Node-Aligned Graph-to-Graph Generation for Retrosynthesis Prediction

Lin Yao
DP Technology, Ltd.
Beijing, China
yaol@dp.tech

Zhen Wang
DP Technology, Ltd.
Beijing, China
wangz@dp.tech

Wentao Guo
University of California, Davis
Davis, United States
wtguo@ucdavis.edu

Shang Xiang
DP Technology, Ltd.
Beijing, China
xiangs@dp.tech

Wentan Liu
DP Technology, Ltd.
Beijing, China
liuwt@dp.tech

Guolin Ke[†]
DP Technology, Ltd.
Beijing, China
kegl@dp.tech

Abstract

Single-step retrosynthesis is a crucial task in organic chemistry and drug design, requiring the identification of required reactants to synthesize a specific compound. With the advent of computer-aided synthesis planning, there is growing interest in using machine learning techniques to facilitate the process. Existing template-free machine learning-based models typically utilize transformer structures and represent molecules as 1D sequences. However, these methods often face challenges in fully leveraging the extensive topological information of the molecule and aligning atoms between the production and reactants, leading to results that are not as competitive as those of semi-template models. Our proposed method, Node-Aligned Graph-to-Graph (NAG2G), also serves as a transformer-based template-free model but utilizes 2D molecular graphs and 3D conformation information. Furthermore, our approach simplifies the incorporation of production-reactant atom mapping alignment by leveraging node alignment to determine a specific order for node generation and generating molecular graphs in an auto-regressive manner node-by-node. This method ensures that the node generation order coincides with the node order in the input graph, overcoming the difficulty of determining a specific node generation order in an auto-regressive manner. Our extensive benchmarking results demonstrate that the proposed NAG2G can outperform the previous state-of-the-art baselines in various metrics.

1 Introduction

The single-step retrosynthesis (SSR) [1] is an essential operation in organic chemistry and *de novo* drug design, involving the reverse synthesis of a target molecule in a single step. This process requires tracing back from the target molecule to determine the reactants needed for its synthesis. SSR serves as the basis for multi-step synthesis planning, which aims to identify a complete synthesis route in which

[†]corresponding author

Preprint. Under review.

the target molecule can be synthesized through a series of one-step reactions. Retrosynthesis demands a thorough understanding of organic chemistry principles and reaction mechanisms. Nevertheless, the advent of computer-aided synthesis planning has spurred increasing interest in employing machine learning techniques to expedite the retrosynthesis process, particularly for the template-free method.

Early machine learning-based SSR models [2, 3, 4, 5, 6] predominantly employ 1D sequences, such as SMILES (Simplified Molecular Input Line Entry System), for molecular representation. Consequently, existing models from Natural Language Processing (NLP), including RNN [7] and Transformer [8], can be readily utilized. Despite its simplicity, the 1D sequence-based model exhibits several limitations. Firstly, as indicated in numerous previous studies [9, 10, 11], the sequence disregards the extensive topological information depicted in molecular graphs. Secondly, the generation of valid molecular sequences is non-trivial due to the intricate rules involved. Lastly, incorporating production-reactant atom mapping alignment information, which significantly contributes to performance, poses a considerable challenge.

Several recent studies have proposed the adaptation of 2D molecular graphs for molecular representation in SSR models [12, 13, 14, 15]. These graph-based models offer improved encoding of molecules, and facilitate the integration of production-reactant atom mapping alignment information. However, generating a graph remains a significant challenge. Prior approaches have employed a repeated graph edit strategy [12] for graph generation, wherein a graph is iteratively modified by predicted edit actions (such as adding, removing, or updating nodes or edges) until no further alterations are required. This method necessitates the advance planning of edit action routes, and the iterative process of predicting actions based on modified graphs results in a significant computational burden.

To improve the efficiency of graph generation, we propose an auto-regressive approach that generates graphs node-by-node, drawing inspiration from language generation techniques. However, unlike language, graphs lack a natural one-dimensional sequence order, which poses a challenge in determining a specific order for node generation. In the context of SSR, this issue can be addressed by utilizing node alignment. Given the small difference between the input (the production molecule's graph) and the output (the reactants' graphs), we enforce the node generation order to match the node order in the input graph, as illustrated in Figure 1.

Building on this concept, we introduce a novel graph-based SSR template-free model, based on Transformer encoder-decoder architecture [8], denoted as Node-Aligned Graph-to-Graph (NAG2G). In NAG2G, the production molecule's graph initially serves as input for the encoder, and subsequently, the decoder generates reactant molecule graphs. For each node, the model generates the atom type, associated hydrogens and charges, and edges connecting to existing nodes. This generation process proceeds node-by-node in an auto-regressive manner, employing the aforementioned node alignment strategy to determine the node generation order. Besides graph structure information, to encode the sequential generation order, the 1D positional encodings are incorporated into the Transformer encoder and decoder. Moreover, data augmentation, like shuffling the node order, is employed to enhance the overall performance. Finally, we propose an efficient method for integrating dynamic graph-level features into self-attention during graph generation.

We conducted experiments using two widely recognized datasets, USPTO-50k [16] and USPTO-Full [17, 4]. The experimental results unequivocally illustrate that the proposed NAG2G substantially surpasses the performance of all prior baseline models. Additionally, we carried out an ablation study to scrutinize the impact of individual components within the proposed methodology. The findings offer compelling evidence of the effectiveness of the proposed NAG2G.

2 Related Work

The single-step retrosynthesis (SSR) [1] is a crucial process in organic chemistry that involves breaking down a target molecule into simpler precursor molecules.

There are several different approaches to modeling retrosynthesis, which can be broadly classified into three categories: template-based, semi-template-based, and template-free.

2.1 Template-free Methods

Template-free methods offer more flexibility than template-based and semi-template methods because they do not rely on pre-defined reaction templates and synthons. Instead, they use machine learning

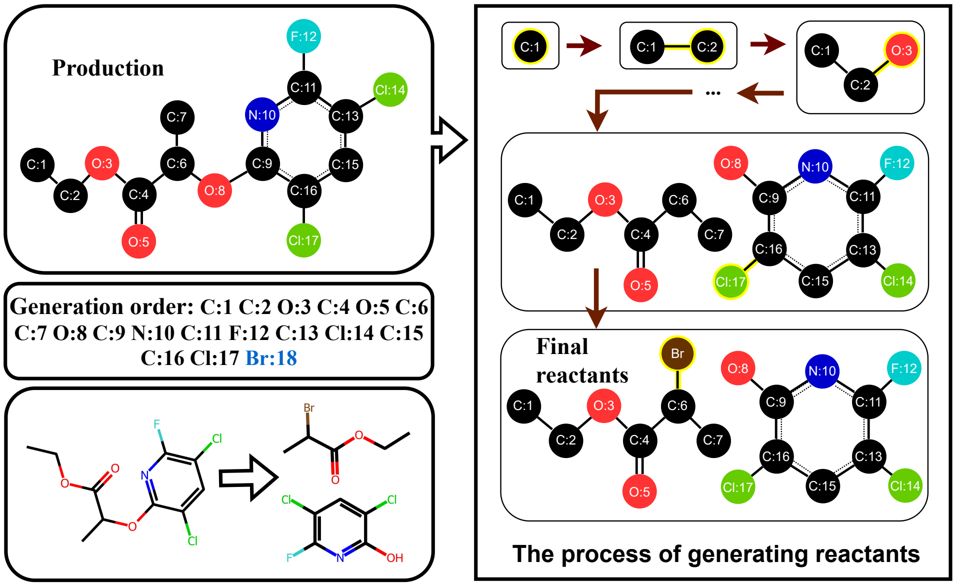

Production
C:1 C:2 O:3 C:4 O:5 C:6 C:7 O:8 C:9 N:10 C:11 F:12 C:13 Cl:14 C:15 C:16 Cl:17
Generation order: C:1 C:2 O:3 C:4 O:5 C:6 C:7 O:8 C:9 N:10 C:11 F:12 C:13 Cl:14 C:15 C:16 Cl:17 Br:18
F
Cl
N
O
O
Cl
Br
O
O
Cl
Cl
F
N
OH
C:1
C:1 C:2
C:1 C:2 O:3
...
Final reactants
Br
The process of generating reactants

Figure 1: The illustration of the node-aligned graph-to-graph generation. Given the high similarity between the input (the production molecule's graph) and the output (the reactants' graphs) graphs, we ensure that the node generation order corresponds to the node order in the input graph. This approach effectively addresses the challenge of determining a specific order for nodes in the auto-regressive graph generation process.

models to directly infer reactants from given productions, allowing for greater adaptability as the model can learn from any available data without being restricted by pre-existing templates. This approach is well-suited for predicting retrosynthesis for complex or novel chemical structures that may not be compatible with existing templates. However, the template-free approach can also present challenges, as models may struggle to learn the transformation rules between productions and reactants, as well as the validness of molecule representations, leading to the generation of invalid reactants. Additionally, predicting the correct reactants in situations where there are multiple possible solutions may result in duplicate, mixed, or incorrect outcomes. Therefore, sophisticated machine learning algorithms and data representations are needed to address these problems.

