MARS: A Motif-based Autoregressive Model for Retrosynthesis Prediction

Anonymous authors
Paper under double-blind review

Abstract

Retrosynthesis is a critical task in drug discovery, aimed at finding a viable pathway for synthesizing a given target molecule. Many existing approaches frame this task as a graph-generating problem. Specifically, these methods firstly identify the reaction center, and break a targeted molecule accordingly to generate synthons. Reactants are generated by either adding atoms sequentially to synthon graphs or directly adding proper leaving groups. However, both of these strategies have limitations. Adding atoms results in a long prediction sequence which increases the complexity of generation, while adding leaving groups can only consider those in the training set which results in poor generalization. In this paper, we propose a novel end-to-end graph generation model for retrosynthesis prediction, which sequentially identifies the reaction center, generates the synthons, and adds motifs to the synthons to generate reactants. Since chemically meaningful motifs are bigger than atoms and smaller than leaving groups, our method enjoys lower prediction complexity than adding atoms and better generalization than adding leaving groups. We evaluate our proposed model on a benchmark dataset and show that it significantly outperforms previous state-of-the-art models. Furthermore, we conduct an ablation study to investigate the contribution of each component of our proposed model to the overall performance on the benchmark dataset. Our results demonstrate the effectiveness of our model in predicting rethresynthesis pathways and suggest its potential as a valuable tool in drug discovery.

1 Introduction

Retrosynthesis prediction is a fundamental problem in the field of organic chemistry, which plays a crucial role in the planning of chemical synthesis and drug discovery. The concept of retrosynthesis was first proposed by E. J. Corey, which triggered extensive research in this area. The aim of retrosynthesis prediction is to identify physically feasible reactants that can be used to synthesize target molecules, given the knowledge of their chemical structure. However, the complexity of the chemical search space make this task highly challenging. There are approximately 10^7 reactions and molecules in the published synthetic-organic knowledge Gothard et al. (2012), leading to an enormous number of possible combinations that need to be considered. Traditionally, chemists relied on their experience and knowledge to derive potential reactants, which was highly inefficient and limited in scope. For example, the complete synthetic route of vitamin B12 required the collaboration of hundreds of chemists led by Robert Woodward Woodward (1973) and took 11 years to complete. To overcome these limitations, chemists have turned to computer-aided synthesis planning (CASP) tools to design synthetic pathways. Several rule-based systems Kayala & Baldi (2012); Marcou et al. (2015) have been developed and achieve excellent results for specific reaction types, but they suffer from high complexities and have limited generalization ability on reactions outside the template library.

With the development of deep learning Wu et al. (2020); Otter et al. (2020), deep models have spawned a series of promising proposals, greatly increasing the efficiency of synthetic route design. Broadly speaking, these models can be categorized into two types: template-based Segler & Waller (2017); Coley et al. (2017); Dai et al. (2019); Yan et al. (2022) and template-free Liu et al. (2017); Zheng et al. (2019); Shi et al. (2020a); Yan et al. (2020); Sun et al. (2021b). Template-based models rely on templates that are either manually extracted by experienced chemists or automatically extracted from large-scale data Coley et al. (2019). The

core task of these methods is to match the product and the reactants to the appropriate template, which reflects the reaction center of the target molecule in a particular type of reaction. While template-based methods offer highly interpretability and can overcome the issues that traditional rule-based systems give conflicting results with functional groups Segler & Waller (2017), they are limited by the costly subgraph matching process Liu et al. (2017) and poor generalization capabilities Thakkar et al. (2020).

Template-free methods can be generally divided into sequence-based and graph-based methods. Sequence-based methods treat retrosynthesis prediction as a machine translation task. These methods use an encoder-decoder model, such as LSTM Liu et al. (2017) and Transformer Karpov et al. (2019); Zheng et al. (2019) to translate SMILES[1] sequences of target molecules into reactants SMILES sequences without atom mapping and subgraph matching. Although sequence-based methods can implicitly learn reaction rules and easily scale to larger datasets, they ignore the rich topological information presented in molecular graphs and are prone to generating invalid reactant molecules. Recently, many graph-based models for retrosynthesis have gained popularity with the development of graph neural networks. These methods typically follow a similar paradigm, consisting of reaction center identification and synthon completion. G2Gs Shi et al. (2020a), RetroXpert Yan et al. (2020), and GraphRetro Somnath et al. (2021) all use a two-stage framework to formulate the above two subtasks. However, due to the different optimization objectives of the two separate models, two-stage methods may not achieve the optimal result and can have poor generalization. Additionally, GraphRetro's use of leaving groups to complete synthons can result in unbalanced training samples and low generalization. MEGAN Sacha et al. (2021) is an end-to-end model that completes synthons with tiny units like single atom and benzene, while the lengthy prediction process makes the reactant generation challenging.

In this work, we propose a novel **M**otif-based **A**utoregressive model for **R**etro**S**ynthesis prediction (**MARS**), which jointly identifies reaction center and completes synthons in an end-to-end graph generation framework. The workflow of the entire model is shown in Fig. 1. For reaction center identification, our MARS automatically predicts which bonds in a product need to be edited, without simply ignoring samples with multiple reaction centers or introducing additional tasks to predict the number of reaction centers. For synthon completion, we employ a predefined motif vocabulary from training reactions, instead of using a single atom or ring. Motifs are fine-grained components that enjoy lower redundancy, more balanced data distribution, and more generative flexibility than leaving groups proposed by GraphRetro Somnath et al. (2021). We describe each step from product to reactants through carefully designed graph editing actions represented as a complete transformation path. Then, we adapt an RNN model to learn to generate a transformation path in an autoregressive manner. Our main contributions in this work can be summarized as:

- We integrate the two subtasks of reaction center identification and synthon completion into a unified framework, and adapt an encoder-decoder architecture for retrosynthesis prediction to train the model in an end-to-end manner.
- We extract a chemically meaningful motif vocabulary from training reactions without additional chemical knowledge, which offers more generative flexibility and greatly improves generalization ability.
- We provide a complete transformation path for each step from product to reactants, which allows for more interpretable and understandable predictions.
- Experiments on the benchmark dataset show that our model could achieve the state-of-the-art retrosynthesis performance with a Top-1 accuracy of 54.6% and 66.2% when w/o and w/ reaction type, respectively.

[1] https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html

2 Related works

2.1 Molecular Graph Generation Methods

The field of molecular graph generation has seen various approaches Madhawa et al. (2019); Zang & Wang (2020); Luo et al. (2021) aimed at generating chemically valid molecules with specific chemical properties. MolGAN De Cao & Kipf (2018) generates molecules via generative adversarial networks. JT-VAE Jin et al. (2018) first decomposes a molecular graph into disconnected subgraphs and then designs a junction tree variational autoencoder for molecule generation. Recently, autoregressive-based models have gained much attention in molecular graph generation. GCPN You et al. (2018) formulates molecular graph generation as a Markov Decision Process. MolecularRNN Popova et al. (2019) utilizes a recurrent neural network to generate the nodes and edges. GraphAF Shi et al. (2020b) designs a flow-based autoregressive model to dynamically generate nodes and edges based on historical subgraph structures. Our proposed method can be considered as a conditional molecular graph generation method based on an autoregressive model.

2.2 Template-free Retrosynthesis Prediction Methods

Template-free methods are data-driven methods that can be divided into sequence-based and graph-based methods. Sequence-based methods Tetko et al. (2020); Wang et al. (2021); Mao et al. (2021) leverage natural language processing (NLP) techniques and treat the retrosynthesis task as a machine translation problem, with molecules represented as SMILES strings. For example, Seq2seq Liu et al. (2017) and Transformer Karpov et al. (2019) simply apply machine translation models to retrosynthesis tasks, resulting in the generation of ineffective molecules. To remedy the grammatically incorrect output in the previous models, SCROP embeddings into the SMILES representation to achieve better performance than the vanilla Transformer. RetroPrime Wang et al. (2021) uses two Transformers to translate products to synthons and synthons to reactants, respectively. Although these methods simplify retrosynthetic models, they ignore the rich structural information in molecular graphs and are poorly interpretable.

