


Sequence analysis

Back translation for molecule generation

Yang Fan¹, Yingce Xia ^{2,*}, Jinhua Zhu¹, Lijun Wu², Shufang Xie² and Tao Qin²

¹University of Science and Technology of China, Hefei, Anhui 230027, China and ²Microsoft Research, Beijing 100080, China

*To whom correspondence should be addressed.

Associate Editor: Jinbo Xu

Received on July 2, 2021; revised on November 11, 2021; editorial decision on November 23, 2021; accepted on December 1, 2021

Abstract

Motivation: Molecule generation, which is to generate new molecules, is an important problem in bioinformatics. Typical tasks include generating molecules with given properties, molecular property improvement (i.e. improving specific properties of an input molecule), retrosynthesis (i.e. predicting the molecules that can be used to synthesize a target molecule), etc. Recently, deep-learning-based methods received more attention for molecule generation. The labeled data of bioinformatics is usually costly to obtain, but there are millions of unlabeled molecules. Inspired by the success of sequence generation in natural language processing with unlabeled data, we would like to explore an effective way of using unlabeled molecules for molecule generation.

Results: We propose a new method, back translation for molecule generation, which is a simple yet effective semi-supervised method. Let \mathcal{X} be the source domain, which is the collection of properties, the molecules to be optimized, etc. Let \mathcal{Y} be the target domain which is the collection of molecules. In particular, given a main task which is about to learn a mapping from the source domain \mathcal{X} to the target domain \mathcal{Y} , we first train a reversed model g for the \mathcal{Y} to \mathcal{X} mapping. After that, we use g to back translate the unlabeled data in \mathcal{Y} to \mathcal{X} and obtain more synthetic data. Finally, we combine the synthetic data with the labeled data and train a model for the main task. We conduct experiments on molecular property improvement and retrosynthesis, and we achieve state-of-the-art results on four molecule generation tasks and one retrosynthesis benchmark, USPTO-50k.

Availability and implementation: Our code and data are available at <https://github.com/fyabc/BT4MolGen>.

Contact: yingce.xia@microsoft.com

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Molecule generation (Imrie *et al.*, 2021), which aims to generate new molecules based on the input, is an important problem in bioinformatics. The input could be some random noise (De Cao and Kipf, 2018), some properties (Xie *et al.*, 2021), or the molecules to be improved (Jin *et al.*, 2020), etc. Various biomedical and chemistry tasks can be formulated as molecule generation. For example, molecular property improvement is to improve specific properties of a molecule, like the quantitative estimation of drug-likeness (Bickerton *et al.*, 2012) (briefly, QED), partition coefficient, etc. The input is the molecule, and the output is another one which is similar to the input with certain properties improved. Retrosynthesis (Corey, 1991) is another example which aims to predict the reactants (i.e. a collection of molecules) that can be used to synthesize a target molecule.

Inspired by the success of deep learning in cognitive tasks such as image classification (Krizhevsky *et al.*, 2012) and machine translation (Vaswani *et al.*, 2017), researchers start using deep neural networks for molecule generation. Olivecrona *et al.* (2017) and Popova *et al.* (2018) leveraged deep reinforcement learning to design *de novo* drugs. Stokes *et al.* (2020) used deep learning for drug repurposing. The more recent neural network architectures like

Transformer (Grechishnikova, 2021; Karpov *et al.*, 2019a), graph neural network (Dai *et al.*, 2019; Shi *et al.*, 2020) and generative adversarial networks (De Cao and Kipf, 2018) are also used in molecule generation and obtain great improvements.

The labeled data of bioinformatics are usually cost to obtain. In comparison, the unlabeled data are much easier to collect. For example, the ZINC (Sterling and Irwin, 2015) database contains more than 750 million compounds, and PubChem has 109 million compounds. There are some work leveraging unlabeled data for molecule modeling and generation. One way is to use variational autoencoder (VAE) to learn the representations of molecules, where an input molecule x is first mapped to some high-dimensional representation h and then reconstructed based on h . The distribution of h is constraint to be close to some prior distribution. Kang and Cho (2019) introduced VAE into molecule generation to leverage unlabeled data, and predict molecular property together with molecule generation. Gómez-Bombarelli *et al.* (2018) also use the VAE framework, and the molecules can be optimized by applying gradient ascent on the predictor. Born *et al.* (2021) jointly use VAE and reinforcement learning, and their model can generate cancer-type-specific candidate drugs that are similar to cancer drugs in drug-likeness, synthesizability and solubility. Moreover,

Kotsias *et al.* (2020) propose a new method that use conditional recurrent neural network (cRNN) to tackle the inverse design problem directly, in which the cRNN may generate molecules near the specified conditions. Witnessing the fast development of pretraining techniques in natural language processing (Devlin *et al.*, 2019; Liu *et al.*, 2019) and computer vision (Chen *et al.*, 2020), researchers also apply pretraining techniques to molecule classification tasks (Chithrananda *et al.*, 2020; Li *et al.*, 2020).

In this work, we propose back translation for molecule generation, that can effectively leverage unlabeled data for the molecule generation. Assume that our main task is to learn a mapping f to convert a molecule from the source domain \mathcal{X} to the target domain \mathcal{Y} . With our method, we need to additionally train a reversed model g on the labeled training corpus, that can map a molecule from \mathcal{Y} to \mathcal{X} .

The high-level idea of our work is shown as follows. Given any molecule $x \in \mathcal{X}$, we can first generate a corresponding molecule in \mathcal{Y} by $\hat{y} = f(x)$, and then use model g to convert \hat{y} back to \hat{x} , i.e. $\hat{x} = g(\hat{y})$. Ideally, x and \hat{x} should be the same, but empirically, they are different since both f and g are learned from data. Therefore, the difference between x and \hat{x} could be used as a training signal to regularize the training. We propose an efficient algorithm for the above strategy: given a collection unlabeled molecules in \mathcal{Y} , we back translate them using g and obtain the corresponding pseudo ‘source’ molecules in \mathcal{X} . After that, we leverage both the labeled dataset and the synthetic dataset to train the model f for our main task. Note that model f and g can be any model architecture for the main task and the reversed task. We verify our algorithm on four molecular property improvement tasks and one retrosynthesis task. For molecular property improvement, we conduct experiments on the four tasks proposed by Jin *et al.* (2020), which is to improve properties of the input molecules while maintaining the similarities. We use two types of models, Transformer (Zheng *et al.*, 2020) and hierarchical graph-to-graph translation (Jin *et al.*, 2020) to verify the algorithm and achieve state-of-the-art results on them. For retrosynthesis, we conduct experiments on USPTO-50k datasets, which is widely adopted in the literature. We implement our method with the Transformer and GLN (Dai *et al.*, 2019) backbones, and obtain significant improvement over the corresponding baselines. We release our data and code at github for reproducibility.

