Effective Sequence Models and Graph Neural Networks for Molecular Data Analysis

by

CHAOCHAO YAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2022

To my dear family and girlfriend for their endless trust, support, and love.

ACKNOWLEDGEMENTS

I would like to express my most sincere thanks to my supervising professor, Dr. Junzhou Huang who guided and supported me to finish this dissertation without whom this dissertation would not have been possible. His consistent support, including but not limited to mental, financial, and intelligent support, and guidance are indispensable contributions to the completion of my doctoral study. His immense knowledge, unlimited support, patient guidance, constructive feedback, and encouraging conversations deeply inspired me. For the last five years, Dr. Huang would always be there when I needed his support and insights, reviewing my progress constantly, and guiding me through my research projects. During this long and fruitful journey, I have learned a lot from Dr. Huang, both academically and in life. I believe all of them will also play a significant role and be continuously influential in my future career and life.

I sincerely express my gratitude to Dr. Jean Gao, Dr. Jia Rao, and Dr. Dajiang Zhu, for serving on my committee. I have benefited a lot from their invaluable feedback and suggestions on my diagnostic evaluation, comprehensive exam, proposal exam, and final dissertation. Furthermore, they have long inspired me through their contributions to biology and bioinformatics, operating systems, and brain imaging and computational neuroscience, respectively.

I am very honored to have an opportunity to work with Dr. Peilin Zhao from Tencent AI Lab. I had gained so many insights from those thoughtful discussion with him. The long-term collaboration with him helped me to become a real researcher.

For the past five years, it is my great pleasure to work with my collaborators and all lab members from the Scalable Modeling & Imaging & Learning Lab (SMILE),

I want to thank all of them for their insightful discussion and collaboration. They are not only my research companions, but also my close friends in life. I am also very lucky to meet a lot of caring and nice friends, in Arlington, TX, Bethesda, MD, and Mountain View, CA. We have had so many delightful moments together. There are so many of them that I can not list all of them comprehensively here. Thank them for making this challenging journey enjoyable and memorable.

Finally, I want to thank my parents, sister, and brother, for their unconditional love and support in my life. Especially thanks to my girlfriend who have supported and encouraged me throughout this long journey. I could not have finished this journey without them.

August 12, 2022

ABSTRACT

Effective Sequence Models and Graph Neural Networks for
Molecular Data Analysis

Chaochao Yan, Ph.D.
The University of Texas at Arlington, 2022

Supervising Professor: Dr. Junzhou Huang

Drug discovery is the process of discovering new candidate medications. New drugs are continually developed by pharmaceutical industries to address increasing medical needs. Drug discovery involves a series of processes including target identification and validation, hit identification, lead generation and optimization, and finally the identification of a candidate for further development. The development further includes optimization of chemical synthesis and its formulation, toxicological studies in animals, clinical trials, and eventually regulatory approval. Both of these processes are time-consuming and cost-expensive.

Computer-aided drug discovery mainly relies on modern computers to model drug molecules, which can speed up the process of drug discovery and reduce costs. In this dissertation, we will investigate two representative applications of drug discovery: molecule generation and retrosynthesis prediction. Since molecules can be represented as either sequences or graphs, therefore different machine learning models (sequence models and graph neural networks) can be adapted for molecular modelling. As the rapid development of machine learning, there are abundant research works try to apply machine learning models on drug discovery. However, these methods are not efficient and effective enough for real-world applications. We propose to improve the efficiency of modern machine learning models for the drug discovery applications. We will explore two representative applications of drug discovery: molecule generation and retrosynthesis prediction. Particularly, we propose new techniques to improve the current sequence models for the molecule generation and graph models for the retrosynthesis prediction, respectively. Extensive experiments prove the efficiency and effectiveness of our methods.

We will first investigate variational autoencoder models for molecule sequence generation. We propose a simple and effective solution to the posterior collapse problem of variational autoencoder models. Then we will study retrosynthesis prediction, and we propose both template-free and template-based methods to overcome the disadvantages of existing methods.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

Molecule Generation With Re-balanced Variational Autoencoder Loss

In this chapter, we investigate the molecule generation task, which is the procedure to generate initial novel molecule proposals for molecule design. Molecule sequences are first projected into continuous vectors in chemical latent space and then these embedding vectors are decoded into molecules under the variational autoencoder (VAE) framework. The continuous latent space of VAE can be utilized to generate novel molecules with desired chemical properties and further optimize the desired chemical properties of molecules. However, there is a posterior collapse problem with the conventional RNN-based VAEs for the molecule sequence generation, which deteriorates the generation performance. We investigate the posterior collapse problem and find that the underestimated reconstruction loss is the main factor in the posterior collapse problem in molecule sequence generation. To support our conclusion, we present both analytical and experimental evidence. What is more, we propose an efficient and effective solution to fix the problem and prevent posterior collapse. As a result, our method achieves competitive reconstruction accuracy and validity score on the benchmark datasets.

Our initial work [10] is published in the Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, and its extension work is published in the Journal of Computational Biology.

1.1 Introduction

The key challenge of material and drug design is to discover novel molecules that have the desired physical or chemical properties. This process can be understood as an optimization problem as described by [11], and the optimization target is to search for molecules with the optimal desired property scores. However, exhaustive exploration in the molecule space is infeasible, since the total number of estimated drug-like molecules is in the order of 10^{60} as estimated by [12]. Besides, molecule synthesis methods like [13] and [14] and molecule validation procedures are also time-consuming and expensive in practice, which makes the brute-force exploration infeasible.

As deep learning methods are making more and more achievements in multiple fields [15, 16, 17, 18, 19, 20, 21, 22], they have also been applied for molecule sequence generation. The majority of existing molecule generation methods heavily rely on the variational autoencoder (VAE) proposed by [23] and [24]. VAE is the combination of a deep latent variable model and an accompanying variational learning technique. As illustrated in Figure 1.1, drug molecules can be represented in the Simplified Molecular-Input Line-Entry System (SMILES) format proposed by [25]. SMILES is a specification in the form of a line notation for describing the structure of chemical molecules. In Figure 1.1, the input SMILES sequence **CO(C)C** is first fed into the VAE encoder composed by Gated Recurrent Unit (GRU) layers by [26] to generate the latent representation. Then the VAE decoder takes the latent vector as the input to reconstruct the original molecule sequence **CO(C)C**. One of the desirable properties of the VAE is that its latent space is continuous and smooth. As a result, it allows both semantically meaningful sampling and smooth interpolation in the latent space. In the case of molecule generation, the latent representations of semantically similar molecules (with similar chemical structures and properties) are often clustered together in the latent space. Thanks to the continuous latent space, novel molecules can be generated by randomly sampling from the latent space since the sampled latent vectors can be regarded as the interpolation of existing molecule representations. What is more, the desired properties can also be further optimized through exploring the

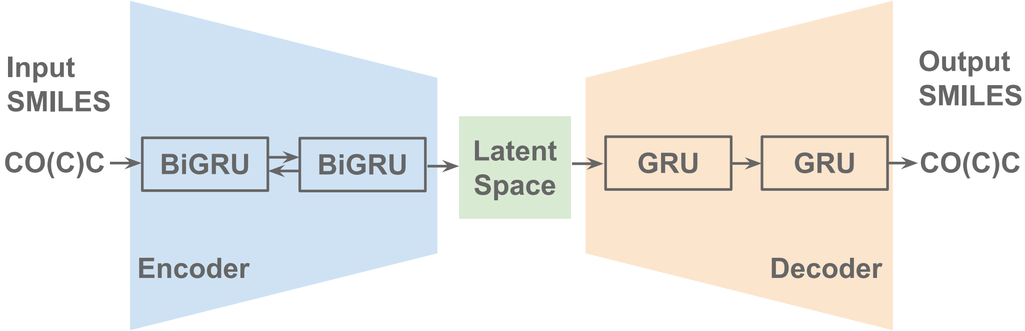

Figure 1.1: Overview of our VAE model implementation. The encoder and decoder are built based on the bi-directional GRU and uni-directional GRU, respectively. Both the input and output of our model are SMILES sequences.

latent space locally. The key idea behind the optimization process is to utilize the smoothness of the latent space to search for molecules that maximize a property score objective by perturbing slightly to the initial latent vector.

However, previous VAE models suffer from the posterior collapse issue, where the decoder tends to ignore latent vectors as described in [27] and [11]. This problem is more frequently observed in Recurrent Neural Network (RNN) based models as in [28]. In consequence, the generated molecules are in low diversity and are hardly relevant to the latent vectors as in [11] and [1]. This phenomenon has also been observed in Natural Language Processing (NLP) tasks, such as the text generation by [27]. The major focus of previous NLP related studies is to propose various training strategies to alleviate this problem, such as the Kullback–Leibler (KL) cost annealing by [27] and optimizing the decoder multiple times before each encoder update in [28]. However, simply extending these methods to molecule generation can not help molecule generation too much, mostly because the molecule sequences are strictly structured according to SMILES grammar rules and any mutation within the molecule sequences lead to invalid sequences. Motivated by the success of attribute grammars in the compiler design and parse trees in the NLP field, following work [1] and [2] propose to incorporate grammar rules to guarantee the validity of generated SMILES sequences. As an alternative, a molecule can also be represented by a graph in order to avoid the posterior collapse as in [29] and [4].

Thanks to the development of NLP text generation, the VAE model is applied for molecule generation for the first time in CVAE by [11]. They build a VAE encoder and decoder with GRU layers, representing molecules in the SMILES sequences. However, their model suffers from generating invalid SMILES sequences which makes their model impracticable. To improve the prior validity, context-free grammars for SMILES are introduced in Grammar VAE (GVAE) by [1] to represent a molecule in the sparse tree. However, the validity score is still unsatisfactory. Inspired by this method, Syntax-directed VAE (SD-VAE) by [2] incorporates extra semantic rules to ensure generated SMILES are valid, and it achieves the best performance among all SMILES-based methods. However, these models did not solve the model posterior collapse problem and there is a large performance gap.

We propose a novel strategy to alleviate the posterior collapse problem considering the essential drawbacks of the contemporary RNN-based VAE models in the molecule generation situation. To achieve this goal, we carefully analyze the posterior collapse problem of the vanilla VAE model for SMILES sequence generation. We point out that the underestimated reconstruction loss triggers the posterior collapse issue in the molecule sequence generation, as the direct consequence of the imbalance between reconstruction loss and KL loss during VAE training. To overcome the problem, we propose a novel loss function to leverage the trade-off between the reconstruction loss and the KL loss in VAE training. Without modifying the VAE network structures or costing extra computational complexity, our proposed strategy is extremely simple yet effective in preventing the posterior collapse in molecule generation. We also provide a detailed analysis of our method[1], and empirically demonstrate its excellent reconstruction accuracy and competitive validity score on the ZINC 250K dataset from [1] and GuacaMol dataset from [30].

In addition to the experimental verification for the statement that the underestimated reconstruction loss causes the posterior collapse of the variational autoencoder models, we also provide theoretical analysis and proof in this work. Besides, to further improve the validity score of our method, we introduce a partial SMILES sequence check toolkit PartialSmiles[2] to verify the validity of the SMILES sequence during the molecule generation process. What is more, to better evaluate the proposed method, we include the results of two extra evaluation metrics novelty and uniqueness in experimental comparison with baseline methods. Last but not least, we conduct experiments on the extra large-scale dataset GuacaMol which consists of 1.6M molecules to demonstrate the scalability and generalization of our proposed method.

1.2 Methods

1.2.1 The Variational Autoencoder

The VAE is a specially regularized variant of the standard autoencoder (AE). It is appealing because it can learn complex distribution in an unsupervised manner and later can act as a generative model defined by a prior distribution $p(z)$ and a conditional distribution $p_\theta(x|z)$. Since the true data likelihood is often intractable, the VAE instead maximizes the evidence lower bound objective (ELBO) $\mathcal{L}(x;\theta,\phi)$

[1] Our implementation is available at `https://github.com/chaoyan1037/Re-balanced-VAE`.

[2] https://github.com/baoilleach/partialsmiles

over the space of all p_θ, and it is a valid lower bound of the true data log likelihood $\log p(x)$:

$$
\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x)] \quad (p_\theta(x) \text{ does not depend on } z)\\
&= \mathbb{E}_z[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}] \quad \text{(Bayes' theorem)}\\
&= \mathbb{E}_z[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)} \frac{q_\phi(z|x)}{q_\phi(z|x)}] \quad \text{(Multiply by a constant)}\\
&= \mathbb{E}_z[\log p_\theta(x|z)] - \mathbb{E}_z[\log \frac{q_\phi(z|x)}{p_\theta(z)}] + \mathbb{E}_z[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}] \qquad (1.1)\\
&= \mathbb{E}_z[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z)) + D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x))\\
&\geq \mathbb{E}_z[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z))\\
&= \mathcal{L}(x;\theta,\phi),
\end{aligned}
$$

where the VAE encoder $q_\phi(z|x)$ is parameterized with ϕ and learns to map the input x to a variational distribution represented by z, and the VAE decoder $p_\theta(x|z)$ parameterized with θ reconstructs the input x given the latent vector z. The inequality holds since the $D_{\mathrm{KL}} \geq 0$. In practice, $q_\phi(z|x)$ is usually modeled as a Gaussian distribution and it is optimized to approximate the true posterior $p_\theta(z|x)$ to reduce the gap between ELBO and true data log-likelihood $\log p(x)$.

The VAE training is optimized to maximize the ELBO, where (i) negative reconstruction loss $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ enforces the encoder to generate meaningful latent vector z, so that the decoder can reconstruct the input x from z, and (ii) the KL regularization loss $D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z))$ minimizes the KL divergence between the approximate posterior $q_\phi(z|x)$ and the prior $p_\theta(z) \sim \mathcal{N}(0, \mathbf{I})$.

1.2.2 Posterior Collapse Problem in VAE

The posterior collapse phenomenon has also been reported in previous work on NLP text generation such as [27], [31], and [32]. When posterior collapse happens, the model training falls into the the local optimum of the ELBO, in which the decoder tends to ignore z when training the VAE model and the variational posterior $q_\phi(z|x)$ naively mimics the model prior $p(z)$. Note that the KL loss in the ELBO objective can be further decomposed as in [33]:

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi(z|x)||p(z))] &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p(z)}] \\ &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p(z)} \frac{q_\phi(z)}{q_\phi(z)}] \\ &= D_{\mathrm{KL}}(q_\phi(z)||p(z)) + I_q(x,z), \end{aligned} \tag{1.2}$$

where $I_q(x,z)$ is the mutual information between x and z given $q_\phi(z|x)$:

$$I_q(x,z) = \mathbb{E}_{q_\phi(z|x)}[\log q_\phi(z|x)] - \mathbb{E}_{q_\phi(z)}[\log q_\phi(z)]. \tag{1.3}$$

When posterior collapse occurs, the KL loss decreases nearly to zero so that I_q is also close to zero (both items on the right-hand side in (1.2) are non-negative) during the VAE model training process. It is especially evident when modelling discrete data with a strong auto-regressive network such as Long Short Term Memory (LSTM) by [34] and GRU by [35], which is exactly our case for molecule sequence generation. This is undesirable since the VAE model fails to learn meaningful latent representations for input molecule sequences.

For text generation task in NLP, the posterior collapse problem has been mainly attributed to the low quality of latent representations z at the early stage of model training as pointed out by [27], [28], and [36]. To be more specific, the decoder $p_\theta(x|z)$ falls behind the encoder $q_\phi(z|x)$ at the initial training stage, and $q_\phi(z|x)$ generates low-quality latent representations so that it is very hard for $p_\theta(x|z)$ to recover the input sequences. As a result, the model is forced to ignore z. Many solutions have been proposed to solve the problem and they have demonstrated satisfactory improvement on various NLP datasets.

However, molecule SMILES generation is a quite different scenario though it appears to be same as the NLP text generation. First of all, its vocabulary size is far less than the NLP text generation datasets. The token size of NLP text data is usually tens of thousands or even larger, while it is less than 100 for chemical molecule data. The smaller token size makes the molecule reconstruction task much easier. Second, the molecule sequence is composed strictly following the SMILES grammar rules, and the reconstructed sequence must be exactly the same as the input to be matched successfully. Any token mutations can result in an invalid sequence. However, there are no such rigid grammar rules applied to the NLP text and the exact match is not required.

We find the existing solutions [28] and [36] for NLP text generation performs poorly in the chemical molecule generation. This motivates us to propose such a solution for molecule sequence generation.

1.2.3 The Trick in Previous Solutions

To avoid the posterior collapse, which will disable the reconstruction ability of VAE models, previous SMILES-based methods such as CVAE [11], GVAE [1], and SD-VAE [2] reduce the standard deviation σ of prior Gaussian distribution to a small value 0.01 (can be found in their public implementations CVAE[3], GVAE[4], SD-VAE[5]), which makes their models more like AEs instead of VAEs. This is why CVAE and GVAE have a decent reconstruction accuracy but extremely low validity scores as

[3] https://github.com/HIPS/molecule-

[4] https://github.com/mkusner/grammarVAE

[5] https://github.com/Hanjun-Dai/sdvae

shown in Table 1.1. If we set σ=1, all these models will suffer from the posterior collapse and can not reconstruct inputs faithfully (similar to the vanilla VAE in Figure 1.2(e)). In the following our analysis and experiments, we will strictly keep σ=1.

1.2.4 Underestimated Reconstruction Loss

To investigate the cause of the posterior collapse in the VAE for molecule sequence generation, we conduct extensive analysis and investigation into the posterior collapse process. We find it is the underestimated reconstruction loss that causes posterior collapse during VAE training process. Both theoretical analysis and experimental support are provided to verify our hypothesis.

The reconstruction loss term $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ in formula (1.1) measures the reconstruction ability of the decoder given the latent vector z. The decoder should only receive information from z and tries to reconstruct the full sequence accurately from the given z. However, in practice, the RNN models are usually trained with the teacher forcing method proposed by [37], in which the RNN input at each step is the ground truth instead of the prediction from a prior time step.

We can rewrite the reconstruction loss term in (1.1) as:

$$\mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log p_\theta(x_t|z, \tilde{x}_{0,\ldots,t-1}), \tag{1.4}$$

where the T is the maximum time step, $\tilde{x}_{0,\ldots,t-1}$ is the predicted sequence prefix before time step t, the current input of the RNN is output $\tilde{x}_{t-1}$ at the previous time step, and $\tilde{x}_0$ is the predefined start symbol.

