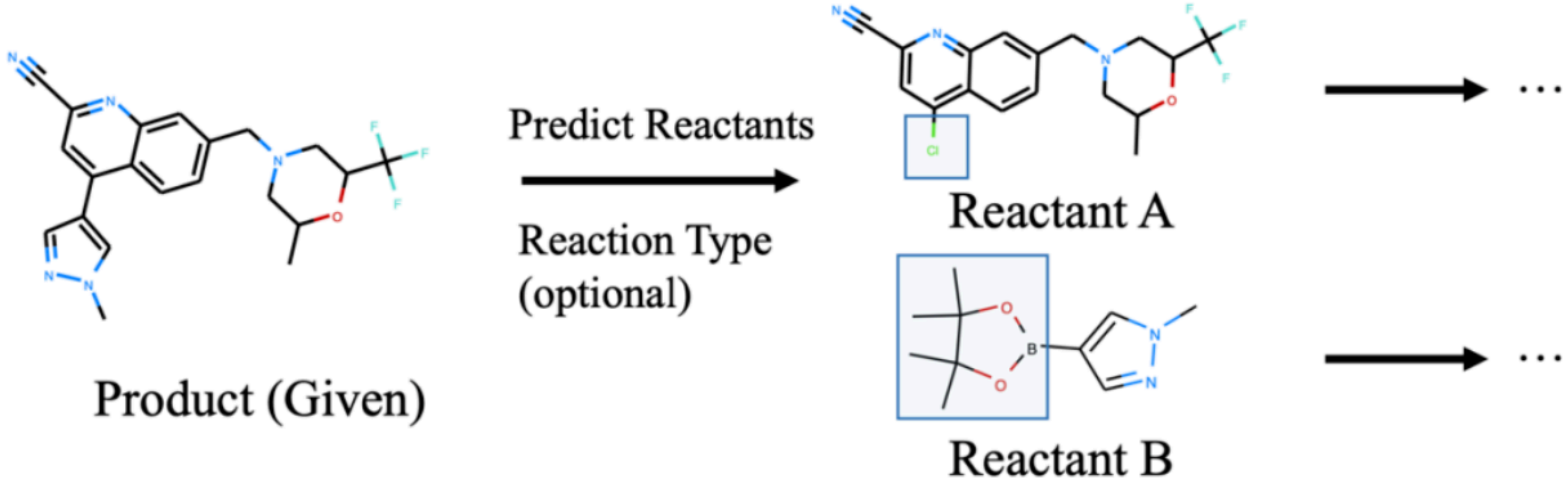
A GRAPH TO GRAPHS FRAMEWORK FOR RETROSYNTHESIS PREDICTION CHANCE

Chence Shi 1 Minkai Xu 2 Hongyu Guo 3 Ming Zhang 1 Jian Tang 4 5 6

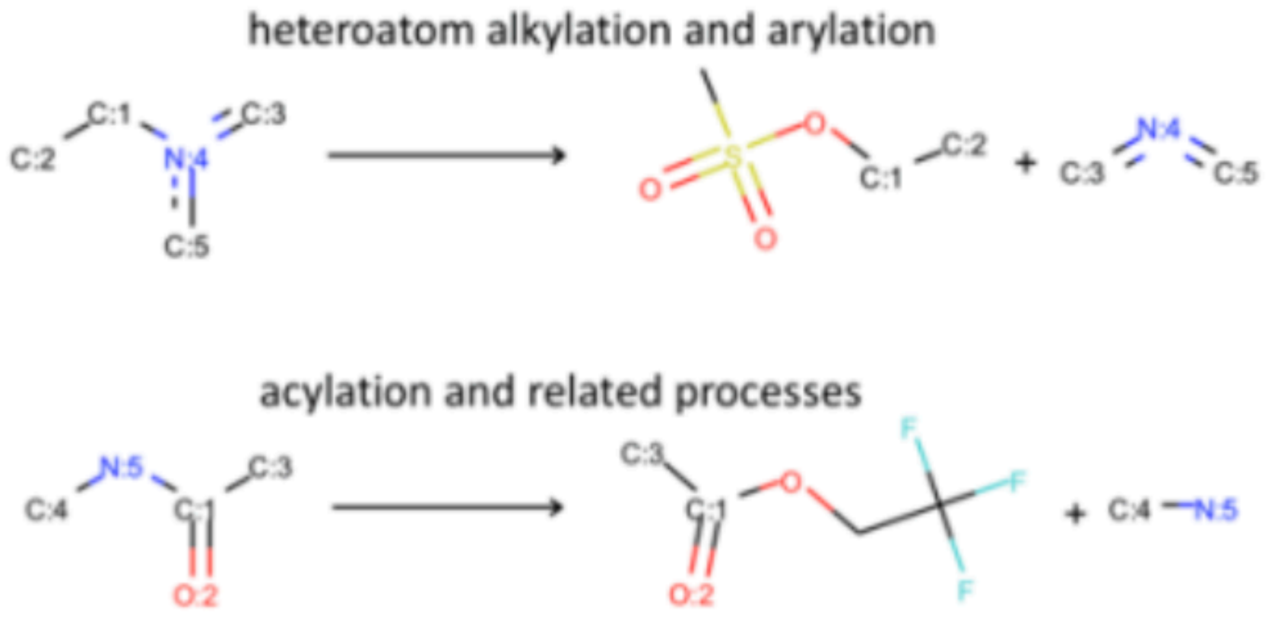
Background: Drug Discovery

- **Retrosynthesis Prediction**
  - **Once a molecular structure is designed, how to synthesize it?**
  - **Goal: Identify a set of reactants that can be used to synthesize a target molecule**



Retrosynthesis Prediction - Template Based

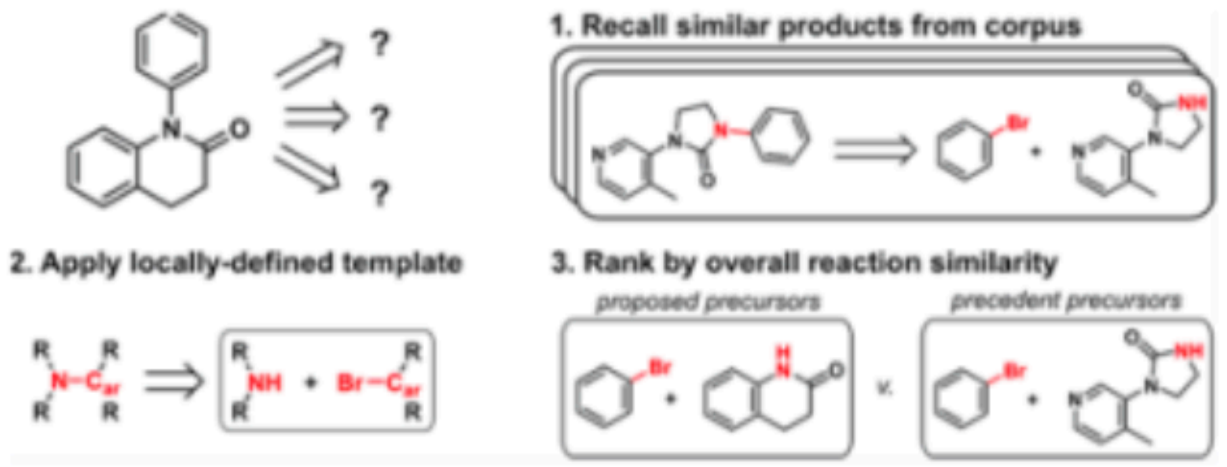
- Retrosim(Corley et al.): template ranking with product-product similarity
- NeuralSymbolic(Segler et al.): template selection as multi-class classification
- GLN(Dai et al.2019): sample template and reactants from conditional joint distribution



heteroatom alkylation and arylation

acylation and related processes

Retrosynthesis Templates. Taken from GLN (Dai et al. 2019)



1. Recall similar products from corpus

2. Apply locally-defined template

3. Rank by overall reaction similarity

Figure from Corley et al. 2017. Computer-Assisted Retrosynthesis Based on Molecular Similarity