2 Methods

In this section, we introduce our proposed method and discuss the relation with related methods.

2.1 Notations and problem setup

Let \mathcal{X} and \mathcal{Y} denote the source domain and target domain, respectively, which are both collections of molecules. For molecular property improvement, \mathcal{X} and \mathcal{Y} are molecules with relatively poor and good properties; for retrosynthesis, \mathcal{X} and \mathcal{Y} are the sets of target molecules and reactants, respectively. Our task is to learn a mapping $f: \mathcal{X} \rightarrow \mathcal{Y}$, that can generate the molecules in the target domain based on the input from the source domain.

Denote the labeled data as $\mathcal{L} \subset \mathcal{X} \times \mathcal{Y}$. Denote the unlabeled data from the target domains as $\mathcal{U}_y \subset \mathcal{Y}$.

2.2 Algorithm

As introduced in Section 1, we introduce a reversed model g , that can achieve the mapping from \mathcal{Y} to \mathcal{X} . Denote the parameters of model f and g as θ_f and θ_g , respectively. Given an unlabeled molecule $y_u \in \mathcal{U}_y$, after mapping by g and f sequentially, we can obtain $\hat{x}_u = g(y_u)$ and $\hat{y}_u = f(\hat{x}_u)$. Let $P(\hat{x}_u|y_u; g)$ denote the probability that we can obtain \hat{x}_u from y_u by using model g , and let $P(\hat{y}_u|\hat{x}_u; f)$ denote the probability that we can recover \hat{y}_u from \hat{x}_u by f . The logarithmic reconstruction probability can be written as

$$\begin{aligned} \log P(y_u = \hat{y}_u|y_u; g, f) &= \log \sum_{\hat{x}_u \in \mathcal{X}} P(y_u = \hat{y}_u, \hat{x}_u|y_u; g, f) \\ &= \log \sum_{\hat{x}_u \in \mathcal{X}} P(\hat{x}_u|y_u; g, f) P(y_u = \hat{y}_u|\hat{x}_u, y_u; g, f). \end{aligned} \quad (1)$$

Considering that \mathcal{X} is exponentially large, it is not practical to work on Equation (1) directly. Alternatively, we can maximize the lower bound of the reconstruction probability (Dempster *et al.*, 1977). Due to the concavity of $\log(\cdot)$, Equation (1) can be further decomposed as

$$\begin{aligned} \log P(y_u = \hat{y}_u|y_u; g, f) \\ \geq \sum_{\hat{x}_u \in \mathcal{X}} P(\hat{x}_u|y_u; g, f) \log P(y_u = \hat{y}_u|\hat{x}_u, y_u; g, f). \end{aligned} \quad (2)$$

Considering \hat{x}_u is only related to y_u and g , we have $P(\hat{x}_u|y_u; g, f) = P(\hat{x}_u|y_u; g)$. When \hat{x}_u is given, reconstructing \hat{y}_u is related to f only, thus $P(y_u = \hat{y}_u|\hat{x}_u, y_u; g, f) = P(y_u = \hat{y}_u|\hat{x}_u; f)$. Therefore, inequality (2) can be further written as

$$\begin{aligned} \log P(y_u = \hat{y}_u|y_u; g, f) &\geq \sum_{\hat{x}_u \in \mathcal{X}} P(\hat{x}_u|y_u; g) \log P(y_u = \hat{y}_u|\hat{x}_u; f) \\ &= \mathbb{E}_{\hat{x}_u \sim P(\cdot|y_u; g)} \log P(y_u = \hat{y}_u|\hat{x}_u; f). \end{aligned} \quad (3)$$

We can use a Monte Carlo method to optimize Equation (3), which consists of three steps:

1. We first train a model f and a model g on \mathcal{L} by

$$\begin{aligned} \min_{\theta_f} \sum_{(x,y) \in \mathcal{L}} -\log P(y|x; \theta_f); \\ \min_{\theta_g} \sum_{(x,y) \in \mathcal{L}} -\log P(x|y; \theta_g). \end{aligned} \quad (4)$$

2. After obtaining θ_g (i.e. g), for each $y_u \in \mathcal{U}_y$, we generate the corresponding \hat{x}_u using g and obtain

$$\hat{\mathcal{L}} = \{(\hat{x}_u, y_u) | y_u \in \mathcal{U}_y, \hat{x}_u \text{ is sampled from } P(\cdot|y_u; \theta_g)\}. \quad (5)$$

3. We retrain the model f on $\mathcal{L} \cup \hat{\mathcal{L}}$, where the parameters are initialized from the θ_f obtained in step (1). That is,

$$\min_{\theta_f^*} \sum_{(x,y) \in \mathcal{L} \cup \hat{\mathcal{L}}} -\log P(y|x; \theta_f^*). \quad (6)$$

We will eventually obtain the θ_f^* , which will be used for our main task. The workflow is also illustrated in Figure 1.

2.3 Discussion

1. We do not have specific requirement for the choice of f and g . Any model that serves for the mapping $\mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{Y} \rightarrow \mathcal{X}$ can be leveraged in our algorithm, which makes it flexible.
2. There are some work about leveraging a reversed model to boost the main task in natural language processing. Sennrich *et al.* (2016) first proposed to use back translation to augment the data and Edunov *et al.* (2018) implement this idea on a large scale of the data. He *et al.* (2016) jointly train the f and g . Wang *et al.* (2019) extend He *et al.* (2016) to a multiagent version, where multiple models for forward translation and back translation are applied. The above works share similar insight with our method. As a first step of applying unlabeled data for molecular generation, we design a simple yet effective method and leave the extension to more comprehensive versions as future work.

