NeuralTPL: a deep learning approach for efficient reaction space exploration

Yue Wan[a], Xin Li[b], Xiaorui Wang[c], Xiaojun Yao[c], Benben Liao[a]*, Chang-Yu Hsieh[a]*, Shengyu Zhang[a]*

[a] *Tencent Quantum Laboratory, Tencent, Shenzhen, PR China*

[b] *Department of Chemistry, Zhejiang University, Hangzhou, PR China*

[c] *State Key Laboratory of Quality Research in Chinese Medicines, Macau University of Science and Technology, Macao*

Abstract

Computer-aided synthesis planning (CASP) has been helping chemists to synthesize novel molecules at an accelerated pace. The recent integration of deep learning with CASP has opened up new avenues for digitizing and exploring the vastly unknown chemical space, and has led to high expectations for fully automated synthesis plannings using machine-discovered novel reactions in the "future". Despite many progresses in the past few years, most deep-learning methods only focus on improving few aspects of CASP (e.g., top-k accuracy). In this work, we target specifically the efficiency of reaction space exploration and its impact on CASP. We propose NeuralTPL, a template-oriented generative approach, that performs impressively across a range of evaluation metrics including chemical validity, diversity, and novelty for various tasks in CASP. In addition, our Transformer-based model bears the potential to learn the core reaction transformation as it can efficiently explore the reaction space. We then perform several experiments and conduct a thorough analysis regarding the three metrics and demonstrate its chemical value for improving the existing deep-learning-driven CASP algorithms.

1 Introduction

Organic synthesis, which involves building up complex molecules from simpler and more accessible precursors, is a cornerstone in modern organic chemistry. Advancements in organic synthesis are crucial for the pharmaceutical industry, as one of the main challenges for drug discovery is to efficiently synthesize novel and complex compounds in the laboratory [1]. Efficient organic synthesis also plays an important role in a wide range of fields from nanotechnology and organic-based functional material design, to supramolecular chemistry and chemical biology. Despite steady progress made in the past few decades, only a tiny fraction of the entire reaction space has been explored. As more molecules with desirable properties were explored by data-driven generative models and more chemical transformations are discovered by experiments, the reaction space has been expanding rapidly, whereas such exponential growth far exceeds the pace of experiment-driven exploration. Although the expansion provides more synthesis options for organic functional molecules, the cost of exploring the reaction space has risen rapidly, making the exploration's efficiency a crucial factor for practical concerns. In this work, we target specifically the efficiency of reaction space exploration and its impact in computer-aided synthesis planning.

In general, the reaction chemical space can be viewed as a gigantic directed knowledge graph composed of both molecule nodes and reactivity nodes (Figure 1: Top), where synthesis planning can be viewed as chemical reasoning within the graph [2]. In this work, we take the practical viewpoint from the synthesis planning by

*Corresponding authors.

E-mail addresses: bliao@tencent.com (B. Liao), kimhsieh@tencent.com (C.-Y. Hsieh), shengyzhang@tencent.com (S. Zhang)

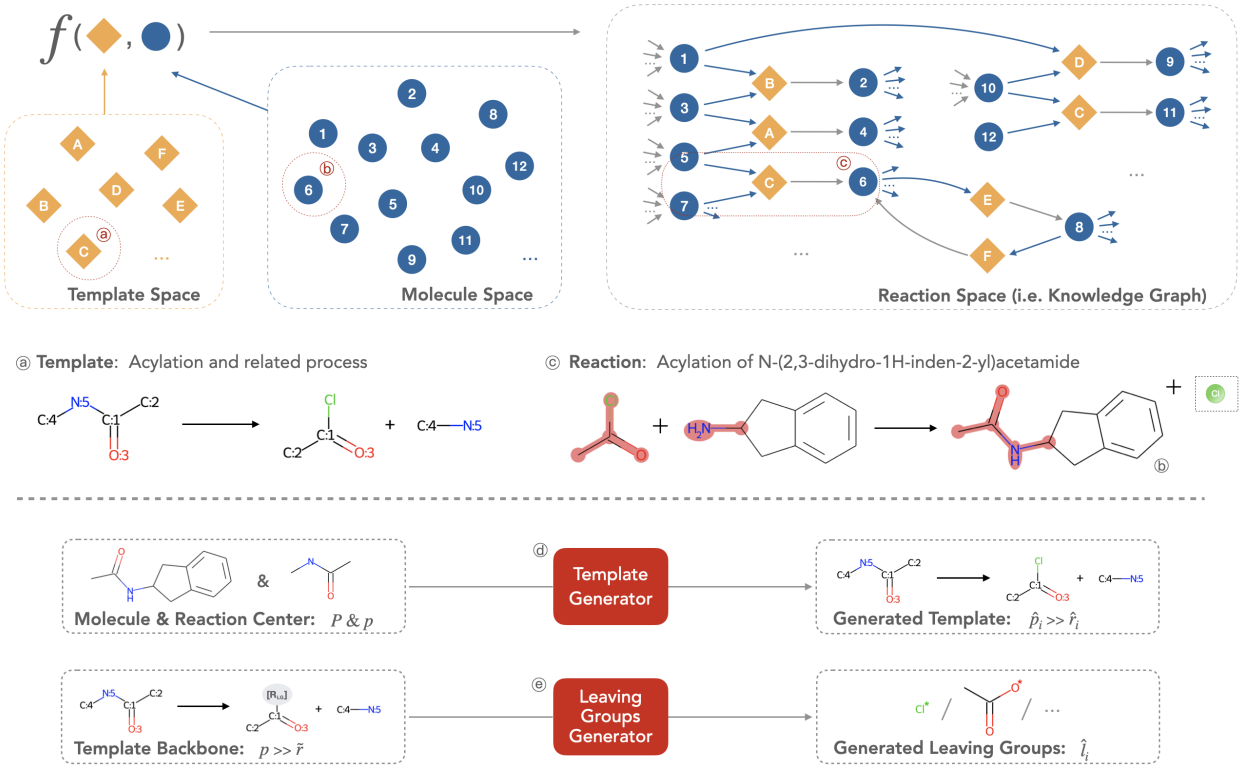


Figure 1: Top: relationship between reaction space, molecule space, and template space with (a) template sample (b) molecule sample and (c) reaction sample. Each reaction in the reaction graph only considers the core product from the synthesis planning perspective. Bottom: overview of our proposed (d) template generation task and (e) leaving groups generation task. The star symbol in the generated leaving groups represents the attachment position.

only considering the core product and its synthetic routes (i.e., reactants) as the chemical transformation. As we mentioned before, both the magnitude of the graph and its largely unexplored ratio have brought challenges and heavy costs of solving the synthesis planning problems. Despite its complexity, we observe that the reaction space can be factorized into the molecule space and the reaction template space, with a likelihood function that captures the reaction likelihood given a molecule and a template. Reaction template is a set of chemical rules that encode the core chemical transformation of the reactions. The molecular fragments where the transformation takes place are called the reaction center. In general, the reaction can be easily constructed by applying its corresponding template to the molecule if matched. Existing templates are either expert-defined or automatically extracted from documented reactions. Although the template space is also not fully explored, its magnitude is much smaller than the reaction space. Because of templates' effectiveness and interpretability in modeling reactions, a group of methods in synthesis planning are built upon them. These template-based methods [3, 4, 5, 6, 7, 8] mainly target to learn the reaction likelihood function given a molecule and a template. GLN [4], as the state-of-the-art retrosynthesis model, uses a conditional graphical model to estimate the joint probability of template and reactants candidates given a product molecule. Although their performance is promising regarding the top-k accuracy, these methods suffer heavily from the coverage issue such that they are limited to the visible portion of the template space from the training dataset. Another group of methods adopt either sequence-based or graph-based deep learning architectures to directly model the entire reactions. These template-free methods [9, 10, 11, 12, 13, 14, 15, 16, 17, 18] first learn the chemical knowledge from the visible reaction space and perform synthetic routes generation directly from the entire reaction space. They hold the potential to be able to infer novel chemical transformations beyond the training scope. Although it sounds promising for its flexibility of chemical knowledge exploration, experiments have shown that chemical reasoning within the entire reaction space is highly inefficient for deep

learning models. For example, candidate synthetic routes generated by Transformer-based models [10, 11, 15] via beam search [19] are often lack of diversity [20, 21, 22]. The work in [23] also concludes that the LSTM Autoencoder [24] is rather limited in designing novel chemical reactions. Another factor of ineffectiveness comes from the high invalidity ratio of the generated outcomes since chemical validity is often not part of the training objective [13].

