We thank all reviewers for their time and thoughtful comments.

1 Response to Reviewer 1

It would also be nice to see some discussion of how the given procedure would have to be modified (or how difficult such a modification should be expected to be) to incorporate some of the other bounds discussed in the paper: Writing the code to incorporate additional bounds is actually very easy. For instance, one can define a new upper bounding function that returns the minimum of two other upper bounding functions. The difficulty is in finding other upper bounding functions that are (1) tighter than the bound we use, (2) efficient to compute, and (3) empirically require few calls to $Refine$. We did not find other upper bounds with implementations that satisfied conditions (1) and (2). We would be happy to include a discussion of bounds that we believe are promising to explore. Particularly, we think that the Bethe upper bound could yield improvements on sparse matrices (as in Fig. 3), which could be useful for multi-target tracking where the matrix of interest is frequently sparse. Unfortunately, the fast c implementation of the bound provided by [1] is numerically unstable for sparse matrices with 0 entries and the numerically stable matlab implementation is prohibitively slow. This difficulty could be overcome by rewriting an efficient implementation. Another potentially interesting bound to explore is the "sharpened" version of the bound we use, described in [4]. This bound is computed by solving an optimization problem, but unfortunately we do not know of an efficient solution. Using gradient descent, we found that this sharpened bound can be significantly tighter than the one we use, but this approach is too computationally expensive. We believe an efficient solution may exist, but have not found it.

It would be nice to see some comparison of the estimates of the values of the previous Law sampling method in addition to the runtime comparison: Both our method and Law's method provide exact samples, so we would obtain the same estimates up to random effects.

We appreciate the additional suggestions to improve our paper.

2 Response to Reviewer 2

My only criticism, really, is that the approach seems to be much more applicable than just a strategy for computing the permanent and it would have been nice to see some experiments on different types of partition function estimation tasks: We believe that implementing AdaPart to work for general graphical models is indeed an interesting direction for future work. We would like to explore implementation using a variety of general upper bounds [5, 2, 3].

3 Response to Reviewer 3

Although the algorithm was written in a general way, i.e., not tied to the permanent problem and particular choice of bound, only a single choice of upper bound from Soules is considered for the permanent problem: Please refer to our discussion in response to reviewer 1.

Thank you for your suggestions, we will clarify Figure 1 and our use of the term "nested upper bound."

References

[1] B. Huang and T. Jebara. Approximating the permanent with belief propagation. *arXiv preprint arXiv:0908.1769*, 2009.

[2] Q. Liu and A. T. Ihler. Bounding the partition function using holder's inequality. In *ICML*, 2011.

[3] Q. Lou, R. Dechter, and A. Ihler. Anytime anyspace and/or search for bounding the partition function. In *AAAI*, 2017.

[4] G. W. Soules. Permanental bounds for nonnegative matrices via decomposition. *Linear algebra and its applications*, 394:73–89, 2005.

[5] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching. In *AISTATS*, 2003.