3 Results on molecular property improvement

In this section, we first verify our approach on the molecular property improvement task.

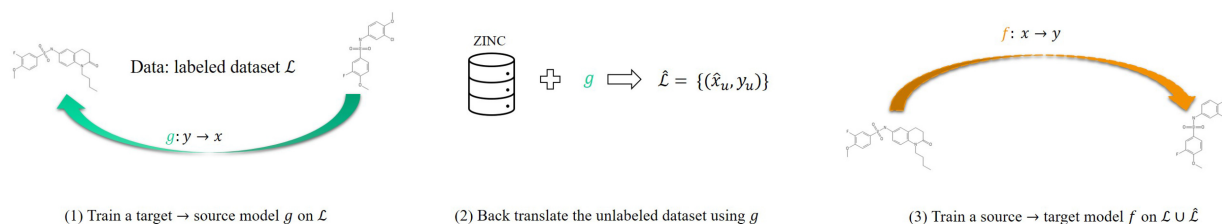


Fig. 1. The workflow of our proposed method

3.1 Setup

The molecular property improvement task [also known as molecule optimization in Jin et al. (2019)] is about to improve specific properties of the input molecule. Following the setup of Jin et al. (2018, 2020), the input molecule x and output molecule y must satisfy the similarity threshold $\text{sim}(x, y) \geq \delta$ to prevent the model from generating arbitrary molecules, where sim is calculated by the Dice similarity provided in RDKit (Landrum, 2016). By default, in this work, we set $\delta = 0.4$. The task is to generate a ‘good’ target molecule y with higher specific property from a ‘bad’ source molecule x under this similarity constraint.

Dataset: We use the four datasets used in Jin et al. (2019, 2020), which are about to improve the LogP, QED and DRD2 of the input molecules. Specifically, (i) the penalized LogP score (Kusner et al., 2017) measures the solubility and synthetic accessibility of a compound. For LogP, we conduct two settings with different similarity thresholds, $\delta = 0.4$ and $\delta = 0.6$; (ii) the qualitative estimation of drug-likeness (briefly, QED), which is utilized to quantify the drug-likeness of a compound. (iii) DRD2, which evaluates the biological activity of a compound against the dopamine type 2 receptor (DRD2). The bioactivity value is assessed by a property prediction model from Olivecrona et al. (2017). All of these properties are expected to get larger values. The statistics of the training, validation and test samples of the above four tasks are summarized in Table 1.

We explore different amounts of unlabeled data, 250K and 1M, that are randomly selected from ZINC. We work on two settings: (i) we keep all the back translated data to enlarge the dataset; (ii) we only keep the (\hat{x}_u, y_u) pairs satisfying $\text{sim}(\hat{x}_u, y_u) \geq \delta$, and the property improvement is the same as that of the labeled training data. For QED, the property of the input molecule lies in $[0.7, 0.8]$, and that of the output molecule lies in $[0.9, 1.0]$. For DRD2, the property of the input molecule is inactivate ($P < 0.5$) and that of output is activate ($P \geq 0.5$).

Network architecture: We adopt a well-known sequence-to-sequence model, Transformer (Vaswani et al., 2017) and a graph-based model HierG2G (Jin et al., 2020) as backbones of our method. For Transformer, both the forward model f and backward model g are Transformers with 6 layers, 4 attention heads, embedding dimension 128 and feed-forward dimension 512. For HierG2G, both f and g follow the settings in Jin et al. (2020).

Evaluation metrics: In each task, given a source molecule x , we use beam search and output top- k predictions $\{y_1, y_2, \dots, y_k\}$ and only keep the translations that satisfy the similarity constraint. In this work, $k = 20$. For LogP tasks, we calculate the highest property score improvement for each data, i.e. $\max_i(\log P(y_i) - \log P(x))$, and report the average property improvement over the test set. For other two tasks, we report the success rate. Specifically, for QED, the task is to translate molecules with QED scores in $[0.7, 0.8]$ into a higher range $[0.9, 1.0]$. For DRD2, the task is to optimize the biological activity against DRD2 of a molecule, where we need to translate a molecule from inactivate state ($P < 0.5$) to activate state ($P \geq 0.5$). The similarity constraint is $\delta \geq 0.4$ for QED and DRD2.

We also report the diversity and novelty scores. Specifically, given a molecule x , the diversity score is defined as the average pairwise molecular distances among those predictions, i.e. $\frac{1}{k(k-1)} \sum_{i \neq j} 1 - \text{sim}(y_i, y_j)$, where $\text{sim}(y_i, y_j)$ is the similarity over

Table 1. Statistics of the datasets of molecular property improvement

Task	#Training	#Validation	#Test
LogP ($\delta \geq 0.4$)	98 851	200	800
LogP ($\delta \geq 0.6$)	74 887	200	800
QED	87 226	360	800
DRD2	34 234	500	1000

Table 2. Results of molecular property improvement

Methods	LogP0.6	LogP0.4	QED (%)	DRD2 (%)
JT-VAE (Jin et al., 2018)	0.28	1.03	8.8	3.4
CG-VAE (Liu et al., 2018)	0.25	0.61	4.8	2.3
GCPN (You et al., 2018)	0.79	2.49	9.4	4.4
MMPA (Dalke et al., 2018)	1.65	3.29	32.9	46.4
seq2seq (Jin et al., 2020)	2.33	3.37	58.5	75.9
JTNN (Jin et al., 2019)	2.33	3.55	59.9	77.8
Transformer (Vaswani et al., 2017) backbone				
Baseline	2.45	3.69	71.9	60.2
+BT(250K)	2.71	4.23	81.5	66.5
+BT(250K, filtered)	2.79	4.34	82.1	67.1
+BT(1M)	2.39	3.71	75.1	50.1
+BT(1M, filtered)	2.86	4.41	82.9	67.4
HierG2G (Jin et al., 2020) backbone				
Baseline	2.49	3.98	76.9	85.9
+BT(250K)	2.67	4.19	79.0	86.9
+BT(250K, filtered)	2.75	4.24	79.1	87.3
+BT(1M)	2.48	4.00	76.4	84.6
+BT(1M, filtered)	2.77	4.21	79.4	87.2

Note: All of the best property values are marked in bold.