Graph-based methods Shi et al. (2020a); Sacha et al. (2021); Han et al. (2022), on the other hand, model the retrosynthesis task as two steps: i) break the target molecule into incomplete molecules called synthons, and then ii) complete them into reactants using subgraph units such as atoms or leaving groups. For instance, methods such as G2Gs Shi et al. (2020a), RetroXpert Yan et al. (2020), and GraphRetro Somnath et al. (2021) build two independent models to implement the above steps respectively. MEGAN Sacha et al. (2021) constructs an end-to-end graph generative model while completing synthon with individual atoms and benzene. Our work is closely related to graph-based models but fundamentally different from the above methods. First, rather than treating reaction center identification and synthon completion as two completely independent subtasks like Shi et al. (2020a); Yan et al. (2020); Somnath et al. (2021), our work integrates these two subtasks into an end-to-end framework. Second, compared to the high prediction complexity of completing synthon with small units Sacha et al. (2021), adding motifs to synthons can greatly reduce the length of the prediction sequence. It is worth noting that motifs are distinct from leaving group proposed by Somnath et al. (2021) and the differences are discussed in subsection 3.2.

3 Proposed method

In this section, we first describe the construction of the transformation path, and then detail our proposed model MARS.

3.1 Notations

In this work, molecules are represented as graph $G = (\mathcal{V}, \mathcal{E})$ with n atoms and m bonds, where $\mathcal{V}$ is the set of atoms (nodes) and $\mathcal{E}$ is the set of bonds (edges). Each atom u has a feature vector $\boldsymbol{x}_u$ indicating its atom type, degree, chiral tag, the number of hydrogen and so on. Similarly, each bond (u, v) has a feature vector $\boldsymbol{x}_{uv}$ indicating bond type, stereo, aromaticity and so on. All features are computed by the RDKit Landrum et al. (2016) package. For convenience, an index is assigned to each bond and atom, where the bond index is its index given by RDKit, and the atom index is its index given by RDKit plus m.

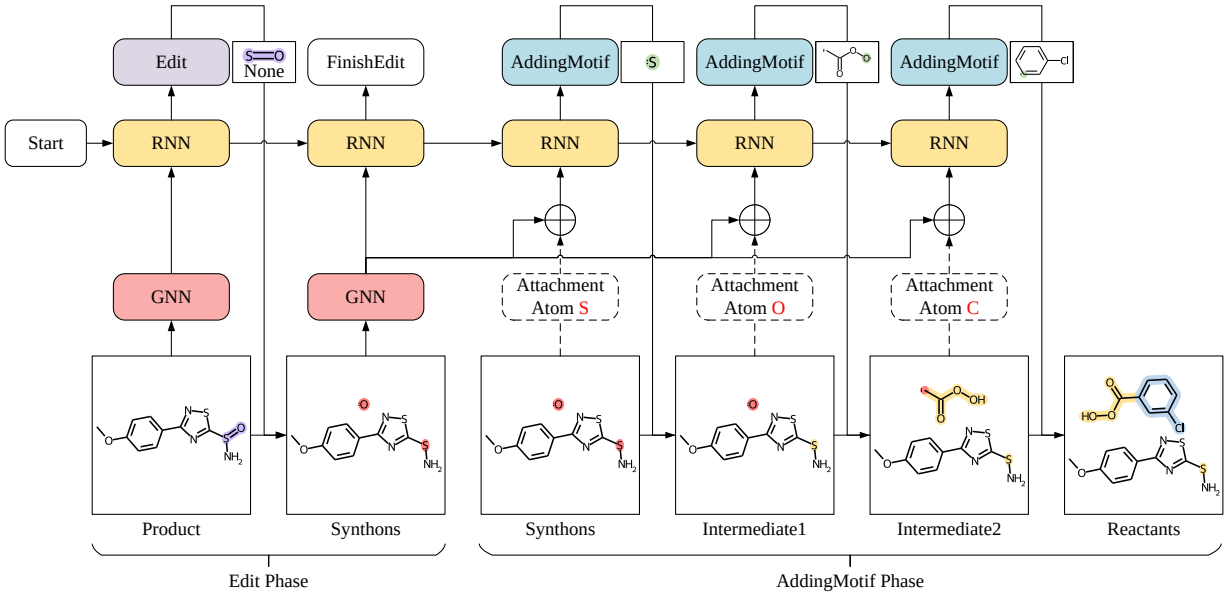

Figure 1: Reactants generation procedure of the proposed MARS. *Edit* and *AddingMotif* indicate graph transformation actions, where the *Edit* phase describes bond and atom changes from product to synthons and plays the role of reaction center identification while *AddingMotif* phase conducts synthon completion by adding proper motifs to synthons. Input molecular graphs are encoded by the Graph Neural Network (GNN), and the Recurrent Neural Network (RNN) predicts graph transformation operations sequentially. In *Edit* phase, the RNN predicts a sequence of *Edit* operations until the *FinishEdit* which indicates the end of *Edit* phase as well as the start of *AddingMotif* phase. In *AddingMotif* phase, the RNN adds motifs sequentially until no attachment atoms (highlighted in pink) remained. In the above example, the first *Edit* operation applies to the S=O bond, and the new bond type is None which indicates removing the bond. For the *AddingMotif* operation, the *interface-atom* (green) in the motif and the *attachment atom* in the synthon/intermediate represents the same atom and are merged into a single atom when attaching the motif to the synthon/intermediate.

In addition, each bond has a 4-dimensional one-hot vector $\boldsymbol{r}_b$ representing its bond type, including none, single, double and triple bonds, respectively. All bonds and atoms have a label $s_i \in \{0, 1\}$ indicating whether they belong to the reaction center.

3.2 Transformation Path Construction

In MARS, we formulate the retrosynthesis prediction as a graph generation problem. Specifically, our MARS involves predicting a sequence of graph editing actions that transform a given product into its corresponding reactants. To achieve this, we pre-construct a transformation path for each product that consist of an *Edit* phase and an *AddingMotif* phase (Fig. 1). The *Edit* phase plays the role in identifying reaction center and describing the bond and atom changes from the product to synthons. The *AddingMotif* phase, on the other hand, constructs synthon completion by adding appropriate pre-defined motifs to synthons. In particular, we introduce a hierarchical structure to represent the connection between synthons and motifs (Fig. 2b), named junction tree Jin et al. (2018), which provides an efficient way to create *AddingMotif* sequences. To integrate *Edit* and *AddingMotif* actions into a complete transformation path, we define four graph transformation tokens: *Start*, *Edit*, *FinishEdit*, and *AddingMotif*. Except for auxiliary actions *Start* and *FinishEdit*, each token in the transformation path contains three parts: edit action π, edit object o, and edit state τ. We then elaborate the representation of the reaction center in the Edit sequence, motif extraction, and junction tree construction.

Table 1: notation

Notation	Short Explanation
G_*	Molecular graph
$\mathcal{V}$	Set of atoms
$\mathcal{E}$	Set of bonds
Z	Motif vocabulary and motif.
a_j	Attachment atom j.
q_j	Interface-atom j.
π_t	Graph editing action at t step.
o_t	Edit object at t step.
τ_t	Edit state at t step.
s_i	Editing score of edit object i.
u_t	The output vector of GRU at t step.
ψ_t	The vector derived from molecular graph embedding and the output of GRU at t step.
$\sigma_*(\cdot)$	Linear layer of neural networks with any nonlinear activation.
$f_*(\cdot)$	Linear layer of neural networks with any nonlinear activation, mapping entities to vectors.
$\boldsymbol{h}_*$	Embedding vector.

