RetCL: A Selection-based Approach for Retrosynthesis via Contrastive Learning

Hankook Lee[†] **Sungsoo Ahn**[†] **Seung-Woo Seo**[§][*] **You Young Song**[‡]
Eunho Yang[†] **Sung Ju Hwang**[†] **Jinwoo Shin**[†]
[†]Korea Advanced Institute of Science and Technology (KAIST)
[§]Standigm [‡]Samsung Advanced Institute of Technology (SAIT)

Abstract

Retrosynthesis, of which the goal is to find a set of reactants for synthesizing a target product, is an emerging research area of deep learning. While the existing approaches have shown promising results, they currently lack the ability to consider commercial availability of the reactants or generalize to unseen reaction templates. In this paper, we propose a new approach that mitigates the issues by reformulating retrosynthesis into a selection problem of reactants from a candidate set of commercially available molecules. To this end, we design an efficient reactant selection framework, named RetCL (retrosynthesis via contrastive learning), for enumerating all of the candidate molecules based on selection scores computed by graph neural networks. For learning the score functions, we propose a novel contrastive training scheme with hard negative mining. Extensive experiments demonstrate the benefits of the proposed selection-based approach over the prior works. For example, when all 671k reactants in the USPTO database are given as candidates, our RetCL achieves top-1 exact match accuracy of 71.3% for the USPTO-50k benchmark, while a recent transformer-based approach achieves 59.6%.

1 Introduction

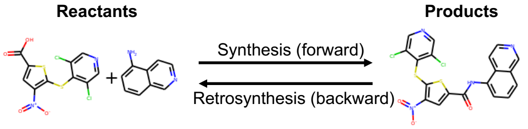


Figure 1: Chemical reaction.

Retrosynthesis [5], finding a synthetic route starting from commercially available reactants to synthesize a target product (see Figure 1), is at the center of focus for discovering new materials in both academia and industry. To address this, researchers have developed computer-aided frameworks to automate the process of retrosynthesis for more than three decades [6]. The recent ML-based approaches mainly fall into two categories depending on their reliance on the reaction templates, i.e., sub-graph patterns describing how the chemical reaction occurs among reactants. Although the template-based approaches [4, 8, 22] can provide chemically interpretable predictions, they limit the search space to known templates and cannot discover novel synthetic routes. In contrast, template-free approaches [13, 16, 23, 28] generate the reactants from scratch to avoid relying on the reaction templates. However, they require to search the entire molecular space, and their predictions could be either unstable or commercially unavailable.

We emphasize that retrosynthesis methods are often required to consider the availability of reactants and generalize to unseen templates in real-world scenarios. For example, when a predicted reactant is not available (e.g., not purchasable) for a chemist or a laboratory, the synthetic path starting from the predicted reactant cannot be instantly used in practice. Moreover, chemists often require

[*] Work done at Samsung Advanced Institute of Technology.

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

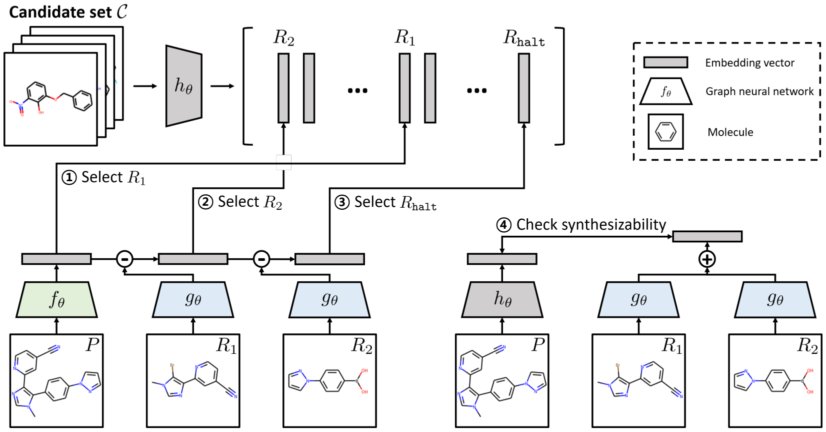


Figure 2: Illustration of the search procedure in RETCL.

retrosynthetic analysis based on unknown reaction rules. This is especially significant due to our incomplete knowledge of chemical reactions; e.g., 29 million reactions were regularly recorded between 2009 and 2019 in a chemical database, Reaxys [18].

Contribution. In this paper, we propose a new *selection-based* framework, named RETCL (retrosynthesis via contrastive learning), which allows considering the commercial availability of reactants. To this end, we reformulate the task of retrosynthesis as a problem where reactants are selected from a candidate set of available molecules. To address the problem, we design selection scores defined by the similarity between GNN-based molecular embeddings of the product and the reactants. We also propose a novel contrastive learning scheme [3, 12, 24] with hard negative mining [11] to overcome a scalability issue while handling a large-scale candidate set. We remark that this selection-based approach has two benefits over the existing ones: (a) it guarantees the commercial availability of the selected reactants; (b) it can generalize to unseen reaction templates and find novel synthetic routes.

To demonstrate the effectiveness of our RETCL, we conduct various experiments based on the USPTO database [17]. Thanks to our prior knowledge on the candidate reactants, our method achieves 71.3% test accuracy and significantly outperforms the baselines without such prior knowledge. Furthermore, our algorithm demonstrates its superiority even when enhancing the baselines with candidate reactants, e.g., our algorithm improves upon the existing template-free approach [1] by 11.7%. Moreover, we evaluate generalization ability of RETCL to unseen templates and demonstrate how our RETCL can be also applied to the multi-step retrosynthesis setting.

