Retroformer: Pushing the Limits of End-to-end Retrosynthesis Transformer

Yue Wan 1  Chang-Yu Hsieh 1  Benben Liao 1  Shengyu Zhang 1

Abstract

Retrosynthesis prediction is one of the fundamental challenges in organic synthesis. The task is to predict the reactants given a core product. With the advancement of machine learning, computer-aided synthesis planning has gained increasing interest. Numerous methods were proposed to solve this problem with different levels of dependency on additional chemical knowledge. In this paper, we propose Retroformer, a novel Transformer-based architecture for retrosynthesis prediction without relying on any cheminformatics tools for molecule editing. Via the proposed local attention head, the model can jointly encode the molecular sequence and graph, and efficiently exchange information between the local reactive region and the global reaction context. Retroformer reaches the new state-of-the-art accuracy for the end-to-end template-free retrosynthesis, and improves over many strong baselines on better molecule and reaction validity. In addition, its generative procedure is highly interpretable and controllable. Overall, Retroformer pushes the limits of the reaction reasoning ability of deep generative models.

1. Introduction

Retrosynthesis (Corey & Cheng, 1989) is one of the major building blocks in organic synthesis, which aims to discover valid and efficient synthetic routes (i.e., reactants) given a target molecule (i.e., product). It is crucial for the pharmaceutical industry as one of the main challenges for drug discovery is to efficiently synthesize novel and complex compounds in the laboratory (Blakemore et al., 2018).

Recently, computer-aided synthesis planning has gained vast attention for its potential to save a tremendous amount of time and efforts from traditional retrosynthesis approaches. Various machine learning approaches were proposed with different levels of dependency on additional chemical knowledge. These methods can be categorized into three groups. First, template-based methods (Coley et al., 2017; Dai et al., 2019; Chen & Jung, 2021) view the retrosynthesis prediction as the template retrieval problem, where a template encodes the core reactive rule (Figure 1). After the templates are retrieved, these methods use cheminformatics tools like RDKit (rdk) to build up full reactions from the templates. Despite the state-of-the-art accuracy and guaranteed molecule validity, these methods are limited to the scope of the existing template database. In contrast, template-free methods, the second class, use deep generative models to directly generate the reactants given the product. Since molecule can be represented by both the graph and the SMILES sequence, existing approaches reframe the retrosynthesis into either sequence-to-sequence (Lin et al., 2020; Chen et al., 2019; Zheng et al., 2020; Tetko et al., 2020; Seo et al., 2021; Kim et al., 2021) or graph-to-sequence problem (Tu & Coley, 2021). These generative methods do not rely on any additional chemical knowledge and can perform chemical reasoning within a larger reaction space. The third class is semi-template-based methods, which combine the advantages of both the generative models and the prior chemical knowledge. Conventional frameworks (Yan et al., 2020; Shi et al., 2020; Somnath et al., 2020; Wang et al., 2021) in this category follow the same idea: They first identify the reactive sites and convert the product into synthons using RDKit. Then, another model completes synthons into reactants. These methods are competitive in accuracy and are interpretable by their stage-wise nature.

In this work, we are interested in the template-free generative approach for retrosynthesis prediction. Existing methods fail to fully explore the potential of deep generative model in terms of reaction reasoning, and we argue that the end-to-end Transformer-based (Vaswani et al., 2017) architecture can reach the same competitive benchmark accuracy as well as good validity and interpretability. We propose Retroformer, a novel end-to-end retrosynthesis Transformer that introduces a special attention head. It is able to jointly encode the sequential and graphical information of the molecule and allow efficient information exchange between the local reactive region and the global reaction context. The generative process is also sensitive to the exact
2. Related Work

2.1. Retrosynthesis Prediction

Existing methods in retrosynthesis prediction can be grouped into three categories: template-based, template-free, and semi-template-based. The reaction template encodes the core reactive rules. As shown in Figure 1, a conventional template tells the potential reactive region within the molecule, as well as its potential chemical transformation. These templates are either expert-defined or automatically extracted by algorithms. In this work, we strictly differentiate the three categories by the levels of dependency on additional chemical knowledge during inference.

**Template-based** methods rely on an external template database. Since the template is a more efficient and interpretable representation for reactions (Heid et al., 2021; Wan et al., 2021), a large body of works (Coley et al., 2017; Dai et al., 2019; Chen & Jung, 2021) focus on capturing the reactive scores between the molecules and templates. Retrosim (Coley et al., 2017) uses molecule fingerprint similarity to rank the candidate templates. GLN (Dai et al., 2019) and LocalRetro (Chen & Jung, 2021) use graph neural network (GNN) to capture the molecule-template and atom/bond-template relationship, respectively. Despite their state-of-the-art top-$k$ accuracy, all template-based methods suffer from the incomplete coverage issue and do not scale well.

**Template-free** methods, in contrast, adopt deep generative models to directly generate the reactants molecules. Besides graph, molecules can be represented using SMILES sequence. Existing works (Lin et al., 2020; Chen et al., 2019; Zheng et al., 2020; Tetko et al., 2020; Seo et al., 2021; Kim et al., 2021) take advantage of the Transformer (Vaswani et al., 2017) architecture and reframe the problem as the sequence-to-sequence translation from product to reactants. Graph2SMILES (Tu & Coley, 2021) replaces the original sequence encoder with a graph encoder to ensure the permutation invariance of SMILES. These methods rely on little additional chemical knowledge for inference. However, chemical validity can be a huge concern because validity is often not part of the training objective. Another factor comes from the ignorance of graphical structure during the sequence generation. Also, generated outcomes from beam search often suffer from the diversity issue (Vijayakumar et al., 2018), which is another practical concern for retrosynthesis.