With teacher forcing training method, now the actual reconstruction loss during VAE training is:

$$\mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log p_\theta(x_t|z, x_{0,\ldots,t-1}), \tag{1.5}$$

where $x_{0,\dots,t-1}$ is the ground-truth prefix before time step t, the ground-truth token at previous time step x_{t-1} is the RNN input at each time step t, and x_0 is also the predefined start symbol.

We posit $\log p_\theta(x_t|z, x_{0,\dots,t-1}) = \log p_\theta(x_t|z, x_{0,\dots,t-1}, \tilde{x}_{0,\dots,t-1})$ since with teacher forcing the RNN training does not rely on the prediction as the input. Then we can prove the log-likelihood (1.5) is larger than (1.4):

$$\begin{aligned}
&\mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log p_\theta(x_t|z, x_{0,\dots,t-1}) \\
&= \mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log p_\theta(x_t|z, x_{0,\dots,t-1}, \tilde{x}_{0,\dots,t-1}) \\
&= \mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log \frac{p_\theta(x_t, x_{0,\dots,t-1}|z, \tilde{x}_{0,\dots,t-1})}{p_\theta(x_{0,\dots,t-1}|z, \tilde{x}_{0,\dots,t-1})} \\
&= \mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} [\log p_\theta(x_t|z, \tilde{x}_{0,\dots,t-1}) - \log p_\theta(x_{0,\dots,t-1}|z, \tilde{x}_{0,\dots,t-1})] \\
&\geq \mathbb{E}_{q_\phi(z|x)} \sum_{t=1}^{T} \log p_\theta(x_t|z, \tilde{x}_{0,\dots,t-1})
\end{aligned} \tag{1.6}$$

The ground-truth information $x_{0,\dots,t-1}$ is incorporated additionally at each time step in (1.5) when training the VAE, and it makes the decoder's prediction task easier, therefore we can expect that the reconstruction ability of the decoder is largely overestimated compared with (1.4). As a result, the reconstruction loss term is underestimated, which will potentially break the balance between reconstruction loss and KL loss in the formula (1.1). We calculate quantitatively how much the reconstruction loss is underestimated in the experiment section.

1.2.5 Re-balanced VAE Loss

Since reconstruction loss is underestimated during training and it breaks the balance with KL loss, which eventually leads to the posterior collapse. We propose to recover the balance by applying a weight α to reconstruction loss:

$$\begin{aligned}\mathcal{L}(x;\theta,\phi) = \alpha\mathbb{E}_{q_\phi(z|x)}&[\log p_\theta(x|z)] \\ &- D_{\text{KL}}(q_\phi(z|x)||p(z)), \alpha > 1,\end{aligned} \tag{1.7}$$

where α can be estimated using Monte Carlo sampling in every training iteration. Specifically, we can sample a batch of data as input and run a VAE with/without teacher forcing, respectively. Since the reconstruction loss without teacher forcing can be regarded as the "true" reconstruction loss (the reconstruction loss it should be in VAE training), we approximate α as the ratio of reconstruction loss without teacher forcing to that with teacher forcing. However, estimating α in every training iteration is too expensive. In practice, we can set α as a hype-parameter for simplicity and efficiency. We show how to decide the optimal value for α in the experiment part.

Inspired by the β-VAE [38] formulation, we can instead reduce KL loss weight β, which is equivalent to increasing reconstruction loss weight α. It is more natural and convenient to weight the KL loss since increasing β from 0 to 1 is a smooth transition from the AE to VAE. So we can have a similarly re-balanced VAE loss formulation:

$$\begin{aligned}\mathcal{L}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}&[\log p_\theta(x|z)] \\ &- \beta D_{\text{KL}}(q_\phi(z|x)||p(z)), 0 \leq \beta < 1.\end{aligned} \tag{1.8}$$

Note that in our case $\beta < 1$, while the β-VAE requires the KL weight $\beta > 1$. The β-VAE is proposed in [38] to learn disentangled representation of generative factors by enforcing a larger penalty on KL loss, since they postulate that $\beta > 1$ could place a stronger constraint on the latent representation to drive the VAE to learn a more efficient latent representation of input x. While we have a completely different motivation and goal of fixing the imbalanced VAE loss by reducing KL weight since we find reconstruction loss is underestimated in ELBO.

Except for the above analysis, our method can also be explained from an intuitive perspective. In previous methods CVAE, GVAE, and SD-VAE, when sampling latent vectors z they have to reduce the standard deviation σ to a small value of 0.01 otherwise the model will collapse and lose the reconstruction ability. However, the validity score is poor in these methods. Instead of reducing sampling σ, we can anneal the KL loss weight β to make the model gradually transform from AE to VAE as in [27] since the AE usually has a strong reconstruction ability. Different from [27], we restrict β to be smaller than 1. By applying the optimal β, we can arrive at a trade-off between the reconstruction accuracy and validity score.

We acknowledge that previous methods such as [2], [28], and [36] have empirically tried to reduce the KL loss weight to avoid the posterior collapse. The β-VAE ($\beta = 0.4$) alleviates the problem and achieves competitive performance on density estimation for NLP text datasets in [28], which proves that reducing β is viable for NLP text task. It is also indicated setting $\beta = 1/\text{LatentDimension}$ could lead to better results in [1] and [2]. However, none of these methods provided any analysis or explanation. We are the first to recognize that the underestimated reconstruction loss leads to the posterior collapse problem in VAE molecule generation, and further, we propose to reduce KL weight to overcome the posterior collapse with detailed analysis and solid experimental support.

1.3 Results

Our proposed solution to the VAE model posterior collapse is simple but extremely effective and efficient. We do not need to modify the network architecture and only adjust the training loss slightly, without introducing much extra computation. In this section, we will first train a vanilla VAE model and track the process of model collapse, as well as experimentally verify that the reconstruction loss is underestimated. Then we will conduct extensive experiments to demonstrate the effectiveness of our proposed method.

1.3.1 Experimental Settings

We build our VAE model based on GRU layers. The VAE encoder is composed of two layers of bi-directional GRU which is good at capturing sequence representation as [39], and hidden size of each GRU layer is 512. The decoder is made up of four layers of uni-directional GRU with the same hidden size 512. Following previous work [11] and [4], we use unit Gaussian prior and set the latent vector dimension to be 56. The ELBO objective is optimized with Adam optimizer by [40] and learning rate is 0.0001. The model is trained with teacher forcing and KL loss annealing. We train the model for 150 epochs and report the performance of the final model. Experiments are conducted on a machine with an Intel Core i7-5930K@3.50GHz CPU and a GTX 1080 Ti GPU.

We experiment on ZINC 250K dataset by [1] which is a subset of the ZINC by [41]. Molecule sequences are tokenized with the regular expression from [42]. We use the same training and testing split as previous work [1] and [4], and have 10K hold-out data out of the training as the validation data. We also experiment on a large-scale dataset GuacaMol by [30] which is derived from the ChEMBL 24 database by [43] to demonstrate the scalability and generalization of our method. GuacaMol dataset consists of 1.6M molecules and we adopt the same data split provided by [30]. We will use the same experimental settings in all our experiments unless explicitly stated.

As for the model evaluation metrics, we report the reconstruction accuracy, validity, novelty, and uniqueness scores following previous work. Following [4], we encode each molecule from test dataset, and then decode obtained latent vector to reconstruct input molecule SMILES. The reconstructed SMILES must be exactly the same as the input to be counted as successful. The reconstruction accuracy is defined to be the ratio of successfully reconstructed molecule sequences to the total tried reconstruction. To calculate validity, 10K latent vectors are randomly sampled from the prior distribution as the input for the decoder. The validity is the portion of chemically valid reconstruction SMILES from the random sampling to the total decoded sequences. We use open-source tool RDKit by [44] to check the validity of SMILES. The novelty is the ratio of generated chemically valid molecules which are not present in the training dataset to the total generated chemically valid molecules. It evaluates the model's ability to generate novel molecules. The uniqueness is used to evaluate to what extent a model generates unique chemically valid molecules, and it is defined as the ratio of generated chemically valid molecules that are unique.

1.3.2 VAE Training Dynamics

We track the training process of a vanilla VAE model, as well as that of our proposed method. We investigate training dynamics including the KL loss weight, KL loss, reconstruction loss, mutual information, reconstruction accuracy, and validity score. Mutual information $I_q(x, z)$ can be calculated using Monte Carlo sampling as proposed in [33] and [45]:

$$I_q = D_{\mathrm{KL}}(q_\phi(z|x)||p_(z)) - D_{\mathrm{KL}}(q_\phi(z)||p(z)), \tag{1.9}$$

which is actually the same as the formula (1.2). We approximate the aggregated posterior $q_\phi(z) = \mathbb{E}_{p_d(x)}[q_\phi(z|x)]$ using Monte Carlo sampling. $D_{\mathrm{KL}}(q_\phi(z)||p_(z))$ can also be estimated by the Monte Carlo sampling, and we can obtain samples from $q_\phi(z)$ by ancestral sampling: first sampling x from the dataset distribution $p_d(x)$ and then sampling z from $q_\phi(z|x)$. More details about $I_q(x, z)$ computation can be found in [33].

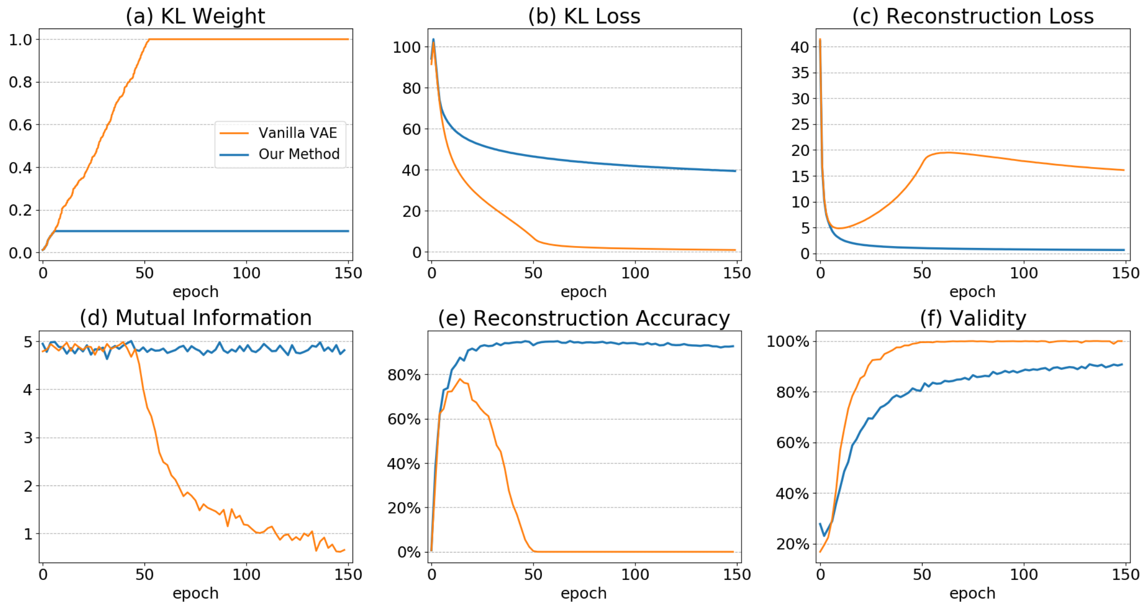

Figure 1.2: Training dynamics of vanilla VAE model and our method on validation data. We track (a) KL weight β, (b) KL loss $D_{\mathrm{KL}}(q_\phi(z|x)||p(z))$, (c) reconstruction loss $-\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$, (d) mutual information $I_q(x, z)$, (e) reconstruction accuracy, and (f) validity score during the full training process. The orange line is the vanilla VAE with KL loss annealing, and the maximum KL weight β is 1. Our method (Blue) reduces the maximum value of β to 0.1. Both models are trained with KL weight annealing and teacher forcing.

As a comparison, we also illustrate the training dynamics of our proposed method. We set KL weight $\beta = 0.1$ which is explained and derived in the next section. We keep all the other experimental settings the same as the vanilla VAE to make a fair comparison.

Results of the two models' training are plotted in the Figure 1.2. The vanilla VAE model performs well on the validation data at the early stage of the KL weight annealing. However, as the KL weight increases, KL loss drops quickly as expected since more penalty is added to the KL loss term, while the reconstruction loss starts to rise at the same time. The mutual information $I_q(x, z)$ decreases to 0.65 at the end, which means the decoder does not absorb much information from the latent vectors when generating the output. This evidence indicates the posterior collapse has happened. When looking at the model performance on validation data, we can notice that the reconstruction accuracy is close to 0 while the validity score is almost perfect. This indicates that too much pressure has been placed on the KL loss, which breaks the balance between the reconstruction loss and KL loss and results in the model posterior collapse.

On the other hand, our method achieves lower reconstruction loss early and can maintain it during model training. Although the KL loss of our method is larger than the vanilla VAE, considering that we have a smaller KL weight β now, the equivalent KL loss added to the training objective should still be in the normal range. Especially, our method maintains the mutual information to be around 4.8, which means output sequences are strongly related to latent vectors. As for the model performance, our method achieves 92.7% reconstruction accuracy and 90.7% validity score, which proves the superiority of our method.

1.3.3 Underestimated Reconstruction Loss

We find that introducing ground-truth information into the decoder will result in underestimated reconstruction loss in previous section, and have provided our detailed analysis previously. In this section, we will experimentally verify that the reconstruction loss is indeed underestimated during the training. We can estimate how much the reconstruction loss has been underestimated using Monte Carlo sampling. Specifically, we can sample a batch of data, then run the model with and without the

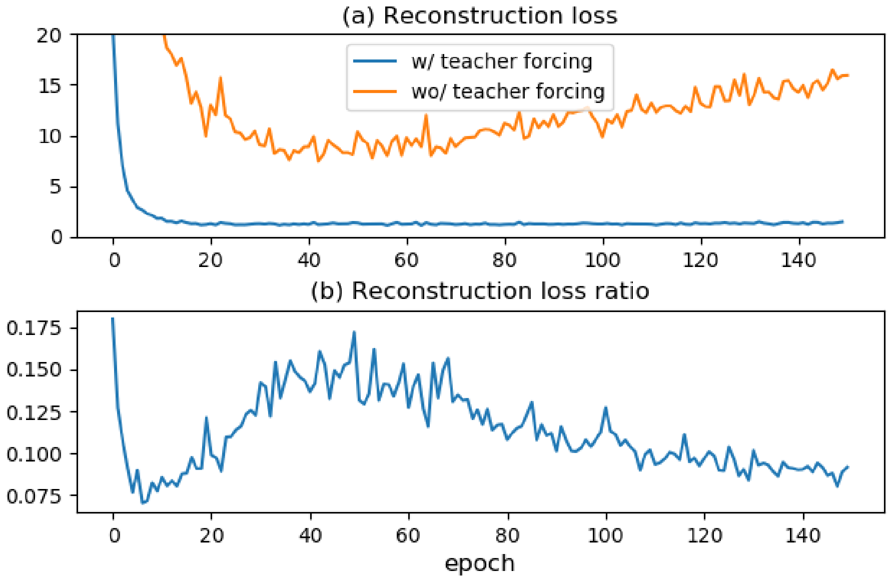

Figure 1.3: (a) Reconstruction loss on validation dataset. At each time step, models parameters are the same when calculating the reconstruction loss. (b) Underestimated ratio of reconstruction loss.

teacher forcing, respectively. The underestimated ratio can be approximated by the ratio of reconstruction loss with teacher forcing to that without teacher forcing.

We track the reconstruction loss on the validation dataset when the teacher forcing is applied and removed, respectively. Results are shown in the Figure 1.3(a). When teacher forcing is applied, the reconstruction loss drops close to 1 quickly, while the loss is much larger (at least 7.5) without teacher forcing. This is not unexpected since the prediction error may be accumulated during the decoding process without teacher forcing. Any wrong token prediction as RNN input at the next time step may result in the following prediction totally different from ground-truth sequences.

To quantitatively evaluate how much the reconstruction loss has been underestimated, we can compute the ratio as reconstruction loss w/ teacher forcing to that wo/ teacher forcing at each time step. Results are shown in Figure 1.3(b). It confirms our conclusion that the reconstruction loss is underestimated. To recover a re-balanced

Table 1.1: Reconstruction accuracy and validity results on ZINC 250K dataset. Baseline results are reported in [1], [2], [3], and [4]. * denotes the SMILES validating parser PartialSmiles is applied during the generation. The novelty and uniqueness scores of baseline methods are copied from [5].

Model	**Reconstruction**	**Validity**	**Novelty**	**Uniqueness**
		SMILES-based		
CVAE	44.6%	0.7%	98.0%	2.1%
GVAE	53.7%	7.2%	100.0%	100.0%
SD-VAE	76.2%	43.5%	100.0%	100.0%
Our method	**92.7**%	90.7%	100.0%	100.0%
Our method*	**92.7**%	93.8%	100.0%	100.0%
		Graph-based		
GraphVAE	-	13.5%	-	-
JT-VAE	76.7%	**100.0**%	99.9%	99.1%

VAE loss, we can set KL loss weight exactly as the underestimated ratio in each epoch. But this requires us to compute the ratio repetitively during training, which is time-consuming. To be simplified, we set $\beta = 0.1$ during training and we find it works very well in practice.

1.3.4 Model Performance Comparison

We summarize the molecule reconstruction accuracy, validity, novelty, and uniqueness scores on ZINC 250K test data in the Table 1.1. Our method outperforms all previous models in reconstruction accuracy by a large margin (16% larger than the second-best model). In the meanwhile, our method achieves 90.7 % validity, which is much better than previous SMILES-based methods. We can further boost our model performance by incorporating a SMILES validating parser PartialSmiles, which can check the validity of the SMILES prefix easily when generating SMILES sequences token by token. The validity score can be boosted to 93.8% with PartialSmiles.

Compared with other SMILES-based methods, our model is much superior in both the reconstruction accuracy and prior validity, even if complex grammar or

Table 1.2: Reconstruction accuracy and validity results on GuacaMol dataset. * denotes the SMILES validating parser PartialSmiles is applied during the generation.

