We thank the reviewers for their positive feedback and thoughtful suggestions. Overall, the reviewers found this research worthwhile and interesting. R2 and R4 requested further demonstration of the scalability of G2SAT. R3 asked for more ablation studies of our techniques. R4 raised concerns regarding our main motivation, questioned the extrapolation ability of G2SAT, and pointed out some related work. We will carefully revise the final paper to clarify these points.

1. Motivation (R4). R4 questioned the contribution of our technique to the SAT community, thus we clarify as follows. **First**, G2SAT manifests an efficient and general technique to generate interesting SAT benchmarks. While we agree with R4 that "*the benchmark situation has improved over the years*", new interesting benchmarks are still demanded and highly welcomed by the SAT community. For example, new benchmarks, both real and synthetic, are called for during each year's SAT competition. **Second**, G2SAT demonstrates a novel data-driven approach to improve SAT-solving. Although, as R4 correctly pointed out, significant progress in SAT solvers has been made over the past few years, we point out that those improvements result mainly from better hand-crafted heuristics and software engineering. On the other hand, the promising result of an experiment in our paper (line 275-284), where we showed that G2SAT formulas can be used to better tune the hyper-parameters of a SAT solver, suggests the exciting opportunities to improve SAT solvers in a data-driven manner. This direction depends on having a large number of realistic formulas, and we propose G2SAT as one way to obtain such formulas. Therefore, we believe our techniques, in addition to being theoretically interesting, are meaningful to the SAT community. We will be more specific about these points in the revised version.

2. Experiment (R2, R3, R4). We conduct additional experiments to address the reviewers' concerns.

Scalability of G2SAT (R2, R4). It is worth noting that existing deep graph generative models can only generate relatively small graphs, ranging from tens of nodes (GraphVAE), hundreds of nodes (Learning Deep Generative Model of Graphs, GCPN) up to 1,000 nodes (GraphRNN, Graphite, NetGAN). In contrast, the novel design of the G2SAT framework (elaborated on in Section 4.2) enables the generation of graphs an order of magnitude larger than those in previous work. Notably, in our additional experiments, the largest graph we generate has 39,578 nodes and 102,927 edges, which only took 489 seconds (data-processing time excluded) to generate on a single GPU. Figure 1 further shows the time-scaling for both training (from 100k batches of node pairs) and formula generation. We found that G2SAT scales roughly linearly for both tasks with respect to the number of clauses.

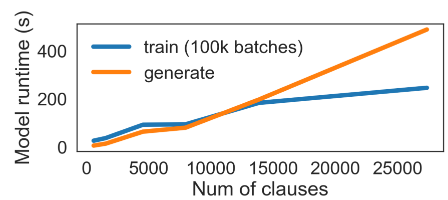

Figure 1: Run time scaling of G2SAT

Extrapolation ability of G2SAT (R4). To address the concern of R4 on whether a trained model can learn to generate SAT instances different from those in the training set, we design an extrapolation experiment as follows. We train on 10 small formulas with 327 to 4,555 clauses, while forcing G2SAT to generate large formulas with 13,028 to 27,360 clauses. Note that none of the baseline methods can accomplish the same task, as they can only mimic a given SAT formula. On the contrary, G2SAT can generate large graphs whose properties are similar to those of the small training graphs, which shows that G2SAT has learned non-trivial properties of real-world SAT problems, and thus can extrapolate beyond the training set. Specifically, the VCG modularity of the large generated formulas is 0.81 ± 0.03, while the modularity of the training formulas is 0.74 ± 0.06.

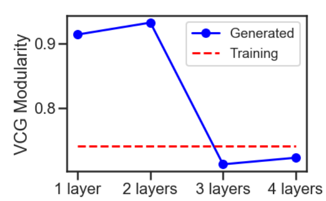

Figure 2: Ablation Study

Ablation study (R3). Figure 2 shows the effect of the number of layers of the GCN neural network model on the modularity of the generated formulas. As the number of layers increases, the average modularity of the generated formulas becomes closer to the training formulas, which indicates that machine learning contributes significantly to the efficacy of G2SAT. The other graph properties that we measured generally follow the same pattern.

3. Related work (R4). We thank R4 for pointing out the related work, a 2-page extended abstract that came out on July 5, which is more than 1 month after the NeurIPS deadline. Our paper differs from that work in at least 4 ways. (1) G2SAT is over bijective LCG representation of SAT, rather than the LIG representation that has ambiguity. (2) G2SAT proposes a novel and scalable bipartite graph generator, while that paper applies an existing model (NetGAN) that can only perturb one given SAT instance. (3) We show that G2SAT-generated formulas resemble the training formulas in SAT-solver performance. (4) We investigate how a SAT generator can potentially help design better SAT solvers. Nevertheless, we recognize the relation of that work to our paper, and will cite that work in the revised version.

4. Code and implementation details (R2, R3). We promise to open-source our code if our paper is accepted. Due to page limits, we did not describe the full details of the model. In summary, we use a standard 3-layer GCN with 32 hidden dimensions in each layer and ReLU activation. We train our model using Adam optimizer with learning rate 0.001, over 10M batches of node pairs for all experiments. We will include these details in the revised version. As requested by R3, we will report the actual total runtime of SAT-solvers on the generated formulas in the revised version.