilon)h_v + \sum_{e \in \mathcal{E}(v)} \text{ReLU}(h_v + h_e)) + b_{\text{vg}}, \quad (5)$$

where the  $h_v^{(g)}$  is the output,  $\epsilon$  and  $W$  are learnable parameters of GIN, and the  $\mathcal{E}(v)$  is the set of edges around  $v$ . Meanwhile, we use the FC layer to project the KNN distances to extract the features that can be formulated as

$$h_v^{(k)} = W_{\text{vk}}(\{d(h_v, h_i)\}_{i=1}^K) + b_{\text{vk}}, \quad (6)$$

$$h_e^{(k)} = W_{\text{ek}}(\{d(h_e, h_i)\}_{i=1}^K) + b_{\text{ek}}, \quad (7)$$

where the brackets  $\{\cdot\}_{i=1}^K$  means building a K-dimensional vector. Finally, the feature from GNN and KNN are combined to a mixed representation, which are

$$h_v^{(o)} = \text{ReLU}(W_{\text{vo}}\text{ReLU}(h_v^{(g)} \| h_v^{(k)}) + b_{\text{vo}}), \quad (8)$$

$$h_e^{(o)} = \text{ReLU}(W_{\text{eo}}\text{ReLU}(h_e^{(g)} \| h_e^{(k)}) + b_{\text{eo}}), \quad (9)$$

where the  $\parallel$  denotes tensor concatenation and  $es$  and  $et$  are start and end node of edge  $e$ .

The  $T_A, \lambda_a$  are predicted by  $h_v^{(o)}$  and the  $T_B, \lambda_b$  are predicted by  $h_e^{(o)}$  by another FC layer. We also use sigmoid function  $\sigma$  to guarantee the  $\lambda_a, \lambda_b \in (0, 1)$  and clamp the  $T_A, T_B$  into range  $[1, 100]$ . Formally, we have

$$T_A = \max(1, \min(100, W_{ta}h_v^{(o)} + b_{ta})), \quad (10)$$

$$\lambda_a = \sigma(W_{la}h_v^{(o)} + b_{la}), \quad (11)$$

$$T_B = \max(1, \min(100, W_{tb}h_e^{(o)} + b_{tb}, 1, 100)), \quad (12)$$

$$\lambda_b = \sigma(W_{lb}h_e^{(o)} + b_{lb}). \quad (13)$$

Because all the formulas used here are differentiable, we optimize the adapter parameters  $W$  with gradient decent to minimize the template classification loss

$$\begin{aligned} \mathcal{L}_M = & -\frac{1}{|\mathcal{V}|} \sum_{a \in \mathcal{V}} \log P(\hat{t}_a | M, a) \\ & -\frac{1}{|\mathcal{E}|} \sum_{b \in \mathcal{E}} \log P(\hat{t}_b | M, b), \end{aligned} \quad (14)$$

for each target molecule  $M$  with node set  $\mathcal{V}$  and edge set  $\mathcal{E}$ . The  $P(\hat{t}_a | M), P(\hat{t}_b | M)$  are computed by Equation (3) and Equation (4). The  $\hat{t}_a, \hat{t}_b$  are the ground truth template.

## 3 Experiments

### 3.1 Experimental Settings

**Data.** Our experiments are based on the chemical reactions extracted from the United States Patent and Trademark Office (USPTO) literature. We use two versions of the USPTO benchmark: the USPTO-50K (Coley et al. 2017) and USPTO-MIT (Jin et al. 2017). The USPTO-50K contains 50k chemical reactions, split into 40k/5k/5k reactions as training, validation, and test, respectively. Meanwhile, the USPTO-MIT consists of about 479k reactions, and the split is 409k/40k/30k. All the partitions are the same as in previous works (Coley et al. 2017; Jin et al. 2017) to make fair comparisons. We also use the preprocess scripts by Chen and Jung (2021) to extract the reaction templates from these reactions, which leads to 658 and 20,221 reaction templates in USPTO-50K and USPTO-MIT.

