We thank all the reviewers for their comments and criticisms. In our response we will answer the specific questions raised and attempt to clear up a few points. In the three sections that follow we respond to the comments in each review.

Response to R1. 1. (Local search for SAT reference) In the paper, we reference WalkSAT as the basic local search algorithm for Boolean satisfiability. Also, in Algorithm 1 we give a generic pseudocode of local search for SAT. Proceedings of the recent SAT conferences and SAT competitions include SLS-based solvers more sophisticated than WalkSAT, however most if not all of them already fit in to the pseudocode that we provided.
2. Doing curriculum learning when there is a measure of difficulty is indeed not new, and we use it solely to accelerate training. As Reviewer 3 observes, the main novelty of this work is in the application domain.
3. R1 writes: "The authors do the first step in replacing CDCL SAT Solvers". This is slightly inaccurate in two ways: We are not comparing at all against CDCL solvers, but rather the algorithms that perform stochastic local search (SLS), in fact our approach can be thought of as an SLS algorithm template where the variable selection heuristic is learned per problem class. Also, we would not yet phrase the goal as replacing CDCL solvers, as they are likely to stay popular at least until the domain-specific hardware such as TPUs becomes common enough for approaches such as ours to become significantly faster. Our focus lies in learning specialized heuristics from scratch with minimal supervision.
4. As for evaluation on more difficult instances, those not solved by WalkSAT are out-of-reach for our approach as well. While the practical value of SAT solving is in solving real-world problems, investigations into entirely different ways of approaching the problem holds scientific value, which we pursue with this work. Our current goal is not to beat the state-of-the-art SAT solvers on industrial instances, but to show that a purely learning-based approach can help us create algorithms with fine-grained specialization. Nevertheless, we are optimistic that a similar approach will hold more practical value in the future.

Response to R2. 1. Longer unrolls were not beneficial enough (and sometimes even hurtful) to justify the increased runtime. $p = 1/2$ is mostly arbitrary and not very important for the conclusions we draw. In our final revision we will add ablation studies to the appendix performed on one of the problem distributions.
2. We have an additional explanation in Appendix C, although it does not answer your question. The specific way we do curriculum is to train on a small problem distribution (say D1) for a fixed number of steps while evaluating on a slightly larger distribution (D2). For the next step, we train on D2 beginning with the best model parameters from before (lowest median number of steps on D2) while evaluating on a larger distribution and so on.
3. We did try using a different head as well as an entirely separate network to predict the value baseline. The MDP that we defined is unorthodox in several ways, and many of our intuitions from deep RL actually seemed to fail. We will release our code to allow people to experiment on their own and verify these.
4. We will include WalkSAT results on Figure 4 in our final revision.
5. Limiting the number of formulas seen during training actually accelerated training (with respect to evaluation performance on larger, unseen problems from the same class). We did experiment with never reusing formulas (checking satisfiability has a relatively low cost for the small problems that we use), which had an adverse effect. Note that even when we recycle problems, the path taken (how the assignment changes) is often wildly different. Our hypothesis is that not reusing the formulas makes the "environment" change too much between episodes (since the state space is already large due to the number of possible assignments), hindering learning.

Response to R3. 1. At a first look, survey propagation (SP) may be worth comparing against, but SP on its own is not particularly strong in solving formulas that are not uniform random k-SAT. Indeed, we performed this experiment at the beginning of this work and SP (without decimation) did not converge to a solution on any of the SAT-encoded graph problems. As a result, it is a baseline that is almost too weak for our purposes of sanity-checking the learned heuristic, which is why we went with WalkSAT instead. As for the solution time compared against WalkSAT, it definitely will be of interest to the broader community, but we do not want to be misleading since we did not focus too much on the efficiency of our implementation (especially the graph neural network). Even with current hardware, once the implementation is properly optimized (moving from Python to C as the first step), the real gap in achievable runtime is probably lower than what we will report.
2. (Generalization from rand3) The learned heuristic is expected to exploit the specific structure of the problem class as much as possible. When there is no structure to exploit, we hypothesized that the learned heuristic is generic, because there is no way to "cheat" by exploiting the structure. WalkSAT, for instance, works well for the uniform random formulas with no structure but is also able to solve some problems that exhibit structure (also see YalSAT and probSAT for other examples of this). This is not a formal enough statement, but we think that the relationship between DPLL–SLS is not the same as [learned heuristic on structured]–[learned heuristic on random]. We will reframe this discussion since it seems to lead to confusion.
3. In Figure 3, the red curves correspond to the smaller distributions used for training and the blue curves correspond to the larger ones used for evaluation. For Table 4, the reported numbers are computed by looking only at the flips that are recommended by the heuristics (although if a variable was previously randomly flipped and the heuristic chooses to flip the same variable again then this is counted as an "undo").