
Orthogonalized ALS: A Theoretically Principled Tensor Decomposition Algorithm for Practical Use

Vatsal Sharan¹ Gregory Valiant¹

Abstract

The popular Alternating Least Squares (ALS) algorithm for tensor decomposition is efficient and easy to implement, but often converges to poor local optima—particularly when the weights of the factors are non-uniform. We propose a modification of the ALS approach that is as efficient as standard ALS, but provably recovers the true factors with random initialization under standard incoherence assumptions on the factors of the tensor. We demonstrate the significant practical superiority of our approach over traditional ALS for a variety of tasks on synthetic data—including tensor factorization on exact, noisy and over-complete tensors, as well as tensor completion—and for computing word embeddings from a third-order word tri-occurrence tensor.

1. Introduction

From a theoretical perspective, tensor methods have become an incredibly useful and versatile tool for learning a wide array of popular models, including topic modeling (Anandkumar et al., 2012), mixtures of Gaussians (Ge et al., 2015), community detection (Anandkumar et al., 2014a), learning graphical models with guarantees via the method of moments (Anandkumar et al., 2014b; Chaganty & Liang, 2014) and reinforcement learning (Azizzadenesheli et al., 2016). The key property of tensors that enables these applications is that tensors have a unique decomposition (decomposition here refers to the most commonly used CANDECOMP/PARAFAC or CP decomposition), under mild conditions on the factor matrices (Kruskal, 1977); for example, tensors have a unique decomposition whenever the factor matrices are full rank. As tensor methods naturally model three-way (or higher-order) relationships, it is not too optimistic to hope that

their practical utility will only increase, with the rise of multi-modal measurements (e.g. measurements taken by “Internet of Things” devices) and the numerous practical applications involving high order dependencies, such as those encountered in natural language processing or genomic settings. In fact, we are already seeing exciting applications of tensor methods for analysis of high-order spatiotemporal data (Yu & Liu, 2016), health data analysis (Wang et al., 2015a) and bioinformatics (Colombo & Vlassis, 2015). Nevertheless, to truly realize the practical impact that the current theory of tensor methods portends, we require better algorithms for computing decompositions—practically efficient algorithms that are both capable of scaling to large (and possibly sparse) tensors, and are robust to noise and deviations from the idealized “low-rank” assumptions.

As tensor decomposition is NP-Hard in the worst-case (Hillar & Lim, 2013; Håstad, 1990), one cannot hope for algorithms which always produce the correct factorization. Despite this worst-case impossibility, accurate decompositions can be efficiently computed in many practical settings. Early work from the 1970’s (Leurgans et al., 1993; Harshman, 1970) established a simple algorithm for computing the tensor decomposition (in the noiseless setting) provided that the factor matrices are full rank. This approach, based on an eigendecomposition, is very sensitive to noise in the tensor (as we also show in our experiments), and does not scale well for large, sparse tensors.

Since this early work, much progress has been made. Nevertheless, many of the tensor decomposition algorithms hitherto proposed and employed have strong provable success guarantees but are computationally expensive (though still polynomial time)—either requiring an expensive initialization phase, being unable to leverage the sparsity of the input tensor, or not being efficiently parallelizable. On the other hand, there are also approaches which are efficient to implement, but which fail to compute an accurate decomposition in many natural settings. The Alternating Least Squares (ALS) algorithm (either with random initialization or more complicated initializations) falls in this latter category and is, by far, the most widely employed decomposition algorithm despite its often poor performance

¹Stanford University, USA. Correspondence to: Vatsal Sharan <vsharan@stanford.edu>.

and propensity for getting stuck in local optima (which we demonstrate on both synthetic data and real NLP data).

In this paper we propose an alternative decomposition algorithm, “Orthogonalized Alternating Least Squares” (Orth-ALS) which has strong theoretical guarantees, and seems to significantly outperform the most commonly used existing approaches in practice on both real and synthetic data, for a number of tasks related to tensor decomposition. This algorithm is a simple modification of the ALS algorithm to periodically “orthogonalize” the estimates of the factors. Intuitively, this periodic orthogonalization prevents multiple recovered factors from “chasing after” the same true factors, allowing for the avoidance of local optima and more rapid convergence to the true factors.

From the practical side, our algorithm enjoys all the benefits of standard ALS, namely simplicity and computational efficiency/scalability, particularly for very large yet sparse tensors, and noise robustness. Additionally, the speed of convergence and quality of the recovered factors are *substantially* better than standard ALS, even when ALS is initialized using the more expensive SVD initialization. As we show, on synthetic low-rank tensors, our algorithm consistently recovers the true factors, while standard ALS often falters in local optima and fails both in recovering the true factors and in recovering an accurate low-rank approximation to the original tensor. We also applied Orth-ALS to a large 3-tensor of word co-occurrences to compute “word embeddings”.¹ The embedding produced by our Orth-ALS algorithm is *significantly* better than that produced by standard ALS, as we quantify via a near 30% better performance of the resulting word embeddings across standard NLP datasets that test the ability of the embeddings to answer basic analogy tasks (i.e. “puppy is to dog as kitten is to ___?”) and semantic word-similarity tasks. Together, these results support our optimism that with better decomposition algorithms, tensor methods will become an indispensable, widely-used data analysis tool in the near future.

Beyond the practical benefits of Orth-ALS, we also consider its theoretical properties. We show that Orth-ALS provably recovers all factors under *random* initialization for worst-case tensors as long as the tensor satisfies an incoherence property (which translates to the factors of the tensors having small correlation with each other), which is satisfied by random tensors with rank $k = o(d^{0.25})$ where d is the dimension of the tensor. This requirement that $k = o(d^{0.25})$ is significantly worse than the best known provable recovery guarantees for polynomial-time algorithms

¹Word embeddings are vector representations of words, which can then be used as features for higher-level machine learning. Word embeddings have rapidly become the backbone of many downstream natural language processing tasks (see e.g. (Mikolov et al., 2013b)).

on random tensors—the recent work Ma et al. (2016) succeeds even in the over-complete setting with $k = o(d^{1.5})$. Nevertheless, our experiments support our belief that this shortcoming is more a property of our analysis than the algorithm itself. Additionally, for many practical settings, particularly natural language tasks, the rank of the recovered tensor is typically significantly sublinear in the dimensionality of the space, and the benefits of an extremely efficient and simple algorithm might outweigh limitations on the required rank for provable recovery.

Finally, as a consequence of our analysis technique for proving convergence of Orth-ALS, we also improve the known guarantees for another popular tensor decomposition algorithm—the tensor power method. We show that the tensor power method with *random* initialization converges to one of the factors with small residual error for rank $k = o(d)$, where d is the dimension. We also show that the convergence rate is *quadratic* in the dimension. Anandkumar et al. (2014c) had previously shown local convergence of the tensor power method with a linear convergence rate (and also showed global convergence via a SVD-based initialization scheme, obtaining the first guarantees for the tensor power method in non-orthogonal settings). Our new results, particularly global convergence from random initialization, provide some deeper insights into the behavior of this popular algorithm.

The rest of the paper is organized as follows—Section 2 states the notation. In Section 3 we discuss related work. Section 4 introduces Orth-ALS, and states the convergence guarantees. We state our convergence results for the tensor power method in Section 4.2. The experimental results, on both synthetic data and the NLP tasks are discussed in Section 5. Proof details have been deferred to the Appendix.

2. Notation

We state our algorithm and results for 3rd order tensors, and believe the algorithm and analysis techniques should extend easily to higher dimensions. Given a 3rd order tensor $T \in \mathbb{R}^{d \times d \times d}$ our task is to decompose the tensor into its factor matrices A, B and C : $T = \sum_{i \in [k]} w_i A_i \otimes B_i \otimes C_i$, where A_i denotes the i th column of a matrix A . Here $w_i \in \mathbb{R}, A_i, B_i, C_i \in \mathbb{R}^d$ and \otimes denotes the tensor product: if $a, b, c \in \mathbb{R}^d$ then $a \otimes b \otimes c \in \mathbb{R}^{d \times d \times d}$ and $(a \otimes b \otimes c)_{ijk} = a_i b_j c_k$. We will refer to w_i as the weight of the factor $\{A_i, B_i, C_i\}$. This is also known as CP decomposition. We refer to the dimension of the tensor by d and denote its rank by k . We refer to different dimensions of a tensor as the modes of the tensor.

We denote $T_{(n)}$ as the mode n matricization of the tensor, which is the flattening of the tensor along the n th direction obtained by stacking all the matrix slices together. For example $T_{(1)}$ denotes flattening of a tensor $T \in \mathbb{R}^{n \times m \times p}$ to

a $(n \times mp)$ matrix. We denote the Khatri-Rao product of two matrices A and B as $(A \odot B)_i = (A_i \otimes B_i)_{(1)}$, where $(A_i \otimes B_i)_{(1)}$ denotes the flattening of the matrix $A_i \otimes B_i$ into a row vector. For any tensor T and vectors a, b, c , we also define $T(a, b, c) = \sum_{i,j,k} T_{ijk} a_i b_j c_k$. Throughout, we say $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n))$ up to poly-logarithmic factors.

Though all algorithms in the paper extend to asymmetric tensors, we prove convergence results under the symmetric setting where $A = B = C$. Similar to other works (Tang & Shah, 2015; Anandkumar et al., 2014c; Ma et al., 2016), our guarantees depend on the incoherence of the factor matrices (c_{\max}), defined to be the maximum correlation in absolute value between any two factors, i.e. $c_{\max} = \max_{i \neq j} |A_i^T A_j|$. This serves as a natural assumption to simplify the problem as it is NP-Hard in the worst case. Also, tensors with randomly drawn factors satisfy $c_{\max} \leq \tilde{O}(1/\sqrt{d})$, and our results hold for such tensors.

3. Background and Related Work

We begin the section with a brief discussion of related work on tensor decomposition. We then review the ALS algorithm and the tensor power method and discuss their basic properties. Our proposed tensor decomposition algorithm, Orth-ALS, builds on these algorithms.

3.1. Related Work on Tensor Decomposition

Though it is not possible for us to do justice to the substantial body of work on tensor decomposition, we will review three families of algorithms which are distinct from alternating minimization approaches such as ALS and the tensor power method. Many algorithms have been proposed for guaranteed decomposition of *orthogonal* tensors, we refer the reader to Anandkumar et al. (2014b); Kolda & Mayo (2011); Comon et al. (2009); Zhang & Golub (2001). However, obtaining guaranteed recovery of non-orthogonal tensors using algorithms for orthogonal tensors requires converting the tensor into an orthogonal form (known as *whitening*) which is ill conditioned in high dimensions (Le et al., 2011; Souloumiac, 2009), and is computationally the most expensive step (Huang et al., 2013). Another very interesting line of work on tensor decompositions is to use simultaneous diagonalization and higher order SVD (Colombo & Vlassis, 2016; Kuleshov et al., 2015; De Lathauwer, 2006) but these are not as computationally efficient as alternating minimization². Recently, there has been in-

²De Lathauwer (2006) prove unique recovery under very general conditions, but their algorithm is quite complex and requires solving a linear system of size $O(d^4)$, which is prohibitive for large tensors. We ran the simultaneous diagonalization algorithm of Kuleshov et al. (2015) on a dimension 100, rank 30 tensor; and the algorithm needed around 30 minutes to run, whereas Orth-ALS converges in less than 5 seconds.

triguing work on provably decomposing random tensors using the *sum-of-squares* approach (Ma et al., 2016; Hopkins et al., 2016; Tang & Shah, 2015; Ge & Ma, 2015). Ma et al. (2016) show that a sum-of-squares based relaxation can decompose highly overcomplete random tensors of rank up to $o(d^{1.5})$. Though these results establish the polynomial learnability of the problem, they are unfortunately not practical.

Very recently, there has been exciting work on scalable tensor decomposition algorithms using ideas such as sketching (Song et al., 2016; Wang et al., 2015b) and contraction of tensor problems to matrix problems (Shah et al., 2015). Also worth noting are recent approaches to speedup ALS via sampling and randomized least squares (Battaglino et al., 2017; Cheng et al., 2016; Papalexakis et al., 2012).

3.2. Alternating Least Squares (ALS)

ALS is the most widely used algorithm for tensor decomposition and has been described as the “workhorse” for tensor decomposition (Kolda & Bader, 2009). The algorithm is conceptually very simple: if the goal is to recover a rank- k tensor, ALS maintains a rank- k decomposition specified by three sets of $d \times k$ dimensional matrices $\{\hat{A}, \hat{B}, \hat{C}\}$ corresponding to the three modes of the tensor. ALS will iteratively fix two of the three modes, say \hat{A} and \hat{B} , and then update \hat{C} by solving a least-squared regression problem to find the best approximation to the underlying tensor T having factors \hat{A} and \hat{B} in the first two modes, namely $\hat{C}_{new} = \arg \min_{C'} \|T - \hat{A} \otimes \hat{B} \otimes C'\|_2$. ALS will then continue to iteratively fix two of the three modes, and update the other mode via solving the associated least-squares regression problem. These updates continue until some stopping condition is satisfied—typically when the squared error of the approximation is no longer decreasing, or when a fixed number of iterations have elapsed. The factors used in ALS are either chosen uniformly at random, or via a more expensive initialization scheme such as SVD based initialization (Anandkumar et al., 2014c). In the SVD based scheme, the factors are initialized to be the singular vectors of a random projection of the tensor onto a matrix.

The main advantages of the ALS approach, which have led to its widespread use in practice are its conceptual simplicity, noise robustness and computational efficiency given its graceful handling of sparse tensors and ease of parallelization. There are several publicly available optimized packages implementing ALS, such as Kossaifi et al. (2016); Vervliet et al.; Bader et al. (2012); Bader & Kolda (2007); Smith & Karypis; Huang et al. (2014); Kang et al. (2012).

Despite the advantages, ALS does not have any global convergence guarantees and can get stuck in local optima (Comon et al., 2009; Kolda & Bader, 2009), even under very realistic settings. For example, consider a setting where the weights w_i for the factors $\{A_i, B_i, C_i\}$ decay

according to a power-law, hence the first few factors have much larger weight than the others. As we show in the experiments (see Fig. 2), ALS fails to recover the low-weight factors. Intuitively, this is because multiple recovered factors will be chasing after the *same* high weight factor, leading to a bad local optima.

3.3. Tensor Power Method

The tensor power method is a special case of ALS that only computes a rank-1 approximation. The procedure is then repeated multiple times to recover different factors. The factors recovered in different iterations of the algorithm are then clustered to determine the set of unique factors. Different initialization strategies have been proposed for the tensor power method. Anandkumar et al. (2014c) showed that the tensor power method converges locally (i.e. for a suitably chosen initialization) for random tensors with rank $o(d^{1.5})$. They also showed that a SVD based initialization strategy gives good starting points and used this to prove global convergence for random tensors with rank $O(d)$. However, the SVD based initialization strategy can be computationally expensive, and our experiments suggest that even SVD initialization fails in the setting where the weights decay according to a power-law (see Fig. 2).

In this work, we prove global convergence guarantees with random initializations for the tensor power method for random and worst-case incoherent tensors. Our results also demonstrate how, with random initialization, the tensor power method converges to the factor having the largest product of weight times the correlation of the factor with the random initialization vector. This explains the difficulty of using random initialization to recover factors with small weight. For example, if one factor has weight less than a $1/c$ fraction of the weight of, say, the heaviest $k/2$ factors, then with high probability we would require at least $k^{\Theta(c^2)}$ random initializations to recover this factor. This is because the correlation between random vectors in high dimensions is approximately distributed as a Normal random variable and if $k/2 + 1$ samples are drawn from the standard Normal distribution, the probability that one particular sample is at least a factor of c larger than the other $k/2$ other samples scales as roughly $k^{-\Theta(c^2)}$.

4. The Algorithm: Orthogonalized Alternating Least Squares (Orth-ALS)

In this section we introduce Orth-ALS, which combines the computational benefits of standard ALS and the provable recovery of the tensor power method, while avoiding the difficulties faced by both when factors have different weights. Orth-ALS is a simple modification of standard ALS that adds an orthogonalization step before each set of ALS steps. We describe the algorithm below. Note that steps 4-6 are just the solution to the least squares problem

expressed in compact tensor notation, for instance step 4 can be equivalently stated as $X = \arg \min_{C'} \|T - \hat{A} \otimes \hat{B} \otimes C'\|_2$. Similarly, step 9 is the least squares estimate of the weight w_i of each rank-1 component $\hat{A}_i \otimes \hat{B}_i \otimes \hat{C}_i$.

Algorithm 1 Orthogonalized ALS (Orth-ALS)

Input: Tensor $T \in \mathbb{R}^{d \times d \times d}$, number of iterations N .

- 1: Initialize each column of \hat{A}, \hat{B} and $\hat{C} \in \mathbb{R}^{d \times k}$ uniformly from the unit sphere
 - 2: **for** $t = 1 : N$ **do**
 - 3: Find QR decomposition of \hat{A} , set $\hat{A} = Q$. Orthogonalize \hat{B} and \hat{C} analogously.
 - 4: $X = T_{(1)}(\hat{C} \odot \hat{B})$
 - 5: $Y = T_{(2)}(\hat{C} \odot \hat{A})$
 - 6: $Z = T_{(3)}(\hat{B} \odot \hat{A})$
 - 7: Normalize X, Y, Z and store results in $\hat{A}, \hat{B}, \hat{C}$
 - 8: **end for**
 - 9: Estimate weights $\hat{w}_i = T(\hat{A}_i, \hat{B}_i, \hat{C}_i), \forall i \in [k]$.
 - 10: **return** $\hat{A}, \hat{B}, \hat{C}, \hat{w}$
-

To get some intuition for why the orthogonalization makes sense, let us consider the more intuitive *matrix* factorization problem, where the goal is to compute the eigenvectors of a matrix. Subspace iteration is a straightforward extension of the matrix power method to recover all eigenvectors at once. In subspace iteration, the matrix of eigenvector estimates is orthogonalized before each power method step (by projecting the second eigenvector estimate orthogonal to the first one and so on), because otherwise all the vectors would converge to the dominant eigenvector. For the case of tensors, the vectors would not all necessarily converge to the dominant factor if the initialization is good, but with high probability a random initialization would drive many factors towards the larger weight factors. The orthogonalization step is a natural modification which forces the estimates to converge to different factors, even if some factors are much larger than the others. It is worth stressing that the orthogonalization step does *not* force the final recovered factors to be orthogonal (because the ALS step follows the orthogonalization step) and in general the factors output will not be orthogonal (which is essential for accurately recovering the factors).