In template-free methods, a popular approach is to use SMILES strings as a compact and standardized way of representing chemical structures. Models such as SCROP [2], Tied Transformer [3], Aug. Transformer [4], RetroDCVAE [5], and Retroformer [6] all consider retrosynthesis as a SMILES sequence-to-sequence prediction problem and use Transformer architecture to address it. Despite the progress made in recent years, SMILES representations suffer from a lack of explicit topological information, and the utilization of production-reactant atom mapping alignments remains difficult. To address these limitations, Graph2SMILES [9] leverages a Graph Neural Network (GNN) to extract topological features, while GET [10] combines graph and SMILES encoders. Additionally, GTA [11] incorporates topological information into attention bias. Notably, models such as GTA [11] and Retroformer [6] aim to fully exploit production-reactant atom alignments to enhance performance. Nevertheless, the complete utilization of topological information and production-reactant atom mapping alignments remains a formidable challenge.

Alternative template-free methods that employ graph-based representations include MEGAN [12] and G2GT [18]. MEGAN [12] utilizes graph edit strategies to generate graphs by iteratively modifying the graph until no further changes are necessary. On the other hand, G2GT [18] aimed to generate graphs in an autoregressive manner, generating nodes sequentially. However, G2GT neglected to consider atom mapping alignment, resulting in suboptimal performance due to the challenges associated with determining a specific order for node generation. Furthermore, the results of G2GT are not directly comparable to those of other methods, as several additional techniques, such as self-training, were employed to achieve high benchmark scores.

2.2 Template-based and Semi-template-based Methods

For template-based methods, the first step is to prepare a template library in advance. These templates are typically based on known chemical reactions and can be either manually curated or generated automatically from reaction databases. Then, a model is trained to learn which templates in the library can be used for synthesizing the given productions. This dependence on the template library poses some challenges. Firstly, the library may not contain all possible reactions. Secondly, the relationship between the productions and templates may not be easily learned, especially when dealing with complex productions. Therefore, they may struggle with complex or novel chemical structures that do not fit well with existing templates. Prior methodologies, exemplified by RetroSim [19] and NeuralSym [20], have employed conventional molecular similarity metrics, such as fingerprints and Tanimoto similarity, to match templates and productions. Nevertheless, contemporary approaches, including GLN [17] and RetroComposer [21], have surpassed their predecessors due to their utilization of graph neural networks (GNN) as a central framework for more efficient data representations.

Since inferring reactants directly from productions is challenging, semi-template-based methods divide the retrosynthesis prediction process into two simpler stages. The first stage involves identifying synthons by detecting reactive bonds or atoms in the production. The synthons are not usually included in the datasets and need to be pre-calculated before training. The second stage involves completing the synthons into reactants using either leaving groups selection [22], SMILES generation [23, 24] or graph generation [13, 14, 15]. Error propagation might be easier in a two-stage model compared to end-to-end models, which means that errors in the first stage may still affect the results of the second stage. However, through the implementation of a two-stage methodology, researchers can attain supplementary information pertaining to synthons, and broaden the searching scope allowing for the exploration of various possibilities from production to synthons and from synthons to reactants, ultimately leading to superior overall performance compared to template-free models. Models including G2G [13], RetroXpert [23], RetroPrime [24], GraphRetro [22] and SemiRetro [25], fall in this categories. In the second stage, promising models such as G2Retro [14] and MARS [15] employ graph edit strategies [12] for graph generation. It is worth noting that while these graph edit strategies have demonstrated their effectiveness in semi-template-based approaches, they generally do not exhibit superiority in template-free methods [15].

3 Approach

3.1 Model Architecture

As shown in Figure 2, the NAG2G constitutes a Transformer-based encoder-decoder architecture, wherein the encoder's purpose is to learn the representation of target molecules. Several potent models, such as Graphormer [26] and Uni-Mol [27], already exist for effectively learning molecular representations. As our focus is not on proposing a new molecular representation model, we directly employ an existing model as the encoder. Specifically, we adopt the model backbone from [28] as the encoder for NAG2G, which is capable of learning molecular representations based on both 2D graph and 3D conformation. Furthermore, to encode the node order, an 1D positional encoding is additionally used. Formally, we denote the process of the encoder as:

$$\boldsymbol{O}^{\text{enc}} = f_{\text{enc}}(\boldsymbol{X}, \boldsymbol{P}^{\text{enc}}, \boldsymbol{E}, \boldsymbol{R}; \boldsymbol{\theta}^{\text{enc}}), \tag{1}$$

where $\boldsymbol{X}$ is the atom feature, $\boldsymbol{P}^{\text{enc}}$ is 1D positional encoding which is additionally added to atom embeddings, $\boldsymbol{E}$ is the edge feature of the 2D graph, $\boldsymbol{R}$ is the atom coordinate of the 3D conformation, $\boldsymbol{\theta}^{\text{enc}}$ is the learnable parameters of the encoder, and $\boldsymbol{O}^{\text{enc}}$ is the learned representation of the encoder.

The primary function of the decoder is to generate the graph node-by-node through an auto-regressive approach. At the i-th time step, the decoder receives three inputs:

1) The output from the encoder. In line with most encoder-decoder Transformer models, the encoder's output serves as the Key and Value in the cross-attention layer between the encoder and decoder. This process enables a more effective information exchange between the encoding and decoding stages, ultimately improving the overall performance of the model.

2) The decoder's outputs from previous time steps, ranging from 1 to $i-1$. This mirrors the approach of most auto-regressive generative models, which utilize outputs from earlier time steps as inputs. Additionally, the 1D positional encoding is added to the inputs as a standard practice in the majority of auto-regressive models. The inclusion of this encoding is crucial for NAG2G, since it facilitates the alignment of the atom order between the encoder inputs and the decoder outputs. During training,

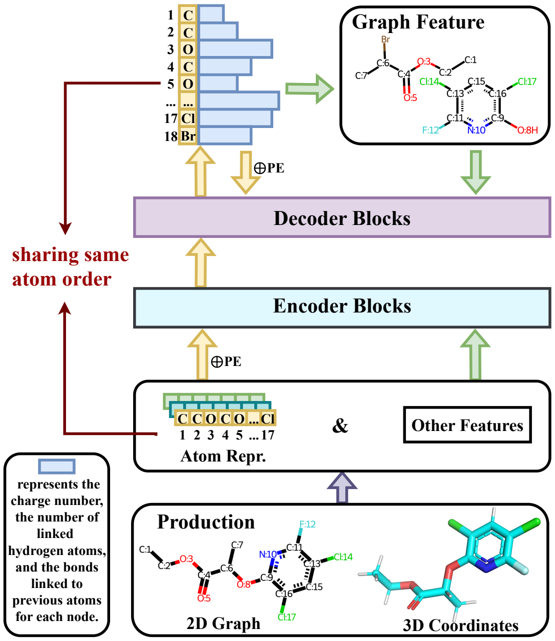


Figure 2: The network architecture of NAG2G.