2 Selection-based retrosynthesis via contrastive learning

A chemical reaction $\mathcal{R} \to P$ is a synthetic process of converting a *reactant-set* $\mathcal{R} = \{R_1, \ldots, R_n\}$, i.e., a set of *reactant* molecules, to a *product* molecule P (see Figure 1). We aim to solve the problem of retrosynthesis by finding the reactant-set $\mathcal{R}$ from a *candidate set* $\mathcal{C}$ which can be synthesized to the target product P. Especially, we consider the case when the candidate set $\mathcal{C}$ consists of *commercially available* molecules. Throughout this paper, we say that the synthetic direction (from $\mathcal{R}$ to P) is *forward* and the retrosynthetic direction (from P to $\mathcal{R}$) is *backward*.

To find a reactant-set $\mathcal{R} = \{R_1, \ldots, R_n\}$, we select each element R_i sequentially from the candidate set $\mathcal{C}$ based on the *backward* (retrosynthetic) selection score $\psi(R|P, \mathcal{R}_{\texttt{given}})$. Note that the score function is also capable of selecting a special reactant $R_{\texttt{halt}}$ to stop updating the reactant-set. Using beam search, we choose top T likely reactant-sets $\mathcal{R}_1, \ldots, \mathcal{R}_T$. Next, we rank the chosen reactant-sets $\mathcal{R}_1, \ldots, \mathcal{R}_T$ based on the backward selection score $\psi(R|P, \mathcal{R}_{\texttt{given}})$ and the *forward* (synthetic) score $\phi(P|\mathcal{R})$. Note that $\psi(R|P, \mathcal{R}_{\texttt{given}})$ and $\phi(P|\mathcal{R})$ correspond to backward and forward directions of a chemical reaction $\mathcal{R} \to P$, respectively (see Figure 1). Using both score functions, we define an overall score on a chemical reaction $\mathcal{R} \to P$ as follows:

$$\texttt{score}(P, \mathcal{R}) = \frac{1}{n+2}\left(\max_{\pi \in \Pi} \sum\nolimits_{i=1}^{n+1} \psi(R_{\pi(i)}|P, \{R_{\pi(1)}, \ldots, R_{\pi(i-1)}\}) + \phi(P|\mathcal{R})\right), \qquad (1)$$

where $R_{n+1} = R_{\texttt{halt}}$ and Π is the space of permutations defined on the integers $1, \ldots, n{+}1$ satisfying $\pi(n{+}1) = n{+}1$. Figure 2 illustrates this search procedure of our framework.

Score design. One can observe that the molecular graph of the product P can be decomposed into subgraphs from each reactant of the reactant-set $\mathcal{R}$. Moreover, when selecting reactants sequentially, the structural information of the previously selected reactants $\mathcal{R}_{\texttt{given}}$ should be ignored to avoid duplicated selections. From these observations, we design the scores ψ and ϕ as follows:

$$\psi(R|P,\mathcal{R}_{\texttt{given}}) = \texttt{CosSim}\left(f_\theta(P) - \sum\nolimits_{S\in\mathcal{R}_{\texttt{given}}} g_\theta(S),\ h_\theta(R)\right),$$
$$\phi(P|\mathcal{R}) = \texttt{CosSim}\left(\sum\nolimits_{R\in\mathcal{R}} g_\theta(R),\ h_\theta(P)\right),$$

where $\texttt{CosSim}$ is the cosine similarity and f_θ, g_θ, h_θ are GNN-based embedding functions from a molecule to a fixed-sized vector with parameters θ. This design allows the search procedure to be processed as an efficient matrix-vector multiplication on the large candidate set $\mathcal{C}$. We also provide a technique for incorporating additional supervision such as reaction types (e.g., carbon-carbon bond formation) in Appendix A. Other implementation details are described in Appendix B.

Training scheme. For learning the score functions ψ and ϕ, we consider two classification tasks with the following probabilities:

$$p(R|P,\mathcal{R}_{\texttt{given}},\mathcal{C}) = \frac{\exp(\psi(R|P,\mathcal{R}_{\texttt{given}})/\tau)}{\sum_{R'\in\mathcal{C}\setminus\{P\}}\exp(\psi(R'|P,\mathcal{R}_{\texttt{given}})/\tau)},$$
$$q(P|\mathcal{R},\mathcal{C}) = \frac{\exp(\phi(P|\mathcal{R})/\tau)}{\sum_{P'\in\mathcal{C}\setminus\mathcal{R}}\exp(\phi(P'|\mathcal{R})/\tau)},$$

where τ is a hyperparameter for temperature scaling and $\mathcal{C}$ is the given candidate set of molecules. Now we define the following losses for a reaction $\mathcal{R} \to P$:

$$\mathcal{L}_{\texttt{backward}}(P,\mathcal{R}|\theta,\mathcal{C}) = -\max_{\pi\in\Pi}\sum\nolimits_{i=1}^{n+1}\log p(R_{\pi(i)}|P,\{R_{\pi(1)},\ldots,R_{\pi(i-1)}\},\mathcal{C}),$$
$$\mathcal{L}_{\texttt{forward}}(P,\mathcal{R}|\theta,\mathcal{C}) = -\log q(P|\mathcal{R},\mathcal{C}),$$

where $R_{n+1} = R_{\texttt{halt}}$ and Π is the space of permutations defined on the integers $1,\ldots,n{+}1$ satisfying $\pi(n{+}1) = n{+}1$. We note that such an objective is known as contrastive loss which has recently gained much attention in various domains [3, 12, 19, 24, 25]. Since the optimization of the above losses is intractable due to the large number of candidate molecules, we approximate $\mathcal{C}$ with the set $\mathcal{C}_\mathcal{B}$ of molecules in a mini-batch $\mathcal{B}$ of reactions where $\mathcal{C}_\mathcal{B} = \bigcup_{(\mathcal{R},P)\in\mathcal{B}}\mathcal{R}\cup\{P\}$. Then we arrive at the following training objective:

$$\mathcal{L}(\mathcal{B}|\theta) = \frac{1}{|\mathcal{B}|}\sum\nolimits_{(\mathcal{R},P)\in\mathcal{B}}\mathcal{L}_{\texttt{backward}}(P,\mathcal{R}|\theta,\mathcal{C}_\mathcal{B}) + \mathcal{L}_{\texttt{forward}}(P,\mathcal{R}|\theta,\mathcal{C}_\mathcal{B}). \tag{2}$$