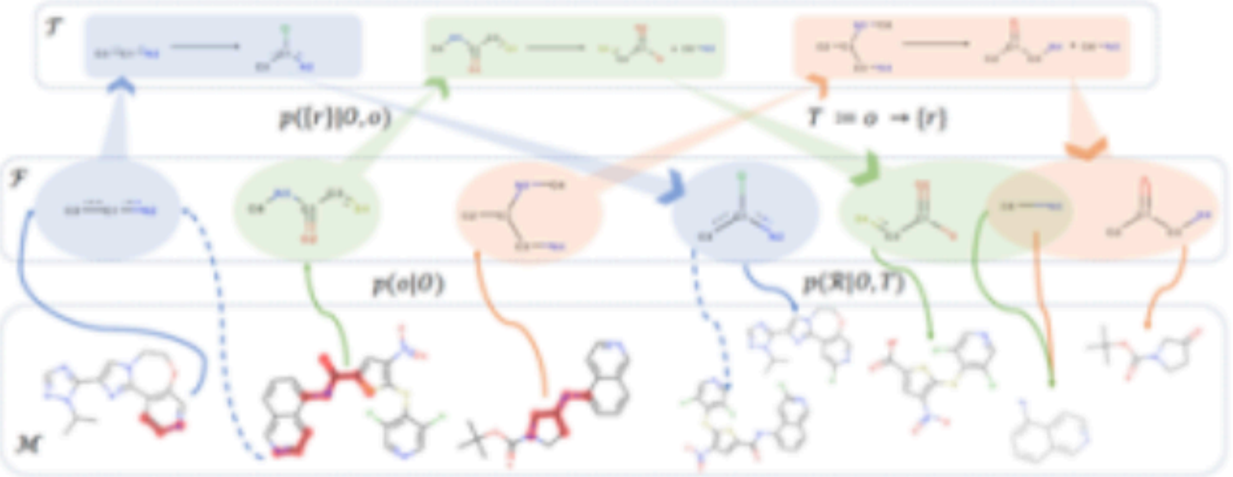


Figure from Dai et al. NeurIPS 2019. Retrosynthesis Prediction with Conditional Graph Logic Network

Retrosynthesis Prediction - Template free

- Sequence to sequence problem (Seq2Seq,Liu et al., 2017)
  - Neural machine translation task
  - SMILES representation of molecules
- Limitations:
  - Not effectively reflect the complex relationships between atoms
  - Unsatisfactory performance

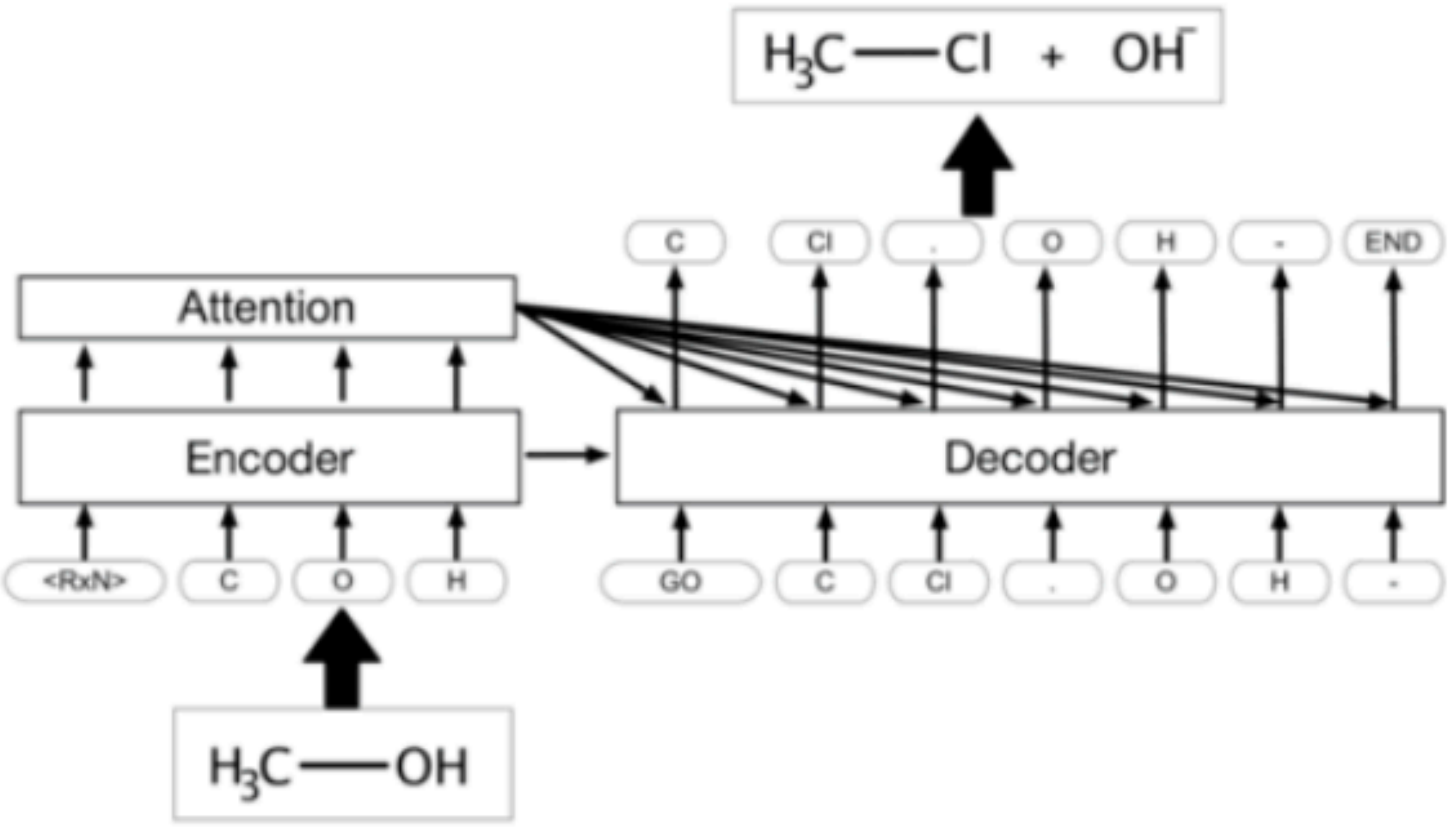
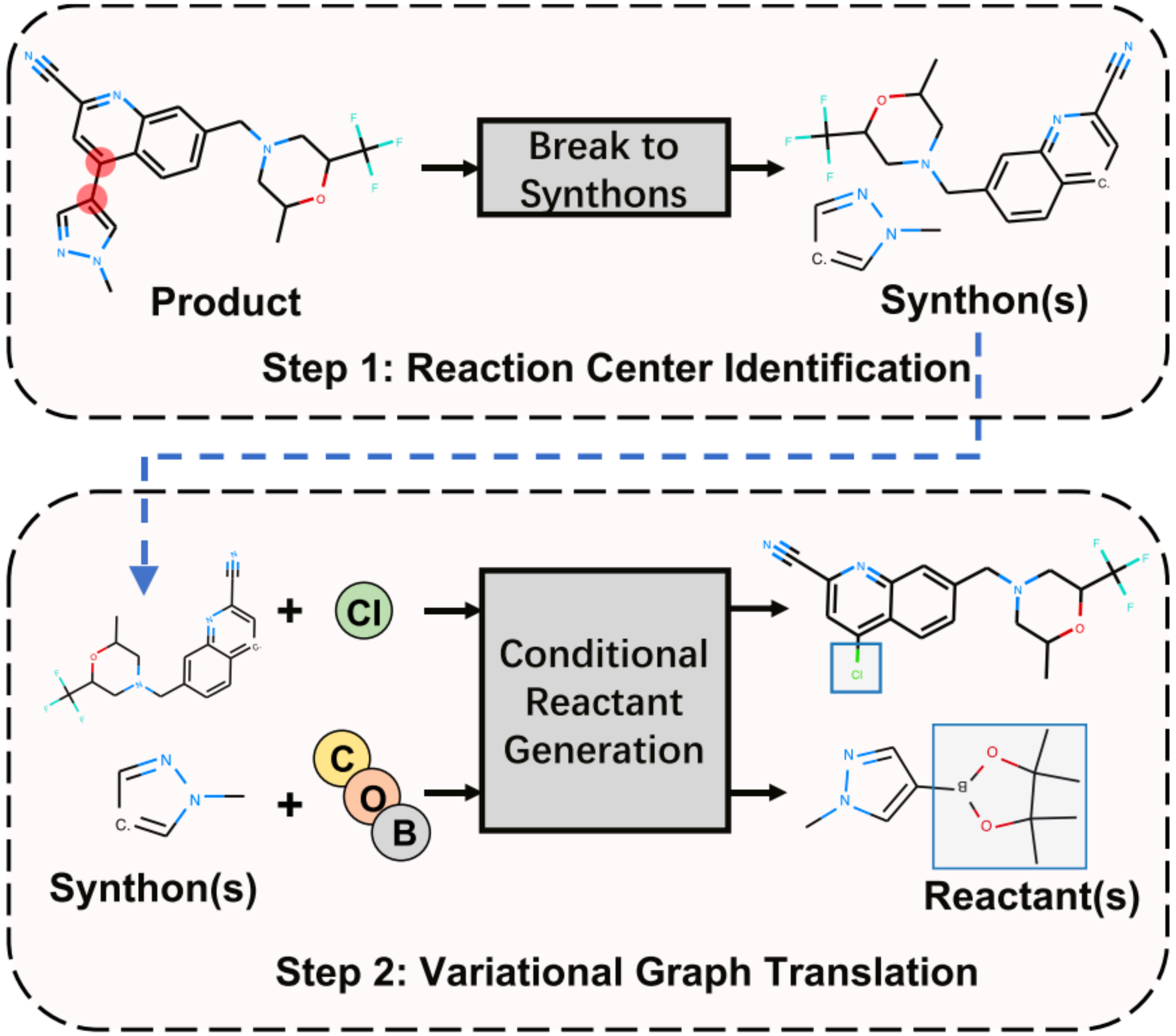