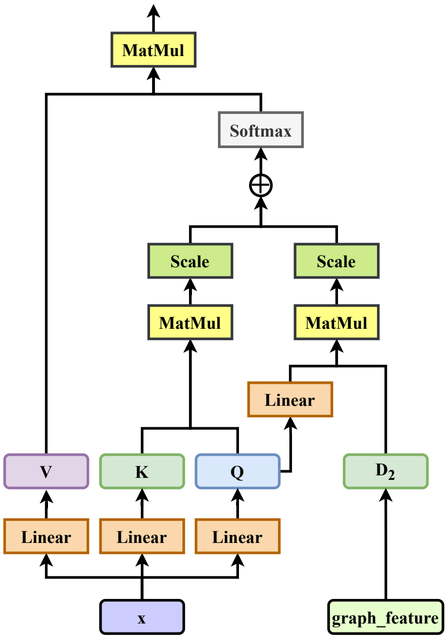


Figure 3: The design of self-attention layer for time-varying graph features.

the teacher-forcing technique is employed to enhance efficiency and stability, under the assumption that the outputs from previous time steps are 100% accurate.

3) The graph-level features, such as degrees and shortest paths, extracted from the existing predicted outcomes. Although these graph-level features hold the potential to enhance the generative performance of the decoder, incorporating them directly into the model presents an efficiency challenge, as the graph features vary across time steps. To address this issue, we propose an efficient method for integrating these graph-level features, with details provided in Section 3.3.

Employing the given inputs, the decoder generates a new node at the i-th time step, which comprises its atomic type and associated formal charge, the number of connected hydrogen atoms, and its edges linked to prior nodes. This procedure is inherently auto-regressive, meaning that the information for each node is produced sequentially. Formally, we can denote this process as

$$\begin{aligned}
t_i &= f_{\text{dec}}(\boldsymbol{P}^{\text{dec}}_{1:i}, \boldsymbol{N}_{1:i-1}, \boldsymbol{G}_{1:i-1}, \boldsymbol{O}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}), \\
c_i &= f_{\text{dec}}(t_i, \boldsymbol{P}^{\text{dec}}_{1:i}, \boldsymbol{N}_{1:i-1}, \boldsymbol{G}_{1:i-1}, \boldsymbol{O}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}), \\
h_i &= f_{\text{dec}}(c_i, t_i, \boldsymbol{P}^{\text{dec}}_{1:i}, \boldsymbol{N}_{1:i-1}, \boldsymbol{G}_{1:i-1}, \boldsymbol{O}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}), \\
e_{i,1} &= f_{\text{dec}}(h_i, c_i, t_i, \boldsymbol{P}^{\text{dec}}_{1:i}, \boldsymbol{N}_{1:i-1}, \boldsymbol{G}_{1:i-1}, \boldsymbol{O}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}), \\
&\cdots \\
e_{i,k} &= f_{\text{dec}}(e_{i,k-1}, ..., e_{i,1}, h_i, c_i, t_i, \boldsymbol{P}^{\text{dec}}_{1:i}, \boldsymbol{N}_{1:i-1}, \boldsymbol{G}_{1:i-1}, \boldsymbol{O}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}),
\end{aligned} \tag{2}$$

where $\boldsymbol{N}_{1:i-1}$ represents the set of nodes generated from the previous $i-1$ time steps, $\boldsymbol{P}^{\text{dec}}_{1:i}$ denotes the 1D positional encoding of the current i nodes, $\boldsymbol{G}_{1:i-1}$ represents the graph feature extracted from previous outputs, and $\boldsymbol{\theta}^{\text{dec}}$ denotes the learnable parameters of the decoder. The atomic type, associated formal charge, and the number of connected hydrogen atoms for the i-th node are represented by t_i, c_i, and h_i, respectively. The d-th edge, denoted by $e_{i,d} = (j, b)$, connects the i-th node and the j-th node with the bond type b. In this context, only the bonds within the molecules are considered as edges. Moreover, to establish a specific generative order for edges, the edges connected to the newly generated nodes (with larger 1D positions) are produced first. To minimize the number

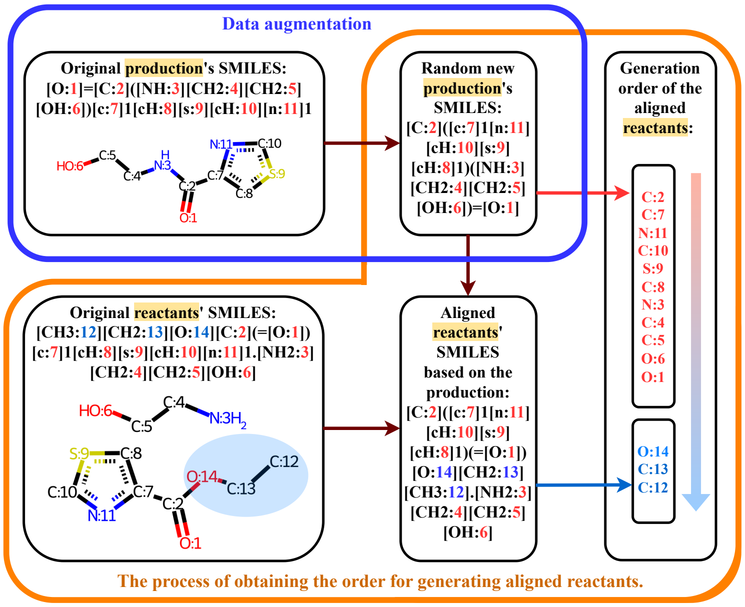

Data augmentation
Original production's SMILES:
[O:1]=[C:2]([NH:3][CH2:4][CH2:5]
[OH:6])[c:7]1[cH:8][s:9][cH:10][n:11]1
HO:6
C:5
C:4
H
N:3
C:2
O:1
C:7
N:11
C:10
S:9
C:8
Random new production's SMILES:
[C:2]([c:7]1[n:11]
[cH:10][s:9]
[cH:8]1)([NH:3]
[CH2:4][CH2:5]
[OH:6])=[O:1]
Generation order of the aligned reactants:
C:2
C:7
N:11
C:10
S:9
C:8
N:3
C:4
C:5
O:6
O:1
O:14
C:13
C:12
Original reactants' SMILES:
[CH3:12][CH2:13][O:14][C:2](=[O:1])
[c:7]1[cH:8][s:9][cH:10][n:11]1.[NH2:3]
[CH2:4][CH2:5][OH:6]
HO:6
C:5
C:4
N:3H2
S:9
C:8
C:10
N:11
C:7
C:2
O:1
O:14
C:13
C:12
Aligned reactants' SMILES based on the production:
[C:2]([c:7]1[n:11]
[cH:10][s:9]
[cH:8]1)(=[O:1])
[O:14][CH2:13]
[CH3:12].[NH2:3]
[CH2:4][CH2:5]
[OH:6]
The process of obtaining the order for generating aligned reactants.

Figure 4: An example to illustrate the process of data augmentation and production-reactant alignment. The red numbers indicate the atoms present in both the production and reactants, while the blue ones represent the atoms found only in the reactants.

of generative steps, the generation of c_i (and h_i) will be omitted if the node has zero formal charges (or no connected hydrogen atoms).

3.2 Node Alignment and Data Augmentation

The Transformer-based model exhibits permutation invariance for input nodes, necessitating the incorporation of 1D positional encodings to distinguish the sequential positions of these nodes. To specify an order, we employ the atom order in the SMILES sequence corresponding to a given production molecule as the 1D position in the encoder. Figure 4 shows the process of node alignment. The atom order of the decoder generations can be bifurcated into two parts. The first part pertains to the atoms that exist in both the productions and reactants, which are arranged in the same order as the production's SMILES. The second part comprises atoms that are present only in the reactants, which are placed at the end of the generations. By align the reactants' SMILES with that of the production based on RDKit [29], the atoms that are exclusively present in the reactants, are selected following the aligned reactants' SMILES orders and appended to the end of the decoder generations. Furthermore, during training, supplementary data augmentation techniques are applied to enhance the model's robustness. Specifically, as shown in Figure 4, RDKit is employed to randomly permute the production's SMILES, and the atom order in the permuted SMILES is used for the encoder. The reactants' SMILES are then aligned with the permuted SMILES. This data augmentation approach allows the model to be more resilient to variations in generation orders.

