JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

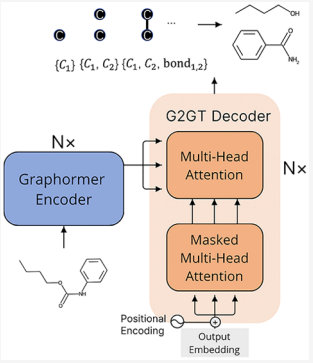
G2GT: Retrosynthesis Prediction with Graph-to-Graph Attention Neural Network and Self-Training

Zaiyun Lin,* Shiqiu Yin, Lei Shi, Wenbiao Zhou, and Yingsheng John Zhang

Cite This: J. Chem. Inf. Model. 2023, 63, 1894−1905 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Retrosynthesis prediction, the task of identifying reactant molecules that can be used to synthesize product molecules, is a fundamental challenge in organic chemistry and related fields. To address this challenge, we propose a novel graph-to-graph transformation model, G2GT. The model is built on the standard transformer structure and utilizes graph encoders and decoders. Additionally, we demonstrate the effectiveness of self-training, a data augmentation technique that utilizes unlabeled molecular data, in improving the performance of the model. To further enhance diversity, we propose a weak ensemble method, inspired by reaction-type labels and ensemble learning. This method incorporates beam search, nucleus sampling, and top-*k* sampling to improve inference diversity. A simple ranking algorithm is employed to retrieve the final top-10 results. We achieved new state-of-the-art results on both the USPTO-50K data set, with a top-1 accuracy of 54%, and the larger more challenging USPTO-Full data set, with a top-1 accuracy of 49.3% and competitive top-10 results. Our model can also be generalized to all other graph-to-graph transformation tasks. Data and code are available at https://github.com/Anonnoname/G2GT_2



INTRODUCTION

Retrosynthesis prediction represents a crucial challenge in the field of organic chemistry and its related disciplines. The task involves identifying an appropriate set of reactants for the synthesis of product molecules. In recent years, the development and application of various computational retrosynthesis tools have been employed to facilitate the design of synthetic routes for novel molecules. These methods can be broadly classified into template-based, template-free, and semi-template approaches, with the latter two categories primarily utilizing deep learning-based techniques.

Template-based methods[1] are rule-based approaches in which the target molecule is applied sequentially to all known reaction templates which are graph transformation rules. If the subgraph of the product molecule matches the product graph of the template, the template is said to be applicable to the molecule and may be used to convert it to a reactant set. Such approaches, which rely on expert knowledge, are unable to keep up with the increasing number of reported reactions. Some other approaches[2−4] employ machine learning to extract reaction templates automatically from reaction data sets. However, this technique requires atom mapping information, that is, a scheme for mapping atoms in reactants to generate atoms in products, which remains an unsolved problem.[5] Such tools rely on libraries of reaction templates and expert rules to function. Eventually, seemingly automatic techniques are still premised on the reaction templates. These techniques share the limitations of all template-based methods.

Semitemplate methods[6−12] decompose the retrosynthesis process into two subtasks: (1) predicting the reaction centers of the product and (2) splitting the product into synthons based on the reaction center and converting them into a complete molecule. The limitations of these methods are apparent: they rely on high-quality atom mapping reaction data sets to extract reaction centers; thus, the training sets of such methods are limited to small subsets of reactions and are not scalable for real-world applications.

Template-free methods,[13−17] in contrast, transform the one-step retrosynthesis prediction into a Seq2Seq translation task, in which the simplified molecular input-line entry system (SMILES)[18] of the product and the SMILES of the reactants are the "source language" and the "target language", respectively. Such methods can use all existing reaction data and generalize the reaction patterns that have not yet been discovered. Liu et al.[13] first adopted such a framework using a long short-term memory (LSTM) network and achieved performance comparable to that of template-based methods. Subsequently, the augmented transformer (AT),[14] which employs the transformer[19] and utilizes SMILES augmentation strategies, provided state-of-the-art results on both the USPTO-50k data set and the larger USPTO-Full data set,

Received: October 20, 2022
Published: March 22, 2023

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

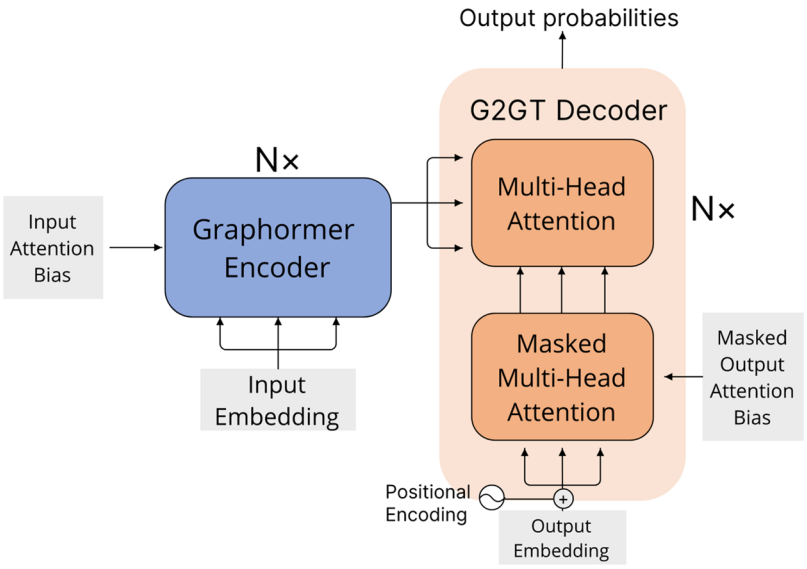


Figure 1. Overview of the Graph2Graph transformer (G2GT) model architecture.

which poses a challenging task for other methods because of the lack of reliable atom-mapping information.

Inspired by the success of template-free methods and recent developments in graph neural networks (GNNs),[20−24] as well as their state-of-the-art (SOTA) applications in molecular graph representation learning,[20,25,26] we formalize the retrosynthesis task as a Graph2Graph problem and propose the Graph2Graph transformer (G2GT) model. This model comprises Graphormer, proposed by Ying et al.,[20] as the graph feature encoder and a novel graph decoder. We adopt graph representation because molecules are naturally represented as graphs, with atoms as nodes and chemical bonds as edges, making them ideal for GNN models. In addition, unlike SMILES, graph models are not affected by atom order.

Our contributions can be summarized as follows:

1. We propose a novel general Graph2Graph model for the one-step retrosynthesis task. The proposed model inherits the benefits of template-free methods and replaces the sequence representation with graph representation. Both the decoder and encoder (Graphormer) are built upon the standard transformer structure to attain a global receptive field and parallelization capability.
2. We propose using self-training as a data-augmentation strategy, combining sampling and beam search results with frequency ranking to retrieve the top-10 results and a novel weak-ensembling method to increase the diversity. Our ablation study shows that these techniques significantly improve the performance of the model.
3. G2GT advances the SOTA for both USPTO-50k and USPTO-Full data sets and proves its robustness and generalization ability on Reaxys[27]—a data set from a different source.

