Author Feedback: Mostly, we would like to address Reviewer #4 who has misunderstood virtually all parts of our paper. First, Reviewer #4 claims that our paper only works for orthogonal tensors. This is not true and is actually the main point of the paper. As we discussed at length in Section 1.1 the only previous works *with provable guarantees* needed to assume orthogonality, near orthogonality or used a very large semidefinite program. We use PCA in each iteration, but as we explained, the goal is to find an approximation to the subspace spanned by the unknown factors (whether they be orthonormal or not) and then in a postprocessing step that works via Jennrich's algorithm to find factors that span this space that work (but again, they need not be orthonormal).

Second, Reviewer #4 has seemingly missed the entire point of the paper that the goal is to provide the first theoretical guarantees for exact tensor completion (and as a bonus we get a practical new algorithm). Instead Reviewer #4 opines that we sacrifice the monotone improvement property of ALS. So? ALS has no known provable guarantees and as we show experimentally can get stuck in suboptimal local minima. Instead we give a new method that provably works and achieves linear convergence rate and never gets stuck. The goal of our paper is to discover new ways to do things. These involve subtle changes and Reviewer #4 has seemingly misunderstood how our algorithm works and instead attempts to draw faulty analogies with other existing methods.

Third, Reviewer #4 makes some bizarre claims, such as arguing that the $r > n$ case is the most interesting and natural setting. Tensor decompositions and completions are used in practice in many applications tensors are at least close to being low rank. In fact a google search for the exact phrase "Low Rank Tensor Completion" yields over 35k results.

That said Reviewer #4 brings up a fair point that we could have included more experimental comparisons, e.g. with other heuristics that are out there that do not have provable guarantees. First, as has been documented many times in the literature (see [1]), standard baselines (ADMM, FaLRTC, TMac, TTN) that require storing the whole tensor in memory run out of memory even on tensors that have dimensions $15000 \times 15000 \times 5$. These tensors have about as many entries as the tensors in our largest experiments. Second we tried out the CPD method in tensorlab which indeed performs much worse than our method. For example, when we have a random tensor with $n = 200$, $r = 4$ and the number of observations is 50000 our method reliably achieves very small 10^{-6} error whereas CPD reliably achieves error rates that are orders of magnitude worse, between 10^{-1} and 10^{0}. Only when the number of observations is much larger (200000) does CPD start to be competitive.

	Kronecker Comp. (50k)	Tensorlab CPD (50k)	Kronecker Comp. (200k)	Tensorlab CPD (200k)
$10^{-1} <$ err $< 10^{0}$	1	36	0	10
$10^{-2} <$ err $< 10^{-1}$	0	4	0	5
$10^{-6} <$ err $< 10^{-2}$	1	0	0	2
err $< 10^{-6}$	38	0	40	23

Figure 1: Entries are the number of trials out of 40 that achieved a certain relative RMSE. Kronecker Comp. (our algorithm) is run for 100 iterations. Tensorlab CPD is run with all default settings.

Reviewer #3 asks about the relation between our work and the paper "Provable tensor factorization with missing data". In fact, they assume that the factors are orthogonal. In contrast, ours is the first algorithm to achieve exact recovery while allowing the factors to be strongly correlated.

Reviewer #1 asks about the complexity of the LS step. The LS problem we are solving involves $r^{O(1)}$ variables and so it can be straightforwardly solved in nearly linear time in the number of observations (when r is polylogarithmic) using Gram-Schmidt. We will clarify this in our paper. The only important point is that, while the vectors have dimension n^2, they are sparse so we just need to store their nonzero entries as lists.

Reviewer #1 also asks about how to set the parameters in practice. We can always terminate the iterative step early if we reach small enough error. The important thing is how the parameters in Theorem 3.2 should be set, and we find that in practice changing the 300 in the exponent to 2 suffices. Our focus was the dependence of our sample complexity and running time on n, but seemingly it does quite well in terms of r too, which is necessary for the kinds of experimental results we get.

[1] Xiawei Guo, Quanming Yao, and James Tin-Yau Kwok. Efficient sparse low-rank tensor completion using the frank-wolfe algorithm. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.