**Semi-template-based** methods combine the advantage of both the generative models and additional chemical knowledge. In this work, we strictly categorize generative models which require additional help from RDKit (rdk) for molecule editing into this method group. MEGAN (Sacha et al., 2021) reframes the generative procedure as a sequence of graph edits that are completed by RDKit. In addition, most existing works (Yan et al., 2020; Shi et al., 2020; Somnath et al., 2020; Wang et al., 2021) approach the task by a two-stage procedure. Despite their architecture differences between GNN and Transformer, they follow the same idea: They first convert the product into synthons by predicting its reactive sites and performing molecule editing via RDKit, then complete the synthons into reactants by either leaving groups selection (Somnath et al., 2020), graph generation (Shi et al., 2020), or SMILES generation (Yan et al., 2020; Wang et al., 2021). Although the framework fits better with chemists’ intuition of solving the problem, it brings several disadvantages. First, it requires two separate models to perform each subtask. RetroPrime (Wang et al., 2021) uses two
Transformers, which doubles the computational complexity. Second, the two learning stages are independent. It means that knowledge behind reaction site prediction cannot propagate to reactants completion, which loses some chemical senses. Third, any subtle mistake from reaction site prediction will directly mislead the results. In contrast, our work attempts to walk around these disadvantages with an end-to-end Transformer while still maintaining its interpretability.

Besides different architecture designs, self-supervised molecule pretraining is also shown to be effective in retrosynthesis prediction. DMP-fusion (Zhu et al., 2021) pretrains the molecule with a dual view of SMILES and graph. Chemformer (Irwin et al., 2021) applies masked SMILES modeling to learn the molecule representation.

2.2. Graph Transformer

The introduction of Transformer into the graph domain has gained increasing interest. The global receptive fields of the self-attention and the local message passing of the graph neural network are inherently complementary and compatible. Attempts have been made in ways of incorporating graph information into the self-attention computation (Ying et al., 2021; Łukasz Maziarka et al., 2020; 2021) and integrating the conventional graph neural networks (Gilmer et al., 2017) with Transformer architecture (Dwivedi & Bresson, 2021; Chen et al., 2021).

3. Preliminary

Let $S = [s_1, s_2, \ldots, s_n]$ be the molecular SMILES sequence with $n$ number of tokens. Let $G_{mol} = (V_m, E_m)$ be the molecular graph. It is formed by $|V_m|$ number of atoms with $|E_m|$ number of bonds. For computation convenience, we further introduce the SMILES graph $G_{smi} = (V_s, E_s)$. $V_s$ is made up of all the SMILES tokens, including the atom tokens (e.g., “C”, “O”) as well as the other special tokens (e.g., “=”, “1”): $V_m \subseteq V_s = S$. In $G_{smi}$, the special tokens are treated as trivial nodes with no neighbors. Its edge $E_s$ represents the graphical connections between atom tokens, which is essentially the same as bond connections $E_m$ in $G_{mol}$. In general, $G_{smi}$ is a larger but sparser graph compared to $G_{mol}$. The introduction of $G_{smi}$ is merely to ensure the alignment relationship between the atoms in graph and the tokens in SMILES.

4. Retroformer

We propose Retroformer, a novel Transformer-based model that is able to perform interpretable retrosynthesis prediction in an end-to-end manner. We propose a special type of local attention head that can support efficient information exchange between the local region of reactive importance and the global reaction context. Its generative procedure is also sensitive to the exact local region. The overall training and inference can be done in an end-to-end manner. It is a fully template-free method without any additional dependency on RDKit for molecule editing. The overall architecture contains an encoder, a decoder, and two reaction center identifiers. We also propose to use SMILES alignment and on-the-fly data augmentation as two additional training strategies.
4.1. Local-Global Encoder with Edge Update

Since molecular graph can provide additional information on top of the SMILES sequence, our encoder takes both $S$ sequence and $G_{smi}$ (i.e., adjacency matrix and bond feature) as inputs. The bond features we considered are listed in Appendix E.1. Different from the existing graph Transformers (Ying et al., 2017; Łukasz Maziarka et al., 2020; 2021) that compute graph self-attention within the entire module, our model encodes the graph information at the head level. We specify two types of attention heads: global head and local head. The global head is the same as the vanilla self-attention head, where its receptive field is the entire SMILES sequence. The local head, on the other hand, considers the topological structure of the molecule. The receptive field of an individual token is restricted to its one-hop neighborhood, which is similar to (Zhang et al., 2021). In addition, we perform element-wise multiplication between the key vector and the edge feature to incorporate the bond information into the calculation. The roll-out form of the local head self-attention at layer $l$ for the $i^{th}$ token is formulated as:

$$x_{i}^{l+1}_{local} = \sum_{j \in N(i)} \sigma\left(\frac{q_i(A_{ij}^{l} \odot k_j)^T}{\sqrt{d}}\right)v_j$$