Model	**Reconstruction**	**Validity**	**Novelty**	**Uniqueness**
Our Method	92.6%	90.6%	100.0%	100.0%
Our Method*	92.6%	93.6%	100.0%	100.0%

syntax rules are incorporated in [1] and [2]. Note that the JT-VAE model assembles molecules by adding sub-graphs step-by-step to make sure the generated molecule graphs are always valid. However, these sub-graphs are extracted from the training dataset, which limits the JT-VAE from generating molecules with unseen sub-graphs. While our method achieves competitive validity performance without any constraints, and is able to generate novel molecules that are not from the same distribution as the training data. That is one important reason why our method achieves better reconstruction accuracy, while JT-VAE suffers from reconstructing testing molecules [46]. Besides, our method is much more efficient than JT-VAE. When generating 10,000 unique valid SMILES from prior random sampling, JT-VAE[6] (faster version) takes about 1450s while our method only needs 9s.

As for the novelty and uniqueness, our method achieves 100.0% for both metrics which are the same as other SMILES-based methods including GVAE and SD-VAE. Note that the novelty and uniqueness are evaluated only on the chemically valid molecules. This indicates that even if both GVAE and SD-VAE achieve the same novelty and uniqueness scores, our method can generate much more valid molecules than GVAE and SD-VAE due to the better validity score. JT-VAE achieves only

99.9% novelty score and 99.1% uniqueness score. This demonstrates that our model is a better chemically valid molecule generator.

We also experiment on a large-scale dataset GuacaMol to evaluate the scalability and generalization of our proposed method. We use the same experimental settings as the ZINC 250K dataset. Our method achieves 92.6% reconstruction accuracy, 90.6% validity score 100.0% novelty, and 100.0% uniqueness on GuacaMol dataset, which are similar to the performance on ZINC 250K. By checking the validity of SMILES during the generation, the validity score can be further boosted to 93.6%. The experiment on the large-scale dataset demonstrates our method scales and generalizes well on a large dataset.

1.3.5 Error Analysis and Visualization

Our model achieves 92.7% reconstruction accuracy and all reconstructed SMILES are valid on the ZINC 250K dataset. We investigate the reconstruction results further and find that our model can predict 97.3% of all tokens correctly, which is measured at the level of the token instead of the sequence. Besides, most of the unmatched sequences (62%) are valid, and it confirms the reconstruction ability of our model. We show some valid but unmatched examples in Figure 1.4. Even for these unmatched examples, there is only a small ratio of the predicted tokens that are different from the ground-truth, which demonstrates the reconstruction ability of our method.

As for the validity sore, we also investigate the model outputs. We find our model can generate complicated and diverse molecules with multiple rings. As for the invalid sequences, from both the reconstruction and prior sampling, there are several typical errors: (1) unkekulized atoms, (2) valence error, (3) unclosed ring, and (4) parentheses error. We believe the grammar-based methods [1] and [2] are complementary to our method, and can be combined together to reduce these errors.

[6] https://github.com/wengong-jin/icml18-jtnn

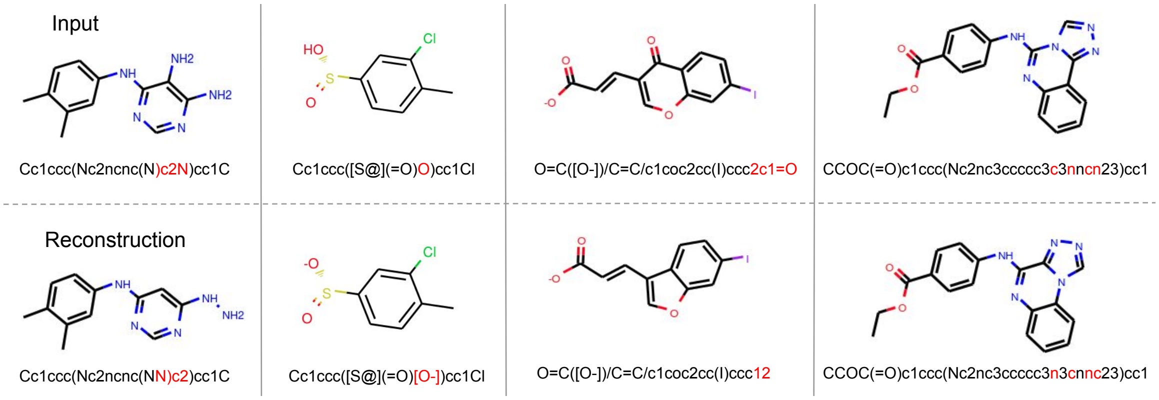
Input
Cc1ccc(Nc2ncnc(N)c2N)cc1C
Cc1ccc([S@](=O)O)cc1Cl
O=C([O-])/C=C/c1coc2cc(I)ccc2c1=O
CCOC(=O)c1ccc(Nc2nc3ccccc3c3nncn23)cc1
Reconstruction
Cc1ccc(Nc2ncnc(NN)c2)cc1C
Cc1ccc([S@](=O)[O-])cc1Cl
O=C([O-])/C=C/c1coc2cc(I)ccc12
CCOC(=O)c1ccc(Nc2nc3ccccc3n3cnnc23)cc1

Figure 1.4: Reconstruction error examples. Unmatched tokens between the input and reconstruction SMILES are shown in red. Note that “[O-]” is a single token.

1.3.6 Bayesian Optimization

One of the important tasks in the drug molecule generation is to make molecules with desired chemical properties. We follow [1] and [4] for all the experimental setting, and the optimization target score is:

$$y(m) = logP(m) - SA(m) - cycle(m), \tag{1.10}$$

where $logP(m)$ is the octanol-water partition coefficients of molecules m, $SA(m)$ is synthetic accessibility score, and $cycle(m)$ is number of large rings with more than six atoms.

We first associate each molecule with a latent vector which is the mean of the learned variational encoding distribution. The latent vector for each molecule will be treated as its feature and we train a Sparse Gaussian Process (SGP) to predict the target score $y(m)$ given its latent vector. After training SGP, five iterations of batched Bayesian optimization (BO) are performed with expected improvement heuristics.

We report the SGP prediction performance when trained on latent representations learned by different models. We train the SGP with 10-fold cross-validation and report the top-3 molecules found by the BO. As shown in Table 1.3, molecules found by our model are much better than that found by previous SMILES-based methods, and our method is even superior to the graph-based method JT-VAE. Figure 1.5 shows top-3 molecules found by our model.

Table 1.3: Top-3 molecule scores found by the BO. Baseline results are copied from [1], [2], and [4].

Model	**1st**	**2nd**	**3rd**
	SMILES-based		
CVAE	1.98	1.42	1.19
GVAE	2.94	2.89	2.80
SD-VAE	4.04	3.50	2.96
Our Method	**5.32**	**5.28**	**5.23**
	Graph-based		
JT-VAE	5.30	4.93	4.49

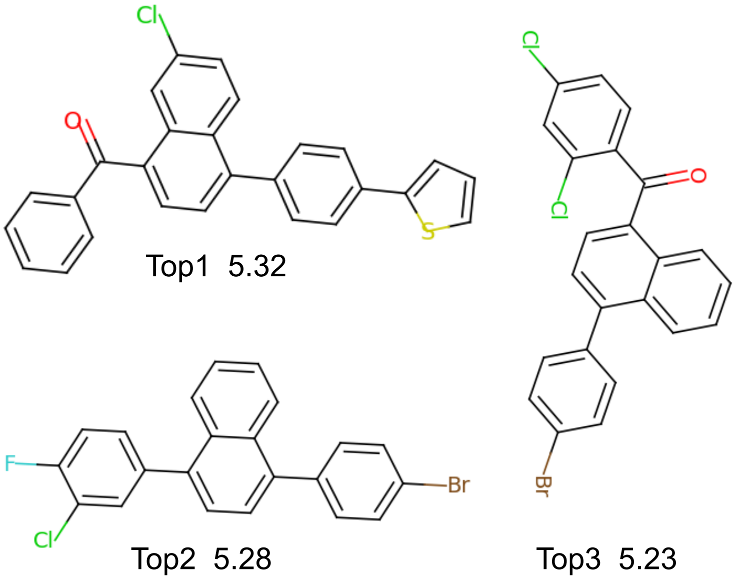

Figure 1.5: Top-3 molecules and associated scores found by our model with Bayesian optimization.

1.4 Discussion

Our method is very efficient and it works extremely well in the molecule generation, in which SMILES sequences are highly structured and grammarly organized. Our experimental results indicate that grammar and syntax rules are necessary to generate more valid SMILES sequences, and they are complementary to our method. Besides, SMILES-based methods and graph-based methods may also be combined together to boost the model performance further.

Though our primary focus is the VAE for molecule generation, our method can also help the NLP task as we mentioned at the end of section 1.2.5. Reducing KL loss weight can help the VAE model for the NLP task avoid the posterior collapse as shown in [28] and [36].

The latent representation learnt by our model can be applied to various downstream tasks, such as molecule property prediction [47, 48, 49, 50, 51]. In the future, we may explore more about this application.

1.5 Conclusions

In this work, we investigate the posterior collapse problem in VAE for molecule sequence generation. Through extensive analysis, we conclude that the underestimated reconstruction loss results in the posterior collapse. The conclusion is supported by both theoretical analysis and experimental results. Based on our analysis, we propose a simple and effective solution to overcome the underestimated reconstruction loss problem by weighting the KL loss term. With the proposed re-balanced VAE loss, the VAE model can avoid the posterior collapse problem and achieve excellent performance in both reconstruction accuracy and validity score on two datasets. We also demonstrate the excellent generalization of our method on a large-scale dataset.

CHAPTER 2

A Two-stage Template-free Retrosynthesis Prediction Method

In this chapter, we start to study another important topic in drug discovery: retrosynthesis prediction. Retrosynthesis is the process of recursively decomposing target molecules into available building blocks. It plays an important role in solving problems in organic synthesis planning. To automate or assist in the retrosynthesis analysis, various retrosynthesis prediction algorithms have been proposed. However, most of them are cumbersome and lack interpretability about their predictions. In this chapter, we devise a novel two-stage template-free algorithm for automatic retrosynthetic expansion inspired by how chemists approach retrosynthesis prediction. Our method disassembles retrosynthesis into two steps: i) identify the potential reaction center of the target molecule through a novel graph neural network and generate intermediate synthons, and ii) generate the reactants associated with synthons via a robust reactant generation model. While outperforming the state-of-the-art baselines by a significant margin, our model also provides chemically reasonable interpretation.

2.1 Introduction

Retrosynthesis of the desired compound is commonly constructed by recursively decomposing it into a set of available reaction building blocks. This analysis mode was formalized in the pioneering work [52, 53] and now have become one of the fundamental paradigms in the modern chemical society. Retrosynthesis is challenging, in part due to the huge size of the search space. The reported synthetic-organic knowledge consists of in the order of 10^7 reactions and compounds [54]. On the other hand, the incomplete understanding of the reaction mechanism also increases the difficulty of retrosynthesis, which is typically undertaken by human experts. Therefore, it is a subjective process and requires considerable expertise and experience. However, molecules may have multiple possible retrosynthetic routes and it is challenging even for experts to select the most appropriate route since the feasibility of a route is often determined by multiple factors, such as the availability of potential reactants, reaction conditions, reaction yield, and potential toxic byproducts.

In this work, we focus on the single-step version (predict possible reactants given the product) of retrosynthesis following previous methods [55, 8, 9]. Our method can be decomposed into two sub-tasks [52, 56]: i) *Breaking down* the given target molecule into a set of **synthons** which are hypothetical units representing potential starting reactants in the retrosynthesis of the target, and ii) *Calibrating* the obtained synthons into a set of reactants, each of which corresponds to an available molecule.

Various computational methods [57, 58, 59, 60, 61, 55, 7, 62, 8, 9, 63, 64] have been developed to assist in designing synthetic routes for novel molecules, and these methods can be broadly divided into two template-based and template-free categories. Template-based methods plan retrosynthesis based on hand-encoded rules or reaction templates. Synthia (formerly Chematica) relies on hand-encoded reaction transformation rules [60], and it has been experimentally validated as an efficient software for retrosynthesis [65]. However, it is infeasible to manually encode all the synthesis routes in practice considering the exponential growth in the number of reactions [62]. Reaction templates are often automatically extracted from the reaction databases and appropriate templates are selected to apply to the target [61, 7, 62, 8]. The key process of these approaches is to select relevant templates for the given target. An obvious limitation is that these methods can only infer reactions within the chemical space covered by the template database, preventing them from discovering novel reactions [66].

On the other hand, template-free methods [55, 9, 63] treat the retrosynthesis as a neural machine translation problem, since molecules can be represented as SMILES [1] strings. Although simple and expressive, these models do not fit into the chemists' analytical process and lack interpretability behind their predictions. Besides, such approaches fail to consider rich chemistry knowledge within the chemical reactions. For example, the generation order of reactants is undetermined in [55, 9, 63] since they ignore the correlation between synthons and reactants, resulting in slower and inferior model convergence. Similar to our method, the concurrent work G2Gs [64] also presents a decomposition and generation two-step framework. G2Gs proposes to incrementally generate reactants from the associated synthons with a variational graph translation model. However, G2Gs can predict at most one bond disconnection which is not universal. Besides, G2Gs independently generates multiple reactants, which ignores the relationship between multiple reactants.

To overcome these challenges, inspired by the expert experience from chemists, we devise a two-step framework named as RetroXpert (**Retro**synthesis e**Xpert**) to automate the retrosynthesis prediction. Our model tackles it in two steps as shown in Figure 3.3. Firstly, we propose to identify the potential reaction center within the target molecule using a novel Edge-enhanced Graph Attention Network (EGAT). The reaction center is referred to as the set of bonds that will be disconnected in the retrosynthesis process. Synthons can be obtained by splitting the target molecule according to the reaction center. Secondly, the Reactant Generation Network (RGN) predicts associated reactants given the target molecule and synthons. Different from previous methods [55, 9, 63], the reactant generation order can be uniquely decided in our method, thanks to the intermediate synthons. What is more, we notice that the robustness of the RGN plays an important role. To robustify the RGN, we propose to

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

augment the training data of RGN by incorporating unsuccessful predicted synthons.

Our main contributions can be summarized as follows:

1) We propose to identify the potential reaction center with a novel Edge-enhanced Graph Attention Network (EGAT), which is strengthened with chemical domain knowledge.
2) By splitting the target molecule into synthons, the RGN is able to determine the generation order of reactants. We further propose to augment training data by introducing unsuccessfully predicted synthons, which makes RGN robust and achieves significant improvement.
3) On the standard USPTO-50K dataset [67], our method achieves 62.1% and 50.4% Top-1 accuracy for w/ and wo/ reaction type, respectively.

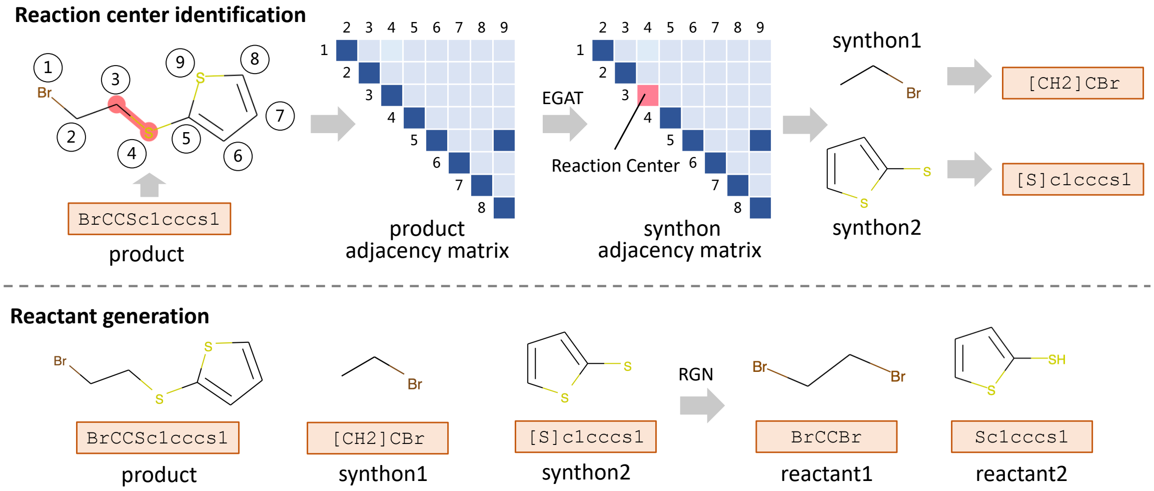

Figure 2.1: Pipeline overview. We conduct retrosynthesis in two closely dependent steps **reaction center identification** and **reactant generation**. The first step aims to identify the reaction center of the target molecule and generates intermediate synthons accordingly. The second step is to generate the desired set of reactants. Note that a molecule can be represented in two equivalent representations: molecule graph and SMILES string.

2.2 Background

2.2.1 Transformer

The transformer [68] is an autoregressive encoder-decoder model built with multi-head attention layers and position-wise feed-forward layers. As illustrated in Figure 2.2, the encoder is composed of stacked multi-head self-attention layers and position-wise feed-forward layers. The encoder self-attention layers attend the full input sequence and iteratively transform it into a latent representation with the self-attention mechanism. The decoder is similar to the encoder. In addition to multi-head self-attention layers and position-wise feed-forward layers, the multi-head encoder-decoder attention layers are inserted to perform cross attention over the encoder output. Different from the encoder self-attention layers, the decoder adopts the masked self-attention which prevents the decoder positions from attending future positions. The encoder-decoder attention and masked self-attention layers enable the decoder to combine the information from the source sequence and the target sequence that has been produced to make the output prediction. We refer readers to [68] and The Illustrated Transformer for comprehensive details about the Transformer.

The transformer removes all recurrent units and introduces a positional encoding to account for the order information of the sequence. Positional encoding adds a position-dependent signal to the token embedding of size d_{emb} to discriminate the position of different tokens in the sequence:

$$PE_{(pos,2i)} = \sin \frac{pos}{10000^{2i/d_{emb}}}, PE_{(pos,2i+1)} = \cos \frac{pos}{10000^{2i/d_{emb}}} \tag{2.1}$$

where pos is the token position and i is the dimension of the positional encoding.