Weighing the pros and cons of the two major groups of data-driven approaches in synthesis planning, we propose NeuralTPL, a template-oriented generative approach for the synthesis planning problem. Instead of directly exploring the entire reaction space, we aim to explore the template space via the deep generative models. Taking advantage of the existing template-based methods [4, 8] that the reaction likelihood function is effectively approximated, template space exploration can lead to reaction space exploration. In addition, it holds advantages over the existing synthesis planning methods in terms of validity, diversity, and novelty, which we will further elaborate on in the **Results** section. In this work, a template is defined as the reverse chemical transformation. The definition is motivated by retrosynthesis [25], which is the task of predicting synthetic routes given a core product molecule. It is the reverse problem of reaction prediction, yet it is more challenging since the model needs to explore a much broader output space with creativity. For the template space exploration, we propose two methods, which are template generation and leaving groups generation (Figure 1: Bottom). Template Generator takes a product molecule plus one of its reaction centers as inputs and attempts to answer the question *"what reaction templates can be applied on the given product that target the given reaction center?"*. Note that different choices of reaction centers can introduce another factor of diversity for template generation, just like the diversity advantage of the template-based methods like GLN. Another method is the leaving groups generation. Leaving groups, referred to as the departing fragments from the reactants without attaching to the core product, play a key role in reactions like nucleophilic substitution. As different choices of leaving groups can affect the reactivity rate, leaving groups selection is a common strategy for reaction design. Leaving Groups Generator takes a template backbone as input and attempt to answer the question *"what leaving groups can be attached to the template backbone?"*. Overall, template generation can be viewed as locating the exact position within the global template space; while leaving groups generation can be viewed as neighborhood exploration within the local template space, since templates that only differ by leaving groups are chemically similar. Both task formations have nice properties such that they can be easily integrated with existing synthesis planning approaches and allows efficient reaction space exploration.

Experiments have shown that our model can reach an impressive 95.9% top-10 template generation accuracy, and 97.6% top-10 leaving groups generation accuracy, indicating that the deep learning model can efficiently learn the chemical transformations at the fragment level. Compared to the conventional template-free methods, the top-10 reaction validity rate of our generated outcomes is 18% higher than directly generating the reactants given the product. We then conduct a thorough analysis on the reconstructed template space with the generated templates, including the spatial relationship between template entries, the interpretability of how the model understands chemical knowledge, as well as the novelty implication of the generated templates. We then showcase a couple of cases that demonstrate the diversity of the synthetic routes endorsed by our method. We also analyze the implication of our method as a replaceable module to improve the state-of-the-art template-based retrosynthesis (e.g., GLN) by extending it to the generative domain. To show the practical value of our template space exploration, we further introduce the technique of template database expansion, which aims to mitigate the template coverage issue of existing template-based methods in synthesis planning. We evaluate our methods on the tasks of reaction prediction, single-step retrosynthesis, and multi-step retrosynthesis. Multi-step retrosynthesis is the recursive version of single-step retrosynthesis, where the model needs to recursively decompose the molecule into purchasable substances. These downstream experiments demonstrate the effectiveness of our method for reaction space exploration that it improves over the existing template-based performance. Finally, we discuss the current challenges and potentials of applying template-oriented methods in a broader context and propose promising future research directions. Our contribution can be summarized as below:

1) We propose NeuralTPL, a template-oriented generative approach for synthesis planning tasks, which combines the advantages of both template-based and template-free methods, and achieves better validity, diversity, and novelty.

2) Experiments have shown that our proposed model can achieve a promising 97.6% top-10 leaving groups generation accuracy, and a 95.9% top-10 template generation accuracy given the reaction center.

3) The reaction template chemical space reconstructed by our approach is able to provide interpretable and comprehensive insights in both template classifications and template generations.

4) The proposed template database expansion technique is able to improve the top-k accuracy performance of synthesis planning tasks using template-based methods by around 40% in the scenario where the original template coverage ratio of test set is only 5.0%.

2 Results

Notation and Task Overview We start by defining the notation that is used in Figure 1 (Bottom) and the experiment results. Let a reaction RXN be the chemical transformation from the reactant set $\mathcal{R}$ to the product set $\mathcal{P}$. A template T is defined as the reverse transformation from the product's reaction center set p to the reactant's reaction center set r. The leaving groups l in r can either be explicitly defined or represented by the general leaving groups indicator $[\mathrm{R_{LG}}]$. We define the latter version of r as $\tilde{r}$. We term $p \gg \tilde{r}$ as *template backbone* T' to differentiate it from the term *template* T (i.e., $p \gg r$). Then, the template generation task is to generate the templates given the product and its reaction center: $<\mathcal{P}, p> \rightarrow \{\hat{p}_i \gg \hat{r}_i\}_{i=1}^{k}$, where the hat symbol represents the predicted value. The leaving groups generation task is to generate the specific leaving groups given the template backbone: $<p \gg \tilde{r}> \rightarrow \{\hat{l}_i\}_{i=1}^{k}$. The generated leaving groups can be reconstructed back into the templates as $\{p \gg \texttt{Attach}(\tilde{r}, \hat{l}_i)\}_{i=1}^{k}$. Note that the output templates of the first task is different than the input template backbone of the second task.

Dataset We use the synthesis planning benchmark dataset USPTO-50K for our experiments. It contains 50k valid atom-mapped chemical reactions from the US patent database, each of which is represented using SMILES sequence and categorized into 10 reaction types. We use the same data split as Coley et al. [5]. For each reaction, we use RDChiral [26] to extract the reaction template, which is represented using SMARTS sequence. We then perform the data processing procedure described in the **Methods** section prior to training the template generator and leaving groups generator.

Top-k Performance Analysis We adopt the conventional top-k generation accuracy as the metric to measure the number of ground truth templates or leaving groups the model is able to recover. This metric examines the model's effectiveness in learning implicit chemical rules. Table 1a shows the top-k accuracy of template generation and leaving groups generation. Our model is able to recover 66.9% and 97.6% of leaving groups from top-1 and top-10 candidates, as well as 64.2% and 88.1% of ground truth templates in the test set from top-1 and top-10 candidates without any constraints. Since we generate the entire template $\hat{p} \gg \hat{r}$ as the transformation from one reaction center to another, $\hat{p}$ is not guaranteed to match with the exact given reaction center p. Although it might sound counter-intuitive if these two are different, for novelty purposes, it provides the model an additional degree of freedom for template space exploration. We then examine the performance where we impose the equality $\hat{p} = p$ and only generate the transformed reaction center $\hat{r}$. This setting is mainly designated for downstream synthesis planning tasks, where validity and accuracy matter more than novelty. Under this setting, our model's performance can reach 72.5% top-1 and 95.9% top-10 accuracy. Note that our template generator is compatible with any template-based method that uses the reaction center as a conditional factor for template proposal. For example, GLN [4] decomposes the retrosynthesis problem into steps of reaction center proposal, template proposal given the reaction center, and reaction likelihood ranking. We could directly replace its second step of template proposal with our template generation. From the accuracy perspective, we pull out GLN's conditional template proposal module and compares its performance given the ground truth reaction center against our methods, which shows that our approach is as competitive as GLN regarding the template top-k accuracy. Detailed table is reported in **Supplementary Information**. This implies the promising potential such that a hybrid model of existing template-based methods and our approach could explore reaction knowledge within a broader chemical

Table 1: Top-k accuracy and validity in test set of each generation task. All generative models are Transformer-based. † indicates the case where we constraint the product reaction center within the template to be exactly the same as the given one.

(a) Top-k template accuracy of the generation tasks.

Tasks	***Top-k Accuracy(%)***				
	1	3	5	7	10
Template†	72.5	91.0	94.2	95.1	95.9
Template	64.2	82.2	85.7	87.4	88.1
Leaving Groups	66.9	91.4	96.0	97.2	97.6

(b) Top-k validity of the generation tasks.

Tasks	***Top-k Validity Rate(%)***				
	1	3	5	7	10
Molecule [15]	95.2	86.9	81.7	78.2	71.3
Template	99.8	99.2	98.1	94.7	89.3
Leaving Groups	98.6	96.6	94.8	92.1	86.6

space for better performance regarding the chemical novelty, while still enjoying the validity and diversity advantages of the template-based methods.