Morgan fingerprints (Rogers and Hahn, 2010) of two molecules. Following Jin et al. (2019), the novelty score is defined as $1 - |\hat{\mathcal{Y}} \cap S|/|S|$, where S is the set of all molecules used in training, and $\hat{\mathcal{Y}}$ is the set of all generated molecules.

3.2 Results

The results are reported in Table 2, where baseline refers to the setting without unlabeled data, and the numbers of unlabeled data are in the brackets after ‘BT’. For the settings where the unlabeled data are filtered, they are tagged by ‘filtered’.

We can see that back translation works for molecular property improvement, no matter for Transformer backbone or HierG2G backbone. Also, simply using more unlabeled data hurts the performance. Taking the QED dataset with Transformer backbone as an example, when using 1M unlabeled data, the performance is worse than that of 250K. However, after applying data filtration, the performance improves to 82.9%, outperforming the results with 250K unlabeled data only. This shows the importance of data filtration when using unlabeled data.

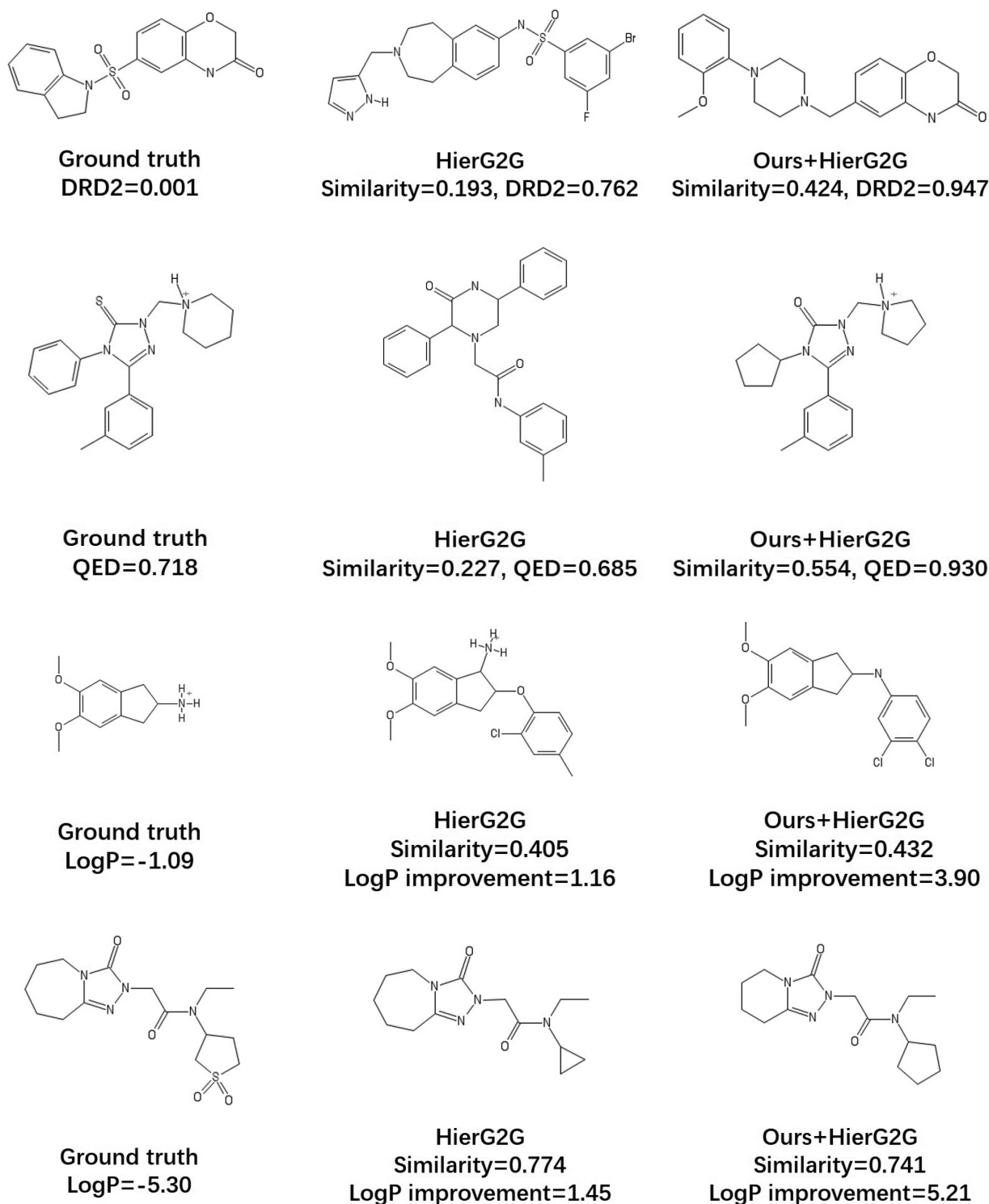


Fig. 2. Comparison of generated molecules between HierG2G with and without our approach. Each row from top to bottom is examples selected from DRD2, QED, LogP (sim \geq 0.4), LogP (sim \geq 0.6), respectively. All DRD2 scores in the first row are predicted by the trained model from Olivecrona *et al.* (2017)

In Table 2, we can see that our method outperforms previous baselines like JTNN (Jin *et al.*, 2019) and HierG2G, and achieved state-of-the-art results on these tasks. We also provide some generated molecules in Figure 2.

Our method leverages a forward translation model f and a back translation model g . We summarize the time of training f and g (denoted as T_0), back translating the unlabeled molecules using g (denoted as T_b), and retraining the forward model (denoted as T_1) in

Table 3. Indeed, our method takes $2.5\times$ more time than conventional supervised method, but brings promising improvement. Note that $T_1 < T_0$ because retraining the forward model on the synthetic data is warm started from the previous forward model, and then it does not require much training time to convergence. We will further improve the efficiency in the future.

The diversity and novelty scores over QED and DRD2 datasets are reported in Table 4. The results show that our method achieves higher

Table 3. Statistics of the training time

Training time (h)	T_0	T_b	T_1
Transformer	8.5	0.9	1.6
HierG2G	16.8	1.1	2.8

Note: All training time is analyzed on DRD2 dataset along with 1M unlabeled data.

