biomolecules

MDPI

Article

RetroComposer: Composing Templates for Template-Based Retrosynthesis Prediction

Chaochao Yan [1], Peilin Zhao [2], Chan Lu [2], Yang Yu [2] and Junzhou Huang [1,*]

[1] Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA
[2] Tencent AI Lab, Shenzhen 518054, China
* Correspondence: jzhuang@uta.edu

Abstract: The main target of retrosynthesis is to recursively decompose desired molecules into available building blocks. Existing template-based retrosynthesis methods follow a template selection stereotype and suffer from limited training templates, which prevents them from discovering novel reactions. To overcome this limitation, we propose an innovative retrosynthesis prediction framework that can compose novel templates beyond training templates. As far as we know, this is the first method that uses machine learning to compose reaction templates for retrosynthesis prediction. Besides, we propose an effective reactant candidate scoring model that can capture atom-level transformations, which helps our method outperform previous methods on the USPTO-50K dataset. Experimental results show that our method can produce novel templates for 15 USPTO-50K test reactions that are not covered by training templates. We have released our source implementation.

Keywords: drug discovery; retrosynthesis; reaction template; machine learning; recurrent neural network; and graph neural network

Citation: Yan, C.; Zhao, P.; Lu, C.; Yu, Y.; Huang, J. RetroComposer: Composing Templates for Template-Based Retrosynthesis Prediction. *Biomolecules* **2022**, *12*, 1325. https://doi.org/10.3390/biom12091325

Academic Editors: Cameron Mura and Lei Xie

Received: 24 June 2022
Accepted: 15 September 2022
Published: 19 September 2022

1. Introduction

Retrosynthesis plays a significant role in organic synthesis planning, in which target molecules are recursively decomposed into available commercial building blocks. This analysis mode was firstly formulated in the pioneering work [1,2] and now is one of the fundamental paradigms in modern chemical society. Since then numerous retrosynthesis prediction algorithms have been proposed to aid or even automate the retrosynthesis analysis. However, the performance of existing methods is still not satisfactory. The massive search space is one of the major challenges of retrosynthesis considering that on the order of 10^7 compounds and reactions [3] have been reported in synthetic–organic knowledge. The other challenge is that there are often multiple viable retrosynthesis pathways and it is challenging to decide the most appropriate route since the feasibility of a route is often compounded by several factors, such as reaction conditions, reaction yield, potential toxic byproducts, and the availability of potential reactants [4].

Most of existing machine-learning-powered retrosynthesis methods focus on the single-step version. These methods are broadly grouped into template-based and template-free major categories. Templates-free methods [4–9] usually rely on deep learning models to directly generate reactants. One effective strategy is to formulate the retrosynthesis prediction as a sequence translation task and generate SMILES [10] sequences directly using sequence-to-sequence models such as Seq2Seq [5], SCROP [6], and AT [11]. SCROP [6] proposes to use a second transformer to correct the initial wrong predictions. Translation-based methods are simple and effective, but lack interpretability behind the prediction. Another representative paradigm is to first find a reaction center and split the target accordingly to obtain hypothetical units called synthons, and then generate reactants incrementally from these synthons, such as in RetroXpert [4], G2Gs [7], RetroPrime [12], and GraphRetro [13].

On the other hand, template-based methods are receiving less attention than the rapid surge of template-free methods. Template-based methods conduct retrosynthesis based on either hand-encoded rules [14] or automatically extracted retrosynthesis templates [15]. Templates encode the minimal reaction transformation patterns, and are straightforwardly interpretable. The key procedure is to select applicable templates to apply to targets [15–18]. Template-based methods have been criticized for the limitation that they can only infer reactions covered by training templates and cannot discover novel reactions [4,19].

In this work, we propose a novel template-based single-step retrosynthesis framework to overcome the mentioned limitation. Unlike previous methods only selecting from training templates, we propose to compose templates with basic template building blocks (molecule subgraphs) extracted from training templates. Specifically, our method composes templates by first selecting appropriate product and reactant molecule subgraphs iteratively, and then annotates atom transformations between the selected subgraphs. This strategy enables our method to discover novel templates from training subgraphs, since the reaction space of our method is the exponential combination of these extracted template subgraphs. What is more, we design an effective reactant scoring model that can capture atom-level transformation information. Thanks to the scoring model, our method achieves state-of-the-art (SOTA) Top-1 accuracy of 54.5% and 65.9% on the USPTO-50K dataset without and with reaction types, respectively. Our contributions are summarized as: (1) we propose a first-ever template-based retrosynthesis framework to compose templates, which can discover novel reactions beyond the training data; (2) we design an effective reactant scoring model that can capture atom-level transformations, which contributes significantly to the superiority of our method; (3) the proposed method achieves 54.5% and 65.9% Top-1 accuracy on the benchmark dataset USPTO-50K without and with reaction types, respectively, which establishes the new SOTA performance. The code is available at https://github.com/uta-smile/RetroComposer (accessed on 10 September 2022).

2. Related Work

Recently there has been an increasing amount of work using machine learning methods to solve the retrosynthesis problem. These methods can be categorized into template-based [15–18,20] and template-free approaches [4,5,7,13,21]. Template-based methods extract templates from training data and build models to learn the corresponding relationship between products and templates. RetroSim [15] selects the templates based on the fingerprint similarity between products and reactions. NeuralSym [16] uses a neural classification model to select corresponding templates. However, this method does not scale well with an increasing number of templates. To mitigate the problem, [20] adopts a multi-scale classification model to select templates according to a manually defined template hierarchy. GLN [18] proposes a graph logic network to model the decomposed template hierarchy by first selecting reaction centers within the targets, and then only consider templates that contain the selected reaction centers. The decomposition strategy can reduce the search space significantly. GLN models the relationship between reactants and templates jointly by applying selected templates to obtain reactants, which are also used to optimize the model simultaneously.

Template-free methods do not rely on retrosynthesis templates. Instead, they construct models to predict reactants from products directly. Translation-based methods [6,11,22,23] use SMILES to represent molecules and treat the problem as a sequence-to-sequence task. MEGAN [8] treats the retrosynthesis problem as a graph transformation task, and trains the model to predict a sequence of graph edits that can transform the product into the reactants. To imitate a chemist's approach to the retrosynthesis, two-step methods [4,7,12,13] first perform reaction center recognition to obtain synthons by disconnecting targets according to the reaction center, and then generate reactants from the synthons. G2Gs [7] treats the reactant generation process as a series of graph editing operations and utilizes a variational graph generation model to implement the generation process. RetroXpert [4] converts the synthon into SMILES to generate reactants as a translation task. GraphRetro [13] also adopts

a similar framework and generates the reactants by attaching leaving groups to synthons. Dual model [9] proposes a general energy-based model framework that integrates both sequence- and graph-based models, and performs consistent training over forward and backward prediction directions.

