Electronic Supplementary Material (ESI) for Chemical Science.

Supporting Information for Root-aligned SMILES: A Tight Representation for Chemical Reaction Prediction

Zipeng Zhong,[†] Jie Song,[‡] Zunlei Feng,[‡] Tiantao Liu,[¶] Lingxiang Jia,[†] Shaolun Yao,[†] Min Wu,[§] Tingjun Hou,[*,¶] and Mingli Song[*,†]

[†]*College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, P.R. China*

[‡]*School of Software Technology, Zhejiang University, Ningbo, Zhejiang, P.R. China*

[¶]*Innovation Institute for Artificial Intelligence in Medicine of Zhejiang University, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, Zhejiang, P.R. China*

[§]*Hangzhou Huadong Medicine Group Pharmaceutical Research Institute, Hangzhou, Zhejiang, P.R. China*

E-mail: tingjunhou@zju.edu.cn; brooksong@zju.edu.cn

Additional Information for Methods

Vanilla transformer We take the vanilla transformer as the backbone of our autoencoder. Vanilla transformer[1] is an end-to-end model following a stepwise and autoregressive encoder-decoder fashion. Taking the product SMILES and partially decoded reactant SMILES as the input, it is trained to predict the next token of reactant SMILES.The key idea of the vanilla transformer is the attention mechanism, which allows each token to capture the global information and is quite suitable for SMILES representation. The encoder and decoder are both composed of multiple stacked multihead attention layers consisting of a multihead attention module and a position-wise feed forward module.

Before passing into the encoder, SMILES tokens are embedded to continuous vector representations. The multihead attention module consists of multiple scaled-dot product layers that run in parallel. A single scaled-dot product calculation works as follows:

$$\begin{aligned}
Attention(Q, K, V) &= softmax(\frac{QK^T}{\sqrt{d_k}})V \\
Q &= W_Q X' \\
K &= W_K X' \\
V &= W_V X'
\end{aligned} \tag{1}$$

where Q, K, V represent query, key, value matrix, respectively; W_Q, W_K, W_V are all trainable parameters; d_k means the dimension of K. Depending on where Q, K, V come from, multihead attention can be the self-attention mechanism or cross-attention mechanism. After the attention calculation of each head, they can be concatenated as follows:

$$\begin{aligned}
Z &= Concat(h_0, h_1, ...)W^0 \\
h_i &= Attention(Q_i, K_i, V_i)
\end{aligned} \tag{2}$$

The position-wise feed forward module is a simple fully connected layer that utilizes the concept of residual block and works as follows:

$$FFN(Z) = max(0, W_1 z + b_1)W_2 + b_2 \tag{3}$$

After the calculation of feed forward module, updated token vectors can be passed to another multihead attention layer. We use the vanilla transformer architecture composed of 6 layers for both encoder and decoder with 8 attention heads for all experiments. During the inference stage, the transformer takes the product SMILES and decoded reactant as the input to predict the probability of the next reactant SMILES token, which can be represented by a conditional probability distribution:

$$p(y|X) = \prod_{i=1}^{m} p(y_i|y_{<i}, X) \tag{4}$$

where m is the maximum number of reactant tokens. The "bos" (begin of sentence) token is the beginning of reactant tokens. When the last predicted token is "eos" (end of sentence), the decoding process completes.