(1)

$$[q_i, k_j, v_j] = [h_i^{l}W^Q, h_j^{l}W^K, h_j^{l}W^V]$$

where $A$ is the bond feature matrix, $W^Q, W^K, W^V$ are the projection matrix for calculating query $q$, key $k$, and value $v$, and $\sigma$ is the softmax operation. The computed representations from the global and local heads are then concatenated along the hidden dimension and passed to a linear layer, which represents the updated token features $h_{i}^{l+1}$. Meanwhile, the edge update module is a fully connected layer (FFN) that takes the concatenation of the updated features of the receiving and sending tokens as inputs:

$$h_i^{l+1} = \text{Linear}([x_{i}^{l+1}_{global}, x_{i}^{l+1}_{local}])$$

(2)

$$A_{ij}^{l+1} = A_{ij}^{l} + \text{FFN}([h_i^{l+1}, h_j^{l+1}])$$

(3)

The integration of the local, global attention heads, and the edge update module allows the model to efficiently exchange information between the local region and global molecular context. Same as the vanilla Transformer (Vaswani et al., 2017), layer normalization and residual connection are enforced between encoder layers. The final encoder outputs are the updated token representation $h$ and the bond representation $A$.

4.2. Reaction Center Detection

A reaction center represents the group of atoms and bonds that are contributing factors to the chemical transformation. However, existing semi-template-based methods (Yan et al., 2020; Somnath et al., 2020; Shi et al., 2020; Wang et al., 2021) simplify this concept as the exact reactive sites. We argue that this simplification may lead to information loss of the reaction context, such as the influence of functional groups. These methods also cannot perform retrosynthesis in an end-to-end manner, since they rely on RDKit to convert the product into synthons. Instead, Retroformer predicts the reactive probability $P_{rc}(\cdot)$ of each atom and bond and converts the reactive region of $S$ into the attention receptive field for the decoder. In short, the detected reaction center $S_{rc}$ is a subset of $S$.

Figure 2b shows a heatmap visualization of the predicted reactive probability. It is done by two fully connected layers named Atom RC Identifier and Bond RC Identifier:

$$P_{rc}(s_i) = \sigma(\text{FFN}_{\text{atom}}(h_i)), s_i \in V_m$$

(4)

$$P_{rc}(e_{ij}) = \sigma(\text{FFN}_{\text{bond}}(A_{ij})), e_{ij} \in E_m$$

(5)

We will show in Section 5.3 that the learned reaction center can be easily visualized and matched with chemical heuristics. We then convert the atom and the bond reactive probability into the reactive indicator of tokens in $S_{rc}$ by either one of the following two strategies:

- **naive**: we naively set a token as reactive if it exists in a reactive edge (i.e., $P_{rc}(e) > 0.5$) and is reactive itself (i.e., $P_{rc}(s) > 0.5$). Note that the special tokens are guaranteed to be non-reactive. This strategy is used at both training and inference stages.

- **search**: we conduct a subgraph search on the molecular graph and rank the subgraphs by their reaction center score: $\sum_{s_i \in S_{rc}} \log P_{rc}(s_i) + \sum_{s_i, s_j \in S_{rc}} \log P_{rc}(e_{ij})$. Only atoms with $P_{rc}(s) > \alpha_{\text{atom}}$ and bonds with $P_{rc}(e) > \alpha_{\text{bond}}$ are considered in the search to reduce the computational time. Detailed algorithm is described in Appendix E.4. Then, top-$n$ subgraphs are selected as reaction center candidates. The model then generates $k/n$ reactants for each reaction center, where $k$ is the total number of predicted reactants. The final results are ranked by the sum of the reaction center score and the generative score. This strategy is only used at inference stage.

4.3. Local-Global Decoder

The decoder takes its generative outcomes from the previous step, the encoder outputs $h$, and the reaction center $S_{rc}$ as inputs. Similar to the encoder, we also introduce two different heads in its cross-attention module. The global head is the same as the vanilla head. The local head, on the contrary, is only visible to the detected reaction center $S_{rc}$. It computes the sparse cross-attention instead of the
Figure 4: Token and atom alignment between product and reactants in SMILES and graph representations, respectively. The right most figure is the ground truth alignment matrix.

4.4. SMILES Alignment

SMILES alignment is an additional learning task of Retroformer. Similar to machine translation, the SMILES sequences of the source and the target molecules are often partially aligned. A large portion of the molecules remains unchanged during the reaction. Figure 4 shows this alignment relationship in both graph and SMILES representations. The node alignment between graphs (i.e., atom mapping) can be easily converted into token alignment between SMILES. Detailed substring matching algorithm with atom-mapping is described in Appendix E.3.

Inspired by the effectiveness of the guided attention in (Deshpande & Narasimhan, 2020), we introduce the attention guidance loss between the ground truth alignment and the attention weights from the decoder’s global cross-attention heads. We treat the computed cross-attention at each decoder step as a probability distribution and impose a label smoothing loss (Pereyra et al., 2017). It is a soft cross entropy loss with the label smoothing technique and is shown to be effective in classification performance. Hypothetically speaking, this guided attention can encourage the model to understand chemical reactions more efficiently.

