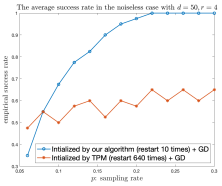
We thank the reviewers for very helpful comments. This letter addresses several major questions raised by the reviewers.

Symmetric tensor models. We will change our title to "symmetric tensor completion ..." as suggested. Note, however, that our results can be extended to the non-symmetric cases straightforwardly. We shall include a new section to discuss this generalization in the final paper. In addition, it has been shown by [4] that any algorithm for symmetric tensors can be used to estimate asymmetric tensors.

Comparisons with prior algorithms. We will add detailed numerical comparisons with prior algorithms to show the advantage of our algorithm (numerical comparison with tensor power method (TPM) [33] is shown in the figure below). The theoretical comparison when $r = O(1)$ (ignoring log factors) is summarized in the table below. Our algorithm achieves optimality in all three aspects (computational cost, sample complexity, statistical accuracy).

	sample complexity	computational cost	ℓ_2 error (noisy case)
our theory	$d^{1.5}$ (computational limit)	pd^3 (linear time)	$\sigma\sqrt{\frac{d}{p}}$ (optimal)
[33] (TPM + alt-min)	$\frac{d^{1.5}}{d^3}$ (initialized by *our scheme*) (initialized by TPM, detailed below)	pd^3	n/a
[4], [50] (SOS)	$d^{1.5}$	d^{10}	$\sigma d^{1.5}$
[63] (spectral init + GD on manifold)	$d^{1.5}$	polynomial ($\gg$ linear time)	n/a
[48] (convex relaxation + unfolding)	d^2	d^5	n/a



Real-data experiments. We will adopt the reviewers' suggestion to include experimental results on real data, including two applications. 1. *Medical images*: We will apply our algorithm to MRI scans to exploit similarities between different slices of MRI. We plan to use the MRI scans from the OsiriX repository as our dataset. We will first stack images into a tensor and then apply our algorithm to complete the tensor. The recovered images as well as the relative square error will be presented. 2. *Collaborative filtering*: We will apply our algorithm to collaborative filtering in the tensor case, where each user corresponds to two categories. For example, we can use our algorithm to predict customer ratings for hotels as well as attributes they care (e.g. room/location/service).

Inadequecy of the tensor power method (TPM) for initialization. In order for GD or other first-order methods to converge fast, it is crucial to provide a careful initialization (as we show in the supplement, random initialized GD does not work optimally). One approach that has been adopted in prior work is TPM. However, as already pointed out by [53], the TPM performs quite suboptimally for tensor problems (even when there is no missing data). As discussed in Section 4.2 of the supplementary material, when using the TPM in the initialization step (e.g. [33]), the perturbation bound in [2, Theorem 5.1] requires the tensor perturbation to be no larger than $o(1/d)$. This cannot possibly hold under the sample size assumption $p \asymp \mathsf{poly}\log d/d^{1.5}$ (since we can only hope to get $1/\mathsf{poly}\log d$ even in the noiseless case [33, Theorem 2.1]). As a result, one would need an unaffordable large number of random restarts if we were to adopt the approach in [33] initialized by the TPM. Instead, the approach in [33] can work as long as we replace the TPM initialization by the initialization procedure proposed in our work (numerical comparison is shown in the figure above).

Specific questions by Reviewer 1: 1. *Initialization*: see the response above about "initialization".

2. *Number of restarts*: We use 10 restarts throughout all experiments, which suffice to obtain satisfactory performances. As predicted by the theoretical analysis, a constant number of restarts are sufficient. This should be contrasted with prior work which requires a huge number of restarts (see discussion about "initialization" above).

Specific questions by Reviewer 2: see the response above for "real-data experiments".

Specific questions by Reviewer 3: 1. *Dependency on the condition number* κ: The quantity κ only affects the sample size requirement. If we remove the assumption A3, then an extra $\mathsf{poly}(\kappa)$ factor will appear in the sample complexity (more specifically, $p \gg \kappa^6 d^{-1.5}\log^6 d$) in Theorem 1.4. In addition, we note that κ does not affect the step size.

2. *Initialization*: see the response above for "initialization". We will separate our theoretical results for initialization and local optimization to improve readability.

3. *Contribution*: The sample complexity $O(d^{1.5})$ is widely conjectured to be optimal among all polynomial algorithms when $r = O(1)$. However, it is unclear how to achieve it within linear time. In particular, many nonconvex algorithms require an initialization stage that is very expensive both in sample complexity and computational complexity (see discussion above about the TPM and theoretical comparisons with other algorithms). Our main contribution is to develop a provably efficient linear-time algorithm with minimal sample complexity.

4. *Implicit regularization of GD*: It has been shown that there is no need to enforce projection or other regularization in various statistical models (e.g. matrix completion and phase retrieval) [46]. Similarly, a projection step is not crucial for the tensor case (which can be proven via a leave-one-out analysis). Regarding the upper bound on the iteration count: this is mainly due to the presence of noise. In the noiseless case, there is no need to impose any upper bound on T; in the noisy case, one can also use a slightly more complicated argument to remove this upper bound (we have chosen to keep this upper bound so as to slightly simplify the analysis, as $T = O(d^{10})$ seems to cover all practical scenarios).