THE G2GT FRAMEWORK

Preliminaries. The **Transformer** is a deep learning model that was introduced in the paper "Attention Is All You Need" by Vaswani et al. (2017).[19] It is a neural network architecture that is primarily used in natural language processing tasks, such as machine translation and language modeling. It is primarily used for natural language processing (NLP) tasks such as language translation, text summarization, and sentiment analysis. The model is based on the idea of self-attention, which allows the model to selectively focus on certain parts of the input while processing it.

The Transformer model uses an attention mechanism, which allows it to selectively focus on certain parts of the input data. The attention mechanism is used in both the encoder and decoder parts of the model. In the encoder, the attention mechanism is used to compute a representation of the input data that takes into account the entire input sequence. In the decoder, the attention mechanism is used to compute a representation of the output data that takes into account both the input sequence and the previously generated output.

Graphormer[20] is built upon the standard Transformer architecture and specifically tailored for graph representation learning tasks. In traditional transformer models, the input data are transformed into a sequence of tokens, which can be processed by the model. However, graphs do not have a natural order, which makes it difficult to use traditional transformer models for graph representation learning.

The authors' key insight in utilizing Transformer in the graph is the necessity of effectively encoding the structural information on a graph into the model. To this end, they propose several simple yet effective structural encoding methods to help Graphormer better model graph-structured data. These methods enable Graphormer to effectively capture the structural information on graphs, which is crucial for graph representation learning tasks. The G2GT's preliminaries, the Transformer and Graphormer, are stated in the Appendix.

G2GT Architecture. As discussed in the Introduction, we define the reaction-prediction problem as a graph-to-graph problem and propose the novel G2GT architecture, which is illustrated in Figure 1, to solve it.

The encoder module adopts Graphormer,[20] which encodes the input graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$, $n = |V|$ is the number of nodes, to a sequence of high-dimensional representations $z = (z_1, ..., z_n)$. Given z, the decoder generates an output graph sequence $(y_1, ..., y_m)$ one element at a time. For each newly generated step, the model calculates new centrality and spatial information based on the updated graph

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

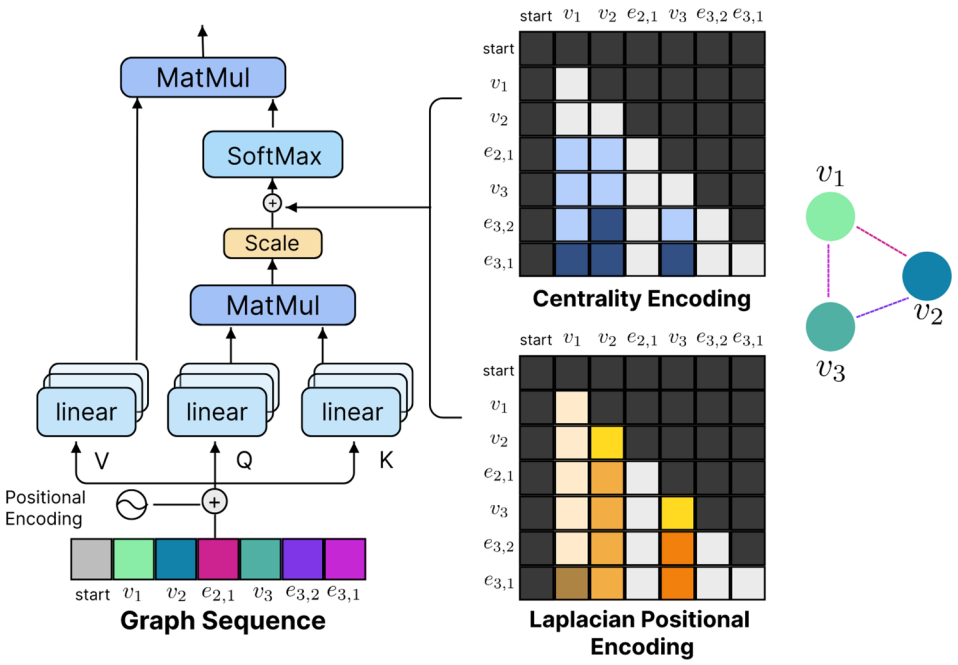


Figure 2. Illustration of G2GT's decoder, including Laplacian positional encoding and centrality encoding.

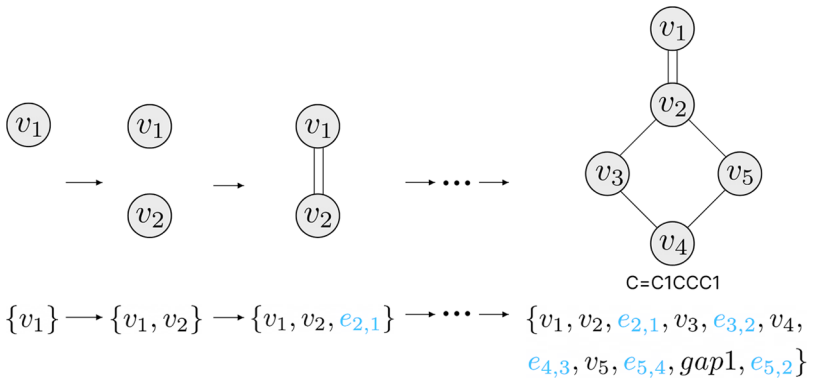


Figure 3. Example of the generation process using the proposed graph sequence representation.

state and feeds the information to the model as the decoders attention bias. The model then generates the next sequence based on the current graph state.

Graphormer Encoder. It is crucial to incorporate structural information on graphs into the model. We utilized Graphormer[20] as our encoder, as it is the first GNN model with a global receptive field that surpasses the performance of other popular GNN architectures, such as Message Passing Neural Networks (MPNN) and Graph Convolution Networks (GCN), in the recent Open Graph Benchmark Large-Scale Challenge[a]. This model effectively incorporates structural information on graphs into the Transformer model through the use of three simple yet effective design choices.

Decoder. This section presents several critical designs of the G2GT decoder, consisting of a novel graph sequence and inductive attention bias that are compatible with the masked attention module. Furthermore, we present an implementation of our proposed novel graph decoder. Finally, we discuss the advantages of our proposed model over previous sequence-based decoders. Figure 2 shows the decoder module.

Graph Representation. Once the input graph is encoded, and the input molecule representation vector, z, is given, the decoder infers the output graph sequences in an autoregressive manner for the given z. The output graph sequence is defined as follows: A molecule is denoted by an undirected graph $G = (V, E)$, where V is the set of atoms and E the set of bonds. Given a fixed atom ordering π, the molecule graph G is uniquely represented by its weighted adjacency matrix $A \in R^{(n\times n)}$, where n is the number of atoms. The weight of an edge is determined based on its attributes. For example, let $e_{(i,i+1)}$ represent a bond between atoms i and $i + 1$. If it is a single bond, $e_{(i,i+1)} = 1$, and if it is a double bond, $e_{(i,i+1)} = 2$.

We can obtain a graph sequence $Y = \{v_1, v_2, e_{2,1}, \cdots, v_n, e_{n,n-1}, \ldots, e_{n,1}\}$ by breaking A into rows. Each atom v_i is followed by the sequence of weighted edges $\{e_{i,i-1}, \cdots, e_{i,1}\}$ connecting the atom v_i to its previous atom v_{i-1}. To compress the length of the whole sequence, we remove each atoms trailing null edges; for example, $\{e_{i,i-1}, \cdots, e_{i,3}, null, null\} \rightarrow \{e_{i,i-1}, \cdots, e_{i,3}\}$. For nontrailing consecutively appearing null edges, we compress them to a single number *gapN* that represents the number of consecutive null edges; for example, $\{e_{i,i-1}, \cdots, null, null, e_{i,j}\} \rightarrow \{e_{i,i-1}, \cdots, gap2, e_{i,j}\}$. An illustrated example is shown in Figure 3.