From a computational perspective, adding the orthogonalization step does not add to the computational cost as the least squares updates in step 4-6 of Algorithm 1 involve an extra pseudoinverse term for standard ALS, which evaluates to identity for Orth-ALS and does not have to be computed. The cost of orthogonalization is $O(k^2d)$, while the cost of computing the pseudoinverse is also $O(k^2d)$. We also observe significant speedups in terms of the number of iterations required for convergence for Orth-ALS as compared to standard ALS in our simulations on random tensors (see the experiments in Section 5).

Variants of Orthogonalized ALS. Several other modifications to the simple orthogonalization step also seem natural. Particularly for low-dimensional settings, in practice we found that it is useful to carry out orthogonalization for a few steps and then continue with standard ALS updates until convergence (we call this variant *Hybrid-ALS*). Hybrid-ALS also gracefully reverts to standard ALS in settings where the factors are highly correlated and orthogonalization is not helpful.

4.1. Performance Guarantees

We now state the formal guarantees on the performance of Orthogonalized ALS. The specific variant of Orthogonalized ALS that our theorems apply to is a slight modification of Algorithm 1, and differs in that there is a periodic (every $\log k$ steps) re-randomization of the factors for which our analysis has not yet guaranteed convergence. In our practical implementations, we observe that *all* factors seem to converge within this first $\log k$ steps, and hence the subsequent re-randomization is unnecessary.

Theorem 1. *Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$. Let $c_{\max} = \max_{i \neq j} |A_i^T A_j|$ be the incoherence between the true factors and $\gamma = \frac{w_{\max}}{w_{\min}}$ be the ratio of the largest and smallest weight. Assume $\gamma c_{\max} \leq o(k^{-2})$, and the estimates of the factors are initialized randomly from the unit sphere. Provided that, at the $i(\log k + \log \log d)$ th step of the algorithm the estimates for all but the first i factors are re-randomized, then with high probability the orthogonalized ALS updates converge to the true factors in $O(k(\log k + \log \log d))$ steps, and the error at convergence satisfies (up to relabelling) $\|A_i - \hat{A}_i\|_2^2 \leq O(\gamma k \max\{c_{\max}^2, 1/d^2\})$ and $|1 - \frac{\hat{w}_i}{w_i}| \leq O(\max\{c_{\max}, 1/d\})$, for all i .*

Theorem 1 immediately gives convergence guarantees for random low rank tensors. For random d dimensional tensors, $c_{\max} = O(1/\sqrt{d})$; therefore Orth-ALS converges globally with random initialization whenever $k = o(d^{0.25})$. If the tensor has rank much smaller than the dimension, then our analysis can tolerate significantly higher correlation between the factors. In the Appendix, we also prove Theorem 1 for the special and easy case of orthogonal tensors, which nevertheless highlights the key proof ideas.

4.2. New Guarantees for the Tensor Power Method

As a consequence of our analysis of the orthogonalized ALS algorithm, we also prove new guarantees on the tensor power method. As these may be of independent interest because of the wide use of the tensor power method, we summarize them in this section. We show a quadratic rate of convergence (in $O(\log \log d)$ steps) with random initialization for random tensors having rank $k = o(d)$. This contrasts with the analysis of Anandkumar et al. (2014c) who showed a linear rate of convergence ($O(\log d)$ steps)

for random tensors, provided an SVD based initialization is employed.

Theorem 2. *Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$ with the factors A_i sampled uniformly from the d -dimensional sphere. Define $\gamma = \frac{w_{\max}}{w_{\min}}$ to be the ratio of the largest and smallest weight. Assume $k \leq o(d)$ and $\gamma \leq \text{polylog}(d)$. If the initialization $x_0 \in \mathbb{R}^d$ is chosen uniformly from the unit sphere, then with high probability the tensor power method updates converge to one of the true factors (say A_1) in $O(\log \log d)$ steps, and the error at convergence satisfies $\|A_1 - \hat{A}_1\|_2 \leq \tilde{O}(1/\sqrt{d})$. Also, the estimate of the weight \hat{w}_1 satisfies $|1 - \frac{\hat{w}_1}{w_1}| \leq \tilde{O}(1/\sqrt{d})$.*

Theorem 2 provides guarantees for random tensors, but it is natural to ask if there are deterministic conditions on the tensors which guarantee global convergence of the tensor power method. Our analysis also allows us to obtain a clean characterization for global convergence of the tensor power method updates for worst-case tensors in terms of the incoherence of the factor matrix—

Theorem 3. *Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$. Let $c_{\max} = \max_{i \neq j} |A_i^T A_j|$ and $\gamma = \frac{w_{\max}}{w_{\min}}$ be the ratio of the largest and smallest weight, and assume $\gamma c_{\max} \leq o(k^{-2})$. If the initialization $x_0 \in \mathbb{R}^d$ is chosen uniformly from the unit sphere, then with high probability the tensor power method updates converge to one of the true factors (say A_1) in $O(\log k + \log \log d)$ steps, and the error at convergence satisfies $\|A_1 - \hat{A}_1\|_2^2 \leq O(\gamma k \max\{c_{\max}^2, 1/d^2\})$ and $|1 - \frac{\hat{w}_1}{w_1}| \leq O(\max\{c_{\max}, 1/d\})$.*

5. Experiments

We compare the performance of Orth-ALS, standard ALS (with random and SVD initialization), the tensor power method, and the classical eigendecomposition approach, through experiments on low rank tensor recovery in a few different parameter regimes, on an overcomplete tensor decomposition task and a tensor completion task. We also compare the factorization of Orth-ALS and standard ALS on a large real-world tensor of word tri-occurrence based on the 1.5 billion word English Wikipedia corpus.³

5.1. Experiments on Random Tensors

Recovering low rank tensors: We explore the abilities of Orth-ALS, standard ALS, and the tensor power method (TPM), to recover a low rank (rank k) tensor that has been constructed by independently drawing each of the k factors independently and uniformly at random from the d dimensional unit spherical shell. We consider several different

³MATLAB, Python and C code for Orth-ALS and Hybrid-ALS is available at <http://web.stanford.edu/~vsharan/orth-als.html>

combinations of the dimension, d , and rank, k . We also consider both the setting where all of the factors are equally weighted, as well as the practically relevant setting where the factor weights decay geometrically, and consider the setting where independent Gaussian noise has been added to the low-rank tensor.

In addition to random initialization for standard ALS and the TPM, we also explore SVD based initialization (Anandkumar et al., 2014c) where the factors are initialized via SVD of a projection of the tensor onto a matrix. We also test the classical technique for tensor decomposition via simultaneous diagonalization (Leurgans et al., 1993; Harshman, 1970) (also known as Jennrich’s algorithm, we refer to it as Sim-Diag), which first performs two random projections of the tensor, and then recovers the factors by an eigenvalue decomposition of the projected matrices. This gives guaranteed recovery when the tensors are noiseless and factors are linearly independent, but is extremely unstable to perturbations.

We evaluate the performance in two respects: 1) the ability of the algorithms to recover a low-rank tensor that is close to the input tensor, and 2) the ability of the algorithms to recover accurate approximations of many of the true factors. Fig. 1 depicts the performance via the first metric. We evaluate the performance in terms of the discrepancy between the input low-rank tensor, and the low-rank tensor recovered by the algorithms, quantified via the ratio of the Frobenius norm of the residual, to the Frobenius norm of the actual tensor: $\frac{\|T - \hat{T}\|_F}{\|T\|_F}$, where \hat{T} is the recovered tensor. Since the true tensor has rank k , the inability of an algorithm to drive this error to zero indicates the presence of local optima. Fig. 1 depicts the performance of Orth-ALS, standard ALS with random initialization and the hybrid algorithm that performs Orth-ALS for the first five iterations before reverting to standard ALS (Hybrid-ALS). Tests are conducted in both the setting where factor weights are uniform, as well as a geometric spacing, where the ratio of the largest factor weight to the smallest is 100. Fig. 1 shows that Hybrid ALS and Orth-ALS have much faster convergence and find a significantly better fit than standard ALS.

Fig. 2 quantifies the performance of the algorithms in terms of the number of the original factors that the algorithms accurately recover. We use standard ALS, Orth-ALS (Algorithm 1), Hybrid-ALS, TPM with random initialization (TPM), ALS with SVD initialization (ALS-SVD), TPM with SVD initialization (TPM-SVD) and the simultaneous diagonalization approach (Sim-Diag). We run TPM and SVD-TPM with 100 different initializations and find a rank $k = 30$ decomposition for ALS, ALS-SVD, Orth-ALS, Hybrid-ALS and Sim-Diag. We repeat the experiment (by sampling a new tensor) 10 times. We perform this evaluation in both the setting where we receive an actual low-

rank tensor as input, as well as the setting where each entry T_{ijk} of the low-rank tensor has been perturbed by independent Gaussian noise of standard deviation equal to $0.05T_{ijk}$. We can see that Orth-ALS and Hybrid-ALS perform significantly better than the other algorithms and are able to recover all factors in the noiseless case even when the weights are highly skewed. Note that the reason the Hybrid-ALS and Orth-ALS fail to recover all factors in the noisy case when the weights are highly skewed is that the magnitude of the noise essentially swamps the contribution from the smallest weight factors.

Recovering over-complete tensors: Overcomplete tensors are tensors with rank higher than the dimension, and have found numerous theoretical applications in learning latent variable models (Anandkumar et al., 2015). Even though orthogonalization cannot be directly applied to the setting where the rank is more than the dimension (as the factors can no longer be orthogonalized), we explore a deflation based approach in this setting. Given a tensor T with dimension $d = 50$ and rank $r > d$, we find a rank d decomposition T_1 of T , subtract T_1 from T , and then compute a rank d decomposition of T_1 to recover the next set of d factors. We repeat this process to recover subsequent factors. After every set of d factors has been estimated, we also refine the factor estimates of all factors estimated so far by running an additional ALS step using the current estimates of the extracted factors as the initialization. Fig. 3a plots the number of factors recovered when this deflation based approach is applied to a dimension $d = 50$ tensor with a mild power low distribution on weights. We can see that Hybrid-ALS is successful at recovering tensors even in the overcomplete setup, and gives an improvement over ALS.

Tensor completion: We also test the utility of orthogonalization on a tensor completion task, where the goal is to recover a large missing fraction of the entries. Fig. 3b suggests Hybrid-ALS gives considerable improvements over standard ALS. Further examining the utility of orthogonalization in this important setting, in theory and practice, would be an interesting direction.

5.2. Learning Word Embeddings via Tensor Factorization

A word embedding is a vector representation of words which preserves some of the syntactic and semantic relationships in the language. Current methods for learning word embeddings implicitly (Mikolov et al., 2013b; Levy & Goldberg, 2014) or explicitly (Pennington et al., 2014) factorize some matrix derived from the matrix of word co-occurrences M , where M_{ij} denotes how often word i appears with word j . We explore tensor methods for learning word embeddings, and contrast the performance of standard ALS and Orthogonalized ALS on standard tasks.

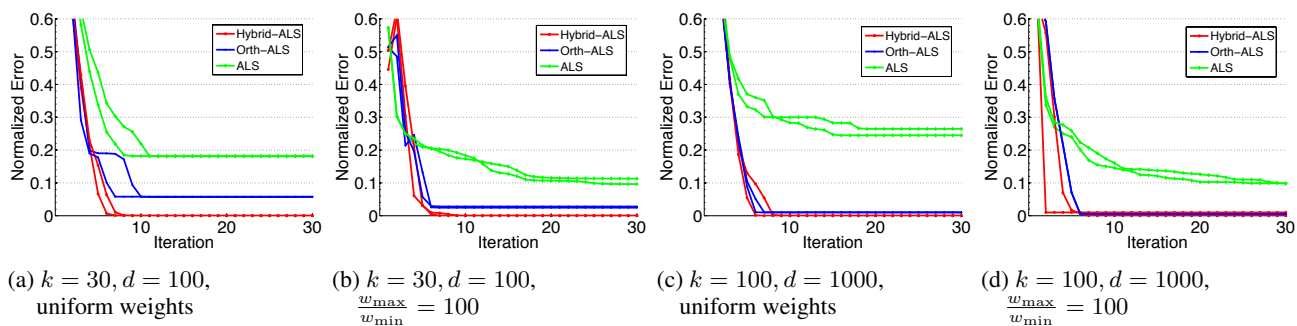


Figure 1. Plot of the normalized discrepancy between the recovered rank k tensor \hat{T} and the true tensor T : $\frac{\|T - \hat{T}\|_F}{\|T\|_F}$, as a function of the iteration. In all settings, the Orth-ALS and the hybrid algorithm drive this discrepancy nearly to zero, with the performance of Orth-ALS improving for the higher dimensional cases, whereas standard ALS algorithm has slower convergence and gets stuck in bad local optima.

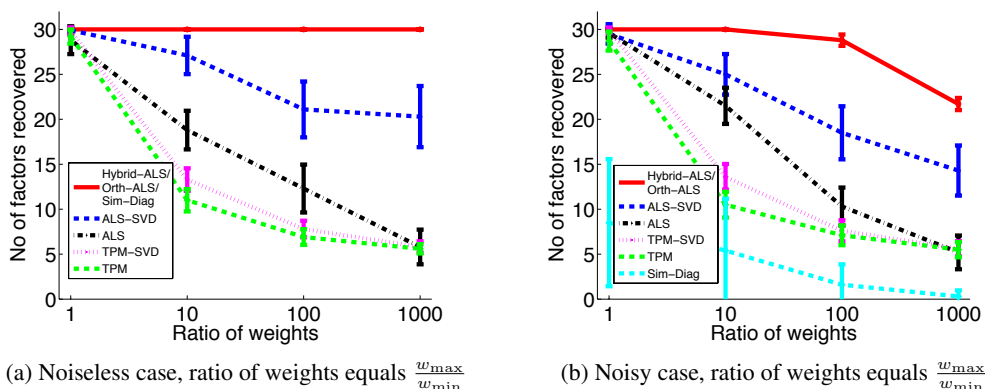


Figure 2. Average number of factors recovered by different algorithms for different values of $\frac{w_{\max}}{w_{\min}}$, the ratio of the maximum factor weight to minimum factor weight (with the weights spaced geometrically), along with error bars for the standard deviation in the number of factors recovered, across independent trials. The true rank $k = 30$, and the dimension $d = 100$. We say a factor $\{A_i, B_i, C_i\}$ of the tensor T is successfully recovered if there exists at least one recovered factor $\{\hat{A}_j, \hat{B}_j, \hat{C}_j\}$ with correlation at least 0.9 in all modes. Orth-ALS and Hybrid-ALS recover all factors in almost all settings, whereas ALS and the tensor power method struggle when the weights are skewed, even with the more expensive SVD based initialization.

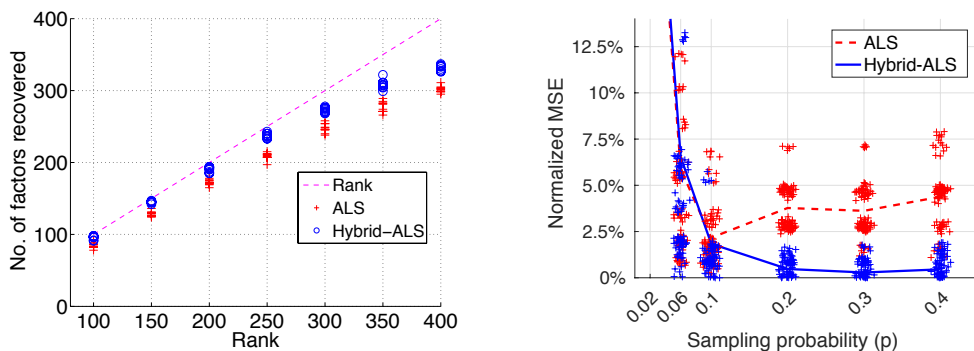


Figure 3. Experiments on overcomplete tensors and tensor completion. Even though our theoretical guarantees do not apply to these settings, we see that orthogonalization leads to significantly better performance over standard ALS.

Methodology. We used the English Wikipedia as our corpus, with 1.5 billion words. We constructed a word co-occurrence tensor T of the 10,000 most frequent words, where the entry T_{ijk} denotes the number of times the words i , j and k appear in a sliding window of length w across the corpus. We consider two different window lengths, $w = 3$ and $w = 5$. Before factoring the tensor, we apply the non-linear element-wise scaling $f(x) = \log(1 + x)$ to the tensor. This scaling is known to perform well in practice for co-occurrence matrices (Pennington et al., 2014), and makes some intuitive sense in light of the Zipfian distribution of word frequencies. Following the application of this element-wise nonlinearity, we recover a rank 100 approximation of the tensor using Orth-ALS or ALS.

We concatenate the (three) recovered factor matrices into one matrix and normalize the rows. The i th row of this matrix is then the embedding for the i th word. We test the quality of these embeddings on two tasks aimed at measuring the syntactic and semantic structure captured by these word embeddings.

We also evaluated the performance of matrix SVD based methods on the task. For this, we built the co-occurrence matrix M with a sliding window of length w over the corpus. We applied the same non-linear element-wise scaling and performed a rank 100 SVD, and set the word embeddings to be the singular vectors after row normalization.

It is worth highlighting some implementation details for our experiments, as they indicate the practical efficiency and scalability inherited by Orth-ALS from standard ALS. Our experiments were run on a cluster with 8 cores and 48 GB of RAM memory per core. Most of the runtime was spent in reading the tensor, the runtime for Orth-ALS was around 80 minutes, with 60 minutes spent in reading the tensor (the runtime for standard ALS was around 100 minutes because it took longer to converge). Since storing a dense representation of the $10,000 \times 10,000 \times 10,000$ tensor is too expensive, we use an optimized ALS solver for sparse tensors (Smith & Karypis; 2015) which also has an efficient parallel implementation.