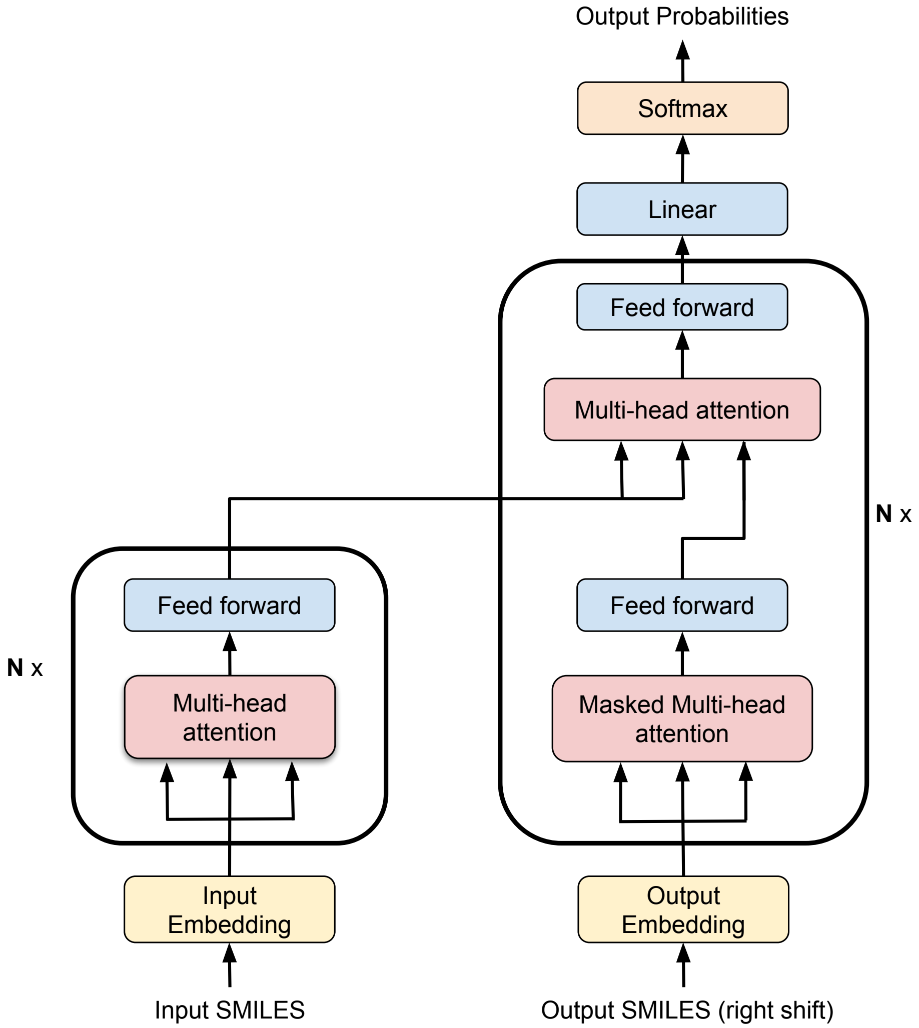

Figure 2.2: Transformer model architecture. The residual connection and layer normalization layer are omitted in the illustration for simplification.

The transformer adopts a scale dot-product attention as the attention formulation, which compute the attention weighted output by taking as input the matrix represented keys K, values V, and queries Q:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

where the d_k is the dimension of Q and K.

2.2.2 Parameters setting

We compose both the encoder and decoder of four layers of size 256. The label smoothing parameter is set to 0 since a nonzero label smoothing parameter will deteriorate the model's discrimination [69]. We adopt eight attention heads as suggested. We set the batch size to 4096 tokens and accumulate gradients over four batches.

2.3 Methodology

Given a molecule graph $\mathbf{G}$ with N nodes (atoms), we denote the matrix representation of node features as $X \in \mathbb{R}^{N \times M}$, the tensor representation of edge features as $E \in \mathbb{R}^{N \times N \times L}$, and the adjacency matrix as $A \in \{0,1\}^{N \times N}$. M and L are feature dimensions of atoms and bonds, respectively. We denote as P, S, R the product, synthons, and reactants in the reaction formulation, respectively. The single-step retrosynthesis problem can be described as given the desired product P, seeking for a set of reactants $R = \{R_1, R_2, ..., R_n\}$ that can produce the major product P through a valid chemical reaction. It is denoted as $P \to R$ (predict R given P), which is the reverse process of the forward reaction prediction problem [70, 71] that predicts the outcome products given a set of reactants.

As illustrated in Figure 3.3, our method decomposes the retrosynthesis task ($P \to R$) into two closely dependent steps **reaction center identification** ($P \to S$) and **reactant generation** ($S \to R$). The first step is to identify the potential reaction bonds which will be disconnected during the retrosynthesis, and then the product P can be split into a set of intermediate synthons $S = \{S_1, S_2, ..., S_n\}$. Note that each synthon S_i can be regarded as the substructure of a reactant R_i. The second step is to transform synthons $S = \{S_1, S_2, ..., S_n\}$ into associated reactants $R = \{R_1, R_2, ..., R_n\}$.

Although the intermediate synthons are not needed in retrosynthesis, decomposing the original retrosynthesis task ($P \rightarrow R$) into two dependent procedures can have multiple benefits, which will be elaborated thoroughly in the following sections.

2.3.1 EGAT for reaction center identification

We treat the reaction center identification as a graph-to-graph transformation problem which is similar to the forward reaction outcome prediction [71]. To achieve this, we propose a graph neural network named Edge-enhanced Graph Attention Network (EGAT) which takes the molecule graph $\mathbf{G}$ as input and predicts disconnection probability for each bond, and this is the main task. Since a product may be produced by different reactions, there can be multiple reaction centers for a given product and each reaction center corresponds to a different reaction. While current message passing neural networks [72] are shallow and capture only local structure information for each node, and it is difficult to distinguish multiple reaction centers without global information. To alleviate the problem, we add a graph-level auxiliary task to predict the total number of disconnection bonds.

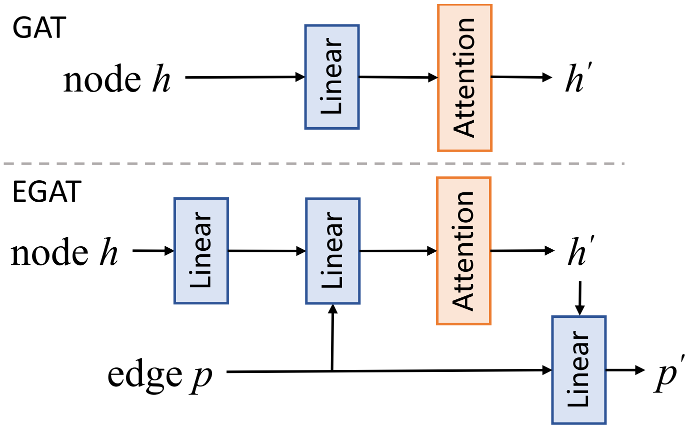

Figure 2.3: Embedding computation flows of GAT and the proposed EGAT.

As shown in Figure 2.3, distinct from the Graph Attention Network (GAT) [73] which is designed to learn node and graph-level embeddings, our proposed EGAT also learns edge embedding. It identifies the reaction center by predicting the disconnection probability for each bond taking its edge embedding as input. Given the target $\mathbf{G} = \{A, E, X\}$, the EGAT layer computes node embedding h'_i and edge embedding $p'_{i,j}$ from previous layer's embeddings h_i and $p_{i,j}$ by following equations:

$$\begin{aligned}
z_i &= \mathbf{W}h_i,\\
c_{i,j} &= \text{LeakyReLU}(\mathbf{a}^T[z_i||z_j||p_{i,j}]),\\
\alpha_{i,j} &= \frac{\exp(c_{i,j})}{\sum_{k\in\mathcal{N}_i}\exp(c_{i,k})},\\
h'_i &= \sigma(\sum_{j\in\mathcal{N}_i}\alpha_{i,j}\mathbf{U}[z_j||p_{i,j}]),\\
p'_{i,j} &= \mathbf{V}[h'_i||h'_j||p_{i,j}],
\end{aligned} \tag{2.3}$$

where $\mathbf{W} \in \mathbb{R}^{F' \times F}$, $\mathbf{a} \in \mathbb{R}^{2F'+D}$, $\mathbf{U} \in \mathbb{R}^{F\times(F'+D)}$, and $\mathbf{V} \in \mathbb{R}^{D\times(2F+D)}$ are trainable parameters, $||$ means concatenation operation, $\mathcal{N}_i$ is all neighbor nodes of the node i, $\alpha_{i,j}$ is the attention weight between the node i and its neighbor node j, and $h'_i \in \mathbb{R}^F$ as well as $p'_{i,j} \in \mathbb{R}^D$ are the output node and edge representations, respectively. Initial input embeddings $h_i, p_{i,j}$ are the input node and edge feature vectors $x_i, e_{i,j}$, respectively, which will be detailed later, and in this special case the dimensions F and D equals to the dimensions of associated features, respectively.

After stacking multiple EGAT layers, we obtain the final edge representation $p_{i,j}$ for the chemical bond between nodes i and j, as well as the node representation h_i for each node i. To predict the disconnection probability for a bond, we perform a fully-connected layer parameterized by $\mathbf{w}_{fc} \in \mathbb{R}^D$ and a *Sigmoid* activation layer to $p_{i,j}$ and its disconnection probability is $d_{i,j} = \text{Sigmoid}(\mathbf{w}_{fc}^T \cdot p_{i,j})$. Note that the multi-head attention mechanism can also be applied like the original GAT. The optimization goal for bond disconnection prediction is to minimize the negative log-likelihood between prediction $d_{i,j}$ and ground-truth $y_{i,j} \in \{0,1\}$ through the binary cross entropy loss function:

$$\mathcal{L}_{\mathrm{M}} = -\frac{1}{K} \sum_{k=1}^{K} \sum_{a_{i,j} \in \mathbf{A}_k} a_{i,j} \left[(1 - y_{i,j}) \log(1 - d_{i,j}) + y_{i,j} \log(d_{i,j}) \right], \tag{2.4}$$

where K is the total number of training reactions and bond (i,j) exists if the associated adjacency element $a_{i,j}$ is nonzero. The ground truth $y_{i,j} = 1$ means the bond (i,j) is disconnected otherwise remaining the same during the reaction. Bond disconnection labels can be obtained by comparing molecule graphs of target and reactants.

The input of auxiliary task is graph-level representation $h_G = \mathrm{READOUT}(\{h_i | 1 \leq i \leq N\})$, which is the output of the READOUT operation over all learned node representations. We adopts an arithmetic mean as the READOUT function $h_G = \frac{1}{N} \sum_{i=1}^{N} h_i$ and it works well in practice.

Similarly, a fully-connected layer parameterized by $\mathbf{W}_s \in \mathbb{R}^{(1+N_{max}) \times F}$ and a *Softmax* activation function are applied to h_G to predict the total number of disconnected bonds, which is solved as a classification problem here. Each category represents the exact number of disconnected bonds, so there are $1{+}N_{max}$ classification categories. N_{max} is the maximum number of possible disconnected bonds in the retrosynthesis. We denote the *Softmax* output as $q = \mathrm{Softmax}(\mathbf{W}_s \cdot h_G)$. The total number of disconnected bonds for each target molecule is predicted as:

$$n^* = \arg\max_n (q_n) = \arg\max_n (\mathrm{Softmax}(\mathbf{W}_s \cdot h_G)_n), 0 \leq n \leq N_{max}. \tag{2.5}$$

The ground truth number of disconnections for molecule k is denoted as N_k, the indicator function $\mathbb{1}(i, N_k)$ is 1 if i equals to N_k otherwise it is 0, and the cross entropy loss for the auxiliary task:

$$\mathcal{L}_{\mathrm{A}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{CrossEntropy}(N_k, q^k) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{i=0}^{N_{max}} \mathbb{1}(i, N_k)\log(q_i^k). \tag{2.6}$$

Finally, the overall loss function for the EGAT is $\mathcal{L}_{\mathrm{EGAT}} = \mathcal{L}_{\mathrm{M}} + \alpha\mathcal{L}_{\mathrm{A}}$, where α is fixed to 1 in our study since we empirically find that α is not a sensitive hype-parameter.

The atom feature consists of a series of general atom information such as atom type, hybridization, and formal charge, while the bond feature is composed of chemical bond information like bond type and conjugation (see Appendix 2.4.4 for details). These features are similar to those used in [6] which is for chemical property prediction. We compute these features using the open-source toolkit RDKit [2]. To fully utilize the provided rich atom-mapping information of the USPTO datasets [67] [74], we add a semi-templates indicator to atom feature. For retrosynthesis dataset with given reaction type, a type indicator is also added to the atom feature.

For atom-mapped USPTO datasets, reaction templates are extracted from reaction data like previous template-based methods [61, 62, 8]. However, we are not interested in full reaction templates since these templates are often too specific. There are as many as 11,647 templates for the USPTO-50K train data [8]. Only the product side of templates are kept instead, which we name as semi-templates. Since reaction templates are closely related to the exact reaction, the semi-templates indicator expected to play a significant role in reaction center identification.

The semi-templates can be considered as subgraph patterns within molecules. We build a database of semi-templates from training data and find all appeared semi-templates within each molecule. For each atom, we mark the indicator bits

[2] `https://www.rdkit.org`

associated with appeared semi-templates. Note that each atom within a molecule may belong to several semi-templates since these semi-templates are not mutually exclusive. Although reaction templates are introduced, our method is still template-free since i) only semi-templates are incorporated and our method does not rely on full templates to plan the retrosynthesis, and ii) our EGAT still works well in the absence of semi-templates, with only slight performance degradation.

2.3.2 Reactant generation network

Once the reaction center has been identified, synthons can be obtained by applying bond disconnection to decompose the target graph. Since each synthon is basically a substructure within the reactant, we are informed of the total number of reactants and substructures of these reactants. The remaining task $S \rightarrow R$ is much simpler than the original $P \rightarrow R$ in which even the number of reactants is unknown.

Specifically, task $S \rightarrow R$ is to generate the set of desired reactants given obtained synthons. Based on commonsense knowledge of chemical reaction, we propose that the ideal RGN should meet following three requirements: R1) be permutation invariant and generate the same set of reactants no matter the order of synthons, R2) all given information should be considered when generating any reactant, and R3) the generation of each reactant also depends on those previously generated reactants.

To fulfill these requirements, we represent molecules in SMILES and formulate $S \rightarrow R$ as a sequence-to-sequence prediction problem. We convert synthon graphs to SMILES representations using RDKit, though these synthons may be chemically invalid. As in Figure 2.4, source sequence is the concatenation of possible reaction types, canonical SMILES of the product, and associated synthons. The target sequence is the desired reactants arranged according to synthons.

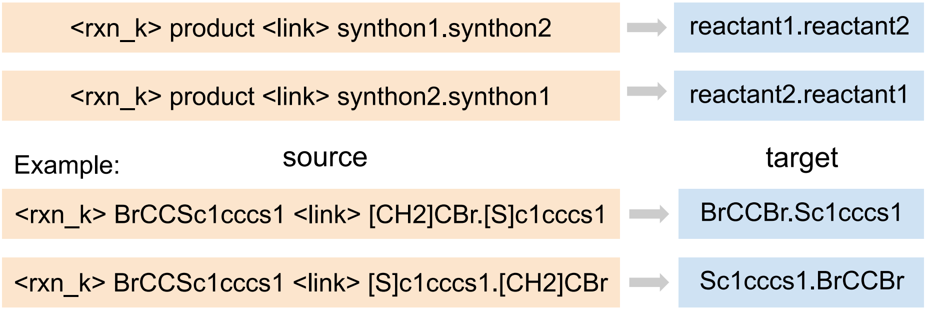
<rxn_k> product <link> synthon1.synthon2
reactant1.reactant2
<rxn_k> product <link> synthon2.synthon1
reactant2.reactant1
Example:
source
target
<rxn_k> BrCCSc1cccs1 <link> [CH2]CBr.[S]c1cccs1
BrCCBr.Sc1cccs1
<rxn_k> BrCCSc1cccs1 <link> [S]c1cccs1.[CH2]CBr
Sc1cccs1.BrCCBr

Figure 2.4: Illustration of source and target sequences. <RXN_K> is the kth reaction type if applicable. The product and synthons are separated with a special <LINK> token. The order of reactants is arranged according to synthons. SMILES strings are joined with a dot following RDkit.

We approximate the requirement R1 by augmenting train samples with reversely arranged synthons and reactants as shown in Figure 2.4. Our empirical studies demonstrate that such approximation works pretty well in practice. To satisfy the requirement R2, the encoder-decoder attention mechanism [75] [68] is employed, which allows each position in the target sequence attends to all positions in the source sequence. A similar masked self-attention mechanism [68], which masks future positions in the decoder, is adopted to make the RGN meet the requirement R3.

Motivated by the great success of Transformer [68] in natural machine translation, we build the RGN based on the Transformer module. Transformer is a sequence-to-sequence model equipped with two types of attention mechanisms: self-attention and encoder-decoder attention [68]. Transformer is also adapted for reaction outcome prediction [69] and retrosynthesis [9], in which both products and reactants are represented in SMILES. We include a brief description of Transformer in section 2.2.1.

For the first time, the generation order of reactants can be determined by aligning reactants in the target with synthons in the source, thanks to intermediate synthons which are associated with reactants uniquely. While the generation order of reactants is undetermined in previous methods [55, 9, 63], which naively treats the sequence-to-sequence model as a black box. The uncertainty of the generation order makes their models hard to train.

2.3.2.1 Robustify the RGN.

We find the EGAT suffers from distinguishing multiple coexisting reaction centers, which is the major bottleneck of our method. As a result of the failure of identifying the reaction center, the generated synthons are different from the ground truth. To make our RGN robust enough and able to predict the desired reactants even if the EGAT fails to recognize the reaction center, we further augment RGN training data by including those unsuccessfully predicted synthons on training data. We do not reverse the order of synthons for these augmentation samples like in Figure 2.4. The intuition behind is that EGAT tends to make similar mistakes on training and test datasets since both datasets follow the same distribution. This method can make our RGN able to correct reaction center prediction error and generate the desired set of reactants.

