We would like to thank all the reviewers for their comprehensive reviews. We clarify the major comments below.

Robustness, Novelty [**R2**] TensorNOODL is a simple online tensor factorization algorithm which provably recovers the constituent factors exactly (upto scalings and permutations) at a linear rate for an inherently non-convex problem. Our main result only states the *sufficient conditions*, and as shown in Fig. 4, the algorithm succeeds for all values of $\{J, K, m, \alpha, \beta\}$ showing its robustness; see also App.E where it shows orders of magnitude superior performance as compared to related techniques. In addition, our analysis of the resulting Khatri-Rao structure is, to the best of our knowledge, also the first to establish guarantees for recovery of constituent factors from the Kronecker structured data.

Model Assumptions [**R1, R2**] For probabilistic analysis like ours, it is common to impose structure on the factors, such assumptions are also used by existing provable algorithms for tensor factorization [1,2,3]. As discussed in Sec. 1.3, the existing provable algorithms for tensor factorization either only apply to the undercomplete case ($m \leq n$) [7,8,9], or when applicable for overcomplete case, use sum of squares [6], or tensor power iterations with multiple initializations [1,3], which are cumbersome for practical use. As noted in Sec.6 (and suggested by **R1,R3**) this is a first of its kind simple provable algorithm, and we aim to work towards relaxing the conditions in our future work. Note that although we leverage results from [11], our analysis works with any dictionary learning algorithm with appropriate guarantees.

Model Selection and Connections to popular algorithms [**R2**] While in general, the model selection for tensor factorization task is NP-hard [5] and is usually based on heuristics, for this particular structured tensor factorization task, our model assumptions ensure that the initialization algorithm such as [2] can recover $\mathbf{A}^{(0)}$ as required by A.2., implicitly accomplishing the model selection (finding m). Further, our aim here is to establish the theoretical underpinnings behind popular heuristics (such as alternating optimization and exploiting the Khatri-Rao structure) employed in tensor factorization tasks to enable development of simple, provable, and practical algorithms. Here, (1) defines a general CP/PARAFAC problem only. As discussed in Sec.1, 1.1, 2-4, and Fig. 1, TensorNOODL accomplishes *online* CP decomposition of a *structured tensor*, useful in applications where batch processing is not viable.

Comparisons with ALS and using Tensor structure [**R3**] We compare TensorNOODL with related techniques which are also agnostic to the tensor structure for a fairness. Out of these, Mairal '09 [4] can be viewed as a variant of ℓ_1-based Alternating Least Squares (ALS). In addition, our result also shows that for this task, where a number of mode-1 fibers are zero, processing only the non-zero fibers may lead to significant gains since there is no need to solve large sparse approximation subproblems as is the case with ALS (which also need to be tuned). Therefore, it seems that leveraging tensor structure may increase the computational complexity. Nevertheless, it can be potentially be useful in presence of noise, where this structure is not obvious. Thank you for this insight.

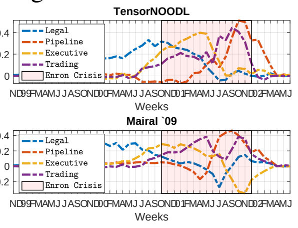

Fig.A Comparing signatures of TensorNOODL & [4].

Arora '15["unbiased"] and Qualitative Real-data Experiments [**R3**] The authors propose an *unbiased* counterpart in [2]. They conjecture that the sample complexity is similar to the *biased* counterpart, but leave the exact analysis to future work; see [2], Thm 2, and footnote 2. In our experiments, we find that given equal number of samples, the unbiased version is only slightly better than the biased counterpart. For quantitative real-world data, we compare the cluster purity of TensorNOODL with other techniques [4,12] in App.E.1 for the Enron dataset. We've added Fig.A which shows the qualitative signatures recovered, note that [12] does not recover the signatures.

Initializing $\mathbf{A}^{(0)}$, Random initializations [**R3,R4**] For real data experiments we use the algorithm proposed by [2] to initialize the algorithm (line 261, App. E.2.). For synthetic data experiments, the initialization algorithm of [2] with a running time of $\tilde{\mathcal{O}}(m^2n^2s)$ becomes the bottleneck (in terms of time) as m and s increase. Therefore, we perturb the ground-truth dictionary (in accordance with our theoretical result) for these experiments, providing all methods with this initialization. Further, TensorNOODL requires the initial dictionary estimate to follow A.2. for exact recovery at a linear rate. Initializations which do not meet these conditions may still converge, albeit not at a linear rate. Empirically, we do see evidence in support of this, and we agree with R4 that using random initializations (along with model selection algorithm) is a very interesting direction for our future work.

FISTA [**R4**] TensorNOODL at its core uses Iterative Hard Thresholding (IHT) for sparse approximation, which essentially solves a projected gradient descent problem for the ℓ_0 problem (counterpart of ISTA which operates for the ℓ_1 case). Indeed, it is possible to build upon our work to use other algorithms for faster convergence. In theory, any algorithm which can preserve the signed-support recovery of $\mathbf{x}$ can be used; see Lem. 3.

[1] Anandkumar, Ge, Janzamin ('15). Learning overcomplete latent variable models through tensor methods. (COLT). [2] Arora, Ge, Ma, Moitra ('15). Simple, efficient, and neural algorithms for sparse coding. (COLT). [3] Sun, Lu, Liu, Cheng ('17). Provable sparse tensor decomposition. (JRSS). [4] Mairal, Bach, Ponce, Sapiro ('09). Online dictionary learning for sparse coding. (ICML). [5] H astad ('90). Tensor rank is np-complete. (JoA) [6] Tang ('15). Guaranteed tensor decomposition: A moment approach. (ICML) [7] Anandkumar, Ge, Hsu, Kakade, Telgarsky ('14). Tensor decompositions for learning latent variable models. (JMLR) [8] Anandkumar, Jain, Shi, Niranjan ('16). Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations. (AISTATS) [9] Sharan and Valiant ('17). Orthogonalized als: A theoretically principled tensor decomposition algorithm for practical use. (ICML) [11] Rambhatla, Li, Haupt ('19). NOODL: Provable Online Dictionary Learning and Sparse Coding. (ICLR) [12] Fu, Huang, Ma, Sidiropoulos, Bro ('15). Joint tensor factorization and outlying slab suppression with applications. (TSP)