Evaluation: Similarity and Analogy Tasks. We evaluated the quality of the recovered word embeddings produced by the various methods via their performance on two different NLP tasks for which standard, human-labeled data exists: estimating the similarity between a pair of words, and completing word analogies.

The word similarity tasks (Bruni et al., 2012; Finkelstein et al., 2001) contain word pairs along with human assigned similarity scores, and the objective is to maximize the correlation between the similarity in the embeddings of the two words (according to a similarity metric such as the dot product) and human judged similarity.

Algorithm	Similarity tasks	Analogy tasks
Standard ALS, $w = 3$	0.50	30.92%
Standard ALS, $w = 5$	0.50	37.38%
Orth-ALS, $w = 3$	0.59	40.00%
Orth-ALS, $w = 5$	0.60	46.37%
Matrix methods, $w = 3$	0.68	53.29%
Matrix methods, $w = 5$	0.67	57.40%

Table 1. Results for word analogy and word similarity tasks for different window lengths w over which the co-occurrences are counted. The embeddings recovered by Orth-ALS are significantly better than those recovered by standard ALS. Despite this, embeddings derived from word co-occurrences using matrix SVD still outperform the tensor embeddings, and we are unsure whether this is due to the relative sparsity of the tensor, sub-optimal element-wise scaling (i.e. the $f(x) = \log(1+x)$ function applied to the counts), or something more fundamental.

The word analogy tasks (Mikolov et al., 2013a;c) present questions of the form “ a is to a^* as b is to $__$?” (e.g. “Paris is to France as Rome is to $__$?”). We find the answer to “ a is to a^* as b is to b^* ” by finding the word whose embedding is the closest to $w_{a^*} - w_a + w_b$ in cosine similarity, where w_a denotes the embedding of the word a .

Results. The performances are summarized in the Table 1. The use of Orth-ALS rather than standard ALS leads to significant improvement in the quality of the embeddings as judged by the similarity and analogy tasks. However, the matrix SVD method still outperforms the tensor based methods. We believe that it is possible that better tensor based approaches (e.g. using better renormalization, additional data, or some other tensor rather than the symmetric tri-occurrence tensor) or a combination of tensor and matrix based methods can actually improve the quality of word embeddings, and is an interesting research direction. Alternatively, it is possible that natural language does not contain sufficiently rich higher-order dependencies among words that appear close together, beyond the co-occurrence structure, to truly leverage the power of tensor methods. Or, perhaps, the two tasks we evaluated on—similarity and analogy tasks—do not require this higher order. In any case, investigating these possibilities seems worthwhile.

6. Conclusion

Our results suggest the theoretical and practical benefits of Orthogonalized ALS, versus standard ALS. An interesting direction for future work would be to more thoroughly examine the practical and theoretical utility of orthogonalization for other tensor-related tasks, such as tensor completion. Additionally, it seems worthwhile to investigate Orthogonalized ALS or Hybrid ALS in more application-specific domains, such as natural language processing.

References

- Anandkumar, Animashree, Liu, Yi-kai, Hsu, Daniel J, Foster, Dean P, and Kakade, Sham M. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pp. 917–925, 2012.
- Anandkumar, Animashree, Ge, Rong, Hsu, Daniel, and Kakade, Sham M. A tensor approach to learning mixed membership community models. *The Journal of Machine Learning Research*, 15(1):2239–2312, 2014a.
- Anandkumar, Animashree, Ge, Rong, Hsu, Daniel, Kakade, Sham M, and Telgarsky, Matus. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014b.
- Anandkumar, Animashree, Ge, Rong, and Janzamin, Majid. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. *arXiv preprint arXiv:1402.5180*, 2014c.
- Anandkumar, Animashree, Ge, Rong, and Janzamin, Majid. Learning overcomplete latent variable models through tensor methods. In *Proceedings of The 28th Conference on Learning Theory*, pp. 36–112, 2015.
- Azizzadenesheli, Kamyar, Lazaric, Alessandro, and Anandkumar, Animashree. Reinforcement learning of POMDPs using spectral methods. In *29th Annual Conference on Learning Theory*, pp. 193–256, 2016.
- Bader, Brett W. and Kolda, Tamara G. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1), December 2007.
- Bader, Brett W., Kolda, Tamara G., et al. Matlab tensor toolbox version 2.5. Available online, January 2012.
- Battaglino, Casey, Ballard, Grey, and Kolda, Tamara G. A practical randomized CP tensor decomposition. *arXiv preprint arXiv:1701.06600*, 2017.
- Bruni, Elia, Boleda, Gemma, Baroni, Marco, and Tran, Nam-Khanh. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.
- Chaganty, Arun Tejasvi and Liang, Percy. Estimating latent-variable graphical models using moments and likelihoods. In *ICML*, pp. 1872–1880, 2014.
- Cheng, Dehua, Peng, Richard, Liu, Yan, and Perros, Ioakeim. SPALS: Fast alternating least squares via implicit leverage scores sampling. In *Advances In Neural Information Processing Systems*, pp. 721–729, 2016.
- Colombo, Nicolo and Vlassis, Nikos. FastMotif: spectral sequence motif discovery. *Bioinformatics*, 31(16), 2015.
- Colombo, Nicolo and Vlassis, Nikos. Tensor decomposition via joint matrix schur decomposition. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2820–2828, 2016.
- Comon, Pierre, Luciani, Xavier, and De Almeida, André LF. Tensor decompositions, alternating least squares and other tales. *Journal of chemometrics*, 23 (7-8):393–405, 2009.
- De Lathauwer, Lieven. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM journal on Matrix Analysis and Applications*, 28(3):642–666, 2006.
- Duembgen, Lutz. Bounding standard gaussian tail probabilities. *arXiv preprint arXiv:1012.2063*, 2010.
- Finkelstein, Lev, Gabrilovich, Evgeniy, Matias, Yossi, Rivlin, Ehud, Solan, Zach, Wolfman, Gadi, and Ruppin, Eytan. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pp. 406–414. ACM, 2001.
- Ge, Rong and Ma, Tengyu. Decomposing overcomplete 3rd order tensors using sum-of-squares algorithms. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 829, 2015.
- Ge, Rong, Huang, Qingqing, and Kakade, Sham M. Learning mixtures of gaussians in high dimensions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 761–770. ACM, 2015.
- Harshman, Richard A. Foundations of the parafac procedure: Models and conditions for an “explanatory” multimodal factor analysis. 1970.
- Håstad, Johan. Tensor rank is NP-Complete. *Journal of Algorithms*, 11(4):644–654, 1990.
- Hillar, Christopher J and Lim, Lek-Heng. Most tensor problems are NP-Hard. *Journal of the ACM*, 60(6), 2013.
- Hopkins, Samuel B, Schramm, Tselil, Shi, Jonathan, and Steurer, David. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, 2016.
- Huang, Furong, Niranjana, UN, Hakeem, Mohammad Umar, and Anandkumar, Animashree. Fast detection of overlapping communities via online tensor methods. *arXiv preprint arXiv:1309.0787*, 2013.
- Huang, Furong, Matushevych, Sergiy, Anandkumar, Anima, Karampatziakis, Nikos, and Mineiro, Paul. Distributed latent dirichlet allocation via tensor factorization. In *NIPS Optimization Workshop*, 2014.

- Kang, U, Papalexakis, Evangelos, Harpale, Abhay, and Faloutsos, Christos. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.
- Kolda, Tamara G and Bader, Brett W. Tensor decompositions and applications. *SIAM review*, 51(3), 2009.
- Kolda, Tamara G and Mayo, Jackson R. Shifted power method for computing tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 32(4), 2011.
- Kossaifi, Jean, Panagakis, Yannis, and Pantic, Maja. Tensorly: Tensor learning in python. *arXiv preprint arXiv:1610.09555*, 2016.
- Kruskal, Joseph B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.
- Kuleshov, Volodymyr, Chaganty, Arun Tejasvi, and Liang, Percy. Tensor factorization via matrix factorization. In *AISTATS*, 2015.
- Le, Quoc V, Karpenko, Alexandre, Ngiam, Jiquan, and Ng, Andrew Y. ICA with reconstruction cost for efficient overcomplete feature learning. In *Advances in Neural Information Processing Systems*, pp. 1017–1025, 2011.
- Leurgans, SE, Ross, RT, and Abel, RB. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.
- Levy, Omer and Goldberg, Yoav. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 2177–2185, 2014.
- Ma, Tengyu, Shi, Jonathan, and Steurer, David. Polynomial-time tensor decompositions with sum-of-squares. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, 2016.
- Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013b.
- Mikolov, Tomas, Yih, Wen-tau, and Zweig, Geoffrey. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pp. 746–751, 2013c.
- Papalexakis, Evangelos E, Faloutsos, Christos, and Sidiropoulos, Nicholas D. Parcube: Sparse parallelizable tensor decompositions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 521–536. Springer, 2012.
- Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- Shah, Parikshit, Rao, Nikhil, and Tang, Gongguo. Sparse and low-rank tensor decomposition. In *Advances in Neural Information Processing Systems*, 2015.
- Smith, Shaden and Karypis, George. SPLATT: The Surprisingly Parallel sparse Tensor Toolkit.
- Smith, Shaden and Karypis, George. DMS: Distributed sparse tensor factorization with alternating least squares. Technical report, 2015.
- Song, Zhao, Woodruff, David, and Zhang, Huan. Sublinear time orthogonal tensor decomposition. In *Advances in Neural Information Processing Systems*, 2016.
- Souloumiac, Antoine. Joint diagonalization: Is non-orthogonal always preferable to orthogonal? In *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2009.
- Tang, Gongguo and Shah, Parikshit. Guaranteed tensor decomposition: A moment approach. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1491–1500, 2015.
- Vervliet, N., Debals, O., Sorber, L., Van Barel, M., and De Lathauwer, L. Tensorlab 3.0, Mar. . Available online.
- Wang, Yichen, Chen, Robert, Ghosh, Joydeep, Denny, Joshua C, Kho, Abel, Chen, You, Malin, Bradley A, and Sun, Jimeng. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1265–1274. ACM, 2015a.
- Wang, Yining, Tung, Hsiao-Yu, Smola, Alexander J, and Anandkumar, Anima. Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems*, pp. 991–999, 2015b.
- Yu, Rose and Liu, Yan. Learning from multiway data: Simple and efficient tensor regression. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, pp. 238–247, 2016.
- Zhang, Tong and Golub, Gene H. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(2):534–550, 2001.

A. Proof Overview: the Orthogonal Tensor Case

In this section, we will consider Orthogonalized ALS for the special case when the factors matrix of the tensor is an orthogonal matrix. Although this special case is easy and numerous algorithms provably work in this setting, it will serve to highlight the high level analysis approach that we apply to the more general settings.

The analysis of Orth-ALS hinges on an analysis of the tensor power method. For completeness we describe the tensor power method in Algorithm 2. We will go through some preliminaries for our analysis of the tensor power method. Let the iterate of the tensor power method at time t be Z_t . The tensor power method update equations can be written as (refer to (Anandkumar et al., 2014c))

$$Z_t = \frac{\sum_{i=1}^k w_i \langle Z_{t-1}, A_i \rangle^2 A_i}{\| \sum_{i=1}^k w_i \langle Z_{t-1}, A_i \rangle^2 A_i \|_2} \quad (\text{A.1})$$

Eq. A.1 is just the tensor analog of the matrix power method updates. For tensors, the updates are quadratic in the previous inner products, in contrast to matrices where the updates are linear in the inner products in the previous step.

Algorithm 2 Tensor power method to recover all factors (Anandkumar et al., 2014c)

Input: Tensor $T \in \mathbb{R}^{d \times d \times d}$, number of initializations L , number of iterations N .

- 1: **for** $\tau = 1 : L$ **do**
 - 2: Initialize $x_0^{(\tau)}, y_0^{(\tau)}, z_0^{(\tau)} \in \mathbb{R}^d$ uniformly from the unit sphere or using the SVD based method
 - 3: **for** $t = 1 : N$ **do**
 - 4: Rank-1 ALS/Power method updates- $x_{t+1}^{(\tau)} = T_{(1)}(\hat{z}_t^{(\tau)} \odot \hat{y}_t^{(\tau)})$
 - 5: Rank-1 ALS/Power method updates- $y_{t+1}^{(\tau)} = T_{(2)}(\hat{z}_t^{(\tau)} \odot \hat{x}_t^{(\tau)})$
 - 6: Rank-1 ALS/Power method updates- $z_{t+1}^{(\tau)} = T_{(3)}(\hat{y}_t^{(\tau)} \odot \hat{x}_t^{(\tau)})$
 - 7: Normalize $x_{t+1}^{(\tau)}, y_{t+1}^{(\tau)}, z_{t+1}^{(\tau)}$ and store results in $\hat{x}_{t+1}^{(\tau)}, \hat{y}_{t+1}^{(\tau)}, \hat{z}_{t+1}^{(\tau)}$.
 - 8: **end for**
 - 9: Estimate weights- $\hat{w}^{(\tau)} = T(\hat{x}_N^{(\tau)}, \hat{y}_N^{(\tau)}, \hat{z}_N^{(\tau)})$
 - 10: **end for**
 - 11: Cluster set $\{(\hat{w}^{(\tau)}, \hat{x}_N^{(\tau)}, \hat{y}_N^{(\tau)}, \hat{z}_N^{(\tau)}), \tau \in [L]\}$ into k clusters.
 - 12: **return** the centers $\{(\hat{w}_i, \hat{a}_i, \hat{b}_i, \hat{c}_i), i \in [k]\}$ of the k clusters as the estimates
-

Observe from Algorithm 1 (Orth-ALS) that the ALS steps in step 4-6 have the same form as tensor power method updates, but on the orthogonalized factors. This is the key idea we use in our analysis of Orth-ALS. Note that the first factor estimate is never affected by the orthogonalization, hence the updates for the first estimated factor have exactly the same form as the tensor power method updates, as this factor is unaffected by orthogonalization. The subsequent factors have an orthogonalization step before every tensor power method step. This ensures that they never have high correlation with the factors which have already been recovered, as they are projected orthogonal to the recovered factors before each ALS step. We then use the incoherence of the factors to argue that orthogonalization does not significantly affect the updates of the factors which have not been recovered so far, while ensuring that the factors which have already been recovered always have a small correlation.

Note that Eq. A.1 is invariant with respect to multiplying the weights of all the factors by some constant. Hence for ease of exposition, we assume that all the weights lie in the interval $[1, \gamma]$, where $\gamma = \frac{w_{\max}}{w_{\min}}$. We also define $\eta = \max\{c_{\max}, 1/d\}$.

Proposition 1. Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$ where the factor matrix A is orthogonal. Define $\gamma = \frac{w_{\max}}{w_{\min}}$ to be the ratio of the largest and smallest weight. If the initial estimates for all the factors are initialized randomly from the unit sphere and the factors $\{A_j, j \geq i + 1\}$ are re-randomized after $i(\log k + \log \log d)$ steps where i is an integer, then with high probability the orthogonalized ALS updates converge to the true factors in $O(k(\log k + \log \log d))$ steps, and the error at convergence satisfies $\|A_i - \hat{A}_i\|_2^2 \leq O(\gamma k/d^2)$ and $|1 - \frac{\hat{w}_i}{w_i}| \leq O(1/d)$ for all i .

Proof. Without loss of generality, we assume that the i th recovered factor converges to the i th true factor. As mentioned earlier, the iterations for the first factor are the usual tensor power method updates and are unaffected by the remaining factors. Therefore to show that orthogonalized ALS recovers the first factor, we only need to analyze the tensor method updates. We show that the tensor power method with random initialization converges in $O(\log k + \log \log d)$ steps with

probability at least $(1 - 1/k^{1+\epsilon})$, for any $\epsilon > 0$. Hence this implies that Orth-ALS correctly recovers the first factor in $O(\log k + \log \log d)$ steps with probability at least $(1 - 1/k^{1+\epsilon})$, for any $\epsilon > 0$.

The main idea of our proof of convergence of the tensor power method is the following – with decent probability, there is some separation between the correlations of the factors with the random initialization. By the tensor power method updates (Eq. A.1), this gap is amplified at every stage. We analyze the updates for all the factors together by a simple recursion. We then show that this recursion converges in $O(\log k + \log \log d)$ steps.

Let Z_t be the iterate of the tensor power method updates at time t . Without loss of generality, we will be proving convergence to the first factor A_1 . Let $a_{i,t}$ be the correlation of the i th factor A_i with Z_t , i.e. $a_{i,t} = \langle A_i, Z_t \rangle$ (note that this should technically be called the inner product, but we refer to it as the correlation). We will refer to $w_i a_{i,t}$ as the weighted correlation of the i th factor.

The first step of the proof is that with decent probability, there is some separation between the weighted correlation of the factors with the initial random estimate. This is Lemma 1.

Lemma 1. *If $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$ for some $\epsilon > 0$, then with probability at least $(1 - \frac{\log^5 k}{k^{1+\epsilon}})$, $\frac{|w_i a_{i,0}|}{\max_i |w_i a_{i,0}|} \leq 1 - 5/k^{1+\epsilon}$ $\forall i \neq \arg \max_i |w_i a_{i,0}|$.*

The proof of Lemma 1 is a bit technical, but relies on basic concentration inequalities for Gaussians. Then using Eq. A.1 the correlation at the end of the $(t + 1)$ th time step is given by

$$a_{i,t+1} = w_i a_{i,t}^2 / \kappa_t$$

where $\kappa_t = \|\sum_{i=1}^k w_i \langle Z_{t-1}, A_i \rangle^2\|_2$ is the normalizing factor at the t th time step.