3. Preliminary Knowledge

3.1. Retrosynthesis and Template

Single-step retrosynthesis predicts a set of reactant molecules given a target product, as shown in Figure 1a. Note that the product and reactant molecules are atom-mapped, which ensures that every product atom is uniquely mapped to a reactant atom. Templates are reaction rules extracted from chemical reactions. They are composed by reaction centers and encode the atom and bond transformations during the reaction process. The illustrated template in Figure 1b consists of a product subgraph (upper) and reactant subgraphs (lower). The subgraph patterns are highlighted in pink within the corresponding molecule graphs.

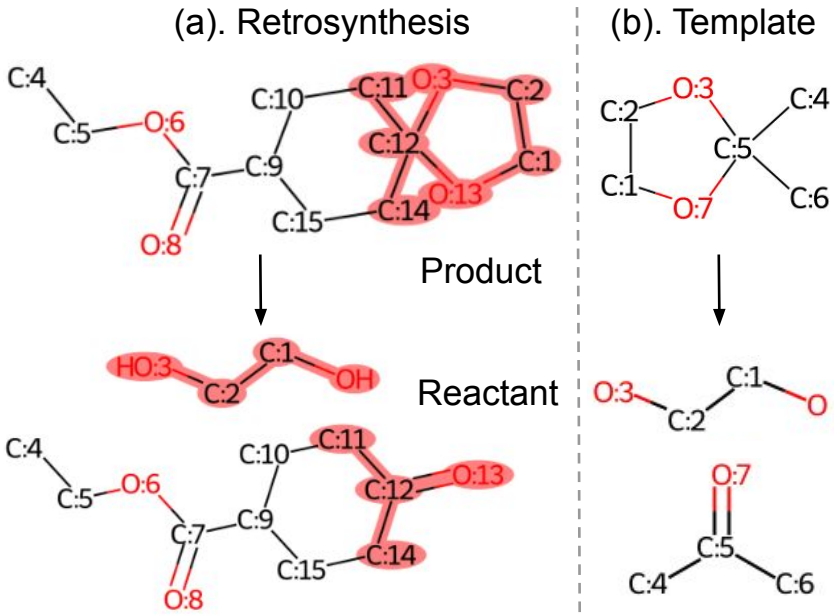

Figure 1. A retrosynthesis example from USPTO-50K dataset and its template extracted using an open-source toolkit. Note that the product and reactant are atom-mapped. The product and reactant subgraphs in (**b**) are highlighted in pink within the product and reactant molecule graphs in (**a**), respectively.

3.2. Molecule Graph Representation

The graph representation of a molecule or subgraph pattern is denoted as $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of graph nodes (atoms) and edges (bonds), respectively. Following previous work [4,18], each bond is represented as two directed edges. Initial node and edge features can be easily collected for the learning purpose.

3.3. Graph Attention Networks

Graph neural networks [24] are especially good at learning node- and graph-level embeddings of molecule data. In this work, we adapt graph attention networks (GATs) [25] to incorporate bond features. The GAT layer updates a node embedding by aggregating its neighbor's information. The modified GAT concatenates edge embeddings with the associated incoming node embeddings before each graph message passing. The input of the GAT layer is node embeddings $\{v_i | \forall i \in \mathcal{V}\}$ and edge features $\{e_{i,j} | (i,j) \in \mathcal{E}\}$, and the output updated node embeddings $\{v'_i | \forall i \in \mathcal{V}\}$. Each node embedding is updated with a shared parametric function t_θ:

$$v_i' = t_\theta(v_i, \text{AGGREGATE}(\{[v_j||e_{i,j}]|\forall j \in \mathcal{N}(i)\})), \quad (1)$$

where $\mathcal{N}(i)$ are neighbor nodes of v_i and $||$ is the concatenation operation. The AGGREGATE of GAT adopts an attention-based mechanism to adaptively weight the neighbor information. A scoring function $c(i,j)$ computes the importance of the neighbor node j to node i:

$$c(i,j) = \text{LeakyReLU}(w^T[\boldsymbol{W}_1 v_i||\boldsymbol{W}_1 v_j||\boldsymbol{W}_2 e_{i,j}]), \quad (2)$$

where w is a learnable vector parameter and each $\boldsymbol{W}$ is a learnable matrix parameter. These importance scores are normalized using the Softmax function across the neighbor nodes $\mathcal{N}(i)$ of the node i to obtain attention weights:

$$\alpha(i,j) = \text{Softmax}_j(c(i,j)) = \frac{\exp(c(i,j))}{\sum_{j' \in \mathcal{N}(i)} \exp(c(i,j'))}. \quad (3)$$

The modified GAT instances t_θ and updates the node embedding as the non-linear function σ activated weighted-sum of the transformed embeddings of its neighbor nodes:

$$v_i' = \sigma(\sum_{j \in \mathcal{N}(i)} \alpha(i,j) * \boldsymbol{W}_3[\boldsymbol{W}_1 v_j||\boldsymbol{W}_2 e_{i,j}]). \quad (4)$$

GAT is usually stacked by multiple layers and enhanced with multi-head attention [26]. Please refer to [25] for more details.

3.4. Graph-Level Embedding

After obtaining the output node embeddings from the GAT, a graph READOUT operation can be used to obtain the graph-level embedding. Inspired by [27], we aggregate and concatenate the output node embeddings from all GAT layers to learn structure-aware node representations from different neighborhood ranges:

$$\text{emb}_G = \text{READOUT}(\{v_{i,1}||v_{i,2}||...||v_{i,L}|\forall i \in \mathcal{V}\}). \quad (5)$$

where $v_{i,l}$ is the output embedding of node i after the lth GAT layer. The READOUT can be any permutation-invariant operation (e.g., mean, sum, max). We adopt the global soft attention layer from [28] as the READOUT function for molecule graphs due to its excellent performance.

4. Methods

We propose to compose retrosynthesis templates from a predefined set of template building blocks; then, these composed templates are applied to target products to obtain the associated reactants. Unlike previous template-based methods [15–18] only selecting from training templates, our method can discover novel templates that are beyond the training templates. To further improve the retrosynthesis prediction performance, we design a scoring model to evaluate the suitability of product and candidate reactant pairs. The scoring procedure acts as a verification step and it plays a significant role in our method.

The overall pipeline of our method is shown in Figure 2. Our method tackles retrosynthesis in two stages. The first stage is to compose retrosynthesis templates with a TCM, which composes retrosynthesis templates by selecting template building blocks and then assembling them. In the second stage, the obtained templates are applied to the target product to generate the associated reactants. After that, we utilize a powerful RSM to evaluate the generated reactants for each product. During evaluation, the probability scores of both stages are linearly combined to rank Top-K reactant predictions. In following sections, we will detail each stage of our method.