**Implementation details.** We follow the same model configuration as LocalRetro (Chen and Jung 2021) to build the backbone GNN model. The feature extractor  $f$  is a 6-layer MPNN (Gilmer et al. 2017) followed by a single GRA layer (Chen and Jung 2021) with 8 heads. We use the hidden dimension 320 and dropout 0.2. The atoms’ and bonds’ input feature is extracted by DGL-LifeSci (Li et al. 2021). The prediction head  $h$  consists two dense layers with ReLU activation. The backbone model is optimized by Adam optimizer with a learning rate of 0.001 for 50 epochs. We also early stop the training when there is no improvement in the validation loss for five epochs. The configurations for backbone are all same as Chen and Jung (2021).

The implementation of KNN is based on the faiss (Johnson, Douze, and Jégou 2019) library with INDEXIVFPQ index for fast embedding searching, and the K of KNN is set

to 32. For the adapter network, we use the same hidden dimension as the backbone GNN. The adapter is also trained with Adam optimizer with a learning rate of 0.001. Considering the data size difference, we train the adapter for ten epochs and two epochs on the validation set of the USPTO-50K and USPTO-MIT datasets, respectively. The adapter with the best validation loss is used for test.

**Evaluation and baselines** Following previous works, our system will predict top-50 results for each target molecule and report the top-K accuracy where  $K=1, 3, 5, 10$ , and 50 by the script from Chen and Jung (2021). We also use representative baseline systems in recent years, include:

- Template-based methods: retrosim (Coley et al. 2017), neuralsym (Segler and Waller 2017), GLN (Dai et al. 2020), Hopfield (Seidl et al. 2021), and LocalRetro (Chen and Jung 2021);
- Semi-template based methods: G2Gs (Shi et al. 2021), RetroXpert (Yan et al. 2020), and GraphRtro (Somnath et al. 2021);
- Template-free methods: Transformer (Lin et al. 2020), MEGAN (Sacha et al. 2021), Chemformer (Irwin et al. 2021), GTA (Seo et al. 2021), and DualTF (Sun et al. 2021).

### 3.2 Main Results

The experimental results of the USPTO-50K benchmark are shown in Table 1 when the reaction type is unknown and in Table 2 when the reaction type is given. Meanwhile, the results on the USPTO-MIT benchmark are in Table 3. In these tables, we sort all systems by their top-1 accuracy and mark their type by filling the circle symbols. Our method (RetroKNN) is in the last row and highlighted in bold.

Comparing these accuracy numbers, we can find that our method outperforms the baseline systems with a large margin. When the reaction type is unknown, we achieved 57.2 points top-1 accuracy and improved the backbone result from LocalRetro by 3.8 points, which is a 7.1% relative gain.

Method	TPL.	K = 1	3	5	10	50
retrosim	●	37.3	54.7	63.3	74.1	85.3
neuralsym	●	44.4	65.3	72.4	78.9	83.1
MEGAN	○	48.1	70.7	78.4	86.1	93.2
G2Gs	◐	48.9	67.6	72.5	75.5	-
RetroXpert	◐	50.4	61.1	62.3	63.4	64.0
GTA	○	51.1	67.6	67.8	81.6	-
Hopfield	●	51.8	74.6	81.2	88.1	94.0
GLN	●	52.5	69.0	75.6	83.7	92.4
LocalRetro	●	53.4	77.5	85.9	92.4	97.7
Dual-TF	○	53.6	70.7	74.6	77.0	-
GraphRetro	◐	53.7	68.3	72.2	75.5	-
Chemformer	○	54.3	-	62.3	63.0	-
<b>RetroKNN</b>	<b>●</b>	<b>57.2</b>	<b>78.9</b>	<b>86.4</b>	<b>92.7</b>	<b>98.1</b>

Table 1: Top-K exact match accuracy on the USPTO-50K dataset when the reaction type is unknown. The ●, ◐, and ○ denote template-based, semi-template, and template-free, respectively. Systems are ordered by top-1 accuracy.

Method	TPL.	K = 1	3	5	10	50
retrosim	●	52.9	73.8	81.2	88.1	-
neuralsym	●	55.3	76.0	81.4	85.1	-
MEGAN	○	60.7	82.0	87.5	91.6	95.3
G2Gs	●	61.0	81.3	86.0	88.7	-
RetroXpert	●	62.1	75.8	78.5	80.9	-
GraphRetro	●	63.9	81.5	85.2	88.1	-
LocalRetro	●	63.9	86.8	92.4	96.3	97.9
GLN	●	64.2	79.1	85.2	90.0	93.2
Dual-TF	○	65.7	81.9	84.7	85.9	-
<b>RetroKNN</b>	<b>●</b>	<b>66.7</b>	<b>88.2</b>	<b>93.6</b>	<b>96.6</b>	<b>98.4</b>

Table 2: Top-K exact match accuracy on the USPTO-50K dataset when the reaction type is given. The ●, ●, and ○ denote template-based, semi-template, and template-free, respectively. Systems are ordered by top-1 accuracy.

