

1 We thank the reviewers for their insightful comments, which we will incorporate into the revised version. We first
2 address some common concerns raised by the reviewers by providing additional experiment results.

3 Design choices of graph neural networks (GNN)

4 Our GLN provides a general graphical model to retrosynthesis problem, which is compatible with many reasonable
5 choices of the representation of graphs. In addition to `structure2vec` with 3 layers (s2v-3) we used in the paper, we
6 provide more ablation studies using different widely used GNNs and different number of "message-passing" layers.

	s2v-3	GGNN	MPNN	GIN	ECFP	s2v-0	s2v-1	s2v-2	retrosim	neuralsym	GLN	
top-1	52.6	51.6	50.4	51.8	51.9	40.7	47.0	51.3	top-1	32.8	35.8	39.3
top-10	83.1	81.8	83.2	83.3	81.5	78.1	80.4	82.2	top-10	56.1	60.8	63.7

Table 1: Ablation study on schneider-50k with different representations. Table 2: Top-k accuracy on USPTO-all.

7 The rationale behind the choices are: 1) the GNNs should be able to take both atom and bond features into consideration;
8 2) according to [1], the family of message-passing GNNs should have similar representation power as WL graph
9 isomorphism check at best. We adopt the s2v in our paper since it satisfies these requirements. Meanwhile, it comes
10 with efficient c++ binding of RDKit. We will elaborate on the details in our revision.

11 We use 2 layers of GNN by default, or use $-k$ after the name in Table 1 to denote k -layer design. We can see that most
12 variations of GNNs can achieve similar performances with enough number of message-passing like propagations.

13 **Reviewer 1 Q1: validate the model on additional real and/or simulated datasets:** We added experiments on a
14 larger USPTO dataset of $\sim 1\text{M}$ reactions (after expanding multi-product reactions and removing duplicates), divided
15 into train/valid/test sets of 800k/100k/100k. The results are presented in Table 2. Despite the noisiness of the full
16 USPTO set relative to the schneider-50k subset, our method still outperforms the two best baselines in top- k accuracies.

17 **Q2: design choices in section 4:** Besides the choice of GNN in Table 1, we also compare the choices of v_1, v_2 and
18 w_2 . Basically all these functions are comparing the compatibility of two vectors \vec{x}, \vec{y} . In the paper, we simply used
19 inner-product $\vec{x}^\top \vec{y}$. Here we also studied $MLP([\vec{x}, \vec{y}])$ and bilinear $\vec{x}^\top A \vec{y}$. For top-1, the inner-prod, MLP and bilinear
20 gets 52.6, 52.7 and 53.5, respectively. So our GLN could be further improved with better design choices.

21 We emphasize that the proposed GLN is general enough which is compatible with other parametrizations. We will
22 make notations and definitions clearer in the paper, and also refine the text for readability.

23 Reviewer 2 Q1: where is the origin of the reactant set \mathcal{R} :

24 We first clarify that we don't limit the reactants to be known in training/test set; this method is able to plan multi-step
25 synthetic routes by recursively making single-step predictions. As in Eq (5), we are modeling the distribution of \mathcal{R} given
26 template T and target product O . With the help of $T = o^T \rightarrow r_1^T + \dots + r_N^T$, we can generate all possible reactants in
27 entire molecule space that satisfy the logic rules. This procedure works as follows: **1)** Enumerate all matches between
28 subgraph pattern o^T and target product O . **2)** Instantiate a copy of the reactant atoms according to r_1^T, \dots, r_N^T for each
29 match. **3)** Copy over all of the connected atoms and atom properties from O .

30 This process is a routine in most cheminformatics packages (we use RDKit's `runReactants`). As mentioned in Line
31 209 in main paper, empirically the # candidate sets is 10 on average, which is not so expensive.

32 **Reviewer 3:** We would first clarify that the appendix file was submitted as part of the supplementary.

33 **Q1: comparison with MLN:** We briefly summarize here, further details can be found in Appendix A. While our GLN
34 and MLN are related—both combine logic rules with graphical models—there are some significant differences:

- 35 • GLN is a directed graphical model while MLN is undirected.
- 36 • MLN treats the predicates of logic rules as latent variables, and the inference task is to get the posterior of them.
37 While in GLN, the task is the structured prediction, and the predicates are implemented with subgraph matching.
- 38 • Due to the above two, GLN can be implemented with efficient hierarchical sampling. However for MLN, generally
39 the expensive MCMC in combinatorial space is needed for both training and inference.

40 Q2: "better to have task agnostic desc...unclear if this is simply an application paper or a truly new method."

41 We are targeting a fundamental problem in ML community, *i.e.*, how to learn distributions on combinatorial discrete
42 variables. The proposed GLN exploits the logic rules to avoid the expensive MCMC sampling over the combinatorial
43 space whose mixing time can be extremely slow. We used the chemical retrosynthesis as our task of focus, which is
44 challenging due to the exact the reason that the modeling objects are discrete and combinatorial. Meanwhile, this task is
45 urgent in need and has been attracting more attentions recently both in ML community and chemistry.

46 We will fix the inline equations and refine the manuscript based on your suggestions.

47 [1] Xu *et.al*, How Powerful are Graph Neural Networks? *ICLR 2019*