Loss Function. Each generation step is a classification problem, where the model predicts the type of the token in the

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

graph sequence. The total number of classes is number of atoms × types of chirality + number of bonds. We use the cross-entropy loss to train our model. This loss function compares the predicted output from the model against the actual output and calculates the difference between them. The result is then used to update the model's parameters to minimize the difference and improve the model's accuracy.

Centrality Encoding (Decoder). Centrality encoding can be directly added to the node input in the encoder stage. However, because of the information leakage problem, the final centrality information cannot be directly added to the graph sequence input during the decoding stage. When the decoder autoregressively generates the following sequence, it must be prevented from knowing future information. Furthermore, as the graph grows, a nodes centrality may increase if it connects to a new node. Therefore, to address this issue, we use attention bias to hold the centrality information on all time steps.

Specifically, for any graph G, we precalculate a centrality matrix $E \in \mathbb{R}^{y\times y\times d}$, where y is the graph sequence length and total generation steps, and d is the number of heads in the multihead attention. While degree is only defined for nodes, in practice the model still requires a centrality input for non-node tokens. Hence, we set the input of the non-node token to 0. Denoting $e_{i,j}$, with dimension d as the (i, j)-element of E, g_i^j as the ith element in the graph sequence at time step j, and b as the learnable embedding indexed by the degree of the node $deg(g_i^j)$, we have

$$e_{i,j} = \begin{cases} b_{deg(g_i^j)} & \text{if } g_i^j \text{ is a node} \\ 0 & \text{if } g_i^j \text{ is not a node} \end{cases} \quad (1)$$

See Figure 4 for an example.

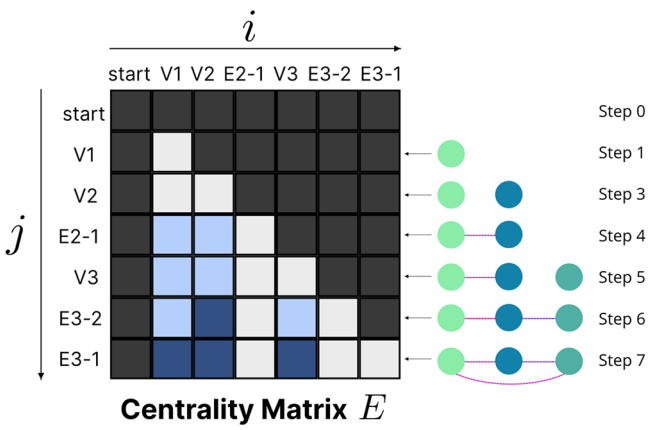


Figure 4. Example of the centrality matrix given each step's corresponding graph, where i is the graph sequence index and j the time step index. The color of the matrix represents the degree of the node. Black = undefined, gray = 0, light blue = 1, and dark blue = 2.

Denoting $A_{i,j}$ as the (i, j) element of the query key product matrix A of the decoder and W_q and W_k as the weight matrices used in self-attention to transform the input vectors into query, key and value vectors, we have

$$A_{ij} = \frac{(z_i W_q)(z_j W_k)^T}{\sqrt{d}} + e_{i,j} \quad (2)$$

Laplacian Positional Encoding (LPE). Traditional methods such as that by Liu et al.[13] represent a graph as a sequence. However, the sequence alone cannot fully capture the structural information on a graph. To preserve this information, previous studies[21,24] used the eigenfunctions of their Laplacian for positional encodings.

Instead of adding the LPE directly to the node input, we use the attention bias to hold the LPE, similar to how the centrality encoding is held. We follow an encoding method similar to that in ref 23. We build an embedding matrix of size $2 \times m$ by concatenating the m-lowest eigenvalues with their associated eigenvectors. The maximum number of eigenvectors to compute is specified by hyperparameter m. We add masked padding for graphs when m is greater than the number of nodes in the graph. Then, on the dimension of size 2, which is the eigenvector and eigenvalue pair, a linear layer is applied to construct new embeddings of size k. In this case, $k = d =$ number of heads. Finally, the sequence is reduced to a fixed k-dimensional node embedding via sum pooling. For any graph G, we precalculate an LPE matrix $L \in \mathbb{R}^{n\times n\times d}$. Denoting $l_{i,j}$ as the (i, j)-element of L and lpe as the node-wise LPE, we have

$$l_{i,j} = \begin{cases} lpe_{g_i^j} & \text{if } g_i^j \text{ is a node} \\ 0 & \text{if } g_i^j \text{ is not a node} \end{cases} \quad (3)$$

Then, we modify eq 2 further with the following equation:

$$A_{ij} = \frac{(z_i W_q)(z_j W_k)^T}{\sqrt{d}} + e_{i,j} + l_{i,j} \quad (4)$$

This approach directly addresses the limitations of the previous methods.[21,24] As described in ref 23, this is achieved by normalizing the eigenvectors, matching them to their eigenvalues, and using the number of eigenvectors as a variable. Furthermore, the model can linearly combine or ignore portions of repeating eigenvalues. In addition, as in other studies,[21,22] to increase the sign ambiguity invariance, we randomly reverse the sign of the precomputed eigenvectors during training. Figure 5 illustrates the proposed LPE module.

Positional Encoding (PE). In the decoder, we add positional encodings to allow the network to learn the generation order of the graph sequence. We use the sine and cosine functions of different frequencies as the positional encodings.

DATA AUGMENTATION AND DECODING STRATEGY

Self-Training. Our graph-to-graph model encoder is graph permutation invariant. Therefore, we cannot use the SMILES augmentation schemes proposed in previous works, such as that by Tetko et al.[14] Instead, we employ an approach called the self-training method, which is similar to AlphaFold,[28] to enhance our models generalizability. We first train a model using the training set; then, the model is used to predict the reactions on the external molecule set and select the high-confidence reactions to add to the training set to retrain the model.

Diversity. Diversity is a major concern in retrosynthesis prediction. A product can undergo different reactions and can be synthesized by a diverse set of alternative reactants. The variation in the reactants leaving groups is another case of retrosynthesis diversity. An example is shown in Figure 6.

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

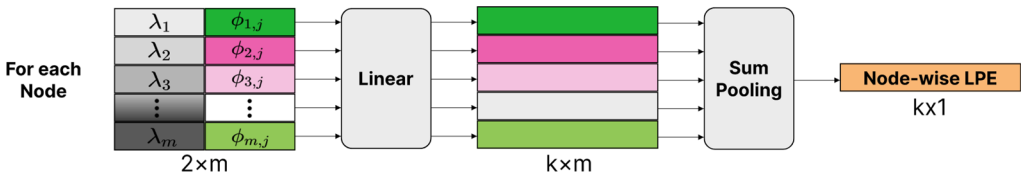


Figure 5. Laplacian positional encoding. $\phi_{i,j}$ denotes the eigenvector of node j paired to the ith lowest eigenvalue. m is the hyperparameter for the maximum number of eigenvalues and eigenvectors, and k is the dimension of the LPE.