We then evaluate the chemical validity of the generated outcomes. For assessing the chemical validity, we consider a template or reaction as valid if RDKit [27] can successfully process it as a reaction and calculate its top-k validity ratio accordingly. Table 1b shows the validity of generating the entire synthetic routes by the vanilla retrosynthesis Transformer [15], our template generation task, and our leaving groups generation task. The top-10 reaction validity ratio of our template generation method is 18% higher than that of the vanilla retrosynthesis Transformer; while that of the leaving groups generation method is 15.3% higher compared to the vanilla retrosynthesis Transformer. The validity improvement is expected. Since retrosynthesis Transformer directly generates the entire reactants SMILES sequence, errors may propagate via the generated molecule fragments that are distant from the reaction center. These fragments should most likely remain unchanged during the reaction. In our method, we only generate the reactive fragments, which leaves us less room for errors. The performance gap between our two generation tasks could be attributed to the potential merging error of the leaving groups and template backbone.

Template Space Analysis Despite being a high-level metric, the top-k accuracy does not reflect any deeper insights behind how the model understands the reaction templates or the potential chemical practicality of other candidates. To explore these values, we extract the template embeddings learned by the model, project and plot them in a low-dimensional space, and conduct a thorough analysis in the spatial relationship between these template data points.

Figure 2 shows the T-SNE plot of the embedding space including both the ground truth templates (labeled by $\bullet$) and the generated templates (labeled by $\times$). During training, we add an additional template classification task so that the model can learn jointly the information of the reaction class and the generative probability. The coloring represents the predicted reaction class. We could easily see from the plot that templates within the same reaction class form clusters. In fact, our template classifier can reach the testing accuracy of 99.21%. Such superior performance indicates that the reaction class information is implicitly and reliably encoded into the template representation. We then analyze the reason behind the reaction class labeling conflicts suggested by the reaction class prediction. There are in total 39 mis-classified templates. The small red box with letter F in Figure 2 represents the samples that are mis-classified. Almost all of them are located along the edge of clusters, indicating themselves as potential overlapping points between clusters. We believe that the mis-classification is mostly because the labeling itself is not mutually exclusive. For example, heteroatom alkylation and arylation is a reaction class with wide coverage. Sample (a1) is originally assigned to this category, whereas the model predicts it to be reaction class (2) (acylation and related processes). As a matter of fact, in terms of templates, the transformations (a1) and (a2) both include acyl groups to the specific N atoms with acyl halides as acylating agents, which is one of the features for acylation and related processes. Although their original category is reasonable, our model provides valid and possibly more appropriate alternatives. Sample (a3) follows the same idea. The template dissociates the ester into carboxylic acid and alcohol, which could be understood as the deprotection (i.e., $[\mathrm{rt}_6]$) process of alcohol. These cases show that the model attains a comprehensive understanding of different reaction classes.

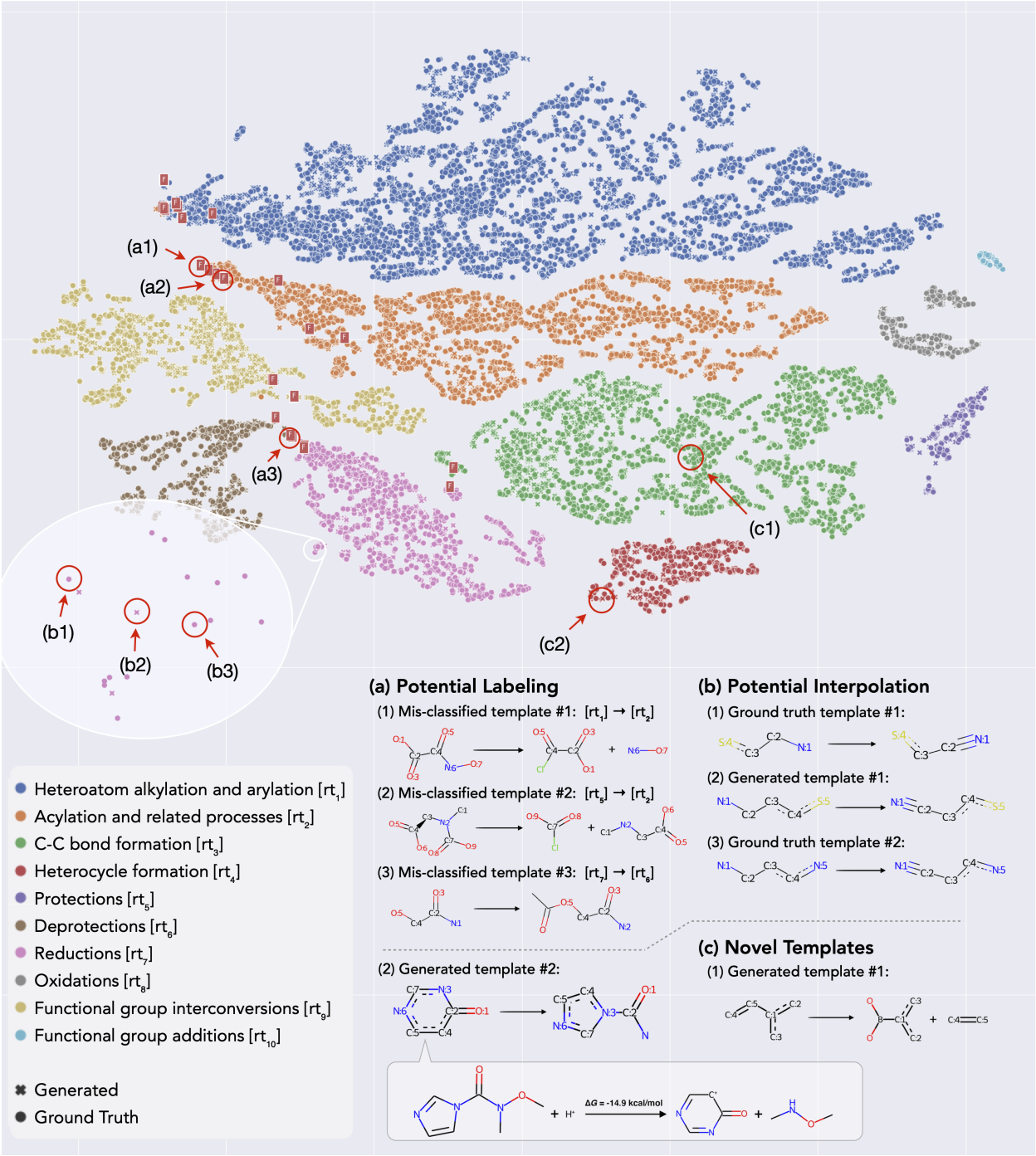


Figure 2: Template embedding space of both the generated and the ground truth templates. Only 11819 newly generated templates are included for plotting purpose.

We then take a look at the generated template and its relationship to the ground truth templates. Group (b) samples indicate the possibility of data interpolation within the learned template space. Sample (b2) is the generated result, while (b1) and (b3) are its ground truth template neighbors in the template space. From the perspective of atomic composition and template structure, sample (b2) can be seen as the combination of (b1) and (b3), indicating the model's ability to infer template modification based on the understanding of the existing ones. This is compatible with chemists' designing schema for new reactions, where the novelty is built on top of the existing reactions. Their relative locations in the embedding space also demonstrate the

possibility of template interpolation for exploring the reaction knowledge.

Group (c) contains samples that are relatively new from both the machine and the human perspective. Sample (c1) shows a C−H Suzuki-Miyaura coupling reaction which is not covered in the training set. Suzuki-Miyaura reaction is a palladium catalyzed cross-coupling reaction between organic boron compounds and organic halides [28, 29, 30, 31, 32]. This relatively simple and versatile C−C bond formation reaction can be extended to various substrates and therefore has a wide range of applications for the synthesis of pharmaceuticals and total synthesis of complex natural products. In the past decade, C−H Suzuki-Miyaura coupling reactions have attracted widespread attention and achieved significant development [33]. Since the bond-dissociation energy (BDE) of C−H (∼110.9 kcal/mol) in olefin is much larger than C−X (∼94.2 kcal/mol for C−Cl, ∼80.9 kcal/mol for C−Br, and ∼61.9 kcal/mol for C−I) [34], the activation of C−H bond is a more challenging task. Sample (c1) shows that our model could learn from the traditional haloolefins and capture the potential C−H Suzuki-Miyaura coupling reactions. We also evaluate the generated templates that are novel and unpublished. Sample (c2) is an interesting example. Since it is difficult to confirm whether an undocumented reaction template is indeed novel and chemically valid, we adopt the criterion for the "novel reaction template" as an undocumented template that has considerable chemical rationality, where chemical rationality is verified through a combination of expert judgment and density functional theory (DFT) calculation. In this investigation for novel templates, we specifically look for generations of a new ring-forming reaction. The proportion of ring-forming reactions in the USPTO-50K dataset is relatively small, only 900 records, accounting for 1.8% of the entire dataset. Such scarcity shows the difficulty of discovering new ring-forming reactions, and it reflects that the design of ring-forming reactions is a very promising field. To our delight, (c2) is successfully observed as we inspect these templates. For rationality evaluation, We build up a complete reaction corresponding to the template (c2) and verify its thermodynamic rationality via DFT calculations. We complete the N atom into a Weinreb amide [35], which is a common amide N leaving group. For this substrate imidazole-carboxamide, the reaction is capable of inducing ring expansion by forming carbocations. Based on this reaction mechanism, we determine the product 4-oxo-4,5-dihydropyrimidin-5-ylium as the calculation endpoint. There are many ways to eliminate carbocations in this product. We calculate the Gibbs free energy difference (ΔG) for this conversion, which is -14.9 kcal/mol, and confirm that this reaction is thermodynamically possible. DFT calculation details are shown in **Supplementary Information**.

