The main objective of this paper is to propose a novel scalable Tucker-based tensor decomposition algorithm and more importantly to provide a convergence rate for such an algorithm. As far as we know, this is the first result of this kind especially for the exact and inexact gradient computations.

The experimental results presented in the main paper and in the supplementary material supports our claims of scalability and efficiency

Reviewer 1. Our main contribution is the scalable algorithm and its theoretical analysis. Adaptivity of the algorithm to constraints is a by-product of the numerical scheme we propose not our main contribution.

Experiments using non-negative constraints are exposed in Figure 2 for both Enron and Movielens datasets. Our comparison show that our approaches can handle larger tensors than competitors. The online capability of our approach has also been illustrated on an online setting in Figure 3.

We have already more than twenty references related to tensor decompositions including papers using SVD and HOSVD but we will be happy to add some other relevant references we have missed and that the reviewer will explicitly point us to.

Reviewer 2. Thanks for acknowledging the strengths of the paper. Again, we want ot stress that as far as we known, we are the first to provide convergence rate on Tucker decomposition algorithm using exact and inexact gradient descent.

Unfortunately, because we have focused our paper on the theoretical results, most experiments have been deported to the supplementary materials. Let us point to some important results

- For $M = 10000$, our approach would still be applicable. As explained in remark 1 and section 3.4, instead of computing gradients from slices of the tensor, we can decompose such a slice in subtensors and then apply our scheme on those subtensors. The Section 5 of the supplementary material provides the mathematical details of this point.
- For large M, we can also consider an online setting as exposed in the last paragraph of the experimental section. Results for this setting is given in Figure 3 and several online (with positive constraints) results are also available in the supplementary material Section 4.3.
- Other experimental analyses provided in the appendix deal with sanity-check of our hypotheses for the inexact gradient approach, some online decomposition performances with respect to rank of the core tensor.

Reviewer 3 As for reviewer 2, thanks for acknowledging that our algorihm and the provided theoretical results on its convergence rate is a strong contribution.

Regarding the experiments, we have decided to sample the original tensors of Movielens and Enron in order to be able to analyze how different competitors behave with increasing size of subtensors. The choice of the third dimension has been made especially for the competitors not to blow-up memory. As we have stated above for reviewer 2, in our case, our algorithm can handle any size of subtensors (as illustrated for instance in Figure 3 for the online setting).

Our experiments have been run on Enron and Movielens and consider as a competitor *TensorSketch* algorithm ([4] Becker et al., Nips 2018) which is tailored for sparse tensors (See their experiments in section 4).

Note also that several experimental analysis have been deported to the supplementary material due to the lack of space and the focus on the theory. Results presented there include non-negative decomposition and online decompositions ...

We have limited ourself to a single node with 32Gb of memory due to lack of better material. However, we believe that our theory and the experiments we carried out is sufficient for making the point that our approach can handle larger tensors owing to our subtensor approach.

Tensor decomposition usually leads to memory blow up during the gradient computation as they involves kronecker matrix products that have high computational spatial complexity.

The fixed mode n as described in Section 3.1 can be chosen arbitrarily. The convergence of the algorithm does not depend on this parameter. However, a relevant heuristic choice is to choose n such that I_n is the largest dimension since this will result in the smallest subtensors to handle.

For the final version, we will improve writings and motivate better the problem we address and improve the illustration of our main contribution.