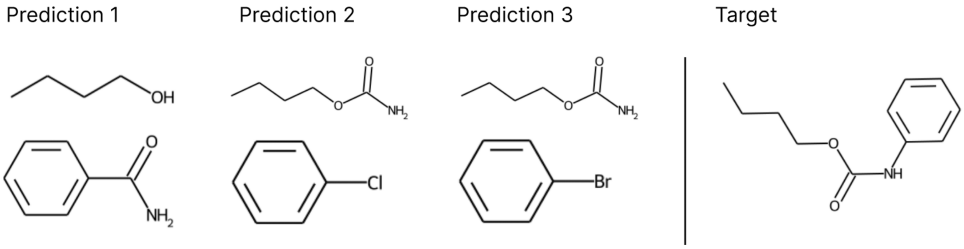


Figure 6. On the right is the input target molecule, and on the left are three possible reactant predictions. The first prediction is a Hofmann rearrangement reaction, whereas the second and third predictions are both Buchwald coupling reactions. The only difference between the latter two is the functional group (Cl vs Br).

Therefore, we designed a combination of strategies to increase prediction diversity.

Beam Search, Top-p, and Top-k Sampling. During inference, we use the top-p sampling[29] (aka nucleus sampling) and top-k sampling as our decoding methods. The top-p sampling selects tokens from the smallest accessible set whose cumulative probability exceeds probability p. Consequently, the size of the collection of tokens may change dynamically depending on the probability distribution of the next token. Given a distribution $P(x|x_{1:i-1})$, we define its top-p token $V(p) \subset V$ as the smallest set, such that

$$\sum_{x \in V(p)} P(x|x_{1:i-1}) \geq p \tag{5}$$

The idea behind top-k sampling is to only consider the top-k most likely tokens at each time step of the generation process, instead of considering all possible tokens. This greatly reduces the number of possibilities that the model needs to explore, which in turn improves the efficiency of the sampling process. For a more concrete definition, see the Appendix

Temperature is also used to adjust the probability. We use a high temperature to change the distribution to a softer one to increase the diversity.

Frequency Ranking. To output the top-n results, we can sample m times, $m > n$. After obtaining m samples, we use the occurrence frequency as the ranking index to retrieve the top-n results. We believe that the frequency of the predicted reactants indicates confidence in the model prediction. Thus, when counting the occurrence, we additionally use predictions decoded by a beam search as complementary samples.

Weak Ensemble. To further increase the diversity of predicted reactants, we explored the influence of the training set's reaction class distribution on the predicted reactants. As shown in Figure 6, a product molecule can contain multiple plausible reactants. Due to the distribution of different reaction classes in the training set, models can be biased toward certain reaction classes.

To address this issue, we propose a novel method inspired by ensemble learning. Specifically, we randomly split the training set into n sets and then concatenate a special node that holds tag ID information to the input of the product molecule. We jointly train all sets together, which exposes the model to different distributions of reaction classes and helps to reduce the bias toward certain reaction classes. At the inference stage, we attach different tag IDs to a product molecule and sample from these variants of the same product. This way, the model can generate different plausible reactants for the same product molecule, thus increasing the diversity of the predicted reactants. To help gain an understanding of how ensemble methods work, we can consider a simple weak ensemble that jointly trains a model with two subsets of the training data set. By attaching each tag ID to the product molecule in a standalone manner, we can make predictions that differ between the two tag IDs. In some cases, one tag ID may predict reactant A while the other predicts reactant B, resulting in increased diversity. Our ablation experiments (Table 4) shows that this technique increases the top-10 accuracy.

EXPERIMENTAL EVALUATION

Settings. We used the same model settings for all the experiments: **G2GT**$(L = 8 \times 2, d = 768)$. Both the number of attention heads and the dimensions of our attention bias modules were set to 24. The maximum number of eigenvectors was set to 30 due to the GPU memory limitation. We used AdamW as the optimizer, with ϵ = 1e-8 and β_1, β_2 = 0.9, 0.999. The peak learning rate was set to 2.5e-4, and the end learning rate was set to 1e-6, followed by a quadratic decay learning rate scheduler. The total number of training steps for USPTO-50K was 120k and for USPTO-Full was 400k. The batch size was set to 10 due to the GPU memory limitation, and we followed the practice recommended by ref 30 to gradually increase the batch size using gradient accumulation. During the inference stage, we used top-p and top-k sampling. After several rounds of parameter tuning on the small validation set, k was set to 5, p set to 0.75, and the temperature set to 6.5. For each molecule, we sampled 400 times. With regard to the weak ensemble, we assigned 20 tags for USPTO-50k and 50 tags for USPTO-Full. We chose these numbers intuitively based on the fact that all reactions in USPTO-50k can be separated into 10 reaction classes. All models were trained on eight NVIDIA V100 32 GB GPUs. We filtered invalid SMILES during the

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

inference process and did not include them in the calculation of the top-*k* accuracy. Table 1 shows the validity rate of generated molecules before filtering. It took approximately 20 h to train on USPTO-50K and 3 days to train on USPTO-Full.

Table 1. Validity Rate of Generated Molecules before Filtering

	validity rate (%)
Top-1	0.46
Top-3	2.56
Top-5	4.49
Top-10	27.10

Data and Software Availability. We mainly used the patent mining work of Lowe,[31] USPTO-50k, and USPTO-Full as our benchmark data sets for comparison with previous work. In addition, we used a nonpublic data set extracted from the Reaxys database to compare its performance with that of a previous end-to-end transformer model.

Data and code are available at https://github.com/Anonnoname/G2GT_2.

For **USPTO-50k**,[13] we used a training set filtered from the USPTO database containing 50k reactions classified into 10 reaction types. We split it into 40k, 5k, and 5k reactions for the training, validation, and test sets, as proposed by Liu et al.[13]

For **USPTO-Full**,[14,32] the original USPTO-Full was created by Dai et al.[32] Reactions with multiple products were duplicated into multiple products with one product each. They also removed duplications in the reactions and those with incorrect mapping. We used a further filtered version developed by Tetko et al.,[14] which eliminates incorrect reactions, such as those with no products or only single ions as reactants. This filtering reduced the size of the train/valid/test sets by an average of 4% to 769/96/96k.

For **Reaxys**,[33] we randomly sampled 5k cleaned reactions from the Reaxys database as an additional test set.

Baselines and Evaluation Metric. The baselines consist of template-based, semi-template, and template-free methods. All results were obtained from the original report using the same experimental settings.

According to the standard evaluation method used by Liu et al.,[13] the prediction is considered correct if and only if all reactants of a reaction are correctly predicted. We used the top-*k* accuracy as the evaluation metric, which is commonly used in the literature. Finally, to compare with the ground truth, we used the canonical SMILES generated by RDKit.