Template Customization In some scenarios, we would like to walk around a particular synthetic route for reasons like avoiding a protected patent route or bypassing the reaction with unacceptable harsh conditions or unsatisfactory yield. Therefore, it is desirable if the model could flexibly target or avoid a given reaction center when performing reasoning. We term this procedure as template customization, which can be achieved by our framework. As we will describe more details in **Methods** section, our template generative framework contains a special argument, which is used like a switch between encouraging $\hat{p} = p$ and $\hat{p} \neq p$. In the latter setting, the model will treat the given reaction center p as noise while randomly selecting another reaction center as input. We apply template customization on out-of-sample molecules to demonstrate the flexibility of our framework, as well as the diversity of the generative outcomes.

We randomly select drug compounds with more than one documented synthesis route as our target molecules from [36], a drug database with documented paths. They are chosen outside of the scope of USPTO-50K to test the generalizability of our method. Figure 3 shows the example template customization of icotinib hydrochloride. Icotinib hydrochloride is a potent and specific epidermal growth factor receptor (EGFR) tyrosine kinase inhibitor (TKI) and is being evaluated as a treatment for EGFR+ Non-Small Cell Lung Cancer (NSCLC) [37]. Let the reaction ① [38] be the initial path that we would like to avoid. The red highlight indicates its original reaction centers. (a) and (b) are two of the candidate reaction centers that are randomly sampled by our generative pipeline. Each of them is paired with the target product molecule and fed to the template generator. We then build up the full reaction from the generated template and the product molecule. Reaction ② and ③ are two of the synthetic routes discovered by our methods. Reaction ② is a documented route [39]. The template corresponding to the reaction ② already exists in USPTO-50K, yet it shows the generalizability of our model to capture the appropriate templates for unseen molecules. To our notice, reaction ③ is one of the unpublished novel candidates that we find worth mentioning. It is essentially a heterocycle formation reaction with triflate as the leaving group. This shows that our template-oriented generative method is well beyond the capability of the template-based methods by discovering out-of-sample

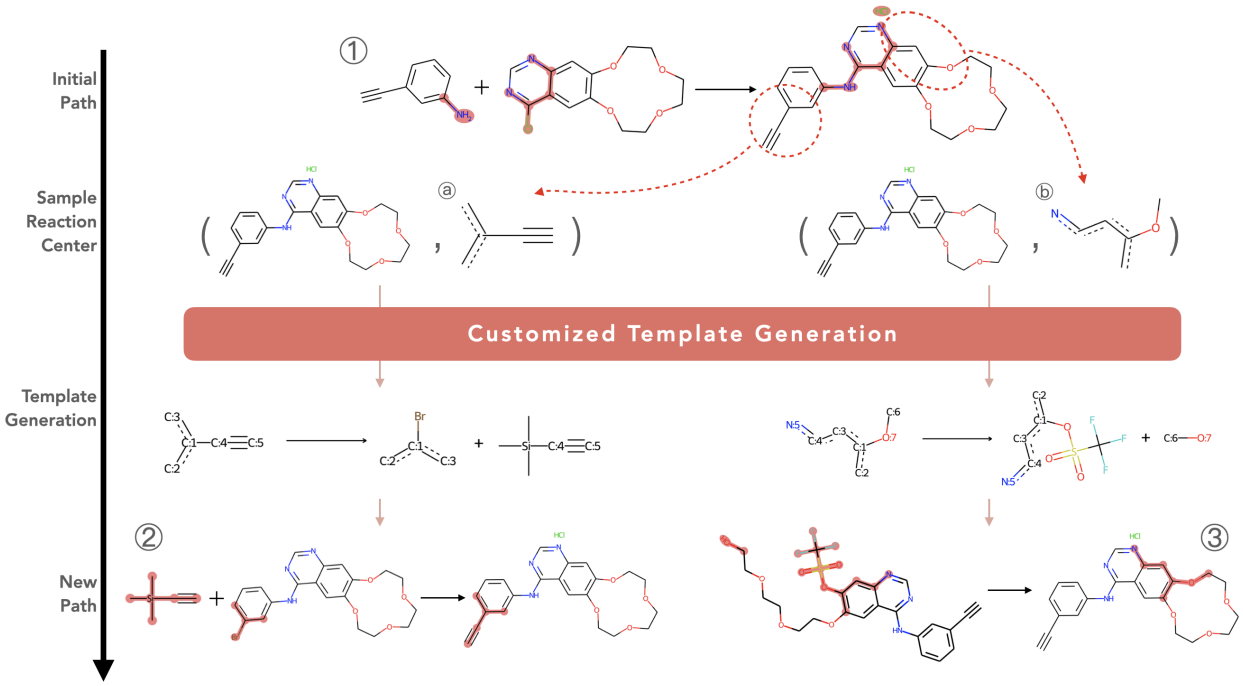


Figure 3: Template customization pipeline with Icotinib Hydrochloride.

templates. To compare the performance with the template-free methods, we train a retrosynthesis Transformer with OpenNMT [40] and compare its top-50 generated outcomes against our discovered path. In this example, neither the path ② nor ③ is found by the retrosynthesis Transformer. We include another example in the **Supplementary Information** with the similar conclusion. These results show the potential of our approach to generate diverse reaction paths with considerable practical values.

Template Database Expansion We further introduce the technique of template database expansion to demonstrate other practical value of the reconstructed template space and its implication of reaction space exploration. This technique is used to mitigate the template coverage limitation of the existing template-based methods. We take reaction prediction, single-step, and multi-step retrosynthesis as our evaluation tasks.

We first evaluate our method for reaction prediction and single-step retrosynthesis. To better illustrate the effect of the database expansion, we build a scenario where the template coverage issue is aggravated. In the conventional USPTO-50K data split used in [5], the template database contains 93.3% of test templates coverage, for which there is little need for the database expansion. To aggravate the coverage issue, we randomly sample 10% of the data with high probability for those whose corresponding template is not covered in the training and validation set but with seen template backbones. The resulted template database contains only 5.0% of test templates coverage. In this case, the original template-based methods perform poorly, while our template database expansion is able to mitigate the coverage issue and improve model's top-k accuracy performance. Prior to the template database expansion, we set a rather conservative template filtering for the generated templates in case of introducing too much noise into the database. We only take the leaving group generator, trained on the new data split, to expand the template database. The new templates are then passed through a strict sanity check ahead of the expansion (see **Methods** section for details). We choose NeuralSym [3] as our evaluation model for the reaction prediction task. To fit its model pipeline, we apply the generated template on all the possible training products to build up the input reaction dataset. Besides NeuralSym, we also choose GLN [4] for the retrosynthesis prediction task. We then compare the model performance between the models trained with the original template database and trained with the expanded template database. Table 2a shows the top-k performance for reaction prediction and Table 2b shows the top-k performance for retrosynthesis prediction. The low accuracy using the original database is

Table 2: Top-k accuracy in downstream synthesis planning tasks.

(a) Reaction prediction of NeuralSym with low template coverage.