Method	TPL.	K = 1	3	5	10	50
Seq2Seq	○	46.9	61.6	66.3	70.8	-
neuralsym	●	47.8	67.9	74.1	80.2	-
Transformer	○	54.1	71.8	76.9	81.8	-
LocalRetro	●	54.1	73.7	79.4	84.4	90.4
<b>RetroKNN</b>	<b>●</b>	<b>60.6</b>	<b>77.1</b>	<b>82.3</b>	<b>87.3</b>	<b>92.9</b>

Table 3: Top-K exact match accuracy on the USPTO-MIT dataset. The ● and ○ denote template-based and template-free methods. Systems are ordered by top-1 accuracy.

When the reaction type is given, we also improve the top-1 accuracy by 2.8 points from 63.9 to 66.7. Meanwhile, on USPTO-MIT, our method shows 60.6 points top-1 accuracy with a 6.5 points improvement or 12% relative gain. More importantly, these top-1 accuracies are also better than other strong baselines and state-of-the-art, demonstrating the effectiveness of our method.

At the same time, we achieved 78.9 points top-3 accuracy and 86.4 points accuracy in USPTO-50K when the reaction type is unknown, which are also much higher than baselines. For the top-10 and top-50 accuracy, we get 92.7 and 98.1 points accuracy. Considering that the accuracy is already very high, the improvement is still significant.

To sum up, the local template retrieval method efficiently improves the retrosynthesis prediction accuracy.

## 4 Study and Analysis

### 4.1 Case Study

**Retrieval case study.** To better understand if we can retrieve useful reactions by the hidden representations, we conducted case studies on the USPTO-50K datasets, and the results are shown in Figure 3. We first select an atom-template reaction and the first bond-template reaction from the data. Next, we query the atom and bond store by the corresponding atom and bond. Finally, for each retrieved template, we show the original target molecule in the training data, where the reaction atom/bond is highlighted by green background.

The bond-template and atom-template reactions are available in the figure’s first and second rows. In each row, we first show the target molecule M of the reaction and then five neighbors of M. From these cases, we can find that the neighborhoods retrieved by hidden representations can effectively capture the local structure of molecules. For example, the carbon-nitrogen bond retrieves all neighbors in the edge-template reaction. Moreover, all carbon atoms are surrounded by oxygen in a double bond ( $=O$ ) and a trifluorocarbon ( $-CF_3$ ), and all nitrogen atoms are connected to an aromatic ring. Meanwhile, for the node-template reaction, all retrieved atoms are the oxygen atoms that are connected to a phenyl. In conclusion, retrieving molecules with hidden representations is efficient because it can capture the local structure well. Therefore, we can improve the prediction accuracy by using the retrieved templates.

**Adapter case study.** We show three representative cases for the effect of adapter in Table 4. In each row, we show the target molecule and ground truth template id, then the  $\lambda$  and T output by the adapter, and finally the GNN prediction and KNN retrieved neighbors. When the GNN prediction is accurate in the first row, the adapter will generate a high  $\lambda$  value (e.g., 0.96) so that the GNN output has a higher weight. However, when that is not the case (the second and third row), the  $\lambda$  tends to be lower (e.g., 0.14), which gives more weight to KNN prediction. Meanwhile, when only the N1 has the correct prediction (the second row), the adapter tends to output a small T (e.g., 7.89) to make the sharp distribution that gives more weight to N1’s prediction. On the contrary (the third row), the adapter tends to output a larger value (e.g., 19.36) so that more neighbors can contribute to the final output. Moreover, our statistics show that when  $\lambda < 0.5$ , the GNN and KNN accuracy are 46.9% and 69.2%, showing that KNN is complementary to GNN prediction.