Main Results. Tables 2 and 3 summarize the performance of G2GT compared to the other methods. We evaluated only the reaction class in unknown settings. We achieved a new SOTA on both USPTO-50k and Full top-1 accuracy and a competitive top-10 accuracy on USPTO-Full. We are primarily interested in USPTO-Full because it is a more realistic data set in which no reliable atom-mapping information exists and contains more diverse and complex reactions with no guarantees of matching templates. We achieved a more significant improvement on USPTO-Full when compared with the results obtained on USPTO-50k. This demonstrates that when confronted with larger and noisier data, the G2GT framework can better generalize the underlying features of the data. Furthermore, despite using different methodologies, this study achieved similar results to many previous studies on USPTO-50k, i.e., approximately 53% top-1 accuracy.[8,10,14,35] These numbers may conceal the potential value of the approaches used. Hence, we argue that USPTO-50k is no longer a good benchmark for the retrosynthesis problem because it does not adequately reflect recent advances in this

Table 3. Top-*k* Accuracy on USPTO-Full Data Set

		Top-*k* accuracy (%)			
		k =			
Method Type	Methods	1	2	5	10
Template-based	GLN[32]	39.3			63.7
	Neuralsym[34]	35.8			60.8
Semi-template	RetroPrime[8]	44.1		62.8	68.5
Template-free	MEGAN[16]	33.6			63.9
	AT[14]	46.2	57.2		**73.3**
	G2GT (this work)	**49.3**	**60.0**	**68.9**	72.7

Table 2. Top-*k* Accuracy on USPTO-50k Data Set

		Top-*k* accuracy (%)			
		k =			
Method Type	Methods	1	3	5	10
Template-based	GLN[32]	**52.5**	69	75.6	83.7
	Neuralsym[34]	44.4	65.3	72.4	78.9
	MHNreact[4]	50.5 ± 0.3	**73.9 ± 0.3**	**81.0 ± 0.1**	**87.9 ± 0.1**
Semi-template	GraphRetro[7]	53.7	68.3	72.2	75.5
	SemiRetro[11]	**54.9**	75.3	80.4	84.1
	LocalRetro[12]	53.4	**77.5**	**85.9**	**92.4**
	RetroPrime[8]	51.4	70.8	74.0	76.1
	G2Gs[9]	48.9	67.6	72.5	75.5
Template-free	SCROP[15]	43.7	60.0	65.2	68.7
	AT[14]	53.2		**80.5**	85.2
	MEGAN[16]	48.1	70.7	78.4	**86.1**
	G2GT (this work)	**54.1**	69.9	74.5	77.7

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

Table 4. Ablation Study

	Top-k accuracy (%)			
	k =			
	1	2	5	10
Transfromer[36]	42.7	52.5	–	69.8
G2GT with beam search	48	57	64	64.5
Self-training with beam search	52.6	63.1	69.8	69.8
Self-training with sampling + frequency ranking (FR)	53.6	64.7	73.5	76.9
Self-training + weak ensembling with sampling + FR	54.1	65.2	74.5	77.7

Table 5. Expert Evaluation Results and Exact-Match Accuracy Belonging to the Given Class

Reaction class	Ours: correct num/num of reaction assessed by expert	Ours: exact-match accuracy (num/total num)	AT[14] exact-match accuracy (num/total num)
Condensation	6/9	0.63(1730/2735)	0.58(1581/2735)
Arylation	11/11	0.47(407/868)	0.40(348/868)
Suzuki	11/13	0.49(270/548)	0.34(186/548)
Reductive	9/11	0.59(130/219)	0.51(112/219)
Rearrangement	18/19	0.18(25/137)	0.09(12/137)
Total	55/63	0.57(2562/4507)	0.50(2239/4507)

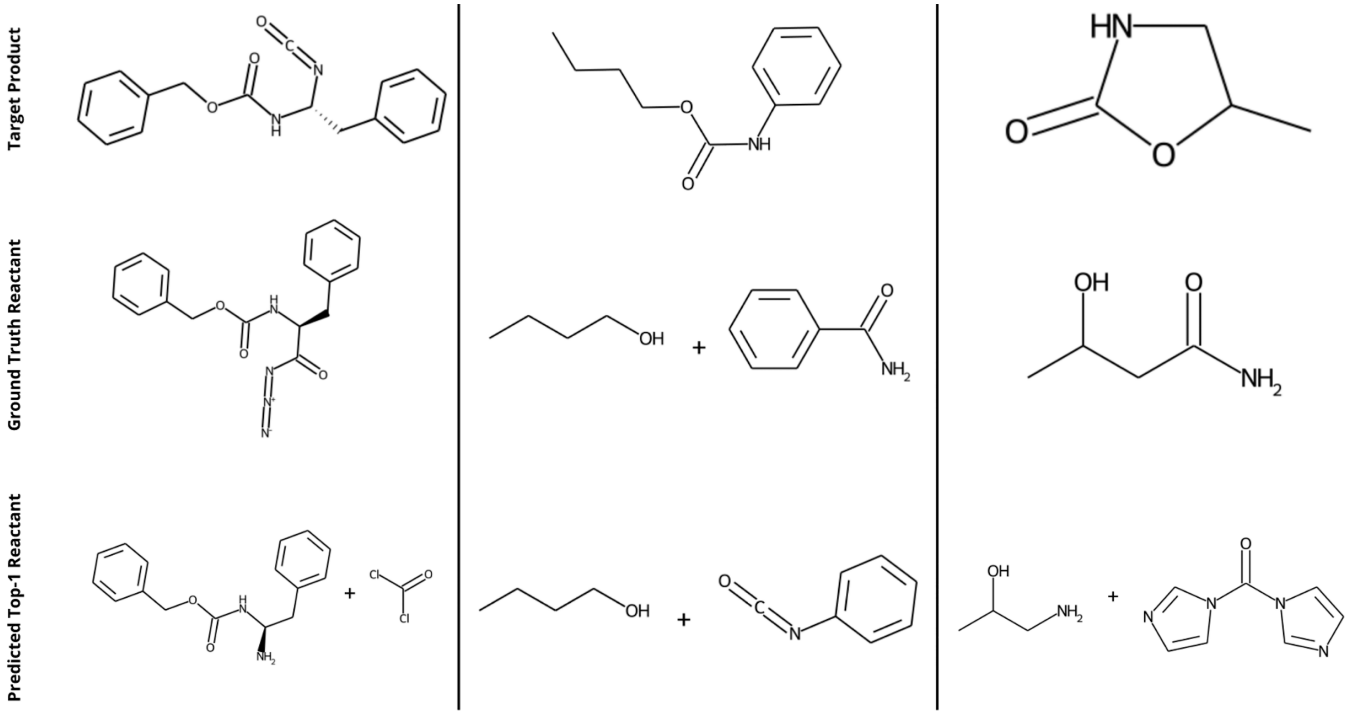


Figure 7. Cases that the model predicted as valid reactants but did not match the ground truth reactants.