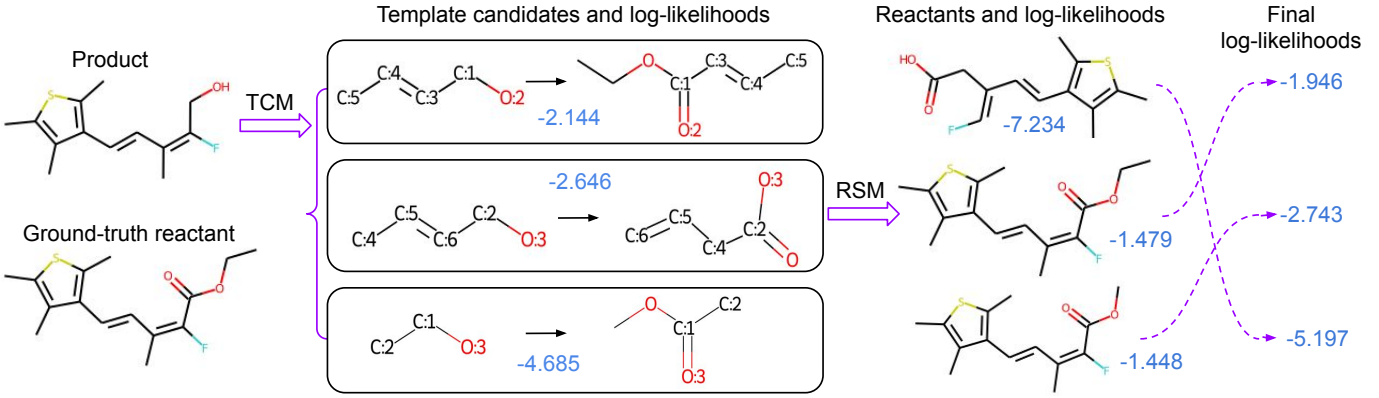

Figure 2. The overall pipeline of our proposed method. Given the desired product as shown at the top left, single-step retrosynthesis finds the ground-truth reactant as shown at the bottom left. Numbers indicated in blue are the corresponding log-likelihoods of our models, and the log-likelihoods of the template composer model (TCM) and the reactant scoring model (RSM) are combined to obtain the final ranking of the reactants. In this example, combining log-likelihoods of TCM and RSM helps to find the correct Top-1 reactant.

4.1. Compose Retrosynthesis Templates

Template-based retrosynthesis methods are criticized for their limitation of not generalizing to unseen reactions, since all existing template-based methods follow a similar procedure to select applicable templates from the extracted training templates. To overcome the above limitation, we propose a different pipeline to find template candidates. As illustrated in Figure 3, our method first selects product and reactant subgraphs sequentially from the corresponding subgraph vocabularies, which is detailed in Section 4.1.1. Then, these selected subgraphs are assembled into templates with properly assigned atom mappings, as detailed in Section 4.1.4. As far as we know, this is the first attempt to compose retrosynthesis templates instead of simple template selection. During evaluation, a beam search algorithm [29] is utilized to find Top-K predicted templates. Reactants can be obtained by applying templates to the target molecule.

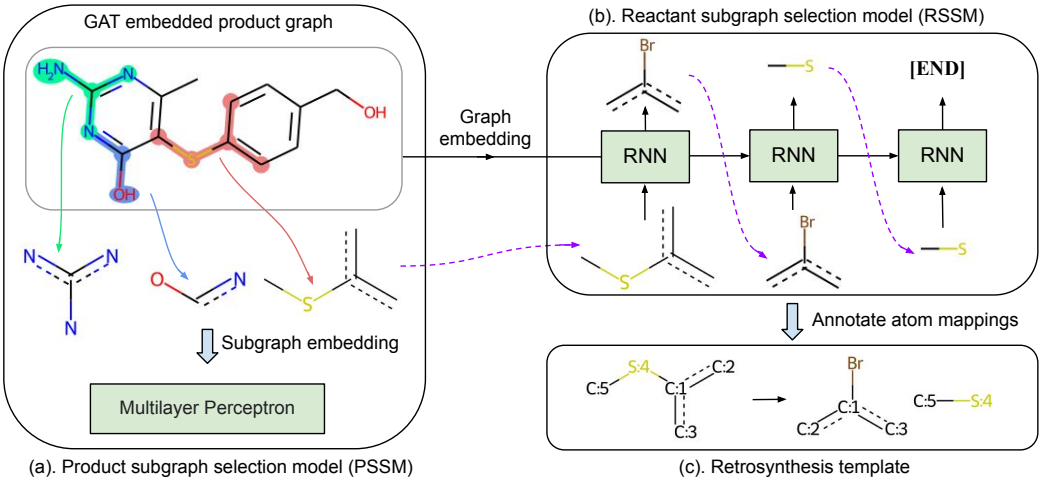

Figure 3. The workflow of our template composer model: (**a**) selecting a proper product subgraph from product subgraph candidates with PSSM, (**b**) selecting reactant subgraphs sequentially from reactant subgraph vocabulary with RSSM, and (**c**) annotating atom mappings between the product and reactant subgraphs to obtain a template.

4.1.1. Subgraph Selection

We denote a subgraph pattern as f, the product and reactant subgraphs for a template as f_p and f_r, respectively, and the product and reactant subgraph vocabulary for the dataset as $\mathcal{F}_P$ and $\mathcal{F}_R$, respectively. To build the product subgraph vocabulary $\mathcal{F}_P$ and reactant subgraph vocabulary $\mathcal{F}_R$, retrosynthesis templates extracted from training data are split into separate subgraphs to collect unique subgraph patterns. We build separate vocabularies for the product and reactant subgraphs due to their essential difference. Product subgraphs represent reaction centers and are more generalizable, while reactant subgraphs may contain extra leaving groups, which are more specific to the reaction type and the desired target. We find this strategy works well in practice.

4.1.2. Product Subgraph Selection

To compose retrosynthesis templates for a desired target, the first step is to choose proper f_p from the vocabulary $\mathcal{F}_P$. In this work, we focus on the single-product reactions; therefore, there is only a single product subgraph pattern. Note that there may be multiple viable retrosynthesis templates for each reaction, so each target may have several applicable product subgraphs. The set of applicable product subgraphs are denoted as $\mathcal{F}_a$. Starting with any applicable product subgraph in $\mathcal{F}_a$ may obtain a applicable retrosynthesis template for the target. Here, $\mathcal{F}_a \subseteq \mathcal{F}_P$ because all applicable product subgraphs must be in the vocabulary $\mathcal{F}_P$.