Because the estimate is normalized at the end of the updates, we only care about the ratio of the correlations of the factors with the estimate rather than the magnitude of the correlations themselves. Hence, it is convenient to normalize all the correlations by the correlation of the largest factor and normalize all the weights by the weight of the largest factor. Therefore, let $\hat{a}_{i,t} = \frac{a_{i,t}}{a_{1,t}}$ and $\hat{w}_i = \frac{w_i}{w_1}$. The new update equation for the ratio of correlations $\hat{a}_{i,t}$ is-

$$\hat{a}_{i,t+1} = \hat{w}_i \hat{a}_{i,t}^2 \tag{A.2}$$

Our goal is to show that $a_{i,t}$ becomes small for all $i \neq 1$ in $O(\log k + \log \log d)$ steps. Instead of separately analyzing the different $a_{i,t}$ for different factors A_i , we upper bound $a_{i,t}$ for all i via a simple recursion. Consider the recursion,

$$\beta_0 = \max_{i \neq 1} |\hat{w}_i \hat{a}_{i,0}| \tag{A.3}$$

$$\beta_{t+1} = \beta_t^2 \tag{A.4}$$

We claim that $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t$ for all t and $i \neq 1$. By Eq. A.3, this is true for $t = 0$ by definition. We prove our claim via induction. Assume that $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t$ for $t = p$. Note that by Eq. A.2, $\hat{w}_i \hat{a}_{i,p+1} = \hat{w}_i^2 \hat{a}_{i,p}^2$. Therefore $w_i \hat{a}_{i,p+1} \leq \beta_{p+1}^2$ for all $i \neq 1$. Hence $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t$ for all t and $i \neq 1$. Note that as the weights lie in the interval $[1, \gamma]$, $\hat{a}_{i,t} \leq \beta_t$.

To show convergence, we will now analyze the recursion in Eq. A.3. We will show that β_t becomes sufficiently small in $O(\log k + \log \log d)$ steps. Note that $\beta_t = (\beta_0)^{2^t}$ and $\beta_0 \leq 1 - 5/k^{1+\epsilon}$. Therefore $\beta_t \leq 0.1$ for $t = 2 \log k$. In another $\log \log d$ steps, $\beta_t \leq 1/d$. Hence $\beta_t \leq 1/d$ in $O(\log k + \log \log d)$ steps. As β_t is an upper bound for the correlation of all but the first factor, hence $|\hat{a}_{i,t}| \leq 1/d$ for all $i \neq 1$ in $O(\log k + \log \log d)$ steps.

To finish the proof of convergence for the tensor power method, we need to show that the estimate Z_t is close to A_1 if it has small correlation with every factor other than A_1 . Lemma 2 shows that if the ratio of the correlation of every other factor with A_1 is small, then the residual ℓ_2 error in estimating A_1 is also small.

Lemma 2. *Let $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$. Without loss of generality assume convergence to the first factor A_1 . Define $\hat{a}_{i,t} = \frac{a_{i,t}}{a_{1,t}}$ - the ratio of the correlation of the i th and 1st factor with the iterate at time t . If $\hat{a}_{i,t} \leq 2\eta \forall i \neq 1$, then $\|A_1 - \hat{A}_1\|_2^2 \leq 10\gamma k \eta^2$ in the subsequent iteration. Also, if $\|A_1 - \hat{A}_1\|_2^2 \leq O(\gamma k \eta^2)$ the relative error in the estimation of the weight w_1 is at most $O(\eta)$.*

Using Lemma 2, it follows that the estimate \hat{A}_1 and \hat{w}_1 for the factor A_1 satisfies $\|A_1 - \hat{A}_1\|_2^2 \leq 10\gamma k/d^2$ and $|1 - \frac{\hat{w}_1}{w_1}| \leq O(1/d)$. Hence we have shown that Orth-ALS correctly recovers the first factor.

We now prove that Orth-ALS also recovers the remaining factors. The proof proceeds by induction. We have already shown that the base case is correct and the algorithm recovers the first factor. We next show that if the first $(m-1)$ factors have converged, then the m th factor converges in $O(\log k + \log \log d)$ steps with failure probability at most $\tilde{O}(1/k^{1+\epsilon})$. The main idea is that as the factors have small correlation with each other, hence orthogonalization does not affect the factors which have not been recovered but ensures that the m th estimate never has high correlation with the factors which have already been recovered. Recall that we assume without loss of generality that the i th recovered factor X_i converges to the i th true factor, hence $X_i = A_i + \hat{\Delta}_i$ for $i < m$, where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k/d^2$. This is our induction hypothesis, which is true for the base case as we just showed that the tensor power method updates converge with residual error at most $10\gamma k/d^2$.

Let $X_{m,t}$ denote the m th factor estimate at time t and let Y_m denote it's value at convergence. We will first calculate the effect of the orthogonalization step on the correlation between the factors and the estimate $X_{m,t}$. Let $\{\bar{X}_i, i < m\}$ denote an orthogonal basis for $\{X_i, i < m\}$. The basis $\{\bar{X}_i, i < m\}$ is calculated via QR decomposition, and can be recursively written down as follows,

$$\bar{X}_i = \frac{X_i - \sum_{j < i} \bar{X}_j^T X_i \bar{X}_j}{\|X_i - \sum_{j < i} \bar{X}_j^T X_i \bar{X}_j\|_2} \quad (\text{A.5})$$

Note that the estimate $X_{m,t}$ is projected orthogonal to this basis. Define $\bar{X}_{m,t}$ as this orthogonal projection, which can be written down as follows –

$$\bar{X}_{m,t} = X_{m,t} - \sum_{j < m} \bar{X}_j^T X_{m,t} \bar{X}_j$$

In the QR decomposition algorithm $\bar{X}_{m,t}$ is also normalized to have unit norm but we will ignore the normalization of $X_{m,t}$ in our analysis because as before we only consider ratios of correlations of the true factors with $\bar{X}_{m,t}$, which is unaffected by normalization.

We will now analyze the orthogonal basis $\{\bar{X}_i, i < m\}$. The key idea is that the orthogonal basis $\{\bar{X}_i, i < m\}$ is close to the original factors $\{A_i, i < m\}$ as the factors are incoherent. Lemma 3 proves this claim.

Lemma 3. *Consider a stage of the Orthogonalized ALS iterations when the first $(m-1)$ factors have converged. Without loss of generality let $X_i = A_i + \hat{\Delta}_i, i < m$, where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k\eta^2$. Let $\{\bar{X}_i, i < m\}$ denote an orthogonal basis for $\{X_i, i < m\}$ calculated using Eq. A.5. Then,*

1. $\bar{X}_i = A_i + \Delta_i, \forall i < m$ and $\|\Delta_i\|_2 \leq 10k\eta$.
2. $|A_j^T \Delta_i| \leq 3\eta, \forall i < m, j < i$.
3. $|A_j^T \Delta_i| \leq 20\gamma k\eta^2, \forall i < m, j > i$.

Using Lemma 3, we will find the effect of orthogonalization on the correlations of the factors with the iterate $X_{m,t}$. At a high level, we need to show that the iterations for the factors $\{A_i, i \geq m\}$ are not much affected by the orthogonalization, while the correlations of the factors $\{A_i, i < m\}$ with the estimate $X_{m,t}$ are ensured to be small. Lemma 3 is the key tool to prove this, as it shows that the orthogonalized basis is close to the true factors.

We will now analyze the inner product between $\bar{X}_{m,t}$ and factor A_i . This is given by-

$$A_i^T \bar{X}_{m,t} = A_i^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_i^T \bar{X}_j$$

As earlier, we normalize all the correlations by the correlation of the largest factor, let $\bar{a}_{i,t}$ be the ratio of the correlations of A_i and A_m with the orthogonalized estimate $\bar{X}_{m,t}$ at time t . We can write $\bar{a}_{i,t}$ as-

$$\bar{a}_{i,t} = \frac{A_i^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_i^T \bar{X}_j}{A_m^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_m^T \bar{X}_j}$$

We can multiply both sides by \hat{w}_i and substitute \bar{X}_j from Lemma 3 and then rewrite as follows-

$$\hat{w}_i \bar{a}_{i,t} = \frac{\hat{w}_i A_i^T X_{m,t} - \sum_{j < m} \hat{w}_i X_{m,t}^T (A_j + \Delta_j) A_i^T \Delta_j}{A_m^T X_{m,t} - \sum_{j < m} X_{m,t}^T (A_j + \Delta_j) A_m^T \Delta_j}$$

We divide the numerator and denominator by $A_m^T X_{m,t}$ to derive an expression in terms of the ratios of correlations. Let

$$\delta_{i,t} = \frac{X_{m,t}^T \Delta_j}{X_{m,t}^T A_m}.$$

$$\hat{w}_i \bar{a}_{i,t} = \frac{\hat{w}_i \hat{a}_{i,t} - \sum_{j < m} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{i,t}) A_i^T \Delta_j}{1 - \sum_{j < m} (\hat{a}_{j,t} + \delta_{i,t}) A_m^T \Delta_j}$$

We now need to show $\hat{w}_i \bar{a}_{i,t}$ is small for all $i < m$ and is close to $\hat{w}_i a_{i,t}$, the weighted correlation before orthogonalization, for all $i > m$. Lemma 4 proves this, and shows that the weighted correlation of factors which have not yet been recovered, $\{A_i, i \geq m\}$, is not much affected by orthogonalization, but the factors which have already been recovered, $\{A_i, i < m\}$, are ensured to be small after the orthogonalization step.

Lemma 4. *Let $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t \forall i \neq m$ at the end of the t th iteration. Let $\bar{a}_{i,t}$ be the ratio of the correlation of the i th and the m th factor with $X_{m,t}$, the iterate at time t after the orthogonalization step. Then,*

1. $|\hat{w}_i \bar{a}_{i,t}| \leq \beta_t (1 + 1/k^{1+\epsilon}), \forall i > m.$
2. $|\hat{w}_i \bar{a}_{i,t}| \leq 50\gamma k \eta \beta_t, \forall i < m.$

We are now ready to analyze the Orth-ALS updates for the m th factor. First, we argue about the initialization step. Lemma 4 shows that an orthogonalization step performed after a random initialization ensures that the factors which have already been recovered have small correlation with the orthogonalized initialization. This is where we need a periodic re-randomization of the factors which have not converged so far.

Lemma 5. *Let $X_{m,0}$ be initialized randomly and the result be projected orthogonal to the $(m-1)$ previously estimated factors, let these be $\{X_i, i < m\}$ without loss of generality. Then $\arg \max_i |w_i a_{i,0}| \geq m$ with high probability. Also, with failure probability at most $(1 - \frac{\log^5 k}{k^{1+\epsilon}})$, $|\frac{w_i \bar{a}_{i,0}}{\max_i \{w_i \bar{a}_{i,0}\}}| \leq 1 - 4/k^{1+\epsilon} \forall i \neq \arg \max_i |w_i a_{i,0}|$ after the orthogonalization step.*

Lemma 5 shows that with high probability, the initialization for the m th recovered factor has the largest weighted correlation with a factor which has not been recovered so far after the orthogonalization step. It also shows that the separation condition in Lemma 1 is satisfied for all remaining factors with probability $(1 - \log^5 k/k^{1+\epsilon})$.

Now, we combine the effects of the tensor power method step and the orthogonalization step for subsequent iterations to show that $X_{m,t}$ converges to A_m . Consider a tensor power method step followed by an orthogonalization step. By our previous argument about the convergence of the tensor power method, if $|\hat{w}_i \hat{a}_{i,t-1}| \leq \beta_{t-1} \forall i \neq m$ at some time $(t-1)$, then $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_{t-1}^2$ for $i \neq m$ after a tensor power method step. Lemma 4 shows that the correlation of all factors other than the m th factor is still small after the orthogonalization step, if it was small before. Combining the effect of the orthogonalization step via Lemma 4, if $|\hat{w}_i \hat{a}_{i,t-1}| \leq \beta_{t-1} \forall i \neq m$ for some time $(t-1)$, then $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_{t-1}^2 (1 + 1/k^{1+\epsilon})$ for $i \neq m$ after both the tensor power method and the orthogonalization steps. By also using Lemma 5 for the initialization, can now write the updated combined recursion analogous to Eq. A.3 and Eq. A.4, but which combines the effect of the tensor power method step and the orthogonalization step.

$$\beta_0 = \max_{i \neq 1} |\hat{w}_i \hat{a}_{i,0}| \tag{A.6}$$

$$\beta_{t+1} = \beta_t^2 (1 + 1/k^{1+\epsilon}) \tag{A.7}$$

By the previous argument, $|\hat{w}_i \bar{a}_{i,t}| \leq \beta_t$. Note that $\beta_0 \leq 1 - 4/k^{1+\epsilon}$ by Lemma 5. By expanding the recursion A.7, $\beta_t = (\beta_0 (1 + 1/k^{1+\epsilon}))^{2^t}$. Hence $\beta_t \leq 1/d$ in $2 \log k + \log \log d$ steps as was the case for the analysis for the tensor power method. This shows that the correlation of the estimate $X_{m,t}$ with all factors other than A_m becomes small in $2 \log k + \log \log d$.

We now again use Lemma 2 to argue that this implies that the recovery error is small, i.e. $\|A_m - \hat{A}_m\|_2^2 \leq 10\gamma k/d^2$ and $|1 - \frac{\hat{w}_m}{w_m}| \leq O(1/d)$.

Hence we have shown that if the first $(m-1)$ factors have converged to $X_i = A_i + \hat{\Delta}_i$ where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k/d^2$, $\forall i < m$ then the m th factor converges to $X_m = A_m + \hat{\Delta}_m$ where $\|\hat{\Delta}_m\|_2 \leq 10\gamma k/d^2$ in $O(\log k + \log \log d)$ steps with probability at least $(1 - \frac{\log^5 k}{k^{1+\epsilon}})$. This proves the induction hypothesis.

We can now do a union bound to argue that each factor converges with ℓ_2 error at most $O(\gamma k/d^2)$ in $O(k(\log k + \log \log d))$ with overall failure probability at most $\tilde{O}(1/k^{-\epsilon})$, $\epsilon > 0$. This finishes the proof of convergence of Orth-ALS for the special case of orthogonal tensors. \square

B. Global convergence of the tensor power method for incoherent tensors

In this section, we will analyze the tensor power method updates for worst-case incoherent tensors. This is a necessary step before analyzing Orth-ALS, because as was pointed out in the proof of convergence of Orth-ALS in the orthogonal tensor case, analyzing Orth-ALS updates reduces to analyzing a perturbed version of the tensor power method updates. Our convergence results for the tensor power method are interesting independent of Orth-ALS though, as they prove global convergence under random initialization. The proof idea is similar to the proof of convergence of the tensor power method in the orthogonal case, but we now need to analyze the cross-terms which come in because the factors are no longer orthogonal.

Theorem 3. Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$. Let $c_{\max} = \max_{i \neq j} |A_i^T A_j|$ and $\gamma = \frac{w_{\max}}{w_{\min}}$ be the ratio of the largest and smallest weight, and assume $\gamma c_{\max} \leq o(k^{-2})$. If the initialization $x_0 \in \mathbb{R}^d$ is chosen uniformly from the unit sphere, then with high probability the tensor power method updates converge to one of the true factors (say A_1) in $O(\log k + \log \log d)$ steps, and the error at convergence satisfies $\|A_1 - \hat{A}_1\|_2^2 \leq O(\gamma k \max\{c_{\max}^2, 1/d^2\})$ and $|1 - \frac{\hat{w}_1}{w_1}| \leq O(\max\{c_{\max}, 1/d\})$.

Proof. Without loss of generality, we will prove convergence to the first factor A_1 . The proof is similar in spirit to the proof of convergence of the tensor power method in the orthogonal case in Section A.

As in the orthogonal case, Lemma 1 states that with high probability there is some separation between the weighted correlation of the largest and second largest factors.

Lemma 1. If $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$ for some $\epsilon > 0$, then with probability at least $(1 - \frac{\log^5 k}{k^{1+\epsilon}})$, $\frac{|w_i a_{i,0}|}{\max_i |w_i a_{i,0}|} \leq 1 - 5/k^{1+\epsilon}$ $\forall i \neq \arg \max_i |w_i a_{i,0}|$.

We normalize all the correlations by the correlation of the largest factor, let $\hat{a}_{i,t+1} = \frac{a_{i,t}}{a_{1,t}}$ and normalize all the weights by the weight of the largest factor, $\hat{w}_i = \frac{w_i}{w_1}$. The new update equations in terms of the ratio of correlations $\hat{a}_{i,t}$ become-

$$\hat{a}_{i,t+1} = \frac{\hat{w}_i \hat{a}_{i,t}^2 + c_{i,1} + \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,t}^2}{1 + \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,t}^2} \quad (\text{B.1})$$

Notice that we have cross terms in Eq. B.1 as compared to Eq. A.2 in the orthogonal case, due to the correlation $c_{i,j}$ between the factors being non-zero. The goal of the analysis for the non-orthogonal case is to bound these cross-terms using the incoherence between the factors.

As in the orthogonal case, we will analyze all the correlations $\hat{a}_{i,t}$ via a single recursion. We define β_t in the non-orthogonal case keeping in mind the cross-terms because of the correlations between the factors being non-zero.

$$\beta_0 = \max_{i \neq 1} |w_i \hat{a}_{i,0}| \quad (\text{B.2})$$

$$\beta_{t+1} = \gamma c_{\max} + \beta_t^2 + 3\gamma k c_{\max} \beta_t^2 \quad (\text{B.3})$$

We now show that $|w_i \hat{a}_{i,t}| \leq \beta_t$, $\forall i \neq 1$ and all t .