3.3 Efficient Time-Varying Graph-Level Features

With the implementation of teacher-forcing technology during training, data at various time steps are processed in parallel to enhance efficiency. The interaction between the current time step and previous time steps is addressed within the decoder's attention layer. To prevent potential leakage from future time steps, the attention matrix is masked using an upper triangular matrix. Formally, we denote this process as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_h}} + \boldsymbol{M}\right)\boldsymbol{V}, \tag{3}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n \times d_h}$ represent the query, key, and value matrices, respectively, and d_h is the dimension of one head, n is the number of time steps. $\boldsymbol{M}$ is an additive mask matrix that ensures that only the relevant information from the current and previous time steps is considered during the attention computation. For the sake of simplicity, we present the calculation for only one head. The multi-head attention process executes the above single-head calculation in parallel. During the calculation of one head, the computational complexity is $\mathcal{O}(n \times n \times d_h)$, and the peak memory consumption is $\mathcal{O}(n \times n)$.

As previously mentioned, graph-level features vary across time steps, and their direct utilization poses an efficiency challenge during model training. Specifically, to maintain the time-varying graph features, a matrix with shape $n \times n \times d_h$ is required*. These time-varying graph features are then employed as additive positional encodings. As a result, the attention layer can be represented as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{D}) = \text{softmax}\left(\frac{\boldsymbol{Q}(\boldsymbol{K}+\boldsymbol{D})^T}{\sqrt{d_h}} + \boldsymbol{M}\right)(\boldsymbol{V}+\boldsymbol{D}), \tag{4}$$

where $\boldsymbol{D} \in \mathbb{R}^{n \times n \times d_h}$ denotes the time-varying graph features, and the shape of $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ is reshaped to $n \times 1 \times d_h$ for broadcasting. In this process, although the computational complexity remains unchanged, the peak memory consumption increases to $\mathcal{O}(n \times n \times d_h)$. Considering that d_h is typical 32 or even larger, this significant increase in peak memory consumption is considered impractical for real-world applications.

To reduce the cost, we can first remove $\boldsymbol{D}$ added to $\boldsymbol{V}$. Then, observe that $\boldsymbol{Q}(\boldsymbol{K}+\boldsymbol{D})^T = \boldsymbol{Q}\boldsymbol{K}^T + \boldsymbol{Q}\boldsymbol{D}^T$, where the cost is bottlenecked at $\boldsymbol{Q}\boldsymbol{D}^T$. Thus, we can reduce the size of the last dimension for this computation. Combining these observations, we obtain:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{D_2}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_h}} + \frac{\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D_2}^T}{\sqrt{d_{h2}}} + \boldsymbol{M}\right)\boldsymbol{V}, \tag{5}$$

where $\boldsymbol{U} \in \mathbb{R}^{d_h \times d_{h2}}$ is employed to reduce the dimension of $\boldsymbol{Q}$, and $\boldsymbol{D}_2 \in \mathbb{R}^{n \times n \times d_{h2}}$ represents the time-varying graph features with a much smaller dimension d_{h2}. With this configuration, the peak memory is reduced to $\mathcal{O}(n \times n \times d_{h2})$. Figure 3 illustrates the design of a self-attention layer for time-varying graph features.

4 Experiment

4.1 Setting

Data We utilize two widely-accepted retrosynthesis benchmark datasets, USPTO-50k [16] and USPTO-Full [17, 4], as our benchmark datasets. USPTO-50k comprises 50,016 atom-mapped reactions, categorized into 10 reaction classes. We employ the same data split as in previous work [23], resulting in 40,008, 5,001, and 5,007 reactions for the training, validation, and test sets, respectively. Following previous studies, we evaluate model performance under two conditions: one with known reaction classes and another without.

The USPTO-Full dataset is a more extensive collection containing approximately 1 million atom-mapped reactions. In our study, we employed the filtered USPTO-Full dataset as described by Tetko et al. [4], instead of the original USPTO-Full dataset created by Dai et al. [17]. This filtered version eliminates incorrect reactions, leading to an approximate 4% reduction in the size of the training, validation, and test sets, which now comprise approximately 769,000, 96,000, and 96,000 reactions respectively. Consistent with previous works, we did not benchmark the results with reaction classes in USPTO-Full.

Model Training We established the model using a 6-layer encoder and a 6-layer decoder. The input embedding dimension was set to 768, and the number of attention heads was set to 24. We employed the Adam optimizer [30] with $(\beta_1, \beta_2) = (0.9, 0.999)$, and utilized linear warmup and decay with a peak learning rate of 2.5e-4. For training on the USPTO-50k dataset, we employed a total of 12,000 training steps, a batch size of 16, and the process took approximately 6 hours on a single NVIDIA A100 GPU. For training on the USPTO-Full dataset, we employed a total of 48,000 training steps, a batch size of 64, and the process took approximately 30 hours on eight NVIDIA A100 GPUs.

*Here, we consider node-wise graph features, such as node degrees. Pair-wise graph features, such as the shortest path, will consume significantly more memory.

Table 1: Top-k accuracy for retrosynthesis prediction on USPTO-50k. The best performance is in **bold**, and the best results in each method type are underlined. Models denoted by an asterisk ($*$) employed supplementary datasets for training or incorporated techniques to enhance the accuracy during inference. In order to maintain a fair comparison, we also present their results without the implementation of these additional techniques.

	Top-k Accuracy (%)							
	USPTO-50k							
	Reaction Class Known				**Reaction Class Unknown**			
Model	1	3	5	10	1	3	5	10
Template-Based								
RetroSim [19]	52.9	73.8	81.2	88.1	37.3	54.7	63.3	74.1
NeuralSym [20]	55.3	76.0	81.4	85.1	44.4	65.3	72.4	78.9
GLN [17]	64.2	79.1	85.2	90.0	52.5	69.0	75.6	83.7
MHNreact [33]	-	-	-	-	50.5	73.9	81.0	87.9
RetroComposer [21]	65.9	85.8	89.5	91.5	54.5	**77.2**	83.2	87.7
Semi-Template-Based								
G2G [13]	61.0	81.3	86.0	88.7	48.9	67.6	72.5	75.5
RetroXpert [23]	62.1	75.8	78.5	80.9	50.4	61.1	62.3	63.4
RetroPrime [24]	64.8	81.6	85.0	86.9	51.4	70.8	74.0	76.1
GraphRetro [22]	63.9	81.5	85.2	88.1	53.7	68.3	72.2	75.5
SemiRetro [25]	65.8	85.7	89.8	92.8	54.9	75.3	80.4	84.1
G2Retro [14]	63.6	83.6	88.4	91.5	54.1	74.1	81.2	86.7
MARS [15]	66.2	85.8	90.2	92.9	54.6	76.4	83.3	88.5
Template-Free								
LV-Transformer [34]	-	-	-	-	40.5	65.1	72.8	79.4
SCROP [2]	59.0	74.8	78.1	81.1	43.7	60.0	65.2	68.7
GET [10]	57.4	71.3	74.8	77.4	44.9	58.8	62.4	65.9
Tied Transformer [3]	-	-	-	-	47.1	67.1	73.1	76.3
MEGAN [12]	60.7	82.0	87.5	91.6	48.1	70.7	78.4	86.1
Aug. Transformer [4]	-	-	-	-	48.3	-	73.4	77.4
Aug. Transformer $*$ [4]	-	-	-	-	53.5	69.4	81	85.7
GTA [11]	-	-	-	-	51.1	67.6	74.8	81.6
Graph2SMILES [9]	-	-	-	-	52.9	66.5	70.0	72.9
RetroDCVAE [5]	-	-	-	-	53.1	68.1	71.6	74.3
DualTF [35]	65.7	81.9	84.7	85.9	53.6	70.7	74.6	77.0
Retroformer [6]	64.0	82.5	86.7	90.2	53.2	71.1	76.6	82.1
G2GT [18]	-	-	-	-	48.0	57.0	64.0	64.5
G2GT $*$ [18]	-	-	-	-	54.1	69.9	74.5	77.7
NAG2G (ours)	**67.2**	**86.4**	**90.5**	**93.8**	**55.1**	76.9	**83.4**	**89.9**

Model Inference & Evaluation During the inference process, we employ the widely-used beam search technique to generate top candidate predictions. Specifically, we set the beam size to 10, with a length penalty of 0 and a temperature of 1. It is important to note that data augmentation is not applied during inference. Additionally, RDChiral [31] is used to address the stereochemistry of reactants based on the stereochemistry of the productions.