4.5. Data Augmentation

We follow the same data augmentation tricks used in (Seo et al., 2021; Tetko et al., 2020) for the SMILES generative models, which are the SMILES permutation of the product and the order permutation of reactants. However, instead of expanding the training dataset off-the-shelf, we choose to perform the augmentation on-the-fly. At each iteration, there is a probability of 50% to permute the product SMILES and the reactants ordering. This dynamic permutation allows the model to focus more on the canonical SMILES and use the permuted SMILES for regularization. Ablation towards different augmentation methods is discussed in Appendix B.

4.6. Loss

The training schema can be viewed as an end-to-end multi-task learning. The overall loss is made up of four parts: $L = L_{LM} + L_{RC_{valid}} + L_{RC_{atom}} + L_{AG}$, where $L_{LM}$ is the language modeling objective, $L_{RC_{valid}}$ is the reactive probability loss, and $L_{AG}$ is the SMILES alignment loss.

5. Experiments

Data We use the conventional retrosynthesis benchmark dataset USPTO-50K (Schneider et al., 2016) to evaluate our method. It contains 50016 atom-mapped reactions that are grouped into 10 reaction classes. We use the same data split as (Coley et al., 2017). We canonicalize the molecule SMILES with atom mapping following the same protocol given in (Somnath et al., 2020). We then use RDChiral (Coley et al., 2019) to extract the ground truth reaction center and use the algorithm in Appendix E.3 to extract the ground truth SMILES token alignment.

Evaluation We adopt the conventional top-k accuracy of the full reactants to evaluate the retrosynthesis performance. We also evaluate the top-k validity of the generated routes. For molecule validity, we treat a candidate as valid if RDKit (rdk) can successfully identify the molecule SMILES. The top-k validity is calculated as: $Valid(k) = \frac{1}{N \times k} \sum_1^N \sum_1^k 1$(SMILES is valid). We further evaluate our method with the round-trip accuracy (Schwaller et al., 2020), which is an approximation metric for reaction validity. It measures the percentage of predicted reactants that can lead back to the original product. We take the pretrained Molecule Transformer (Schwaller et al., 2019) as the oracle reaction prediction model because of its state-of-the-art performance. Our top-k round-trip accuracy calculation is slightly different from the definition adopted by RetroPrime (Wang et al., 2021) and LocalRetro (Chen & Jung, 2021): $RoundTrip(k) = \frac{1}{N \times k} \sum_1^N \sum_1^k 1$(Reach Ground Truth Product).
### Table 1: Top-$k$ accuracy for retrosynthesis prediction on USPTO-50K. * indicates the model with SMILES augmentation. For comparison purpose, the Aug. Transformer is evaluated without any test augmentation. Best performance is in **bold**.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-$k$ accuracy (%)</th>
<th>Reaction class known</th>
<th>Reaction class unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Template-Based</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLN (Dai et al., 2019)</td>
<td>64.2</td>
<td>79.1</td>
<td>85.2</td>
</tr>
<tr>
<td>LocalRetro (Chen &amp; Jung, 2021)</td>
<td>63.9</td>
<td><strong>86.8</strong></td>
<td><strong>92.4</strong></td>
</tr>
<tr>
<td><strong>Template-Free</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformer</td>
<td>57.1</td>
<td>71.5</td>
<td>75.0</td>
</tr>
<tr>
<td>SCROP (Zheng et al., 2020)</td>
<td>59.0</td>
<td>74.8</td>
<td>78.1</td>
</tr>
<tr>
<td>Tied Transformer (Kim et al., 2021)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Aug. Transformer* (Tetko et al., 2020)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GTA* (Seo et al., 2021)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Graph2SMILES (Tu &amp; Coley, 2021)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Retroformer$_{base}$ (Ours)</td>
<td>61.5</td>
<td>78.3</td>
<td>82.0</td>
</tr>
<tr>
<td>Retroformer$_{aug}$* (Ours)</td>
<td><strong>64.0</strong></td>
<td><strong>81.8</strong></td>
<td><strong>85.4</strong></td>
</tr>
<tr>
<td>Retroformer$_{aug}+$* (Ours)</td>
<td><strong>64.0</strong></td>
<td><strong>82.5</strong></td>
<td><strong>86.7</strong></td>
</tr>
<tr>
<td><strong>Semi-Template-Based</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetroXpert* (Yan et al., 2020)</td>
<td>62.1</td>
<td>75.8</td>
<td>78.5</td>
</tr>
<tr>
<td>G2G (Shi et al., 2020)</td>
<td>61.0</td>
<td>81.3</td>
<td>86.0</td>
</tr>
<tr>
<td>GraphRetro (Somnath et al., 2020)</td>
<td>63.9</td>
<td>81.5</td>
<td>85.2</td>
</tr>
<tr>
<td>RetroPrime* (Wang et al., 2021)</td>
<td><strong>64.8</strong></td>
<td><strong>81.6</strong></td>
<td><strong>85.0</strong></td>
</tr>
<tr>
<td>MEGAN (Sacha et al., 2021)</td>
<td>60.7</td>
<td><strong>82.0</strong></td>
<td><strong>87.5</strong></td>
</tr>
</tbody>
</table>