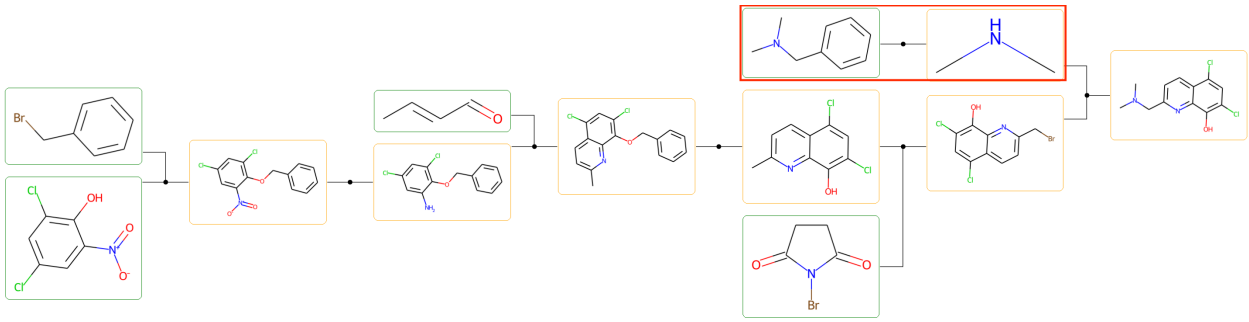
Method	**Template Database**	***Top-k Accuracy (%)***					
		1	3	5	10	20	50
NeuralSym	Original	16.9	26.8	30.9	35.7	39.2	42.5
	Expanded	53.4	72.3	78.3	84.4	88.4	92.6

(b) Retrosynthesis prediction of NeuralSym and GLN with low template coverage.

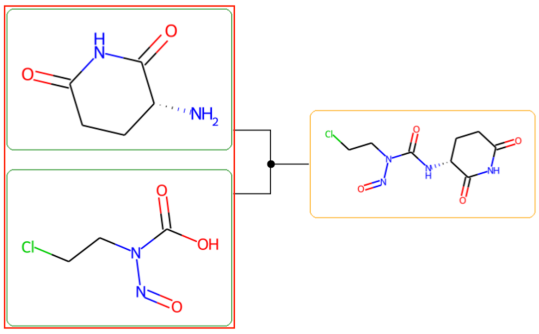
Method	**Template Database**	***Top-k Accuracy (%)***					
		1	3	5	10	20	50
NeuralSym	Original	6.4	13.5	17.1	23.1	27.3	31.6
	Expanded	5.7	15.3	24.3	41.1	56.9	72.2
GLN	Original	15.5	24.7	30.4	37.7	42.8	45.7
	Expanded	26.6	43.4	53.6	67.0	80.4	89.3

expected, indicating the template-based performance is jeopardized by the low template coverage. For the reaction prediction task, the template database expansion is able to improve NeuralSym's performance by $40 \sim 50\%$. For retrosynthesis, there is also a 40.6% increase for the top-50 accuracy. As for GLN, the results show consistent top-k accuracy improvement from 11.1% to 43.6% by our template expansion. Note that there is a performance gap between our results after the expansion and the reported results in the literature, but is expected for the following reasons. First of all, our technique could not completely solve the template coverage issue. Second, the current technique is rather naive and it also introduces additional challenges to these models. For NeuralSym, it is a neural-network based multi-class classification model. The additional challenges are: 1. The output class number increases drastically with the expanded database's size; 2. During building up reaction entries that fit NeuralSym's pipeline, the same product will be mapped to multiple reactions. It implicitly converts the multi-class classification problem into the multi-label classification problem, especially for the retrosynthesis prediction task; 3. Despite the templates' validity filtering ahead of the experiments, noise within the generations might still exist. Building up potential reactions by these noisy templates will enlarge the effect of the noise. These challenges explain the accuracy drop in NeuralSym's top-1 retrosynthesis performance and the slight accuracy boost in top-3$\sim$5 accuracy. For GLN, since it directly uses the learned template embedding to optimize its objective, it does not encounter the second and the third challenges that NeuralSym faces. However, since it uses a graph neural network to capture the parameterization of the templates, its learned embeddings are similar to each other for templates who only differ by leaving groups. It becomes more challenging for the model to differentiate them while performing the ranking. Despite these challenges, the current performance improvement still shows the practical value of the template space exploration.

We then examine the positive effects of the database expansion on the multi-step retrosynthesis problem. Multi-Step retrosynthesis is a more practical and challenging task of recursively decomposing the target molecule into purchasable substances. We adopt the framework of AiZynthFinder [41], which uses the Monte Carlo tree search algorithm over a precursor proposal system. We embed two separate NeuralSym models into this Monte Carlo tree framework: 1. the model trained with the original database; 2. the model trained with the expanded template database. We use a purchasable molecule dataset of approximately 17,422,831 molecules in total from the ZINC database. In addition, we use AiZynthFinder's default In-scope filter [6] to assist the synthesis planning. The specific configuration of the Monte Carlo tree search is described in **Supplementary Information**. Here we enumerate the potential synthesis routes for two commercial substances (*PBT2*, *Zifrosilone*) discovered using the expanded database. *PBT2* is a safe-for-human-use Zinc ionophore, which is an experimental drug candidate [42, 43] and potential treatment of Alzheimer's disease and Huntington's disease. *Zifrosilone* is a potent acetylcholinesterase inhibitor that has been tested in clinical trials [44]. The proposed synthesis routes are displayed in Figure 4. The red box indicates that



(a) Discovered synthesis pathway of commercial substance **PBT2**



(b) Discovered synthesis pathway of commercial substance **Zifrosilone**

Figure 4: Sample cases of multi-step retrosynthesis where the expanded template database is able to discover synthesis pathway that the original template database fails to discover. The rightmost node is the target molecule. The red box indicates that the transformation is not covered in the original database.

such transformation is from our generated template and is not covered in the original template database. This means that the model fails to decompose them when using the original database. It shows that with the task of multi-step retrosynthesis, our expanded template database is able to provide more chemical insights into reaction designs and potentially valuable synthesis routes.

3 Discussion

We propose NeuralTPL, a template-oriented generative method to approach the synthesis planning problems, which combines the advantages of both template-based and template-free methods. Our Transformer-based model is able to effectively capture the existing chemical rules. In addition to the model's ability to reconstruct the ground truth templates, we also show that our approach is superior to the existing methods in terms of validity, diversity, and novelty, indicating its ability to efficiently reconstruct a more valid and comprehensive chemical space. Our approach is able to improve the top-10 reaction validity by 18% compared to the vanilla retrosynthesis Transformer. Our conditional generative pipeline, illustrated by the template customization samples, shows the high diversity of the generated outcomes. We then conduct a thorough template analysis within the learned template space and discover several interesting cases indicating how the model understands and generates the reaction templates.

We further extend our approach to the technique of template database expansion. It shows that NeuralTPL can provide better template space support for the template-based methods, which leads to better synthesis planning performance by mitigating the template coverage issue. We also demonstrate the implication of our approach as a module that can replace any template proposal method that uses reaction center as a conditional factor (e.g., GLN's [4] template proposal). This implies the promising potential of a hybrid template-oriented retrosynthesis model that not only has the ability of reaction space exploration but also

enjoys the validity and diversity advantages of the template-based methods.

One of the challenges of scaling up is the precision issue of the template atom-mapping information from the generated outcomes, since template atom-mapping is required for tools like RDChiral [26] to build full reactions from templates. So far, our experiments rely on additional atom-mapping tools such as RXNMapper [45] to assign such information to the generated templates. All of the existing tools only focus on the mapping for the full reaction rather than the reaction template, and their performance on template atom-mapping is far from being satisfactory. We have experienced at least a 10% percent performance drop from template top-1 accuracy to reaction (built from atom-mapped template) top-1 accuracy because of the noisy mapping. However, template atom-mapping is essentially a subtask of reaction atom-mapping and should be straightforward to adapt these tools for such purpose. We believe it is not a huge impediment for future template applications in a broader context. Another future work direction is the template format standardization, including identifying the most appropriate reaction center and determining the amount of atom attributes that needs to be explicitly shown. Currently, atom radius is one of the hyperparameters of the template extraction algorithm, which determines how large the reaction center should be. It is a simple heuristic that does not contain any deep analysis of whether a particular atom contributes to the reactive process. We also notice that, in some cases, two templates are essentially identical but with different levels of characterization of the reaction center's environment, whereas the model cannot capture the consistency from their SMARTS representation. In addition, the explicit level of atom attributes is also important, especially for sequence-based generative models. So far, our template formation does not contain any explicit atom attribute besides the explicit number of hydrogen. This is because of a trade-off consideration of the sequence generative model between token's frequency versus token's level of granularity. Therefore, for future work, it is promising that a more efficient and interpretable template format is able to provide better chemical insights and application performance driven by deep learning.

Overall, despite the current challenges, we have demonstrated the promising potential of NeuralTPL, as it is able to efficiently explore the reaction space by focusing on the reactive fragments (i.e., template) rather than the full reaction for better synthesis planning algorithms.