Each product molecule graph G_p contains only a limited set of candidate subgraphs $\mathcal{F}_c$ predefined in the vocabulary $\mathcal{F}_P$. Three candidate subgraphs are illustrated in Figure 3a. The candidate subgraphs for each target can be obtained offline by checking the existence of every product subgraph from $\mathcal{F}_P$ in the product graph G_p. Therefore, we only need to consider the candidate subgraphs $\mathcal{F}_c$ to guide the selection process [18] when selecting a product subgraph. Here, $\mathcal{F}_a \subseteq \mathcal{F}_c \subseteq \mathcal{F}_P$ since the candidate subgraphs $\mathcal{F}_c$ must contain all applicable subgraphs.

In this situation, the product subgraph selection can be regarded as a multi-label classification problem and starting with any applicable product subgraph in $\mathcal{F}_a$ can obtain a viable retrosynthesis template. However, it is not optimal to train the product subgraph selection model with binary cross-entropy loss (BCE) as in the multi-label classification setting, since it predicts the applicability score independently for each $f \in \mathcal{F}_c$ without considering their interrelationship. Note that the absolute applicability scores of subgraphs in $\mathcal{F}_c$ do not matter here; what really matters is the ranking of these applicability scores, since the beam search is adopted to find a series of template candidates during model inference. While a Softmax classifier can consider the relationship of all subgraphs in $\mathcal{F}_c$, it cannot be directly applied to PSSM, since it is not suitable for the multi-label case. Inspired by Softmax, we propose a novel negative log-likelihood loss for the PSSM:

$$L_{\text{PSSM}} = \log \frac{\arg\min_{f \in \mathcal{F}_a} o_f}{\arg\min_{f \in \mathcal{F}_a} o_f + \sum_{f \in \mathcal{F}_c \setminus \mathcal{F}_a} o_f}, \tag{6}$$

where o_f is the exponential of PSSM output logits for subgraphs in $\mathcal{F}$, $|\mathcal{F}|$ is the size of $\mathcal{F}$, and $\setminus$ is set subtraction. In the above loss function, the numerator is the minimal exponential output for all applicable subgraphs in $\mathcal{F}_a$, which is considered as the ground-truth class proxy in the Softmax classifier. The extra item in denominator is the summation of exponential output of all inapplicable subgraphs in $\mathcal{F}_c$. The intuition is that we always optimize the PSSM to increase the prediction probability for the least probable applicable subgraph, so the model is driven to generate large scores for all applicable subgraphs $\mathcal{F}_c$ while considering interrelationships of candidate subgraphs. The novel loss outperforms BCE loss in our experiments. Detailed experimental comparison results between the proposed loss function Equation (6) and BCE loss can be found in the experiment section.

PSSM scores candidate subgraphs $\mathcal{F}_c$ based on their subgraph embeddings. As shown in Figure 3a, to obtain subgraph embeddings, the nodes of product molecule graph G_p

are first encoded with the modified GAT that is detailed in Section 3.3. The embedding emb_f of the subgraph f is gathered as the average embedding of subgraph $f-$associated nodes in G_p, and then these embeddings are fed into a multilayer perceptron (MLP) for subgraph selection. Here, for a subgraph f, the READOUT function is implemented as the arithmetic average for its simplicity and efficiency. Note that this is different from GLN [18], in which product graphs and candidate subgraphs are considered as separate graphs and embedded independently. Our strategy to reuse node embeddings is more efficient and can learn more informative subgraph embedding since the neighboring structure of a subgraph is also incorporated during the message passing procedure of GAT. Besides, our method can naturally handle multiple equivalent subgraph situations in which the same subgraph appears multiple times within the product graph.

4.1.3. Reactant Subgraph Selection

The second step of the subgraph selection is to choose reactant subgraphs f_r from the vocabulary $\mathcal{F}_R$, which is ordered according to the subgraph frequency in training data, so that f_r is also determinedly ordered. With minor notation abuse, f_r also denotes an ordered sequence of reactant subgraphs in the following content.

Since the number of reactant subgraphs is undetermined, we build the reactant subgraph selection model based on the recurrent neural network (RNN), as illustrated in Figure 3b, and formulate reactant subgraph selection as the sequence generation. The hidden state of RNN is initialized from the product graph embedding emb_{G_p} as defined in Equation (5) to explicitly consider the target product, and the start token is the product subgraph f_p selected in the previous procedure (Section 4.1.2). Furthermore, an extra end token [END] is appended to reactant subgraph sequence f_r. At each time step, the RNN output is fed into a MLP for the token classification. For the start token f_p, we reuse product subgraph embeddings obtained previously (Section 4.1.2) since we find it provides better performance than embedding the token in the traditional one-hot embedding manner.

4.1.4. Annotate Atom Mappings

Given f_p and f_r, the final step is to annotate the atom mappings between f_p and f_r to obtain the retrosynthesis template, as shown in Figure 3c. A subgraph pattern f can also be represented in the SMARTS string, and we use open source toolkit Indigo's (https://github.com/epam/Indigo (accessed on 20 March 2022)) automap() function to build atom mappings. We empirically find about 70% of USPTO-50K training templates can be successfully annotated with correct atom mappings. To remedy this deficiency, we keep a memo of training templates and associated f_p and f_r. During evaluation, the predicted f_p and f_r are processed with automap() if not found in the memo.

4.2. Score Predicted Reactants

After a retrosynthesis template is composed, reactants can be easily obtained by applying the template to the target using RunReactants from RDKit [30] or the run_reaction() function from RDChiral [31]. To achieve superior retrosynthesis prediction performance, it is important to verify that the predicted reactants can generate the target successfully. The verification is achieved by scoring the reactants and target pair, which is formulated as a multi-class classification task where the true reactant set is the ground-truth class.

To serve the verification purpose, we build a reactant scoring model based on the modified GAT. Product molecule graph G_p and reactant molecule graph G_r are first input into a GAT to learn atom embeddings. Since the target and generated reactants are atom-mapped as in Figure 1a, for each atom in G_p, we can easily find its associated atom in G_r. Inspired by WLDN [32], we define a fusion function $\text{F}(n_a^p, n_{a'}^r)$ to combine embeddings of a product atom a and its associated reactant atom a':

$$\text{F}(n_a^p, n_{a'}^r) = \boldsymbol{W}_4(n_a^p - n_{a'}^r)||\boldsymbol{W}_5(n_a^p + n_{a'}^r), \tag{7}$$

where $||$ indicates the concatenation operation and $\boldsymbol{W}$ is a matrix that halves the node embedding dimension so that the concatenated embedding restores the original dimension.