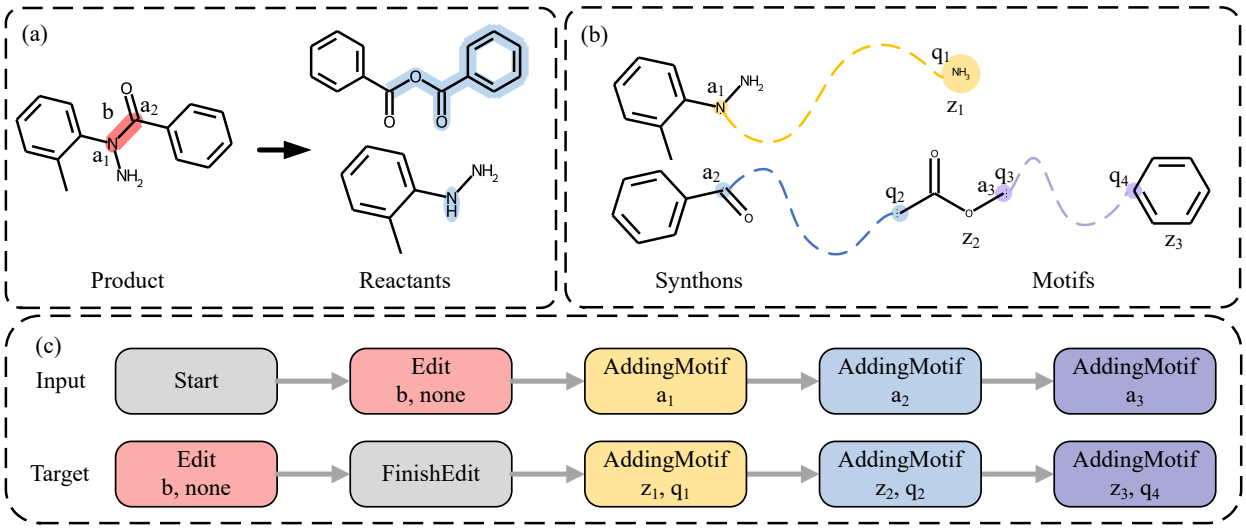

Figure 2: (a) The conversion of products to reactants. The bond b marked pink represents the reaction center, while the atoms a_1 and a_2 are attachment atoms; (b) The junction tree constructed by synthons and motifs z. The atoms of the same color represent connected attachment atom and interface atom q, which are the same atom in reactant. The arrows represent the parent node pointing to the child node. After connecting synthons with z_2, the interface atom q_3 becomes the attachment atom a_3; (c) The input and target transformation paths for training RNN, constructed according to (a) and (b).

3.2.1 Edit Sequence Construction

The *Edit* phase of our MARS involves two types of edits: bond edit and atom edit. Bond edit is the process of adding or removing a bond or changing the bond type between two heavy atoms, while atom edit is the process of changing the number of hydrogen or charge of an atom. These edits are applied to a target molecule to obtain synthons, and we refer to the atoms at both ends of the changed bond, as well as the atoms with changed hydrogen or charges as *attachment atoms*. This is because we need to attach a motif

to every attachment atom to complete synthons. The reaction center of the target molecule can be encoded as an *Edit* sequence, where each *Edit* token in the sequence is a tuple of *(edit action, edit object, edit state)*. For example, in Fig. 2a, the bond marked in pink is the reaction center of the given product and the *Edit* tuple is denoted as *(Edit, b, none)*, where b is the bond index and *none* describes the new bond type which indicates removing the bond.

Once we obtain synthons resulting from the *Edit* operations, we can add motifs to synthons to generate reactants. *AddingMotif* token represents adding a motif from the vocabulary and attaching the motif to a specific attachment. A motif can contains one or more attachments. We add motifs sequentially until all attachments are properly processed.

3.2.2 Motif Extraction

Through *Edit*, a product molecular graph can be decomposed into a group of incomplete subgraphs called synthons. By combining appropriate motifs and attachments, synthons can be reconstructed into valid reactant molecular graphs. In other words, motifs are the subgraphs of reactant molecular graphs. The details of motif extraction are summarised as follows:

- Bonds that connect the synthon in the reactants are broken to obtain a set of subgraphs. Each subgraph retains the attachment atoms connected to it on the synthon, resulting in a coarse-grained motif.
- If two connected atoms belong to two rings, the bond between them is broken, resulting in two independent motifs.
- If one atom belongs to a ring and the other has a degree greater than 1, the bond between them is broken, resulting in two independent motifs.

Finally, a motif vocabulary Z of size $|Z| = 210$ is obtained from the USPTO-50K training set Schneider et al. (2016). It is worth noting that motifs are fundamentally different from leaving groups proposed by the previous method Somnath et al. (2021). i) A motif is connected to an attachment atom, whereas a leaving group is combined with a synthon. Since one synthon may contain more than one attachment atom, and a leaving group may consist of multiple disconnected subgraphs (i.e. motifs). ii) Motif retains the corresponding attachment atom on synthon, referred to as the *interface-atom*. We observe that a large portion of the added leaving groups is single hydrogen, which results in an extremely unbalanced frequency of leaving groups. iii) A large leaving group may contain multiple rings or large branched chains, which appear infrequently in the dataset. To reduce redundancy, We cut these into multiple small motifs that are common in the dataset.

3.2.3 Junction Tree Construction

Based on chemical intuition, it is postulated that reactants can be decomposed into synthons and motifs, where synthons represent molecule fragments obtained by breaking the bonds in the product, and motifs represent subgraphs of reactants. to maintain the connection between synthons and motifs, the junction tree method is introduced, as described in Jin et al. (2018). The junction tree method represents synthons and motifs as a hierarchical tree structure, where the group of synthons is set as the root node and motifs are set as children nodes (Fig. 2b). The connected edge between two nodes indicates that they are directly linked in the reactants, denoted as *(attachment, motif, interface-atom)*. The trees are traversed using depth-first search (DFS) to preserve the linked edges between nodes, and to obtain the training input and target AddingMotif paths. The input path consists of each token containing an action *AddingMotif* and an object *attachment atom*, while for target path, the object consists of the *motif* z and the *interface-atom* q.

By combining the aforementioned Edit sequence with the AddingMotif path, as well as other auxiliary actions, the input and target transformation paths corresponding to each product can be obtained.(Fig. 2c).

3.3 Graph Encoder

Graph neural networks (GNNs) Kipf & Welling (2016); Hamilton et al. (2017); Veličković et al. (2017); Xu et al. (2018) are a series of neural network architectures specifically designed to acting on graph structure and properties, updating the representation vectors (i.e. embeddings) of nodes via a message passing mechanism. In this work, we employ an L-Layer graph transformer network (GTN) Shi et al. (2020c) to encode the latent representation of molecular graph. GTN utilizes improved multi-head self-attention modules in vanilla transformer Vaswani et al. (2017) to aggregate both atom and bond features. In the head c of layer l, the query $\boldsymbol{q}_{c,u}^{(l)}$, keyword $\boldsymbol{k}_{c,u}^{(l)}$, and value $\boldsymbol{v}_{c,u}^{(l)}$ vectors correspond to the atom representation vector $\boldsymbol{h}_u^{(l)}$, which are transformed by the query, keyword, and value matrices. The GTN integrates bond features $\boldsymbol{x}_{uv}$, calculates an attention score to the bond (u, v), and updates the representation of atom u by message passing scheme as follows:

$$\begin{aligned}
\boldsymbol{h}_{c,uv} &= \boldsymbol{W}_{c,e}\boldsymbol{x}_{uv} + \boldsymbol{b}_{c,e}, \\
\alpha_{c,uv}^{(l)} &= \frac{\langle \boldsymbol{q}_{c,u}^{(l)}, \boldsymbol{k}_{c,v}^{(l)} + \boldsymbol{h}_{c,uv} \rangle}{\sum_{v \in \mathcal{N}(u)} \langle \boldsymbol{q}_{c,u}^{(l)}, \boldsymbol{k}_{c,v}^{(l)} + \boldsymbol{h}_{c,uv} \rangle}, \\
\hat{\boldsymbol{h}}_u^{(l+1)} &= \Big\|_{c=1}^{C} [\sum_{v \in \mathcal{N}(u)} \alpha_{c,uv}^{(l)} (\boldsymbol{v}_{c,v}^{(l)} + \boldsymbol{h}_{c,uv})],
\end{aligned} \tag{1}$$

where $\langle \boldsymbol{q}, \boldsymbol{k} \rangle = \exp(\frac{\boldsymbol{q}^T \boldsymbol{k}}{\sqrt{d}})$, d is the dimension of each head, $\mathcal{N}(u)$ denotes the neighbors of atom u and $\|$ is the concatenation operation. Then the gated residual connection is introduced to avoid over-smoothing.

Finally, the atom representations $\{\boldsymbol{h}_u \in \mathbb{R}^D | u \in G\}$ are gained. The self-loop representations of atoms and bond representations are expressed as

$$\boldsymbol{h}_{uu} = \mathrm{MLP}_{bond}(\boldsymbol{h}_u \parallel \boldsymbol{h}_u), \tag{2}$$

$$\boldsymbol{h}_{uv} = \mathrm{MLP}_{bond}(\boldsymbol{h}_u \parallel \boldsymbol{h}_v), \tag{3}$$

where $\mathrm{MLP}_*(\cdot)$ denotes a Multilayer Perceptron with a Mish Misra (2019) activation function.

For convenience, we use the same notation $\boldsymbol{e}_i \in \{\boldsymbol{h}_{uv}\}_{v \in \mathcal{N}(u) \bigcup \{u\}}$ to represent both the bond and atom representations, where i is the index of the bond or atom. The final graph representations $\boldsymbol{h}_G \in \mathbb{R}^D$ is defined by aggregating the whole atom representations using a readout function (i.e. mean, max, sum and attention pooling) as follows:

$$\boldsymbol{h}_G = \mathrm{Readout}(\{\boldsymbol{h}_u | u \in G\}). \tag{4}$$

Similarly, the graph representation of synthons $\boldsymbol{h}_{syn} \in \mathbb{R}^D$ can also be computed.

3.4 Autoregressive Model

Motivated by previous works Popova et al. (2019); Shi et al. (2020b), the task of retrosynthesis prediction is formulated as an autoregressive conditional molecule generation problem. In this method, the autoregressive model generates a new graph structure G_t based on the incomplete graph of the previous steps, until the reactant graph G_R is finally obtained. This general process can be defined as a jointly conditional likelihood function:

$$\mathcal{P}(G_R | G_P) = \prod_{t=1}^{N} \mathcal{P}(G_t | G_0, \ldots, G_{t-1}) = \prod_{t=1}^{N} \mathcal{P}(G_t | G_{<t}), \tag{5}$$

where N is the length of generated sequence and G_0 is the given product graph G_P. It is important to note that the intermediate graph structure G_t is not directly generated by the model. Instead, the model generates a graph editing action π, an edit object o (i.e. bond, atom, or motif), and its edit state τ (e.g. new bond type or interface-atom) from the historical graph editing sequence. These are then applied to G_{t-1} to

obtain a new graph structure. Therefore, given the historical edited objects, edited states, and incomplete graphs, the likelihood in Eq. (5) can be modified as:

$$\mathcal{P}(G_R) = \prod_{t=1}^{N} \mathcal{P}(\pi_t, o_t, \tau_t | o_{<t}, \tau_{<t}, G_{<t}). \tag{6}$$

We utilize a recurrent neural network (RNN) to model the likelihood function Eq. (6), where the object, state and incomplete graph of the previous step are encoded to decode the D-dimension output $\boldsymbol{u}_t \in \mathbb{R}^D$. To incorporate the global topological information of G_P in generation process, we concatenate $\boldsymbol{h}_G$ and $\boldsymbol{u}_t$ for subsequent prediction. Specifically, the Gated Recurrent Unit Chung et al. (2014); Li et al. (2015b), denoted as GRU$(\cdot)$, is employed as follows:

$$\boldsymbol{u}_t = \text{GRU}(\boldsymbol{input}_t), \ \textit{where } \boldsymbol{input}_0 = \mathbf{0}, \tag{7}$$

where $\boldsymbol{input}_t \in \mathbb{R}^D$ is the input embedding of GRU at step t. The hidden state of GRU is initialized by $\sigma_G(\boldsymbol{h}_G)$, where $\sigma_*(\cdot)$ is a linear layer without nonlinear activation. The resulting vector $\boldsymbol{u}_t$ is then combined with $\boldsymbol{h}_G$ using the concatenation operation as:

$$\boldsymbol{\psi}_t = \boldsymbol{h}_G \parallel \boldsymbol{u}_t, \tag{8}$$

The generation process starts with the *Start* action, and at each step t, we generate graph editing action $\hat{\pi}_t$ by follow:

$$\hat{\pi}_t = \text{softmax}(\text{MLP}_{act}(\boldsymbol{\psi}_t)). \tag{9}$$

Edit Phase When the predicted action is *Edit*, the process enters the Edit phase. At step t, the model firstly assigns an editing score $\hat{s}_i$ to each bond and atom, indicating the likelihood that the bond or atom is a suitable candidate for editing. The editing score is computed as follows:

$$\hat{s}_i^t = \text{sigmoid}(\text{MLP}_{target}(\boldsymbol{\psi}_t \parallel \sigma_e(\boldsymbol{e}_i))). \tag{10}$$

The atom or bond with the largest editing score is selected as the edit object, and the atom or atoms at both ends of the selected bond are set as attachment atoms. The model then predicts the new bond type $\hat{r}_b$ for the edit object as:

$$\hat{r}_b^t = \text{softmax}(\text{MLP}_{type}(\boldsymbol{\psi}_t \parallel \sigma_e(\boldsymbol{e}_{\arg\max_i(\hat{\boldsymbol{s}}_i^t)}))). \tag{11}$$

The synthon structure is then modified by applying the edit object and its new bond type, and the resulting structure is embedded using GTN$(\cdot)$ to obtain the synthon embedding $\boldsymbol{h}_t^{syn}$. Finally, $\boldsymbol{input}_t$ is updated by synthon embedding, edit object and its new bond type:

$$\boldsymbol{input}_{t+1} = f_\pi(\hat{\pi}_t) + \sigma_e(\boldsymbol{e}_{\arg\max_i(\hat{\boldsymbol{s}}_i^t)}) + f_b(\hat{r}_b^t) + \boldsymbol{h}_{syn}^t, \tag{12}$$

where $f_*(\cdot)$ is a linear layer without activation functions, mapping entities to vectors. The model iterates this process to generate an Edit sequence that covers all reaction centers. When the model predictes the action to be a *FinishEdit*, the Edit phase ends and AddingMotif phase begins. The synthon structure is fixed and its embedding is denoted as $\boldsymbol{h}_{syn}$. Assume that after N_1 Edit operations, a total of N_2 attachment atoms $\{a_1, ..., a_{N_2}\}$ are obtained, where a_j is the atom index in target molecular graph G_P. Then the set of attachment atoms is sorted, and $\boldsymbol{input}_{t+1}$ is updated as:

$$\boldsymbol{input}_{t+1} = f_\pi(\hat{\pi}_t) + \sigma_{att}(\boldsymbol{e}_{m+a_t}) + \boldsymbol{h}_{syn}. \tag{13}$$