Hard negative mining. In our setting, molecules in the candidate set $\mathcal{C}_\mathcal{B}$ are easily distinguishable. Hence, learning to discriminate between them is often not informative. To alleviate this issue, we replace the $\mathcal{C}_\mathcal{B}$ with its augmented version $\widetilde{\mathcal{C}}_\mathcal{B}$ by adding *hard* negative (i.e., similar) samples where $\widetilde{\mathcal{C}}_\mathcal{B} = \mathcal{C}_\mathcal{B}\cup\bigcup_{M\in\mathcal{C}_\mathcal{B}}\{\text{Top-}K \text{ nearest neighbors of } M \text{ from } \mathcal{C}\}$. The nearest neighbors are defined with respect to the cosine similarity on $\{h_\theta(M)\}_{M\in\mathcal{C}}$. We found that the hard negative mining plays a significant role in improving the performance of RETCL as shown in Appendix C.

3 Experiments

Table 1 evaluates our RETCL and other baselines using the top-k exact match accuracy in USPTO-50k. Our framework significantly outperforms a concurrent selection-based (SB) approach, Bayesian-Retro [10], by 23.8% in terms of top-1 accuracy when reaction type is unknown. Furthermore, ours also outperforms template-based (TB) approaches utilizing the different knowledge, i.e., reaction templates instead of candidates, with a large margin, e.g., 18.8% over GLN [8]. The full results and failure case study are provided in Appendix D and E, respectively.

Incorporating the knowledge of candidates into baselines. It is hard to fairly compare between methods operating under different assumptions. For example, template-based approaches require the knowledge of reaction templates, while our selection-based approach requires that of available reactants. To alleviate such a concern, we improve the baselines to additionally use our prior

Table 1: Top-k exact match accuracy (%) of computer-aided approaches in USPTO-50k.

Category	Method	Top-1	Top-5	Top-10
TF	Transformer [13]	37.9	62.7	-
	SCROP [28]	43.7	65.2	68.7
	Transformer [1]	44.8	67.7	71.1
	G2Gs [23]	**48.9**	**72.5**	**75.5**
TB	retrosim [4]	37.3	63.3	74.1
	neuralsym [22]	44.4	72.4	78.9
	GLN [8]	**52.5**	**75.6**	**83.7**
SB	Bayesian-Retro [10]	47.5	77.0	80.3
	RetCL (Ours)	**71.3**	**92.0**	**94.1**

Table 2: Top-k exact match accuracy (%) of computer-aided approaches with discarding predictions not in the candidate set $\mathcal{C}$.

Method	Top-1	Top-10	Top-100
Transformer [1]	59.6	77.0	79.5
RetCL (Ours)	71.3	**94.1**	**96.7**
GLN [8]	**77.3**	92.5	93.3

Table 3: Generalization to USPTO-full.

Method	Top-1	Top-10	Top-50
GLN [8]	26.7	42.2	46.7
RetCL (Ours)	39.9	57.1	60.9

Table 4: Top-10 exact match accuracy (%) of our RetCL and GLN [8] trained on USPTO-50k without reaction types from 6 to 10.

Method	Average	1	2	3	4	5	6	7	8	9	10
GLN [8]	39.7	84.3	92.2	70.7	59.3	89.7	0.0	0.0	0.0	0.5	0.0
RetCL (Ours)	**55.6**	**93.9**	**97.6**	**86.4**	**67.0**	**95.6**	**59.1**	**11.9**	**18.3**	**26.1**	0.0

knowledge of candidates $\mathcal{C}$; we filter out reactants outside the candidates $\mathcal{C}$ from the predictions made by the baselines. As reported in Table 2, our framework still outperforms the template-free (TF) approach with a large margin. Moreover, the performance of the template-based approach, GLN [8], is saturated to 93.3% due to the limited coverage of reaction templates while ours continues to increase the top-k accuracy as k increases.

Generalization to unseen templates. The advantage of our framework over the template-based approaches is the generalization ability to unseen reaction templates. To demonstrate it, we remove reactions of classes (i.e., reaction types) from 6 to 10 in training/validation splits of USPTO-50k. In this case, the templates extracted from the modified dataset cannot be applied to the reactions of different classes. Hence the template-based approaches suffer from the generalization issue; for example, GLN [8] cannot provide correct predictions for reactions of unseen types as reported in Table 4, while our RetCL is able to provide correct answers.

We also conduct a more realistic experiment: testing on a larger dataset, the test split of USPTO-full provided by [8], using a model trained on a smaller dataset, USPTO-50k. As reported in Table 3, our framework provides a consistent benefit over the template-based approaches. These results show that our strength of generalization ability. More details of datasets are provided in Appendix F.

Table 5: Multi-step retrosynthesis.