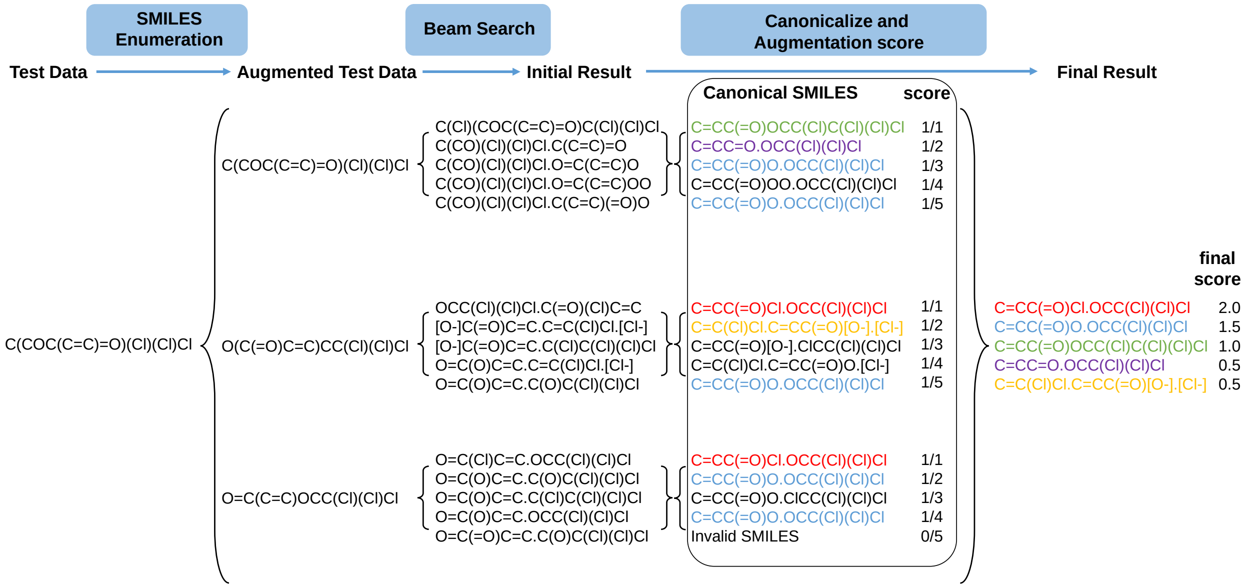
SMILES Enumeration
Beam Search
Canonicalize and Augmentation score
Test Data
Augmented Test Data
Initial Result
Final Result
Canonical SMILES
score
C(COC(C=C)=O)(Cl)(Cl)Cl
C(COC(C=C)=O)(Cl)(Cl)Cl
O(C(=O)C=C)CC(Cl)(Cl)Cl
O=C(C=C)OCC(Cl)(Cl)Cl
C(Cl)(COC(C=C)=O)C(Cl)(Cl)Cl
C(CO)(Cl)(Cl)Cl.C(C=C)=O
C(CO)(Cl)(Cl)Cl.O=C(C=C)O
C(CO)(Cl)(Cl)Cl.O=C(C=C)OO
C(CO)(Cl)(Cl)Cl.C(C=C)(=O)O
OCC(Cl)(Cl)Cl.C(=O)(Cl)C=C
[O-]C(=O)C=C.C=C(Cl)Cl.[Cl-]
[O-]C(=O)C=C.C(Cl)C(Cl)(Cl)Cl
O=C(O)C=C.C=C(Cl)Cl.[Cl-]
O=C(O)C=C.C(O)C(Cl)(Cl)Cl
O=C(Cl)C=C.OCC(Cl)(Cl)Cl
O=C(O)C=C.C(O)C(Cl)(Cl)Cl
O=C(O)C=C.C(Cl)C(Cl)(Cl)Cl
O=C(O)C=C.OCC(Cl)(Cl)Cl
O=C(=O)C=C.C(O)C(Cl)(Cl)Cl
C=CC(=O)OCC(Cl)C(Cl)(Cl)Cl 1/1
C=CC=O.OCC(Cl)(Cl)Cl 1/2
C=CC(=O)O.OCC(Cl)(Cl)Cl 1/3
C=CC(=O)OO.OCC(Cl)(Cl)Cl 1/4
C=CC(=O)O.OCC(Cl)(Cl)Cl 1/5
C=CC(=O)Cl.OCC(Cl)(Cl)Cl 1/1
C=C(Cl)Cl.C=CC(=O)[O-].[Cl-] 1/2
C=CC(=O)[O-].ClCC(Cl)(Cl)Cl 1/3
C=C(Cl)Cl.C=CC(=O)O.[Cl-] 1/4
C=CC(=O)O.OCC(Cl)(Cl)Cl 1/5
C=CC(=O)Cl.OCC(Cl)(Cl)Cl 1/1
C=CC(=O)O.OCC(Cl)(Cl)Cl 1/2
C=CC(=O)O.ClCC(Cl)(Cl)Cl 1/3
C=CC(=O)O.OCC(Cl)(Cl)Cl 1/4
Invalid SMILES 0/5
final score
C=CC(=O)Cl.OCC(Cl)(Cl)Cl 2.0
C=CC(=O)O.OCC(Cl)(Cl)Cl 1.5
C=CC(=O)OCC(Cl)C(Cl)(Cl)Cl 1.0
C=CC=O.OCC(Cl)(Cl)Cl 0.5
C=C(Cl)Cl.C=CC(=O)[O-].[Cl-] 0.5