4 Methods

Notations We re-illustrate the notations defined in previous sections to remind the readers the meaning of each symbols. For synthesis planning tasks, a reaction RXN contains a product molecule $\mathcal{P}$ and a set of reactants molecule $\mathcal{R} = \{\mathcal{R}_i\}_{i=0}^{N}$. The entire reactants are represented by a single SMILES sequence with each individual reactant SMILES concatenated together by the separator ".". RXN is represented by the SMILES sequence with the separator "»" that links the product SMILES and the reactants SMILES. Reaction template T is represented by the SMARTS sequence with the separator "»" that links the product's reaction centers SMARTS and the reactants' reaction centers SMARTS. Template backbone T' is similar to T except its leaving groups are replaced with the general leaving groups indicator, which the [MASK] token in our experiment.

Data Processing Ground truth templates are extracted from USPTO-50K using RDChiral [26]. Different from its default configuration, we take the explicit number of hydrogen as the only explicit atom attributes shown in SMARTS token. This is because of the trade-off consideration between token's frequency and token's level of granularity for the sequence-based generative models. There are in total 16328 unique templates automatically extracted by the algorithm.

For template generation, we create a special data processing pipeline ahead of the model's training and inference pipeline to merge the input information of both the target molecule and the proposed reaction center. Inspired by the mask language modeling (MLM) proposed in BERT [46], we introduce the molecule's reaction center masking (RCM) to facilitate better learning. The masking function $f_{\text{RCM}}(.)$ takes a molecule SMILES, a reaction center SMARTS, and a mask token (either [MASK] or [NOISE]) as inputs. The function first identifies the reaction center within the molecule. Then, it converts all the atoms and bonds that relate to the reaction center into wildcard characters, groups these characters, and replaces them with the mask

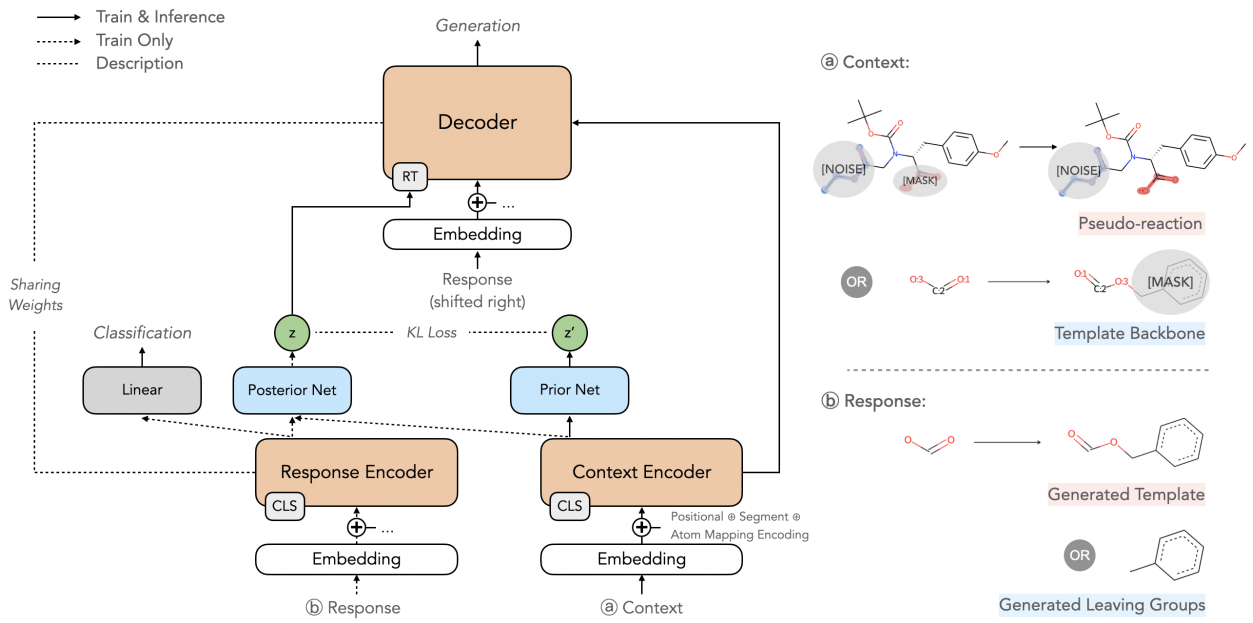


Figure 5: Architecture of the proposed model. The flow described using dashed arrow happens only in train stage. The flow described using solid arrow happens in both train and inference stage. The right part of the figure shows model's inputs and outputs for template generation (red highlight) and leaving groups generation (blue highlight) settings.

token. The template generation data processing pipeline is described by the following steps:

1. Duplicate product $\mathcal{P}$ into the format of $\mathcal{P} \gg \mathcal{P}$;
2. Apply RCM on the left side of the equation with the proposed reaction center p'_0 and the [MASK] token, which ends up with $f_{\text{RCM}}(\mathcal{P}, p'_0, [\text{MASK}]) \gg \mathcal{P}$;
3. (Optional for inference) Apply reaction center masking on both sides with another reaction center p'_1 and the [NOISE] token, which ends up with $f_{\text{RCM}}(f_{\text{RCM}}(\mathcal{P}, p'_0, [\text{MASK}]), p'_1, [\text{NOISE}]) \gg f_{\text{RCM}}(\mathcal{P}, p'_1, [\text{NOISE}])$.

During training, $p'_0 = p$. By masking the ground truth reaction center of $\mathcal{P}$, the model will learn to understand the target reaction center by comparing the difference between the two sides. In the third step, p'_1 is a randomly selected reaction center within $\mathcal{P}$ that is different from p'_0. As implied by the token's name, this masking type is mainly used for introducing noise within the data. One of its goals is to prevent over-fitting, while another functionality is to teach the model to avoid generating the templates with the "noisy" reaction center masked by [NOISE]. As our training strategy, the third step only happens at a probability of 80%. During inference, p'_0 is the proposed reaction center. It can be either manually provided or randomly selected from all the possible reaction centers with the product $\mathcal{P}$. p'_1 is optional for inference. If provided, p'_1 indicates the reaction center which we explicitly want the model to avoid. This operation is to simulate a scenario in which we want to avoid a particular synthetic route during synthesis planning. We term the model input after the data processing as "pseudo-reaction", represented by RXN', since it shares the similar format as $\mathcal{R} \gg \mathcal{P}$, except it does not contain any information regarding the actual transformation. For leaving groups generation, the data processing is simpler. The leaving groups converting (LGC) function $f_{\text{LGC}}(.)$ takes the reactants' reaction centers r as input. Then, it first identifies the leaving groups l and converts them into the general leaving groups indicator token. Here, we use [MASK] token as the indicator $[\text{R}_{\text{LG}}]$ mentioned before. Its output is $\tilde{r}$. The leaving groups generation data processing pipeline proceeds in the following steps:

1. Extract reaction templates $T = p \gg r$ from reaction using RDChiral [26];

2. Apply LGC on reactants' reaction centers: $f_{\text{LGC}}(r) = \tilde{r}$;
3. Build template backbone $T' = p \gg \tilde{r}$.

Model Overview Since the task is essentially a one-to-many generative problem conditioned on the context information, it is analogous to the story completion task in natural language processing. Our design is built on top of the Transformer architecture [47] (briefly described in **Supplementary Information**), with the sequence of pseudo-reaction or template backbone as input, and the sequence of reaction template or leaving groups as output. The overall model contains a context encoder, a response encoder, a decoder, and two latent nets. There are also two embedding layers that capture the continuous vector representation of SMILES and SMARTS tokens. Different from the vanilla version, we redesign the encoding module of the token's attributes. We also adopt the idea of the latent variable z in VT [48] and T-CVAE [49] to better capture the modality of the generative outcomes. Overall, it is a Transformer-based Conditional Variational Autoencoder architecture. Figure 5 is the overview architecture of the model. During training, the context sequence (pseudo-reaction or template backbone) is first passed to the context encoder. Its encoder outputs, including both the sequence representation and the token representations, are passed to the decoder. Meanwhile, the response encoder takes the response sequence (template or leaving groups) and outputs the response sequence representation. The response sequence representation and context sequence representation are concatenated together for the posterior net to learn the posterior distribution of latent variable z. The response sequence representation itself is also passed to a linear classifier for reaction type classification. At the same time, the prior net takes the context sequence representation to learn the prior distribution of latent variable z'. KL-divergence is adopted to enforce the similarity between the two distributions. Then, z is sampled from the posterior distribution and added to the embedding of the initial token of the decoder's input sequence. Finally, the decoder takes the context encoder outputs, the latent variable z, and the response sequence (shifted right) for the paralleled language modeling training. During inference, the response encoder does not participate in the outcome generation. It only acts as an individual module that outputs the template embedding. The molecule sequence is generated step-by-step with latent variable z sampled from the prior distribution.