Lemma 6. *If $|\hat{w}_i \hat{a}_{i,m}| \leq \beta_m$ for some time m and for all $i \neq 1$, then at time $(m+1)$ for all $i \neq 1$,*

1. $|\hat{w}_i \hat{a}_{i,m+1}| \leq \beta_{m+1}$.
2. $|\hat{a}_{i,m+1} - c_{i,1}| \leq 2\beta_m^2$

Proof. Note that by Lemma 7, $\beta_t < 1 \forall t \implies \hat{w}_i \hat{a}_{i,m}^2 \leq 1$. Therefore $\sum_j |c_{i,j} \hat{w}_j \hat{a}_{j,m}^2| \leq k c_{\max} \leq 1/k^{1+\epsilon} \forall i$. Hence we can write,

$$\frac{1}{1 + \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2} = 1 - \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 + \epsilon_1$$

where ϵ_1 is the residual term, and $|\epsilon_1| \leq \left| \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 \right|^2 \leq k^2 c_{\max}^2 \leq 1/k^2$. We can now rewrite the updates for $\hat{a}_{i,m+1}$ as-

$$\hat{a}_{i,m+1} = \left(c_{i,1} + \hat{w}_i \hat{a}_{i,m}^2 + \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 \right) \left(1 - \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 + \epsilon_1 \right)$$

Let $\rho_{i,m} = c_{i,1} + \hat{w}_i \hat{a}_{i,m}^2 + \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2$. We can write,

$$\hat{a}_{i,m+1} = c_{i,1} + \hat{w}_i \hat{a}_{i,m}^2 + \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 - \rho_{i,m} \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 + \rho_{i,m} \epsilon_1 \quad (\text{B.4})$$

$$\implies |\hat{a}_{i,m+1}| \leq |c_{i,1}| + |\hat{w}_i \hat{a}_{i,m}^2| + \left| \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 \right| + \left| \rho_{i,m} \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 \right| + |\rho_{i,m} \epsilon_1| \quad (\text{B.5})$$

We claim that $\rho_{i,m} \leq 1$. We verify this as follows,

$$\begin{aligned} \rho_{i,m} &= c_{i,1} + \hat{w}_i \hat{a}_{i,m}^2 + \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 \\ \implies |\rho_{i,m}| &\leq |c_{i,1}| + |\hat{w}_i \hat{a}_{i,m}^2| + \left| \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 \right| \\ \implies |\hat{w}_i \rho_{i,m}| &\leq \gamma |c_{i,1}| + |\hat{w}_i^2 \hat{a}_{i,m}^2| + \left| \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_i \hat{w}_j \hat{a}_{j,m}^2 \right| \\ &\leq \gamma c_{\max} + \beta_m^2 + \gamma k c_{\max} \beta_m^2 \\ &\leq \beta_{m+1} \leq 1 \\ \implies |\rho_{i,m}| &\leq 1 \end{aligned}$$

where we used the fact that the weights lie in the interval $[1, \gamma]$. Hence $|\rho_{i,m}| \leq 1$. Therefore, by Eq. B.5,

$$\begin{aligned} |\hat{a}_{i,m+1}| &\leq |c_{i,1}| + |\hat{w}_i \hat{a}_{i,m}^2| + \left| \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_j \hat{a}_{j,m}^2 \right| + \left| \sum_{j:j \neq 1} c_{1,j} \hat{w}_j \hat{a}_{j,m}^2 \right| + |\epsilon_1| \\ \implies |\hat{w}_i \hat{a}_{i,m+1}| &\leq \gamma |c_{i,1}| + |\hat{w}_i^2 \hat{a}_{i,m}^2| + \left| \sum_{j:j \neq \{i,1\}} c_{i,j} \hat{w}_i \hat{w}_j \hat{a}_{j,m}^2 \right| + \left| \sum_{j:j \neq 1} c_{1,j} \hat{w}_i \hat{w}_j \hat{a}_{j,m}^2 \right| + \gamma |\epsilon_1| \\ &\leq \gamma c_{\max} + \beta_m^2 + 3\gamma k c_{\max} \beta_m^2 = \beta_{m+1} \quad (\text{B.6}) \end{aligned}$$

To show that $|\hat{a}_{i,m+1} - c_{i,1}| \leq 2\beta_m^2$ we use Eq. B.4 and repeat the steps used to show that $|\hat{w}_i \hat{a}_{i,m+1}| \leq \beta_{m+1} \forall t$. \square

By using induction and Lemma 6, the iterates at all time t satisfy the following properties, for all $i \neq 1$,

1. $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t \forall t$.
2. $|\hat{a}_{i,t} - c_{i,1}| \leq 2\beta_{t-1}^2$

This allows us to analyze the iterations of β_t instead of keeping track of the different $a_{i,t}$. We will now analyze the recursion for β_t . The following Lemma shows that β_t becomes sufficiently small in $O(\log k + \log \log d)$ steps.

Lemma 7. $\beta_t \leq 3\gamma\eta \forall t \geq O(\log k + \log \log d)$, also $\beta_t < 1 \forall t$

Proof. We divide the updates into three stages.

1. $0.1 \leq \beta_t \leq 1 - 5/k^{1+\epsilon}$:

As $\beta_t \geq 0.1$, therefore $k\beta_t^2 \geq 1$ in this regime and hence $\gamma c_{\max} \leq \gamma k c_{\max} \beta_t^2$, and we can write-

$$\begin{aligned}\beta_{t+1} &= \gamma c_{\max} + \beta_t^2 + 3\gamma k c_{\max} \beta_t^2 \\ \beta_{t+1} &\leq \beta_t^2 + 4\gamma k c_{\max} \beta_t^2\end{aligned}$$

We claim that $\beta_t < 0.1$ for $t = O(\log d)$. To verify, note that-

$$\begin{aligned}\beta_t &\leq (\beta_0(1 + 4\gamma k c_{\max}))^{2^t} \\ &\leq \left((1 - 5/k^{1+\epsilon})(1 + 1/k^{1+\epsilon}) \right)^{2^t} \\ &\leq \left(1 - 1/k^{1+\epsilon} \right)^{2^t}\end{aligned}\tag{B.7}$$

where we used the fact that $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$. Note that $(1 - 1/k^{1+\epsilon})^{2^t} \leq 0.1$ for $t = 2 \log k$ and hence we stay in this regime for at most $2 \log k$ steps.

2. $\sqrt{\gamma\eta} \leq \beta_t \leq 0.1$:

For notational convenience, we restart t from 0 in this stage. Because $\gamma c_{\max} \leq \gamma\eta \leq \beta_t^2$ in this regime and $3\gamma k c_{\max} \beta_t^2 \leq 0.1\beta_t^2$ as $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$, we can write-

$$\begin{aligned}\beta_{t+1} &= \gamma c_{\max} + \beta_t^2 + 3\gamma k c_{\max} \beta_t^2 \\ &\leq \beta_t^2 + \beta_t^2 + 0.3\beta_t^2 \leq 2.5\beta_t^2\end{aligned}$$

We claim that $\beta_t < \sqrt{\gamma\eta}$ for $t = O(\log \log \gamma\eta^{-1})$. To verify, note that-

$$\beta_t \leq (2.5\beta_0)^{2^t} \leq (0.25)^{2^t}\tag{B.8}$$

Note that $(0.25)^{2^t} \leq \sqrt{\gamma\eta}$ for $t = O(\log \log(\gamma\eta)^{-1})$ and hence we stay in this stage for at most $O(\log \log(\gamma\eta)^{-1})$ steps. As $\eta^{-1} = O(d)$, this stage continues for at most $O(\log \log d)$ steps.

3. Note that in the next step, $\beta_t \leq \gamma c_{\max} + 1.1\gamma\eta \leq 3\gamma\eta$. This is again because $3\gamma k c_{\max} \beta_t^2 \leq 0.1\beta_t^2$ and $\beta_t \leq \sqrt{\gamma\eta}$ at the end of the previous stage.

□

Hence $\beta_t \leq 3\gamma\eta$ for $t = O(\log \log d + \log k)$. By Lemma 6, $|\hat{a}_{i,t} - c_{i,1}| \leq 18\gamma^2\eta^2, i \neq 1$. Hence $|\hat{a}_{i,t}| \leq 2\eta$. By Lemma 2, the error at convergence satisfies $\|A_1 - \hat{A}_1\|_2 \leq 10\gamma k \eta^2$ and the estimate of the weight \hat{w}_1 satisfies $|1 - \frac{\hat{w}_1}{w_1}| \leq O(\eta)$. □

C. Global convergence of the tensor power method for random tensors

The previous section gives global convergence guarantees for the tensor power method for incoherent tensors. Applying Theorem 3 to a tensor whose factors are chosen uniformly at random, we can say that the tensor power method converges with random initialization whenever the rank $k = o(d^{0.25})$. Theorem 3 also proves a linear convergence rate. However, this is quite suboptimal for random tensors. In this section, we use the randomness in the tensor to get much stronger convergence results.

The techniques used in this section are very different from the rest of the paper. Instead of recursively analyzing the tensor power method updates by showing that the algorithm makes progress at every step by boosting its correlation with some fixed factor, we directly express the correlation of the factors with the estimate Z_τ after a fixed number of $\tau = O(\log \log d)$ time steps in terms of the initial correlations of the factors with the random initialization. This allows us to then skillfully use the randomness in the factors to get strong results. The difficulty with the recursive approach is that all the randomness in the tensor is “lost” after just one tensor power method update, i.e. the correlations of different factors with the estimate are no longer independent of each other, which makes the analysis much more difficult.

Theorem 2. *Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$ with the factors A_i sampled uniformly from the d -dimensional sphere. Define $\gamma = \frac{w_{\max}}{w_{\min}}$ to be the ratio of the largest and smallest weight. Assume $k \leq o(d)$ and $\gamma \leq \text{polylog}(d)$. If the initialization $x_0 \in \mathbb{R}^d$ is chosen uniformly from the unit sphere, then with high probability the tensor power method updates converge to one of the true factors (say A_1) in $O(\log \log d)$ steps, and the error at convergence satisfies $\|A_1 - \hat{A}_1\|_2 \leq \tilde{O}(1/\sqrt{d})$. Also, the estimate of the weight \hat{w}_1 satisfies $|1 - \frac{\hat{w}_1}{w_1}| \leq \tilde{O}(1/\sqrt{d})$.*

Proof. Without loss of generality, we will prove convergence to the first factor A_1 . Let $\tau = 5 \log \log d^2$. As before, define $a_{i,t} = \langle A_i, Z_t \rangle$ where Z_t is the iterate at time t . For the analysis of the tensor power method updates for random tensors we ignore the normalization step of the updates, till the last iteration. It is easy to see that this makes no difference in the analysis, though in practice it is important to normalize after every step to prevent the vectors from becoming too small and causing numerical errors. Recall that the update equations for $a_{i,t}$ for any t are-

$$a_{i,t} = w_i a_{i,t-1}^2 + c_{i,1} w_1 a_{1,t-1}^2 + \sum_{j:j \neq \{i,1\}} c_{i,j} w_j a_{j,t-1}^2 \quad (\text{C.1})$$

and the iterate $X_{\tau+1}$ at time $\tau + 1$ can be written as

$$Z_{\tau+1} = w_1 a_{1,\tau}^2 A_1 + \sum_{i \neq 1} w_i a_{i,\tau}^2 A_i$$

On expanding $w_1 a_{1,\tau}^2$ recursively using Eq. C.1, one of the terms that appears in the expansion is $(w_1 a_{1,0})^{2^\tau} / w_1$. Define $\alpha_\tau = |w_1 a_{1,0}|^{2^\tau} / w_1$. Let $\Delta_\tau = (1/\alpha_\tau) \sum_{i \neq 1} w_i a_{i,\tau}^2 A_i$. We show that $\|\Delta_\tau\|_2 \leq \tilde{O}(1/\sqrt{d})$ with failure probability at most $\log^{-1} d$. We can write $(1/\alpha_\tau) w_1 a_{1,\tau}^2$, the coefficient for first factor A_1 normalized by α_τ , as follows

$$\begin{aligned} \frac{w_1 a_{1,\tau}^2}{\alpha_\tau} &= \frac{(w_1 a_{1,0})^{2^\tau}}{w_1 \alpha_\tau} + \frac{1}{\alpha_\tau} \left(w_1 a_{1,\tau}^2 - \frac{(w_1 a_{1,0})^{2^\tau}}{w_1} \right) \\ &= 1 + \frac{1}{\alpha_\tau} \left(w_1 a_{1,\tau}^2 - \frac{(w_1 a_{1,0})^{2^\tau}}{w_1} \right) \end{aligned}$$

Let $\lambda_\tau = \frac{1}{\alpha_\tau} \left(w_1 a_{1,\tau}^2 - \frac{(w_1 a_{1,0})^{2^\tau}}{w_1} \right)$. Let $Z'_{\tau+1} = Z_{\tau+1} / \alpha_\tau$. We can write $Z'_{\tau+1}$ as

$$Z'_{\tau+1} = (1 + \lambda_\tau) A_1 + \Delta_\tau$$

Note that $\frac{Z'_{\tau+1}}{\|Z'_{\tau+1}\|_2} = \frac{Z_{\tau+1}}{\|Z_{\tau+1}\|_2}$. Let $\frac{Z'_{\tau+1}}{\|Z'_{\tau+1}\|_2} = \tilde{Z}_{\tau+1}$. We desire to show that the residual $\|\tilde{Z}_{\tau+1} - A_1\|_2 \leq \tilde{O}(1/\sqrt{d})$. We can bound $\|\tilde{Z}_{\tau+1} - A_1\|_2$ as follows using the triangle inequality,

$$\|\tilde{Z}_{\tau+1} - A_1\|_2 \leq \left| \frac{1 + \lambda_\tau}{\|(1 + \lambda_\tau)A_1 + \Delta_\tau\|_2} - 1 \right| + \frac{\|\Delta_\tau\|_2}{\|(1 + \lambda_\tau)A_1 + \Delta_\tau\|_2}$$

If $\|\Delta_\tau\|_2 \leq \tilde{O}(1/\sqrt{d})$ and $|\lambda_\tau| \leq d^{-\epsilon}$ then,

$$\begin{aligned} \|\tilde{Z}_{\tau+1} - A_1\|_2 &\leq \left| \frac{1}{\|A_1 + \Delta_\tau/(1 + \lambda_\tau)\|_2} - 1 \right| + \frac{\|\Delta_\tau\|_2}{1 - |\lambda_\tau| - \|\Delta_\tau\|_2} \\ &\leq \frac{2\|\Delta_\tau\|_2}{1 - |\lambda_\tau|} + \frac{\|\Delta_\tau\|_2}{1 - |\lambda_\tau| - \|\Delta_\tau\|_2} \leq \tilde{O}(1/\sqrt{d}) \end{aligned}$$

If $\|\tilde{Z}_{\tau+1} - A_1\|_2 \leq \tilde{O}(1/\sqrt{d})$ then, by Lemma 2, the estimate of the weight \hat{w}_1 satisfies $|1 - \frac{\hat{w}_1}{w_1}| \leq \tilde{O}(1/\sqrt{d})$.

Hence we will show that $\|\Delta_\tau\|_2 \leq \tilde{O}(1/\sqrt{d})$ and $|\lambda_\tau| \leq d^{-\epsilon}$ with failure probability at most $\log^{-1} d$. Let $\epsilon_\tau = \|\Delta_\tau\|_2^2$. We can write ϵ_τ as

$$\epsilon_\tau = \|\Delta_\tau\|_2^2 = \sum_{i \neq 1, j \neq 1} (1/\alpha_\tau^2) w_i w_j a_{i,\tau}^2 a_{j,\tau}^2 c_{i,j}$$

We can also write λ_τ^2 as follows-

$$\lambda_\tau^2 = (1/\alpha_\tau^2) w_1^2 \left(a_{1,\tau}^2 - \frac{(w_1 a_{1,0})^{2\tau}}{w_1^2} \right)^2$$

Note that λ_τ has the same form as ϵ_τ with the restriction that $i = j = 1$ and the $\frac{(w_1 a_{1,0})^{2\tau}}{w_1^2}$ in the expansion of $a_{1,\tau}^2$ is removed.

Our approach will be to recursively expand the $a_{i,\tau}^2$ terms to express ϵ_τ and λ_τ only in terms of $a_{i,0}$ (the initial correlations at time 0), the correlation between factors $c_{i,j}$ and the weights w_i . We use the recursion Eq. C.1 to do this.

We first consider the expansion of $a_{i,\tau}^2$ for any i using recursion Eq. C.1. $a_{i,t}^2$ can be written as a weighted sum of correlations of the factors with the iterate at the $(t-1)$ st time step as follows using recursion Eq. C.1-

$$a_{i,t}^2 = \left(w_i a_{i,t-1}^2 + \sum_{j \neq i} c_{i,j} w_j a_{j,t-1}^2 \right)^2 \quad (\text{C.2})$$

$$= \sum_{j,k} w_j w_k c_{i,j} c_{i,k} a_{j,t-1}^2 a_{k,t-1}^2 \quad (\text{C.3})$$

Each term in the summation corresponds to two choices for the terms at time $(t-1)$, the j and k variables. Continuing this recursive expansion for τ time steps, we can represent each monomial in the expansion by a complete binary tree with depth τ . We label a node of the binary tree as j if it corresponds to factor A_j . For ease of exposition, we will consider the initialization Z_0 as the 0th factor for the graph representation, hence $c_{i,0} = a_{i,0}$. The root of the tree is labeled as i as it corresponds to the factor A_i . The descendants of the root i are labelled as j and k if $a_{i,\tau}^2$ is expanded into $a_{j,\tau-1}^2$ and $a_{k,\tau-1}^2$ using recursion Eq. C.1. The process is repeated at any step of the recursion, by expanding $a_{i,t}^2$ in terms of $a_{k,t-1}^2$ and $a_{l,t-1}^2$ for some k and l . Refer to Fig. 4 for an example of a monomial and its binary tree representation.

Given any complete binary tree B , the monomial associated with the tree can be written down recursively. We write down the procedure for finding the monomial corresponding to a binary tree B explicitly in Algorithm 3 for clarity.

Therefore, by successively using Eq. C.1, we expand $w_i a_{i,1}^2$ in terms of the correlations of the factors with the random initialization Z_0 (the $a_{j,0}^2$ factors) and define a complete binary tree B_f for every monomial f in the expansion. We also define a graph G_f for the monomial f by coalescing nodes of the binary tree having the same label and removing self-loops. We allow more than one edge between two nodes.

For any monomial f in the expansion of $(1/\alpha_\tau^2) w_i w_j a_{i,\tau}^2 a_{j,\tau}^2 c_{i,j}$ in ϵ_t , we construct two binary trees corresponding to the expansion of $w_i a_{i,\tau}^2$ and $w_j a_{j,\tau}^2$. We construct the graph G_f by adding an edge between the roots of the two binary trees (this corresponds to the $c_{i,j}$ term) and then coalescing nodes of the new graph having the same label and removing self-loops, while allowing multiple edges between two nodes. The same procedure is followed for the expansion of λ_τ^2 , with the difference that now $i = j = 1$, and the $\frac{(w_1 a_{1,0})^{2\tau}}{w_1^2}$ term in the expansion of $a_{1,\tau}^2$ is removed.