To evaluate prediction accuracy, we adopt the method proposed by Liu et al. [32], which considers a prediction accurate only if all reactants in a reaction are correctly predicted. We measure the top-k accuracy of predictions, defined as the proportion of test cases in which the correct answer appears among the top k candidates of the beam search results.

4.2 Result

USPTO-50k We evaluate our proposed method, NAG2G, by comparing it with recent baseline approaches, including template-based, semi-template-based, and template-free methods. The results are summarized in Table 1. Based on these findings, we draw the following conclusions. (1)

Table 2: Top-k accuracy for retrosynthesis prediction on the USPTO-Full dataset. Models denoted by an asterisk ($*$) used supplementary datasets for training or incorporated techniques to improve accuracy during inference. For models denoted by a circle ($\circ$), the invalid reactions are excluded from the test set, following the setting of [4]. To align our methods with the previous baselines, we adopted the approach from [4], assuming that the methods failed on the removed test data, as evidenced by the results of our methods without a circle ($\circ$).

Model		**Top-k Accuracy (%)**			
Model Type	Methods	1	3	5	10
Template-Based	RetroSim [19]	32.8	-	-	56.1
	NeuralSym [20]	35.8	-	-	60.8
	GLN [17]	39.3	-	-	63.7
Semi-Template-Based	RetroPrime [24]	44.1	59.1	62.8	68.5
Template-Free	MEGAN [12]	33.6	-	-	63.9
	NAG2G (ours)	**47.7**	**62.0**	**66.6**	**71.0**
	Aug. Transformer $*\circ$ [4]	46.2	-	-	73.3
	G2GT $*\circ$ [11]	49.3		68.9	72.7
	NAG2G (ours)$\circ$	**49.7**	**64.6**	**69.3**	**74.0**

Table 3: Ablation study, on USPTO-50k with reaction class unknown.

Strategies			**Top-k Accuracy (%)**			
Node Alignment	Data Augmentation	Graph Features	1	3	5	10
✓	✓	✓	**55.1**	**76.9**	**83.4**	**89.9**
✓	✓	×	54.1	75.9	82.6	88.8
✓	×	✓	49.2	69.2	75.3	80.4
×	✓	✓	46.1	47.6	48.5	49.9
×	×	✓	40.3	54.9	58.9	62.6

Within the template-free category, NAG2G significantly outperforms all previous baselines across all metrics. Although some baselines employ additional data or techniques to enhance the benchmark results (denoted with $*$), the proposed method still outperforms them substantially. (2) Despite the additional use of pre-defined rules in template-based and semi-template-based methods, NAG2G still surpasses them, demonstrating a considerable improvement. Notably, this is the first instance of a template-free model outperforming both template-based and semi-template-based methods, as previous template-free baselines have been unable to achieve this goal.

USPTO-Full Table 2 presents the evaluation results of various models on the USPTO-Full dataset. From these results, it is evident that NAG2G significantly outperforms previous baselines in all metrics. Furthermore, we observe that template-based methods exhibit poor performance on the USPTO-Full dataset, despite demonstrating strong performance on the USPTO-50k dataset. This suggests that template-based methods, which heavily rely on pre-defined rules, face challenges in scaling up to larger datasets. On the other hand, template-free models possess greater advantages when applied to large-scale datasets.

In summary, the proposed template-free model, NAG2G, is highly effective. Not only does it outperform previous template-free models, but it also surpasses both previous template-based and semi-template-based methods.

4.3 Ablation Study

We conduct an ablation study on the USPTO-50k dataset, considering reactions with unknown classes. Specifically, we investigate the effectiveness of node alignment, data augmentation, and time-varying graph features. The results, as summarized in Table 3, lead to the following conclusions: (1) the incorporation of time-varying graph features considerably improves the final performance, and (2) both node alignment and the data augmentation that enhances it significantly contribute to the final performance. Overall, the ablation study demonstrates the necessity of employing these techniques in the proposed method.

5 Conclusion

In this paper, we have introduced a novel graph-based SSR template-free model, Node-Aligned Graph-to-Graph (NAG2G), which leverages Transformer encoder-decoder architecture to generate reactant molecule graphs in an auto-regressive manner. By utilizing node alignment strategy, we effectively address the challenge of determining node generation order in molecular graphs. Experimental results on widely recognized datasets, USPTO-50k and USPTO-Full, demonstrate that NAG2G significantly outperforms previous state-of-the-art baseline models. Ablation studies provide insights into the impact of individual components within our methodology, further validating the effectiveness of the proposed NAG2G model.

Our work represents a significant advancement in the application of machine learning techniques for single-step retrosynthesis, with the potential to greatly expedite the retrosynthesis process and contribute to the broader fields of organic chemistry and *de novo* drug design. Future research may explore additional enhancements to the proposed model, as well as the integration of our approach into multi-step synthesis planning for even more complex and diverse chemical synthesis tasks.

Limitations

The proposed graph-to-graph generation method is primarily designed for single-step retrosynthesis prediction, as there is a small difference between the input and the output graphs. However, for general graph-to-graph generation problems, particularly those with significant differences between inputs and outputs, the proposed method may not perform optimally.

References

[1] E J Corey and Xue-Min Cheng. *The logic of chemical synthesis*. John Wiley & Sons, Nashville, TN, June 1995.

[2] 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, 2020. PMID: 31825611.

[3] Eunji Kim, Dongseon Lee, Youngchun Kwon, Min Sik Park, and Youn-Suk Choi. Valid, plausible, and diverse retrosynthesis using tied two-way transformers with latent variables. *Journal of Chemical Information and Modeling*, 61(1):123–133, 2021. PMID: 33410697.

[4] 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 2020.

[5] Huarui He, Jie Wang, Yunfei Liu, and Feng Wu. Modeling diverse chemical reactions for single-step retrosynthesis via discrete latent variables, 2022.

[6] Yue Wan, Benben Liao, Chang-Yu Hsieh, and Shengyu Zhang. Retroformer: Pushing the limits of interpretable end-to-end retrosynthesis transformer, 2022.

[7] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[9] Zhengkai Tu and Connor W. Coley. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction, 2021.

[10] 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.

[11] Seung-Woo Seo, You Young Song, June Yong Yang, Seohui Bae, Hankook Lee, Jinwoo Shin, Sung Ju Hwang, and Eunho Yang. Gta: Graph truncated attention for retrosynthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):531–539, May 2021.

[12] Mikołaj Sacha, Mikołaj Błaż, Piotr Byrski, Paweł Dąbrowski-Tumański, Mikołaj Chromiński, Rafał Loska, Paweł Włodarczyk-Pruszyński, and Stanisław Jastrzębski. Molecule edit graph attention network: Modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021. PMID: 34251814.

[13] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 8818–8827, 2020.

[14] Ziqi Chen, Oluwatosin R. Ayinde, James R. Fuchs, Huan Sun, and Xia Ning. $\mathsf{G}^2\mathsf{Retro}$: Two-step graph generative models for retrosynthesis prediction, 2022.

[15] Jiahan Liu, Chaochao Yan, Yang Yu, Chan Lu, Junzhou Huang, Le Ou-Yang, and Peilin Zhao. Mars: A motif-based autoregressive model for retrosynthesis prediction, 2022.

[16] Nadine Schneider, Daniel M. Lowe, Roger A. Sayle, Michael A. Tarselli, and Gregory A. Landrum. Big data from pharmaceutical patents: A computational analysis of medicinal chemists' bread and butter. *Journal of Medicinal Chemistry*, 59(9):4385–4402, 2016. PMID: 27028220.

[17] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems*, pages 8870–8880, 2019.