AddingMotif Phase In this phase, the model proceeds by traversing all the attachment atoms $\{a_1, ..., a_{N_2}\}$ sequentially and assigning a appropriate motif to each attachment. Motif prediction is treated as a multi-classification task on the motif vocabulary Z. Once the predicted motif $\hat{z}_t$ is obtained, the model determines

which interface-atom on the motif corresponds to the attachment atom a_t. To achieve this, the model predicts both the motif $\hat{z}_t$ and interface-atom index $\hat{q}_t$ as follows:

$$\hat{z}_t = \text{softmax}(\text{MLP}_{motif}(\boldsymbol{\psi}_t)), \tag{14}$$

$$\hat{q}_t = \text{softmax}(\text{MLP}_{interface}(\boldsymbol{\psi}_t \parallel f_z(\hat{z}_t))). \tag{15}$$

If the predicted motif $\hat{z}_t$ contains only one interface-atom, the input representation $\boldsymbol{input}_{t+1}$ is computed as Eq. (13). However, if $\hat{z}_t$ contains multiple interface-atoms, and $\hat{q}_t$ is the $\boldsymbol{input}_{t+1}$ is updated as follows:

$$\boldsymbol{input}_{t+1} = f_\pi(\hat{\pi}_t) + f_z(\hat{z}_t) + f_{interface}(\hat{q}_t) + \boldsymbol{h}_{syn}. \tag{16}$$

It is worth noting that there is no need to incorporate an action to indicate the end of the process. The generation process continues until all attachments on the synthons and added motifs have been traversed. Finally, the model generates a transformation path, which is applied to the product to obtain reactants.

3.5 Training and Inference

Training MARS is trained to predict target transformation paths given training transformation paths. The model is optimized using cross-entropy loss $\mathcal{L}_c$ for predicting new types, motifs, and interface-atom indexes, and binary cross-entropy loss $\mathcal{L}_b$ for predicting reaction centers. The overall loss function for MARS is the summation of these losses over all the steps in the Edit and AddingMotif sequences. Specifically, for the Edit phase, the loss function includes both the binary cross-entropy loss for the reaction center prediction and the cross-entropy loss for predicting the reaction type. For the AddingMotif phase, the loss function includes the cross-entropy losses for predicting the motif and interface-atom indexes. Thus, the overall loss function for MARS is summarized as:

$$\mathcal{L} = \sum_{t=0}^{N_1+N_2} \mathcal{L}_c(\hat{\pi}_t, \pi_t) + \sum_{t=0}^{N_1} \left[\sum_{i=0}^{n+m-1} \mathcal{L}_b(\hat{s}_i^t, s_i^t) + \mathcal{L}_c(\hat{r}_b^t, r_b^t) \right] + \sum_{t=0}^{N_2} \left[\mathcal{L}_c(\hat{z}_t, z_t) + \mathcal{L}_c(\hat{q}_t, q_t) \right], \tag{17}$$

where N_1 and N_2 denote the lengths of the Edit and AddingMotif sequences respectively. To reduce convergence difficulties, we adopt an efficient strategy teacher-forcing Williams & Zipser (1989) to train our model. When training model, the strategy utilizes ground truth as the input for the model instead of directly uses the output from the last time step as the input at current time step.

Inference During inference, the beam search algorithm Tillmann & Ney (2003) with hyperparameter k is used to rank the predictions. At each time step, the Top-k best results are selected as the input at next time step based on the log-likelihood score function. In other words, the process can be described as the construction of a search tree, in which the leaf nodes with the highest scores are expanded with their children nodes while the other leaf nodes are dropped. It is important to note that atom-mapping in the testing set is unnecessary in the inference phase.

4 Results

4.1 Experiment Setup

4.1.1 Data

We evaluate the effectiveness of our proposed approach on a widely used benchmark dataset called USPTO-50K Schneider et al. (2016). This dataset includes a collection of 50K reactions from the US patent literature, which are categorized into ten different classes. We follow the same training/validation/testing splits in an 8:1:1 ratio, as previously established in Coley et al. (2017); Dai et al. (2019). Notably, the USPTO dataset has been reported to contain a shortcut in 75% of the product molecules, where the atom of atom-mapping "1" is part of the reaction center. To address this issue, we eliminate these shortcuts by canonicalizing product SMILES and reassigning atom-mapping to reactant atoms.

Algorithm 1: Framework of MARS.

Input: The SMILES string of product.
Output: The SMILES string of reactants.

Convert the SMILES string of the product to a molecular graph G_P, and compute atom feature $\{\boldsymbol{x}_u\}$ and bond feature $\{\boldsymbol{x}_{uv}\}$ by using RDKit;
$\{\boldsymbol{e}_i\}, \boldsymbol{h}_G \leftarrow \text{GraphEncoder}(\{\boldsymbol{h}_u | u \in G_P\})$;
Initialize $\pi_0 \leftarrow \textit{Start}$, $t \leftarrow 0$, *hidden state of* $\text{GRU}(\cdot) \leftarrow \sigma_G(\boldsymbol{h}_G)$, $\boldsymbol{input}_0 \leftarrow \mathbf{0}$;
while π_t *is not FinishEdit* **do**
  $\boldsymbol{u}_t \leftarrow \text{GRU}(\boldsymbol{input}_t)$;
  $\boldsymbol{\psi}_t \leftarrow \boldsymbol{h}_G \| \boldsymbol{u}_t$;
  $\hat{\pi}_t \leftarrow \text{softmax}(\text{MLP}_{act}(\boldsymbol{\psi}_t))$;
  $\hat{s}_i \leftarrow \text{sigmoid}(\text{MLP}_{target}(\boldsymbol{\psi}_t \parallel \sigma_e(\boldsymbol{e}_i)))$;
  $\hat{r}_b \leftarrow \text{softmax}(\text{MLP}_{type}(\boldsymbol{\psi}_t \parallel \sigma_e(\boldsymbol{e}_{\underset{i}{\arg\max(\hat{s}_i)}})))$;
  $\boldsymbol{input}_{t+1} \leftarrow f_\pi(\hat{\pi}_t) + \sigma_e(\boldsymbol{e}_{\underset{i}{\arg\max(\hat{s}_i)}}) + f_b(\hat{r}_b) + \boldsymbol{h}_t^{syn}$;
  $t \leftarrow t + 1$
$\boldsymbol{h}_{syn} \leftarrow \text{GraphEncoder}(\{\boldsymbol{h}_u | u \in G_{syn}\})$;
The set of attachment atoms is sorted according to their atom index in G_P;
$\boldsymbol{input}_t = f_\pi(\hat{\pi}_t) + \sigma_{att}(a_1) + \boldsymbol{h}_{syn}$;
foreach *set of attachment atoms* $\{a\}$ **do**
  $\boldsymbol{u}_t \leftarrow \text{GRU}(\boldsymbol{input}_t)$;
  $\boldsymbol{\psi}_t \leftarrow \boldsymbol{h}_G \| \boldsymbol{u}_t$;
  $\hat{\pi}_t \leftarrow \text{softmax}(\text{MLP}_{act}(\boldsymbol{\psi}_t))$.;
  $\hat{z}_t \leftarrow \text{softmax}(\text{MLP}_{motif}(\boldsymbol{\psi}_t))$;
  $\hat{q}_t \leftarrow \text{softmax}(\text{MLP}_{interface}(\boldsymbol{\psi}_t \parallel f_z(\hat{z}_t)))$;
  if *motif* $\hat{z}_t$ *has only one interface-atom* **then**
    $\boldsymbol{input}_{t+1} \leftarrow f_\pi(\hat{\pi}_t) + \sigma_{att}(\boldsymbol{e}_{a_t}) + \boldsymbol{h}_{syn}$;
  else
    $\boldsymbol{input}_{t+1} = f_\pi(\hat{\pi}_t) + f_z(\hat{z}_t) + f_{interface}(\hat{q}_t) + \boldsymbol{h}_{syn}$;
  $t \leftarrow t + 1$
Convert the reactant molecular graph to SMILES string;

4.1.2 Evaluation

We employ a standard evaluation metric known as Top-k accuracy. This metric is calculated as the percentage of ground truth reactants that appear in the Top-k suggestions provided by our model. Specifically, the accuracy is determined by comparing the predicted reactants with the ground truth reactants, both of which are represented in canonical SMILES format.