Table 4. Comparison of diversity and novelty scores between previous methods and our method on QED and DRD2 datasets

Methods	QED		DRD2	
	Diversity	Novelty (%)	Diversity	Novelty (%)
GCPN	0.216	100	0.152	100
MMPA	0.236	99.9	0.275	99.9
seq2seq	0.331	99.6	0.176	79.7
JTNN	0.373	98.3	0.156	83.4
Transformer backbone				
Baseline	0.349	98.8	0.182	89.8
+BT(250K)	0.377	99.3	0.218	92.4
+BT(250K, filtered)	0.380	99.3	0.221	92.7
+BT(1M)	0.396	99.7	0.242	95.1
+BT(1M, filtered)	0.390	99.5	0.238	94.9
HierG2G backbone				
Backbone	0.477	98.6	0.192	84.2
+BT(250K)	0.484	99.1	0.250	88.7
+BT(250K, filtered)	0.481	99.2	0.244	87.9
+BT(1M)	0.496	99.6	0.275	92.4
+BT(1M, filtered)	0.489	99.3	0.270	92.1

Note: For clarity, we have omitted some invalid previous results. Bold fonts indicate the best results of each setting.

diversity and novelty scores for both backbone models. This is because the large-scale unlabeled data enables the model to widely explore the hidden representations of molecules and generate grand new molecules. Our models still have less novelty than GCPN and MMPA; however, these methods have much lower property improvement. The results of LogP0.6 and LogP0.4 are left in [Supplementary Appendix A](#).

4 Results on retrosynthesis prediction

We also verify our approach on the retrosynthesis prediction task. A chemical reaction can be viewed as a transformation from a reactant set $\mathcal{R} = \{r_1, r_2, \dots, r_N\}$ to the product x . In retrosynthesis prediction, given the product x , the goal is to predict the reactant set \mathcal{R} .

4.1 Setup

Dataset: Following [Dai et al. \(2019\)](#), [Shi et al. \(2020\)](#) and [Yan et al. \(2020\)](#), we conduct experiments on the widely used benchmark dataset, USPTO-50k, which has 50k reactions with 10 reaction types in total. For fair comparison with previous work, we use the data released by [Dai et al. \(2019\)](#), where the training, validation and test have been split in advance and each part contains 80%, 10% and 10% of the total data respectively. (Data and code: <https://github.com/HanJun-Dai/GLN>.) For experiments, we work on two settings where the reaction type is given or not.

The unlabeled data are selected from ZINC ([Sterling and Irwin, 2015](#)). Note that for retrosynthesis prediction, each source molecule is made up of l target molecules. In USPTO-50k, $l \in \{1, 2, 3\}$. Let N_l denote the number of training samples whose target side has l molecules. According to statistics, $N_1 : N_2 : N_3 = 29.3\% : 70.4\% : 0.3\%$. Therefore, to create a reactant set \mathcal{R} in our unlabeled dataset \mathcal{U}_y , we sample the number of reactants l w.p. $\frac{N_l}{N_1+N_2+N_3}$, then sample l molecules from ZINC,

concatenate them as a large molecule and put it into \mathcal{U}_y . The default value of $|\mathcal{U}_y|$ is 250k, and we also vary this number to verify the effect on the unlabeled data size.

Network architecture: We apply our approach to a sequence-based model, Transformer ([Vaswani et al., 2017](#)) and a graph-based model, GLN ([Dai et al., 2019](#)). For Transformer, we first tokenize the SMILES strings using regular expression provided by [Schwaller et al. \(2018\)](#) and then feed them into Transformers (tokenizer details are reported in [Supplementary Appendix C](#)). Both the forward model f and reverse model g are Transformers with four layers, 8 attention heads, embedding dimension 256 and feed-forward dimension 2048. Therefore, we still use the Transformer as the reversed model when working on GLN. To sample pseudo pairs, i.e. [Equation \(5\)](#), we use random sampling to speed up the process.

Evaluation metrics: Following the common practice of retrosynthesis prediction ([Dai et al., 2019](#); [Shi et al., 2020](#)), we evaluate the models by the top- k exact match accuracy (briefly, top- k accuracy), which verifies that given a product, whether one of the k generated reactant sets exactly matches the ground truth reactant set. The k ranges from $\{1, 3, 5, 10, 20, 50\}$. For all $k > 1$, we use beam search to generate the reactant sets, and rank them by log likelihoods.

4.2 Results

The top- k accuracy of baselines and our approach is reported in [Table 5](#), including both the reaction type is given or not. The Δ in [Table 5](#) denotes the gap between ‘ours + alg’ and alg, where alg can be Transformer and GLN.

We have the following observations:

1. After combined with our method, the performance of both Transformer and GLN can be improved, no matter the reaction type is given or not. This demonstrates the effectiveness and generalization ability of our method.
2. Across different top- k accuracy metrics, our approach generally brings the most improvement for smaller k 's like 1, 3 and 5, no matter whether the reaction type is given or not. For example, when we use the GLN model with reaction type known, the improvement decreases w.r.t. k . This shows that the usage of unlabeled data helps improve the high-precision decisions.
3. Our proposed method outperforms existing baselines, including: (i) expertSys ([Liu et al., 2017](#)), an expert system that automatically extract retrosynthetic reaction rules from the training set and applies the rules to a target molecule to obtain the reactants; (ii) the sequence-to-sequence models, whose encoders/decoders are LSTM ([Liu et al., 2017](#)) and Transformer ([Karpov et al., 2019b](#)); (iii) the template-based methods retrosim ([Coley et al., 2017](#)) and neuralsym ([Segler and Waller, 2017](#)); (iv) the graph-to-graph translation model ([Shi et al., 2020](#)), which leverages graph neural network for translation. The success of our method shows the effectiveness of using unlabeled data.

4.3 Analysis

Effect of unlabeled data: To verify the effect on the amount of unlabeled data, we use all parallel data in USPTO-50k and use $\{50k, 100k, 150k, 250k, 500k\}$ unlabeled data. We conduct experiments on Transformer model and report the top-1 and top-10 accuracy. The results are in [Figure 3](#). As the number of unlabeled data increased from 50k to 250k, both the top-1 and top-10 accuracy increases, which shows that leveraging unlabeled data can effectively improve the performances. However, when the unlabeled data are larger than 250k, the performance is not further improved. Our conjecture is that the noise in the unlabeled data hurts more than the benefits brought by the unlabeled data.