Figure from Liu et al. 2017. Retrosynthetic reaction prediction using neural sequence-to-sequence models

A Graph to Graphs Framework for Retrosynthesis Prediction(Shi et al.ICML2020)

- Represent each molecule as a graph
- Formulate retrosynthesis prediction as a graph-to-graphs translation problem.
- G2Gs first splits the target molecular graph into a set of synthons by identifying the reaction centers, and then translates the synthons to the final reactant graphs via a variational graph translation framework.


Break to Synthons
Product
Synthon(s)
Step 1: Reaction Center Identification
+
Cl
+
C
O
B
Conditional Reactant Generation
Synthon(s)
Reactant(s)
Step 2: Variational Graph Translation

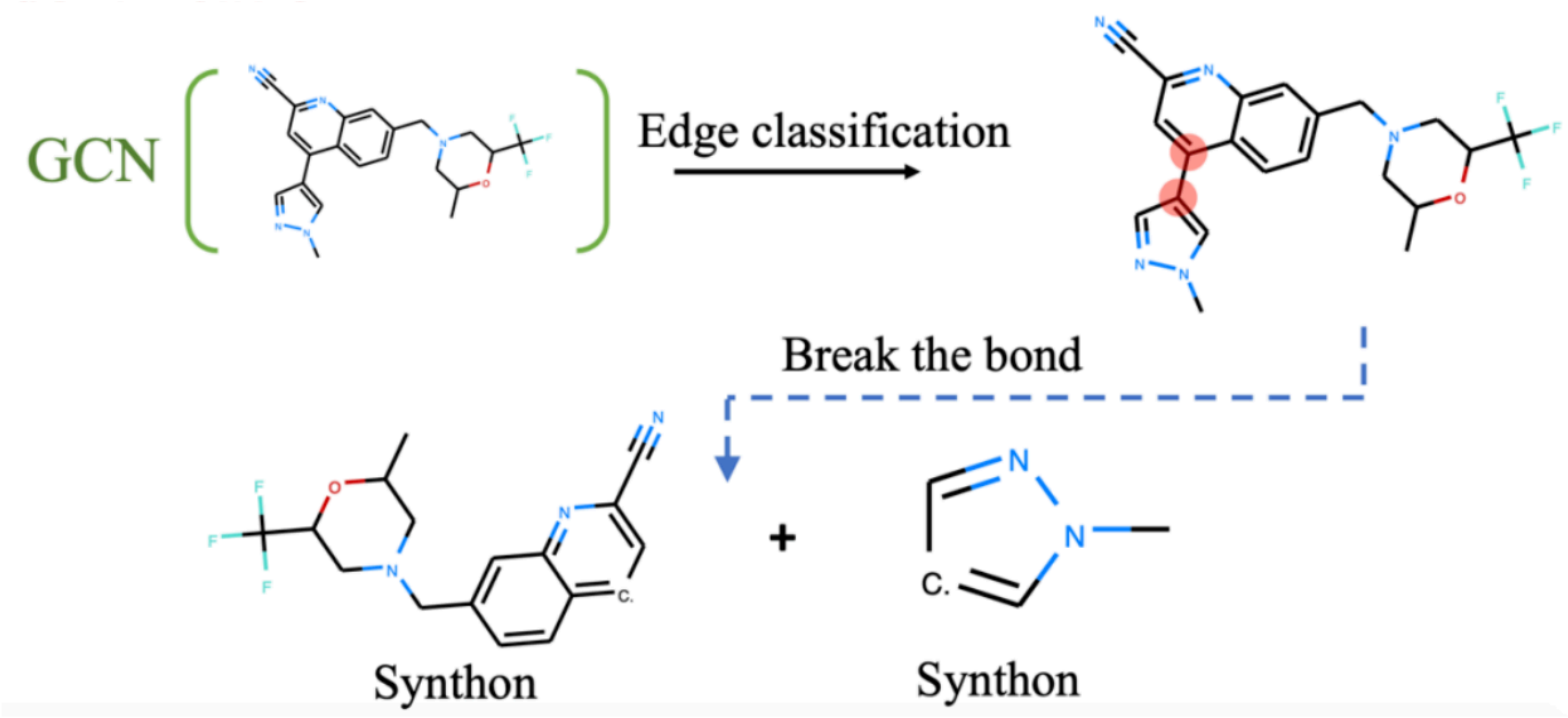
Reaction Center Identification

- Estimate the reactivity score of all atom pairs of the product graph(R-GCN), and the atom pair with the highest reactivity score above a threshold will be selected as the reaction center.

- Split the product graph into synthons by disconnecting the bonds of the reaction center resulted.

- one-to-many graph translation problem

- multiple one-to-one translation processes



R-GCN(Schlichtkrull et al., 2018)

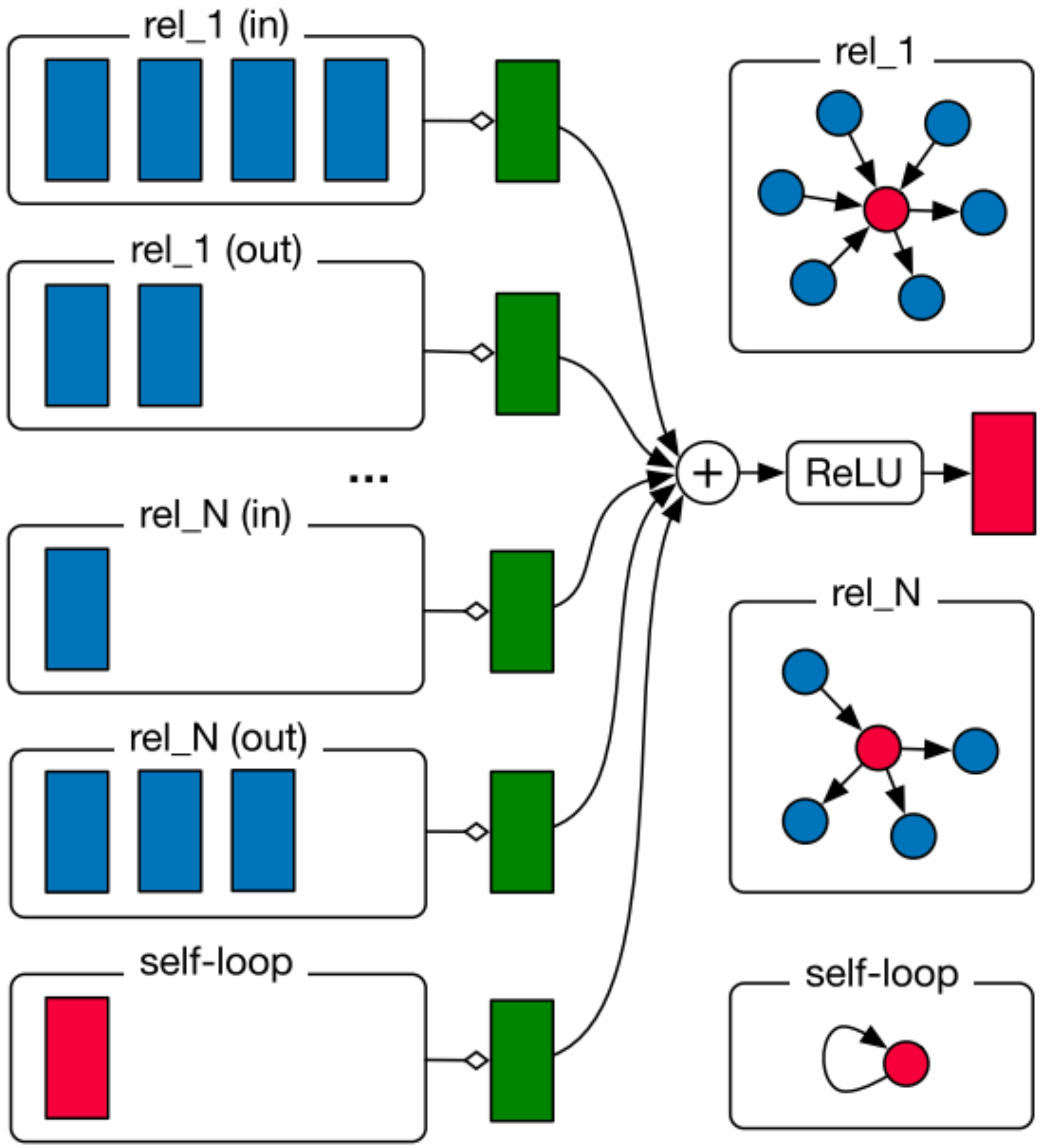
- **message-passing framework (Gilmer et al. 2017)**