4.1.3 Implementation Details

We use PyTorch Paszke et al. (2019) and Pytorch Geometric Fey & Lenssen (2019) library to implement our model. For Graph Transformer, we stack six eight-head self-attention modules, and the attention pooling Li et al. (2015a) is used as the readout function. The GRU network is implemented with three layers. All embedding size D in our model is set to 512. In all experiments, we train on USPTO-50K for 100 epochs, using a batch size of 32 and the Adam Kingma & Ba (2014) optimizer with initial learning rate of 0.0003. The learning rate is adjusted with the strategy of cosine annealing learning rate with restart Loshchilov & Hutter (2016), and the restart cycle is set to 20 epochs. The training on USPTO-50K takes approximately 17h on a single NVIDIA Tesla V100 GPU. The beam size k is set to 10 in the inference phase.

4.1.4 Baseline

We take three template-based and eight template-free methods as our competitors.

For template-based models,

- **RetroSim** Coley et al. (2017) selects reaction centers based on Morgan fingerprint similarity between target molecules and known precedents.

- **NeuralSym** Segler & Waller (2017) combines a fully-connect layer and a deep highway network to learn knowledge of potential correlations between molecular functional groups and reactions.
- **GLN** Dai et al. (2019) models the joint probability of single-step retrosynthesis to select templates and generate reactants.

Template-free models can be divided into four sequence-based models and four graph-based models. For sequence-based models,

- **SCROP** Zheng et al. (2019) combines an extra Transformer to correct predicted SMILES strings.
- **LV-Transformer** Chen et al. (2019) uses a pre-training strategy and introduces latent variables to improve prediction diversity.
- **RetroPrime** Wang et al. (2021) uses two Transformers to model reaction center identification and synthons completion, respectively.
- **DualTF** Sun et al. (2021a) unifies sequence-based and graph-based models using energy functions and uses an extra order model to help inference.

For graph-based models,

- **G2Gs** Shi et al. (2020a) employs a graph neural network to select reaction centers and generates reactants using a variational autoencoder.
- **RetroXpert** Yan et al. (2020) leverages a graph neural network to predict disconnections and regards reactant generation as a sequence translation task.
- **GraphRetro** Somnath et al. (2021) determines the synthon through an edit prediction model and then performs a single full-connected network to complete the synthons by using predefined leaving groups.
- **MEGAN** Sacha et al. (2021) defines five graph editing actions, using two stacked graph attention networks to perform retrosynthesis predictions.

All results are derived from their original reports, except for NeuralSym Segler & Waller (2017) reported by GLN Dai et al. (2019), and corrected results reported by RetroXpert Yan et al. (2020) on their website [2].

4.2 Overall Performance

We present the Top-k accuracy in Table 2, where N ranges from $\{1, 3, 5, 10\}$. We evaluate both the reaction type unknown and reaction type known.

4.2.1 Reaction Type Unknown

When reaction type is unknown, our model surpasses both template-based and template-free models. Our model outperforms GraphRetro by 0.9% and MEGAN by 6.5% in terms of Top-1 accuracy. Moreover, for larger k, our model still enjoys high performance, which is over 8.1% than GraphRetro and 2.4% than MEGAN. We notice that both our model and MEGAN outperform two-stage models when $n \geq 3$. Benefiting from an end-to-end model, our model is able to explore the underlying relationship between reaction centers and synthon completion, rather than relying on two separate modules with different optimization objectives. Furthermore, our model takes advantage of motifs that are larger than individual atoms and smaller than leaving groups, allowing it to avoid the high complexity of long prediction sequences without sacrificing flexibility.

[2] https://github.com/uta-smile/RetroXpert

Table 2: Top-k accuracy for retrosynthesis prediction on USPTO-50K.

Methods		Top-k Accuracy (%)							
		Reaction Type Known				Reaction Type Unknown			
		1	3	5	10	1	3	5	10
Template-based	RetroSim	52.9	73.8	81.2	88.1	37.3	54.7	63.3	74.1
	NeuralSym	55.3	76	81.4	85.1	44.4	65.3	72.4	78.9
	GLN	64.2	79.1	85.2	90.0	52.5	69	75.6	83.7
Sequence-based	SCROP	59.0	74.8	78.1	81.1	43.7	60.0	65.2	68.7
	LV-Transformer	-	-	-	-	40.5	65.1	72.8	79.4
	RetroPrime	64.8	81.6	85.0	86.9	51.4	70.8	74.0	76.1
	DualTF	65.7	81.9	84.7	85.9	53.6	70.7	74.6	77.0
Graph-based	G2Gs	61.0	81.3	86.0	88.7	48.9	67.6	72.5	75.5
	RetroXpert	62.1	75.8	78.5	80.9	50.4	61.1	62.3	63.4
	MEGAN	60.7	82.0	87.5	91.6	48.1	70.7	78.4	86.1
	GraphRetro	63.9	81.5	85.2	88.1	53.7	68.3	72.2	75.5
	Ours	**66.2**	**85.8**	**90.2**	**92.9**	**54.6**	**76.4**	**83.3**	**88.5**

4.2.2 Reaction Type Known

When reaction type is given, our model outperforms MEGAN and GraphRetro by 5.5% and 2.3% in Top-1 accuracy. For larger k, our model also achieves state-of-the-art Top-k accuracy of 85.8%, 90.2% and 92.9%, which is over 4.3% higher than GraphRetro. Even though template-based methods can exploit the reaction type to narrow the template space and improve the accuracy, they also suffer from poor generalization. In contrast, our model can improve accuracy while still maintaining high generalization performance.

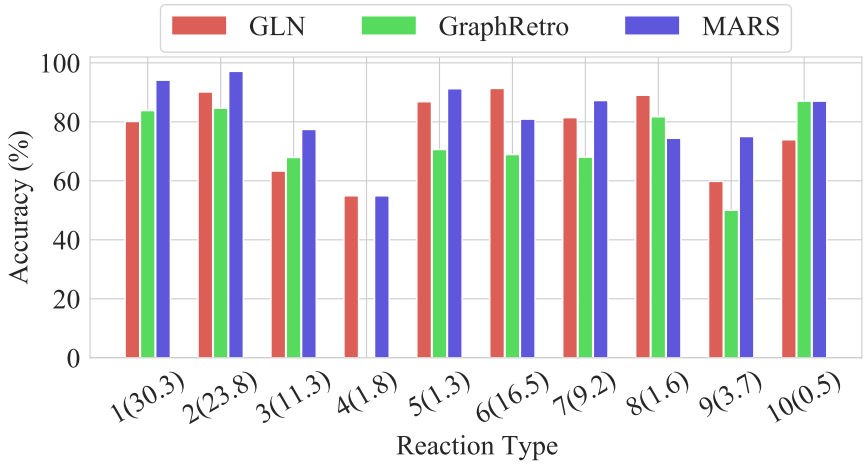

Figure 3: Comparison of the Top-10 accuracy across the USPTO-50K reaction types. We report the results of GLN and GraphRetro with beam size of 10. The labels on the x-axis represent reaction types and their proportion in USPTO-50K dataset.