Effect of labeled data: To verify the improvement of our method w.r.t. amounts of labeled data, we conduct experiments on

Table 5. Results of top- k exact match accuracy on USPTO-50K dataset

Methods	Top- k accuracy (%)					
	1	3	5	10	20	50
Reaction class given as prior						
expertSys (Liu <i>et al.</i> , 2017)	35.4	52.3	59.1	65.1	68.6	69.5
seq2seq (Liu <i>et al.</i> , 2017)	37.4	52.4	57.0	61.7	65.9	70.7
retrosim (Coley <i>et al.</i> , 2017)	52.9	73.8	81.2	88.1	91.8	92.9
neuralsym (Segler and Waller, 2017)	55.3	76.0	81.4	85.1	86.5	86.9
G2Gs	61.0	81.3	86.0	88.7	—	—
Transformer (Vaswani <i>et al.</i> , 2017)	52.2	68.2	72.7	77.4	80.1	82.3
Ours + Transformer	55.9	72.8	77.8	79.7	82.2	83.2
Δ	3.7	4.6	5.1	2.3	2.1	0.9
GLN (Dai <i>et al.</i> , 2019)	64.2	79.1	85.2	90.0	92.3	93.2
Ours + GLN	67.9	82.5	87.3	91.5	92.9	93.6
Δ	3.7	3.4	2.1	1.5	0.6	0.4
Reaction class not known						
retrosim (Coley <i>et al.</i> , 2017)	37.3	54.7	63.3	74.1	82.0	85.3
neuralsym (Segler and Waller, 2017)	44.4	65.3	72.4	78.9	82.2	83.1
G2Gs (Shi <i>et al.</i> , 2020)	48.9	67.6	72.5	75.5	—	—
Transformer (Vaswani <i>et al.</i> , 2017)	37.9	57.3	62.7	68.1	72.4	75.1
Ours + Transformer	43.5	58.8	64.6	69.7	73.6	75.8
Δ	5.6	1.5	1.9	1.6	1.2	0.7
GLN (Dai <i>et al.</i> , 2019)	52.5	69.0	75.6	83.7	89.0	92.4
Ours + GLN	54.7	70.2	77.0	84.4	89.5	92.7
Δ	2.2	1.2	1.4	0.7	0.5	0.3

Bold fonts indicate the best results of each setting.

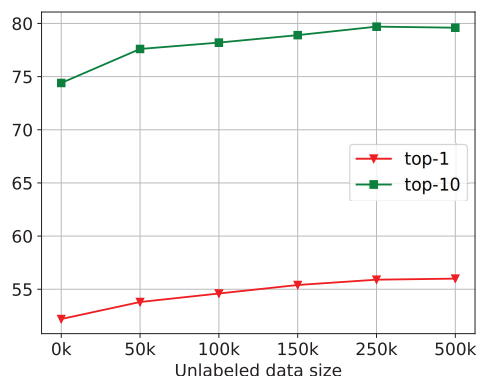


Fig. 3. Top-1 and top-10 accuracy w.r.t. different unlabeled data sizes

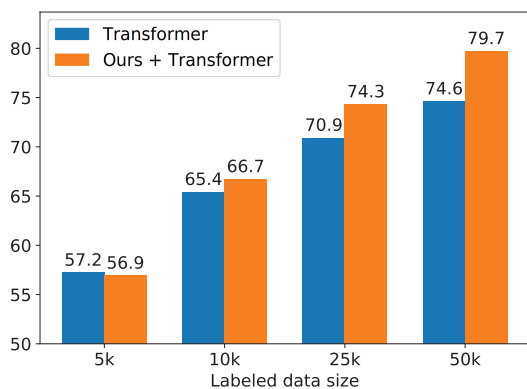


Fig. 4. Comparison of the top-10 accuracy between Transformer and the combination with our approach w.r.t. different amount of labeled data.

Transformer models with different settings, where there are L_p labeled samples $L_p \in \{5k, 10k, 25k, 50k\}$ and 250k unlabeled molecules. The results are in Figure 4. When only 5k data are available, leveraging monolingual data hurts the performance. This is because the Transformer model itself is not well trained, therefore the synthetic pseudo labels are not good enough to improve the performances. As the number of labeled data increased, we can see that leveraging more unlabeled data becomes more and more helpful. Our conjecture is that when the amount of labeled data is large enough, leveraging unlabeled data will become less helpful. We will verify it on larger datasets in the future.

Case study: We visualize the predicted results of GLN and our method built on top of GLN in Figure 5. We show the top-3 molecules and their Dice similarity (calculated by RDKit) to the ground truth. We can see that the reactants generated by our method are more close to the ground truth one, and our method generates the only correct candidate.

4.4 Examples of back translated molecules

We visualize some back translated examples of unlabeled data on molecular property improvement and retrosynthesis prediction in Figure 6. In Figure 6a and b, we can see that the backward models generate molecules \hat{x}_u whose property is not as good as y_u , thus (\hat{x}_u, y_u) can be used as additional training data for f . Similarly, in Figure 6c, the model can also generate synthetic data for retrosynthesis. We also show more back translation examples in Supplementary Appendix B.

