We thank the reviewers for their constructive feedback. However, we strongly believe that our work has been misinterpreted by the reviewers. We do not claim that we solved the boolean SAT problem, as we stress in the paper. If we did, we would have submitted to an applied SAT conference. We chose NeurIPS because, as we say in the introduction, we are mainly interested in the machine learning (ML) side of the problem. And we believe that our work should be judged based on its two key contributions.

First, our paper is not a statement that Graph-Q-SAT is a state of the art SAT heuristic. It is a proof of concept, and we should not expect state-of-the-art performance right out of the gate. From this perspective, we believe that our paper should be published because it provides a proof-of-concept validation for a very different approach to devising heuristics for SAT solvers. This validation was not trivial to obtain and opens up a lot of new research opportunities, some of which ultimately could improve the state of the art on SAT. But that won't happen if the paper is not published.

Second, our contribution from the ML perspective is answering the question in the title. It is the study of the properties of a value-based RL algorithm arising in a challenging setting of the boolean SAT problem, studying its generalisation and transfer behaviour, and its data efficiency properties. From this perspective, SAT is not more than a gym environment, similar to how the researchers in the field use MuJoCo or Atari – a simplification enabling the study of some abstract problems. We use boolean SAT because it has the features that are hard to find in conventional RL environments:

- Elements of the state/action set are of different dimensions, which is a challenging case for conventional function approximation techniques.
- Structured, object-oriented state-space representation.
- Ability to vary problem sizes without changing the task family, and to change the task family without changing problem sizes.
- Experiments do not take weeks and are easy to iterate on.

Here, we are not interested in the absolute number of iterations per se or the total ratio between VSIDS and Graph-Q-SAT. We use these numbers as a common scale to show the generalisation, transfer and data efficiency properties of our approach. We believe that our contributions are significant for NeurIPS community because:

- we show surprising generalisation properties of a deep RL algorithm that are usually considered brittle and overfitting for the training task;
- we show the data efficiency properties of the algorithm, which pinpoints the attractiveness of RL compared to the supervised learning approach;
- we demonstrate positive transfer behaviour, which is a rare thing in deep RL. Often, the task differences studied in transfer for RL are as small as different reward functions. In case of structured state/action spaces, adding two legs to an agent and showing the score higher than of a random policy is considered an achievement.
- the code of the environment we improve and refactor can spark further development of the algorithms in multi-task RL, curriculum, transfer learning and other subfields of RL dealing with multiple tasks.

We submit that our paper deserves a much higher score at an ML conference and we think that our experiments are sufficiently thorough and easy to replicate.

Response to specific comments In this section, we address some specific comments from the reviewers. We will address the rest in the final version of our paper.

"Adapting the deep Q networks in [22] to learn heuristic for CDCL looks straightforward to me. So the authors should provide a deeper discussion concerning the novel aspects of their approach."

The problem studied in [22] is much easier from the RL perspective than the one considered in our work. The environment dynamics in [22] is straightforward with the next state easily inferred, given the current state and the chosen action. In our case, there are CDCL steps after the decision, and the next state might be totally different from the current one making the problem harder in terms of learning the Q-function. We disagree that adapting DQN to our case is a straightforward task, and our work has to address two nontrivial issues. First, most of the existing insights about DQN are related to its main testbed — Atari 2600. And many pieces of the common knowledge did not hold in our case. For example, the target network, which is considered of utmost importance for DQN, was harmful to us when updated less frequently. The same holds about the memory replay size. We found out that large replay buffer size is destructive in our setting, which was not discussed before in the literature. Second, GNN research, especially using them in deep RL, is a young area. In our work, we use [4], which is more expressive than `structure2vec` used in [22]. The global attribute in [4] can facilitate message passing in case of a bigger graph. Having separate updaters for edges and nodes leads to more powerful representations. And, finally, an edge updater in [4] can learn better pairwise interaction between the sender and the receiver, enabling sending different messages to different nodes.