### Table 2: Top-$k$ SMILES validity for retrosynthesis prediction on USPTO-50K with reaction class unknown.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-$k$ validity (%)</th>
<th>Reaction class unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>97.2</td>
<td>87.9</td>
</tr>
<tr>
<td>Graph2SMILES</td>
<td><strong>99.4</strong></td>
<td>90.9</td>
</tr>
<tr>
<td>RetroPrime</td>
<td>98.9</td>
<td>98.2</td>
</tr>
<tr>
<td>Retroformer$_{aug}$</td>
<td>99.3</td>
<td>98.5</td>
</tr>
<tr>
<td>Retroformer$_{aug}+$</td>
<td>99.2</td>
<td><strong>98.5</strong></td>
</tr>
</tbody>
</table>

### Table 3: Top-$k$ round-trip accuracy for retrosynthesis prediction on USPTO-50K with reaction class unknown.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-$k$ round-trip acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>71.9</td>
</tr>
<tr>
<td>Graph2SMILES</td>
<td>76.7</td>
</tr>
<tr>
<td>RetroPrime</td>
<td><strong>79.6</strong></td>
</tr>
<tr>
<td>Retroformer$_{aug}$</td>
<td>78.6</td>
</tr>
<tr>
<td>Retroformer$_{aug}+$</td>
<td>78.9</td>
</tr>
</tbody>
</table>

**Baseline** We take GLN (Dai et al., 2019) and LocalRetro (Chen & Jung, 2021) as two strong baseline models from the template-based group. We take SCROP (Zheng et al., 2020), Tied Transformer (Kim et al., 2021), Augmented Transformer (Tetko et al., 2020), GTA (Seo et al., 2021), and Graph2SMILES (Tu & Coley, 2021) as the baseline models from the template-free group. We also train a vanilla retrosynthesis Transformer from scratch using OpenNMT (Klein et al., 2017). We take RetroXpert (Yan et al., 2020), G2G (Shi et al., 2020), GraphRetro (Somnath et al., 2020), RetroPrime (Wang et al., 2021), and MEGAN (Sacha et al., 2021) as strong semi-template-based baselines. We do not include the pretraining approach in the performance comparison. We experiment with three variants of the proposed model: Retroformer$_{base}$ represents the model with no data augmentation and the naive reaction center detection strategy; Retroformer$_{aug}$ represents the model with data augmentation and the naive strategy; Retroformer$_{aug}+$ represents the model with data augmentation and the search strategy.

**Implementation Details** Built on top of the vanilla Transformer (Vaswani et al., 2017), our model has 8 encoder layers and 8 decoder layers. The model is trained using the Adam optimizer (Kingma & Ba, 2017) with a fixed learning rate of $1e^{-4}$, and a dropout rate of 0.3. The embedding dimension is set to 256, and the total amount of heads is set to 8. We split the heads by half for global and local heads. Retroformer$_{base}$ is trained on 1 NVIDIA Tesla V100 GPU for 24 hours. Our code is available at https://github.com/yuewan2/Retroformer.
We further demonstrate the strength of the reaction center detection. With the top-n subgraphs proposed, Retroformer\textsubscript{aug} can further boost the performance to the new state-of-the-art accuracy for template-free retrosynthesis in both experiment settings. We provide further ablation and interpretation of the \textit{search} strategy in Section 5.2, Section 5.3 and Appendix E.4.

### Top-k SMILES Validity

We take the vanilla retrosynthesis Transformer, Graph2SMILES, and RetroPrime as strong SMILES generative baselines for validity comparison with our model. We do not include template-based methods in this evaluation since molecule SMILES built from templates are guaranteed to be valid. As we mentioned before, SMILES generative models are more likely to struggle with the validity issue. Without knowing the proper reaction center, the models may modify the molecule fragments that are distant from the core reactive region, which is chemically trivial. As shown in Table 2, both of our model variants enjoy better molecule validity than others. It improves the top-10 validity over the vanilla Transformer by 23.6%. It shows that being aware of the local reactive region can encourage the model to avoid errors that propagate via the non-reactive regions.

### Top-k Round-trip Accuracy

To measure the reaction validity, we take the pretrained Molecule Transformer (Schwaller et al., 2019) as the oracle reaction prediction model to measure the percentage of top-\textit{k} proposed synthetic routes that can lead back to the ground truth product. Table 3 shows the performance comparison of the round-trip accuracy. It shows that our method improves over the existing methods by a large margin. Our model exceeds the vanilla Transformer by 22.0% top-10 round-trip accuracy, and it also improves over RetroPrime by 12.2%. It shows that our model is more likely to propose valid and efficient synthetic routes for downstream usage.

### 5.2 Ablation Study

We further conduct ablation study to evaluate the effects of each component on retrosynthesis performance. As for the guided alignment loss, we experiment with two settings: Guided\textsubscript{all}: the alignment loss is enforced at the first global heads of all decoder layers; Guided\textsubscript{last}: the loss is enforced at the first global head of the last decoder layer.