Figure S1: The workflow of model prediction with the beam search and data augmentation. The example applies 3× augmentation, and sets *beam size* = 5, *topk* = 5.

Data augmentation with R-SMILES We successively perform data augmentation and root alignment on the training data, and only perform data augmentation on the test data. When inferring on the validation and test data, we input multiple SMILES of a molecule respectively and get multiple sets of outputs correspondingly. After removing invalid SMILES that cannot be recognized by Rdkit[2] and converting all outputs to canonical SMILES, we refer to Tekto *et al.*'s approach[3] that scores these outputs uniformly as follows:

$$score(output) = \sum_{n=1}^{augmentation} \sum_{i=k}^{topk} \frac{1}{k} \tag{5}$$

where *augmentation* represents the augmentation times of the test set, and *beam* represents the beam size. After scoring uniformly, we can select outputs with top-K scores as the final result.

Here we show an example of performing data augmentation for the test data in Figure S1. When the SMILES "C(COC(C=C)=O)(Cl)(Cl)Cl" was input, first we performed SMILES enumeration to get three different SMILES representing the same molecule. Then we started model prediction using the beam search strategy with the beam size of 5, and got the top-5 prediction for each SMILES. To score uniformly, we converted them to the canonical SMILES and removed those invalid SMILES. According to the ranking of the initial results, we can give each output an initial score $\frac{1.0}{k}$. For example, the rank-1 prediction "C=CC(=O)OCC(Cl)C(Cl)(Cl)Cl" of "C(COC(C=C)=O)(Cl)(Cl)Cl" was scored one. If the prediction was an invalid SMILES, we would score it zero. After getting the scores for each output, we can score them uniformly by adding the scores of the same output. For example, since "C=CC(=O)Cl.OCC(Cl)(Cl)Cl" was the rank-1 prediction for two inputs "O(C(=O)C=C)CC(Cl)(Cl)Cl" and "O=C(C=C)OCC(Cl)(Cl)Cl", its final score was two. Therefore, we can acquire the scores for all the output and got a uniform ranking, i.e, the final result.

Table S1: The effect of choosing different values of α on the top-K single-step retrosynthesis.

	USPTO-50K top-K Accuracy (%)				**USPTO-MIT top-K Accuracy (%)**				**USPTO-FULL top-K Accuracy (%)**			
α	K = 1	3	5	10	K = 1	3	5	10	K = 1	3	5	10
0.001	54.2	75.7	85.0	89.5	59.4	76.0	81.3	86.0	47.4	63.5	69.5	74.8
0.01	54.1	75.7	85.1	89.5	58.8	74.7	79.2	84.3	47.8	63.7	68.3	73.4
0.1	55.1	77.6	84.0	90.0	59.3	76.6	81.9	86.4	48.8	64.9	70.1	76.0
1	**56.0**	**79.4**	**86.2**	**91.2**	**60.4**	**78.4**	83.4	**87.6**	**49.0**	66.4	71.8	**76.4**
10	55.9	79.2	86.1	91.1	60.2	**78.4**	**83.5**	87.5	48.9	**66.5**	71.6	**76.4**
100	55.9	79.1	**86.2**	**91.2**	60.3	78.2	83.4	87.5	48.9	**66.5**	**71.9**	**76.4**

