1 Rebuttal

1.1 Response to skYN

We thank the reviewer for their comments.

1.2 Response to 7JBT

Regarding the questions about linear behavior in \( n \) and the right side of the figures with computation time, we note that our claim is that the behavior would be linear in numerical tolerance. We have revised our manuscript to be more clear that the algorithm is linear in numerical tolerance. We have not characterized the scaling of the algorithm in \( n \), but as you point out the numerical results suggest that it is not linear. However, we believe this does not detract from the practical utility of our algorithm because tensor completion is often used in the low \( n \) setting.

Regarding numerical evidence of scalability and efficiency, we do note that our experiments scale to tensors with 10^8 entries, which is comparable to the problem sizes used in research papers to show scalability for matrix completion.

Regarding your surprise that an algorithm that performs a local optimization by flipping single variables one at a time is able to reach the optimal solution in general: We were also surprised by the empirical success of the simple, heuristic separation oracle, and we have checked our code carefully to confirm. Some of this can be explained by the nature of the BCG algorithm: the separation problem does NOT have to be solved to optimality – any feasible solution crossing the zero objective value threshold suffices. Therefore, especially in early iterations, it is relatively easy to find a separating cut.

1.3 Response to 4C1A

Question 1: We have changed the language to say "achieve substantially closer to the information-theoretic rate" for these past works. Two of the cited methods do not have corresponding results, but we believe an analysis would show this to be the case.

Question 2: We have added a remark to clarify this point.

Question 3: We use two oracles in our algorithm. The oracle that gets used the most is implemented as a heuristic, and that is backed up by an oracle that uses an exact MIP solver. The MIP solver is, naturally, worst-case exponential-time since the problem is NP-Hard. Nonetheless our computational experiments demonstrate highly-scalable and data-efficient practical performance. This is because the heuristic oracle is called much more often. The code we include with our paper keeps count of
the number of times the MIP-based oracle is called, and it gets called a small number of times when solving problems.

Question 4: We have not focused on a polynomial-time special case, as our contribution focuses on a practical, data-efficient numerical method for the general case. Now, we can show that the separation oracle (and consequently the overall algorithm) can be executed in polynomial-time for the special case of matrix completion. However, in practice it is likely that more specialized methods would still preferred for nonnegative matrix completion. Establishing polynomial-time subclasses of tensor completion via the complexity of the separation problem is an interesting topic, and certainly something we would like to consider in future work.

Question 5: Regarding the difference between $\psi$ and $\hat{\psi}$ in Prop. 4.1 and Cor 4.3, we have corrected a minor type on the left side of the equation in Prop 4.1 where the “hat” was missing. This should now clarify the difference.

Question 7: There was a typo, and little-o should be big-o. We have corrected this typo. Regarding the comment that in tensor completion we have $2^n > \pi > \sqrt{n}$, this may not be true in the noisy setting where you could potentially observe several different noisy measurements of each tensor entry: Tensor completion would give an improvement over simply averaging each observed entry.

Question 9: We have revised the wording to “…practical numerical computation….”

Question 10: We report normalized MSE, and the theoretical results are for MSE. Because of how the instances are constructed, each entry in the “true” tensor lie between zero and one. Hence, normalized MSE is larger than MSE. Also, the algorithm uses a specified numerical tolerance the determine when to terminate the algorithm, and we note that the number of oracle calls is linear in this numerical tolerance. However, the finite numerical tolerance introduces some amount of error in the reconstruction. Convergence to an optima within a finite numerical tolerance is still considered convergence to global optima within the optimization community.