Embedding and Encoding Layers In addition to the trainable SMILES and SMARTS token embedding layer, positional encoding is applied to handle the sequential information within the input. We also apply the segment encoding used in BERT [46] since the model needs to differentiate the product and the reactants portion within the reaction or template. We then propose the atom mapping encoding to capture the mapping relationship between reactive atoms. One of the issues with directly using the atom-mapped token is that model could treat two atoms differently even if they are essentially the same from the chemical perspective. For example, [NH2:1] is the SMARTS representation of primary amine with atom mapping number "1". Note that this number does not contain any additional chemical meaning besides how it is mapped within the reaction. In other words, from the chemical view, the model should treat [NH2:x] with $x \in \{1, 2, ...\}$, the same. Therefore, we extract the atom mapping number of each token and embed it separately into the token attribute encoding layer. The final token vector representation is a summation of token embedding, positional encoding, segment encoding, and atom-mapping encoding.

Encoders The basic encoder architecture follows the same structure as the vanilla Transformer. We adopt the design in BERT [46] and add the [CLS] token at the beginning of the sequence to capture the sequence representation for both the context encoder and the response encoder. We take the sequence representation from the response encoder as the template embedding. The response encoder shares the same embedding layer and multi-head self-attention module as the decoder, which is used primarily for learning the template sequence representation.

Latent Nets The latent variable z is used to better capture the modality of generative outcomes given the input sequence. Such variable is another conditional factor for both the template and leaving groups generation. The template generation's objective $p(T|RXN')$ is illustrated as followed. For leaving groups

generation setting, the formulation is simply modified by replacing RXN' with T', RXN with T, and T with l.

$$p(T|RXN') = \int_z p(T|RXN', z)p(z|RXN')dz \tag{1}$$

In our architecture, $p(T|RXN', z)$ is the decoder and $p(z|RXN')$ is the prior net. Since the likelihood $p(T|RXN')$ is intractable [50], the typical CVAE objective is to optimize its evidence lower bound (ELBO), by assuming z follows a multivariate Gaussian distribution with a diagonal co-variance matrix,

$$\log p(T|RXN') \leq \mathcal{L}_{ELBO} = \mathbb{E}_{q(z|RXN',T)}\Big[\log p(T|RXN', z)\Big] - KL\Big(q(z|RXN', T)||p(z|RXN')\Big) \tag{2}$$

where $q(z|RXN', T)$ is the posterior net (i.e., recognition net). In order to estimate $p(z|RXN')$ and $q(z|RXN', T)$ by neural networks, it requires a fixed dimension representation for the inputs, which is the sequence representation (corresponding to the [CLS] token) of RXN' and T learned by our response encoder and context encoder.

Decoder The decoder architecture follows the same structure as the vanilla Transformer [47]. Each layer contains a masked multi-head self-attention module, a feed-forward module, and a multi-head attention module that attends to the outputs of the context encoder. The initialize token of the decoder's input is the reaction class token. This is different from how other Transformer-based synthesis planning models treat the reaction class where they treat it as the encoder's input. We find that initializing the decoder generation directly with the reaction class token can lead to more diverse outcomes. It also matches with the intuition such that the input sequence understanding should not be affected by the generation condition. Additionally, we integrate the latent variable z into the decoder. Common approach in RNN-based CVAE [51] uses latent variable to initialize the decoder state. In our case, we adopt the idea from VT [48] and incorporate the latent variable z into the embedding of the initial reaction class token [RT].

$$e'_{RT} = e_{RT} + z \tag{3}$$

Losses The overall training loss is composed of the language modeling cross-entropy loss, the reaction type classification cross-entropy loss, and the KL-divergence loss: $\mathcal{L} = \mathcal{L}_{LM} + \mathcal{L}_{RT} + \mathcal{L}_{KL}$. The traditional language modeling objective of the sequence-to-sequence tasks is essentially the next token classification given the existing sequence, which is to learn the probability $P(x_t|x_0, ..., x_{t-1})$ at each time step t. $\mathcal{L}_{LM}$ is then the cross-entropy loss of these token classifications. Additional to the language modeling objective of the target sequence, we also introduce the template classification objective. With the shared weights between the response encoder and the decoder, it means that template representation is learned jointly under the generation task and the classification task. $\mathcal{L}_{RT}$ is the cross-entropy between the predicted reaction class and the ground truth label. KL-divergence loss of the latent variable z is the exact term shown as above: $\mathcal{L}_{KL} = KL(q(z|RXN', T)||p(z|RXN'))$.

Template Generation Algorithm The pseudo-code of the general template space exploration is shown below. There are two settings for the template generation: main template generation TemplateGeneration(.), and the template generation formed by novel leaving groups attachment TemplateGeneration_lvgp(.). GetCenter(.) takes a molecule SMILES and reaction class rt and returns all its potential reaction centers. It relies on a fragment database, which follows the same idea as GLN [4]. GenTemplate(.) is the main template generation function. It is a wrapper of the model generation module that takes a pseudo-reaction RXN', reaction class rt, and the number of output candidates k as inputs. GenLvgp(.) is the leaving group generation function, with input T' as the template backbone. Attach(.) takes a leaving group l and a template backbone T' and merge them into T. BasicSanityCheck(.) takes a list of templates and returns only those who can be successfully identified as a reaction by RDKit. LvgpSanityCheck(.) is an additional sanity check that is used specifically under the leaving group generation setting. It applies three filtering rules as described in the section below.

Algorithm 1 Template generation for reaction space exploration

```
function TemplateGeneration(molData, k)
    Initialize result as an empty list;
    for molecule ∈ molData do
        for rt = 1, 2, ..., 10 do
            RCS = GetCenter(molecule, rt)
            for rc ∈ RCS do
                Build RXN′;
                T̂s = GenTemplate(RXN′, rt, k);
                result += BasicSanityCheck(T̂s);
            end for
        end for
    end for
    return result
end function
```

```
function TemplateGeneration_lvgp(backboneData, k)
    Initialize result as an empty list;
    for T′ ∈ backboneData do
        for rt = 1, 2, ..., 10 do
            l̂s = GenLvgp(T′, rt, k)
            T̂s = Attach(T′, l̂s)
            T̂s = BasicSanityCheck(T̂s);
            result += LvgpSanityCheck(T̂s);
        end for
    end for
    return result
end function
```

During template analysis, to explore the chemical novelty, generated outcomes with high similarity as the ground truth templates are screened out. We use cosine similarity between template embedding to calculate the similarity. Then, the remaining templates are manually screened by the professional regarding its validity. Next, templates are turned into specific reactions by either applying them on a list of molecule or manual design. At last, these reactions will be searched against the Reaxys database, and their thermodynamic feasibility will be verified through DFT calculations. As for database expansion experiments, we add a couple of additional rules to make validity check as strict as possible corresponding to the LvgpSanityCheck(.) function, since false positive has a much higher cost than false negative under the downstream task setting.

- **Rule 1** Whether there exists a product $\mathcal{P}$ in the training set such that $\hat{T}$ can successfully apply on it using RDChiral [26].
- **Rule 2** Whether the reactants $\mathcal{R}$ generated by applying $\hat{T}$ on $\mathcal{P}$ is a stable molecule (by identifying [CH] in molecule SMILES).
- **Rule 3** Whether the new leaving group $\hat{l}$ exists in a larger scale of chemical reaction dataset, which is in our case the USPTO-full dataset [4].

References

[1] Blakemore, D. C. *et al.* Organic synthesis provides opportunities to transform drug discovery. *Nature Chemistry* **10**, 383–394 (2018). URL https://doi.org/10.1038/s41557-018-0021-z.

[2] Segler, M. H. S. & Waller, M. P. Modelling chemical reasoning to predict and invent reactions. *Chemistry - A European Journal* **23**, 6118–6128 (2017). URL http://dx.doi.org/10.1002/chem.201604556.

[3] Segler, M. H. S. & Waller, M. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry* **23 25**, 5966–5971 (2017).

[4] Dai, H., Li, C., Coley, C., Dai, B. & Song, L. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems*, 8870–8880 (2019).

[5] Coley, C. W., Rogers, L., Green, W. H. & Jensen, K. F. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science* **3**, 1237–1245 (2017). URL https://doi.org/10.1021%2Facscentsci.7b00355.