	Single		Hybrid	
Single-step model	MLP	TF	TF+TF	RetCL+TF
Succ. rate (%)	86.84	91.05	90.54	96.84
Avg. length	-	4.30	4.31	3.90

Multi-step retrosynthesis. To consider a more practical scenario, we evaluate our algorithm for the task of *multi-step retrosynthesis*. To this end, we use the synthetic route benchmark provided by [2]. Especially, we focus on demonstrating how our method could be used to improve the existing template-free Transformer model (TF, [1]). Given a target product, the hybrid algorithm operates as follows: (1) our RetCL proposes a set of reactants from the candidates $\mathcal{C}$; (2) TF proposes additional reactants outside the candidates $\mathcal{C}$; (3) TF chooses the top-K reactants based on its log-likelihood of all the proposed reactants. As an additional baseline, we replace RetCL by another independently trained TF in the hybrid algorithm. We use Retro* [2] for efficient route search with the retrosynthesis models and evaluate the discovered routes based on the metrics used by [2, 14]. As reported in Table 5, our model can enhance the search quality of the existing template-free model in the multi-step retrosynthesis scenarios. The detailed description of this multi-step retrosynthesis experiment and the discovered routes are provided in Appendix G.

4 Conclusion

In this paper, we propose RetCL for solving retrosynthesis. To this end, we reformulate retrosynthesis as a selection problem of commercially available reactants, and propose a contrastive learning scheme with hard negative mining to train our RetCL. Through the extensive experiments, we show that our framework achieves outstanding performance for the USPTO benchmarks. We believe that extending our framework to multi-step retrosynthesis or combining with various contrastive learning techniques in other domains could be interesting future research directions.

References

[1] Benson Chen, Tianxiao Shen, Tommi S Jaakkola, and Regina Barzilay. Learning to make generalizable and diverse predictions for retrosynthesis. *arXiv preprint arXiv:1910.09688*, 2019.

[2] Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided a* search. In *ICML*, 2020.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[4] Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS central science*, 3(12):1237–1245, 2017.

[5] Elias James Corey. The logic of chemical synthesis: multistep synthesis of complex carbogenic molecules (nobel lecture). *Angewandte Chemie International Edition in English*, 30(5):455–465, 1991.

[6] Elias James Corey, Alan K Long, and Steward D Rubenstein. Computer-assisted analysis in organic synthesis. *Science*, 228(4698):408–418, 1985.

[7] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[8] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8872–8882. Curran Associates, Inc., 2019.

[9] Zsombor Gonda and Zoltan Novak. Transition-metal-free n-arylation of pyrazoles with diaryliodonium salts. *Chemistry (Weinheim an der Bergstrasse, Germany)*, 21(47):16801–16806, 2015.

[10] Zhongliang Guo, Stephen Wu, Mitsuru Ohno, and Ryo Yoshida. A bayesian algorithm for retrosynthesis. *arXiv preprint arXiv:2003.03190*, 2020.

[11] Ben Harwood, BG Kumar, Gustavo Carneiro, Ian Reid, Tom Drummond, et al. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2821–2829, 2017.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[13] Pavel Karpov, Guillaume Godin, and Igor V Tetko. A transformer model for retrosynthesis. In *International Conference on Artificial Neural Networks*, pages 817–830. Springer, 2019.

[14] Akihiro Kishimoto, Beat Buesser, Bei Chen, and Adi Botea. Depth-first proof-number search with heuristic edge cost and application to chemical synthesis planning. In *Advances in Neural Information Processing Systems*, pages 7226–7236, 2019.

[15] Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. Automatic retrosynthetic pathway planning using template-free models. *arXiv preprint arXiv:1906.02308*, 2019.

[16] Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS central science*, 3(10):1103–1113, 2017.

[17] Daniel Mark Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 2012.

[18] Troy Mutton and Damon D Ridley. Understanding similarities and differences between two prominent web-based chemical information and data retrieval tools: Comments on searches for research topics, substances, and reactions. *Journal of Chemical Education*, 96(10):2167–2179, 2019.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Autodiff Workshop*, 2017.

[21] Nadine Schneider, Nikolaus Stiefl, and Gregory A Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of chemical information and modeling*, 56(12):2336–2346, 2016.

[22] Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

[23] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. *arXiv preprint arXiv:2003.12725*, 2020.

[24] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

[25] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[26] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. *arXiv preprint arXiv:1909.01315*, 2019.

[27] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[28] Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling*, 2019.

A Incorporating reaction types

A human expert could have some prior information about a reaction type, e.g., carbon-carbon bond formation, for the target product P. To utilize this prior knowledge, we add trainable embedding bias vectors $u^{(t)}$ and $v^{(t)}$ for each reaction type t into the query embeddings of ψ and ϕ, respectively. To be specific,

$$\psi(R|P,\mathcal{R}_{\texttt{given}},t) = \texttt{CosSim}\left(f_\theta(P) - \sum\nolimits_{S\in\mathcal{R}_{\texttt{given}}} g_\theta(S) + u^{(t)},\ h_\theta(R)\right),$$
$$\phi(P|\mathcal{R},t) = \texttt{CosSim}\left(\sum\nolimits_{R\in\mathcal{R}} g_\theta(R) + v^{(t)},\ h_\theta(P)\right),$$

where bias vectors u and v are initialized by zero at beginning of training. We show that this additional knowledge provides a significant gain as reported in Section D.