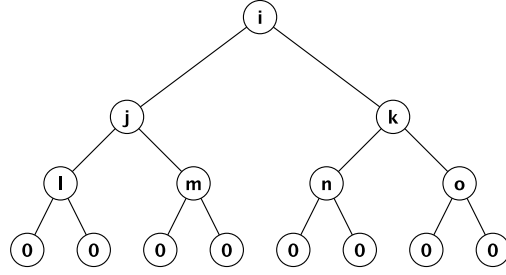


Figure 4. Example of a monomial in the expansion of $w_i a_{i,2}^2$ represented as a binary tree. The monomial is $c_{i,j} c_{i,j} c_{j,l} c_{j,m} c_{i,k} c_{k,n} c_{k,o} w_i w_j w_l w_m w_k w_n w_o a_{l,0}^2 a_{m,0}^2 a_{n,0}^2 a_{o,0}^2$

Algorithm 3 Finding monomial f from binary tree B

Input: Binary tree B , root u

monomial(B, u)

- 1: **while** u is not a leaf **do**
 - 2: Set i to be the factor corresponding to u
 - 3: Set v to be the left child of u , set j to be the factor corresponding to v
 - 4: Set w to be the right child of u , set j to be the factor corresponding to w
 - 5: $f = f w_i$
 - 6: $f = f c_{i,j} \text{monomial}(B, v)$
 - 7: $f = f c_{i,j} \text{monomial}(B, w)$
 - 8: **end while**
 - 9: **return** f
-

C.1. Choosing a suitable basis for the factors

Without loss of generality, assume that the first $(n-1)$ factors are present in G_f , for some n . The $(n-1)$ vectors corresponding to the $(n-1)$ factors and the initialization Z_0 span a n dimensional subspace. We will choose a particular basis $\{v_i\}, i \in [n]$ for the n dimensional subspace and express the factors with respect to that basis. $v_1 = Z_0$, and v_i is unit vector along the projection of A_{i-1} orthogonal to $\{A_j, j < i-1\}$. In terms of this basis, $Z_0 = (1, 0, \dots, 0)$. Let the 1st factor A_1 have component $x_{1,1}$ along the first coordinate axis and $u_{1,2}$ along the second coordinate axis. Note that $x_{1,1}$ is distributed as $\frac{\tilde{x}_{1,1}}{r_1}$ and $u_{1,2}$ is distributed as $\frac{\tilde{u}_{1,2}}{r_1}$ where $\tilde{x}_{1,1} \sim N(0, 1/d)$, $\tilde{u}_{1,2} \sim v_1 \sqrt{\sum_{i=2}^d \tilde{y}_{1,i}^2}$ and $r_1 = \sqrt{\tilde{x}_{1,1}^2 + \tilde{u}_{1,2}^2}$. Here $\tilde{y}_{1,i} \sim N(0, 1/d)$ and v_1 is uniform on $\{-1, +1\}$. Similarly, the 2nd factor A_2 has components $(x_{2,1}, x_{2,2}, u_{2,3}) \sim \left(\frac{\tilde{x}_{2,1}}{r_2}, \frac{\tilde{x}_{2,2}}{r_2}, \frac{\tilde{u}_{2,3}}{r_2}\right)$ along the first three coordinate axes. Here $\tilde{x}_{2,1}, \tilde{x}_{2,2} \sim N(0, 1/d)$ and $\tilde{u}_{2,3} \sim v_2 \sqrt{\sum_{i=3}^d \tilde{y}_{2,i}^2}$ and $r_2 = \sqrt{\tilde{x}_{2,1}^2 + \tilde{x}_{2,2}^2 + \tilde{u}_{2,3}^2}$, where $\tilde{y}_{2,i} \sim N(0, 1/d)$ and v_2 is uniform on $\{-1, +1\}$. We continue this projection for all subsequent factors.

We first prove a Lemma that bounds the magnitude of the projection of any factor along the basis vectors.

Lemma 8. *The projection of n factors along the basis defined above has the following properties-*

1. $1 - \frac{1}{d^{0.25}} \leq r_i^2 \leq 1 + \frac{1}{d^{0.25}} \forall i \in [n]$ with failure probability at most $2ne^{-\sqrt{d}/8}$.
2. $|\tilde{x}_{i,j}| \leq \log^5 d / \sqrt{d}$ for all valid i, j (i.e. for all $j < i, i \in [n]$) with failure probability at most $n(\frac{1}{d})^{\log^8 d}$.

Proof. The proof relies on basic concentration inequalities.

1. Consider the vector $(x_{i,1}, \dots, u_{i,i+1}, 0 \dots, 0)$ corresponding to factor i . The squared scaling factor r_i^2 is distributed as $r_i^2 \sim (\tilde{x}_{i,1}^2 + \dots + \tilde{x}_{i,i}^2 + \tilde{u}_{i,i+1}^2)$, where $\tilde{u}_{i,i+1}^2 \sim \tilde{y}_{i,i+1}^2 + \dots + \tilde{y}_{i,d}^2$, the $\tilde{y}_{i,j}$ are independent $N(0, 1/d)$ random variables. r_i^2 is the sum of squares of independent zero mean Gaussian random variables each having variance $1/d$, and hence $x_i^2 = d r_i^2$ is a χ^2 random variable with d degrees of freedom. We use the following tail bound on a χ^2

random variable x with d degrees of freedom (the bound follows from the sub-exponential property of the χ^2 random variable)

$$\mathbb{P}[|x^2 - d| \geq dt] \leq 2e^{-dt^2/8}$$

Choosing $t = d^{-0.25}$, $\mathbb{P}[|x_i^2 - d| \geq d^{0.75}] \leq 2e^{-\sqrt{d}/8}$. Therefore $\mathbb{P}[|r_i - 1| \geq d^{-0.25}] \leq 2e^{-\sqrt{d}/8}$. By a union bound, $|r_i^2 - 1| \leq \frac{1}{d^{0.25}} \forall i \in [n]$ with failure probability at most $2ne^{-\sqrt{d}/8}$.

2. The bound follows directly from basic Gaussian tail bounds (refer to Eq. E.1) and a union bound. □

Note that as $\tau = 5 \log \log d^2$, the total number of nodes of the binary tree corresponding to a monomial is at most $2^{\tau+1} = 2 \log^5 d^2$. As each monomial corresponds to two binary trees, the number of number in the graph G_f can be at most $4 \log^5 d^2$. Let $N = 4 \log^5 d^2$. We can now use a union bound to argue that the properties of the factors in Lemma 8 hold with high probability for any set of N factors. We define $\beta_0 = \max \left\{ \left| \frac{w_i x_{i,1}}{w_1 x_{1,1}} \right|, i \neq 1 \right\}$ and $\beta_t = \beta_0^{2^t}$ for any t .

Lemma 9. *Consider the projection of any set of $N = 4 \log^5 d^2$ factors. With failure probability at most $1/d^{\log d}$, $|x_{i,j}| \leq 2 \log^3 d / \sqrt{d}$ for all valid i, j (i.e. for all $j < i, i \in [N]$). Also, with failure probability at most $1/\log d$, $\beta_0 \leq 1 - 1/\log^4 k$.*

Proof. Using Lemma 8 and a union bound, $|\tilde{x}_{i,j}| \leq (\log d)^5 / \sqrt{d}$ for all valid i, j and $|r_i^2 - 1| \leq \frac{1}{d^{0.25}} \forall i \in [N]$ with failure probability at most $N(\frac{1}{d})^{\log^8 d} + 2Ne^{-\sqrt{d}/8} \leq 2N(\frac{1}{d})^{\log^8 d}$. As $x_{i,j} = \tilde{x}_{i,j}/r_i$, therefore $x_{i,j} \leq 2 \log^5 d / \sqrt{d}$ whenever $\tilde{x}_{i,j} \leq \log^5 d / \sqrt{d}$ and $r_i \geq 1 - \frac{1}{d^{0.25}}$. Therefore, as the total number of sets of N factors is at most $k^N \leq d^N$, by doing a union bound over all possible sets of N factors, $|x_{i,j}| \leq 2 \log^5 d / \sqrt{d}$ for all valid i, j with failure probability at most $2d^N N(\frac{1}{d})^{\log^8 d} \leq 1/d^{\log d}$.

Using Lemma 14, with failure probability at most $1/\log d$, $\left| \frac{w_i \tilde{x}_{i,1}}{w_1 \tilde{x}_{1,1}} \right| \leq 1 - 1/\log^5 k$ for all $i \neq 1$. As $|r_i^2 - 1| \leq \frac{1}{d^{0.25}} \forall i \in [k]$ with failure probability at most $2ke^{-\sqrt{d}/8}$, therefore with failure probability at most $2/\log d$, $\left| \frac{w_i x_{i,1}}{w_1 x_{1,1}} \right| \leq 1 - 0.5/\log^5 k$ for all $i \neq 1$. □

Let E be the event that for any projection of up to N factors $|x_{i,j}| \leq 2 \log^3 d / \sqrt{d}$ for all valid i, j (i.e. for all $j < i, i \in [n]$) and $\beta_0 \leq 1 - 1/\log^4 k$. By Lemma 9, probability of the event E is at least $(1 - 3/\log d)$. We condition on the event E for the rest of the proof.

C.2. Characterizing when the monomial has non-zero expectation

Let f_2 refer to the product of all $a_{i,0}^2$ terms, all the weights w_i for any i appearing in f and $1/\alpha_\tau^2$. Let f_1 refer to all the terms in f not present in f_2 , hence $f = f_1 f_2$. Let G'_f be the graph obtained by removing the node corresponding to the initialization X_0 and all its edges from G_f . Note that G'_f is a connected graph, as the 0th factors only appears in the leaves of the binary tree.

As the $c_{i,j}$ terms are inner products between the factors, we can write $c_{i,j}$ in terms of the co-ordinates of the vectors A_i and A_j , in terms of the basis we described previously. Note that $a_{i,0} = x_{i,1}$ hence there is only one term in the inner product $a_{i,0}$. f_1 is a product of the cross-correlation terms $c_{i,j}$, hence it can be written as the summation of a product of a choices of coordinate for every $c_{i,j}$ term. Let the terms obtained on rewriting f_1 in terms of the coordinates of the vectors be g_i , hence $f_1 = \sum_{i=1}^K g_i$.

Lemma 10. *f has non-zero expectation only if G_f is Eulerian. Also, every term g_i having non-zero expectation corresponds to choosing a split of G'_f into a disjoint union of cycles and then choosing a single coordinate for all inner products $c_{i,j}$ which are part of a particular cycle.*

Proof. We claim that every node in G_f must have even degree for f to have non-zero expectation. To verify, consider any node j which has odd degree. Note that the 0th node corresponding to the initialization X_0 always has even degree, hence

$j \neq 0$. $\mathbb{E}[f]$ is the expectation of the product of all correlation terms $c_{i,j}$ and $a_{i,0}$ appearing in the monomial. Each inner product $c_{i,j}$ involves a $x_{i,t}$ term or $u_{i,t}$ term for some coordinate t . Hence if node i has odd degree, then there is at least some t such that $x_{i,t}$ or $u_{i,t}$ is raised to an odd power. Note that the sign of $x_{i,t}$ or $u_{i,t}$ is an independent zero mean random variable, hence the expectation evaluates to 0 in this case. Hence every node in G_f must have even degree for f to have non-zero expectation. By Euler's theorem every node in a graph has even degree if and only if the graph is Eulerian (there exists a trail in the graph which uses every edge exactly once and returns to its starting point). Also, an Eulerian graph can be written as a disjoint union of cycles (Veblen's theorem).

G'_f is also Eulerian and can be written as a disjoint union of cycles as every node has an even number of edges to node 0 and hence removal of these edges preserves the Eulerian property.

We now prove the second part of the Lemma, that every term g_i having non-zero expectation corresponds to choosing a split of G'_f into a disjoint union of cycles and then choosing a single coordinate for all inner products $c_{i,j}$ which are part of a particular cycle. To verify this, let's start at any node i and consider its inner product with a neighbor j . Say we choose coordinate t for the inner product $c_{i,j}$ which leads to a $x_{i,t}x_{j,t}$ term in g_i . To ensure that g_i has non-zero expectation, $x_{j,t}$ must appear in the term an even number of times (as the sign of $x_{j,t}$ is an independent zero mean random variable). Hence the coordinate t must be chosen in the inner product of node j with some neighbor of j . By repeating this argument, there must exist a cycle C with node i such that the coordinate t is chosen for all correlation terms in that cycle C . We then repeat the process on the graph obtained by removing the edges corresponding to cycle C from G'_f . Hence every g_i term having non-zero expectation corresponds to choosing a split of G'_f into a disjoint union of cycles and then choosing a single coordinate for all inner products $c_{i,j}$ which are part of a particular cycle. \square

We let $f'_1 = \sum_{i:\mathbb{E}[g_i] \neq 0} g_i$ and $f' = f'_1 f_2$. We claim that $\mathbb{E}[f] = \mathbb{E}[f']$. Consider any term g_i , such that $\mathbb{E}[g_i] = 0$. We claim that $\mathbb{E}[g_i f_2]$ also equals 0, hence $\mathbb{E}[f] = \mathbb{E}[f']$. This is because if g_i has zero expectation, then there is some $x_{i,t}$ term raised to an odd power, as otherwise the expectation is non-zero. But, as all terms are raised to an even power in f_2 , the $x_{i,t}$ term is also raised to an odd power in $g_i f_2$, which implies that $\mathbb{E}[g_i f_2] = 0$. This verifies the claim that $\mathbb{E}[g_i f_2] = 0$ if $\mathbb{E}[g_i] = 0$.

C.3. Bounding expected value of monomial

We are now ready to bound the expected value of f . Note that $\mathbb{E}[f] = 0$ if G_f is not Eulerian. If G_f and hence G'_f are Eulerian, split G'_f into some disjoint union of cycles. Say we split G'_f into p cycles $\{C_1, C_2, \dots, C_p\}$ with m_1, m_2, \dots, m_p edges. Let $D(C_j)$ refer to the choice of coordinate $D(C_j)$ for cycle C_j . Let $g(\cup_j C_j(D(C_j)))$ be the term in the expansion of f corresponding to a split of G_f into cycles $\{C_1, C_2, \dots, C_p\}$ and the choice of coordinate $D(C_j)$ for cycle C_j . We also define $h(C_j(D(C_j)))$ as the product of terms corresponding to cycle C_j and the choice of coordinate $D(C_j)$ for the cycle C_j . Note that $g(\cup_j C_j(D(C_j))) = \prod_{j=1}^p h(C_j(D(C_j)))$. We can write

$$g(\cup_j C_j(D(C_j))) = \prod_j h(C_j(D(C_j)))$$

$h(C_j(D(C_j)))$ is the product of the square of the $D(C_j)$ -th co-ordinate of all the factors appearing in the cycle C_j . Conditioned on the event E , there is only one factor having a component greater than $\log^5 d / \sqrt{d}$ in absolute value along the $D(C_j)$ -th co-ordinate axis, hence

$$h(C_j(D(C_j))) \leq \frac{(\log^{10} d)^{m_1-1}}{d^{m_1-1}}$$

Hence, conditioned on event E , we can bound $g(\cup_j C_j(D(C_j)))$ as-

$$g(\cup_j C_j(D(C_j))) \leq \frac{(\log^{10} d)^{m/2}}{d^{m-p}}$$

Let $c(G_f)$ be the largest p such that G'_f can be decomposed into a union of p disjoint cycles. There can be at most $m/2$ disjoint cycles in G'_f as there are m edges, therefore $c(G_f) \leq m/2$. Each edge can be placed in one of the total number of possible cycles, hence the total number of ways of splitting G'_f into a disjoint union of cycles is at most $(m/2)^m$. Also, there are n possible choices for a coordinate for each cycle, hence there are at most $n^{(m/2)}$ terms corresponding to the same split of G'_f into a disjoint union of cycles. Hence for any particular monomial f , the number of possible g_i terms having

non-zero expectation is at most $(m/2)^m n^{(m/2)}$. Note that $m \leq 2 \log^5 d^2$ as the graph G'_f is constructed by collapsing the two binary trees corresponding to monomial f . Each binary tree has depth $\tau = 5 \log \log d^2$, hence the number of edges is at most $2 \log^5 d^2$. Hence the total number of edges in graph G'_f is at most $4 \log^5 d^2 \leq \log^6 d$. Hence we can bound $\mathbb{E}_{|E}[f]$ as-

$$\mathbb{E}_{|E}[f] \leq f' \leq (m/2)^m n^{(m/2)} \frac{(\log^{10} d)^{m/2}}{d^{m-c(G_f)}} f_2 \leq \frac{(\log^{10} d)^{5m/2}}{d^{m-c(G_f)}} f_2 \quad (\text{C.4})$$

We will now bound the f_2 term. Let $\theta = \max \left\{ \left| \frac{w_i x_{i,1}}{w_1 x_{1,1}} \right| \right\}$ over all nodes $i \in G'_f$. Clearly $\theta \leq \beta_0$ if node 1 is not in G and is at most 1 otherwise. We will consider the representation of the monomial f as two complete binary trees. Recall that the leaves of the binary tree correspond to the 0th factor. Each pair of leaves having the factor i as their parent corresponds to a $a_{i,0}^2$ term. We will pair every leaf node with its successor, regarding the binary tree as a binary search tree. Note that the left child of any node has the same node as its successor. Let the right child of the node with factor i have a node with factor j as its successor. We group the w_i term due to the successor of the left child and w_j term due to the successor of the right child together with the $a_{i,0}^2$ term. We bound the $w_j w_i a_{i,0}^2$ term by $\gamma(w_i a_{i,0})^2$ whenever $j \neq i$ and by $(w_i a_{i,0})^2$ when $j = i$. If all the edges from the successor to the leaf are self-loops of the form $c_{j,j}$, then $j = i$. Note that the paths of all leaves of a binary tree to their successor are disjoint, hence each cross-correlation term $c_{i,j}, i \neq j$ can lead to at most one leaf with $j \neq i$. The number of cross-correlation terms equals m , the number of edges in the graph G'_f . Recall that $\alpha_\tau = |w_1 a_{1,0}|^{2^\tau} / w_1$. Therefore the product of all the w_i and $a_{i,0}^2$ terms normalized by α_τ is at most $\gamma^m \theta^{2^\tau}$.

As an example, consider the monomial $f = w_2^2 w_1^4 (c_{1,2})^4 (a_{1,0})^8$. The binary tree B_f corresponding to f is given in Fig. 6. Both binary trees are the same in this case. The graph G'_f obtained by coalescing the two binary trees is given in Fig. 5.