$$h_i^{(l+1)} = \sigma\left(\sum_{m\in\mathcal{M}_i} g_m\left(h_i^{(l)}, h_j^{(l)}\right)\right) \qquad g_m\left(h_i, h_j\right) = Wh_j$$

- **For relational(directed and labels) multi-graph**

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right) \qquad W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)},$$

$\mathcal{N}_i^r$: the set of neighbor indices of node i under relation $r \in \mathcal{R}$
$c_{i,r}$: normalization constant: (eg. $c_{i,r} = \left|\mathcal{N}_i^r\right|$).

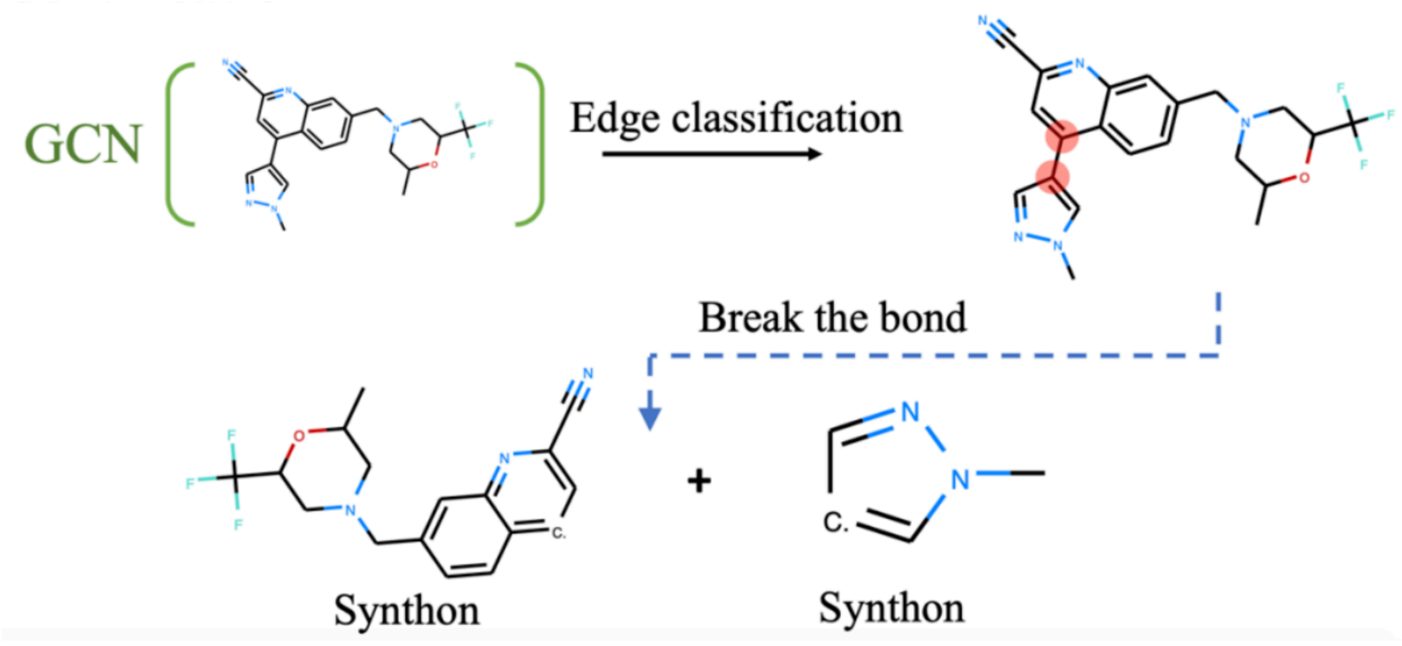


Retrosynthesis Prediction Notation

Notation	Explaination
A	Adjacency matrix $A \in \{0,1\}^{n\times n\times b}$
X	Matrix of node features $X \in \{0,1\}^{n\times d}$
$G=(A,X)$	A molecule representation
G_i ; G_j	reactant graph; product graph
$\left\{G_i\right\}_{i=1}^{N_1}$; $\left\{G_j\right\}_{j=1}^{N_2}$	the set of reactants ; the set of products
$\left(\left\{G_i\right\}_{i=1}^{N_1}, G_p\right)$	A chimical reaction

Molecular Graph Representation Learning

Notation	Explanation
$k \in \mathbb{R}$	embedding dimension
$H^l \in \mathbb{R}^{n \times k}$	node embeddings at the l^{th} layer
H_i^l	the embedding of the l^{th} atom
$A_{[:,:,i]}$	adjacency matrix
I	Identity matrix



node representation:

$$H^l = \mathrm{Agg}\Big(\mathrm{ReLU}\big(\{E_i H^{l-1} W_i^l\} \mid i \in (1, \ldots, b)\big)\Big)$$

$$E_i = A_{[:,:,i]} + I$$

The entire graph-level embedding h_G :
Readout($\cdot$) function to H^L (Hamilton et al., 2017)
e.g., summation.

Reaction Center Identification in G2Gs

- **Chemical reaction:** $\left(\{G_i\}_{i=1}^{N_1}, G_p\right)$ **binary label matrix Y:** $Y \in \{0, 1\}^{n\times n}$
- **Reaction centers :Each atom pair (i.e., bond) in the product Gp**

  employ L -layer R-GCN

  $$H^L = \mathrm{R} - \mathrm{GCN}(G_p), h_{G_p} = \mathrm{Readout}\left(H^L\right).$$