B Implementation details

We here provide a detailed description of our implementation. Since the USPTO datasets provide molecule information based on the SMILES [27] format, we convert a SMILES representation to a bidirectional graph with atom and bond features. To this end, we use RDKit[2] and Deep Graph Library (DGL) [26]. Let $G = (V, E)$ be the molecular graph, and $X(v) \in \mathbb{R}^{d_{\texttt{atom}}}$ and $X(uv) \in \mathbb{R}^{d_{\texttt{bond}}}$ are features for an atom $v \in V$ and a bond $uv \in E$ in the molecular graph G, respectively. The atom feature $X(v)$ includes the atom type (e.g., C, I, B), degree, formal charge, and so on; the bond feature $X(uv)$ includes the bond type (single, double, triple or aromatic), whether the bond is in a ring, and so on. For more details, we highly recommend to see DGL and its extension, DGL-LifeSci.[3]

Architecture. We build our graph neural network (GNN) architecture based on the molecular graph G with features X as follows:

$$\begin{aligned}
H^{(0)}(v) &\leftarrow \texttt{ReLU}\left(\texttt{BN}\left(W^{(0)}_{\texttt{atom}} X(v) + \sum\nolimits_{u \in \mathcal{N}(v)} W^{(0)}_{\texttt{bond}} X(uv)\right)\right), \\
H^{(l)}(v) &\leftarrow \texttt{ReLU}\left(\texttt{BN}\left(W^{(l)}_{1} \sum\nolimits_{u \in \mathcal{N}(v)} H^{(l-1)}(u) + \sum\nolimits_{u \in \mathcal{N}(v)} W^{(l)}_{\texttt{bond}} X(uv)\right)\right), \\
H^{(l)}(v) &\leftarrow \texttt{ReLU}\left(\texttt{BN}\left(W^{(l)}_{2} H^{(l)}(v) + H^{(l-1)}(v)\right)\right), \text{ for } l = 1, 2, \ldots, L, \\
H(v) &\leftarrow W_{\texttt{last}} H^{(L)}(v),
\end{aligned}$$

where $\mathcal{N}(v)$ is the set of adjacent vertices with v. This architecture is based on structure2vec [8, 7], however it is slightly different with the model used by [8]: we use `ReLU` after `BN` instead of `BN` after `ReLU`; we append a last linear model $W_{\texttt{last}}$. Based on the atom-level embeddings $H(v)$, we construct query and key embeddings f_θ, g_θ, and h_θ using three separate residual blocks as follows:

$$\begin{aligned}
f_\theta(M) &\leftarrow \sum_{v \in V} \left(H(v) + \texttt{BN}(W^{(f)}_2 \texttt{ReLU}(\texttt{BN}(W^{(f)}_1 \texttt{ReLU}(H(v)))))\right), \\
g_\theta(M) &\leftarrow \sum_{v \in V} \left(H(v) + \texttt{BN}(W^{(g)}_2 \texttt{ReLU}(\texttt{BN}(W^{(g)}_1 \texttt{ReLU}(H(v)))))\right), \\
h_\theta(M) &\leftarrow \sum_{v \in V} \left(H(v) + \texttt{BN}(W^{(h)}_2 \texttt{ReLU}(\texttt{BN}(W^{(h)}_1 \texttt{ReLU}(H(v)))))\right),
\end{aligned}$$

where M is the corresponding molecule with the molecular graph G. Note that θ includes all W defined above, and we omit bias vectors of the linear layers due to the notational simplicity. We found that these design choices, e.g., sharing GNN layers and using residual layers, also provide an accuracy gain. Therefore, more sophisticated architecture designs could provide further improvements; we leave it for future work.

Architecture. To parameterize the graph neural networks f_θ, g_θ and h_θ, we use a single shared 5-layer structure2vec [7, 8] and three separate 2-layer residual blocks with an embedding size of 256. We first apply the residual blocks on vertex-level embedding vectors obtained from the sturecture2vec, and then aggregate them via `sum` pooling to obtain the graph-level embedding.

Optimization. For learning the parameter θ, we use the stochastic gradient descent (SGD) with a learning rate of 0.01, a momentum of 0.9, a weight decay of 10^{-5}, a batch size of 64, and a gradient clip of 5.0. We train our model for $200k$ iterations and evaluate on the validation split every 1000 iterations. The information of the nearest neighbors is also updated every 1000 iterations and we choose $K = 4$ hard-negative samples during training. When evaluating on the test split, we use the best validation model with a beam size of 200.

To sum up, we use Pytorch [20] for automatic differentiation, Deep Graph Library [26] for building graph neural networks, and RDKit for processing SMILES [27] representations. All our models can be executed on single NVIDIA RTX 2080 Ti GPU.

[2] Open-Source Cheminformatics Software, https://www.rdkit.org/

[3] Bringing Graph Neural Networks to Chemistry and Biology, https://lifesci.dgl.ai/

C Ablation study

Table 6: Ablation study.

$\phi(P\|\mathcal{R})$	K	`sum`	Top-1	Top-10
✓			59.5	79.8
✓	1		69.6	92.2
✓	2		70.9	92.7
✓	4		71.1	92.9
	4		69.8	90.3
✓	4	✓	71.3	94.1

Table 6 shows the effect of components of our framework. First, we found that the hard negative mining as described increases the performance significantly. This is because there are many similar molecules in the candidate set $\mathcal{C}$, thus a model could predict slightly different reactants without hard negative mining. We also demonstrate the effect of checking the synthesizablity of the predicted reactants with $\phi(P|\mathcal{R})$. As seen the fourth and fifth rows in Table 6, using $\phi(P|\mathcal{R})$ provides a 2.6% gain in terms of top-10 accuracy. Moreover, we empirically found that `sum` pooling for aggregating node embedding vectors is more effective than `mean` pooling. This is because the former can capture the size of molecules as the norm of graph embedding vectors.

D Full results

Table 7: The top-k exact match accuracy (%) of computer-aided approaches in USPTO-50k. The template-based approaches use the knowledge of reaction templates while others do not. †The results are reproduced using the code of [1].