The fused atom embeddings are regarded as new atom features of G_p, which are input into another GAT to learn the graph-level embedding emb_G. In this way, the critical difference between the product and reactant can be better captured since our RSM can incorporate higher order interactions between fused atom embeddings through the message passing process of GAT. Previous retrosynthesis methods score reactants by modeling the compatibility of reactant and product at the molecule level without considering the atom-level embedding.

The graph-level embedding emb_G is then fed into a simple MLP composed of two fully-connected layers to output a compatibility score. The final probability score is obtained by applying a Softmax function to the compatibility scores of all candidate reactants associated to the target.

Our scoring model is advantageous since it operates on atom-level embeddings and is sensitive to local transformations between the product and reactants, while the existing method GLN [18] takes only molecule-level representations as the input. Therefore, GLN cannot capture atom-level transformations and has a weaker distinguishing ability.

The log-likelihoods of our TCM and RSM model predictions are denoted as $l_{TCM} = \log(\mathcal{P}(\mathcal{T}|P))$ and $l_{RSM} = \log(\mathcal{P}(R|P))$, respectively. The predicted reactants are finally ranked according to the linear combination value of $\lambda * l_{TCM} + (1-\lambda) * l_{RSM}$, $0 \leq \lambda \leq 1$. The formulation can be understood as:

$$\begin{aligned} &\lambda * \log(\mathcal{P}(\mathcal{T}|P)) + (1-\lambda) * \log(\mathcal{P}(R|P)) \\ =&\log(\mathcal{P}(\mathcal{T}|P)^{\lambda} * \mathcal{P}(R|P)^{1-\lambda}), \end{aligned} \tag{8}$$

where $\mathcal{P}(\mathcal{T}|P)$ is the probability of that the template $\mathcal{T}$ is applicable to the given product P and $\mathcal{P}(R|P)$ is the probability of the reactant set R for the given product P. When combined together, $\mathcal{P}(\mathcal{T}|P) * \mathcal{P}(R|P)$ approximates the joint probability distribution $\mathcal{P}(\mathcal{T}, R|P)$. Hyper-parameter λ regulates the relative importance of $\mathcal{P}(\mathcal{T}|P)$ and $\mathcal{P}(R|P)$. The optimal λ can be determined by the validation.

5. Experiment and Results

5.1. Dataset and Preprocessing

Our method is evaluated on the standard benchmark dataset USPTO-50K [33] under two settings (with or without reaction types) to demonstrate its effectiveness. USPTO-50K is derived from USPTO granted patents [34] and is composed of 50,000 reactions annotated with 10 reaction types. More detailed dataset information can be found in the Appendix A.1. We split reaction data into training/validation/test sets at an 8:1:1 ratio, in the same way as previous work [15,18]. Since the original annotated mapping numbers in the USPTO dataset may result in unexpected information leakage (https://github.com/uta-smile/RetroXpert (accessed on 20 March 2022)), we first preprocess the USPTO reactions to re-assign product mapping numbers according to the canonical atom order, as suggested by RetroXpert [4]. The atom and bond features are similar to the previous work [4] and reaction types are converted into one-hot vectors concatenated with the original atom features.

Following RetroXpert [4], we extract templates from training reactions using RDChiral [31]. We can obtain 10386 unique templates in total for the USPTO-50K training data and 94.08% of test reactions are covered by these training templates. The gathered templates are split into product and reactant subgraphs, from which mapping numbers are further removed to obtain the subgraph vocabularies $\mathcal{F}_P$ of size 7766 and $\mathcal{F}_R$ of size 4391.

For each target molecule, we find its candidate subgraphs $\mathcal{F}_c$ using graph matching algorithms and applicable templates by checking if the ground-truth reactant can be obtained when each training template is applied to the target. The applicable subgraphs $\mathcal{F}_a$ then can be obtained easily from the acquired applicable templates. Since the exact graph matching process might be time-consuming, we extract the fingerprint for each

molecule/sub-molecule to filter those impossible subgraphs. For the subgraph screening purpose, we adopt the PatternFingerprint from RDKit and use a fingerprint size of 1024.

5.2. Evaluation

Following previous methods [4,18], we use beam search [29] to find Top-50 template predictions during evaluation, which are applied to targets to collect candidate reactants. The collected reactants and targets are the experimental data for RSM. The predicted reactants are finally ranked according to the combined log-likelihood of TCM and RSM. The evaluation metric for retrosynthesis prediction is the Top-K exact match accuracy, which is the percentage of reactions where the ground truth reactant set is within the top K predictions.

5.3. Implementation

Our model is implemented using PyTorch [35] and PyTorch Geometric [36]. The adapted GAT model is built based on the source implementation of Pretrain-GNN [37]. The TCM model is composed of a modified GAT and a simple RNN model. The embedding dimension is set as 300 for all embeddings for simplicity. The number of GAT layers is six. We adopt GRU [38] as the RNN implementation in TCM; the number of GRU layers is two and both its embedding and hidden size are 300. We add a self-loop to each graph node following [4,18]. We use the parametric rectified linear unit (PReLU) [39] comprehensively as the activation function in our model. We replace the original batch normalization [40] layer with a layer normalization [41] layer after each GAT layer, since we find layer normalization provides better performance in our experiments. We adopt Equation (5) as the graph READOUT operation. A simple MLP is applied to product subgraph embeddings to select the proper product subgraph. The MLP is composed of two linear layers, and the PReLU activation function is placed between the two linear layers. We also use a dropout [42] layer with a dropout rate of 0.3 in the MLP.

The RSM model is composed of two GATs and a MLP head, and the GAT uses the same settings as in the TCM except that each GAT is composed of three layers. Product and reactant graphs are embedded with the first GAT model. Note that for reactions with multiple reactants, we regard the disconnected molecule graphs as a single large graph. Once the fused atom embeddings are obtained, the new product molecule graphs with fused atom embeddings are input into the second GAT. The composition of the MLP head is similar to that in TCM. The RSM model is also trained in multi-process mode for acceleration.

Both TCM and RSM are optimized with the Adam [43] optimizer with default settings, and the initial learning rates are 0.0003 and 0.00005 for TCM and RSM, respectively. The learning rate is adjusted with the CosineAnnealingLR scheduler during training. The models are trained in multi-process mode on a single GTX 1080 Ti GPU for acceleration. TCM is trained with batch size 32; it only takes about two hours to train TCM for 80 epochs. RSM training takes about 6 hours for 20 epochs. The final model parameters are saved and loaded later for inference. We repeat all experiments three times and report the mean performance as default. We find our model is quite robust to the hyper-parameters, and most of the model settings are copied from [37] as they are given. We slightly tune the model hyper-parameters, such as learning rate and batch size, manually on validation data to achieve the best results.