There is a scoring trade-off in the test set data augmentation, that is, how to weigh the number of prediction occurrences against the ranking of predictions. Suppose there are two predictions, one that is predicted by one input and ranks first, and the other that is predicted by two different inputs and ranks second and third respectively, which of them should get a higher final score? This problem can be expressed by the following equation:

$$score(output) = \sum_{n=1}^{augmentation} \sum_{i=k}^{topk} \frac{1}{1 + \alpha * (k - 1)} \tag{6}$$

where α is the weighing parameter. The higher α, the more important the ranking and the less important the number of appearances, and vice versa. We tested different values of α and show the accuracy of the validation set in the Table S1. It can be seen that for all datasets, the best results are obtained when the α value is equal to or greater than one, which demonstrates the ranking is more important than the number of occurrences. Moreover, when the α value is one, most of the top-K accuracies are the highest. In fact, Eq. 5 is the case where α takes the value of one in Eq. 6.

Table S2: Training time and steps for different tasks. "R2P" denotes the forward reaction prediction, and "P2R" denotes the retrosynthesis prediction. "From Scratch" denotes the model is trained without pretraining.

Dataset	**USPTO-50K**	**USPTO-MIT**	**USPTO-Full**
Pretrain	-	-	130 hours / 1,000,000 steps
Finetune - R2P	-	30 hours / 500, 000 steps	-
Finetune - P2R	20 hours / 300,000 steps	50 hours / 500,000 steps	50 hours / 500,000 steps
From Scratch - R2P	-	60 hours / 100,000 steps	-
From Scratch - P2R	30 hours / 600,000 steps	110 hours / 2,000,000 steps	120 hours / 1,500,000 steps

Training settings We use the same data split as previous researchers[4–6] for all the datasets. During the pretraining stage. Depending on whether it is a forward or retrosynthesis prediction, products or reacatants in the training set of USPTO-FULL are used for self-supervised training, where molecules in the test set of USPTO-50K and USPTO-MIT are removed. We apply 20× augmentation at training and test sets of USPT0-50K, and 5× augmentation at training and test sets of USPTO-MIT and USPTO-FULL. We set the embedding and hidden size as 512 except that the dimension of Q , K , V is 64. We also use the Adam optimizer and a varied learning rate with 8,000 warm up steps. The input and output share the same vocabulary, but their embedding layers are independent. We conducted all experiments with one NVIDIA GeForce RTX 3090 GPU, and the approximate training time and steps for different tasks are displayed in Fig. S2. The more detailed settings can be found at https://github.com/otori-bird/retrosynthesis.

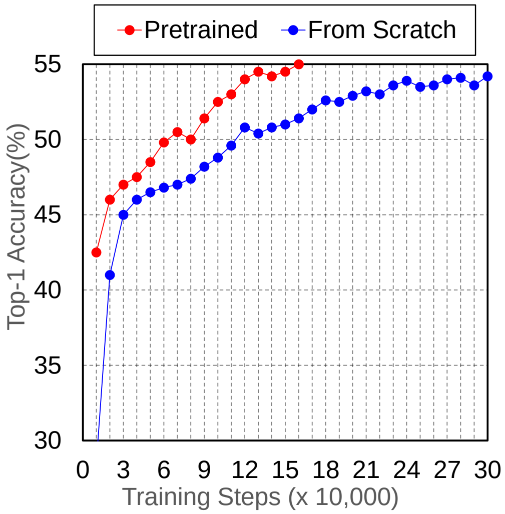

Figure S2: Training steps with/without the pretrained model on USPTO-50K for the P2R stage. The training set of USPTO-50K is applied 20× augmentation.

Analysis of Pretrained Transformer

We used a similar approach to the masked language model of BERT[7] for pretraining. Specifically, 15% of tokens in SMILES are masked. Every masked token has an 80% probability of being replaced with the "unknown" token, a 10% probability of being replaced with any token in the vocabulary, and keeps unchanged for the rest of the cases. After pretraining, we can see in Figure S2 and Table S2 that the training time has been dramatically reduced, which helps a lot in the case of very limited computational resources.