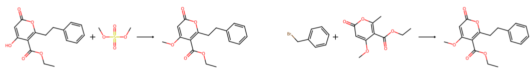
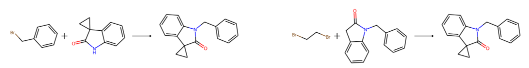
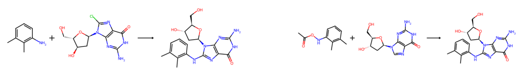
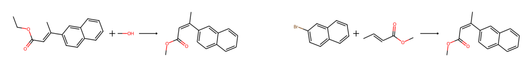
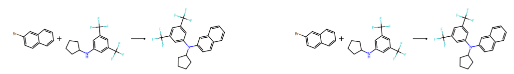
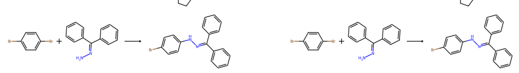
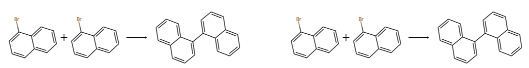
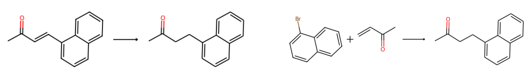
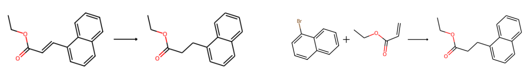
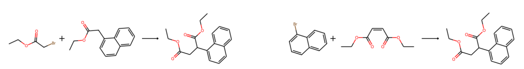
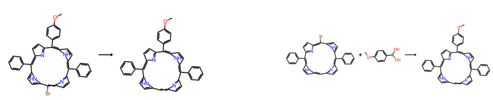
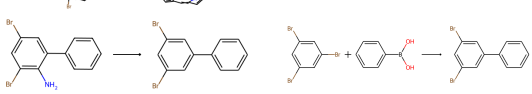
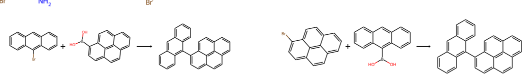
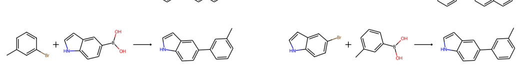
field. Therefore, we suggest that the community focus more on USPTO-Full.

It is worth noting that as top-k increases, G2GT's advantage gradually diminishes. The diversity problems are outstanding (see Table 4) when no data augmentation and decoding strategies are used.

Ablation Study on USPTO-50k. We developed various techniques to enhance model performance. This section examines the impact of these techniques on the final results. Table 4 presents the ablation results for USPTO-50k. The results demonstrate that the proposed self-training, sampling, frequency ranking, and weak ensemble effectively improved the top-1 accuracy by 6% and the top-10 by 13.2%. **Self-training** as a data augmentation technique helps the model to generalize to different scaffolds and is therefore beneficial when training on smaller data sets like USPTO-50k. We see a relatively small improvement on USPTO-Full as the data set already contains enough molecules to help the model generalize. During the inference stage, we propose to use **sampling and frequency ranking** to enhance diversity. Unlike beam search, sampling can produce a diverse set of predictions with relatively low quality. To successfully retrieve the top-k possible unique predictions, frequency ranking is used. This technique increases the top-10 and top-1 accuracies by 7% and 1%, respectively. Finally, the results show that the novel **weak-ensembling** method consistently increased the top-k accuracy. This method combines the idea of "prompting" in natural language processing (NLP), where the model takes in an input

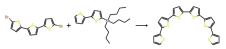
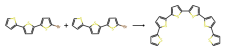
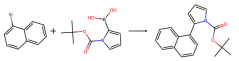
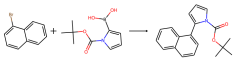
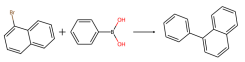
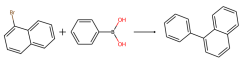
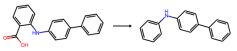
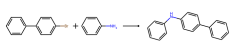
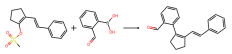
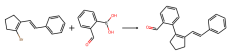
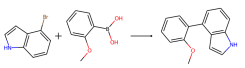
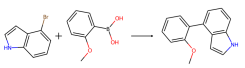
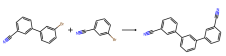
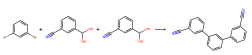
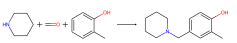
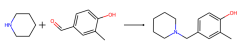
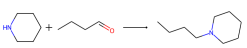
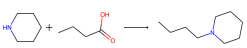
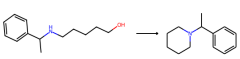
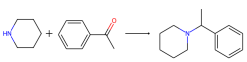
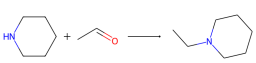
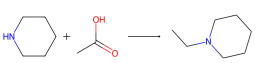
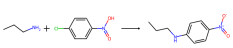
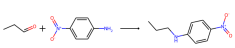
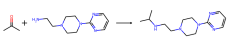
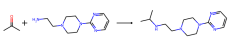
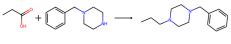
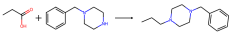
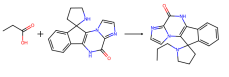
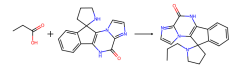
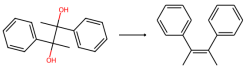
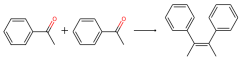
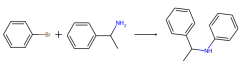
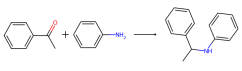
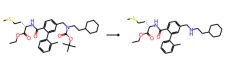
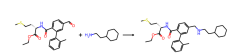
https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

Table 6. Expert Evaluation of 63 Reactions from Reaxys Predicted by Our Model

expert evaluation	ground truth reaction type	our prediction / ground truth
F	condensation	
T	condensation	
F	condensation	
F	condensation	
T	condensation	
T	condensation	
T	condensation	
T	condensation	
T	condensation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	arylation	
T	Suzuki	
T	Suzuki	
T	Suzuki	
T	Suzuki	

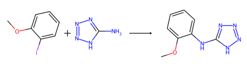
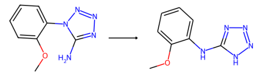
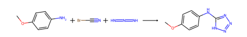
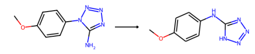
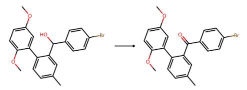
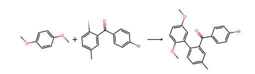
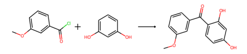
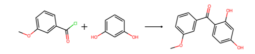
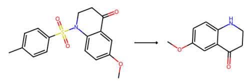
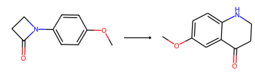
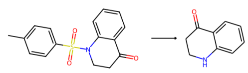
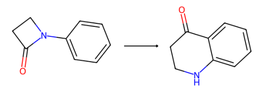
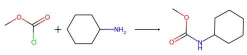
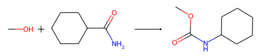
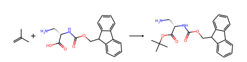
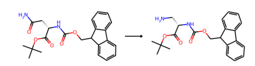
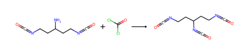
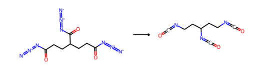
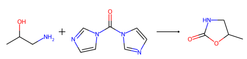
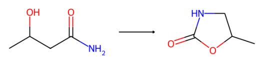
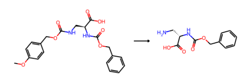
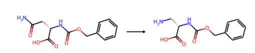
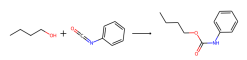
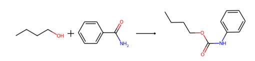
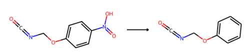
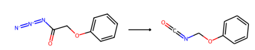
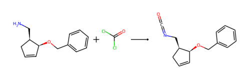
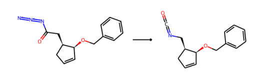
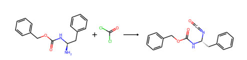
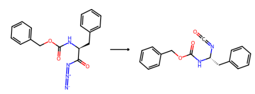
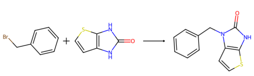
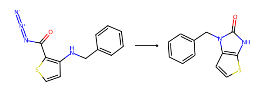
https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

