As pointed out by reviewers, we need to devote more time to highlight our contributions. We add here some aspects that we did not highlight with enough detail:

1. Learning a proposal function is an important contribution to meta-learning.
   (a) Two of the main applications of meta-learning are learning new tasks from few examples and learning to be fast at optimizing new functions. To the best of our knowledge, typical few-shot learning algorithms do not simultaneously learn to optimize the loss function of each task quickly. In contrast, we learn from few examples by doing structure search and speed it up by orders of magnitude by learning a proposal function.
   (b) Usually, few-shot learning algorithms either rely on local search to perform fast optimization or perform more complex, but slow, optimizations; limiting either capacity or speed. By also learning how to optimize, we can have high capacity and fast search.
   (c) During meta-training, the optimization of each task generates a lot of data (such as gradients, intermediate weights or, in our case, structures). However, most meta-learning algorithms only use the final loss performance. Learning our proposal function shows how we can profit from aggregating these other types of information.
2. Batched modular meta-learning enables shared computation between tasks.
   (a) Main idea: modular meta-learning runs the same module weights across all tasks. Therefore, if a module is present in multiple tasks, we can batch all its evaluations into a single one. This gives similar compute gains as GPUs evaluating regular neural networks in batches of examples, but now for batches of tasks as well and leads to a factor of 5 speed-up.
   (b) Coding this parallelization is very easy for many modular meta-learning compositions. For instance, as reviewer 1 points out, parallelizing GNNs is a simple and standard idea. We see this simplicity as a positive aspect.
   (c) Note that gradient-based few-shot learning algorithms cannot do this parallelization, because they change their network weights differently in each task.
3. A model-based approach to Neural Relational Inference.
   (a) We can tackle new types of problems for which we did not explicitly train. We highlight this by finding the trajectory of an unseen particle using only its influence on seen particles, which the original NRI approach can not do.
   (b) We can predict all edges jointly instead of independently of one another. This is important because many graphs have some high-level structure, such as being connected, symmetric or bipartite. Predicting them jointly adds more capacity to the search and makes it easier to unearth these high-level relations.

We will make sure to better balance our related work section to make higher emphasis on non-graph modular networks. We also agree that learning proposal functions has been studied in other fields, such as in particle filters; we thank reviewer 1 for the relevant citations, which we will certainly include.

Given the reviews, we acknowledge the paper would benefit from more clarity explaining the methods and the experiment (like section 5.2 and figure 4). We will expand both sections, if needed moving details (such as section 5.1) to the appendix. With respect to the experiments, we would like to mention the following things:

1. We do not include comparisons with the original modular meta-learning because we expect results to be similar, but training to take about one month. In particular, we believe one of the contributions of this work is making modular meta-learning practical, going from the toy datasets of 150 tasks of the original paper by Alet et al., to 50.000 in this work.
2. None of the baselines, including the original NRI method, can directly tackle the new problem of finding an unseen node via its influence on others. Thanks to our model-based approach we can make the trajectory of the unseen node part of the inner-loop optimization. To show that our results are good, we show that our prediction errors are as good or better than some of the baselines that do know the position of all nodes, showing good generalization beyond the single problem of finding relations.