5.4. Main Results

We decide the optimal value of λ according to validation performance. Specifically, we set λ as 0.4 for both experimental settings (with/without reaction types). We use these optimal settings in all experiments unless explicitly stated. Detailed ablation study about λ are included in Section 5.4.3.

5.4.1. Retrosynthesis Prediction Performance

We compare our RetroComposer with existing methods on the standard benchmark dataset USPTO-50K, and report comparison results in Table 1. The results of RetroXpert have been updated by the authors (https://github.com/uta-smile/RetroXpert (accessed on 20 March 2022)). For both evaluation settings (with or without reaction types), our method outperforms previous methods by a significant margin in seven out of eight compared Top-K metrics.

Table 1. Retrosynthesis evaluation results (%) on USPTO-50K. Existing methods are grouped into two categories. Our method RetroComposer belongs to the template-based methods. The best results in each column are highlighted in bold. RetroXpert* results have been updated by the authors in their GitHub repository (https://github.com/uta-smile/RetroXpert (accessed on 20 March 2022)).

Methods	Without Reaction Types				With Reaction Types			
	Top-1	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**
Template-free methods								
SCROP [6]	43.7	60.0	65.2	68.7	59.0	74.8	78.1	81.1
G2Gs [7]	48.9	67.6	72.5	75.5	61.0	81.3	86.0	88.7
MEGAN [8]	48.1	70.7	78.4	86.1	60.7	82.0	87.5	**91.6**
RetroXpert* [4]	50.4	61.1	62.3	63.4	62.1	75.8	78.5	80.9
RetroPrime [12]	51.4	70.8	74.0	76.1	64.8	81.6	85.0	86.9
AT [11]	53.5	-	81.0	85.7	-	-	-	-
GraphRetro [13]	53.7	68.3	72.2	75.5	63.9	81.5	85.2	88.1
Dual model [9]	53.6	70.7	74.6	77.0	65.7	81.9	84.7	85.9
Template-based methods								
RetroSim [15]	37.3	54.7	63.3	74.1	52.9	73.8	81.2	88.1
NeuralSym [16]	44.4	65.3	72.4	78.9	55.3	76.0	81.4	85.1
GLN [18]	52.5	69.0	75.6	83.7	64.2	79.1	85.2	90.0
Ours	**54.5**	**77.2**	**83.2**	**87.7**	**65.9**	**85.8**	**89.5**	91.5
TCM only	49.6	71.7	80.8	86.4	60.9	82.3	87.5	90.9
RSM only	51.8	75.7	82.4	87.3	64.3	84.8	88.9	91.4

Specially, our RetroComposer achieves 54.5% Top-1 accuracy without reaction types, which improves on the previous best template-based method GLN [18] significantly by 2.0% and also outperforms existing SOTA template-free methods Dual model and GraphRetro. Besides, our method achieves 77.2% Top-3 accuracy, which improves on the Top-3 accuracy 70.8% of RetroPrime [12] by 6.4%, and 87.7% Top-10 accuracy, which improves on the Top-10 accuracy 85.7% of AT [11] by 2.0%.

When reaction types are given, our method also obtains the best Top-1 accuracy, 65.9%, among all methods and outperforms GLN by 1.7%. Compared with template-free methods GraphRetro and Dual model, our method outperforms the SOTA Dual model (65.7%) by 0.2% and GraphRetro significantly by 2.0% in Top-1 accuracy. As for the Top-10 accuracy, our method achieves 91.5%, which is slightly lower than 91.6% of MEGAN [8].

As the ablation study, we report results with only TCM or RSM. With only either TCM or RSM, the model performance is largely degraded. Without reaction types, TCM only achieves 49.6% Top-1 accuracy while RSM achieves only 51.8%. With reaction types, TCM only achieves 60.9% Top-1 accuracy while RSM achieves only 64.3%. Since TCM and RSM score retrosynthesis from different perspectives and are complementary, their results can be combined to achieve the best performance. Particularly, our method achieves 54.5% and 65.9% Top-1 accuracy when combining TCM and RSM according to Equation (8).

The superior performance demonstrates the effectiveness of our method. Particularly, the superiority of our method is more significant in real world applications where reaction types are unknown. What is more, our Top-10 accuracy is already quite high. This indicates that our method can usually find the best reactant set for the target in a few candidates.

This is especially important for multi-step retrosynthesis scenarios, in which the number of predicted reaction paths may grow exponentially with the retrosynthesis path length.

5.4.2. Ablation Study of PSSM Loss

We experimentally show that our proposed loss function Equation (6) for PSSM outperforms the BCE loss. For all ablation experiments, we find the optimal value of hyper-parameter λ independently and report the best results for a fair comparison. The comprehensive experimental results are reported in Table 2.

Without given reaction types, our method with Equation (6) as PSSM loss achieves the best Top-1 and Top-3 accuracy results, outperforming the BCE loss in Top-1 and Top-3 accuracy by 1.4% and 1.5%, respectively. With known reaction types, our method with Equation (6) as PSSM loss outperforms BCE loss by 0.6% in Top-1 accuracy. While BCE loss can achieve better Top-5 and Top-10 results in both settings, our proposed loss function Equation (6) can achieve better Top-1 accuracy. The retrosynthesis prediction emphasizes more Top-1 accuracy, therefore, we adopt Equation (6) as the PSSM loss in our method.

For all experiments, combining the TCM and RSM scores can always achieve the best performance, which proves the effectiveness of our strategy.

Table 2. Ablation study results (%) of two different PSSM loss functions: our proposed Equation (6) and BCE. The bold indicates the best results.

Types	L_{PSSM}	Methods	Top-1	Top-3	Top-5	Top-10
Without	Equation (6)	Ours	**54.5**	**77.2**	83.2	87.7
		TCM only	49.6	71.7	80.8	86.4
		RSM only	51.8	75.7	82.4	87.3
	BCE	Ours	53.1	77.1	**83.8**	**89.2**
		TCM only	46.5	69.9	78.5	86.9
		RSM only	51.2	75.7	82.9	88.6
With	Equation (6)	Ours	**65.9**	85.8	89.5	91.5
		TCM only	60.9	82.3	87.5	90.9
		RSM only	64.3	84.8	88.9	91.4
	BCE	Ours	65.3	**85.9**	**90.3**	**92.6**
		TCM only	58.5	81.8	87.6	91.5
		RSM only	64.2	85.4	89.6	92.4

5.4.3. Ablation Study of Hyper-Parameter λ

We conduct the ablation study of λ and report results in Table 3; when $\lambda = 0.4$, the best Top-1 accuracy is achieved for both settings. Note that with only RSM ($\lambda = 0$), the Top-1 accuracy 64.3% already outperforms the previous best template-based method GLN of 63.2% [18] with given reaction types. This demonstrates the effectiveness of our RSM. With only TCM ($\lambda = 1.0$), the performance has an appreciable gap with the existing methods. In our method, each generated set of subgraphs may have multiple associated templates due to the uncertainty of product subgraphs and atom transformations. Therefore, there may be multiple top-tier predictions that cannot be distinguished with only TCM. With a little help from RSM ($\lambda = 0.9$), these top-tier predictions can be differentiated and the Top-1 accuracy significantly boosted.