Table S3: The edit distance and top-K accuracy of single-step retrosynthesis for ring and non-ring reactions on the USPTO-MIT and USPTO-FULL datasets.

	USPTO-MIT					**USPTO-FULL**				
Reaction Type	**edit distance**	**K=1**	**3**	**5**	**10**	**edit distance**	**K=1**	**3**	**5**	**10**
Overall[a]	26.7	53.8	70.4	75.1	77.7	29.2	44.7	61.0	65.8	68.7
Non-ring reaction[a]	25.9	55.9	73.6	78.5	81.3	27.3	49.6	67.7	72.9	76.0
Ring-opening reaction[a]	33.5	39.7	48.0	50.4	52.0	39.1	25.3	33.3	35.8	37.8
Ring-forming reaction[a]	23.7	37.9	51.6	56.1	58.7	28.4	34.1	49.2	54.3	57.5
Overall[b]	13.5 (-49%)	60.4	78.0	83.0	86.9	16.6 (-43%)	48.9	66.5	71.8	76.7
Non-ring reaction[b]	12.5 (-52%)	62.8	80.5	85.3	89.4	13.9 (-49%)	54.1	72.6	77.9	83.0
Ring-opening reaction[b]	21.8 (-35%)	43.3	57.2	62.9	67.2	28.4 (-28%)	27.3	39.1	43.9	48.8
Ring-forming reaction[b]	15.2 (-36%)	46.8	63.8	70.4	76.8	20.8 (-27%)	39.3	56.7	63.2	69.6

[a] Without root alignment; [b] With root alignment.

Limitations for ring-opening and ring-forming reactions

The root alignment strategy does not always work well. Taking the ring-opening reaction "[CH2:1]1[CH:2]([NH2:3])[CH:4]([OH:5])[O:6][CH2:7]1>> [CH2:1]([CH:2]([NH2:3])[CH:4]=[O:5])[CH2:7]" as an example, where the bond between "[O:6]" and "[CH:4]" is split, we can get both the reasonable aligned result and largely unaligned one. Aligning at the root atom "[O:6]" yields reasonable results "O1CCC(N)C1O" and "OCCC(N)C=O", whereas aligning at root atom "[O:5]" yields largely unaligned strings "OC1OCCC1N" and "O=CC(N)CCO". This obviously increases the edit distance between inputs and outputs, which leads to a decrease in prediction performance for this type of data.We calculated the accuracy of retrosynthesis for ring-opening and forming reactions in different datasets. Results are shown in Table S3. It can be seen that the accuracy of R-SMILES, as pointed out by the reviewer, is not so high as that of other reactions. To make it clearer, we also calculated the edit distance between the input and the output SMILES for these reactions. Compared with that of non-ring reaction R-SMILES, the edit distance of ring reactions is significantly larger. These results again verify our main motivation in this work that large distance between input and output strings will degrade the reaction prediction performance. We will be devoted to dealing with this problem in our future work by trying to align more than one atom.