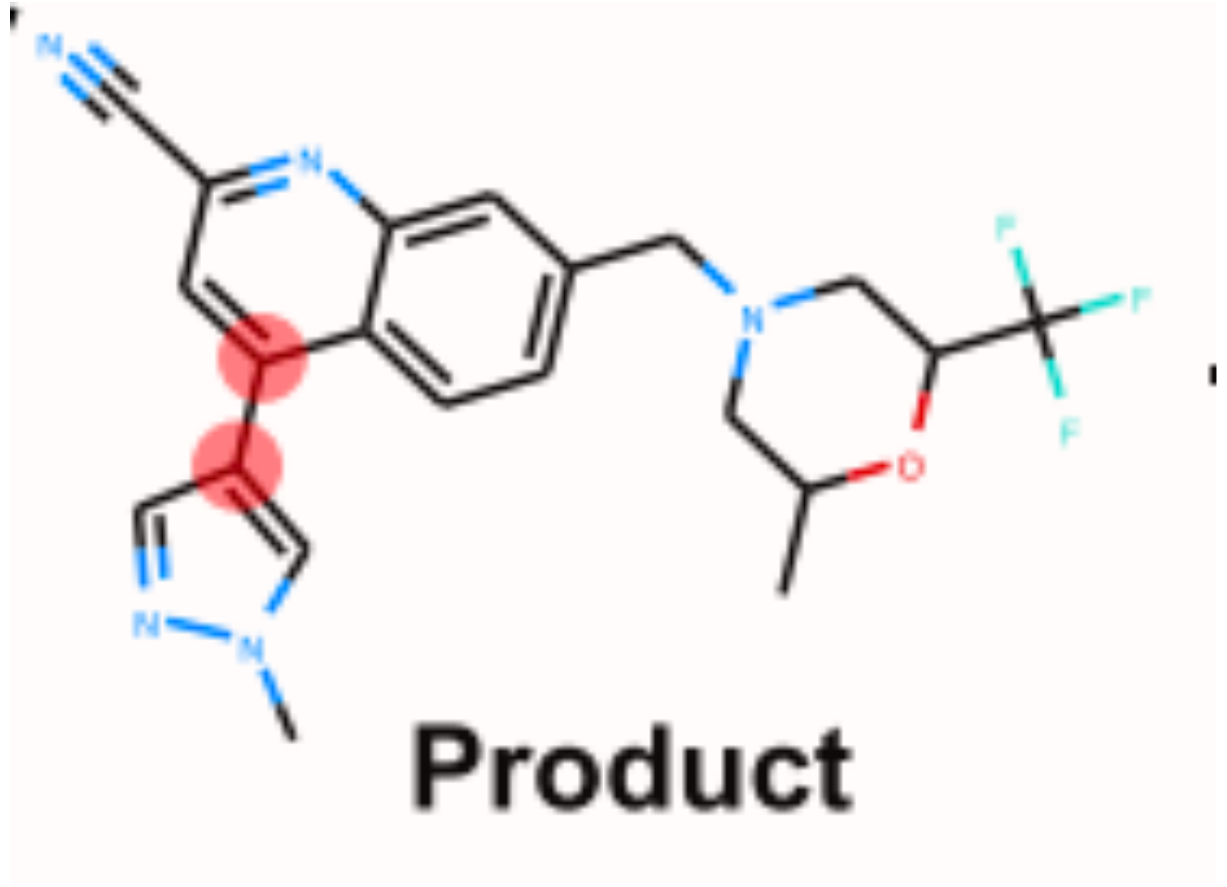
- **Reaction center <-> remote atoms?**

  $$e_{ij} = H_i^L \parallel H_j^L \parallel A_{ij} \parallel h_{G_p}$$

- **Reactivity score** $s_{ij} = \sigma\left(m_r\left(e_{ij}\right)\right)$
- **Learning: maximizing the cross entropy of the binary label matrix Y**

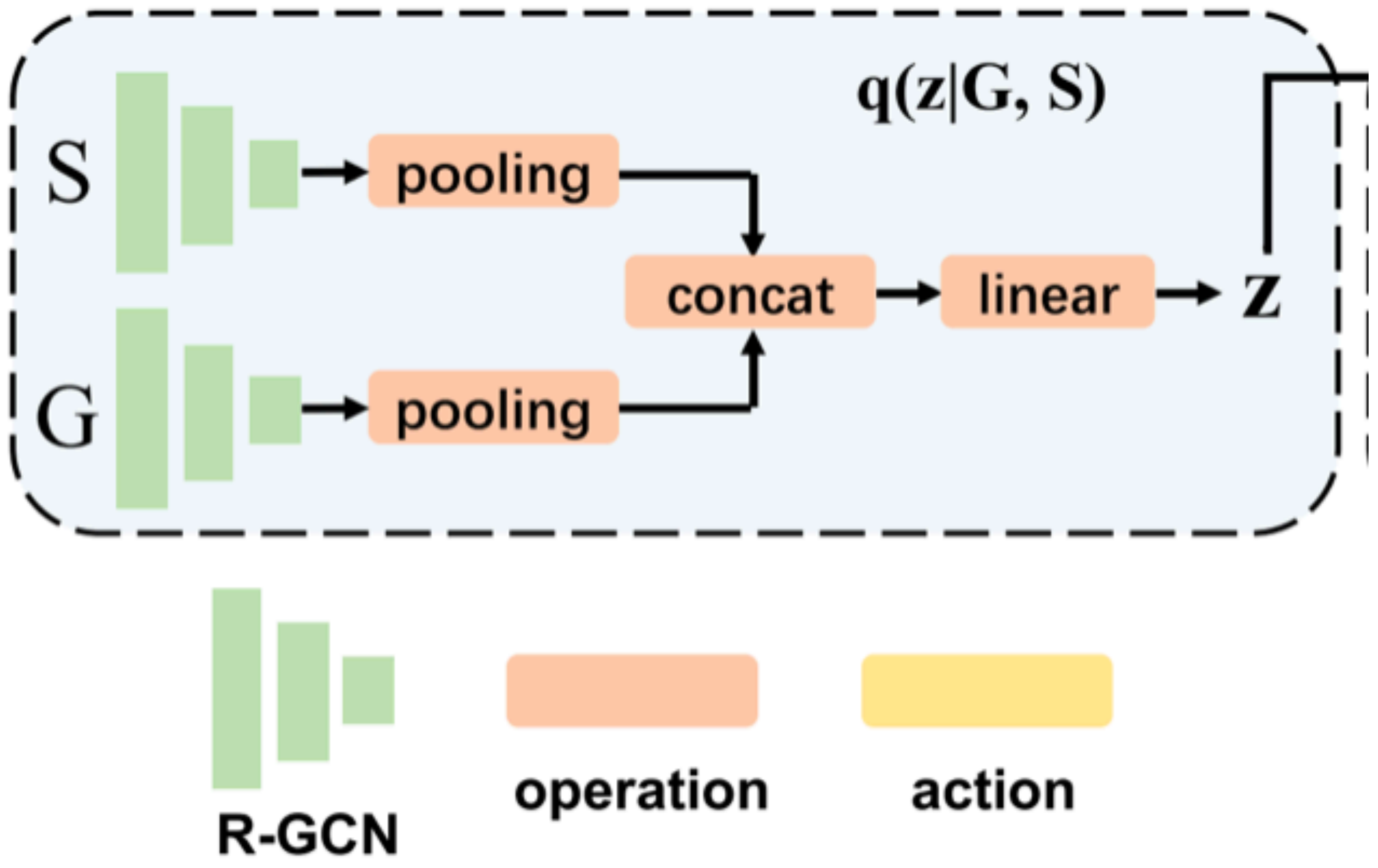
  $$\mathscr{L}_1 = -\sum_r \sum_{i\neq j} \lambda Y_{ij}\log\left(s_{ij}\right) + \left(1 - Y_{ij}\right)\log\left(1 - s_{ij}\right)$$