### 4.2 Zero-shot and Few-shot Study

We modify the USPTO-50K dataset to zero-shot and few-shot versions to study the domain adaptation ability of our method. Specifically, in the USPTO-50K data, each reaction has its reaction class available in class 1 to 10. To build the zero-shot data, we filter the train and validation data by removing all reactions with reaction class 6 to 10 and only keeping those with reaction class 1 to 5. Similarly, to build the few-shot data, we only keep 10% of reactions that have class 6 to 10. Finally, we evaluate the performance of these new data with the LocalRetro baseline and our RetroKNN method. The results are summarized in Figure 4.

From these plots, we notice that zero-shot is a challenging setting for conventional template-based methods, which is a known shortcoming of this kind of methods. However, when combined with KNN, our system can generate meaningful results. For example, in reaction class 8, the RetroKNN has 6.1 points top-5 accuracy and 9.8 points top-10 accuracy in the zero-shot data. The few-shot setting is easier than the zero-shot because a few examples are available during training. Nevertheless, the RetroKNN also outperforms baseline on all reaction types. On average, the RetroKNN improved 8.56 points top-5 accuracy and 5.64 points top-10 accuracy. These results show that our method is can also

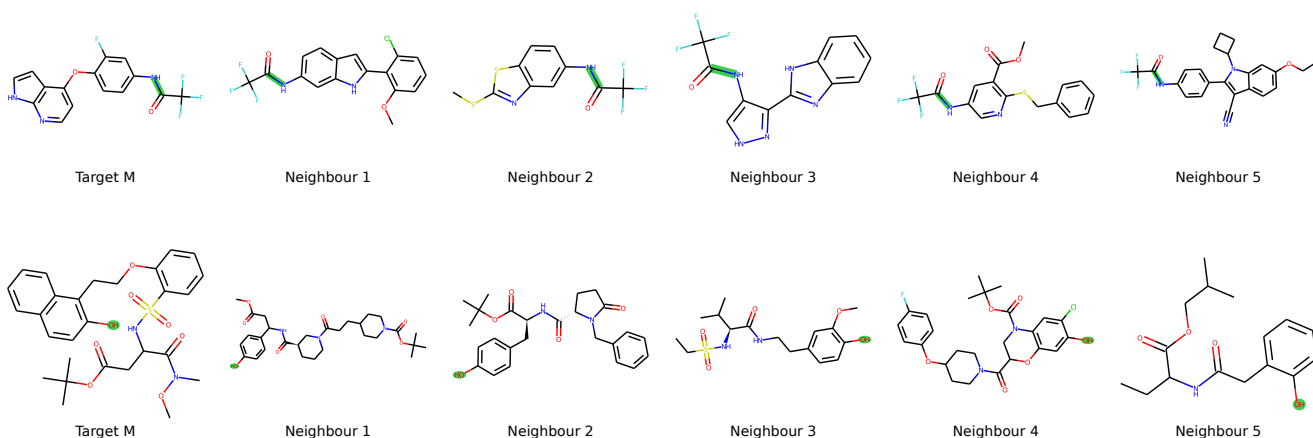


Figure 3: Case study of retrieved molecules. The bonds and atoms used in retrieval are highlighted by green background. The first column shows the target molecules, and the rest show five neighbourhood targets from the training data.

Target Molecule	GT.	$\lambda$	T	GNN	N1	N2	N3	N4	N5
<chem>Cc1ccc(-c2cccnc2C#N)cc1</chem>	<b>b542</b>	0.96	21.42	<b>b542</b>	b519 (67.51)	b519 (77.35)	b519 (77.35)	b519 (77.35)	b0 (104.00)
<chem>CCOc1ccc(C[C@H](NC(=O)C(F)(F)F)C(=O)O)cc1</chem>	<b>b524</b>	0.14	7.89	b495	<b>b524</b> (22.79)	b523 (33.84)	b495 (67.3)	b495 (76.21)	b495 (76.55)
<chem>CC1(C)CC(=O)N(Cc2ccccc2)c2ccc(C#Cc3ccc(C(=O)O)cc3)cc21</chem>	<b>a121</b>	0.02	19.36	a124	<b>a121</b> (34.41)	<b>a121</b> (57.3)	<b>a121</b> (58.4)	a0 (59.91)	a0 (61.17)

Table 4: Case study of parameter T and  $\lambda$ . The GT. denotes ground truth template id, GNN denotes the GNN prediction, and N1 to N5 denotes five neighbors. The prefix a, b of template id means it is an atom or bond template. We show each neighbor’s distance in the brackets below template id. The correct predictions are highlighted in bold.