4.2.3 Reaction Type Performance

The performance of MARS for each reaction type is investigated in Fig. 3. The results show that MARS achieves competitive performance in eight categories compared with the template-based method GLN. Additionally, MARS outperforms the baseline methods on reaction types with fewer samples, such as class 5 and 9. This suggests that MARS does not suffer from overfitting even on an imbalanced dataset. Notably,

the reaction type 4 is heterocycle formation, which contains multiple reaction centers. GraphRetro only considered the samples with a single reaction center, resulting in inaccurate predictions for such samples. Our model requires no additional chemical knowledge and reaches the Top-10 accuracy of 54.9%, the same as GLN.

4.3 Parameter Analysis

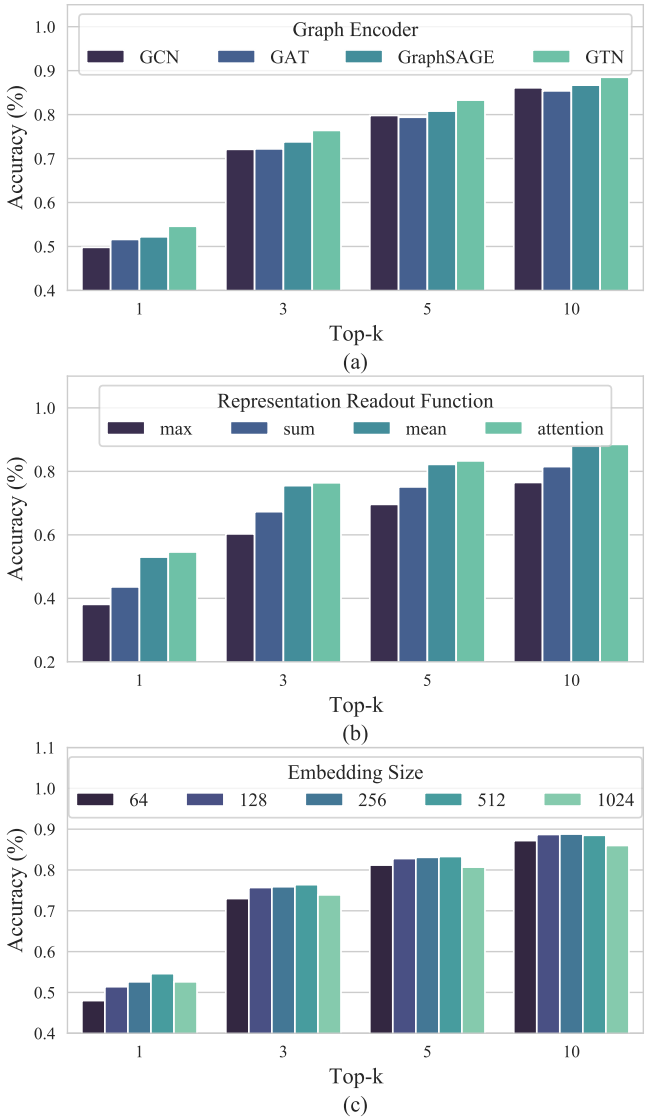

Figure 4: (a). Performance comparison of different graph encoders. (b). Performance comparison of different representation readout functions. (c). Performance comparison of different dimension sizes.

4.3.1 Effects of Graph Encoder

The graph encoder is responsible for learning the representations of nodes by aggregating their neighbors' information, which is used to capture the topological information of molecular graphs. We investigate the effect of different graph encoders on the performance of MARS, including GCN Kipf & Welling (2016), GAT Veličković et al. (2017), GraphSAGE Hamilton et al. (2017) and GTN Shi et al. (2020c). As shown in Fig.

4(a), GTN outperforms the other graph encoders. This is because the self-attention module in GTN can better fuse bond features into the atom representations, leading to more effective learning of the molecular graph.

4.3.2 Effects of Representation Readout Function

The representation readout function is utilized to aggregate all atom representations to obtain the molecular graph representation. We compare the effects of several readout functions on performance of MARS, including max, sum, mean and attention pooling. Fig. 4(b) shows the Top-k accuracy of different readout functions. We find that attention pooling achieves the best performance for learning the representation of molecular graphs. This is because attention pooling can better extract information from atom representations that is beneficial to downstream tasks.

4.3.3 Effects of Dimension Size

Embedding size K has a significant impact on performance. As shown in Figure 4(c), we test the performance of MARS when K takes on values of 64, 128, 256, 512, 1024. We find that the Top-1 accuracy is optimal when the embedding size is 512, and the performance does not increase further when the embedding size is 1024. For larger k, the accuracy of MARS is not sensitive to K. This demonstrates that 512 dimensions are sufficient for our model to perform optimally.

4.4 Ablation Study

To gain insight into the importance of synthon embedding, we conduct an ablation study by removing it in MARS. As shown in Table 3, when the synthon embedding is not included, the top-1 accuracy drops by 4.9% if the reaction type is given and 10.5% otherwise. This demonstrates that synthon embedding plays an indispensable role in the generation process. We observe that synthon structure information helps the model determine *FinishEdit* action, while the model without synthon embedding suffers from the problem of repeatedly predicting edit objects in *Edit* phase.

In addition, we also examined the importance of bond features by removing them from MARS and testing the performance of the model. The Top-k accuracy of MARS without the bond feature is shown in Table 3. The Top-1 accuracy of MARS-w/o B is 2.8% lower than the MARS when the reaction type is unknown, and 2.2% lower when the reaction type is known. This demonstrates that the use of bond features enables MARS to better learn molecular representations, which can improve the accuracy of downstream predictions.

Table 3: Top-k accuracy of synthon embedding ablation study. *MAR-w/o S* indicates MAR without synthon embedding. *MAR-w/o B* indicates MAR without bond feature.

Method	Top-k Accuracy (%)							
	Reaction Type Known				Reaction Type Unknown			
	1	3	5	10	1	3	5	10
MARS-w/o S	61.3	73.5	76.3	81.8	44.1	58.5	63.0	69.3
MARS-w/o B	64.0	84.4	89.3	92.4	51.8	74.6	81.5	86.8
MARS	**66.2**	**85.6**	**90.2**	**92.9**	**54.6**	**76.4**	**83.3**	**88.5**

4.5 Prediction Visualization

To provide a more comprehensive understanding of the prediction performance of our model, we visualize four ground truth reactants and Top-1 predicted reactants from USPTO-50K test set in Fig. 5. In Fig. 5a and 5b, our model correctly predicts the reactants with accurate identification of the reaction centers and addition of appropriate motifs. Notably, our model is insensitive to the size of motifs, indicating its ability to assign the correct motifs for the synthons. Compared to methods that add atoms or benzene rings one by one, our model's predictions demonstrate high accuracy and chemical rationality. Fig. 5c shows a

failure case in which the reaction center is correctly predicted but motifs are different from the ground truth. However, the predicted reactants are chemically reasonable, since they can be more conveniently obtained in some cases. In Fig. 5d, our model predicts another disconnection site and adds corresponding motifs based on the predicted synthons. The predictions are also correct (checked by chemists), as the prediction and ground truth differ only in the disconnection order from multi-step retrosynthesis perspective. These examples illustrate that our model can inherently learn underlying reaction rules and provide predictions with high chemical rationality.