[6] Segler, M. H. S., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature* **555**, 604–610 (2018). URL https://doi.org/10.1038/nature25978.

[7] Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. & Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS Central Science* **3**, 434–443 (2017). URL https://doi.org/10.1021/acscentsci.7b00064. PMID: 28573205, https://doi.org/10.1021/acscentsci.7b00064.

[8] Chen, S. & Jung, Y. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au* (2021). URL https://doi.org/10.1021/jacsau.1c00246.

[9] Liu, B. *et al.* Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Science* **3**, 1103–1113 (2017). URL https://doi.org/10.1021/acscentsci.7b00303. PMID: 29104927, https://doi.org/10.1021/acscentsci.7b00303.

[10] Chen, B., Shen, T., Jaakkola, T. S. & Barzilay, R. Learning to make generalizable and diverse predictions for retrosynthesis (2019). 1910.09688.

[11] Tetko, I. V., Karpov, P., Van Deursen, R. & Godin, G. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature Communications* (2020). URL https://doi.org/10.1038/s41467-020-19266-y.

[12] Schwaller, P. *et al.* Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science* **5**, 1572–1583 (2019). URL https://doi.org/10.1021/acscentsci.9b00576. https://doi.org/10.1021/acscentsci.9b00576.

[13] Zheng, S., Rao, J., Zhang, Z., Xu, J. & Yang, Y. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling* **60**, 47–55 (2020). URL https://doi.org/10.1021/acs.jcim.9b00949. PMID: 31825611, https://doi.org/10.1021/acs.jcim.9b00949.

[14] Wang, X. *et al.* Retroprime: A diverse, plausible and transformer-based method for single-step retrosynthesis predictions. *Chemical Engineering Journal* **420**, 129845 (2021). URL https://www.sciencedirect.com/science/article/pii/S1385894721014303.

[15] Lin, K., Xu, Y., Pei, J. & Lai, L. Automatic retrosynthetic route planning using template-free models. *Chem. Sci.* **11**, 3355–3364 (2020). URL http://dx.doi.org/10.1039/C9SC03666K.

[16] Somnath, V. R., Bunne, C., Coley, C. W., Krause, A. & Barzilay, R. Learning graph models for template-free retrosynthesis (2020). 2006.07038.

[17] Shi, C., Xu, M., Guo, H., Zhang, M. & Tang, J. A graph to graphs framework for retrosynthesis prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 8818–8827 (2020).

[18] Yan, C. *et al.* Retroxpert: Decompose retrosynthesis prediction like a chemist. In *Advances in Neural Information Processing Systems*, 11248–11258 (Curran Associates, Inc., 2020).

[19] Graves, A. Sequence transduction with recurrent neural networks (2012). 1211.3711.

[20] Vijayakumar, A. K. *et al.* Diverse beam search: Decoding diverse solutions from neural sequence models (2018). 1610.02424.

[21] Spero, M. Improved beam search diversity for neural machine translation with k-dpp sampling (2019).

[22] Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M. & Callison-Burch, C. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3752–3762 (Association for Computational Linguistics, Florence, Italy, 2019). URL https://aclanthology.org/P19-1365.

[23] Bort, W. *et al.* Discovery of novel chemical reactions by deep generative recurrent neural network. *Scientific Reports* **11**, 6118–6128 (2021). URL https://doi.org/10.1038/s41598-021-81889-y.

[24] Xu, Z., Wang, S., Zhu, F. & Huang, J. 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*, ACM-BCB '17, 285–294 (Association for Computing Machinery, New York, NY, USA, 2017). URL https://doi.org/10.1145/3107411.3107424.

[25] Corey, E. & Cheng, X. *The Logic of Chemical Synthesis* (Wiley, 1989).

[26] Coley, C. W., Green, W. H. & Jensen, K. F. Rdchiral: An rdkit wrapper for handling stereochemistry in retrosynthetic template extraction and application. *Journal of Chemical Information and Modeling* **59**, 2529–2537 (2019). URL https://doi.org/10.1021/acs.jcim.9b00286. https://doi.org/10.1021/acs.jcim.9b00286.

[27] Rdkit: Open-source cheminformatics URL http://www.rdkit.org.

[28] Suzuki, A. Carbon–carbon bonding made easy. *Chemical Communications* 4759–4763 (2005). URL http://dx.doi.org/10.1039/B507375H.

[29] Suzuki, A. Recent developments of biaryl synthesis via cross-coupling reactions of areneboronic acid derivatives. *Yuki Gosei Kagaku Kyokaishi(Journal of Synthetic Organic Chemistry)* **63**, 312–324 (2005).

[30] Suzuki, A. Organoborane coupling reactions (suzuki coupling). *Proceedings of the Japan Academy, Series B* **80**, 359–371 (2004).

[31] Miyaura, N. & Suzuki, A. Palladium-catalyzed cross-coupling reactions of organoboron compounds. *Chemical Reviews* **95**, 2457–2483 (1995). URL https://doi.org/10.1021/cr00039a007.

[32] Lennox, A. J. J. & Lloyd-Jones, G. C. Selection of boron reagents for suzuki-miyaura coupling. *Chemical Society Reviews* **43**, 412–443 (2014). URL http://dx.doi.org/10.1039/C3CS60197H.

[33] Nishikata, T., Abela, A. R., Huang, S. & Lipshutz, B. H. Cationic palladium(ii) catalysis: C-h activation/suzuki-miyaura couplings at room temperature. *Journal of the American Chemical Society* **132**, 4978–4979 (2010). URL https://doi.org/10.1021/ja910973a.

[34] Luo, Y. Comprehensive handbook of chemical bond energies (CRC Press., 2007).

[35] Nahm, S. & Weinreb, S. M. N-methoxy-n-methylamides as effective acylating agents. *Tetrahedron Letters* **22**, 3815–3818 (1981). URL https://www.sciencedirect.com/science/article/pii/S0040403901913164.

[36] Drugdataexpy. URL https://data.yaozh.com. Accessed: 2021-12-02.

[37] Ma, X. H. *et al.* Efficacy and influence factors of icotinib hydrochloride in treating advanced non-small cell lung cancer. *Eur Rev Med Pharmacol Sci* **21**, 266–274 (2017).

[38] Hu, S. *et al.* Synthesis and biological evaluation of crown ether fused quinazoline analogues as potent egfr inhibitors. *Bioorganic & Medicinal Chemistry Letters* **22**, 6301–6305 (2012). URL https://www.sciencedirect.com/science/article/pii/S0960894X12008189.

[39] Don, Z. *et al.* Novel fused quinazoline derivatives useful as tyrosine kinase inhibitors (U.S. Patent 7078409B2, Jul. 18 2006). URL https://worldwide.espacenet.com/publicationDetails/biblio?CC=WO&NR=03082830&KC=&FT=E&locale=en_EP.

[40] Klein, G., Kim, Y., Deng, Y., Senellart, J. & Rush, A. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, 67–72 (Association for Computational Linguistics, Vancouver, Canada, 2017). URL https://aclanthology.org/P17-4012.

[41] Genheden, S. *et al.* Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics* 70 (2020).

[42] Bohlmann, L. *et al.* Chemical synergy between ionophore pbt2 and zinc reverses antibiotic resistance. *MBio* **9** (2018).

[43] Crouch, P. J. *et al.* The alzheimer's therapeutic pbt2 promotes amyloid-β degradation and gsk3 phosphorylation via a metal chaperone activity. *Journal of neurochemistry* **119**, 220–230 (2011).

[44] Cutler, N. R. *et al.* Acetylcholinesterase inhibition by zifrosilone: pharmacokinetics and pharmacodynamics. *Clinical Pharmacology & Therapeutics* **58**, 54–61 (1995).

[45] Schwaller, P., Hoover, B., Reymond, J.-L., Strobelt, H. & Laino, T. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances* **7**, eabe4166 (2021).

[46] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). URL https://aclanthology.org/N19-1423.

[47] Vaswani, A. *et al.* Attention is all you need. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017). URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[48] Lin, Z., Winata, G. I., Xu, P., Liu, Z. & Fung, P. Variational transformers for diverse response generation (2020). 2003.12738.

[49] Wang, T. & Wan, X. T-cvae: Transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5233–5239 (International Joint Conferences on Artificial Intelligence Organization, 2019). URL https://doi.org/10.24963/ijcai.2019/727.

[50] Kingma, D. P. & Welling, M. Auto-encoding variational bayes (2014). 1312.6114.

[51] Bowman, S. R. *et al.* Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 10–21 (Association for Computational Linguistics, Berlin, Germany, 2016). URL https://aclanthology.org/K16-1002.