Category	Method	Top-1	Top-3	Top-5	Top-10	Top-20	Top-50
Reaction type is unknown							
Template-free	Transformer [13]	37.9	57.3	62.7	-	-	-
	SCROP [28]	43.7	60.0	65.2	68.7	-	-
	Transformer [1]	44.8	62.6	67.7	71.1	-	-
	G2Gs [23]	**48.9**	**67.6**	**72.5**	**75.5**	-	-
Template-based	retrosim [4]	37.3	54.7	63.3	74.1	82.0	85.3
	neuralsym [22]	44.4	65.3	72.4	78.9	82.2	83.1
	GLN [8]	**52.5**	**69.0**	**75.6**	**83.7**	**89.0**	**92.4**
Selection-based	Bayesian-Retro [10]	47.5	67.2	77.0	80.3	-	-
	RetCL (Ours)	**71.3**	**86.4**	**92.0**	**94.1**	**95.0**	**96.4**
Reaction type is given as prior							
Template-free	seq2seq [16]	37.4	52.4	57.0	61.7	65.9	70.7
	Transformer† [1]	54.1	70.0	74.2	77.8	80.4	83.3
	SCROP [28]	59.0	74.8	78.1	81.1	-	-
	G2Gs [23]	**61.0**	**81.3**	**86.0**	**88.7**	-	-
Template-based	retrosim [4]	52.9	73.8	81.2	88.1	91.8	92.9
	neuralsym [22]	55.3	76.0	81.4	85.1	86.5	86.9
	GLN [8]	**64.2**	**79.1**	**85.2**	**90.0**	**92.3**	**93.2**
Selection-based	Bayesian-Retro [10]	55.2	74.1	81.4	83.5	-	-
	RetCL (Ours)	**78.9**	**90.4**	**93.9**	**95.2**	**95.8**	**96.7**

Table 8: The top-k exact match accuracy (%) of our RetCL, Transformer [1] and GLN [8] with discarding predictions not in the candidate set $\mathcal{C}$.

Category	Method	Top-1	Top-5	Top-10	Top-50	Top-100	Top-200
Reaction type is unknown							
Template-free	Transformer [1]	59.6	74.3	77.0	79.4	79.5	79.6
	RetCL (Ours)	71.3	**92.0**	**94.1**	**96.4**	**96.7**	**97.1**
Template-based	GLN [8]	**77.3**	90.0	92.5	93.3	93.3	93.3
Reaction type is given as prior							
Template-free	Transformer [1]	68.4	82.4	84.3	85.9	86.0	86.1
	RetCL (Ours)	78.9	**93.9**	**95.2**	**96.7**	**97.1**	**97.5**
Template-based	GLN [8]	**82.0**	91.7	92.9	93.3	93.3	93.3

E Failure case study

Figure 3 shows examples of wrong predictions from our framework. We found that the reactants of wrong predictions are still similar to the ground-truth ones. For example, the top-3 predictions of the examples A and B are partially correct; the larger reactant is correct while the smaller one is slightly different. In the example C, the ring at the center of the product is broken in the ground-truth reactants while our RETCL predicts non-broken reactants. Surprisingly, in a chemical database, Reaxys, we found a synthetic route starting from reactants in the top-2 prediction to synthesize the target product.

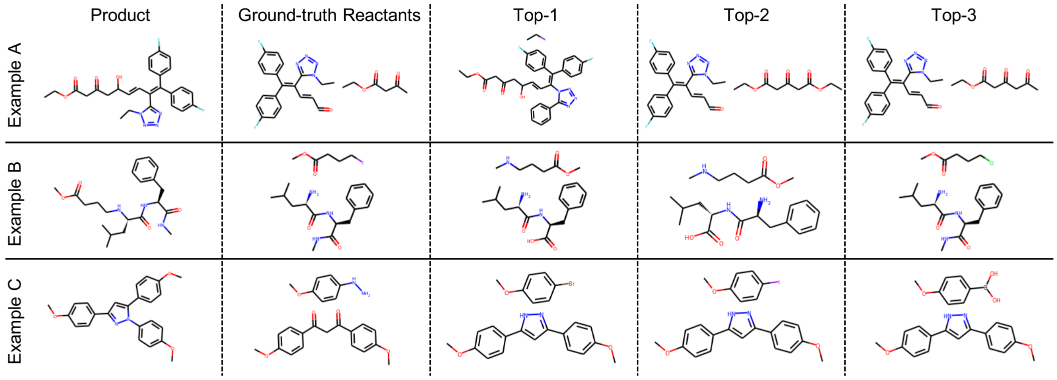


Figure 3: Failure cases of RETCL.

As illustrated in Figure 4, the prediction exists as a 3-step reaction with two reagents (sodium acetate and thiophene) in the chemical literature [9].[4] Note that our framework currently does not consider reagent prediction. Therefore, our prediction can be regarded as an available (i.e., correct) synthetic path in practice.