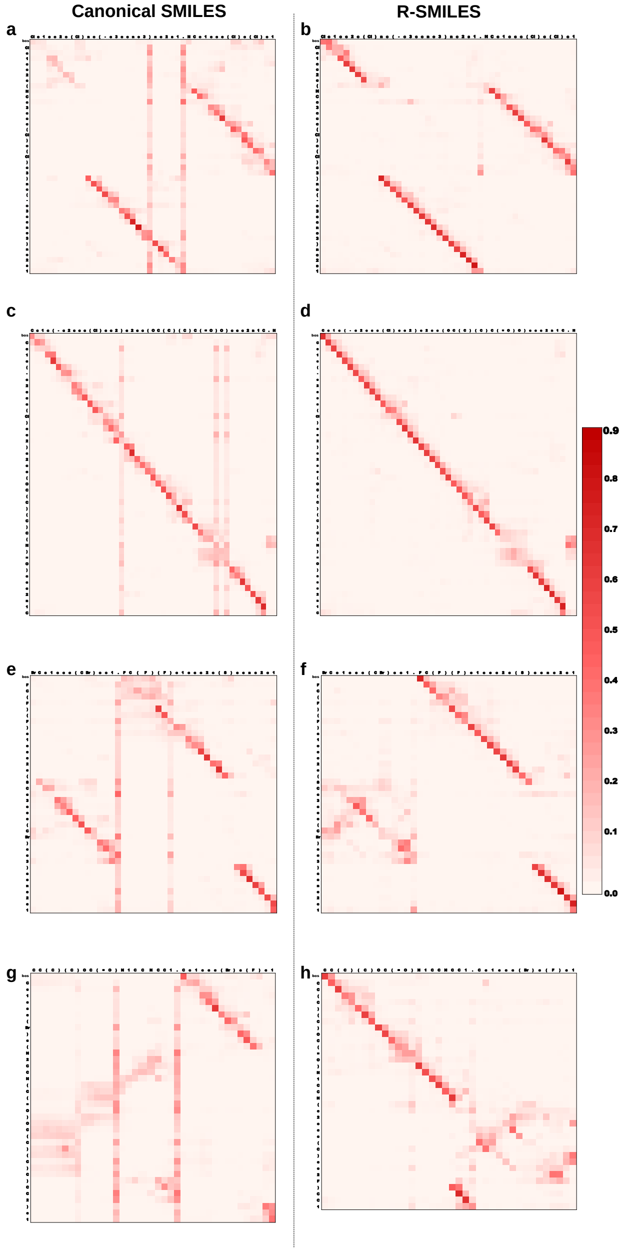

Figure S3: Visualization of the cross-attention obtained by the canonical SMILES (Left) and the proposed R-SMILES (Right) in the forward reaction prediction. (a, c, e, g) The attention maps obtained by the model trained with canonical SMILES. (b, d, f, h) The attention maps obtained by the model trained with R-SMILES. The input tokens are along the x axis, and the output tokens are along the y axis. Each row in the attention map represents the attention over the input tokens for predicting the next output token. Each column represents the attention between an input token with each output token. The "bos" token is the beginning of output tokens and will be removed after the decoding process completes.