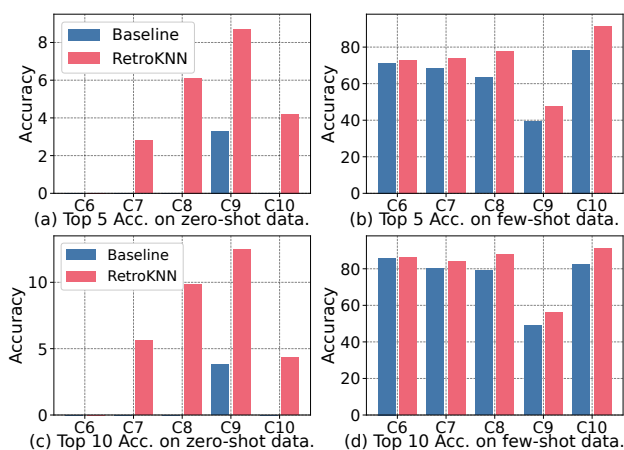


Figure 4: Top-5 (a, b) and top-10 (c, d) accuracy (Acc.) on the zero-shot (a, c) and few-shot (b, d) data. The columns C6 to C10 denote different reaction classes.

improve the performance on zero/few-shot data, which are important scenarios in this field.

### 4.3 Ablation Study

We conducted an ablation study on the USPTO-50K dataset to study the contributions of different components, and the results are shown in Table 5. We show the top-1 accuracy in the table by comparing different systems. The system ① is the LocalRetro baseline without using KNN, which achieved 53.4 points accuracy. In system ②, we add the KNN without using the adapter. To find the optimal paramters, we conduct comprehensive grid search on by  $T \in \{1, 5, 25, 50\}$  and  $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ , which leads to total 20 combinations. We select the parameters by the validation loss and finally get the 56.3 points accuracy. Furthermore, in system ③, we add the adapter only for T and keep the  $\lambda$  same as system ②. Similarly, we only add the adapter only for  $\lambda$  in system ④. The system ⑤ is the full RetroKNN model.

Comparing the system ① with others that using KNN, we can find that introducing KNN to this task can effectively improve the model performance. These numbers show that

ID	System	Accuracy
①	Baseline	53.4
②	+ KNN	56.3
③	+ KNN, adaptive T	56.7
④	+ KNN, adaptive $\lambda$	56.8
⑤	+ KNN, adaptive T, adaptive $\lambda$	57.2

Table 5: Ablation study on the USPTO-50K dataset when the reaction type is unknown.

#Retrieved reactions	1	4	8	16	32
Accuracy	55.6	57.4	57.1	56.9	57.2

Table 6: Study on the number of retrieved reactions by KNN.

the local template retrieval is vital for the system. Meanwhile, comparing system ③④ to system ②, we notice that adding both T and  $\lambda$  adapter is helpful. Finally, when both parameters are adaptively predicted in system ⑤, the accuracy can be boosted to 57.2, showing that they can work together effectively. Therefore, all components are necessary for this system.

#### 4.4 Retrieved Templates Size

In Table 6, we show how the number of retrieved reactions (i.e., K of KNN) affects the model performance. More specifically, in the KNN search, we set the  $K \in [1, 4, 8, 16, 32]$ , then train adapters for each of them. Finally, we report the top-1 accuracy in the table.

From these results, we first observe that only adding one retrieved template ( $K=1$ ) can improve the accuracy from 53.4 to 55.6. When K is  $\geq$  than 4, the accuracy can be further improved to around 57 points. There will be no further significant improvement when more reactions are retrieved, nor will more received templates hurt the performance. We suppose it is because there is already enough information to improve the accuracy as the templates far from the query will contribute less to the prediction.