Table 4 shows that all proposed components are necessary for Retroformer\textsubscript{base} to reach the best retrosynthesis performance. The improvement from (a) to (b) and (a) to (c) shows that the model can better capture the reaction knowledge from learning the SMILES alignment. We choose (c) over (b) as our final alignment loss because of its comparable performance and its lighter training duty. The 2.9% top-1 improvement from (d) to (c) indicates the effectiveness of

<table>
<thead>
<tr>
<th>Settings</th>
<th>Top-k accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(a)</td>
<td>45.5</td>
</tr>
<tr>
<td>(b)</td>
<td>47.0</td>
</tr>
<tr>
<td>(c)</td>
<td>47.9</td>
</tr>
<tr>
<td>(d)</td>
<td>44.1</td>
</tr>
<tr>
<td>(e)</td>
<td>46.7</td>
</tr>
<tr>
<td>(f)</td>
<td>48.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Settings</th>
<th>Top-n accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Ablation (d)</td>
<td>55.4</td>
</tr>
<tr>
<td>Ablation (e)</td>
<td>63.0</td>
</tr>
<tr>
<td>Retroformer\textsubscript{aug}</td>
<td>67.5</td>
</tr>
</tbody>
</table>
Retroformer: Pushing the Limits of Interpretable End-to-end Retrosynthesis Transformer

5.3. Qualitative Analysis

In addition to its competitive benchmark performance, Retroformer is fully interpretable and controllable by external chemical instruction.

To evaluate the interpretability and the quality of the detected reaction center, we randomly select a product molecule from the test set of USPTO-50K and predict the reactants with the search strategy. We also evaluate the setting where we explicitly specify a reaction center and give it to the decoder. We term this setting as the reaction center retrieval. As shown in Figure 5, the search algorithm proposes three different reaction centers (highlighted in green) given the raw reactive log probabilities. The numbers represent the reactive scores of each subgraph candidate. In this example, the top candidate matches the ground truth reaction center in the data. The third column indicates the top-2 verified predicted reactants given both the reaction center and the encoded molecule. The numbers represent the generative scores of each reactants. It shows that the model can understand the concept of reaction center and propose chemical transformations compatible with it. The outcomes from the retrieved reaction center (highlighted in blue) also demonstrate that the generation is fully controllable by external instruction (i.e., the specification of a particular reaction center). Additional analysis with respect to the learned token alignment is described in Appendix D.
6. Conclusion

We propose Retroformer, a novel Transformer-based architecture that reaches the new state-of-the-art performance for template-free retrosynthesis. With the proposed local attention heads and the incorporation of the graph information, the model is able to identify local reactive regions and generate reactants conditionally on the detected reaction center. Being aware of the reaction center also encourages the model to generate reactants with improved molecule validity, reaction validity, and interpretability. We plan to further research the multi-step template-free retrosynthesis planning problem using Retroformer as the single-step retrosynthesis prediction backbone.

References


Tu, Z. and Coley, C. W. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction, 2021.


A. Appendix: Validity and Round-trip Accuracy

Table 6 shows a detailed comparison between all variants of Retroformer in terms of top-k SMILES validity and round-trip accuracy. It helps to understand the effects of search strategy and data augmentation alone on validity performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-k validity</th>
<th>Top-k round-trip acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 3 5 10</td>
<td>1 3 5 10</td>
</tr>
<tr>
<td>Retroformer_base</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.4 97.0 95.4 90.5</td>
<td>76.4 69.4 64.2 53.6</td>
</tr>
<tr>
<td>Retroformer_base+</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.5 97.4 96.3 94.0</td>
<td>77.4 69.3 64.2 53.9</td>
</tr>
<tr>
<td>Retroformer_aug</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>99.3 98.5 97.2 92.6</td>
<td>78.6 71.8 67.1 57.6</td>
</tr>
<tr>
<td>Retroformer_aug+</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>99.2 98.5 97.4 96.7</td>
<td>78.9 72.0 67.1 57.2</td>
</tr>
</tbody>
</table>

B. Appendix: Data Augmentation

Although SMILES augmentation has become a common trick that improves over the SMILES-based molecular deep learning model, the implementation varies. Here, we perform an ablation study with our augmentation method, 2P2R_s (Seo et al., 2021), and x5M (Tetko et al., 2020). 2P2R_s represents adding an additional permuted SMILES (i.e., change the starting atom of the product and shuffle the reactants ordering) for each original SMILES into the dataset. x5M represents adding four additional permuted SMILES as well as its inverted format. The inverted format is essentially the forward reaction SMILES, with a special token added at the beginning of its string for retrosynthesis training. In our work, we apply random permutations at each training iteration instead of expanding the dataset ahead since we want the canonical form and the permuted form to be equally weighted, while still allowing the model to see various permutations. It also avoids storing a much larger dataset ahead.

Table 7 shows the performance comparison of different augmentation methods applied on Retroformer. All methods show significant performance improvement over the non-augmented setting. 2P2R_s is slightly less effective than x5M and ours, which can be explained by its small augmentation amount. Our method is slightly less effective than x5M, even though the augmentation amount from random permutation is much larger. It implies the reducing marginal improvement brought by brutally increasing the permutation amount, which is consistent with the finding in (Tetko et al., 2020). It also indicates the effectiveness of data augmentation from the forward reaction.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Top-k accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 3 5 10</td>
</tr>
<tr>
<td>None</td>
<td>47.9 62.9 66.6 70.7</td>
</tr>
<tr>
<td>2P2R_s</td>
<td>51.9 67.7 72.3 77.0</td>
</tr>
<tr>
<td>x5M</td>
<td>52.8 68.7 72.9 78.6</td>
</tr>
<tr>
<td>Ours</td>
<td>52.9 68.2 72.5 76.4</td>
</tr>
</tbody>
</table>