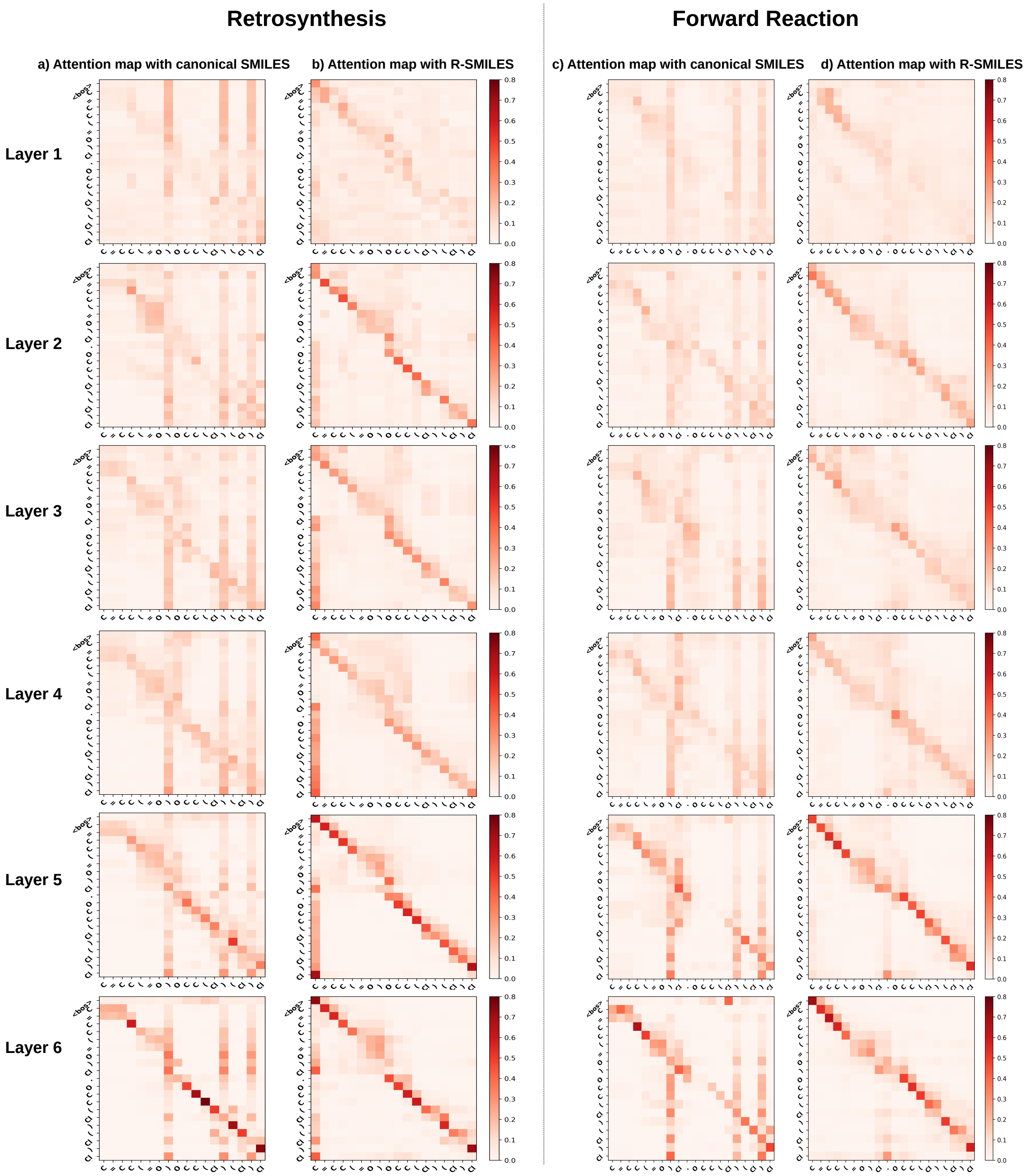

Figure S4: Comparison of the attention of the all the layers. (a, c) are the attention maps obtained by the model trained with canonical SMILES. (b, d) are the attention maps obtained by the model trained with R-SMILES.

Comparison of the attention of all layers

To give a more detailed discussion about attention mechanism, we fed the reactant or the product of the reaction “C=CC(=O)Cl.OCC(Cl)(Cl)Cl>>C=CC(=O)OCC(Cl)(Cl)Cl” to four different models and visualized attention maps of all layers in Fig. S4. For these four models, one is trained with R-SMILES for forward reaction, one trained with R-SMILES for retrosynthesis, one trained with canonical SMILES for forward reaction, and one trained with SMILES for retrosynthesis. With SMILES (Fig. S4a, c), much attention is paid to syntactic tokens at the shallow layers. However, for R-SMILES, the attention maps are always cleaner and nearly diagonal at different layers. Attention maps of the 200 examples can be found at https://github.com/otori-bird/retrosynthesis.