#### 4.5 Inference Latency

In Table 7, we study the datastore size and the inference latency. The last two rows present the latency with or without retrieval during inference, which are measured on a machine with a single NVIDIA A100 GPU. Each latency value, which is the average run time per reaction, is measured with ten independent runs. In the USPTO-50K dataset, we observe that the average latency increased from 2.71 ms to 3.31 ms, which is about 0.6 ms for each reaction. The extra latency is a little more prominent for the USPTO-MIT dataset because it is about ten times larger than the USPTO-50K. However, considering the hours or even days that a more accurate system can save for chemists, the extra ten-millisecond cost is not a real obstacle to the practical use of this method. Finally, some work (He, Neubig, and Berg-Kirkpatrick 2021; Meng et al. 2021) show that the KNN

Dataset	USPTO-50K	USPTO-MIT
$ \mathcal{D}_{\text{train}} $	40k	409k
$ \mathcal{S}_A $	1,039k	10,012k
$ \mathcal{S}_B $	2,241k	21,495k
Latency w/o KNN	$2.71 \pm 0.02$ ms	$3.51 \pm 0.05$ ms
Latency w/ KNN	$3.31 \pm 0.09$ ms	$14.69 \pm 0.29$ ms

Table 7: Study of the datastore size and inference latency.

speed can be further accelerated, and we would like to add these techniques in future work.

## 5 Related Work

### 5.1 Retrosynthesis Prediction

Retrosynthesis prediction is an essential task for scientific discovery and have achieved promising results in recent years (Segler and Waller 2017; Liu et al. 2017; Coley et al. 2017; Tetko et al. 2020; Irwin et al. 2021; Dai et al. 2020; Yan et al. 2020; Seidl et al. 2021; Chen and Jung 2021; Shi et al. 2021; Somnath et al. 2021; Wan et al. 2022). A few research also use retrieval mechanisms for this task. For example, Seidl et al. (2021) use Hopfield networks to select templates, and Lee et al. (2021) use retrieval method to fetch molecules from a database. Being differently, we are the first to combine deep learning and KNN retrieval in this task.

### 5.2 Retrieval Methods

Retrieving from data store or memory to improve the machine learning model’s performance is an important research topic. SVM-KNN (Zhang et al. 2006) first combines the SVM and KNN for recognition tasks. Furthermore, the KNN-LM (Khandelwal et al. 2020) and KNN-MT (Khandelwal et al. 2021) have shown promising results when combining KNN with Transformer networks. Meanwhile, He, Neubig, and Berg-Kirkpatrick (2021); Meng et al. (2021) study the speed of retrieval methods and Zheng et al. (2021) study the adaptation problem. However, we are the first to combine the strong capability of KNN with GNN and use them on the retrosynthesis task.

## 6 Conclusion

Retrosynthesis prediction is essential for scientific discovery, especially drug discovery and healthcare. In this work, we propose a novel method to improve prediction accuracy using local template retrieval. We first build the atom and bond stores with the training data and a trained GNN and retrieve templates from these stores during inference. The retrieved templates are combined with the original GNN predictions to make the final output. We further leverage a lightweight adapter to adaptively predict the weights to integrate the GNN predictions and retrieved templates. We greatly advanced the prediction performance on two widely used benchmarks, the USPTO-50K and USPTO-MIT, reaching 57.2 and 60.6 points for top-1 accuracy. These results demonstrate the effectiveness of our methods.



## Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. This work was supported by National Natural Science Foundation of China (NSFC Grant No. 62122089 and No. 61876196), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. We also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC. Rui Yan is supported by Beijing Academy of Artificial Intelligence (BAAI).