  Alleviate imbalanced class distributions problem: few reaction center



Reactants Generation via Variational Graph Translation

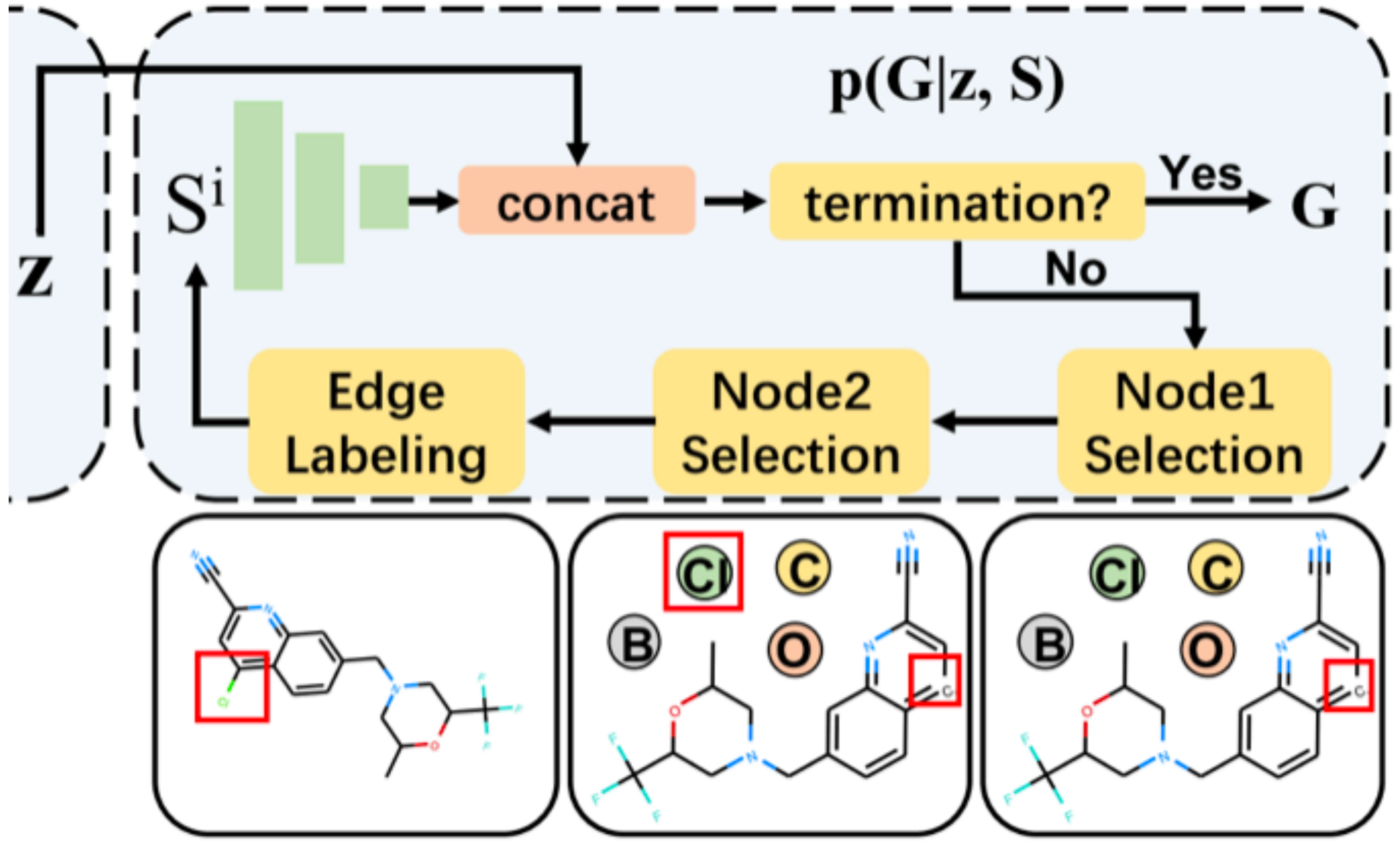
- Disconnect the bonds of the reaction centers in Gp, and treat each connected subgraph in Gp as a synthon. $\{S_i\}_{i=1}^{N_1}$
- Translation pair (S, G)
- Goal: Translates a synthon to a final reactant graph.
  - conditional generative model p(G|S)
- Issue: multi-modality problem. Same synthon can be translated to different reactants
  - low-dimensional latent vector z



Variational Graph Translation: Generative Model

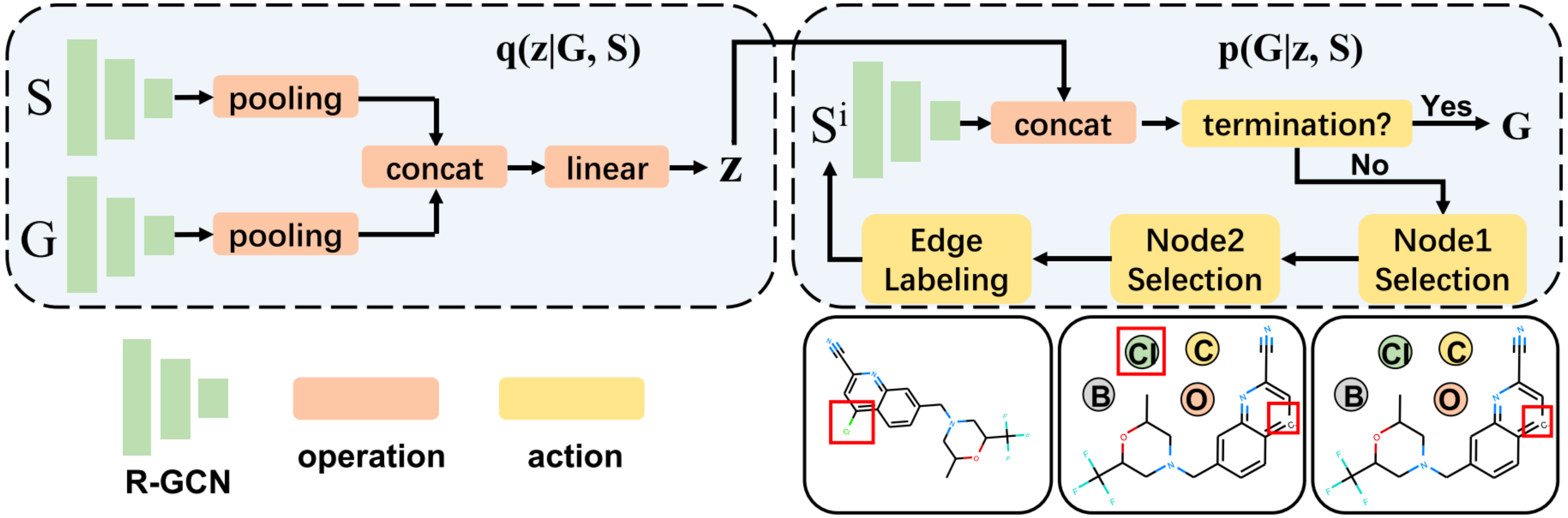
- **The generation of graph G is conditioned on both the S and the latent vector z.**

- synthon-> reactant: $p(G \mid z, S)$
  $\mathcal{T}$: $(a_1, \cdots, a_T), t \in \mathcal{T}$: graph transformation actions
  translate synthons S to target reactants G
  a_t : action. a modification to the graph.
  $p(G \mid z, S)$ ->sampling action sequences from the distribution
  -> joint distribution over $p(t \mid z, S)$.



S^i : apply $a_{1:i}$ to S.
$S^0 = S$; $p(S^i \mid S^{i-1}, z) = p(a_i \mid S^{i-1}, z)$.
Markov Decision Process (MDP): $p(S^i \mid S^{i-1}, z) = p(S^i \mid S^{i-1}, \cdots, S^0, z)$.
Graph translation model: $p(t \mid z, S) = p(a_{1:T} \mid z, S) = \prod_{i=1}^{T} p(a_i \mid z, S^{i-1})$

Variational Graph Translation: Definition of an action



number of atom types: m

$a_i = \left(a_i^1, a_i^2, a_i^3, a_i^4\right)$

$a_i^1 \in \{0,1\}^2$ predicts the termination of the graph translation procedure;

$a_i^2 \in \{0,1\}^n$ indicates the first node to focus;

$a_i^3 \in \{0,1\}^{n+m}$ indicates the second node to focus;

$a_i^4 \in \{0,1\}^b$ predicts the type of bond between two nodes.

Variational Graph Translation: Three parts of distribution $p\left(a_i \mid z, S^{i-1}\right)$

1) Termination Prediction:

$$H=\mathcal{R}\left(S^{i-1}\right), h_S=\operatorname{Readout}(H)$$

$$p\left(a_i^1 \mid z, S^{i-1}\right)=\tau\left(m_t\left(h_S, z\right)\right)$$

2) Nodes Selection:

add the set of possible atoms $\{v_1, \cdots, v_m\}: V=\bigcup_{i=1}^m v_i$.

$$\tilde{S}^{i-1}=S^{i-1} \bigcup V.$$

first node <- S^{i-1}, second node <- $\tilde{S}^{i-1}$ conditioned on the first node

$$p\left(a_i^2 \mid z, S^{i-1}, a_i^1\right)=\tau\left(\beta_1 \odot m_f\left(\mathcal{R}\left(\tilde{S}^{i-1}\right), z\right)\right)$$

$$a_i^2 \sim p\left(a_i^2 \mid z, S^{i-1}, a_i^1\right)$$

$$p\left(a_i^3 \mid z, S^{i-1}, a_i^{1:2}\right)=\tau\left(\beta_2 \odot m_s\left(\mathcal{R}\left(\tilde{S}^{i-1}\right), z, a_i^2\right)\right)$$

