We are very thankful to all reviewers for their time and valuable comments. In response, we propose to use the additional page allowed in the camera-ready version to: **I)** expand our literature review to include other related approaches; **II)** incorporate more details about our dataset generation procedure, as well as training and inference times; **III)** include a "Discussion" section to answer the questions raised by each reviewer; **IV)** add an additional problem, *maximum independent set*, to our benchmark; **V)** fix the missing value in Table 2 which is confusing, and should be read as N/A (not available), since FSB did not solve any instance in that case.

Answer to reviewer #3: I) We agree that "lots of works have used GCNN for different combinatorial optimization problems", but we emphasize that combinatorial optimization can mean many things. We are the first to use GCNNs (as well as behavioral cloning) for *exact* solving, namely by using them to model branching policies: this is novel and not trivial to do, since one must also come up with a way to encode branch-and-bound states appropriately. It is also an important problem to tackle, since exact optimization is ubiquitous in industry. Furthermore, we provide strong, reproducible empirical evidence that this bipartite GCNN is better suited than previously proposed models for this important task. **II)** Regarding the policy quality vs inference speed: we agree both aspects are important, and comparing number of nodes vs total running time in Table 2 already illustrates this trade-off. We propose to clarify this question explicitly in the text, especially regarding time-consuming methods like strong branching. **III)** Regarding the RL part, we understand the issue, and we propose to keep the MDP formulation but remove any references to rewards, such as Eq. 3. **IV)** Regarding interpretability of our learned model, that is a hard question to answer: how to define an easy/hard branching case? We measured the entropy of both FSB and our learned policy and find that they strongly correlate, indicating that "obvious" choices for FSB seem to be also "obvious" for our model. We will include this result in the supplementary materials – we hope this can provide more insights on the behavior of the trained policy, as requested.

Answer to reviewer #4: We agree that our benchmark is artificial, and using real-world instances would bring value. Such datasets could be collected, e.g. similar unit commitment problems are solved every 15 minutes by electrical power production companies: however, no public one is known to us. At the same time, it is common to evaluate OR algorithms on artificial problems, and we have used for our benchmark peer-reviewed, published instance generation algorithms for well-recognized problems, which were designed to model real-world applications. Regarding additional structure, since our original submission we have run experiments on a fourth problem, maximum independent set, and here again the GCNN dominates (see below). We intend to include those new results in the final version of the paper.

	Easy			Medium			Hard		
Model	Time	Wins	Nodes	Time	Wins	Nodes	Time	Wins	Nodes
FSB	34.82 ±27.2%	5 / 100	7 ±35.7%	2434.80 ±15.8%	0 / 52	67 ±37.3%	3600.00 ± 0.0%	0 / 0	n/a ± n/a %
RPB	12.01 ±14.2%	3 / 100	**20** ±56.0%	175.00 ±20.0%	28 / 100	1292 ±29.1%	2759.82 ±11.3%	11 / 34	8156 ±34.4%
TREES	11.77 ±19.9%	4 / 100	79 ±47.6%	1691.76 ±37.4%	0 / 44	9441 ±58.1%	3600.03 ± 0.0%	0 / 0	n/a ± n/a %
SVM	9.70 ±13.1%	9 / 100	43 ±33.6%	434.34 ±37.4%	0 / 80	**867** ±36.0%	3499.30 ± 1.8%	0 / 4	10 256 ±12.2%
LMART	8.36 ±11.3%	18 / 100	48 ±39.6%	318.38 ±33.0%	6 / 84	1042 ±47.4%	3493.27 ± 2.5%	0 / 3	15 368 ±48.5%
GCNN	**7.81** ± 9.0%	**61** / 100	38 ±31.2%	**149.12** ±54.8%	**66** / 93	955 ±59.0%	**2281.58** ±29.1%	**28** / 32	**5070** ±79.5%

Maximum Independent Set

Answer to reviewer #5: I) We agree that we should discuss references [a-c] in our literature review. We propose to expand this section and better compare our approach to those related works. **II)** Regarding feature engineering and embeddings, we agree and will clarify this point. **III)** Regarding experimental details, we agree and will add a detailed description of our dataset generation procedure, the total number of variables/nodes/instances represented in each dataset, the evaluation metrics and training/inference times of each method. **IV)** Regarding evaluation on even bigger instances, it seems that GCNN (and other ML methods) fail to generalize at some point, e.g., training on easy (100 items x 500 bids) combinatorial auction instances does not generalize to huge (400 x 2000) instances, however re-training our model on medium (200 x 1000) does. We propose to add this discussion in the paper. **V)** Regarding evaluation set size, we agree 20 instances x 5 seeds is small, but benchmarking already takes more than a hundred hours of computing time. Unfortunately, we do not currently have the resources to do more. **V)** Regarding pseudocost filtering in [29], thank you for pointing this out (it was a hard catch from the text). Indeed, by filtering the top-10 variables for training and testing we were able to train competing models using the whole dataset, but the performance was 1.15-1.5x worse in time, and 2-8x worse in number of nodes. This might be because PC scores require an initialization with SB at the beginning of the solving process, which is present in [29] due to their hybrid strategy, while in our context we are trying to replace SB completely. PC scores are then very poor, especially at the root of the tree where branching decisions are most crucial. Nonetheless, we can include such results in the supplementary materials. **VI)** Regarding the unfair reduced training size for other methods, that is a good point, and we propose to add a discussion in the paper. However, we believe that the inability to exploit huge amounts of data is an intrinsic limitation of other methods (e.g. it would require >500GB RAM for SVMrank, >1 week training for LambdaMART), and is not a good reason for limiting ourselves to the same amount of data. This being said, we also ran preliminary experiments using small datasets (10x smaller), and our performance was only slightly worse. Therefore we believe that the superiority of GCNN cannot come from the dataset size alone.

Again, we thank the reviewers for their time, and hope our proposed modifications will answer to their concerns.