Table 6. continued

expert evaluation	ground truth reaction type	our prediction	ground truth
F	Suzuki		
T	Suzuki		
T	Suzuki		
T	Suzuki		
F	Suzuki		
T	Suzuki		
T	Suzuki		
T	Suzuki		
T	Suzuki		
T	reductive		
T	reductive		
F	reductive		
T	reductive		
T	reductive		
T	reductive		
T	reductive		
T	reductive		
F	reductive		
T	reductive		
T	reductive		
T	rearrangement		
T	rearrangement		
T	rearrangement		

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

Table 6. continued

expert evaluation	ground truth reaction type	our prediction	ground truth
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		
F	rearrangement		
T	rearrangement		
T	rearrangement		
T	rearrangement		

prompt x and predicts an output y as $P(y|x)$, with the model ensemble to overcome the distribution bias of the training set.

Model Generalization Verification. The two experiments outlined above verified the performance of our model on standard data sets. However, information leakage is unavoidable because the training and test sets come from the same source, the USPTO. Therefore, we selected a 4507 reaction data set containing five common reaction types from Reaxys[27] to further evaluate the generalization and robustness of our proposed method. Furthermore, an organic synthesis expert analyzed 63 randomly selected cases to investigate the performance. The expert evaluation results and the top-1 exact-match accuracy compared to AT[14] are shown in Table 5. We discovered that, although the exact-match accuracy was low for some reaction types, such as the rearrangement reaction, the expert evaluation revealed that most of the predicted reactants were valid. We conclude that the proportion of reaction types in the training set is related to this phenomenon. Because rearrangement reactions are rare in the training set, our model tends to predict a nonrearrangement reaction type for these products. Figure 7 depicts a few corresponding cases. For a more concrete evaluation result, see Table 6 in the Appendix.

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

The exact-match accuracy cannot fully reflect the prediction's actual validity, but it does reflect the distribution of the reaction class in the training set to some extent.

CONCLUSION AND LIMITATION

We implemented a graph decoder structure based on the Transformer model for the first time that supports the parallel prediction of all time steps in the training stage and successfully applied it to the one-step retrosynthesis problem. In addition, self-training was introduced into reaction prediction to increase the number of training samples. We proposed various techniques to further enhance diversity performance and proved the superiority of template-free and graph-based methods and their real-world applicability. The G2GT framework achieved a new SOTA result on USPTO-50k with a top-1 accuracy of 54.1%. It also improved the prior USPTO-Full top-1 accuracy by 3%−49.3%, while maintaining a competitive top-10 accuracy. The improvement on USPTO-Full is especially meaningful because the data set is larger, more challenging, and closer to real-world data. Because there is no reliable atom-mapping information, no prior reaction class information and no matching templates exist for all the reactions in USPTO-Full. Furthermore, the reactions in it are not completely predictable by templates alone. As a result, our approach may perform better in real-world usage.

Limitation. Although our proposed method consistently outperformed the previous methods in terms of top-1 accuracy, we still fell short of the top-10 accuracy. Furthermore, G2GT's application to large molecules is limited because of the quadratic complexity of the self-attention module and the memory consumed by the LPE module. Further studies are required to confirm this hypothesis.

FUTURE WORK

We know that the same functional group undergoes the same or similar chemical reactions regardless of the composition of the rest of the molecule.[37] It is reasonable to believe that a model that can predict the results of a chemical reaction can reasonably characterize the functional groups. In addition to chemical reactions, functional groups inherently carry certain physical and chemical properties of molecules. Hence, this study can be viewed as a molecular representation pretraining task. We obtained preliminary results on some molecular property prediction tasks, which will be discussed in future work.

APPENDIX

Preliminaries. In this section, we discuss the basics of Transformer and Graphormer.[20]

Transformer. For illustrative purposes, we simply present a single-head attention function.Self-Attention Module

Let $Z = [z_1^T, \cdots, z_n^T]^T \in \mathbb{R}^{n\times d}$ be the input of the self-attention module, where z_i is the hidden representation at position i, and d represents the hidden dimension. Subsequently, the attention layer linearly projects Z to create three learned representations: key, query, and value vectors. The attention function is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

The scalar products of Q and K can capture their similarity and represent the relationships that matter.

Graphormer. Three main designs allow Graphormer to learn the graph representation: spatial encoding, edge encoding, and centrality encoding.Spatial Encoding

Nodes in graphs are not ordered sequentially, rather they can exist in a multidimensional spatial space and are connected by edges. If two nodes, v_i and v_j, are connected, Graphormer chooses $\phi(v_i, v_j)$ as the shortest path distance between them. Denoting $A_{(i,j)}$ as the (i, j)-element of query key product matrix A, we obtain $b_\phi(vi, vj)$, a learnable scalar indexed by $\phi(vi, vj)$ in eq 7.Edge Encoding

For each node pair (v_i, v_j), Graphormer identifies the shortest path $SP_{ij} = (e_1, e_2, ..., e_N)$ from v_i to v_j and then calculates the average of the dot products of the edge feature and a learnable embedding along the path. The proposed edge encoding adds edge features to the attention module through a bias term $c_{i,j}$. We thus arrive at the final equation:

$$A_{ij} = \frac{(z_i W_q)(z_j W_k)^T}{\sqrt{d}} + b_{\phi(v_i,v_j)} + c_{i,j} \tag{7}$$

Centrality Encoding

Graphormer uses degree centrality as an additional signal to the transformer network. The softmax attention can capture the node significance signal in queries and keys by employing centrality encoding in the input, as shown in eq 8, where d is the degree of the node in the graph that corresponds to an atom's connectivity in a molecule.

$$z_i^{(0)} = x_i + d_{deg(v_i)} \tag{8}$$

Consequently, Graphormer can capture both the semantic correlation and node significance in the attention mechanism.

Model Architecture Details. Top-k Sampling. Given a discrete vocabulary of size V, and a probability distribution over the vocabulary, P, at each time step of the generation process, the model computes the top-k probabilities, where k is a hyperparameter. These probabilities are represented as a vector P_{topk}, where the elements of this vector are the top-k probabilities of the vocabulary, and the remaining elements are zero. Then the model will sample a token from this probability distribution P_{topk}, instead of the original probability distribution P.

Formally, the top-k sampling can be described as For a given probability distribution P over the vocabulary, the top-k sampling is defined as $P_{topk} = topk(P, k)$ where $topk(P, k)$ is an operator that selects the top-k elements of the probability distribution P and returns a vector with the same shape of P but with only k nonzero elements, and these elements are the top-k elements of P.