The l_{RSM} indicates the likelihood of retrosynthesis templates, while l_{TCM} scores each reaction by looking at the detailed atom transformations. These two terms are complementary and combined together to achieve the best performance.

Table 3. Top-1 accuracy (%) with different λ values. The bold indicates the best results.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Without types	51.8	53.3	53.9	**54.5**	**54.5**	54.4	54.1	53.6	53.0	52.3	49.6
With types	64.3	65.2	65.6	65.7	**65.9**	**65.9**	65.6	65.1	64.7	64.4	60.9

5.4.4. Novel Templates

Different from existing methods, our method can find novels templates that are not in training data. Our model predicts different templates based on different possible reaction centers for a given target. For example, an amide formation template and alkylation template may both be applied in the same target molecule, and our model can predict suitable templates very well and give reasonable corresponding reactants for such cases. For the 5.92% of test reactions that are not covered by training templates, our algorithm can predict relevant templates very well for most reaction types, although it fails in some heterocyclic formation reactions. This is because there are very few reaction data on such reactions in USPTO-50K. Particularly, our method successfully discovers chemically valid templates for 15 uncovered test reactions, which confirms that our method can find novel reactions. Two such examples are illustrated in Figure 4.

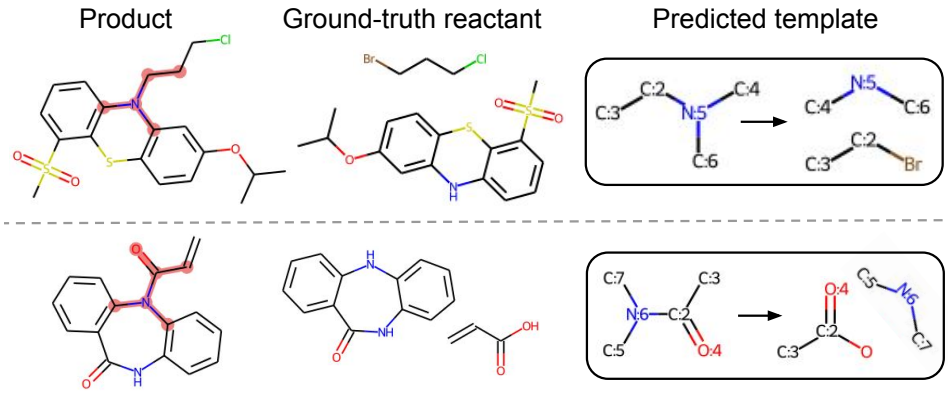

Figure 4. Our method successfully finds valid templates for two test reactions that are not covered by training data. The matched product subgraphs are highlighted in pink for better visualization.

6. Discussion and Conclusions

In this work, we propose a novel template-based retrosynthesis prediction framework that composes templates by selecting and assembling molecule subgraphs. Besides, experimental results confirm that the proposed strategy can discover novel reactions. Although currently our method can find only a few novel templates, we believe our method can inspire the community to explore further in this direction to improve models' ability to find more novel reactions. To further improve the ranking accuracy, we present a novel reactant scoring model to rank candidate reactants by taking into account atom-level transformations. Our method significantly outperforms previous methods and sets new SOTA performance on the USPTO-50K, which proves the effectiveness of our method.

We tried to adapt our method to run on the USPTO-full dataset [34], but find it needs non-trivial effort to manually handle edge cases due to noisy reactions (such as wrong mapping numbers) from USPTO-full, since our methods rely on correct mapping numbers to extract templates as well as build the reactant scoring model. We have released our source implementation and encourage the community to help adapt our method to the USPTO-full dataset.

Author Contributions: Conceptualization, C.Y. and P.Z.; methodology, C.Y. and P.Z.; formal analysis, Y.Y.; investigation, C.L.; writing—original draft preparation, C.Y. and C.L.; writing—review and editing, C.Y. and P.Z.; supervision, J.H.; project administration, J.H.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by US National Science Foundation IIS-1553687 and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental dataset USPTO-50K can be downloaded at http://pubs.acs.org/doi/suppl/10.1021/acs.jcim.6b00564/suppl_file/ci6b00564_si_002.zip (accessed on 20 March 2022).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

Appendix A.1. USPTO-50K Dataset Information

The USPTO-50K consists of 50,000 reactions that are annotated with 10 reaction types; the detailed distribution of reaction types is displayed in the below Table A1. The imbalanced reaction type distribution makes the retrosynthesis prediction more challenging.

Table A1. Distribution of 10 recognized reaction types.

Type	Reaction Type Name	Number of Reactions
1	Heteroatom alkylation and arylation	15,204
2	Acylation and related processes	11,972
3	C-C bond formation	5667
4	Heterocycle formation	909
5	Protections	672
6	Deprotections	8405
7	Reductions	4642
8	Oxidations	822
9	Functional group interconversion	1858
10	Functional group addition (FGA)	231

We can extract 10,386 unique templates from the training data, and 94.08% of test reactions are covered by these templates. For each product molecule, there are an average of 35.19 candidate subgraphs, which are denoted as $\mathcal{F}_c$ in Section 4.1.2. Among these subgraphs, there are an average of 2.02 applicable subgraphs denoted as $\mathcal{F}_a$ for each target.

Table A2. Statistical results of templates and reactions. # is the short for "number".

# total templates	10,386
# unique product subgraphs	7766
# unique reactant subgraphs	4391
Test reactions coverage by training templates	94.08%
Average # contained product subgraphs per mol	35.19
Average # applicable product subgraphs per mol	2.02
Average # templates per reaction	2.23
Average # reactants per reaction	1.71

Appendix A.2. Atom and Bond Features

Following [4], we use similar bond and atom features to build molecule graphs as listed in Tables A3 and A4. These features can be easily extracted using the chemistry toolkit RDKit.

Table A3. Bond features used in our method. These features are one-hot encoding.

Feature	Description	Size
Bond type	Single, double, triple, or aromatic.	4
Conjugation	Whether the bond is conjugated.	1
In ring	Whether the bond is part of a ring.	1
Stereo	None, any, E/Z or cis/trans.	6

Table A4. Atom features used in our method. All features are one-hot encoding, except the atomic mass is a real number scaled to be on the same order of magnitude. The reaction type is applicable for type conditional setting.