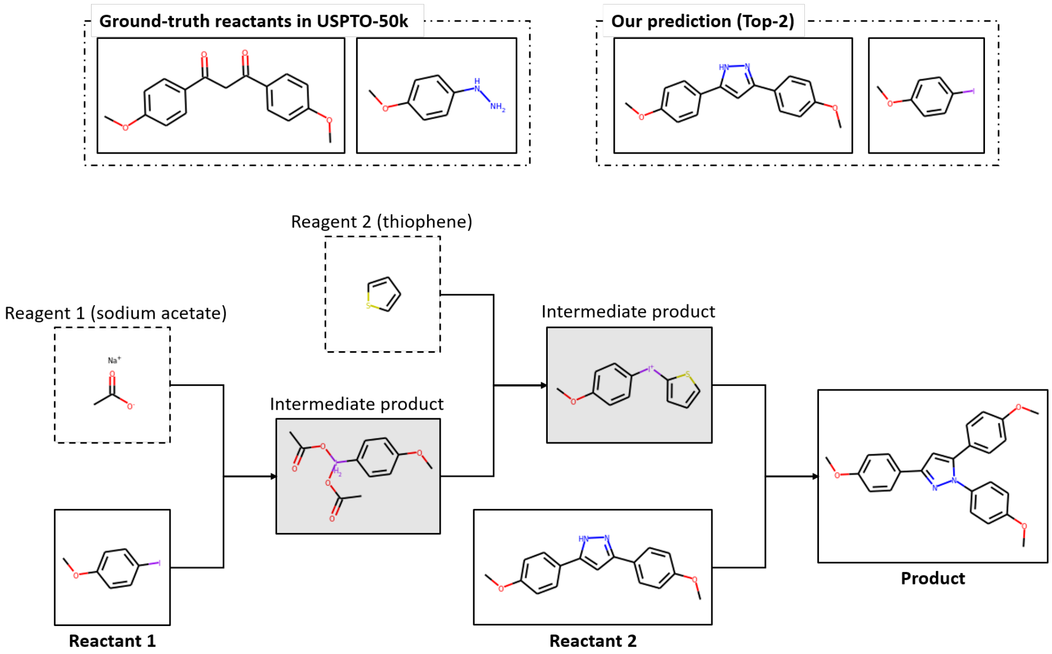


Figure 4: A synthetic path existing in Reaxys based on RETCL's prediction.

[4] We found this synthetic path and the corresponding literature from a chemical database, Reaxys. Note that the sodium acetate and the thiophene are considered as reagents in Reaxys.

F Dataset details

We here describe the details of USPTO datasets. The reactions in the USPTO datasets are derived from the US patent literature [17]. The entire set, USPTO 1976-2016, contains 1.8 million raw reactions. The commonly-used benchmark of single-step retrosynthesis is USPTO-50k containing 50k clean atom-mapped reactions which can be classified into 10 broad reaction types [21]. See Table 9a for the information of the reaction types. For generalization experiments, we introduce a filtered dataset, USPTO-50k-modified, which contains reactions of reaction types from 1 to 5. We report the number of reactions of the modified dataset in Table 9b. We also use the USPTO-full dataset, provided by [8], which contains 1.1 million reactions. Note that we use only the test split of USPTO-full (i.e., only 101k reactions) for testing generalizability. Note that we do not use atom-mappings in the USPTO benchmarks. Moreover, we do not consider reagents for single-step retrosynthesis following prior work [8, 13, 15, 16].

Table 9: The detailed information on USPTO datasets.

(a) The information about reaction types in USPTO-50k.

ID	Fraction (%)	Description
1	30.3	heteroatom alkylation and arylation
2	23.8	acylation and related processes
3	11.3	C-C bond formation
4	1.8	heterocycle formation
5	1.3	protections
6	16.5	deprotections
7	9.2	reductions
8	1.6	oxidations
9	3.7	functional group interconversion (FGI)
10	0.5	functional group addition (FGA)

(b) The number of reactions in USPTO datasets.

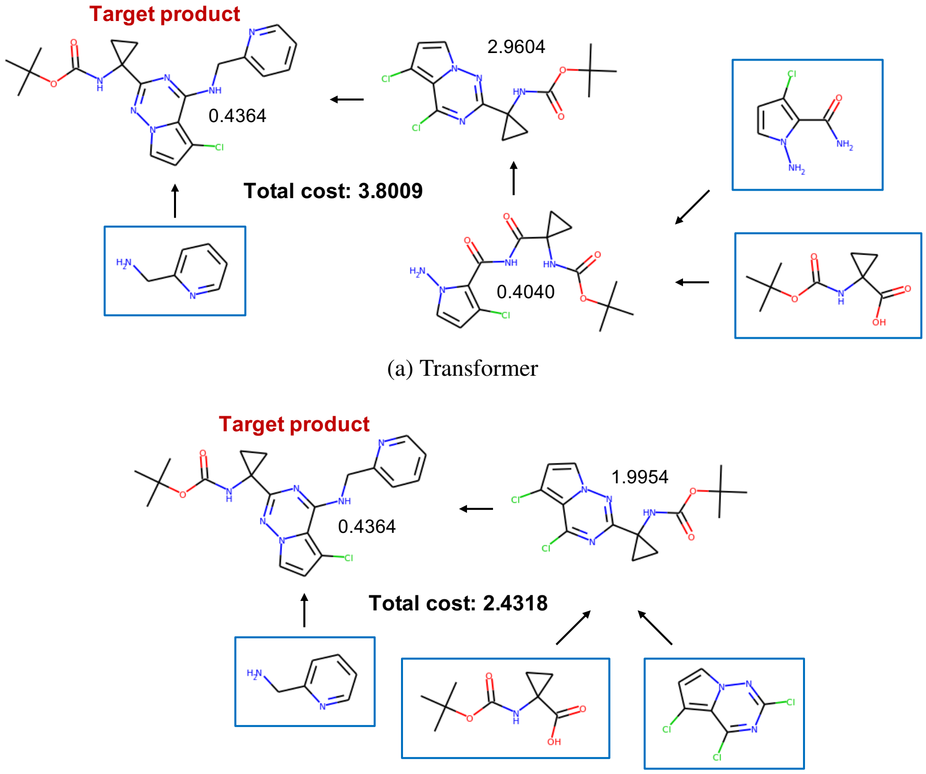
Dataset	Split	# of reactions
USPTO-50k	Train	40,008
	Val	5,001
	Test	5,007
USPTO-50k-modified	Train	27,429
	Val	3,429
	Test	5,007
USPTO-full	Train	810,496
	Val	101,311
	Test	101,311

G Multi-step retrosynthesis

For the multi-step retrosynthesis experiment, we use a synthetic route dataset provided by [2]. This dataset is constructed from the USPTO [17] database like other benchmarks. We recommend to see [2] for the construction details. The dataset contains 299202 training routes, 65274 validation routes, and 190 test routes. We first extract single-step reactions and molecules from the training and validation splits of the dataset. The extracted reactions are used for training our RETCL and Transformer (TF, [1]), and the molecules are used as the candidate set $\mathcal{C}_{\texttt{train}}$[5] for ours. When testing the single-step models with Retro* [2], we use all starting molecules (i.e., 114802 molecules) in the routes in the dataset as the candidate set $\mathcal{C}$. This reflects more practical scenarios because intermediate reactants often be unavailable in multi-step retrosynthesis. We remark that TF also uses the candidate set $\mathcal{C}$ as the prior knowledge for finishing the search procedure.