2.4 Experiments

2.4.1 Dataset

We use USPTO-50K [67] and USPTO-full [74] to verify the effectiveness and scalability. USPTO-50K consists of 50K reactions annotated with 10 reaction types, which is derived from USPTO granted patents [76]. It is widely used in previous retrosynthesis work. The USPTO-50K dataset is annotated with 10 reaction types, the distribution of reaction types is displayed in Table 2.1. The distribution is extremely unbalanced. We also report the statistics of the number of disconnection bonds for training reactions in Tables 2.2 and 2.3.

Table 2.1: Distribution of 10 recognized reaction types.

Reaction type	Reaction type name	# Examples
1	Heteroatom alkylation and arylation	15204
2	Acylation and related processes	11972
3	C-C bond formation	5667
4	Heterocycle formation	909
5	Protections	672
6	Deprotections	8405
7	Reductions	4642
8	Oxidations	822
9	Functional group interconversion (FGI)	1858
10	Functional group addition (FGA)	231

Table 2.2: Statistics of the number of disconnection bonds for the USPTO-50K training reactions.

# Disconnection bonds	0	1	2	≥ 3
# Reactions	11296	27851	849	12
Accumulative percent	28.23%	97.85%	99.97%	100.00%

Table 2.3: Statistics of the number of disconnection bonds for the USPTO-full training reactions.

# Disconnection bonds	0	1	2	3	4	5	≥ 6
# Reactions	161500	485449	88146	19303	5687	2032	1000
Accumulative percent	21.16%	84.77%	96.33%	98.86%	99.60%	99.87%	100.00%

2.4.2 Dataset and Preprocessing.

We evaluate our method on We adopt the same training/validation/test splits in 8:1:1 as [61, 8]. For RGN training data, we add an extra 28K samples of which synthons are reversed as shown in Figure 2.4 if there are at least two synthons. There are 68K training samples for RGN, which is still denoted as USPTO-50K in the following content. The USPTO-full consists of 950K cleaned reactions from the USPTO 1976-2016 [74], which has 1,808,937 raw reactions without reaction types. Reactions with multiple products are duplicated into multiple single-product ones.

After removing invalid reactions (empty reactant and missing atom mappings) and deduplication, we can obtain 950K reactions [3], which are randomly partitioned into training/validation/test sets in 8:1:1.

For the EGAT, we build molecule graphs using DGL [77] and extract atom and bond features with RDkit. By comparing molecule graphs of product and reactants, we can identify disconnection bonds within the product graph and obtain training labels for both main and auxiliary tasks. This comparison can be easily done for atom-mapped reactions. For reactions without atom-mapping, a substructure matching algorithm in RDKit can be utilized to accomplish the comparison. We use RDChiral [78] to extract super general reaction templates, and obtain 1859 semi-templates for USPTO-50K training data. Semi-templates that appear less than twice are filtered and finally 654 semi-templates are obtained. As for the RGN, the product molecule graph is divided into synthon graphs according to the ground truth reaction center, then are converted into SMILES strings. The input sequence of RGN is the concatenation of the possible reaction type, product SMILES string, and synthon SMILES strings as illustrated in Figure 2.4.

2.4.3 Implementation.

All reactions are represented in canonical SMILES, which are tokenized with the regular expression in [42]. We use DGL [77] and OpenNMT [79] to implement our EGAT and RGN models, respectively. As for the EGAT, we stack three identical four-head attentive layers of which the hidden dimension is 128. All embedding sizes in EGAT are set to 128, such as F, F', and D. The N_{max} is set to be two to cover 99.97% training samples. We train the EGAT on USPTO-50K for 80 epochs. EGAT

[3] Code and processed USPTO-full data are available at `https://github.com/uta-smile/RetroXpert`

parameters are optimized with Adam [40] with default settings, and the initial learning rate is 0.0005 and it is scheduled to multiply 0.2 every 20 epochs. We train the RGN for $300,000$ time steps, and it takes about 30 hours on two GTX 1080 Ti GPUs. We save a checkpoint of RGN parameters every $10,000$ steps and average the last 10 checkpoints as the final model. We run all experiments for three times and report the means of their performance in default.

2.4.4 Atom and bond features

Table 2.4: Atom Features used in EGAT. All features are one-hot encoding, except the atomic mass is a real number scaled to be on the same order of magnitude. The upper part is general atom feature following [6], the lower part is specifically extended for the retrosynthesis prediction. Semi-templates size is 654 for the USPTO-50K dataset.

Feature	Description	Size
Atom type	Type of atom (ex. C, N, O), by atomic number.	100
# Bonds	Number of bonds the atom is involved in.	6
Formal charge	Integer electronic charge assigned to atom.	5
Chirality	Unspecified, tetrahedral CW/CCW, or other.	4
# Hs	Number of bonded Hydrogen atom.	5
Hybridization	sp, sp2, sp3, sp3d, or sp3d2.	5
Aromaticity	Whether this atom is part of an aromatic system.	1
Atomic mass	Mass of the atom, divided by 100.	1
Semi-templates	Semi-templates that the atom is within.	654
Reaction type	The specified reaction type if it exists.	10

2.4.4.0.1 Evaluation metric. The Top-N accuracy is used as the evaluation metric for retrosynthesis. Beam search [80] strategy is adopted to keep top K predictions throughout the reactant generation process. K is set to 50 in all experiments.

Table 2.5: Bond features used in EGAT. All features are one-hot encoding.

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

The generated reactants are represented in canonical SMILES. A correct predicted set of reactants must be exactly the same as the ground truth reactants.

2.4.5 Reaction center identification results

Table 2.6: Results of EGAT on USPTO-50K dataset. *EAtt* and *Aux* are the short for edge-enhanced attention and auxiliary task, respectively. *EGAT* consists of both main and auxiliary tasks. The prediction is binarized with a threshold of 0.5 if main task alone.

Type	EAtt	Accuracy (%)		
		Main	Aux	EGAT
✓	✗	73.9	99.1	85.7
✓	✓	**74.4**	**99.2**	**86.0**
✗	✗	50.0	86.1	64.3
✗	✓	**51.5**	**86.4**	**64.9**

To verify the effectiveness of edge-enhanced attention mechanism, we also include the ablation study by removing edge embedding $p_{i,j}$ when computing the coefficient $c_{i,j} = \text{LeakyReLU}(\mathbf{a}^T[z_i||z_j])$. Results are reported in Table 2.6. The auxiliary task (**Aux**) can successfully predict the number of disconnection bonds for 99.2% test molecules given the reaction type (**Type**) while 86.4% if not given.

As for the main task (**Main**) alone, its prediction accuracy is 74.4% w/ reaction type and 51.5% wo/ reaction type. However, if we adopt the prediction from the auxiliary task as the prior of the number of disconnection bonds, and select the most probable disconnection bonds (**EGAT**), then the prediction accuracy can be boosted to 86.0% (w/) and 64.9% (wo/), respectively. The edge-enhanced attention (**EAtt**) can consistently improve the model's performance in all tasks. The improvement is more significant when the reaction type is unknown, so our EGAT is more practical in real world applications without reaction types. This demonstrates that the reaction type information plays an important role in the retrosynthesis. The reactions of the same type usually share similar reaction patterns (involved atoms, bonds, and functional groups), it is much easier to recognize the reaction center if the reaction type is given as the prior knowledge. We also verify the importance of semi-templates in Appendix 2.7.

2.4.6 Reactant prediction results

To robustify the RGN as described in the paragraph **Robustify the RGN**, we also conduct the $P \rightarrow S$ prediction on the EGAT training data for USPTO-50K (40K), and the prediction accuracy is 89.0% for the reaction type conditional setting. We can obtain about 4K unsuccessful synthon predictions as augmentation samples (**Aug**), adding the original 68K RGN training data, the total RGN training data size is 72K. For the unconditional setting, the EGAT accuracy is 70.0% and there are 12K augmentation samples, and the total RGN training size is 80K in this case. We train RGN models on the USPTO-50K with/without the augmentation (**Aug**), and report results in Table 2.7.

For the RGN evaluation, the RGN input consists of the ground truth synthons. Therefore the results in Table 2.7 indicate the upper bound of our method's overall retrosynthesis performance. The proposed augmentation strategy does not always improve the upper bound. Without given reaction type, the RGN generally performs worse with the augmentation due to the introduced dirty training samples. However, when given reaction type, this augmentation boosts its prediction accuracy. We presume that it is because the reaction type plays a significant role. The RGN learns to put more attention on the reaction type and product instead of synthons to generate the reactants.

Table 2.7: $S \rightarrow R$ prediction results. *Aug* denotes training data augmentation. Evaluation results are based on ground-truth synthons as the RGN input.

Type	Aug	Training size	Top-n accuracy (%)					
			1	3	5	10	20	50
✓	✗	68K	72.9	86.5	88.3	89.5	90.4	91.6
✓	✓	72K	**73.4**	**86.7**	**88.5**	**89.7**	**90.9**	**92.1**
✗	✗	68K	**71.9**	**85.7**	**87.5**	**88.9**	**90.0**	**91.0**
✗	✓	80K	70.9	84.6	86.4	88.2	89.4	90.6

To evaluate the overall retrosynthesis prediction accuracy, the generated synthons from $P \rightarrow S$ instead of the ground truth are input into the RGN. In this way, we only need to compare the predicted reactants with the ground truth ones, without considering if the reaction center predictions correct or not. We report the retrosynthesis results in Tables 3.1. Our method RetroXpert achieves impressive performance on the test data. Specifically, when given reaction types, our proposed method achieves 62.1% Top-1 accuracy. As for results wo/ given reaction type, our model achieves 50.4% Tpo-1 accuracy.

Table 2.8: Retrosynthesis results compared with the existing methods. NeuralSym [7] results are copied from [8]. *We run the self-implemented SCROP [9] and official implementation of GLN [8] on the USPTO-full dataset.

Methods	Top-n accuracy (%)					
	1	3	5	10	20	50
Reaction types given as prior on USPTO-50K						
Seq2Seq [55]	37.4	52.4	57.0	61.7	65.9	70.7
RetroSim [61]	52.9	73.8	81.2	88.1	91.8	92.9
NeuralSym [7]	55.3	76.0	81.4	85.1	86.5	86.9
SCROP [9]	59.0	74.8	78.1	81.1	-	-
GLN [8]	63.2	77.5	83.4	89.1	92.1	93.2
RetroXpert	62.1	75.8	78.5	80.9	82.8	83.5
Reaction type unknown on USPTO-50K						
RetroSim [61]	37.3	54.7	63.3	74.1	82.0	85.3
NeuralSym [7]	44.4	65.3	72.4	78.9	82.2	83.1
SCROP [9]	43.7	60.0	65.2	68.7	-	-
GLN [8]	52.6	68.0	75.1	83.1	88.5	92.1
RetroXpert	50.4	61.1	62.3	63.4	63.9	64.0
Retrosynthesis results on USPTO-full.						
GLN* [8]	39.0	50.1	55.3	61.3	65.9	69.1
SCROP* [9]	45.7	60.7	65.3	70.1	73.3	76.0
RetroXpert	49.4	63.6	67.6	71.6	74.6	77.0

While our RetroXpert is currently designed to find the best set of reactants. To increase the diversity, we can design new strategies to enumerate multiple reaction centers for each product. This is left as the feature work.

2.5 Large scale experiments

To demonstrate the scalability of our method, we also experiment on the USPTO-full dataset, which consists of 760K training data. We extract 75,129 semi-templates and keep only 3,788 ones that appear at least 10 times. We set N_{max} as 5 to cover 99.87% training data. We obtain 1.35M training data after reversing synthons. The final accuracy of the $P \rightarrow S$ on training set is 60.5%, and there are 0.3M unsuccessful synthon data and the total RGN training data size is 1.65M. We train the RGN for 500,000 time steps on USPTO-full while keeping the other settings the same as those in section 2.4. We run the official implementation of GLN following their instructions [8], as well as the self-implemented SCROP [9] on the USPTO-full dataset. Experimental results are reported at the bottom of Table 3.1. Our method again significantly outperforms the SCROP and GLN, which demonstrates that our model scales well to the large real-world dataset. Note that both template-free methods SCROP and RetroXpert outperform the GLN significantly, which may indicate the scalability of template-based methods is very limited.

2.6 Prediction visualization

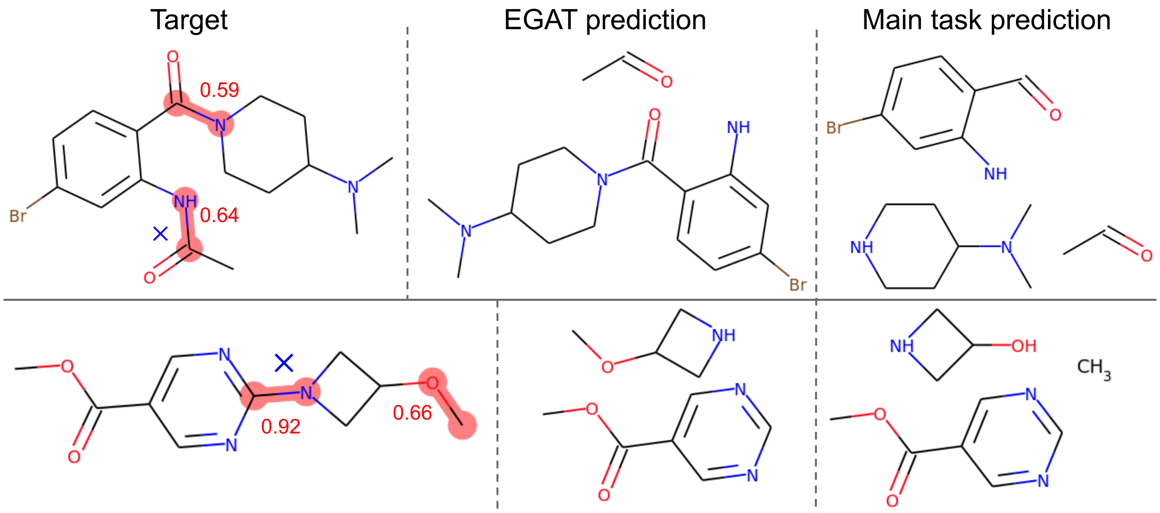

Figure 2.5: Importance of the auxiliary task. Pink indicates the reaction center along with disconnection probability predicted by the EGAT main task. Blue cross indicates the ground truth disconnection. Our EGAT successfully finds the desired reaction center under the guidance of the auxiliary task.

For EGAT, how the auxiliary task helps to identify the reaction center is illustrated in Figure 2.5. Note that in the first example the two colored bonds and their surrounding structures are very similar. Current shallow GNNs consider only local information and fails to distinguish the true reaction center. Under the guidance of the auxiliary task, EGAT is able to identify the true reaction center. Figure **??** demonstrates the robustness of our method. Even if the predicted synthons are different from the ground truth, the RGN still successfully generates desired reactants.

2.7 Ablation study of atom features

Our method can also work without semi-templates. When removing semi-templates, the EGAT performance drops slightly as listed in Table 2.9. The semi-templates feature is not a must component of our method, but it is definitely helpful for finding the reaction center.

Table 2.9: Results of atom features ablation study. *Aux* is the short for auxiliary. *EGAT* consists of both main and auxiliary tasks. The prediction is binarized with a threshold of 0.5 if the main task alone.

Type	Semi-templates	Accuracy (%)		
		Main	Aux	EGAT
✓	✗	70.0	**99.2**	84.0
✓	✓	**74.4**	**99.2**	**86.0**
✗	✗	43.3	83.8	59.9
✗	✓	**51.5**	**86.4**	**64.9**

2.8 Top-1 and Top-2 predictions

About 10% Top-1 predictions by our model have been considered as wrong predictions while the associated Top-2 predictions are the same to the ground-truth.

However, 9 in 10 of these Top-1 predictions are re-considered as reasonable and valid predictions checked by experienced chemists from the synthetic chemistry perspective. As Figure 2.6 shows, the major retro-predictions that both Top-1 and Top-2 can be thought correct, are among metal-catalyzed cross-coupling reactions, N- and O-alkylation reactions, saponification of ethyl esters and methyl esters, different sources of reactants, esterification of alcohol with acyl chlorides or carboxylic acid, and deprotection of different protecting groups to same alcohols.

There are some deprotection reactions with different protecting groups, such as deprotecting O-THP ether and O-Bn ether to free alcohol in Figure 2.6(a). They are prevalent strategies in chemistry utilizing different protecting groups. In Figure 2.6(b), both bromoarenes and iodoarenes are reactive enough to initiate Suzuki coupling reactions, similar to N- and O-alkylation of propargyl like or benzyl chloride and bromide in Figure 2.6(c). In Figure 2.6(d), hydrolysis of ethyl ester and methyl ester to corresponding carboxylic acid can both occur under certain conditions, although saponification of methyl ester is faster than ethyl ester. Real reactants that participated in the reactions are predicted in our Top-1 predictions, such as allyl Grignard reagent and acyl chloride in cases shown in Figure 2.6(e). Last but not least, in Figure 2.6(f), methyl boronic acid or its trimer form and trimethyl borate are very common reagents used by chemists in Suzuki coupling reaction to introduce methyl group.