[18] Zaiyun Lin, Shiqiu Yin, Lei Shi, Wenbiao Zhou, and Yingsheng John Zhang. G2gt: Retrosynthesis prediction with graph-to-graph attention neural network and self-training. *Journal of Chemical Information and Modeling*, 63(7):1894–1905, 2023. PMID: 36946514.

[19] 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.

[20] 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.

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

[22] Vignesh Ram Somnath, Charlotte Bunne, Connor W. Coley, Andreas Krause, and Regina Barzilay. Learning graph models for template-free retrosynthesis, 2020.

[23] Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, JINYU YANG, Yang Yu, and Junzhou Huang. Retroxpert: Decompose retrosynthesis prediction like a chemist. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, pages 11248–11258. Curran Associates, Inc., 2020.

[24] 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.

[25] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. Semiretro: Semi-template framework boosts deep retrosynthesis prediction, 2022.

[26] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation?, 2021.

[27] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Shuqi Lu, Zhifeng Gao, Di He, Linfeng Zhang, and Guolin Ke. Highly accurate quantum chemical property prediction with uni-mol+, 2023.

[29] Rdkit: Open-source cheminformatics, http://www.rdkit.org.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[31] 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.

[32] 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. PMID: 29104927.

[33] Philipp Seidl, Philipp Renz, Natalia Dyubankova, Paulo Neves, Jonas Verhoeven, Jörg K. Wegner, Marwin Segler, Sepp Hochreiter, and Günter Klambauer. Improving few- and zero-shot reaction template prediction using modern hopfield networks. *Journal of Chemical Information and Modeling*, 62(9):2111–2120, 2022. PMID: 35034452.

[34] Benson Chen, Tianxiao Shen, T. Jaakkola, and Regina Barzilay. Learning to make generalizable and diverse predictions for retrosynthesis. *ArXiv*, abs/1910.09688, 2019.

[35] Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Energy-based view of retrosynthesis, 2021.

[36] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *NeurIPS Workshop on Bayesian Deep Learning*, 2016.

[37] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer, 2018.

[38] Daniel Flam-Shepherd, Tony Wu, and Alan Aspuru-Guzik. Graph deconvolutional generation. *arXiv preprint arXiv:2002.07087*, 2020.

[39] Tengfei Ma, Jie Chen, and Cao Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 7113–7124, 2018.

[40] Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *International Conference on Machine Learning*, pages 2434–2444, 2019.

[41] Xiaojie Guo, Liang Zhao, Zhao Qin, Lingfei Wu, Amarda Shehu, and Yanfang Ye. Node-edge co-disentangled representation learning for attributed graph generation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[42] Jia Li, Jianwei Yu, Jiajin Li, Honglei Zhang, Kangfei Zhao, Yu Rong, Hong Cheng, and Junzhou Huang. Dirichlet graph variational autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.

[43] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. In *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[44] Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, and Pan Li. Conditional structure generation through graph variational generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 1340–1351, 2019.

[45] Shanchao Yang, Jing Liu, Kai Wu, and Mingming Li. Learn to generate time series conditioned graphs with generative adversarial nets. *arXiv preprint arXiv:2003.01436*, 2020.

[46] Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. In *Advances in Neural Information Processing Systems*, pages 13578–13588, 2019.

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

[48] Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):33, 2018.

[49] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pages 5708–5717, 2018.

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

[51] Davide Bacciu, Alessio Micheli, and Marco Podda. Graph generation by sequential edge prediction. In *Proc. ESANN*, 2019.

[52] Davide Bacciu, Alessio Micheli, and Marco Podda. Edge-based sequential graph generation with recurrent neural networks. *Neurocomputing*, 2020.

[53] Nikhil Goyal, Harsh Vardhan Jain, and Sayan Ranu. Graphgen: A scalable approach to domain-agnostic labeled graph generation. In *Proceedings of The Web Conference 2020*, pages 1253–1263, 2020.

[54] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems*, pages 4257–4267, 2019.

[55] Wataru Kawai, Yusuke Mukuta, and Tatsuya Harada. Scalable generative models for graphs with graph attention mechanism. *arXiv preprint arXiv:1906.01861*, 2019.

[56] Shuangfei Fan and Bert Huang. Attention-based graph evolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 436–447. Springer, 2020.

[57] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[58] Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In *International Conference on Machine Learning*, 2020.

Appendix for "Node-Aligned Graph-to-Graph Generation for Retrosynthesis Prediction"

A Related Work on Graph Generation

Graph generation can be broadly categorized into two classes: global models and sequential models. Global models generate the entire graph structure simultaneously, as opposed to sequentially. These models often employ a pairwise matrix, such as an adjacency matrix, to represent the graph and learn to generate the matrix. Commonly used global graph generation techniques include autoencoder-based models [36, 37, 38, 39, 40, 41, 42], Generative Adversarial Networks (GANs) [43, 44, 45], and flow-based generative models [46, 47].

In contrast, sequential models generate graphs incrementally. Many previous works fall into this category, such as MolRNN [48], GraphRNN [49], MolecularRNN [50], Bacciu et al.'s works [51, 52], GraphGen [53], MolMP [48], GRAN [54], GRAM [55], AGE [56], DeepGMG [57], and BiGG [58]. Sequential models are simple and effective, but suffer from the need to specify a generation order for nodes.

The proposed NAG2G leverages the node alignment strategy to determine node generation order, thus addressing the challenge in sequential generation. It is important to note that the proposed NAG2G model is primarily designed for single-step retrosynthesis prediction, as the input and output graphs exhibit small differences, allowing for node alignment. However, for general graph generation problems, the proposed method may not perform optimally.

B More Experimental Results

B.1 Ablation Study on Encoders

In NAG2G, we directly employ the existing model backbones from previous works as encoders. To investigate the performance improvement attributed to the encoders, we conduct an ablation study. Table 4 demonstrates that, apart from the default Uni-Mol+ [28], we also evaluate the performance of Graphormer [26]. The results indicate that the overall performance is comparable.

This observation suggests that the choice of encoders does not significantly impact the final performance of NAG2G.

Table 4: The performance with different encoders on USPTO-50k with reaction class unknown.

Encoder Type	**Top-k Accuracy (%)**			
	1	3	5	10
Default (Uni-Mol+) [28]	**55.1**	76.9	**83.4**	**89.9**
Graphormer [26]	54.3	**77.0**	**83.4**	89.0

B.2 Validity of the Generated Molecules

In order to accurately represent a molecule, it is crucial to take into account not only the types of atoms and bonds but also the formal charges of atoms and the number of attached hydrogen atoms. Although software tools such as RDKit have the potential to provide this information, they often fail to do so or deliver inaccurate data. To tackle this problem, NAG2G generates these features end-to-end, alongside atom types and bonds. The impact of incorporating this additional information is assessed through two ablation studies.

The first study evaluates the validity of the generated molecules, as depicted in Table 5. It can be observed that the model featuring additional atom characteristics enhances the validity, surpassing the previous baselines. The second study examines the overall performance, which is presented in Table 6. Owing to the increased validity, the model is capable of generating a greater number of valid results, thereby significantly improving the overall performance.

Table 5: Top-k validity of the generated molecules on USPTO-50k with reaction class unknown.

Model	**Top-k Validity (%)**			
	1	3	5	10
NAG2G (ours)	**99.7**	**98.6**	**97.1**	**92.9**
NAG2G w/o charge	89.9	90.2	86.1	75.9
NAG2G w/o hydrogen	89.6	88.4	87.6	83.4
NAG2G w/o charge or hydrogen	80.8	82.5	81.5	77.6
GET [10]	97.8	86.6	80.5	70.7
Graph2SMILES [9]	99.4	90.9	84.9	74.9
RetroPrime [24]	98.9	98.2	**97.1**	92.5

Table 6: The performance of NAG2G with/without generating additional atomic features, on USPTO-50k with reaction class unknown.