Feature	Description	Size
Atom type	Type of atom (ex. C, N, O), by atomic number.	17
# Bonds	Number of bonds the atom is involved in.	6
Formal charge	Integer electronic charge assigned to atom.	5
Chirality	Unspecified, tetrahedral CW/CCW, or other.	4
# Hs	Number of bonded Hydrogen atom.	5
Hybridization	sp, sp2, sp3, sp3d, or sp3d2.	5
Aromaticity	Whether this atom is part of an aromatic system.	1
Atomic mass	Mass of the atom, divided by 100.	1
Reaction type	The specified reaction type if it exists.	10

References

1. Corey, E.J.; Wipke, W.T. Computer-assisted design of complex organic syntheses. *Science* **1969**, *166*, 178–192. [CrossRef] [PubMed]
2. Corey, E.J. The logic of chemical synthesis: Multistep synthesis of complex carbogenic molecules (Nobel Lecture). *Angew. Chem. Int. Ed. Engl.* **1991**, *30*, 455–465. [CrossRef]
3. Gothard, C.M.; Soh, S.; Gothard, N.A.; Kowalczyk, B.; Wei, Y.; Baytekin, B.; Grzybowski, B.A. Rewiring chemistry: Algorithmic discovery and experimental validation of one-pot reactions in the network of organic chemistry. *Angew. Chem. Int. Ed.* **2012**, *51*, 7922–7927. [CrossRef] [PubMed]
4. Yan, C.; Ding, Q.; Zhao, P.; Zheng, S.; Yang, J.; Yu, Y.; Huang, J. RetroXpert: Decompose Retrosynthesis Prediction Like A Chemist. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11248–11258.
5. Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Cent. Sci.* **2017**, *3*, 1103–1113. [CrossRef]
6. Zheng, S.; Rao, J.; Zhang, Z.; Xu, J.; Yang, Y. Predicting Retrosynthetic Reactions using Self-Corrected Transformer Neural Networks. *J. Chem. Inf. Model.* **2020**, *60*, 47–55. [CrossRef]
7. Shi, C.; Xu, M.; Guo, H.; Zhang, M.; Tang, J. A Graph to Graphs Framework for Retrosynthesis Prediction. *arXiv* **2020**, arXiv:2003.12725.
8. Sacha, M.; Błaż, M.; Byrski, P.; Włodarczyk-Pruszyński, P.; Jastrzebski, S. Molecule Edit Graph Attention Network: Modeling Chemical Reactions as Sequences of Graph Edits. *J. Chem. Inf. Model.* **2021**, *61*, 3273–3284. [CrossRef]
9. Sun, R.; Dai, H.; Li, L.; Kearnes, S.; Dai, B. Towards understanding retrosynthesis by energy-based models. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 10186–10194.
10. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36. [CrossRef]
11. Tetko, I.V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **2020**, *11*, 5575. [CrossRef] [PubMed]
12. Wang, X.; Li, Y.; Qiu, J.; Chen, G.; Liu, H.; Liao, B.; Hsieh, C.; Yao, X. RetroPrime: A Diverse, plausible and Transformer-based method for Single-Step retrosynthesis predictions. *Chem. Eng. J.* **2021**, *420*, 129845. [CrossRef]
13. Somnath, V.R.; Bunne, C.; Coley, C.; Krause, A.; Barzilay, R. Learning graph models for retrosynthesis prediction. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9405–9415.
14. Szymkuć, S.; Gajewska, E.P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B.A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem. Int. Ed.* **2016**, *55*, 5904–5937. [CrossRef]
15. Coley, C.W.; Rogers, L.; Green, W.H.; Jensen, K.F. Computer-assisted retrosynthesis based on molecular similarity. *ACS Cent. Sci.* **2017**, *3*, 1237–1245. [CrossRef]
16. Segler, M.H.; Waller, M.P. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chem.-Eur. J.* **2017**, *23*, 5966–5971. [CrossRef]
17. Segler, M.H.; Preuss, M.; Waller, M.P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604–610. [CrossRef]

18. Dai, H.; Li, C.; Coley, C.; Dai, B.; Song, L. Retrosynthesis Prediction with Conditional Graph Logic Network. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8870–8880.
19. Segler, M.H.; Waller, M.P. Modelling chemical reasoning to predict and invent reactions. *Chem.-Eur. J.* **2017**, *23*, 6118–6128. [CrossRef]
20. Baylon, J.L.; Cilfone, N.A.; Gulcher, J.R.; Chittenden, T.W. Enhancing retrosynthetic reaction prediction with deep learning using multiscale reaction classification. *J. Chem. Inf. Model.* **2019**, *59*, 673–688. [CrossRef]
21. Tu, Z.; Coley, C.W. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction. *arXiv* **2021**, arXiv:2110.09681.
22. Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E.J. Chemformer: A Pre-Trained Transformer for Computational Chemistry. *Mach. Learn. Sci. Technol.* **2021**, *3*, 015022. [CrossRef]
23. Mao, K.; Xiao, X.; Xu, T.; Rong, Y.; Huang, J.; Zhao, P. Molecular graph enhanced transformer for retrosynthesis prediction. *Neurocomputing* **2021**, *457*, 193–202. [CrossRef]
24. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
25. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.i.; Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5453–5462.
28. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv* **2015**, arXiv:1511.05493.
29. Tillmann, C.; Ney, H. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.* **2003**, *29*, 97–133. [CrossRef]
30. Landrum, G. RDKit: Open-Source Cheminformatics. 2021. Available online: https://github.com/rdkit/rdkit/tree/Release_2021_03_1 (accessed on 14 September 2022).
31. Coley, C.W.; Green, W.H.; Jensen, K.F. RDChiral: An RDKit wrapper for handling stereochemistry in retrosynthetic template extraction and application. *J. Chem. Inf. Model.* **2019**, *59*, 2529–2537. [CrossRef]
32. Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting organic reaction outcomes with weisfeiler-lehman network. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
33. Schneider, N.; Stiefl, N.; Landrum, G.A. What's what: The (nearly) definitive guide to reaction role assignment. *J. Chem. Inf. Model.* **2016**, *56*, 2336–2346. [CrossRef]
34. Lowe, D.M. Extraction of Chemical Structures and Reactions from the Literature. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2012.
35. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
36. Fey, M.; Lenssen, J.E. Fast graph representation learning with PyTorch Geometric. *arXiv* **2019**, arXiv:1903.02428.
37. Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for pre-training graph neural networks. *arXiv* **2019**, arXiv:1905.12265.
38. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
40. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 448–456.
41. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
42. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.