C. Appendix: Model Complexity

We measure the model complexity (i.e., total number of parameters and inference time) of Retroformer compares to other Transformer-based retrosynthesis baselines: vanilla Transformer, GTA (Seo et al., 2021), and RetroPrime (Wang et al., 2021). Note that the complexity of Augmented Transformer (Tetko et al., 2020) and Tied Transformer (Kim et al., 2021) should be roughly the same as the vanilla Transformer since they only differ by training strategies. Table 8 shows the complexity comparison measured on a NVIDIA T4 GPU with batch size equaling 8. The second column represents the vanilla Transformer based on our implementation. Its difference in inference time compared to the vanilla Transformer by OpenNMT (Klein et al., 2017) is mainly caused by the implementation difference in dataloading and beam search. It implies that Retroformer can be further sped up by efficient designs of the decoding algorithm.
Table 8: Model complexity comparison. * means the model is built by OpenNMT.

<table>
<thead>
<tr>
<th></th>
<th>vanilla*</th>
<th>vanilla GTA*</th>
<th>RetroPrime*</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>#params</td>
<td>23.2M</td>
<td>23.2M</td>
<td>17.4M</td>
<td>75.4M</td>
</tr>
<tr>
<td>sec/mol</td>
<td>0.17</td>
<td>0.47</td>
<td>0.59</td>
<td>2.18</td>
</tr>
</tbody>
</table>

D. Appendix: Qualitative Analysis, Atom Mapping

Since our model is trained to learn the token alignment between the source and the target SMILES, the predicted attention can be easily converted to atom mapping. Different from the RXNMapper (Schwaller et al., 2021) that uses an additional neighbor attention multiplier to calculate the atom mapping from the attention weights, we directly use the attention weights to do the assignment while ignoring the molecular graphical structure. Note that it does not guarantee either the one-to-one mapping from product atom to reactants atom or the equivalence of the mapped element. Figure 6(a) shows a success case of the inferred atom mapping. Figure 6(b) shows a typical failure case. This assignment mistakenly aligns [O:11] with [N:11]. The mistake is explainable since [HN:8] and [O:11] within the reactants are the exact position where chemical transformation happens. Also, this naive atom mapping fails to assign the one-to-one mapping, which is also reasonable because of the symmetry present in the second reactant.

![Figure 6(a) Success Case](image1)

(a) Success Case

![Figure 6(b) Failure Case](image2)

(b) Failure Case

Figure 6: Sample Atom Mapping.

E. Appendix: Implementation Details

E.1. Appendix: Bond Features

Table 9 shows the bond features considered in the proposed Retroformer, more specifically in the local-global encoder.

Table 9: Bond features.

<table>
<thead>
<tr>
<th>Bond Feature</th>
<th>Possible Values</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond Type</td>
<td>Single, Aromatic, Double, Triple</td>
<td>4</td>
</tr>
<tr>
<td>Aromatic</td>
<td>True, False</td>
<td>1</td>
</tr>
<tr>
<td>Conjugated</td>
<td>True, False</td>
<td>1</td>
</tr>
<tr>
<td>Part of Ring</td>
<td>True, False</td>
<td>1</td>
</tr>
</tbody>
</table>

E.2. Appendix: SMILES Graph Construction

To ensure the alignment between the SMILES token and the atoms in graph, we expand the original molecular graph $G_{mol}$ into the SMILES graph $G_{smi}$ by Algorithm 1. `readSmiles()`, `getAtoms()`, `getNeighbors()`, and `writeSmiles()` are functions supported in RDKit (rdk). The tagging procedure is to inform the connection relationship between SMILES
tokens. Rewriting the tagged SMILES without canonicalization is to ensure that the SMILES syntax does not change after the tagging.

**Algorithm 1** SMILES Graph Construction

**Input:** molecule canonical SMILES $S$

Initialize $V$ as a token list of $S$.

Initialize $E$ as an empty set.

Initialize $A$ as an empty list.

$M = \text{readSmiles}(S)$

for $a_i \in \text{getAtoms}(M)$ do

Assign tag #1 to the SMILES symbol of $a_i$.

for $a_j \in \text{getNeighbors}(M, a_i)$ do

Assign tag #2 to the SMILES symbol of $a_j$.

end for

Get the tagged SMILES $S' = \text{writeSmiles}(M, \text{canonical}=\text{False})$

Retrieve the token connections $e(#1,#2)$ by the tagged $S'$ and add them to $E$.

Retrieve the bond feature of $e(#1,#2)$ and add them to $A$.

end for

**Output:** $V, E, A$

**Algorithm 2** SMILES Token Alignment Computation

**Input:** atom-mapped product SMILES $S_p$ and atom-mapped reactants SMILES $S_r$.

Initialize the token mapping dictionary $r2s$.

for $s_r \in S_r$ do

if $s_r$ is not visited and $s_r$ is an atom token then

Locate the token $s_p$ in $S_p$ with the same atom mapping number as $s_r$: $am(s_p) == am(s_r)$.

while $s_r == s_p$ or $am(s_p) == am(s_r)$ do

Add alignment relationship $\{i : j\}$ into $r2s$.