2.9 Discussion

One major common limitation of current retrosynthesis work is the lack of reasonable evaluation metrics. There may be multiple valid ways to synthesize a product, while the current evaluation metric considers only the given reaction. More evaluation metrics should be proposed in the future. Different evaluation metrics such

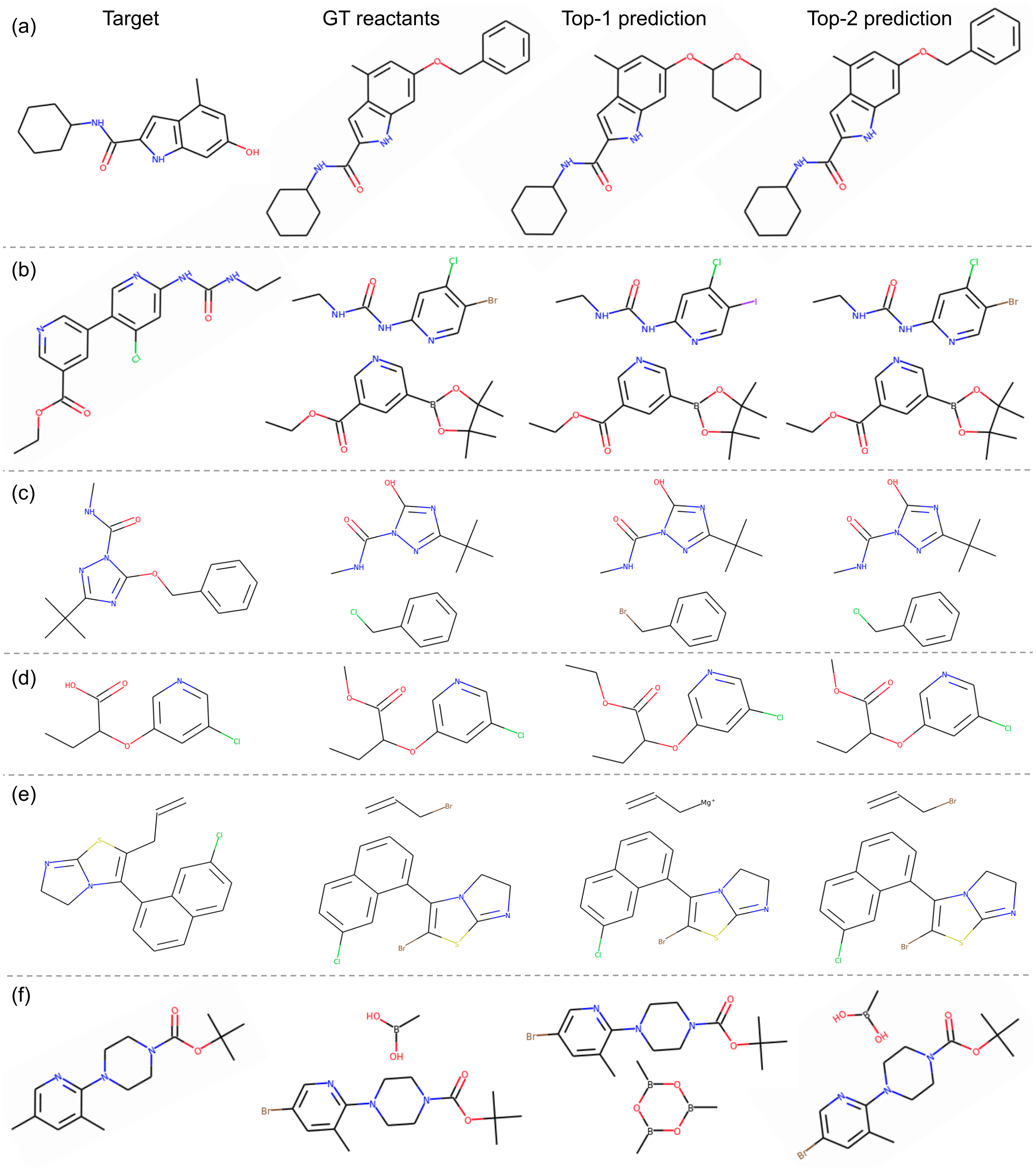

Figure 2.6: Top-1 and Top-2 predictions are both reasonable reactants.

as coverage and round trip accuracy are proposed in [81], which is a good start, but there is still a long way to go.

CHAPTER 3

Template-based Retrosynthesis Prediction by Composing Templates

In this chapter, we will study template-based retrosynthesis prediction. Existing template-based retrosynthesis methods follow a template selection stereotype and suffer from the limited training templates, which prevents them from discovering novel reactions. To overcome the limitation, we propose an innovative retrosynthesis prediction framework that can compose novel templates beyond training templates. So far as we know, this is the first method that uses machine learning to compose reaction templates for retrosynthesis prediction. Besides, we propose an effective reactant candidates scoring model that can capture atom-level transformations, which helps our method outperform previous methods on the USPTO-50K dataset. Experimental results show our method can produce novel templates for 15 USPTO-50K test reactions that are not covered by training templates.

3.1 Introduction

Retrosynthesis plays a significant role in the organic synthesis planning, in which target molecules are recursively decomposed into available commercial building blocks. This analysis mode was firstly formulated in the pioneering work [52, 53] and now is one of the fundamental paradigms in the modern chemical society. As the development of deep learning and its applications [82, 83, 84, 85, 86], numerous retrosynthesis prediction algorithms have been proposed to aid or even automate the retrosynthesis analysis. However, the performance of existing methods is still not satisfactory. The massive search space is one of the major challenges of retrosynthesis considering that the order of 10^7 compounds and reactions [54] have been reported in synthetic-organic knowledge. The other challenge is that there are often multiple viable retrosynthesis pathways and it is challenging to decide the most appropriate route since the feasibility of a route is often compounded by several factors, such as reaction conditions, reaction yield, potential toxic byproducts, and the availability of potential reactants [13].

Most of existing machine-learning empowered retrosynthesis methods focus on the single-step version. These methods are broadly grouped into template-based and template-free major categories. Templates-free methods [55, 9, 13, 64, 87, 88] usually rely on deep learning models to directly generate reactants. One effective strategy is to formulate the retrosynthesis prediction as a sequence translation task, and generate SMILES [25] sequences directly using sequence-to-sequence models such as Seq2Seq [55], SCROP [9], and AT [89]. SCROP [9] proposes to use a second Transformer to correct the initial wrong predictions. Translation-based methods are simple and effective, but lack interpretability behind the prediction. Another representative paradigm is to first find a reaction center and the target is split accordingly to obtain hypothetical units named synthons, and then generate reactants incrementally from these synthons such as RetroXpert [13], G2Gs [64], RetroPrime [90], and GraphRetro [91].

On the other hand, template-based methods are receiving less attention as the rapid surge of template-free methods. Template-based methods conduct retrosynthesis based on either hand-encoded rules [60] or automatically extracted retrosynthesis templates [61]. Templates encode the minimal reaction transformation patterns, and are straightforwardly interpretable. The key procedure is to select applicable templates to apply to targets [61, 7, 62, 8]. Template-based methods have been criticised for the limitation that they can only infer reactions covered by training templates and can not discover novel reactions [66, 13].

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

3.2 Related Work

Recently there has been an increasing number of work using machine learning methods to solve the retrosynthesis problem. These methods can be categorized into template-based [61, 7, 62, 92, 8] and template-free approaches [55, 64, 13, 91, 93]. Template-based methods extract templates from training data and build models to learn the corresponding relationship between products and templates. RetroSim

[61] selects the templates based on the fingerprint similarity between products and reactions. NeuralSym [7] uses a neural classification model to select corresponding templates. However, this method does not scale well with increasing number of templates. To mitigate the problem, [92] adopts a multi-scale classification model to select templates according to a manually defined template hierarchy. GLN [8] proposes a graph logic network to model the decomposed template hierarchy by first selecting reaction centers within the targets and then only consider templates that contain the selected reaction centers. The decomposition strategy can reduce the search space significantly. GLN models the relationship between reactants and templates jointly by applying selected templates to obtain reactants which are also used to optimize the model simultaneously.

Template-free methods do not rely on retrosynthesis templates. Instead, they construct models to predict reactants from products directly. Translation based methods [9, 89, 94, 95] use SMILES to represent molecules and treat the problem as a sequence-to-sequence task. MEGAN [87] treats the retrosynthesis problem as a graph transformation task, and train the model to predict a sequence of graph edits that can transform the product into the reactants. To imitate a chemist's approach to the retrosynthesis, two-step methods [64, 13, 90, 91] first perform reaction center recognition to obtain synthons by disconnecting targets according to the reaction center, and then generate reactants from the synthons. G2Gs [64] treats the reactant generation process as a series of graph editing operations and utilizes a variational graph generation model to implement the generation process. RetroXpert [13] converts the synthon into SMILES to generate reactants as a translation task. GraphRetro [91] also adopts a similar framework and generates the reactants by attaching leaving groups to synthons. Dual [88] proposes a general energy-based model framework that integrates both sequence- and graph-based models, and performs consistent training over forward and backward prediction directions.

3.3 Preliminary Knowledge

3.3.1 Retrosynthesis and Template

Single-step retrosynthesis is to predict a set of reactant molecules given a target product as shown in Figure 3.1(a). Note that the product and reactant molecules are atom-mapped, which ensures that every product atom is uniquely mapped to a reactant atom. Templates are reaction rules extracted from chemical reactions. They are composed by reaction centers and encode the atom and bond transformations during the reaction process. The illustrated template in Figure 3.1(b) consists of a product subgraph (upper) and reactant subgraphs (lower). The subgraph patterns are highlighted in pink within the corresponding molecule graphs.

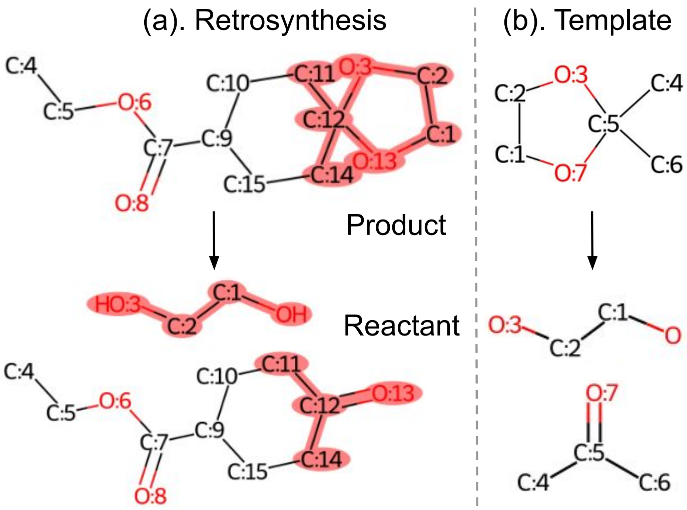

Figure 3.1: A retrosynthesis example from USPTO-50K dataset and its template. Note that the product and reactant are atom-mapped. The product and reactant subgraphs in (b) are highlighted in pink within the product and reactant molecule graphs in (a), respectively.

3.3.2 Molecule Graph Representation

The graph representation of a molecule or subgraph pattern is denoted as $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of graph nodes (atoms) and edges (bonds), respectively. Following previous work [8, 13], each bond is represented as two directed edges. Initial node and edge features can be easily collected for the learning purpose.

3.3.3 Graph Attention Networks

Graph Neural Networks [72] are especially good at learning node- and graph-level embeddings of molecule data. In this work, we adapt the Graph Attention Networks (GATs) [73] to incorporate bond features. The GAT layer updates a node embedding by aggregating its neighbor's information. The modified GAT concatenates edge embeddings with the associated incoming node embeddings before each graph message passing. The input of the GAT layer is node embeddings $\{v_i|\forall i \in \mathcal{V}\}$ and edge features $\{e_{i,j}|(i,j) \in \mathcal{E}\}$, and the output updated node embeddings $\{v_i'|\forall i \in \mathcal{V}\}$. Each node embedding is updated with a shared parametric function t_θ:

$$v_i' = t_\theta(v_i, \text{AGGREGATE}(\{[v_j||e_{i,j}]|\forall j \in \mathcal{N}(i)\})), \tag{3.1}$$

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

$$c(i,j) = \text{LeakyReLU}(w^T[\boldsymbol{W}_1 v_i||\boldsymbol{W}_1 v_j||\boldsymbol{W}_2 e_{i,j}]), \tag{3.2}$$

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

$$\alpha(i,j) = \text{Softmax}_j(c(i,j)) = \frac{\exp(c(i,j))}{\sum_{j' \in \mathcal{N}(i)} \exp(c(i,j'))}. \tag{3.3}$$

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

$$v_i' = \sigma(\sum_{j \in \mathcal{N}(i)} \alpha(i,j) * \boldsymbol{W}_3[\boldsymbol{W}_1 v_j || \boldsymbol{W}_2 e_{i,j}]). \tag{3.4}$$

GAT is usually stacked by multiple layers and enhanced with multi-head attention [68]. Please refer to [73] for more details.

3.3.4 Graph-level Embedding

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

$$\text{emb}_G = \text{READOUT}(\{v_{i,1}||v_{i,2}||...||v_{i,L}|\forall i \in \mathcal{V}\}). \tag{3.5}$$

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

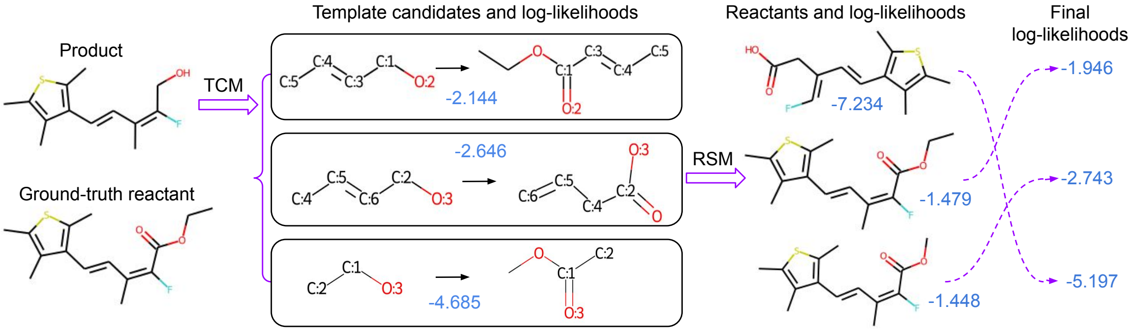

Figure 3.2: The overall pipeline of our proposed method. Given the desired product as shown at the top left, single-step retrosynthesis is to find the ground-truth reactant as shown at the bottom left. Numbers indicated in blue are the corresponding log-likelihoods of our models, and the log-likelihoods of template composer model (TCM) and reactant scoring model (RSM) are combined to get the final ranking of the reactants. In this example, combining log-likelihoods of TCM and RSM helps to find the correct Top-1 reactant.

3.4 Methods

We propose to compose retrosynthesis templates from a predefined set of template building blocks, and then these composed templates are applied to target products to obtain the associated reactants. Unlike previous template-based methods [61, 7, 62, 8] only selecting from training templates, our method can discover novel templates which are beyond the training templates. To further improve the retrosynthesis prediction performance, we design a scoring model to evaluate the suitability of product and candidate reactants pair. The scoring procedure acts as a verification step, and it plays a significant role in our method.

The overall pipeline of our method is shown in Figure 3.2. Our method tackle retrosynthesis in two stages. The first stage is to compose retrosynthesis templates with a TCM, which composes retrosynthesis templates by selecting template building blocks and then assembling them. In the second stage, the obtained templates are applied to the target product to generate associated reactants. After that, we utilize a powerful RSM to evaluate the generated reactants for each product. During evaluation, the probability scores of both stages are linearly combined to rank Top-K reactants prediction. In following sections, we will detail each stage of our method.

3.4.1 Compose Retrosynthesis Templates

The template-based retrosynthesis methods are criticized for their limitation not generalizing to unseen reactions, since all existing template-based methods follow the similar procedure to select applicable templates from the extracted training ones. To overcome the above limitation, we propose a different pipeline to find template candidates. As illustrated in Figure 3.3, our method first selects product and reactant subgraphs sequentially from the corresponding subgraph vocabularies, which is detailed in section 3.4.1.1. Then these selected subgraphs are assembled into templates with properly assigned atom mappings as detailed in section 3.4.1.4. As far as we know, this is the first attempt to compose retrosynthesis templates instead of simple template selection. During evaluation, beam search algorithm [80] is utilized to find Top-K predicted templates. Reactants can be obtained by applying templates to the target molecule.

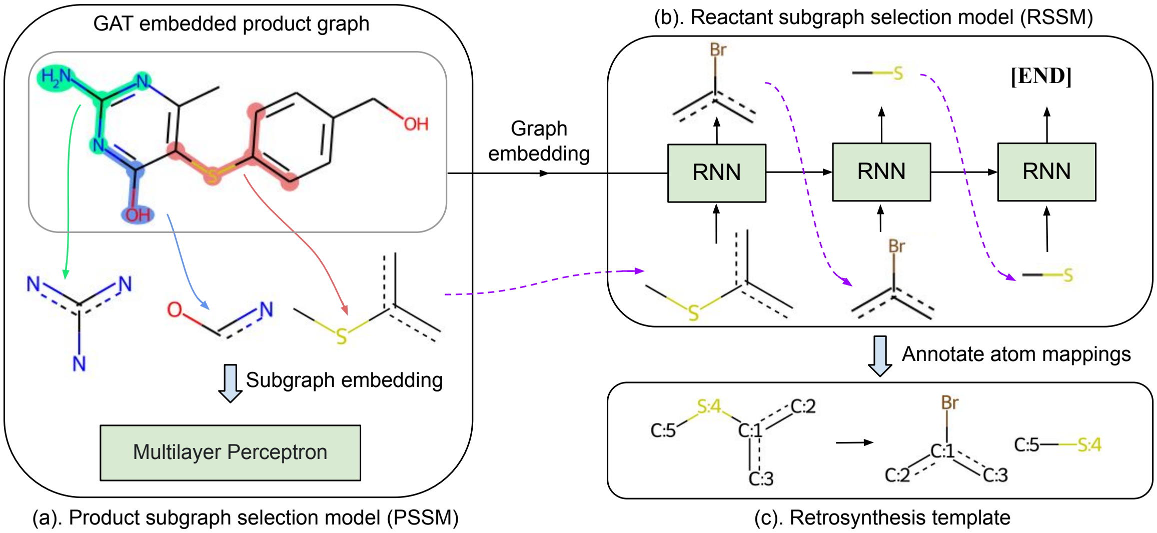
GAT embedded product graph
Graph embedding
Subgraph embedding
Multilayer Perceptron
(a). Product subgraph selection model (PSSM)
(b). Reactant subgraph selection model (RSSM)
RNN
RNN
RNN
[END]
Annotate atom mappings
(c). Retrosynthesis template
C:5
S:4
C:1
C:2
C:3
Br

Figure 3.3: The workflow of our template composer model: (a) selecting a proper product subgraph from product subgraph candidates with PSSM, (b) selecting reactant subgraphs sequentially from reactant subgraph vocabulary with RSSM, and (c) annotating atom mappings between the product and reactant subgraphs to obtain a template.