$$a_i^3 \sim p\left(a_i^3 \mid z, S^{i-1}, a_i^{1:2}\right)$$

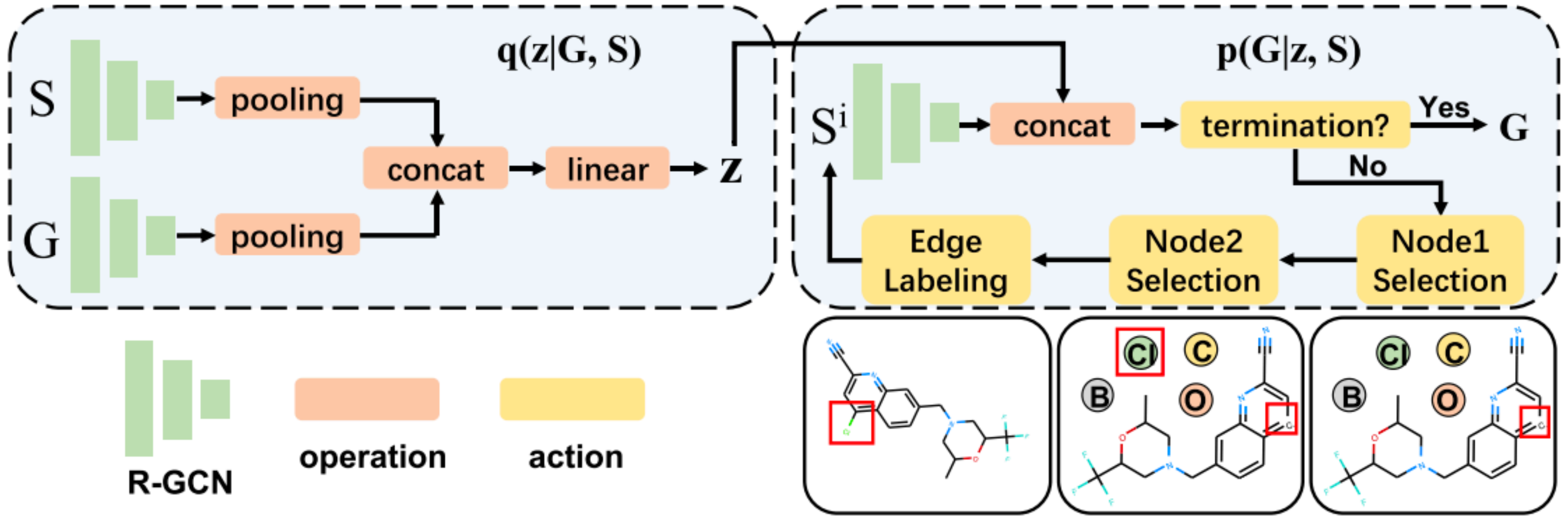
β_1 and β_2: masks to zero out the probability of certain atoms being selected.
only the second node can be selected from V

3) Edge Labeling

$$p\left(a_i^4 \mid z, S^{i-1}, a_i^{1:3}\right)=\tau\left(m_e\left(\mathcal{R}\left(\tilde{S}^{i-1}\right), z, a_i^{2:3}\right)\right)$$

$$a_i^4 \sim P\left(a_i^4 \mid z, S^{i-1}, a_i^{1:3}\right)$$

Enumerating all possible graph transformation sequences that translate S to G :
$P(G \mid z, S)=\sum_{t \in \mathcal{T}} P(t \mid z, S)$



Variational Graph Translation: Learning

maximize $\log P(G \mid S)$
Issue: marginalizing the latent variable z

$$\mu = m_\mu(h_G \| h_S)$$
$$\log \sigma^2 = m_\sigma(h_G \| h_S)$$
$$q(z \mid G, S) = \mathcal{N}\left(z \mid \mu, \operatorname{diag}\left(\sigma^2\right)\right)$$

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

Let $x_1, \ldots, x_n \in \mathbb{R}$ and let $a_1, \ldots, a_n \geq 0$ satisfy $a_1 + \cdots + a_n = 1$. Then

If F is a concave function, we have:

$$F(a_1 x_1 + \cdots + a_n x_n) \geq a_1 F(x_1) + \cdots + a_n F(x_n)$$

The evidence lower bound (ELBO):

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{z \sim q}[\log P(G \mid z, S)] - \mathrm{KL}[q(z \mid G, S) \| p(z \mid S)]$$

$\mathrm{KL}[q(\cdot) \| p(\cdot)]$: Kullback-Leibler divergence
prior $p(z \mid S)$: standard Gaussian $\mathcal{N}(z \mid 0, I)$.

computation of $\log P(G \mid z, S)$-> expensive
Jensen's inequality:

$$\begin{aligned} \log P(G \mid z, S) &= \log \sum_{t \in \mathcal{T}} P(t \mid z, S) \\ &\geq \log |t| + \frac{1}{|t|} \sum_{t \in \mathcal{T}} \log P(t \mid z, S) \end{aligned}$$

$|t|$: the number of different action traces

Variational Graph Translation: Generation

beam search： size: k

For the graph generation in the i^{th} step, maintain a candidate set $\mathcal{S} = \left\{S^{i,j}\right\}_{j=1}^{k}$

At the i^{th} transformation step:

1) calculate the probabilities of all possible actions and sort them.

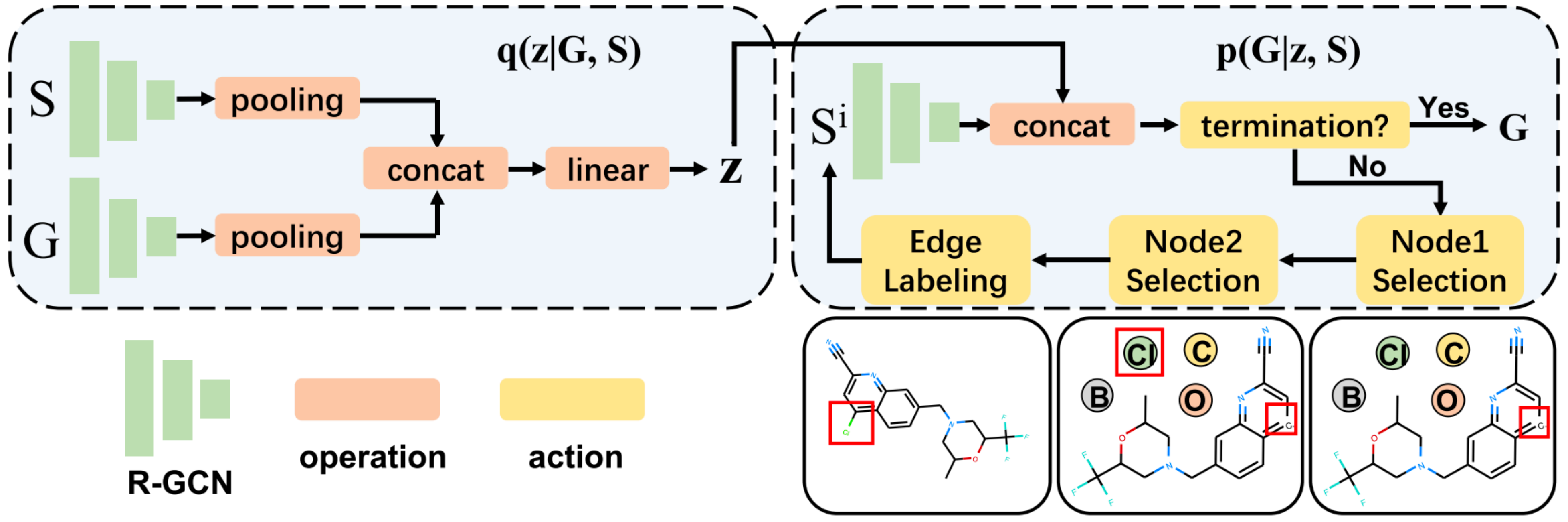
2) select top k ranked valid actions for each candidate graph $S^{i-1,j}$ in $\mathcal{S}$.

3) top k graphs among all the generated 2 graphs -> candidates for the next $i+1)^{th}$ transformation step.

beam search stop when :

1) if i reaches the predefined maximum transformation step

2) a_i^1 indicates a termination.



