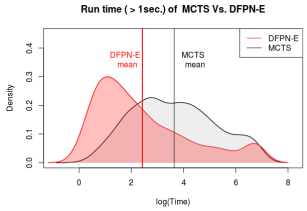
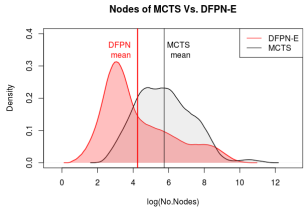
Method	MCTS	DFPN	DFPN$_H$	DFPN-1	DFPN-E
Runtime (s)	18,552	20,133	30,822	46,542	**5,654**
Node expansion	184,347	730,241	668,421	460,738	**68,719**
Num solved	**852**	770	619	691	842
Longest pathway	21	2,000	455	**8**	19
Average pathway	5.58	227.42	21.09	**3.59**	5.72





Thank you for your constructive comments. Please find our response below.

To reviewer 1. We are in the process of releasing our source code, but are still waiting for internal approval. We agree on the missing details about our algorithm in lines 254-266 and will add the missing details in the revised version. Regarding our enhancements to MCTS of Segler et al.: assuming two reactants B and C are generated from a product A by applying a reaction rule in a retrosynthetic manner, their state always includes both B and C. In contrast, our MCTS regards it as an AND node with two OR children B and C. That way the choice of B and C is more dynamic, and based on which one is more promising to prove. The remaining parts of the algorithms (e.g., template extraction) are the same standard algorithms. The data is split based on the date of filing as described in line 236, "The training set contains 681,866 reactions in patents filed before ones in the test set."

The table and figures int this rebuttal present the search performance for the instances with best known pathway lengths ≥ 3. These results are confirming claims made based on Table 1 and Fig. 4 in the paper. There are 897 instances in total and 483 instances are solved by all methods. Furthermore, *there are 2 instances with a pathway length of 2 which are not solved by MCTS but solved by DFPN-E*, which shows that even finding extremely short pathways can fail if MCTS continues examining the wrong branches due to slow node expansion rates and a large branching factors. We will incorporate the specific change requests in the revised version.

In the paper we did not intend to say that there are more uni-molecular reactions than bi-molecular reactions and we are updating it in the revised version. These statistics result from writing the reaction templates in the reaction direction of large molecules "reacting" into smaller molecules to facilitate the search from a usually larger target molecule towards the usually smaller starting materials.

To reviewer 2. Once MCTS/DFPN prove that a subtree of a node is a win/loss, that subtree is no longer examined. For MCTS the partially solved states of MCTS by Segler et al. are prioritized and updated, based on their formula and *Heifets' and Jurisica's game solving model*. The game solving model already incorporates the idea of Segler et al. by regarding their state with multiple molecules as one AND node with multiple OR children. Each of the OR children accumulates its own reward. In addition, if a proof tree of a node is a sequence of moves and a MCTS sampling finds that sequence, the node is regarded as a win which no longer needs to be examined either. Furthermore, we evaluated standard rewards for MCTS, but did not include it in the paper because of space constraints and because it performed poorly by only solving 658 instances (table above) with longer pathways than the version we presented in the paper. There are still hard to solve instances that remain unsolved in our benchmark, therefore search speed is one of the next most important criteria to consider to evaluate performance.

To reviewer 3. As reviewer #2 pointed out, achieving better performance with a combination of evaluation function and PNS variants is not trivial. We plan to emphasize it in the revised version. Evaluation functions for DFPN need to be called in a different way than for PNS variants. Both MCTS and DFPN-E can become a choice in chemical synthesis planning and both are important contributions. At present, MCTS has been much more studied than the PNS variants. Our results will allow other researchers to consider both MCTS and PNS as strong candidates in domains with lopsided search spaces like chemical synthesis planning. Without our contributions, they might refrain from investigating PNS, since the standard way of using an evaluation function for PNS results in poor performance, as we demonstrate with this paper.