1. *Projecting factors onto suitable basis:* We can write the initialization Z_0 as the vector $(1, 0 \cdots, 0)$. We write the factor A_1 as $(x_{1,1}, u_{1,2}, 0, \cdots, 0)$. Similarly, the 2nd factor A_2 has components $(x_{2,1}, x_{2,2}, u_{2,3})$. Using Lemma 9, $\max\{|x_{1,1}|, |x_{2,1}|, |x_{2,2}|\} \leq \log^5 d / \sqrt{d}$.
2. *Writing expectation of f as product of expectation of cycles:* Let $f_1 = (c_{1,2})^4$. Let $f_2 = (a_{1,0})^8 = (x_{1,0})^8$. f can be expanded by choosing a coordinate corresponding to each $c_{1,2}$ term, and then summing across all choices. Let the terms obtained on rewriting f_1 in terms of the co-ordinates of the factors A_1 and A_2 be g_i , hence $f = \sum_{j=1}^K g_i$. By Lemma 10, every term g_i having non-zero expectation corresponds to choosing a split of G'_f into a disjoint union of cycles and then choosing a single coordinate for all inner products $c_{i,j}$ which are part of a particular cycle. Say we split G'_f into the union of cycles C_1 and C_2 where C_1 and C_2 are 2 edge cycles between node 1 and node 2. Say we choose the 2nd coordinate for both the cycles C_1 and C_2 . Following the notation of subsection C.3, $D(C_1) = D(C_2) = 2$ and $g(C_1(2) \cup C_2(2))$ is the term in the expansion of f corresponding to split of G'_f into cycles C_1 and C_2 and then choosing the second coordinate for both cycles. $g(C_1(2) \cup C_2(2)) = h(C_1(2))h(C_2(2)) = x_{1,2}^4 u_{1,2}^4 \leq \log^{10} d / d^2$, again following the notation of subsection C.3. Recalling the definition of $c(G_f)$ be the largest p such that G'_f can be decomposed into a union of p disjoint cycles, for our example, $c(G_f) = 2$. As each edge can be placed in one of the two possible cycles and there are 4 edges, the total number of ways of splitting G'_f into a disjoint union of cycles is at most 2^4 . There are 2 possible choices for coordinates for each cycle as we have two factors. Hence we can bound f' and $\mathbb{E}[f]$ as -

$$\mathbb{E}[f] \leq f' \leq 2^4 4^2 \frac{(\log^{10} d)^2}{d^2} f_2 \leq \frac{(\log^{10} d)^{10}}{d^2} f_2 \quad (\text{C.5})$$

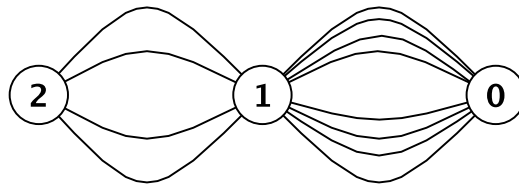
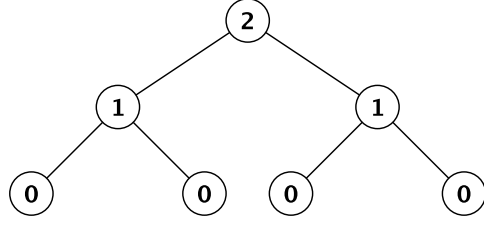


Figure 5. Graph G_f for $f = w_2^2 w_1^4 (c_{1,2})^4 (a_{1,0})^8$

 Figure 6. Binary tree B_f for $f = w_2^2 w_1^4 (c_{1,2})^4 (a_{1,0})^8$ (both binary trees for f are the same)

We are now ready to bound ϵ_τ . We will divide ϵ_τ into 2 sets of monomials and bound each one of them separately-

1. All monomials with root nodes i and j and with either no path from node i to node 1 or no path from node j to node 1. We call this set S_1 .
2. All monomials with root nodes i and j and at least two paths from node i to node 1 and at least two paths from node j to node 1. We call this set S_2 .

Note that the number of paths between two nodes in the graph G'_f is always even if f has non-zero expectation, as G'_f is Eulerian in that case. We need to relate the number of nodes and edges of an Eulerian graph for the rest of the proof, Lemma 11 provides a simple bound.

Lemma 11. *For any connected Eulerian graph G , let N be the number of nodes and M be the number of edges. Consider any decomposition of G into a edge-disjoint set of p cycles. Then, $N \leq M - p + 1$. Moreover, if G has four edge-disjoint paths between a pair of nodes then $N \leq M - p$.*

We first consider the set S_1 . As there are no paths from node i to node 1 or from node j to node 1, therefore $\theta \leq \beta_0$ for at least one of the binary trees. Therefore $f_2 \leq \gamma^m \beta_0^{2\tau} = \gamma^m \beta_\tau$. For any graph G_f with n nodes, there can be at most $k^n \leq d^n$ monomials having a graph isomorphic to G_f as their representation. By Lemma 11, $n \leq m - c(G_f) + 1$. The total number of graphs with n nodes and m edges is at most $(n^2)^m$. As the graph G'_f is connected, $n \leq m$. Note that the number of edges can be at most $4 \log^5 d^2 \leq \log^6 d$. Hence we can bound the contribution of all monomials in the set S_1 as follows-

$$\begin{aligned}
 \sum_{f: f \in S_1} \mathbb{E}[f] &\leq \sum_{m=0}^{\log^6 d} k^{m-c(G_f)+1} \frac{(m^2)^m (\log^{10} d)^{5m/2} \gamma^m}{d^{m-c(G_f)}} \beta_t \\
 &\leq \sum_{m=0}^{\log^6 d} k^{m-c(G_f)+1} \frac{(\gamma \log^{55} d)^m}{d^{m-c(G_f)}} \frac{1}{d^2} \\
 &\leq \frac{1}{d} \sum_{m=0}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{k}{d}\right)^{m-c(G_f)} \\
 &\leq \frac{1}{d} \sum_{m=0}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{1}{d^\epsilon}\right)^{m/2} \leq \frac{1}{d} \sum_{m=0}^{\infty} \left(\frac{\gamma \log^{55} d}{d^{0.5\epsilon}}\right)^m \leq \frac{2}{d}
 \end{aligned}$$

We next consider the set S_2 . For any graph G_f with n nodes with at least one of the nodes corresponding to factor A_1 , there can be at most $nk^{n-1} \leq nd^{n-1}$ monomials having a graph isomorphic to G_f as their representation as there are n possible positions to place the factor A_1 and at most d^{n-1} ways to label the remaining nodes. We claim that by Lemma 11, $n \leq m - c(G_f)$. This is because there are two paths from node i to node 1 and two paths from node j to node 1. Note that

there is always an edge between nodes i and j , as we connect the roots of the binary trees by an edge. Hence there are at least three edge-disjoint paths between nodes i and j . But there cannot be an odd number of edge-disjoint paths between 2 nodes in an Eulerian graph, hence there must be at least four edge-disjoint paths between nodes i and j . Hence by Lemma 11, $n \leq m - c(G_f)$. Also, note that the number of edges $m \geq 4$ for monomials in S_2 as there are two paths from node i to node 1 and two paths from node j to node 1. Hence we can bound the contribution of all monomials in the set S_2 as follows-

$$\begin{aligned}
 \sum_{f: f \in S_2} \mathbb{E}[f] &\leq \sum_{m=4}^{\log^6 d} k^{m-c(G_f)-1} \frac{m(m^2)^m (\log^{10} d)^{5m/2} \gamma^m}{d^{m-c(G_f)}} \\
 &\leq \sum_{m=4}^{\log^6 d} k^{m-c(G_f)-1} \frac{(\gamma \log^{55} d)^m}{d^{m-c(G_f)}} \\
 &\leq \frac{1}{d} \sum_{m=4}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{k}{d}\right)^{m-c(G_f)-1} \\
 &\leq \frac{1}{d} \sum_{m=4}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{1}{d^\epsilon}\right)^{m/2-1} \leq \frac{1}{d} \sum_{m=4}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{1}{d^\epsilon}\right)^{m/10} \\
 &\leq \frac{1}{d} \sum_{m=4}^{\log^6 d} \left(\frac{\gamma \log^{55} d}{d^{0.1\epsilon}}\right)^m \leq \frac{1}{d} \sum_{m=4}^{\infty} \left(\frac{\gamma \log^{55} d}{d^{0.1\epsilon}}\right)^m \leq \frac{2}{d}
 \end{aligned}$$

λ_τ is composed of monomials with at least one correlation $(c_{1,i})$ term for $i \neq 1$. Also, all graphs for monomials corresponding to the expansion of λ_τ must include a node with label A_1 . As before, for any graph G_f with n nodes with at least one of the nodes corresponding to factor A_1 , there can be at most $nk^{n-1} \leq nd^{n-1}$ monomials having a graph isomorphic to G_f as their representation. By Lemma 11, $n \leq m - c(G_f) + 1$. Hence we can bound λ_τ as follows,

$$\begin{aligned}
 \mathbb{E}[\lambda_\tau] &\leq \sum_{m=1}^{\log^6 d} k^{m-c(G_f)} \frac{m(m^2)^m (\log^{10} d)^{5m/2} \gamma^m}{d^{m-c(G_f)}} \\
 &\leq \sum_{m=1}^{\log^6 d} k^{m-c(G_f)} \frac{(\gamma \log^{55} d)^m}{d^{m-c(G_f)}} \\
 &\leq \sum_{m=1}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{k}{d}\right)^{m-c(G_f)} \\
 &\leq \sum_{m=1}^{\log^6 d} (\gamma \log^{55} d)^m \left(\frac{1}{d^\epsilon}\right)^{m/2} \leq \sum_{m=1}^{\log^6 d} \left(\frac{\gamma \log^{55} d}{d^{0.5\epsilon}}\right)^m \\
 &\leq \sum_{m=1}^{\infty} \left(\frac{\gamma \log^{55} d}{d^{0.5\epsilon}}\right)^m \leq \frac{1}{d^{\epsilon'}}
 \end{aligned}$$

for some $\epsilon' > 0$. We now use Markov's inequality to get high probability guarantees

$$\begin{aligned}
 \mathbb{P}\left[\epsilon_\tau \geq \log d/d\right] &\leq 4/\log^2 d \\
 \mathbb{P}\left[\lambda_\tau \geq \log d/d^{\epsilon'}\right] &\leq 1/\log^2 d
 \end{aligned}$$

Hence we have shown that $\|\Delta_\tau\|_2 \leq \tilde{O}(1/\sqrt{d})$ and $|\lambda_\tau| \leq d^{-\epsilon}$ with failure probability at most $\log^{-1} d$. \square

D. Proof of convergence for Orth-ALS

The proof of convergence of Orth-ALS for incoherent tensors mirrors the proof for orthogonal tensors in Section A. For clarity, we will try to stick to the proof for the orthogonal case as far as possible, while also providing proofs for intermediate Lemmas which were stated without proof in Section A.

Theorem 1. Consider a d -dimensional rank k tensor $T = \sum_{i=1}^k w_i A_i \otimes A_i \otimes A_i$. Let $c_{\max} = \max_{i \neq j} |A_i^T A_j|$ be the incoherence between the true factors and $\gamma = \frac{w_{\max}}{w_{\min}}$ be the ratio of the largest and smallest weight. Assume $\gamma c_{\max} \leq o(k^{-2})$, and the estimates of the factors are initialized randomly from the unit sphere. Provided that, at the $i(\log k + \log \log d)$ th step of the algorithm the estimates for all but the first i factors are re-randomized, then with high probability the orthogonalized ALS updates converge to the true factors in $O(k(\log k + \log \log d))$ steps, and the error at convergence satisfies (up to relabelling) $\|A_i - \hat{A}_i\|_2^2 \leq O(\gamma k \max\{c_{\max}^2, 1/d^2\})$ and $|1 - \frac{\hat{w}_i}{w_i}| \leq O(\max\{c_{\max}, 1/d\})$, for all i .

Proof. Without loss of generality, we assume that the i th recovered factor converges to the i th true factor. Note that the iterations for the first factor are the usual tensor power method updates and are unaffected by the remaining factors. Hence by Theorem 3, Orth-ALS correctly recovers the first factor $O(\log k + \log \log d)$ steps with probability at least $(1 - \log^5 k/k^{1+\epsilon})$, for any $\epsilon > 0$.

We now prove that Orth-ALS also recovers the remaining factors. The proof proceeds by induction. We have already shown that the base case is correct and the algorithm recovers the first factor. We next show that if the first $(m-1)$ factors have converged, then the m th factor converges in $O(\log k + \log \log d)$ steps with failure probability at most $\tilde{O}(1/k^{1+\epsilon})$. The main idea is that as the factors have small correlation with each other, hence orthogonalization does not affect the factors which have not been recovered but ensures that the m th estimate never has high correlation with the factors which have already been recovered. Recall that we assume without loss of generality that the i th recovered factor X_i converges to the i th true factor, hence $X_i = A_i + \hat{\Delta}_i$ for $i < m$, where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k\eta^2$. This is our induction hypothesis, which is true for the base case as by Theorem 3 the tensor power method updates converge with residual error at most $10\gamma k\eta^2$.

Let $X_{m,t}$ denote the m th factor estimate at time t and let Y_m denote it's value at convergence. We will first calculate the effect of the orthogonalization step on the correlation between the factors and the estimate $X_{m,t}$. Let $\{\bar{X}_i, i < m\}$ denote an orthogonal basis for $\{X_i, i < m\}$. The basis $\{\bar{X}_i, i < m\}$ is calculated via QR decomposition, and can be written down as follows,

$$\bar{X}_i = \frac{X_i - \sum_{j < i} \bar{X}_j^T X_i \bar{X}_j}{\|X_i - \sum_{j < i} \bar{X}_j^T X_i \bar{X}_j\|_2}$$

Note that the estimate $X_{m,t}$ is projected orthogonal to this basis. Define $\bar{X}_{m,t}$ as this orthogonal projection, which can be written down as follows –

$$\bar{X}_{m,t} = X_{m,t} - \sum_{j < m} \bar{X}_j^T X_{m,t} \bar{X}_j$$

In the QR decomposition algorithm $\bar{X}_{m,t}$ is also normalized to have unit norm but we will ignore the normalization of $X_{m,t}$ in our analysis because as before we only consider ratios of correlations of the true factors with $\bar{X}_{m,t}$, which is unaffected by normalization.

We will now analyze the orthogonal basis $\{\bar{X}_i, i < m\}$. The key idea is that the orthogonal basis $\{\bar{X}_i, i < m\}$ is close to the original factors $\{X_i, i < m\}$ as the factors are incoherent. Lemma 3 proves this claim.

Lemma 3. Consider a stage of the Orthogonalized ALS iterations when the first $(m-1)$ factors have converged. Without loss of generality let $X_i = A_i + \hat{\Delta}_i, i < m$, where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k\eta^2$. Let $\{\bar{X}_i, i < m\}$ denote an orthogonal basis for $\{X_i, i < m\}$ calculated using Eq. A.5. Then,

1. $\bar{X}_i = A_i + \Delta_i, \forall i < m$ and $\|\Delta_j\|_2 \leq 10k\eta$.
2. $|A_j^T \Delta_i| \leq 3\eta, \forall i < m, j < i$.
3. $|A_j^T \Delta_i| \leq 20\gamma k\eta^2, \forall i < m, j > i$.

Proof. We argue the result by induction. As the first estimate converges to $A_1 + \hat{\Delta}_1$ where $\|\hat{\Delta}_1\|_2 \leq 10\gamma k\eta^2$, the base case is correct. Assume that the result is true for the first $p-1$ vectors in the basis. After orthogonalization, the p th basis vector has the following form-

$$\bar{X}_p = \frac{1}{\kappa} \left((A_p + \hat{\Delta}_p) - \sum_{j < p} ((A_p + \hat{\Delta}_p)^T \bar{X}_j) \bar{X}_j \right)$$

where κ is the normalizing factor which ensures $\|\bar{X}_p\|_2 = 1$. Define $\mu_{p,j} = (A_p^T(A_j + \Delta_j))$. As $|A_p^T \Delta_j| \leq 20\gamma k\eta^2$ by the induction hypothesis and $|A_p^T A_j| \leq \eta$ by definition of η , $|\mu_{p,j}| \leq 2\eta$. Using the induction hypothesis, we can write

$$\begin{aligned}\kappa \bar{X}_p &= A_p - \sum_{j < p} \left(A_p^T(A_j + \Delta_j) \right) (A_j + \Delta_j) + \hat{\Delta}_p - \sum_{k < p} \left(\hat{\Delta}_p^T(A_j + \Delta_j) \right) (A_j + \Delta_j) \\ &= A_p - \sum_{j < p} \mu_{p,j} (A_j + \Delta_j) + \hat{\Delta}_\epsilon\end{aligned}$$

where $\hat{\Delta}_\epsilon = \hat{\Delta}_p - \sum_{k < p} \left(\hat{\Delta}_p^T(A_j + \Delta_j) \right) (A_j + \Delta_j)$. As $\hat{\Delta}_\epsilon$ is a projection of $\hat{\Delta}_p$ orthogonal to the basis $\{\bar{X}_i, i < p\}$, $\|\hat{\Delta}_\epsilon\|_2 \leq \|\hat{\Delta}_p\|_2 \leq 10\gamma k\eta^2$. We can write-

$$\begin{aligned}\kappa \bar{X}_p &= A_p - \sum_{j < p} \mu_{p,j} A_j - \sum_{j < p} \mu_{p,j} \Delta_j + \hat{\Delta}_\epsilon \\ &= A_p + \Delta'_p\end{aligned}$$

where $\Delta'_p = -\sum_{j < p} \mu_{p,j} A_j - \sum_{j < p} \mu_{p,j} \Delta_j + \hat{\Delta}_\epsilon$. We bound $\|\Delta'_p\|_2$ as follows-

$$\begin{aligned}\|\Delta'_p\|_2 &\leq \sum_{j < p} \|\mu_{p,j} A_j\|_2 + \sum_{j < p} \|\mu_{p,j} \Delta_j\|_2 + \|\hat{\Delta}_\epsilon\|_2 \\ &\leq 2k\eta + 20k^2\eta^2 + 10\gamma k\eta^2 \leq 3k\eta\end{aligned}$$

Note that $\kappa = \|A_p + \Delta'_p\|_2 \implies 1 - 3k\eta \leq \kappa \leq 1 + 3k\eta$ by the triangle inequality. Hence $1 - 3k\eta \leq 1/\kappa \leq 1 + 6k\eta$. Therefore we can rewrite \bar{X}_p as-

$$\begin{aligned}\bar{X}_p &= \frac{1}{\kappa} (A_p + \Delta'_p) \\ &= A_p + \left(1 - \frac{1}{\kappa}\right) A_p + \frac{1}{\kappa} \Delta'_p \\ &= A_p + c_1 A_p + c_2 \Delta'_p \\ &= A_p + \Delta_p\end{aligned}$$

where $c_1 = (1 - \frac{1}{\kappa})$, $c_2 = \frac{1}{\kappa}$ and $\Delta_p = c_1 A_p + c_2 \Delta'_p$. Note that $|c_1| \leq 6k\eta$ and $1 - 3k\eta \leq c_2 \leq 1 + 6k\eta$. Hence $\|\Delta_p\|_2 \leq 10k\eta$.