		Top-k Accuracy (%)			
Charges	**Hydrogen Atoms**	1	3	5	10
✓	✓	**55.1**	**76.9**	**83.4**	**89.9**
✓	×	49.0	68.9	74.7	80.8
×	✓	48.1	68.2	73.9	79.4
×	×	42.6	60.8	65.9	70.9

B.3 Accuracy of the Primary Reagent (MaxFrag)

The MaxFrag metric, previously proposed in the Aug. Transformer [4], is utilized to assess the accuracy of the primary (largest) reactant prediction. This metric was introduced due to its significance in the classical procedure, where focusing solely on the main compound transformations provides the minimal information necessary to derive an efficient retrosynthesis route. As demonstrated in Table 7, our proposed model, NAG2G, outperforms the Aug. Transformer, signifying an improved ability to accurately predict the primary reagent based on a given product.

Table 7: The top-k Accuracy of MaxFrag on USPTO-50k with reaction class unknown.

Model	**Top-k MaxFrag Accuarcy (%)**			
	1	2	5	10
Aug. Transformer [4]	58.0	73.4	84.8	89.1
NAG2G (ours)	**59.7**	**73.6**	**86.3**	**91.9**

B.4 Accuracy of Each Reaction Class

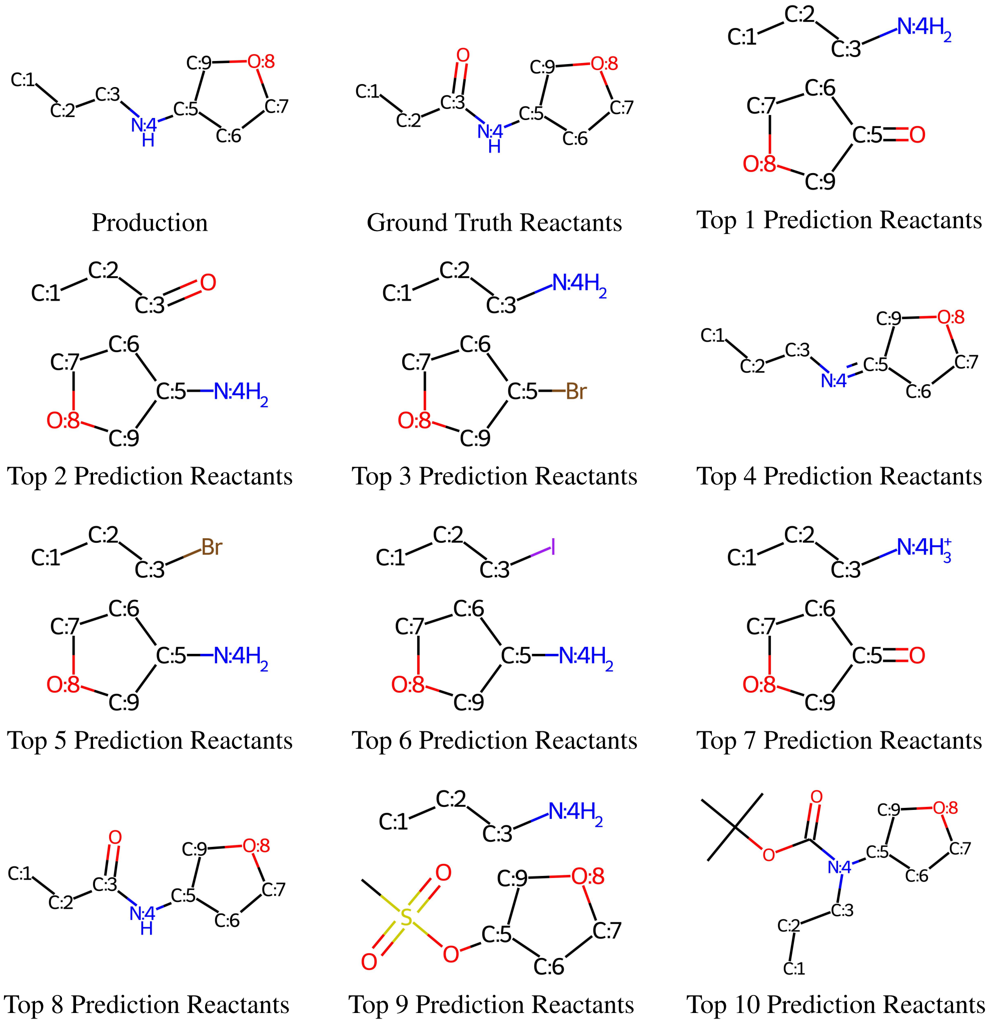
In Table 8, we exhibit the top-k accuracy results of NAG2G on the USPTO-50k dataset, wherein the reaction classes were undisclosed during the training process, and the results are stratified according to the ground truth reaction classes.

B.5 Case Study

We present three examples of NAG2G prediction results on the USPTO-50k test dataset without given reaction classes, as shown in Figure 5, Figure 6, and Figure 7. These examples demonstrate the capability of NAG2G to predict plausible reactants, even when an exact match with the ground truth reactants is achieved at a lower ranking. The model is able to generate multiple predictions that can result in the desired product, highlighting its potential for application in real-world scenarios.

Table 8: The top-k accuracy of each reaction class on USPTO-50k dataset, when trained with reaction class unknown.

Reaction Class	**Reaction Fraction(%)**	**Top-k Accuracy (%)**			
		1	3	5	10
heteroatom alkylation and arylation	30.3	54.4	75.5	82.3	90.5
acylation and related processes	23.8	67.3	87.5	91.1	94.5
deprotections	16.5	51.2	81.1	86.9	92.1
C-C bond formation	11.3	40.0	61.6	71.8	81.7
reductions	9.2	55.4	74.5	81.8	87.7
functional group interconversion	3.7	34.8	53.3	64.1	76.1
heterocycle formation	1.8	44.0	66.0	74.7	82.4
oxidations	1.6	69.5	81.7	91.5	96.3
protections	1.3	67.6	85.3	89.7	92.6
functional group addition	0.5	87.0	87.0	87.0	91.3


C:1 C:2 C:3 N:4 H C:5 C:6 C:7 O:8 C:9
C:1 C:2 C:3 O N:4 H C:5 C:6 C:7 O:8 C:9
C:1 C:2 C:3 N:4H2 C:7 C:6 C:5 O O:8 C:9
C:1 C:2 C:3 O C:7 C:6 C:5 N:4H2 O:8 C:9
C:1 C:2 C:3 N:4H2 C:7 C:6 C:5 Br O:8 C:9
C:1 C:2 C:3 N:4 C:5 C:9 O:8 C:7 C:6
C:1 C:2 C:3 Br C:7 C:6 C:5 N:4H2 O:8 C:9
C:1 C:2 C:3 I C:7 C:6 C:5 N:4H2 O:8 C:9
C:1 C:2 C:3 N:4H3+ C:7 C:6 C:5 O O:8 C:9
C:1 C:2 C:3 O N:4 H C:5 C:6 C:7 O:8 C:9
C:1 C:2 C:3 N:4H2 S O O O C:5 C:6 C:7 O:8 C:9
O O N:4 C:5 C:6 C:7 O:8 C:9 C:3 C:2 C:1