Experiments

- **Experiment Setups**
  - **Benchmark dataset USPTO-50K, containing 50k atom-mapped reactions**
  - **Evaluation metrics: top-k exact match (based on canonical SMILES) accuracy**

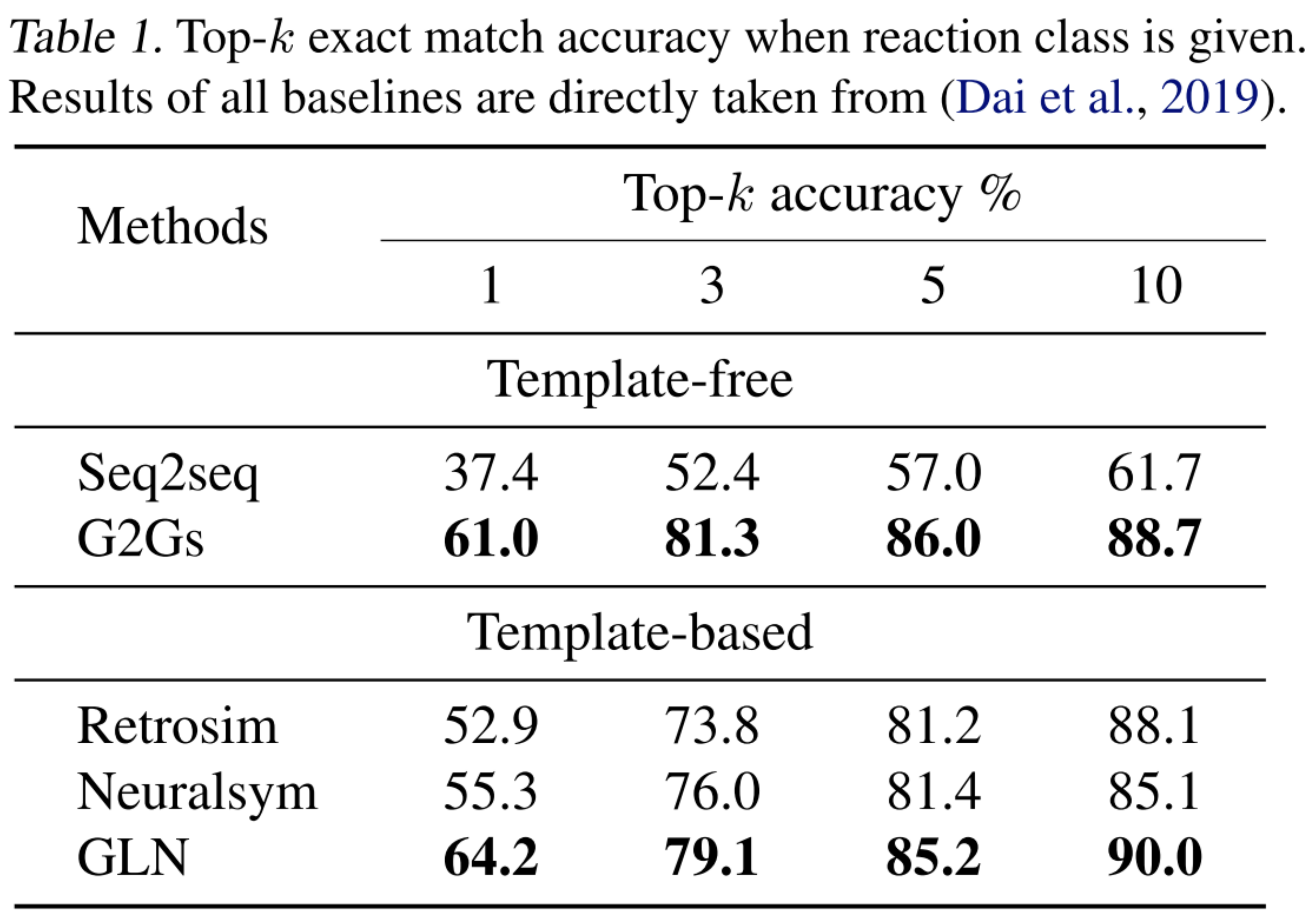


Table 1. Top-k exact match accuracy when reaction class is given. Results of all baselines are directly taken from (Dai et al., 2019).

Methods	Top-k accuracy %			
	1	3	5	10
Template-free				
Seq2seq	37.4	52.4	57.0	61.7
G2Gs	**61.0**	**81.3**	**86.0**	**88.7**
Template-based				
Retrosim	52.9	73.8	81.2	88.1
Neuralsym	55.3	76.0	81.4	85.1
GLN	**64.2**	**79.1**	**85.2**	**90.0**

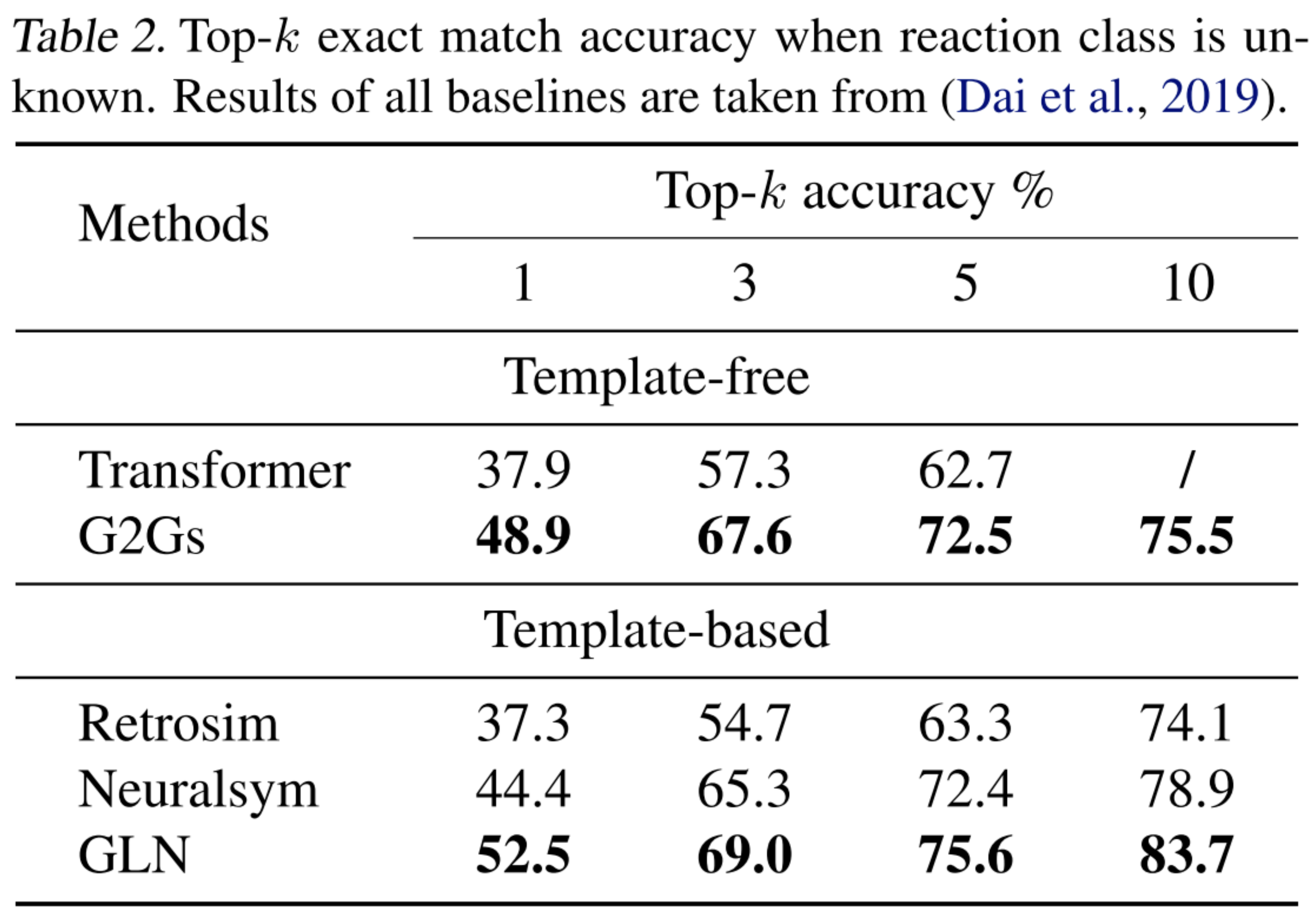


Table 2. Top-k exact match accuracy when reaction class is unknown. Results of all baselines are taken from (Dai et al., 2019).

Methods	Top-k accuracy %			
	1	3	5	10
Template-free				
Transformer	37.9	57.3	62.7	/
G2Gs	**48.9**	**67.6**	**72.5**	**75.5**
Template-based				
Retrosim	37.3	54.7	63.3	74.1
Neuralsym	44.4	65.3	72.4	78.9
GLN	**52.5**	**69.0**	**75.6**	**83.7**

References

- Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P. Learning deep generative models of graphs. arXiv preprint arXiv:1803.03324, 2018.
- Shi, C., Xu, M., Guo, H., Zhang, M., & Tang, J. (2020). A graph to graphs framework for retrosynthesis prediction. In *37th International Conference on Machine Learning, ICML 2020* (Vol. PartF168147-12, pp. 8777–8786). International Machine Learning Society (IMLS).
- milton, W., Ying, Z., and Leskovec, J. Inductive repre- sentation learning on large graphs. In Advances in Neural Information Processing Systems, pp. 1024–1034, 2017.
- Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In European Semantic Web Conference, pp. 593–607. Springer, 2018.