5 Conclusions and future work

In this paper, we propose a new semisupervised method called back translation for molecule generation, where a reversed model is trained to back translate unlabeled data into the source domain, and the translated synthetic data were combined with the labeled data to enhance the model for the main task. Experiments on

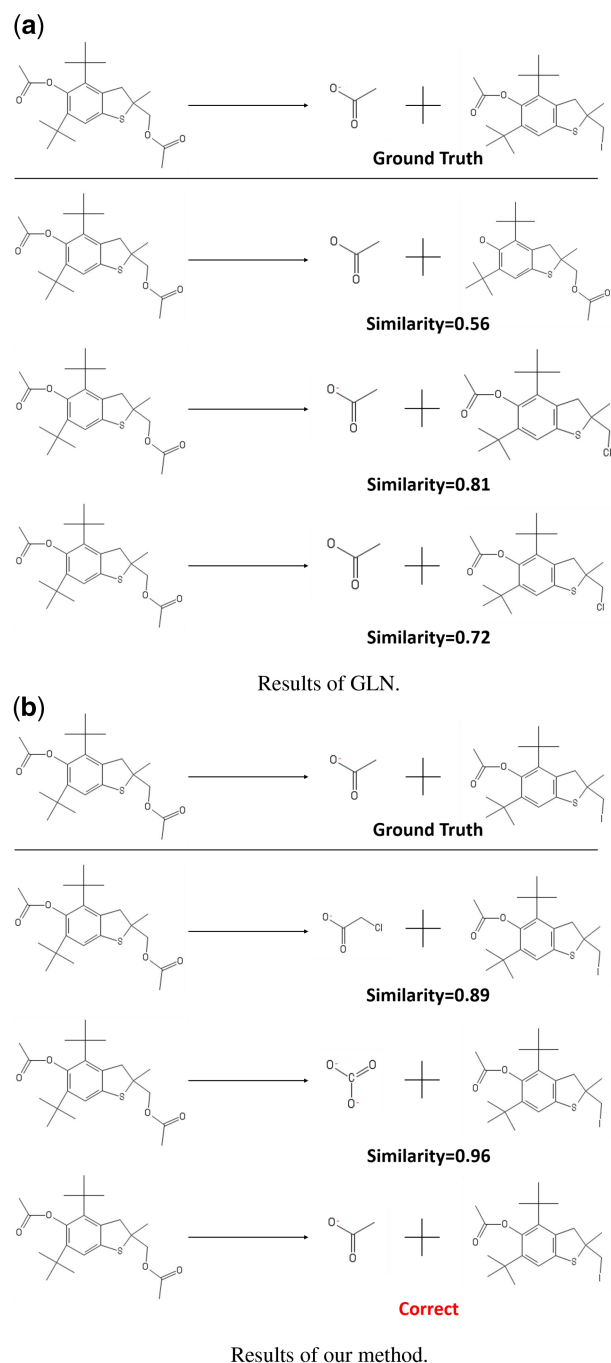
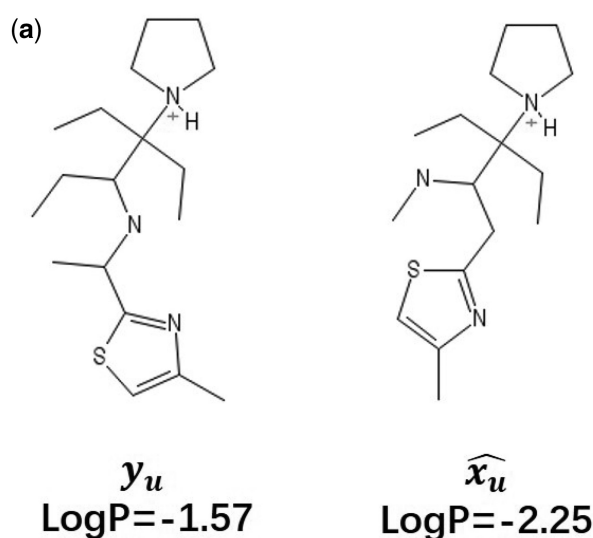


Fig. 5. Comparison of generated reactants between GLN and GLN + our approach.

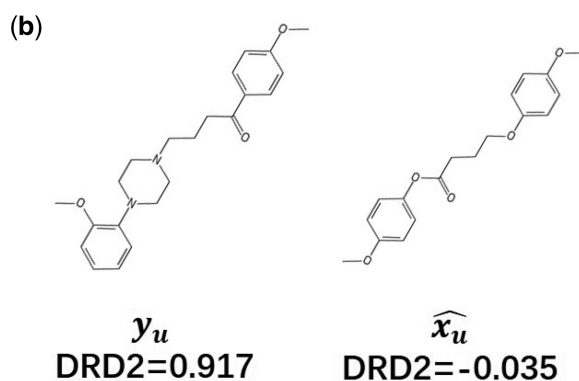
molecular property improvement and retrosynthesis prediction demonstrate the effectiveness of our approach, and rich ablation studies on the method were also conducted in our work. For future work, the first is to improve our algorithm to complete the source and target domains learning closed loop, as mentioned in He et al. (2016) and Xia et al. (2017). Second, we will try to apply more heuristic or machine-learning-based data filtration rules on generated synthetic data to further improve the performance of the algorithm.

Financial Support: none declared.

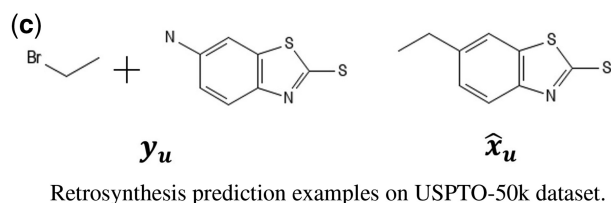
Conflict of Interest: none declared.



Molecular property improvement examples on LogP dataset.



Molecular property improvement examples on DRD2 dataset.



Retrosynthesis prediction examples on USPTO-50k dataset.

Fig. 6. Back translation examples. y_u are unlabeled data sampled randomly selected from ZINC, and \hat{x}_u are back translated outputs generated by the backward model g

References

- Bickerton, G.R. et al. (2012) Quantifying the chemical beauty of drugs. *Nat. Chem.*, 4, 90–98.
- Born, J. et al. (2021) Pacmannrl: *de novo* generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. *iScience*, 24, 102269.
- Chen, T. et al. (2020) A simple framework for contrastive learning of visual representations. In: H. D. III and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 119. PMLR, pp. 1597–1607. <http://proceedings.mlr.press/v119/chen20j/chen20j.pdf>
- Chithrananda, S. et al. (2020) Chemberta: large-scale self-supervised pretraining for molecular property prediction. https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_67.pdf

