We warmly thank the reviewers for their careful reading of the paper and for their feedback which will help us to significantly improve the final version. We agree with the reviewers that we can present our work more clearly, that some additional experiments will provide valuable information and that we can give a more accurate perspective about our future work and the current limitations of AlphaNPI.

1 Clarity

We thank the reviewers for their insightful suggestions to improve the clarity of our paper.

In Section 3, we will focus on clarifying the key principles of the algorithm, and the way the hierarchies of programs are exploited. We will add a short description of Algorithms 1 and 2, the MCTS simulation and tree search, to improve readability without necessarily referring to the appendix. Besides, as requested by Reviewer #2 , we have already worked on a new figure that will illustrate better the main mechanisms of AlphaNPI.

We will also add a paragraph to recap with more details the ways the π^{mcts} output is computed and used. In a nutshell, for each observation e, the π^{mcts} vector, computed with AlphaZero via a guided MCTS simulation, corresponds to a distribution over actions. This is assumed to be better than the network output for this same observation, where better means closer to the one generated by an optimal policy. Therefore, for all observations in the generated traces, we minimize a distance between the distribution represented by π^{mcts} and the current network output distribution to adjust the model weights. The exact computation of the π^{mcts} vector can be found in Appendix A.5, which we will improve.

The comments of the reviewers also helped us to realize that the role of M_{prog} could be better explained. We will clarify the training and use of program embeddings, and the slight differences between our work and the original NPI.

Finally, Reviewer #1 comments that, compared to the work of Cai et al., a previous extension of NPI, we have traded strong supervision of one kind for strong supervision of another. However, we would like to clarify that with our method, the sizes of instances are limited but random during training. There is no ordering of instances, as the curriculum uses only the hierarchy of programs. This hierarchy is not explicit in Cai et al. and in Reed & de Freitas, but it could be easily inferred from the observed execution traces. In our work, traces are not given in advance, they are generated via interactions between the system and the environment. Since execution traces somehow require the problem to be already solved, replacing them by a hierarchy of programs is, in our opinion, a significantly weaker supervision requirement. In the final version of the paper, we will make this supervision difference clearer and more explicit.

2 Additional experiments

We agree with Reviewer #3 that training curves will be a useful addition to our paper. We generated a set of curves to include in the final version, which show how the progress on non-elementary programs starts once a good performance has been reached with all the programs of lower level in the hierarchy. It can be observed that, for the BUBBLESORT program, the performance converges towards almost 100% of success after approximately 250 iterations, which corresponds to the training on $250 \times 20 = 5000$ traces. In the original NPI work, the network was trained on 1216 BUBBLESORT execution traces, which represents much more data as each trace contains also sub-traces corresponding to all the executed sub-programs, while in our setting traces are limited to the actions of a single program. Yet, our results exhibit better generalization properties. In future work, we will estimate more precisely the sample efficiency and generalization of AlphaNPI.

Regarding additional baselines, we will include the results of Cai et al. and Xiao et al.

3 Future work

The necessary verifiability of post-conditions is a current limitation of our work, as in some contexts post-conditions may not be easy to check. However, without post-conditions, new sub-programs would have to be defined during training, and the discovery of relevant sub-programs is a very difficult challenge. Removing the requirement of post-conditions while keeping the global program easily interpretable would probably be a very difficult extension to our work. Instead, in immediate future work we intend to remove the need for the hierarchy of program levels, resulting in the need for more exploration and a less immediate curriculum learning strategy. We intend to use curriculum learning based on learning progress (LP) to overcome the need for an explicit hierarchy given in advance. With this strategy, the agent should be able to focus its training on low-level programs which are easier to learn and thus provide high LP before attempting training on higher-level ones.