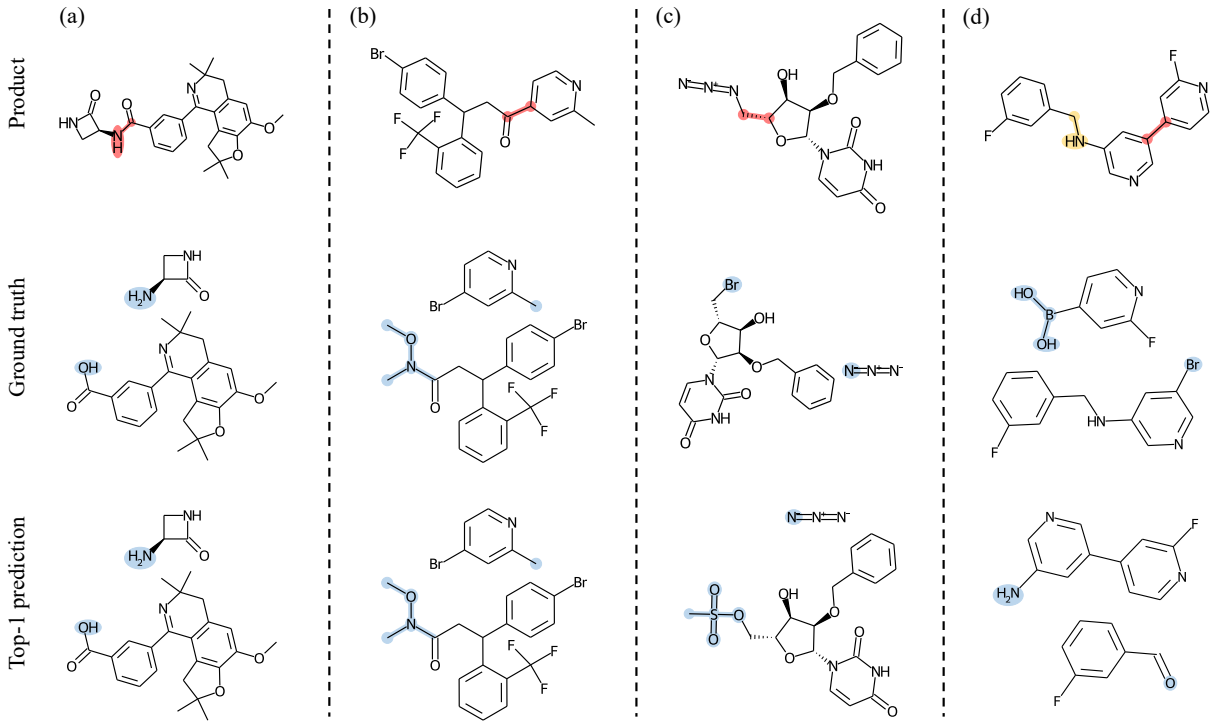

Figure 5: Example Predictions. *Red* indicates the correct reaction centers while *yellow* represents the error one predicted by our model, and *blue* indicates the added motifs. (a)(b). Examples of successful predictions by our model. (c). Correctly predicted reaction center but added wrong motif. (d). Incorrectly predicted reaction center.

5 Conclusion

Motivated by the classical retrosynthesis theory proposed by Nobel Prize laureate E.J.Corey, we have proposed a graph generative model MARS for retrosynthetic analysis. Our model benefits from the flexibility and low prediction complexity of motifs. The end-to-end architecture enables our model to explore latent relationships between reaction centers and motifs. Moreover, the motifs are functional groups in chemistry. It is very reasonable to treat them as elementary entities in retrosynthetic prediction task. All these account for why the high accuracy and excellent generalization performance are obtained by our model. In the future, pre-training a model to learn more reasonable motifs from existing chemical compounds will be tried.

Our work and the existing retrosynthesis methods share the limitation that lacking a more reasonable evaluation metric. Our work is capable of obtaining multiple chemically plausible reactants to synthesize products, but existing evaluation metrics only consider a given reaction.

References

Benson Chen, Tianxiao Shen, Tommi S Jaakkola, and Regina Barzilay. Learning to make generalizable and diverse predictions for retrosynthesis. *arXiv preprint arXiv:1910.09688*, 2019.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS central science*, 3(12):1237–1245, 2017.

Connor W Coley, William H Green, and Klavs F Jensen. Rdchiral: An rdkit wrapper for handling stereochemistry in retrosynthetic template extraction and application. *Journal of chemical information and modeling*, 59(6):2529–2537, 2019.

Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. *Advances in Neural Information Processing Systems*, 32:8872–8882, 2019.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Chris M Gothard, Siowling Soh, Nosheen A Gothard, Bartlomiej Kowalczyk, Yanhu Wei, Bilge Baytekin, and Bartosz A Grzybowski. Rewiring chemistry: algorithmic discovery and experimental validation of one-pot reactions in the network of organic chemistry. *Angewandte Chemie International Edition*, 51(32): 7922–7927, 2012.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Peng Han, Peilin Zhao, Chan Lu, Junzhou Huang, Jiaxiang Wu, Shuo Shang, Bin Yao, and Xiangliang Zhang. Gnn-retro: Retrosynthetic planning with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4014–4021, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Pavel Karpov, Guillaume Godin, and Igor V Tetko. A transformer model for retrosynthesis. In *International Conference on Artificial Neural Networks*, pp. 817–830. Springer, 2019.

Matthew A Kayala and Pierre Baldi. Reactionpredictor: Prediction of complex chemical reactions at the mechanistic level using machine learning. *Journal of chemical information and modeling*, 52(10):2526–2540, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Greg Landrum et al. Rdkit: Open-source cheminformatics software. 2016.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015a.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015b.

Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS central science*, 3(10):1103–1113, 2017.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*, pp. 7192–7203. PMLR, 2021.

Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. Graphnvp: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*, 2019.

Kelong Mao, Xi Xiao, Tingyang Xu, Yu Rong, Junzhou Huang, and Peilin Zhao. Molecular graph enhanced transformer for retrosynthesis prediction. *Neurocomputing*, 457:193–202, 2021.

Gilles Marcou, João Aires de Sousa, Diogo ARS Latino, Aurélie de Luca, Dragos Horvath, V Rietsch, and Alexandre Varnek. Expert system for predicting reaction conditions: the michael reaction case. *Journal of chemical information and modeling*, 55(2):239–250, 2015.

Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*, 2019.

Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61 (7):3273–3284, 2021.

Nadine Schneider, Nikolaus Stiefl, and Gregory A Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of chemical information and modeling*, 56(12):2336–2346, 2016.

Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. In *International Conference on Machine Learning*, pp. 8818–8827. PMLR, 2020a.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020b.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020c.

Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. *Advances in Neural Information Processing Systems*, 34, 2021.

Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Towards understanding retrosynthesis by energy-based models. *Advances in Neural Information Processing Systems*, 34, 2021a.

Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Towards understanding retrosynthesis by energy-based models. *Advances in Neural Information Processing Systems*, 34, 2021b.

Igor V Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature communications*, 11(1):5575, 2020.

Amol Thakkar, Thierry Kogej, Jean-Louis Reymond, Ola Engkvist, and Esben Jannik Bjerrum. Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain. *Chemical science*, 11(1):154–168, 2020.

Christoph Tillmann and Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics*, 29(1):97–133, 2003.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Xiaorui Wang, Yuquan Li, Jiezhong Qiu, Guangyong Chen, Huanxiang Liu, Benben Liao, Chang-Yu Hsieh, and Xiaojun Yao. Retroprime: A diverse, plausible and transformer-based method for single-step retrosynthesis predictions. *Chemical Engineering Journal*, 420:129845, 2021.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Robert Burns Woodward. The total synthesis of vitamin b12. *Pure and Applied Chemistry*, 33(1):145–178, 1973.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, JINYU YANG, Yang Yu, and Junzhou Huang. Retroxpert: Decompose retrosynthesis prediction like a chemist. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11248–11258. Curran Associates, Inc., 2020.

Chaochao Yan, Peilin Zhao, Chan Lu, Yang Yu, and Junzhou Huang. Retrocomposer: Composing templates for template-based retrosynthesis prediction. *Biomolecules*, 12(9):1325, 2022.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 617–626, 2020.

Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling*, 60(1):47–55, 2019.