- Coley, C.W. *et al.* (2017) Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Sci.*, **3**, 1237–1245.
- Corey, E.J. (1991) The logic of chemical synthesis: multistep synthesis of complex carbogenic molecules (nobel lecture). *Angew. Chem. Int. Ed. Engl.*, **30**, 445–465.
- Dai, H. *et al.* (2019) Retrosynthesis Prediction with Conditional Graph Logic Network. In: Wallach, H. *et al.* (eds.), *Advances in Neural Information Processing Systems, Vancouver, Canada*, Vol. 32. Curran Associates, Inc., Red Hook, NY.
- Dalke, A. *et al.* (2018) mmpdb: an open-source matched molecular pair platform for large multiproperty data sets. *J. Chem. Inf. Model.*, **58**, 902–910.
- De Cao, N. and Kipf, T. (2018) Molgan: an implicit generative model for small molecular graphs. In: *ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models*.
- Dempster, A.P. *et al.* (1977) Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. Ser. B*, **39**, 1–22.
- Devlin, J. *et al.* (2019) 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)*. Association for Computational Linguistics, Minneapolis, MN, pp. 4171–4186.
- Eduov, S. *et al.* (2018) Understanding back-translation at scale. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pp. 489–500.
- Gómez-Bombarelli, R. *et al.* (2018) Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Sci.*, **4**, 268–276.
- Grechishnikova, D. (2021) Transformer neural network for protein-specific *de novo* drug generation as a machine translation problem. *Sci. Rep.*, **11**, 321.
- He, D. *et al.* (2016) Dual learning for machine translation. In: Lee, D. *et al.* (eds.), *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc., Red Hook, NY, pp. 820–828.
- Imrie, F. *et al.* (2021) Generating property-matched decoy molecules using deep learning. *Bioinformatics*, **37**, 2134–2141.
- Jin, W. *et al.* (2018) Junction tree variational autoencoder for molecular graph generation. In: Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 80. PMLR, Stockholm, Sweden, pp. 2323–2332.
- Jin, W. *et al.* (2019) Learning Multimodal Graph-to-Graph Translation for Molecule Optimization. In: *International Conference on Learning Representations, New Orleans, LA, USA*.
- Jin, W. *et al.* (2020) Hierarchical generation of molecular graphs using structural motifs. In: *International Conference on Machine Learning*. PMLR, pp. 4839–4848. <http://proceedings.mlr.press/v119/jin20a/jin20a.pdf>
- Kang, S. and Cho, K. (2019) Conditional molecular design with deep generative models. *J. Chem. Inf. Model.*, **59**, 43–52.
- Karpov, P. *et al.* (2019a) A transformer model for retrosynthesis. In: Tetko, I.V. *et al.* (eds.), *Artificial Neural Networks and Machine Learning—ICANN 2019: Workshop and Special Sessions*. Springer International Publishing, Cham, pp. 817–830.
- Karpov, P. *et al.* (2019b) A transformer model for retrosynthesis. In: *International Conference on Artificial Neural Networks*, pp. 817–830. Springer, Cham.
- Kotsias, P.-C. *et al.* (2020) Direct steering of *de novo* molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.*, **2**, 254–265.
- Krizhevsky, A. *et al.* (2012) Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Proc. Syst.*, **25**, 1097–1105.
- Kusner, M.J. *et al.* (2017) Grammar variational autoencoder. In *International Conference on Machine Learning*. PMLR, Sydney, NSW, Australia, pp. 1945–1954.
- Landrum, G. (2016) Rdkit: open-source cheminformatics software. <https://www.bibsonomy.org/bibtex/28d01fceecc66bf2486e47d7c4207b108/salot>
- Li, P. *et al.* (2020) Learn molecular representations from large-scale unlabeled molecules for drug discovery. arXiv preprint arXiv:2012.11175.
- Liu, B. *et al.* (2017) Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Sci.*, **3**, 1103–1113.
- Liu, Q. *et al.* (2018) Constrained graph variational autoencoders for molecule design. *Adv. Neural Inf. Process. Syst.*, **31**, 7795–7804.
- Liu, Y. *et al.* (2019) Roberta: a robustly optimized BERT pretraining approach. arXiv:1907.11692.
- Olivecrona, M. *et al.* (2017) Molecular de-novo design through deep reinforcement learning. *J. Cheminf.*, **9**, 1–14.
- Popova, M. *et al.* (2018) Deep reinforcement learning for *de novo* drug design. *Sci. Adv.*, **4**, eaap7885.
- Rogers, D. and Hahn, M. (2010) Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, **50**, 742–754.
- Schwaller, P. *et al.* (2018) “found in translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.*, **9**, 6091–6098.
- Segler, M.H. and Waller, M.P. (2017) Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–Eur. J.*, **23**, 5966–5971.
- Sennrich, R. *et al.* (2016) Improving neural machine translation models with monolingual data. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pp. 86–96.
- Shi, C. *et al.* (2020) A graph to graphs framework for retrosynthesis prediction. In: *International Conference on Machine Learning*. PMLR, pp. 8818–8827. <http://proceedings.mlr.press/v119/shi20d/shi20d.pdf>
- Sterling, T. and Irwin, J.J. (2015) Zinc 15—ligand discovery for everyone. *J. Chem. Inf. Model.*, **55**, 2324–2337.
- Stokes, J.M. *et al.* (2020) A deep learning approach to antibiotic discovery. *Cell*, **180**, 688–702.e13.
- Vaswani, A. *et al.* (2017) Attention is all you need. In: Guyon, I. *et al.* (eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., Red Hook, NY, pp. 5998–6008.
- Wang, Y. *et al.* (2019) Multi-agent dual learning. In: *Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA*. <https://openreview.net/forum?id=HyGhN2A5tm>.
- Xia, Y. *et al.* (2017) Dual supervised learning. In: *International Conference on Machine Learning*. PMLR, Sydney, NSW, Australia, pp. 3789–3798.
- Xie, Y. *et al.* (2021) MARS: Markov molecular sampling for multi-objective drug discovery. In: *International Conference on Learning Representations (ICLR)*.
- Yan, C. *et al.* (2020) Retroxpert: decompose retrosynthesis prediction like a chemist. In: Larochelle, H. *et al.* (eds.), *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., Red Hook, NY, pp. 11248–11258.
- You, J. *et al.* (2018) Graph convolutional policy network for goal-directed molecular graph generation. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates, Inc., Red Hook, NY, pp. 6412–6422.
- Zheng, S. *et al.* (2020) Predicting retrosynthetic reactions using self-corrected transformer neural networks. *J. Chem. Inf. Model.*, **60**, 47–55.