Production

Ground Truth Reactants

Top 1 Prediction Reactants

Top 2 Prediction Reactants

Top 3 Prediction Reactants

Top 4 Prediction Reactants

Top 5 Prediction Reactants

Top 6 Prediction Reactants

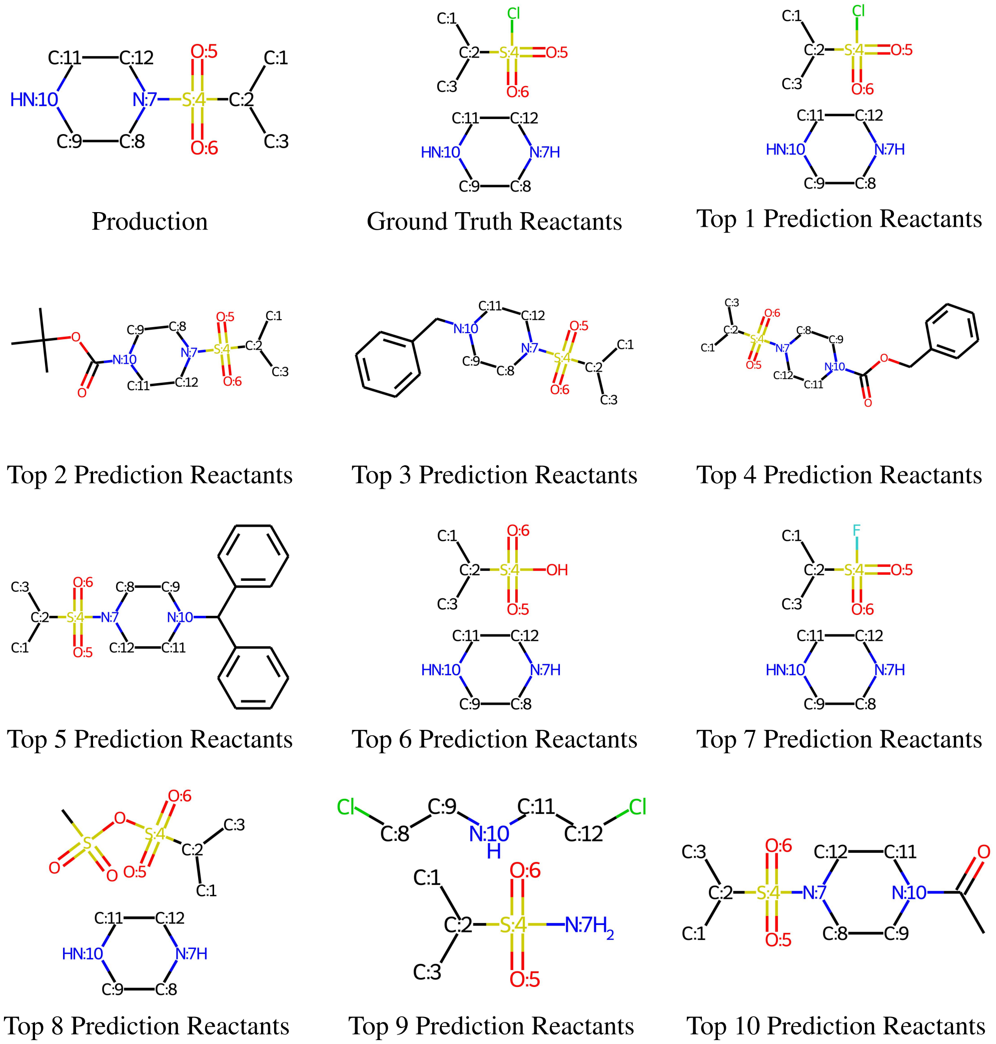
Top 7 Prediction Reactants

Top 8 Prediction Reactants

Top 9 Prediction Reactants

Top 10 Prediction Reactants

Figure 5: This is example 1 of NAG2G prediction on the USPTO-50k test dataset with the reaction class unknown. Although only the eighth predicted reactants precisely correspond to the ground truth reactants, all ten predicted reactions are chemically valid within reaction mechanisms. Specifically, the first, second, fourth, seventh, and eighth reactions can be classified as reduction reactions, while the third, fifth, sixth, and ninth reactions belong to the heteroatom alkylation and arylation reaction type. The tenth reaction is categorized as a deprotection reaction.



Production

Ground Truth Reactants

Top 1 Prediction Reactants

Top 2 Prediction Reactants

Top 3 Prediction Reactants

Top 4 Prediction Reactants

Top 5 Prediction Reactants

Top 6 Prediction Reactants

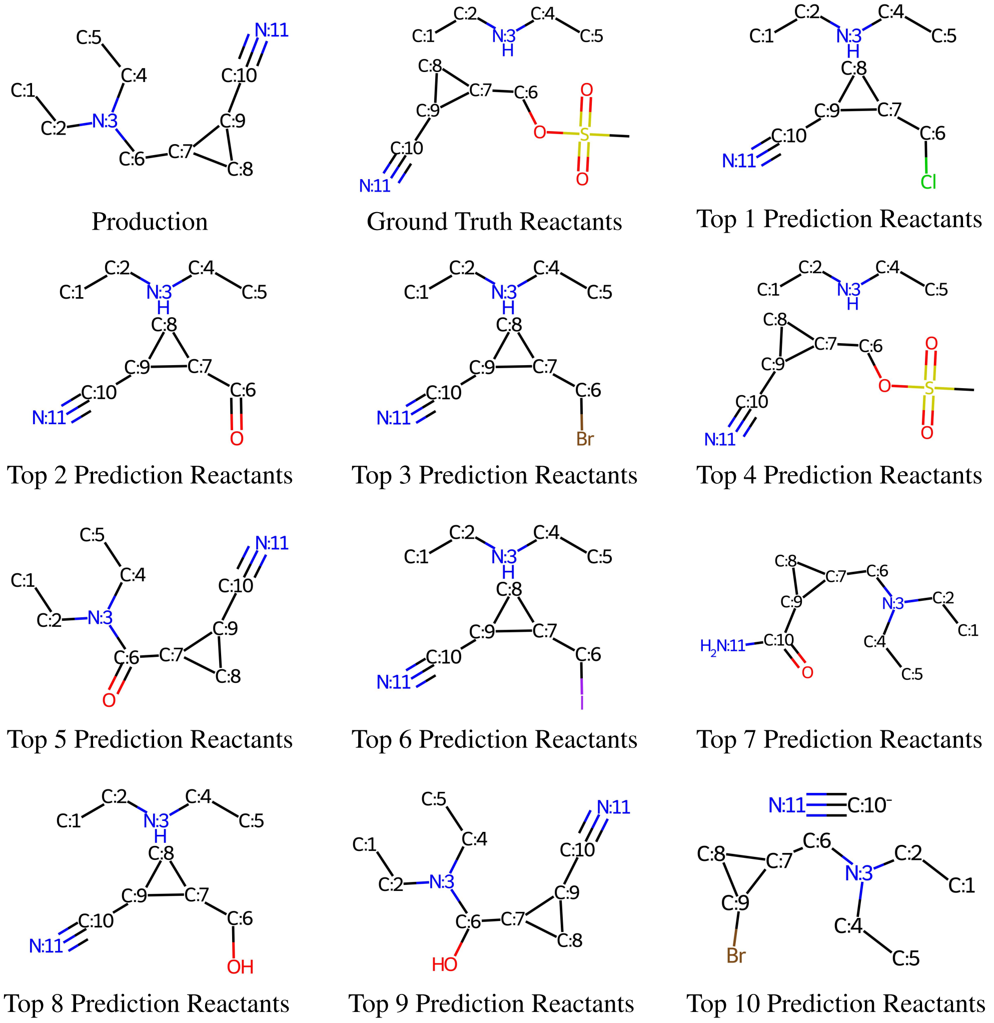
Top 7 Prediction Reactants

Top 8 Prediction Reactants

Top 9 Prediction Reactants

Top 10 Prediction Reactants

Figure 6: This is example 2 of NAG2G prediction on the USPTO-50k test dataset with the reaction class unknown. All the reactions are chemically valid in terms of mechanisms except for the sixth reaction. The first predicted reactants exact match the ground truth. The first, seventh, eighth, and ninth reactions fall under the category of heteroatom alkylation and arylation reaction, while the second, third, fourth, fifth and tenth are classified as deprotections reactions. It is worth noting that the sixth predicted reactants can be easily transformed to the ground truth by adding one extra step.



Production

Ground Truth Reactants

Top 1 Prediction Reactants

Top 2 Prediction Reactants

Top 3 Prediction Reactants

Top 4 Prediction Reactants

Top 5 Prediction Reactants

Top 6 Prediction Reactants

Top 7 Prediction Reactants

Top 8 Prediction Reactants

Top 9 Prediction Reactants

Top 10 Prediction Reactants

Figure 7: This is example 3 of NAG2G prediction on the USPTO-50k test dataset with the reaction class unknown. Aside from the eighth reaction, all the other reactions are chemically valid in terms of mechanisms. The fourth predicted reactants exactly match the ground truth. The first, third, fourth, and sixth reactions can be categorized as heteroatom alkylation and arylation reactions, while the second, fifth, and ninth reactions are classified as reduction reactions. The seventh reaction uniquely belongs to the functional group interconversion reaction category, and the tenth reaction is a C-C bond formation reaction. It is worth noting that the eighth prediction can be easily converted to the ground truth by adding just one extra step.