AUTHOR INFORMATION

Corresponding Author

Zaiyun Lin − *Stone Wise, Haidian District, Beijing, China 100089*; Email: linzaiyun@stonewise.cn

Authors

Shiqiu Yin − *Stone Wise, Haidian District, Beijing, China 100089*; orcid.org/0000-0001-8125-0367

Lei Shi − *Stone Wise, Haidian District, Beijing, China 100089*

Wenbiao Zhou − *Stone Wise, Haidian District, Beijing, China 100089*

https://doi.org/10.1021/acs.jcim.2c01302
J. Chem. Inf. Model. 2023, 63, 1894−1905

Yingsheng John Zhang − *Stone Wise, Haidian District, Beijing, China 100089;* orcid.org/0000-0003-2520-3923

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c01302

Author Contributions

Zaiyun Lin, Shiqiu Yin, and Lei Shi contributed equally to this paper.

Notes

The authors declare no competing financial interest.

■ ADDITIONAL NOTE

[a]https://ogb.stanford.edu/kddcup2021/pcqm4m/.

■ REFERENCES

(1) Corey, E. J.; Long, A. K.; Rubenstein, S. D. Computer-assisted analysis in organic synthesis. *Science* **1985**, *228*, 408−418.

(2) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. Route designer: a retrosynthetic analysis tool utilizing automated retrosynthetic rule generation. *J. Chem. Inf. Model.* **2009**, *49*, 593−602.

(3) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS Cent. Sci.* **2017**, *3*, 434−443.

(4) Seidl, P.; Renz, P.; Dyubankova, N.; Neves, P.; Verhoeven, J.; Segler, M.; Wegner, J. K.; Hochreiter, S.; Klambauer, G. Modern hopfield networks for few-and zero-shot reaction template prediction. *arXiv Preprint*, arXiv:2104.03279, 2021. DOI: 10.48550/arXiv.2104.03279.

(5) Chen, W. L.; Chen, D. Z.; Taylor, K. T. Automatic reaction mapping and reaction center detection. *WIREs Comput. Mol. Sci.* **2013**, *3*, 560−593.

(6) Liu, X.; Li, P.; Song, S. Decomposing Retrosynthesis into Reactive Center Prediction and Molecule Generation. *bioRxiv Preprint*, 2020. DOI: 10.1101/677849.

(7) Somnath, V. R.; Bunne, C.; Coley, C.; Krause, A.; Barzilay, R. Learning graph models for retrosynthesis prediction. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, Vol. *34*.

(8) Wang, X.; Li, Y.; Qiu, J.; Chen, G.; Liu, H.; Liao, B.; Hsieh, C.-Y.; Yao, X. RetroPrime: ADiverse, plausible and Transformer-based method for Single-Step retrosynthesis predictions. *Chem. Eng. J.* **2021**, *420*, 129845.

(9) Shi, C.; Xu, M.; Guo, H.; Zhang, M.; Tang, J. *A Graph to Graphs Framework for Retrosynthesis Prediction*; ICML, 2020; pp 8818−8827.

(10) Yan, C.; Ding, Q.; Zhao, P.; Zheng, S.; Yang, J.; Yu, Y.; Huang, J.Retroxpert: Decompose retrosynthesis prediction like a chemist. *Adv. Neural Inf. Process. Syst.*, **2020**331124811258

(11) Gao, Z.; Tan, C.; Wu, L.; Li, S. Z. SemiRetro: Semi-template framework boosts deep retrosynthesis prediction. *arXiv Preprint*, arXiv:2202.08205, 2022. DOI: 10.48550/arXiv.2202.08205.

(12) Chen, S.; Jung, Y. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au* **2021**, *1*, 1612−1620.

(13) Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Cent. Sci.* **2017**, *3*, 1103−1113.

(14) Tetko, I. V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **2020**, *11*, 1−11.

(15) Zheng, S.; Rao, J.; Zhang, Z.; Xu, J.; Yang, Y. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *J. Chem. Inf. Model.* **2020**, *60*, 47−55.

(16) Sacha, M.; Błaz, M.; Byrski, P.; Dabrowski-Tumanski, P.; Chrominski, M.; Loska, R.; Włodarczyk-Pruszynski, P.; Jastrzebski, S. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *J. Chem. Inf. Model.* **2021**, *61*, 3273−3284.

(17) Chen, B.; Shen, T.; Jaakkola, T. S.; Barzilay, R. Learning to make generalizable and diverse predictions for retrosynthesis. *arXiv Preprint*, arXiv:1910.09688, 2019. DOI: 10.48550/arXiv.1910.09688

(18) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.

(19) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, na.

(20) Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do Transformers Really Perform Badly for Graph Representation. *Adv. Neural Inf. Process. Syst* **2021**, *34*, na.

(21) Dwivedi, V. P.; Bresson, X. A generalization of transformer networks to graphs. *arXiv Preprint*, arXiv:2012.09699, 2020. DOI: 10.48550/arXiv.2012.09699.

(22) Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking graph neural networks. *arXiv Preprint*, arXiv:2003.00982, 2020. DOI: 10.48550/arXiv.2003.00982.

(23) Kreuzer, D.; Beaini, D.; Hamilton, W.; Létourneau, V.; Tossou, P. Rethinking graph transformers with spectral attention. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, na.

(24) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv Preprint*, arXiv:1710.10903, 2017. DOI: 10.48550/arXiv.1710.10903.

(25) Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12559−12571.

(26) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for pre-training graph neural networks. *arXiv Preprint*, arXiv:1905.12265, 2019. DOI: 10.48550/arXiv.1905.12265.

(27) Lawson, A. J.; Swienty-Busch, J.; Géoui, T.; Evans, D. The Making of Reaxys—Towards Unobstructed Access to Relevant Chemistry Information. In *The Future of the History of Chemical Information*; McEwen, L. R., Buntrock, R. E., Eds.; ACS Symposium Series 1164; American Chemical Society, 2014; pp 127−148.

(28) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Zidek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.

(29) Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. Curious case of neural text degeneration. *arXiv Preprint*, arXiv:1904.09751, 2019 . DOI: 10.48550/arXiv.1904.09751.

(30) Smith, S. L.; Kindermans, P.-J.; Ying, C.; Le, Q. V. Don't decay the learning rate, increase the batch size. *arXiv Preprint*, arXiv:1711.00489, 2017. DOI: 10.48550/arXiv.1711.00489.

(31) Lowe, D. M. Extraction of chemical structures and reactions from the literature. Ph.D. Thesis, University of Cambridge, 2012.

(32) Dai, H.; Li, C.; Coley, C.; Dai, B.; Song, L. Retrosynthesis prediction with conditional graph logic network. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, na.

(33) Reaxys. www.reaxys.com, Reaxys is a registered trademark of RELX Intellectual Properties SA used under license.

(34) Segler, M. H.; Waller, M. P. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chem.—Eur. J.* **2017**, *23*, 5966−5971.

(35) Sun, R.; Dai, H.; Li, L.; Kearnes, S.; Dai, B. Energy-based View of Retrosynthesis. arXiv Preprint, arXiv:2007.13437, 2020. DOI: 10.48550/arXiv.2007.13437.

(36) Karpov, P.; Godin, G.; Tetko, I. V. *A Transformer Model for Retrosynthesis*; ICANN, 2019; pp 817−830.

(37) Nic, M.; Jirat, J.; Kosata, B.; McNaught, A.; Wilkinson, A.; Jenkins, A. *Compendium of Chemical Terminology*; IUPAC, Gold Book, 1997.