3.4.1.1 Subgraph Selection

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

3.4.1.2 Product Subgraph Selection

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

Each product molecule graph G_p contains only a limited set of candidate subgraphs $\mathcal{F}_c$ predefined in the vocabulary $\mathcal{F}_P$. Three candidate subgraphs are illustrated in Figure 3.3(a). The candidate subgraphs for each target can be obtained in offline by checking the existence of every product subgraph from $\mathcal{F}_P$ in the product graph G_p. Therefore, we only need to consider the candidate subgraphs $\mathcal{F}_c$ to guide the selection process [8] when selecting a product subgraph. Here $\mathcal{F}_a \subseteq \mathcal{F}_c \subseteq \mathcal{F}_P$ since the candidate subgraphs $\mathcal{F}_c$ must contain all applicable subgraphs.

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

$$L_{\text{PSSM}} = -\log \frac{\arg\min_{f \in \mathcal{F}_a} o_f}{\arg\min_{f \in \mathcal{F}_a} o_f + \sum_{f \in \mathcal{F}_c \setminus \mathcal{F}_a} o_f}, \tag{3.6}$$

where o_f is the exponential of PSSM output logits for subgraphs in $\mathcal{F}$, $|\mathcal{F}|$ is the size of $\mathcal{F}$, and $\setminus$ is set subtraction. In the above loss function, the numerator is the minimal exponential output for all applicable subgraphs in $\mathcal{F}_a$, which is considered as the ground-truth class proxy in the Softmax classifier. The extra item in denominator is the summation of exponential output of all inapplicable subgraphs in $\mathcal{F}_c$. The intuition is that we always optimize the PSSM to increase the prediction probability for the least probable applicable subgraph, so the model is driven to generate large scores for all applicable subgraphs $\mathcal{F}_c$ while considering interrelationships of candidate subgraphs. The novel loss outperforms BCE loss in our experiments. Detailed experimental comparison results between the proposed loss function Equation (3.6) and BCE loss can be found in the Table 3.2.

PSSM scores candidate subgraphs $\mathcal{F}_c$ based on their subgraph embeddings. As shown in Figure 3.3(a), to obtain subgraph embeddings, the nodes of product molecule graph G_p are first encoded with the modified GAT that is detailed in section 3.3.3. The embedding emb_f of the subgraph f is gathered as the average embedding of subgraph f associated nodes in G_p, and then these embeddings are fed into a multilayer perceptron (MLP) for subgraph selection. Here for a subgraph f, the READOUT function is implemented as the arithmetic average for its simplicity and efficiency. Note that this is different from GLN [8] in which product graph and candidate subgraphs are considered as separate graphs and embedded independently. Our strategy to reuse node embeddings is more efficient and can learn more informative subgraph embedding since the neighboring structure of a subgraph is also incorporated during message passing procedure of GAT. Besides, our method can naturally handle multiple equivalent subgraphs situation in which the same subgraph appears multiple times within the product graph.

3.4.1.3 Reactant Subgraph Selection

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

Since the number of reactant subgraphs is undetermined, we build the reactant subgraph selection model based on the recurrent neural network (RNN) as illustrated in Figure 3.3(b), and formulate reactant subgraph selection as the sequence generation. The hidden state of RNN is initialized from the product graph embedding emb_{G_p} as defined in Equation (3.5) to explicitly consider the target product, and the start token is the product subgraph f_p selected in the previous procedure (Section 3.4.1.2), as well as an extra end token [END] is appended to reactant subgraph sequence f_r. At each time step, the RNN output is fed into a MLP for the token classification. For the start token f_p, we reuse product subgraph embeddings obtained previously (Section 3.4.1.2) since we find it provides better performance than embedding the token in the traditional one-hot embedding manner.

3.4.1.4 Annotate Atom Mappings

Given f_p and f_r, the final step is to annotate the atom mappings between f_p and f_r to obtain the retrosynthesis template as shown in Figure 3.3(c). A subgraph pattern f can also be represented in the SMARTS string, and we use open source toolkit

Indigo[1]'s automap() function to build atom mappings. We empirically find about 70% of USPTO-50K training templates can be successfully annotated with correct atom mappings. To remedy this deficiency, we keep a memo of training templates and associated f_p and f_r. During evaluation, the predicted f_p and f_r are processed with automap() if not found in the memo.

3.4.2 Score Predicted Reactants

After a retrosynthesis template is composed, reactants can be easily obtained by applying the template to the target using RunReactants from RDKit [44] or run_reaction() function from RDChiral [78]. To achieve superior retrosynthesis prediction performance, it is important to verify that the predicted reactants can generate the target successfully. The verification is achieved by scoring reactants and target pair, which is formulated as a multi-class classification task where the true reactant set is the ground-truth class.

To serve the verification purpose, we build a reactant scoring model based on the modified GAT. Product molecule graph G_p and reactant molecule graph G_r are first input into a GAT to learn atom embeddings. Since the target and generated reactants are atom-mapped as in Figure 3.1(a), for each atom in G_p we can easily find its associated atom in G_r. Inspired by WLDN [71], we define a fusion function $\mathrm{F}(n_a^p, n_{a'}^r)$ to combine embeddings of a product atom a and its associated reactant atom a':

$$\mathrm{F}(n_a^p, n_{a'}^r) = \boldsymbol{W}_4(n_a^p - n_{a'}^r)||\boldsymbol{W}_5(n_a^p + n_{a'}^r), \tag{3.7}$$

[1] https://github.com/epam/Indigo

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

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

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

Our scoring model is advantageous since it operates on atom-level embeddings and is sensitive to local transformations between the product and reactants, while existing method GLN [8] takes only molecule-level representations as the input. So GLN can not capture atom-level transformations and has a weaker distinguishing ability.

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

$$\begin{aligned} &\lambda * \log(\mathcal{P}(\mathcal{T}|P)) + (1-\lambda) * \log(\mathcal{P}(R|P)) \\ =&\log(\mathcal{P}(\mathcal{T}|P)^{\lambda} * \mathcal{P}(R|P)^{1-\lambda}), \end{aligned} \tag{3.8}$$

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

3.5 Experiment and Results

3.5.1 Dataset and Preprocessing

Our method is evaluated on the standard benchmark dataset USPTO-50K [67] under two settings (with or without reaction types) to demonstrate its effectiveness. USPTO-50K is derived from USPTO granted patents [76], and it is composed of 50K reactions annotated with 10 reaction types. We split reaction data into training/validation/test sets into 8:1:1 in the same way as previous work [61, 8]. Since the original annotated mapping numbers in the USPTO dataset may result in unexpected information leakage[2], we first preprocess the USPTO reactions to re-assign product mapping numbers according to its canonical atom order as suggested by RetroXpert [13]. The atom and bond features are similar to the previous work [13], and reaction types are converted into one-hot vectors concatenated with the original atom features.

Following RetroXpert [13], we extract templates from training reactions using RDChiral [78]. We can obtain 10386 unique templates in total for the USPTO-50K training data, and 94.08% of test reactions are covered by these training templates. The gathered templates are split into product and reactant subgraphs of which mapping numbers are further removed to obtain the subgraph vocabularies $\mathcal{F}_P$ of size 7766 and $\mathcal{F}_R$ of size 4391.

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

For each target molecule, we find its candidate subgraphs $\mathcal{F}_c$ using graph matching algorithms and applicable templates by checking if the ground-truth reactant can be obtained when each training template is applied to the target. The applicable subgraphs $\mathcal{F}_a$ then can be obtained easily from the acquired applicable templates. Since the exact graph matching process might be time-consuming, we extract the fingerprint for each molecule/sub-molecule to filter those impossible subgraphs. For the subgraph screening purpose, we adopt the PatternFingerprint from RDKit and use a fingerprint size of 1024.

3.5.2 Evaluation

Following previous methods [8, 13], we use beam search [80] to find Top-50 template predictions during evaluation, which are applied to targets to collect candidate reactants. Collected reactants and targets are the experimental data for RSM. Predicted reactants are finally ranked according to the combined log-likelihood of TCM and RSM. The evaluation metric for retrosynthesis prediction is the Top-K exact match accuracy, which is the percentage of reactions where the ground truth reactant set is within the top K predictions.

3.5.3 Implementation

Our model is implemented using PyTorch [96] and PyTorch Geometric [97]. The adapted GAT model is built based on the source implementation of Pretrain-GNN [98]. The TCM model is composed of a modified GAT and a simple RNN model. The embedding dimension is set as 300 for all embeddings for simplicity. The number of GAT layers is 6. We adopt GRU [26] as the RNN implementation in TCM, the number of GRU layers is 2 and both its embedding and hidden size are 300. We add a self-loop to each graph node following [8, 13]. We use the Parametric Rectified

Linear Unit (PReLU) [99] comprehensively as the activation function in our model. We replace the original Batch Normalization [100] layer with Layer Normalization [101] layer after each GAT layer, since we find Layer Normalization provides better performance in our experiments. We adopt Equation (3.5) as the graph READOUT operation. A simple MLP is applied to product subgraph embeddings to select the proper product subgraph. The MLP is composed of two linear layers and the PReLU activation function is placed between the two linear layers. We also use a Dropout [102] layer with a dropout rate of 0.3 in the MLP.

The RSM model is composed of two GATs and a MLP head, and the GAT uses the same settings as in the TCM except that each GAT is composed of 3 layers. Product and reactant graphs are embedded with the first GAT model. Note that for reactions with multiple reactants, we regard the disconnected molecule graphs as a single large graph. Once obtaining the fused atom embeddings, the new product molecule graphs with fused atom embeddings are input into the second GAT. The composition of the MLP head is similar to that in TCM. The RSM model is also trained in multi-process mode for acceleration.

Both TCM and RSM are optimized with Adam [40] optimizer with default settings, and the initial learning rate is 0.0003 and 0.00005 for TCM and RSM, respectively. The learning rate is adjusted with CosineAnnealingLR scheduler during training. The models is trained in multi-process mode on a single GTX 1080 Ti GPU for acceleration. TCM is trained with batch size 32 and it only takes about two hours to train TCM for 80 epochs. RSM training costs about 6 hours for 20 epochs. The final model parameters are saved and loaded later for inference. We repeat all experiments for three times and report the mean performance in default. We find our model is quite robust to the hyper-parameters, and most of the model settings are copied from [98] as they are given. We slightly tune the model hyper-parameters such as learning rate and batch size to achieve the best results.

3.5.4 Main Results

We decide the optimal value of λ according to validation performance. Specifically, we set λ as 0.4 for both experimental settings (with/without reaction types). We use these optimal settings in all experiments unless explicitly stated. Detailed ablation study about λ are included in the 3.5.4.3.

Table 3.1: Retrosynthesis evaluation results (%) on USPTO-50K. Existing methods are grouped into two categories. Our method RetroComposer belongs to the template-based methods. The best results in each column are highlighted in bold. RetroXpert* results have been updated by the authors in their GitHub repository[2].

Methods	Without reaction types				With reaction types			
	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10
Template-free methods								
SCROP [9]	43.7	60.0	65.2	68.7	59.0	74.8	78.1	81.1
G2Gs [64]	48.9	67.6	72.5	75.5	61.0	81.3	86.0	88.7
MEGAN [87]	48.1	70.7	78.4	86.1	60.7	82.0	87.5	**91.6**
RetroXpert* [13]	50.4	61.1	62.3	63.4	62.1	75.8	78.5	80.9
RetroPrime [90]	51.4	70.8	74.0	76.1	64.8	81.6	85.0	86.9
AT [89]	53.5	-	81.0	85.7	-	-	-	-
GraphRetro [91]	53.7	68.3	72.2	75.5	63.9	81.5	85.2	88.1
Dual [88]	53.6	70.7	74.6	77.0	65.7	81.9	84.7	85.9
Template-based methods								
RetroSim [61]	37.3	54.7	63.3	74.1	52.9	73.8	81.2	88.1
NeuralSym [7]	44.4	65.3	72.4	78.9	55.3	76.0	81.4	85.1
GLN [8]	52.5	69.0	75.6	83.7	64.2	79.1	85.2	90.0
Ours	**54.5**	**77.2**	**83.2**	**87.7**	**65.9**	**85.8**	**89.5**	91.5
TCM only	49.6	71.7	80.8	86.4	60.9	82.3	87.5	90.9
RSM only	51.8	75.7	82.4	87.3	64.3	84.8	88.9	91.4

3.5.4.1 Retrosynthesis Prediction Performance

We compare our RetroComposer with existing methods on the standard benchmark dataset USPTO-50K, and report comparison results in Table 3.1. Results of RetroXpert have been updated by the authors[2]. For both evaluation settings (with or without reaction types), our method outperforms previous methods by a significant margin in seven out of eight compared Top-K metrics.

Specially, our RetroComposer achieves 54.5% Top-1 accuracy without reaction types, which improves the previous best template-based method GLN [8] significantly by 2.0% and also outperforms existing SOTA template-free methods Dual [88] and GraphRetro [91]. Besides, our method achieves 77.2% Top-3 accuracy which improves the Top-3 accuracy 70.8% of RetroPrime [90] by 6.4%, and 87.7% Top-10 accuracy which improves the Top-10 accuracy 85.7% of AT [89] by 2.0%.

When reaction types are given, our method also obtains the best Top-1 accuracy 65.9% among all methods and outperforms GLN by 1.7%. Compared with template-free methods GraphRetro and Dual, our method outperforms the SOTA Dual (65.7%) by 0.2% and outperforms GraphRetro significantly by 2.0% in Top-1 accuracy. As for the Top-10 accuracy, our method achieves 91.5%, which is slightly lower than 91.6% of MEGAN [87].

As the ablation study, we report results with only TCM or RSM. With only either TCM or RSM, the model performance is largely degraded. Without reaction types, TCM only achieves 49.6% Top-1 accuracy while RSM only 51.8%. With reaction types, TCM only achieves 60.9% Top-1 accuracy while RSM only 64.3%. Since TCM and RSM scores retrosynthesis from different perspectives and are complementary, their results can be combined to achieve the best performance. Particularly, our method achieves 54.5% and 65.9% Top-1 accuracy when combining TCM and RSM according to Equation (3.8).

The superior performance demonstrates the effectiveness of our method. Particularly, the superiority of our method is more significant in real world applications where reaction types are unknown. What is more, our Top-10 accuracy is already quite high.It indicates that our method can usually find the best reactant set for the target in a few candidates. This is especially important for multi-step retrosynthesis scenario, in which the number of predicted reaction paths may grow exponentially with the retrosynthesis path length.

3.5.4.2 Ablation Study of PSSM Loss

We experimentally show that our proposed loss function Equation (3.6) for PSSM outperforms the BCE loss. For all ablation experiments, we find the optimal value of hyper-parameter λ independently and report the best results for a fair comparison. The comprehensive experimental results are reported in Table 3.2.

Without given reaction types, our method with Equation (3.6) as PSSM loss achieves the best Top-1 and Top-3 accuracy results, outperforming the BCE loss in Top-1 and Top-3 accuracy by 1.4% and 1.5%, respectively. With known reaction types, our method with Equation (3.6) as PSSM Loss outperforms BCE loss by 0.6% in Top-1 accuracy. While BCE loss can achieve better Top-5 and Top-10 results in both settings, our proposed loss function Equation (3.6) can achieve better Top-1 accuracy. The retrosynthesis prediction emphasizes more Top-1 accuracy, therefore we adopt Equation (3.6) as the PSSM loss in our method.

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

Table 3.2: Ablation study results (%) of two differentPSSMloss functions: our proposed Equation (3.6) and BCE.

Types	L_{PSSM}	Methods	Top-1	Top-3	Top-5	Top-10
Wo/	Eq. (3.6)	Ours	**54.5**	**77.2**	83.2	87.7
		TCM only	49.6	71.7	80.8	86.4
		RSM only	51.8	75.7	82.4	87.3
	BCE	Ours	53.1	77.1	**83.8**	**89.2**
		TCM only	46.5	69.9	78.5	86.9
		RSM only	51.2	75.7	82.9	88.6
W/	Eq. (3.6)	Ours	**65.9**	85.8	89.5	91.5
		TCM only	60.9	82.3	87.5	90.9
		RSM only	64.3	84.8	88.9	91.4
	BCE	Ours	65.3	**85.9**	**90.3**	**92.6**
		TCM only	58.5	81.8	87.6	91.5
		RSM only	64.2	85.4	89.6	92.4

3.5.4.3 Ablation study of hyper-parameter λ

We conduct the ablation study of λ and report results in Table 3.3, when $\lambda = 0.4$ the best Top-1 accuracy is achieved for the both settings. Note that with only RSM ($\lambda = 0$), the Top-1 accuracy 64.3% already outperforms the previous best template-based method GLN of 63.2% [8] with given reaction types. This demonstrates the effectiveness of our RSM. While with only TCM ($\lambda = 1.0$), the performance has an appreciable gap with the existing methods. In our method, each generated set of subgraphs may have multiple associated templates due to the uncertainty of product subgraphs and atom transformations. Therefore there may be multiple top-tier predictions that can not be distinguished with only TCM. With a little help from RSM ($\lambda = 0.9$), these top-tier predictions can be differentiated and the Top-1 accuracy is significantly boosted.

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

Table 3.3: Top-1 accuracy (%) with different λ values.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Wo/ types	51.8	53.3	53.9	**54.5**	**54.5**	54.4	54.1	53.6	53.0	52.3	49.6
W/ types	64.3	65.2	65.6	65.7	**65.9**	**65.9**	65.6	65.1	64.7	64.4	60.9

3.5.4.4 Novel Templates

Different from existing methods, our method can find novels templates that are not in training data. Our model predicts different templates based on different possible reaction centers for a given target. For example, an amide formation template and alkylation template may both be applied in the same target molecule, and our model can predict suitable templates very well and give reasonable corresponding reactants for such cases. For 5.92% of test reactions that are not covered by training templates, our algorithm can predict relevant templates very well for most of reaction types, although it fails in some heterocyclic formation reactions. This is because there are very few of such reaction data in USPTO-50K. Particularly, our method successfully discovers chemically valid templates for 15 uncovered test reactions, which confirms that our method can find novel reactions. Two of such examples are illustrated in Figure 3.4.