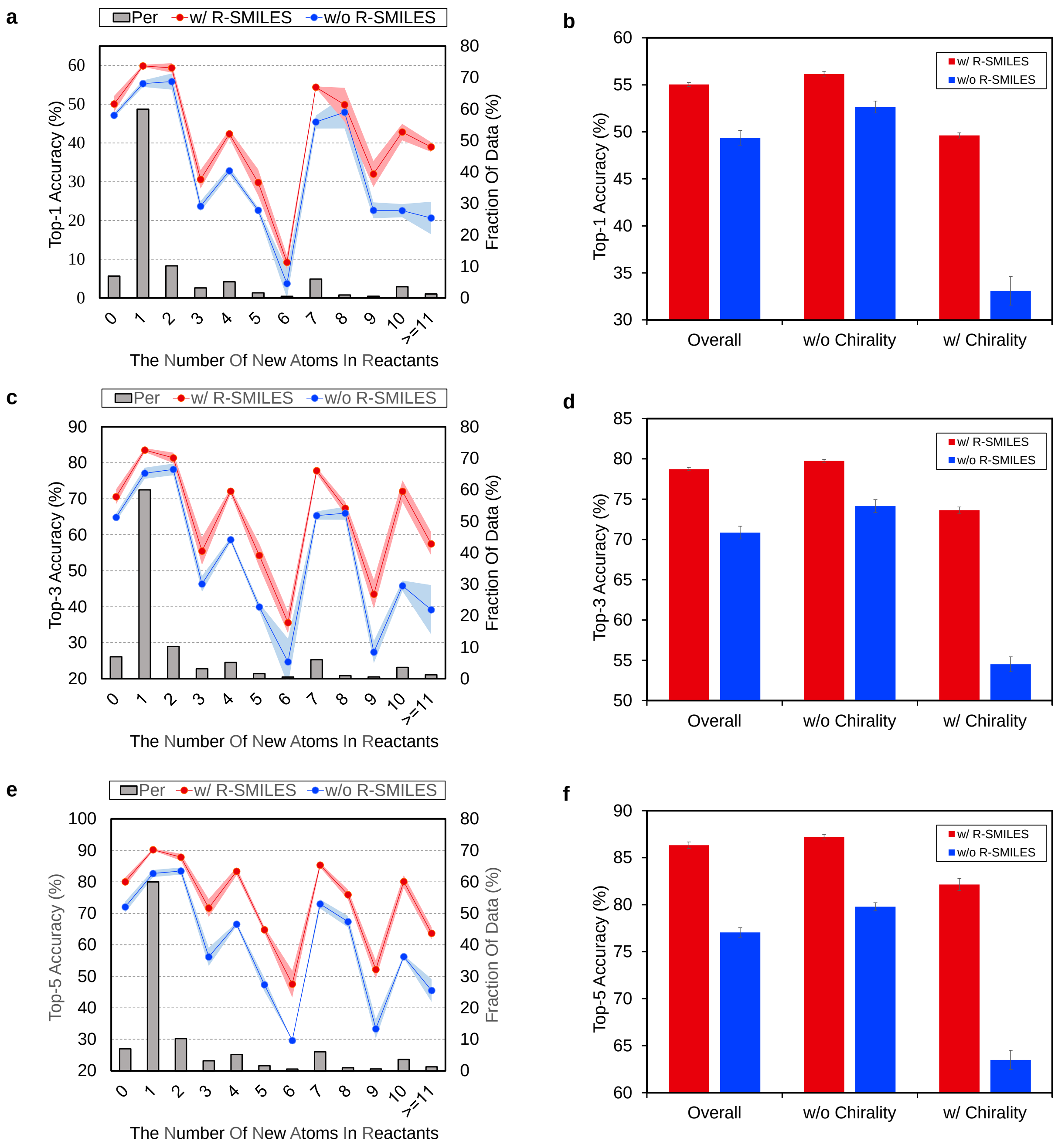

Figure S5: Extra top-K accuracy (%) for complex reactions. **a, c, e,** Top-1, top-3, and top-5 accuracies according to the number of new atoms in reactants. The red and blue lines represent the performance with/without R-SMILES. The gray bar means the percentage of this kind of reaction in the test set. **b, d, f,** Top-1, top-3, and top-5 accuracies for reactions involving with or without chirality. The red and blue bars represent the performance with or without R-SMILES.

References

(1) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; Polosukhin, I. Attention is All you Need. Advances in Neural Information Processing Systems. 2017.

(2) Landrum, G. RDKit: Open-Source Cheminformatics Software. `https://rdkit.org`.

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

(4) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. Computer-assisted retrosynthesis based on molecular similarity. *ACS Cent. Sci.* **2017**, *3*, 1237–1245.

(5) Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. Advances in Neural Information Processing Systems. 2017.

(6) Dai, H.; Li, C.; Coley, C.; Dai, B.; Song, L. Retrosynthesis Prediction with Conditional Graph Logic Network. Advances in Neural Information Processing Systems. 2019.

(7) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *article preprint article:1810.04805* **2018**, DOI: `https://doi.org/10.48550/arXiv.1810.04805`.