Thank you all for your reviews and constructive comments. We will revise the manuscript based on your suggestions.

Reviewer #1: ▶ *Add more examples showing that the new GNNs are more expressive than previously considered classes of GNNs.* The finding single leaf problem is the only known problem that does not belong to $\mathcal{P}_{\text{VV}_\text{C}\text{-GNNs}} \setminus \mathcal{P}_{\text{MB-GNNs}}$. It has been a long-standing open problem to find other such problems in the field of distributed local algorithms [10]. If this open problem is solved in the distributed algorithm community in the future, we can give an example thanks to Theorem 1. It should be noted that the approximation of the minimum vertex cover problem provably belongs to $\mathcal{P}_{\text{VV}_\text{C}\text{-GNNs}}$ (Theorem 7) whereas it is not known whether this problem belongs to $\mathcal{P}_{\text{MB-GNNs}}$ or not. ▶ *Another extension of GNNs was proposed in https://arxiv.org/abs/1810.02244 - it would be interesting to compare these two approaches ...* As you pointed out, their approach is orthogonal to ours. For example, k-GNNs cannot solve the finding single leaf problem (Line 229) whereas ours can. Therefore, we can make k-GNNs more powerful using port numbering. Examining the expressive power of k-GNNs with port numbering more precisely is an interesting future work. ▶ *Why authors have selected the Reinforce algorithm for training?* We followed the existing work [4].

Reviewer #2: Important results in this paper are the inapproximability results (e.g., Theorem 4 and 8) rather than the approximability results. The best approximation ratios that GNNs can achieve are far worse than many researchers considered. Moreover, as Reviewer #1 pointed out, the most important contribution is to show a link between GNNs and distributed local algorithms (Theorem 1). These surprising results must have a large impact on the NeurIPS community. ▶ *Definition of "solving" a given combinatorial task seems tricky (L121-122). If my understanding is correct, a GNN model class is considered to be able to solve the task as long as it contains a single model instance that solves the task.* As you pointed out, the definition of solvability is fairly loose in this paper. Therefore, the inapproximability results become extremely strong. It indicates that there exist no model instances that can solve these graph problems, and any elaborated training procedures cannot find any model instance that solve these problems. ▶ *the paper can be strengthened if the authors could provide more insights/explanations for those unsatisfactory ratios.* We show an illustrative example of the minimum dominating set problem in Figure 1. ▶ *I believe experiments are crucial to verify the correctness of the theorems...* We gave a mathematical proof for each theorem, which verifies the correctness of the theorem more rigorously than any empirical experiments.

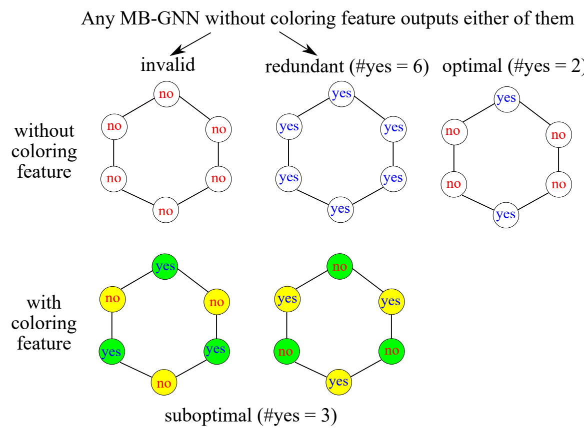

Figure 1: **Minimum Dominating Set Problem:** GNNs output invalid or redundant solutions without coloring because the input graph is symmetrical. With coloring, GNNs can distinguish adjacent nodes, but cannot identify the global structure. Thus GNNs output suboptimal solutions.

Reviewer #3: ▶ *a DNN runs in polynomial time and we have inapproximability results for polynomial-time algorithms, we already know that it cannot beat known approximation algorithms in terms of approximation ratio.* We showed the approximation ratios of GNNs are far worse than known inapproximablity results for polynomial time algorithms. For example, there exists a $(\mathcal{H}_{\Delta+1} - \frac{1}{2})$-approximation algorithm for the minimum dominating set problem [A], where $\mathcal{H}_i$ is the i-th harmonic number. Considering $\mathcal{H}_{\Delta+1} = O(\log \Delta)$, the best approximation ratio $(\Delta + 1)$ of GNNs is far worse than this algorithm. Moreover, GNNs cannot solve even an easy instance as Figure 1 shows. This fact has been overlooked in the GNN community. ▶ *I guess the reason why people try to use DNNs for combinatorial problems is its empirical performance. ... Why do we want to identify the best approximation ratio we can obtain with a DNN when we know that it won't be better than those of known approximation algorithms?* Indeed, GNNs are popular for its empirical performance. However, we consider providing a theoretical guarantee is also important. For example, when one determines the schedule of product releases using a combinatorial solver without any theoretical guarantee, it may output a far worse solution than the optimal solution and causes an enormous loss. We proved GNN cannot use such applications that need a theoretical guarantee. ▶ *in Theorem 7, can we use a single choice of parameters to achieve 2-approximation or we have to change parameters depending on the input graph?* In all theorems, we use a single choice of parameters to achieve the approximation ratios (see Line 114 and 121). ▶ *Line 8: As "GNN" is not a well-defined term, it does not make much sense to say "no GNN can perform better than ..."* We intended GNN meant MB-GNN, which include most of GNNs in the literature (Line 155 - 161). We will clarify it.

References: [A] Miroslav Chlebík and Janka Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Inf. Comput.,* 2008.