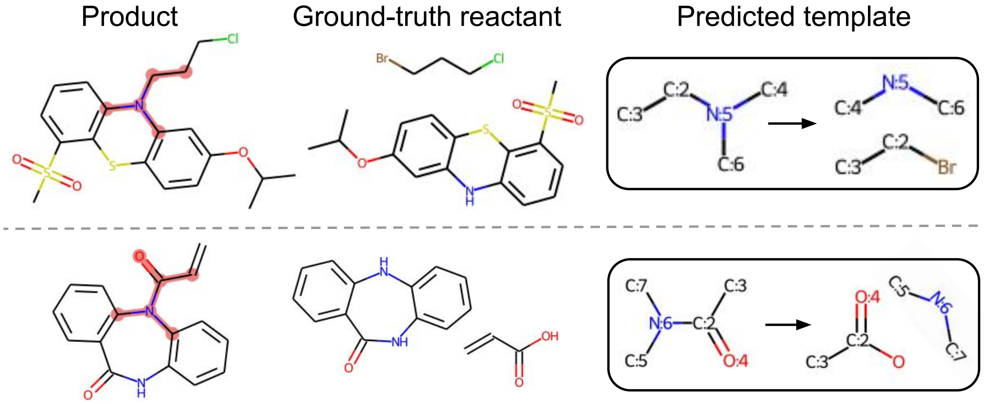

Figure 3.4: Our method successfully finds valid templates for two test reactions that are not covered by training data. The matched product subgraphs are highlighted in pink for better visualization.

3.6 Discussion and conclusion

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

REFERENCES

[1] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder," in *Proceedings of the 34th ICML-Volume 70*. JMLR. org, 2017, pp. 1945–1954.

[2] H. Dai, Y. Tian, B. Dai, S. Skiena, and L. Song, "Syntax-directed variational autoencoder for structured data," *arXiv preprint arXiv:1802.08786*, 2018.

[3] M. Simonovsky and N. Komodakis, "Graphvae: Towards generation of small graphs using variational autoencoders," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 412–422.

[4] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *ICML*, 2018, pp. 2328–2337.

[5] B. Samanta, A. De, G. Jana, V. Gómez, P. K. Chattaraj, N. Ganguly, and M. Gomez-Rodriguez, "Nevae: A deep generative model for molecular graphs," *Journal of machine learning research. 2020 Apr; 21 (114): 1-33*, 2020.

[6] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, *et al.*, "Analyzing learned molecular representations for property prediction," *Journal of chemical information and modeling*, vol. 59, no. 8, pp. 3370–3388, 2019.

[7] M. H. Segler and M. P. Waller, "Neural-symbolic machine learning for retrosynthesis and reaction prediction," *Chemistry–A European Journal*, vol. 23, no. 25, pp. 5966–5971, 2017.

[8] H. Dai, C. Li, C. Coley, B. Dai, and L. Song, “Retrosynthesis prediction with conditional graph logic network,” in *Advances in Neural Information Processing Systems*, 2019, pp. 8870–8880.

[9] S. Zheng, J. Rao, Z. Zhang, J. Xu, and Y. Yang, “Predicting retrosynthetic reactions using self-corrected transformer neural networks,” *Journal of Chemical Information and Modeling*, 2020.

[10] C. Yan, S. Wang, J. Yang, T. Xu, and J. Huang, “Re-balancing variational autoencoder loss for molecule sequence generation,” in *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2020, pp. 1–7.

[11] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, and e. al, “Automatic chemical design using a data-driven continuous representation of molecules,” *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.

[12] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek, “Estimation of the size of drug-like chemical space based on gdb-17 data,” *Journal of computer-aided molecular design*, vol. 27, no. 8, pp. 675–679, 2013.

[13] C. Yan, Q. Ding, P. Zhao, S. Zheng, J. Yang, Y. Yu, and J. Huang, “Retroxpert: Decompose retrosynthesis prediction like a chemist,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 248–11 258, 2020.

[14] C. Yan, P. Zhao, C. Lu, Y. Yu, and J. Huang, “Retrocomposer: Discovering novel reactions by composing templates for retrosynthesis prediction,” *arXiv preprint arXiv:2112.11225*, 2021.

[15] X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang, “Direct shape regression networks for end-to-end face alignment,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5040–5049.

[16] C. Yan, J. Yao, R. Li, Z. Xu, and J. Huang, "Weakly supervised deep learning for thoracic disease classification and localization on chest x-rays," in *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, 2018, pp. 103–110.

[17] S. Wang, Z. Xu, C. Yan, and J. Huang, "Graph convolutional nets for tool presence detection in surgical videos," in *International Conference on Information Processing in Medical Imaging.* Springer, 2019, pp. 467–478.

[18] X. Miao, X. Yuan, Y. Pu, and V. Athitsos, "l-net: Reconstruct hyperspectral images from a snapshot measurement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4059–4069.

[19] X. Miao, X. Yuan, and P. Wilford, "Deep learning for compressive spectral imaging," in *Digital Holography and Three-Dimensional Imaging.* Optica Publishing Group, 2019, pp. M3B–3.

[20] J. Yang, W. An, C. Yan, P. Zhao, and J. Huang, "Context-aware domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 514–524.

[21] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *European conference on computer vision.* Springer, 2020, pp. 480–498.

[22] J. Yang, P. Zhao, Y. Rong, C. Yan, C. Li, H. Ma, and J. Huang, "Hierarchical graph capsule network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 603–10 611.

[23] P. K. Diederik, M. Welling, *et al.*, "Auto-encoding variational bayes," in *Proceedings of the ICLR*, 2014.

[24] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” in *ICML*, 2014, pp. 1278–1286.

[25] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[26] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[27] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, “Generating sentences from a continuous space,” in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[28] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, “Lagging inference networks and posterior collapse in variational autoencoders,” in *Proceedings of the ICLR*, 2019.

[29] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia, “Learning deep generative models of graphs,” *arXiv preprint arXiv:1803.03324*, 2018.

[30] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher, “Guacamol: benchmarking models for de novo molecular design,” *Journal of chemical information and modeling*, vol. 59, no. 3, pp. 1096–1108, 2019.

[31] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, “Improved variational autoencoders for text modeling using dilated convolutions,” in *Proceedings of the 34th ICML-Volume 70*. JMLR. org, 2017, pp. 3881–3890.

[32] Y. Kim, S. Wiseman, A. Miller, D. Sontag, and A. Rush, “Semi-amortized variational autoencoders,” in *ICML*, 2018, pp. 2683–2692.

[33] M. D. Hoffman and M. J. Johnson, “Elbo surgery: yet another way to carve up the variational evidence lower bound,” in *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2016.

[34] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[36] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, “Cyclical annealing schedule: A simple approach to mitigating KL vanishing,” in *NAACL*, 2019.

[37] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[38] I. Higgins, L. Matthey, A. Pal, C. Burgess, and e. al, “beta-vae: Learning basic visual concepts with a constrained variational framework.” in *Proceedings of the ICLR*, 2017.

[39] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[41] T. Sterling and J. J. Irwin, “Zinc 15–ligand discovery for everyone,” *Journal of chemical information and modeling*, vol. 55, no. 11, pp. 2324–2337, 2015.

[42] P. Schwaller, T. Gaudin, D. Lanyi, C. Bekas, and T. Laino, ““found in translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models,” *Chemical science*, vol. 9, no. 28, pp. 6091–6098, 2018.

[43] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, *et al.*, "Chembl: towards direct deposition of bioassay data," *Nucleic acids research*, vol. 47, no. D1, pp. D930–D940, 2019.

[44] G. Landrum *et al.*, "Rdkit: Open-source cheminformatics," 2006.

[45] A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, "Avoiding latent variable collapse with generative skip models," in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2397–2405.

[46] S. Mohammadi, B. O'Dowd, C. Paulitz-Erdmann, and L. Goerlitz, "Penalized variational autoencoder for molecular design," *ChemRxiv*, 2019.

[47] Z. Xu, S. Wang, F. Zhu, and J. Huang, "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery," in *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*, 2017, pp. 285–294.

[48] H. Ma, C. Yan, Y. Guo, S. Wang, Y. Wang, H. Sun, and J. Huang, "Improving molecular property prediction on limited data with deep multi-label learning," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, 2020, pp. 2779–2784.

[49] H. Ma, Y. Rong, B. Liu, Y. Guo, C. Yan, and J. Huang, "Gradient-norm based attentive loss for molecular property prediction," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, 2021, pp. 497–502.

[50] H. Ma, Y. Bian, Y. Rong, W. Huang, T. Xu, W. Xie, G. Ye, and J. Huang, "Cross-dependent graph neural networks for molecular property prediction," *Bioinformatics*, vol. 38, no. 7, pp. 2003–2009, 2022.

[51] H. Ma, F. Jiang, Y. Rong, Y. Guo, and J. Huang, "Robust self-training strategy for various molecular biology prediction tasks," in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2022, pp. 1–5.

[52] E. J. Corey and W. T. Wipke, "Computer-assisted design of complex organic syntheses," *Science*, vol. 166, no. 3902, pp. 178–192, 1969.

[53] E. J. Corey, "The logic of chemical synthesis: multistep synthesis of complex carbogenic molecules (nobel lecture)," *Angewandte Chemie International Edition in English*, vol. 30, no. 5, pp. 455–465, 1991.

[54] C. M. Gothard, S. Soh, N. A. Gothard, B. Kowalczyk, Y. Wei, B. Baytekin, and B. A. Grzybowski, "Rewiring chemistry: Algorithmic discovery and experimental validation of one-pot reactions in the network of organic chemistry," *Angewandte Chemie International Edition*, vol. 51, no. 32, pp. 7922–7927, 2012.

[55] B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender, and V. Pande, "Retrosynthetic reaction prediction using neural sequence-to-sequence models," *ACS central science*, vol. 3, no. 10, pp. 1103–1113, 2017.

[56] D. A. Pensak and E. J. Corey, "Lhasa—logic and heuristics applied to synthetic analysis," 1977.

[57] E. J. Corey, "General methods for the construction of complex molecules," *Pure and Applied chemistry*, vol. 14, no. 1, pp. 19–38, 1967.

[58] C. D. Christ, M. Zentgraf, and J. M. Kriegl, "Mining electronic laboratory notebooks: Analysis, retrosynthesis, and reaction based enumeration," *Journal of chemical information and modeling*, vol. 52, no. 7, pp. 1745–1756, 2012.

[59] A. Bogevig, H.-J. Federsel, F. Huerta, M. G. Hutchings, H. Kraut, T. Langer, P. Low, C. Oppawsky, T. Rein, and H. Saller, "Route design in the 21st century:

The icsynth software tool as an idea generator for synthesis prediction," *Organic Process Research & Development*, vol. 19, no. 2, pp. 357–368, 2015.

[60] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, and B. A. Grzybowski, "Computer-assisted synthetic planning: The end of the beginning," *Angewandte Chemie International Edition*, vol. 55, no. 20, pp. 5904–5937, 2016.

[61] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen, "Computer-assisted retrosynthesis based on molecular similarity," *ACS central science*, vol. 3, no. 12, pp. 1237–1245, 2017.

[62] M. H. Segler, M. Preuss, and M. P. Waller, "Planning chemical syntheses with deep neural networks and symbolic ai," *Nature*, vol. 555, no. 7698, pp. 604–610, 2018.

[63] K. Lin, Y. Xu, J. Pei, and L. Lai, "Automatic retrosynthetic route planning using template-free models," *Chemical Science*, vol. 11, no. 12, pp. 3355–3364, 2020.

[64] C. Shi, M. Xu, H. Guo, M. Zhang, and J. Tang, "A graph to graphs framework for retrosynthesis prediction," *arXiv preprint arXiv:2003.12725*, 2020.

[65] T. Klucznik, B. Mikulak-Klucznik, M. P. McCormack, H. Lima, S. Szymkuć, M. Bhowmick, K. Molga, Y. Zhou, L. Rickershauser, E. P. Gajewska, *et al.*, "Efficient syntheses of diverse, medicinally relevant targets planned by computer and executed in the laboratory," *Chem*, vol. 4, no. 3, pp. 522–532, 2018.

[66] M. H. Segler and M. P. Waller, "Modelling chemical reasoning to predict and invent reactions," *Chemistry–A European Journal*, vol. 23, no. 25, pp. 6118–6128, 2017.

[67] N. Schneider, N. Stiefl, and G. A. Landrum, “What’s what: The (nearly) definitive guide to reaction role assignment,” *Journal of chemical information and modeling*, vol. 56, no. 12, pp. 2336–2346, 2016.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[69] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, “Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction,” *ACS central science*, vol. 5, no. 9, pp. 1572–1583, 2019.

[70] J. Bradshaw, M. J. Kusner, B. Paige, M. H. Segler, and J. M. Hernández-Lobato, “A generative model for electron paths,” *arXiv preprint arXiv:1805.10970*, 2018.

[71] W. Jin, C. Coley, R. Barzilay, and T. Jaakkola, “Predicting organic reaction outcomes with weisfeiler-lehman network,” in *Advances in Neural Information Processing Systems*, 2017.

[72] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.

[73] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” *International Conference on Learning Representations*, 2018.

[74] D. Lowe, “Chemical reactions from us patents (1976-sep2016),” 2018.

[75] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[76] D. M. Lowe, “Extraction of chemical structures and reactions from the literature,” Ph.D. dissertation, University of Cambridge, 2012.

[77] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, *et al.*, “Deep graph library: Towards efficient and scalable deep learning on graphs,” *arXiv preprint arXiv:1909.01315*, 2019.

[78] C. W. Coley, W. H. Green, and K. F. Jensen, “Rdchiral: An rdkit wrapper for handling stereochemistry in retrosynthetic template extraction and application,” *Journal of chemical information and modeling*, vol. 59, no. 6, pp. 2529–2537, 2019.

[79] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “OpenNMT: Open-source toolkit for neural machine translation,” in *Proc. ACL*, 2017. [Online]. Available: https://doi.org/10.18653/v1/P17-4012

[80] C. Tillmann and H. Ney, “Word reordering and a dynamic programming beam search algorithm for statistical machine translation,” *Computational linguistics*, vol. 29, no. 1, pp. 97–133, 2003.

[81] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano, and T. Laino, “Predicting retrosynthetic pathways using a combined linguistic model and hyper-graph exploration strategy,” *arXiv preprint arXiv:1910.08036*, 2019.

[82] A. Raju, C.-T. Cheng, Y. Huo, J. Cai, J. Huang, J. Xiao, L. Lu, C. Liao, and A. P. Harrison, “Co-heterogeneous and adaptive segmentation from multi-source and multi-phase ct imaging data: a study on pathological liver and lesion segmentation,” in *European Conference on Computer Vision.* Springer, 2020, pp. 448–465.

[83] A. Raju, J. Yao, M. M. Haq, J. Jonnagaddala, and J. Huang, “Graph attention multi-instance learning for accurate colorectal cancer staging,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 529–539.

[84] A. Raju, Z. Ji, C. T. Cheng, J. Cai, J. Huang, J. Xiao, L. Lu, C. Liao, and A. P. Harrison, "User-guided domain adaptation for rapid annotation from user interactions: a study on pathological liver segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 457–467.

[85] A. Raju, S. Miao, D. Jin, L. Lu, J. Huang, and A. P. Harrison, "Deep implicit statistical shape models for 3d medical image delineation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2135–2143.

[86] Y. Guo, J. Wu, H. Ma, and J. Huang, "Self-supervised pre-training for protein embeddings using tertiary structures," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, pp. 6801–6809, Jun. 2022.

[87] M. Sacha, M. Błaz, P. Byrski, P. Dabrowski-Tumanski, M. Chrominski, R. Loska, P. Włodarczyk-Pruszynski, and S. Jastrzebski, "Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits," *Journal of Chemical Information and Modeling*, vol. 61, no. 7, pp. 3273–3284, 2021.

[88] R. Sun, H. Dai, L. Li, S. Kearnes, and B. Dai, "Towards understanding retrosynthesis by energy-based models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 186–10 194, 2021.

[89] I. V. Tetko, P. Karpov, R. Van Deursen, and G. Godin, "State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis," *Nature communications*, vol. 11, no. 1, pp. 1–11, 2020.

[90] X. Wang, Y. Li, J. Qiu, G. Chen, H. Liu, B. Liao, C.-Y. Hsieh, and X. Yao, "Retroprime: A diverse, plausible and transformer-based method for single-step retrosynthesis predictions," *Chemical Engineering Journal*, vol. 420, p. 129845, 2021.

[91] V. R. Somnath, C. Bunne, C. Coley, A. Krause, and R. Barzilay, "Learning graph models for retrosynthesis prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9405–9415, 2021.

[92] J. L. Baylon, N. A. Cilfone, J. R. Gulcher, and T. W. Chittenden, "Enhancing retrosynthetic reaction prediction with deep learning using multiscale reaction classification," *Journal of chemical information and modeling*, vol. 59, no. 2, pp. 673–688, 2019.

[93] Z. Tu and C. W. Coley, "Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction," *Journal of Chemical Information and Modeling*, 2021.

[94] R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum, "Chemformer: A pre-trained transformer for computational chemistry," *Machine Learning: Science and Technology*, 2021.

[95] K. Mao, X. Xiao, T. Xu, Y. Rong, J. Huang, and P. Zhao, "Molecular graph enhanced transformer for retrosynthesis prediction," *Neurocomputing*, vol. 457, pp. 193–202, 2021.

[96] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[97] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[98] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.

[99] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[100] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[101] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[102] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

BIOGRAPHICAL STATEMENT

Chaochao Yan received his Ph.D. degree in Computer Science from the University of Texas at Arlington in 2022. Prior to the Ph.D. program at UT Arlington, he received his M.S degree in Computer Technology from University of Chinese Academy of Sciences in 2017. He received his B.S degree in Automation from Huazhong University of Science and Technology in 2014. His research mainly lies in the areas of machine learning, graph neural networks, and their applications in drug discovery and medical image analysis. During his Ph.D. study, he has published more than 10 conference and journal papers, such as Conference on Neural Information Processing Systems (NeurIPS), ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM BCB), International Conference on Information Processing in Medical Imaging (IPMI), IEEE International Conference on Bioinformatics and Biomedicine (BIBM), European Conference on Computer Vision (ECCV), AAAI Conference on Artificial Intelligence (AAAI), IEEE Winter Conference on Applications of Computer Vision (WACV). He has been invited to serve as a reviewer for many top-tier conferences, such as ICML, NeurIPS, ICLR, AAAI, CVPR, ICCV, ECCV, WACV, and MICCAI.