## References

- Chen, S.; and Jung, Y. 2021. Deep Retrosynthetic Reaction Prediction using Local Reactivity and Global Attention. *JACS Au*.
- Coley, C. W.; Rogers, L.; Green, W. H.; and Jensen, K. F. 2017. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Central Science*, 3(12): 1237–1245.
- Corey, E. J. 1991. *The logic of chemical synthesis*. Ripol Classic.
- Dai, H.; Li, C.; Coley, C. W.; Dai, B.; and Song, L. 2020. Retrosynthesis Prediction with Conditional Graph Logic Network. *arXiv:2001.01408*.
- Dong, J.; Zhao, M.; Liu, Y.; Su, Y.; and Zeng, X. 2021. Deep learning in retrosynthesis planning: datasets, models and tools. *Briefings in Bioinformatics*, 00(August): 1–15.
- Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. PMLR.
- He, J.; Neubig, G.; and Berg-Kirkpatrick, T. 2021. Efficient Nearest Neighbor Language Models. *EMNLP*.
- Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for Pre-training Graph Neural Networks. *arXiv:1905.12265*.
- Irwin, R.; Dimitriadis, S.; He, J.; and Bjerrum, E. J. 2021. Chemformer: A Pre-Trained Transformer for Computational Chemistry. *Machine Learning: Science and Technology*, 3.
- Jin, W.; Coley, C.; Barzilay, R.; and Jaakkola, T. 2017. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in neural information processing systems*, 30.
- Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3): 535–547.
- Khandelwal, U.; Fan, A.; Jurafsky, D.; Zettlemoyer, L.; and Lewis, M. 2021. Nearest Neighbor Machine Translation. In *International Conference on Learning Representations*.
- Khandelwal, U.; Levy, O.; Jurafsky, D.; Zettlemoyer, L.; and Lewis, M. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*.
- Lee, H.; Ahn, S.; Seo, S.-W.; Song, Y. Y.; Yang, E.; Hwang, S. J.; and Shin, J. 2021. RetCL: A Selection-based Approach for Retrosynthesis via Contrastive Learning. In *IJ-CAI*, 2673–2679.
- Li, M.; Zhou, J.; Hu, J.; Fan, W.; Zhang, Y.; Gu, Y.; and Karypis, G. 2021. DGL-LifeSci: An Open-Source Toolkit for Deep Learning on Graphs in Life Science. *ACS Omega*.
- Lin, K.; Pei, J.; Lai, L.; and Xu, Y. 2020. Automatic Retrosynthetic Pathway Planning Using Template-free Models. *Chemical Science*.
- Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; and Pande, V. 2017. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Central Science*, 3(10): 1103–1113.
- Meng, Y.; Li, X.; Zheng, X.; Wu, F.; Sun, X.; Zhang, T.; and Li, J. 2021. Fast Nearest Neighbor Machine Translation. *arXiv:2105.14528*.
- Sacha, M.; Błaz, M.; Byrski, P.; Dabrowski-Tumanski, P.; Chrominski, M.; Loska, R.; Włodarczyk-Pruszyński, P.; and Jastrzebski, S. 2021. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7): 3273–3284.
- Segler, M. H. S.; and Waller, M. P. 2017. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chemistry - A European Journal*, 23(25): 5966–5971.
- Seidl, P.; Renz, P.; Dyubankova, N.; Neves, P.; Verhoeven, J.; Segler, M.; Wegner, J. K.; Hochreiter, S.; and Klambauer, G. 2021. Modern Hopfield Networks for Few- and Zero-Shot Reaction Template Prediction. *arXiv:2104.03279*.
- Seo, S.-W.; Young Song, Y.; Yong Yang, J.; Bae, S.; Lee, H.; Shin, J.; Ju Hwang, S.; and Yang, E. 2021. GTA: Graph Truncated Attention for Retrosynthesis. *AAAI*.
- Shi, C.; Xu, M.; Guo, H.; Zhang, M.; and Tang, J. 2021. A Graph to Graphs Framework for Retrosynthesis Prediction. *arXiv:2003.12725*.
- Somnath, V. R.; Bunne, C.; Coley, C. W.; Krause, A.; and Barzilay, R. 2021. Learning Graph Models for Retrosynthesis Prediction. *arXiv:2006.07038*.
- Sun, R.; Dai, H.; Li, L.; Kearnes, S.; and Dai, B. 2021. Towards understanding retrosynthesis by energy-based models. *NeurIPS*, 9.
- Tetko, I. V.; Karpov, P.; Van Deursen, R.; and Godin, G. 2020. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1): 5575.
- Wan, Y.; Liao, B.; Hsieh, C.-Y.; and Zhang, S. 2022. Retroformer: Pushing the Limits of Interpretable End-to-end Retrosynthesis Transformer. *arXiv:2201.12475*.
- Yan, C.; Ding, Q.; Zhao, P.; Zheng, S.; Yang, J.; Yu, Y.; and Huang, J. 2020. Retroxpert: Decompose retrosynthesis prediction like a chemist. *Advances in Neural Information Processing Systems*, 33: 11248–11258.
- Zhang, H.; Berg, A. C.; Maire, M.; and Malik, J. 2006. SVM-KNN: Discriminative nearest neighbor classification



for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 2126–2136. IEEE.

Zheng, X.; Zhang, Z.; Guo, J.; Huang, S.; Chen, B.; Luo, W.; and Chen, J. 2021. Adaptive Nearest Neighbor Machine Translation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.