The evaluation metrics are *success rate* and *average length of routes*. The success means that a synthetic route for a target product is successfully discovered under a limit of the number of expansions. We set the limit by 100 and use only the top-5 predictions of a single or hybrid model for each expansion. When computing the average length, we only consider the cases where all the single-step models discover routes successfully. As [2] did, we use the negative log-likelihood computed by TF as the reaction cost.

Figure 5 and 6 illustrate the discovered routes by TF and RETCL+TF under the aforementioned setting. The molecules in the blue boxes are building blocks (i.e., available reactants) and the numbers indicate the reaction costs (i.e., the negative log-likelihoods computed by TF). As shown in the figures, our algorithm allows to discover a shorter and cheaper route.

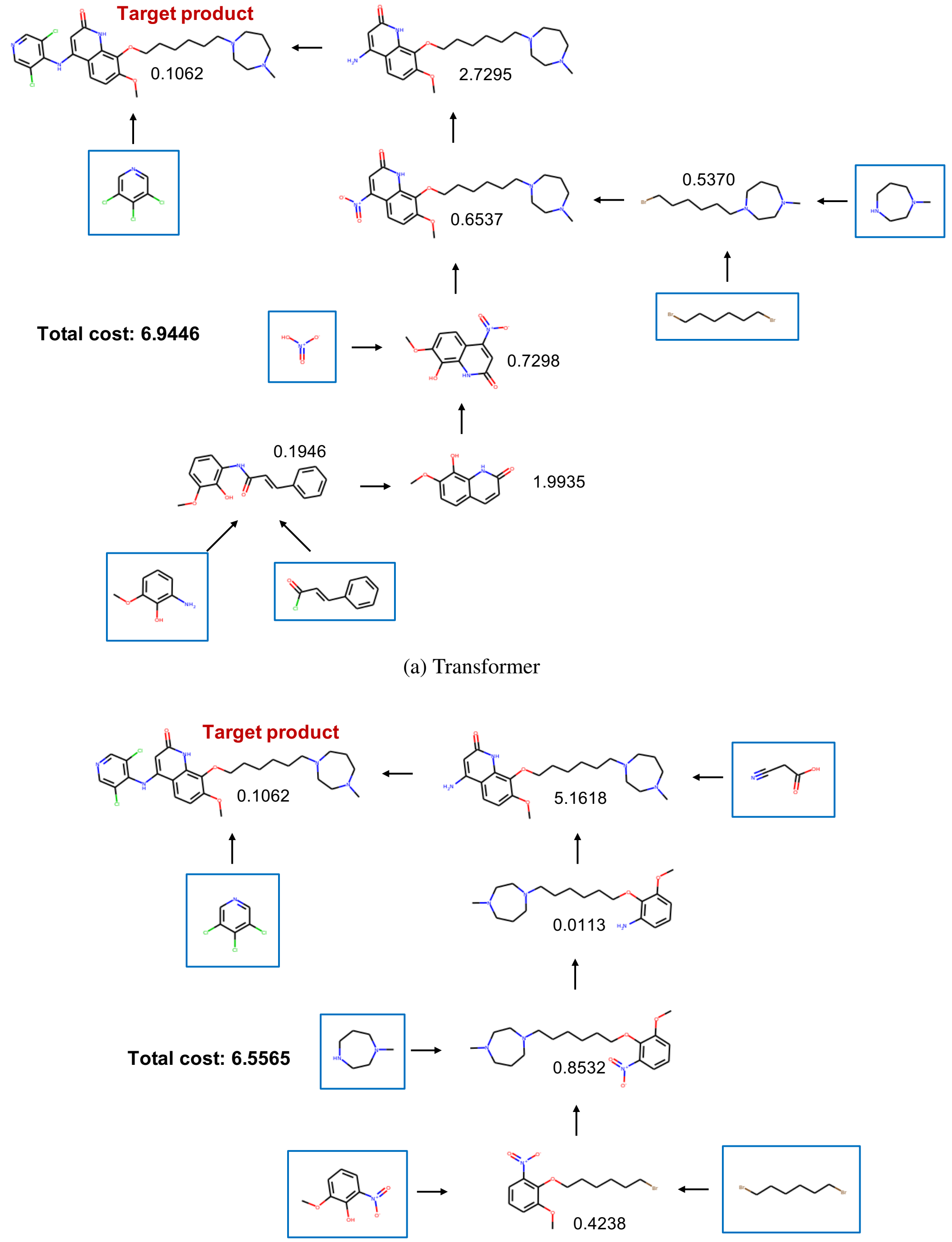


(a) Transformer

(b) RETCL+Transformer (ours)

Figure 5: Synthetic routes discovered by (a) Transformer and (b) our RETCL+Transformer.

[5] Note that this candidate set is used only for training.



(a) Transformer

(b) RETCL+Transformer (ours)

Figure 6: Synthetic routes discovered by (a) Transformer and (b) our RETCL+Transformer.