Increment $i$ and $j$.

end while

end if

end for

for $\{i : j\} \in r2s$ do

Decrement $i$ and $j$.

while $s_r == s_p$ and $s_r$ is not an atom symbol do

Add alignment relationship $\{i : j\}$ into $r2s$.

Decrement $i$ and $j$.

end while

end for

**Output:** $r2s$

**Algorithm 3** shows the detailed search reaction center subgraph search algorithm, and Figure 7 shows a visualization of its procedure. In general, it searches for the candidate subgraphs (i.e., reaction centers) within the molecular graph via recursive pruning. It adopts a set of hyperparameters to avoid searching over the entire subgraph space.
Figure 7: Visualization of the reaction center subgraph search algorithm: (1) Predict raw atom and bond reactive probability (i.e., reaction center detection); (2) Retrieve connected components based on the reactive probability; (3) Iterative pruning with $maxBranch = 3$. The number represents the reactive probability of the pruned atom; (4) Retrieve and rank all the candidate subgraphs (i.e. reaction centers); (5) Select the top-$n$ diverse candidates from all subgraphs.

The reaction center subgraph search is done in multiple stages: First, the algorithm removes all nodes whose $P_{rc}(s_i) < \alpha_{atom}$ and edges whose $P_{rc}(e_{ij}) < \alpha_{bond}$, and retrieves all the connected components $C$ from the edited graph. Second, for each connected component $c = (V_c, E_c)$, the algorithm retrieves all its subgraphs and the corresponding reactive scores via recursive pruning. The recursive pruning takes $maxRootSize, minLeafSize, maxBranch$ as three arguments to control its search space. If the number of nodes $|V_c|$ is larger than $maxRootSize$, then the algorithm directly removes $|V_c| - maxRootSize$ number of nodes with the lowest reactive scores. At each iteration, the algorithm considers nodes that lie along the border of the current graph as pruning candidates to ensure that the pruned graph is still a connected graph. $maxBranch$ is the maximum branching factor of the recursive pruning. The algorithm first ranks the pruning candidates by their atom reactive probability, and keeps only the top-$maxBranch$ candidates for pruning. The recursion stops when $|V_c| = minLeafSize$. After all subgraphs are retrieved from a root connected component $c$, we rank them by their reactive scores. Prior to the overall subgraph ranking, we remove all subgraphs (excluding the top-1 subgraph) that share at least two common nodes with the top-1 subgraph to ensure the candidates’ diversity. At last, we gather the remaining subgraphs from all the root connected components $C$ and retrieve the top-$n$ reaction center candidates by their reactive scores.

In our experiments, we set $n = 3$. Note that it only guarantees the maximum amount of reaction center candidates. We set the temperature $T = 10$ to flatten the reactive probabilities $P_{rc}(s)$ and $P_{rc}(e)$. As for $\alpha_{atom}$ and $\alpha_{bond}$, instead of having a fixed value, we dynamically set the two parameters as the $k_{s}^{th}$ and $k_{e}^{th}$ percentile of $P_{rc}(s)$ and $P_{rc}(e)$, respectively. Based on the best validation performance, we set $k_s = 40, k_e = 40$ for reaction class unknown setting, and $k_s = 40, k_e = 55$ for reaction class known setting; $\beta$ is the parameter that controls the $minLeafSize$. For simplicity reason, we set $\beta = 0.5, maxRootSize = 25$, and $maxBranch = 5$.

The exact reactive score for a candidate subgraph $G = (V, E)$ is computed as follow:
Algorithm 3 Reaction Center Subgraph Search

Input: \( P_{rc}(s), P_{rc}(e), G_{smi}, \alpha_{atom}, \alpha_{bond}, \beta \).

Remove all nodes whose \( P_{rc}(s_i) < \alpha_{atom} \) from \( G_{smi} \).
Remove all edges whose \( P_{rc}(e_{ij}) < \alpha_{bond} \) from \( G_{smi} \).

Retrieve all the connected components \( C \) from the edited SMILES graph.

for \( c = (V_c, E_c) \in C \) do
    Set \( max\text{RootSize} = 25 \).
    Set \( max\text{Branch} = 5 \).
    Set \( min\text{LeafSize} = \sum_{s_i \in c} 1(P_{rc}(s_i) > \beta) \), where \( s_i \in c \).
    if \( |V_c| > max\text{RootSize} \) then
        Remove \( |V_c| - max\text{RootSize} \) nodes with the lowest \( P_{rc}(s_i) \).
    end if

    Retrieve all subgraphs of \( c \) (with scores) via recursive pruning with \( max\text{Branch} \) and \( min\text{LeafSize} \).
    Remove all subgraphs who share more than two common nodes with the top-1 subgraph.
end for

Output: Subgraphs with reactive scores.

\[
\frac{1 + \phi(|V|, \mu, \sigma^2)}{M} \left( \sum_{s_i \in V} \log P_{rc}(s_i) + \sum_{e_{ij} \in E} \log P_{rc}(e_{ij}) \right) \tag{8}
\]

where \( M = |V| + |E| \) is the normalization factor, and \( \phi(\cdot) \) is the density function of a normal distribution of the size of reaction centers. We set \( \mu = 5.55 \) and \( \sigma = 1.2 \), which are computed from the training dataset. This is a heuristic factor taken from the observation that the size of the reaction centers has little relationship with the size of the molecule, but is rather normally distributed (Figure 8).