We now show that $|A_i^T \Delta_p| \leq 3\eta$, $i < p$,

$$\begin{aligned}\Delta_p &= c_1 A_p + c_2 \left(-\sum_{j < p} \mu_{p,j} A_j - \sum_{j < p} \mu_{p,j} \Delta_j + \hat{\Delta}_\epsilon \right) \\ \implies |A_i^T \Delta_p| &= \left| c_1 A_i^T A_p \right| + c_2 \left| \sum_{j < p} \mu_{p,j} A_i^T A_j - \sum_{j < p, j \neq i} \mu_{p,j} A_i^T \Delta_j - \mu_{p,i} A_i^T \Delta_i + A_i^T \hat{\Delta}_\epsilon \right| \\ &\leq 6k\eta^2 + (1 + 6k\eta)(2\eta(1 + k\eta) + 6k\eta^2 + 20k\eta^2 + 10\gamma k\eta^2) \\ &\leq 3\eta\end{aligned}$$

Finally, we show that $|A_i^T \Delta_p| \leq 20\gamma k\eta^2$, $i > p$,

$$\begin{aligned}|A_i^T \Delta_p| &= c_1 |A_i^T A_p| + c_2 \left| \sum_{j < p} \mu_{p,j} A_i^T A_j - \sum_{j < p} \mu_{p,j} A_i^T \Delta_j + A_i^T \hat{\Delta}_\epsilon \right| \\ &\leq 6k\eta^2 + (1 + 6k\eta)(2k\eta^2 + 40\gamma k^2\eta^3 + 10\gamma k\eta^2) \\ &\leq 20\gamma k\eta^2\end{aligned}$$

□

Using Lemma 3, we will find the effect of orthogonalization on the correlations of the factors with the iterate $X_{m,t}$. At a high level, we need to show that the iterations for the factors $\{A_i, i \geq m\}$ are not much affected by the orthogonalization, while the correlations of the factors $\{A_i, i < m\}$ with the estimate $X_{m,t}$ are ensured to be small. Lemma 3 is the key tool to prove this, as it shows that the orthogonalized basis is close to the true factors.

We will now analyze the inner product between $\bar{X}_{m,t}$ and factor A_i . This is given by-

$$A_i^T \bar{X}_{m,t} = A_i^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_i^T \bar{X}_j$$

As earlier, we normalize all the correlations by the correlation of the largest factor, let $\bar{a}_{i,t}$ be the ratio of the correlations of A_i and A_m with the orthogonalized estimate $\bar{X}_{m,t}$ at time t . We can write $\bar{a}_{i,t}$ as-

$$\bar{a}_{i,t} = \frac{A_i^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_i^T \bar{X}_j}{A_m^T X_{m,t} - \sum_{j < m} X_{m,t}^T \bar{X}_j A_m^T \bar{X}_j}$$

We can multiply both sides by \hat{w}_i and substitute \bar{X}_j from Lemma 3 and then rewrite as follows-

$$\hat{w}_i \bar{a}_{i,t} = \frac{\hat{w}_i A_i^T X_{m,t} - \sum_{j < m} \hat{w}_i X_{m,t}^T (A_j + \Delta_j) A_i^T \Delta_j}{A_m^T X_{m,t} - \sum_{j < m} X_{m,t}^T (A_j + \Delta_j) A_m^T \Delta_j}$$

We divide the numerator and denominator by $A_m^T X_{m,t}$ to derive an expression in terms of the ratios of correlations. Let $\delta_{i,t} = \frac{X_{m,t}^T \Delta_j}{X_{m,t}^T A_m}$.

$$\hat{w}_i \bar{a}_{i,t} = \frac{\hat{w}_i \hat{a}_{i,t} - \sum_{j < m} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{i,t}) A_i^T \Delta_j}{1 - \sum_{j < m} (\hat{a}_{j,t} + \delta_{i,t}) A_m^T \Delta_j}$$

Lemma 12 upper bounds $\delta_{i,t}$.

Lemma 12. *Let $|\hat{w}_i \hat{a}_{i,t-1}| \leq \beta_{t-1} \forall i \neq m$ and some time $(t-1)$. Also, let $\beta_t \leq \gamma\eta + \beta_{t-1}^2$. Then for all $i < m$, $\delta_{i,t} \leq 40\gamma k\eta\beta_t$.*

Proof. By the power method updates $X_{m,t} = \frac{\sum_i w_i \lambda_i A_i}{\|\sum_i w_i \lambda_i A_i\|_2}$ where $\lambda_i = a_{i,t-1}^2$. Note that $\delta_{i,j}$ is normalized by the correlation of the largest factor A_m , hence the normalizing factor $\|\sum_i w_i \lambda_i A_i\|_2$ does not matter and we will ignore it. We use Lemma 3 to bound $|A_i^T \Delta_j|$. Hence,

$$\begin{aligned} \left| \frac{X_{m,t}^T \Delta_j}{X_{m,t}^T A_m} \right| &\leq \frac{\sum_i \hat{w}_i \hat{a}_{i,t-1}^2 |A_i^T \Delta_j|}{\sum_i \hat{w}_i \hat{a}_{i,t-1}^2 A_i^T A_m} \\ &= \frac{|A_m^T \Delta_j| + \sum_{i \neq j, m} \hat{w}_i \hat{a}_{i,t-1}^2 |A_i^T \Delta_i| + \hat{w}_i \hat{a}_{j,t-1}^2 |A_j^T \Delta_j|}{1 + \sum_{i \neq m} c_{i,m} \hat{w}_i \hat{a}_{i,t-1}^2} \\ &\leq \frac{20\gamma k\eta^2 + 3k\eta\beta_{t-1}^2 + 10k\eta\beta_{t-1}^2}{1 - \gamma k\eta\beta_{t-1}^2} \\ &\leq \frac{k\eta(20\gamma\eta + 13\beta_{t-1}^2)}{1 - 0.5} \\ &\leq 40\gamma k\eta\beta_t \end{aligned}$$

□

We now need to show $\hat{w}_i \bar{a}_{i,t}$ is small for all $i < m$ and is close to $\hat{w}_i a_{i,t}$, the weighted correlation before orthogonalization, for all $i > m$. Lemma 4 proves this, and shows that the weighted correlation of factors which have not yet been recovered, $\{A_i, i \geq m\}$, is not much affected by orthogonalization, but the factors which have already been recovered, $\{A_i, i < m\}$, are ensured to be small after the orthogonalization step.

Lemma 4. Let $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t \forall i \neq m$ at the end of the t th iteration. Let $\bar{a}_{i,t}$ be the ratio of the correlation of the i th and the m th factor with $X_{m,t}$, the iterate at time t after the orthogonalization step. Then,

1. $|\hat{w}_i \bar{a}_{i,t}| \leq \beta_t(1 + 1/k^{1+\epsilon}), \forall i > m.$
2. $|\hat{w}_i \bar{a}_{i,t}| \leq 50\gamma k\eta\beta_t, \forall i < m.$

Proof. We can bound $\bar{a}_{i,t}$ for all $i \geq m$ as-

$$\begin{aligned} |\hat{w}_i \bar{a}_{i,t}| &\leq \frac{|\hat{w}_i \hat{a}_{i,t}| + \left| \sum_{j < m} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{j,t}) A_i^T (A_j + \Delta_j) \right|}{1 - \left| \sum_{j < m} (\hat{a}_{j,t} + \delta_{j,t}) A_m^T (A_j + \Delta_j) \right|} \\ &\leq \frac{|\hat{w}_i \hat{a}_{i,t}| + \sum_{j < m} \left| (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{j,t}) \right| \left| A_i^T (A_j + \Delta_j) \right|}{1 - \sum_{j < m} \left| (\hat{a}_{j,t} + \delta_{j,t}) \right| \left| A_m^T (A_j + \Delta_j) \right|} \\ &\leq \frac{|\hat{w}_i \hat{a}_{i,t}| + \sum_{j < m} \left(\left| \hat{w}_i \hat{a}_{j,t} \right| + \left| \hat{w}_i \delta_{j,t} \right| \right) \left| A_i^T (A_j + \Delta_j) \right|}{1 - \sum_{j < m} \left(\left| \hat{a}_{j,t} \right| + \left| \delta_{j,t} \right| \right) \left| A_m^T (A_j + \Delta_j) \right|} \end{aligned}$$

Note that $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t$, $|\hat{w}_i \hat{a}_{j,t}| \leq \gamma\beta_t$ and $|\hat{w}_i \delta_{j,t}| \leq 40\gamma k\eta\beta_t$. Also, $|A_i^T (A_j + \Delta_j)| \leq 4\eta$ using Lemma 3. Hence we can write,

$$\begin{aligned} |\hat{w}_i \bar{a}_{i,t}| &\leq \beta_t \frac{1 + 8\gamma k\eta}{1 - 4k\eta\beta_t} \\ &\leq \beta_t(1 + 8\gamma k\eta)(1 + 8k\eta\beta_t) \\ &\leq \beta_t(1 + 20\gamma k\eta) \\ &\leq \beta_t(1 + 1/k^{1+\epsilon}) \end{aligned}$$

Similarly, we can bound $\bar{a}_{i,t}$ for all $i < m$ as-

$$\begin{aligned} \hat{w}_i \bar{a}_{i,t} &= \frac{\hat{w}_i \hat{a}_{i,t} - \sum_{j < m} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{j,t}) A_i^T (A_j + \Delta_j)}{1 - \sum_{j < m} (\hat{a}_{j,t} + \delta_{j,t}) A_m^T (A_j + \Delta_j)} \\ &= \frac{\hat{w}_i \hat{a}_{i,t} - (\hat{w}_i \hat{a}_{i,t} + \hat{w}_i \delta_{j,t}) A_i^T (A_i + \Delta_i) - \sum_{j < m, j \neq i} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{j,t}) A_i^T (A_j + \Delta_j)}{1 - \sum_{j < m} (\hat{a}_{j,t} + \delta_{j,t}) A_m^T (A_j + \Delta_j)} \\ &= \frac{\hat{w}_i \delta_{j,t} A_i^T (A_i + \Delta_i) - \sum_{j < m, j \neq i} (\hat{w}_i \hat{a}_{j,t} + \hat{w}_i \delta_{j,t}) A_i^T (A_j + \Delta_j)}{1 - \sum_{j < m} (\hat{a}_{j,t} + \delta_{j,t}) A_m^T (A_j + \Delta_j)} \\ &\leq \frac{\left| \hat{w}_i \delta_{j,t} \right| \left| A_i^T (A_i + \Delta_i) \right| + \sum_{j \neq i} \left(\left| \hat{w}_i \hat{a}_{j,t} \right| + \left| \hat{w}_i \delta_{j,t} \right| \right) \left| A_i^T (A_j + \Delta_j) \right|}{1 - \sum_{j, m} \left(\left| \hat{a}_{j,t} \right| + \left| \delta_{j,t} \right| \right) \left| A_m^T (A_j + \Delta_j) \right|} \\ &\leq \frac{40\gamma k\eta\beta_t + 8\gamma k\eta\beta_t}{1 - 4k\eta\beta_t} \\ &\leq 50\gamma k\eta\beta_t \end{aligned}$$

where we have again used the relations $|\hat{w}_i \hat{a}_{i,t}| \leq \beta_t$, $|\hat{w}_i \hat{a}_{j,t}| \leq \gamma\beta_t$, $|\hat{w}_i \delta_{j,t}| \leq 40\gamma k\eta\beta_t$ and $|A_i^T (A_j + \Delta_j)| \leq 4\eta$. \square

We are now ready to analyze the Orth-ALS updates for the m th factor. First, we argue about the initialization step. Lemma 4 shows that an orthogonalization step performed after the initialization ensures that the factors which have already been recovered have small correlation with the orthogonalized initialization –

Lemma 5. Let $X_{m,0}$ be initialized randomly and the result be projected orthogonal to the $(m-1)$ previously estimated factors, let these be $\{X_i, i < m\}$ without loss of generality. Then $\arg \max_i |w_i a_{i,0}| \geq m$ with high probability. Also, with failure probability at most $\left(1 - \frac{\log^5 k}{k^{1+\epsilon}}\right)$, $\left|\frac{w_i \bar{a}_{i,0}}{\max_i \{w_i \bar{a}_{i,0}\}}\right| \leq 1 - 4/k^{1+\epsilon} \forall i \neq \arg \max_i |w_i a_{i,0}|$ after the orthogonalization step.

Proof. We first show that $\arg \max_i |w_i a_{i,0}| \geq m$. From Lemma 4, the ratio of the weighted correlation of all factors $\{A_i, i < m\}$ with the random initialization and the weighted correlation of all factors $\{A_i, i \geq m\}$ with the random initialization is shrunk by a factor of $O(k^{-(1+\epsilon)})$ after the orthogonalization step. Hence no factor $\{A_i, i < m\}$ will have maximum weighted correlation after the orthogonalization step.

Lemma 1 can now be applied on all remaining factors, to get the initialization condition. Without loss of generality, assume that $\arg \max |w_i a_{i,0}| = m$. Consider the set of factors $\{A_i, m \leq i \leq n\}$. From Lemma 1, with probability at least $\left(1 - \frac{\log^5 k}{k^{1+\epsilon}}\right)$, $\left|\frac{w_i a_{i,0}}{w_m a_{m,0}}\right| \leq 1 - 5/k^{1+\epsilon}, \epsilon > 0 \forall i \neq m$. Applying Lemma 4 once more, $|\hat{w}_i \bar{a}_{i,t}| \leq \beta_t(1 + 1/k^{1+\epsilon}), \forall i > m$. Therefore combining Lemma 1 and Lemma 4, with failure probability at most $\left(1 - \frac{\log^5 k}{k^{1+\epsilon}}\right)$, $\left|\frac{w_i \bar{a}_{i,0}}{w_m \bar{a}_{m,0}}\right| \leq 1 - 4/k^{1+\epsilon} \forall i \neq m$ after the orthogonalization step. \square

Lemma 5 shows that with high probability, the initialization for the m th recovered factor has the largest weighted correlation with a factor which has not been recovered so far after the orthogonalization step. It also shows that the separation condition in Lemma 1 is satisfied for all remaining factors with probability $(1 - \log^5 k/k^{1+\epsilon})$.

Now, we combine the effects of the tensor power method step and the orthogonalization step for subsequent iterations to show that that $X_{m,t}$ converges to A_m . Consider a tensor power method step followed by an orthogonalization step. By Lemma 6, if $|\hat{w}_i \hat{a}_{i,t-1}| \leq \beta_{t-1} i \neq m$ at some time $(t-1)$, then $|\hat{w}_i \hat{a}_{i,t}| \leq (\gamma c_{\max} + \beta_{t-1}^2 + 3\gamma k c_{\max} \beta_{t-1}^2)$ for $i \neq m$ after a tensor power method step. Lemma 4 shows that the correlation of all factors other than the m th factors is still small after the orthogonalization step if it was small before. Combining the effect of the orthogonalization step via Lemma 4, if $|\hat{w}_i \hat{a}_{i,t-1}| \leq \beta_{t-1} i \neq m$ for some time $(t-1)$, then $|\hat{w}_i \hat{a}_{i,t}| \leq (\gamma c_{\max} + \beta_{t-1}^2 + 3\gamma k c_{\max} \beta_{t-1}^2)(1 + 1/k^{1+\epsilon})$ for $i \neq m$ after both the tensor power method and the orthogonalization steps. By also using Lemma 5 for the initialization, can now write the updated combined recursion analogous to Eq. B.2 and Eq. B.2, but which combines the effect of the tensor power method step and the orthogonalization step.

$$\beta_0 = \max_{i \neq 1} |w_i \hat{a}_{i,0}| \quad (\text{D.1})$$

$$\beta_{t+1} = (\gamma c_{\max} + \beta_t^2 + 3\gamma k c_{\max} \beta_t^2)(1 + 1/k^{1+\epsilon}) \quad (\text{D.2})$$

By the previous argument, $|w_i \bar{a}_{i,t}| \leq \beta_t$. Note that $\beta_0 \leq 1 - 4/k^{1+\epsilon}$ by Lemma 5

Lemma 13. $\beta_t \leq 3\gamma c_{\max} \forall t \geq O(\log k + \log \log d)$, also $\beta_t < 1 - 1/k^{1+\epsilon} \forall t$

Proof. The proof is very similar to the proof for Lemma 7. We divide the updates into three stages.

1. $0.1 \leq \beta_t \leq 1 - 4/k^{1+\epsilon}$:

As $\beta_t \geq 0.1$ therefore $k\beta_t^2 \geq 1$ in this regime and hence $\gamma c_{\max} \leq \gamma k \beta_t^2 c_{\max}$, and we can write-

$$\beta_{t+1} = (\gamma c_{\max} + \beta_t^2 + 3\gamma k c_{\max} \beta_t^2)(1 + 1/k^{1+\epsilon})$$

$$\beta_{t+1} \leq (\beta_t^2 + 4\gamma k c_{\max} \beta_t^2)(1 + 1/k^{1+\epsilon})$$

We claim that $\beta_t < 0.1$ for $t = 2 \log k$. To verify, note that-

$$\begin{aligned} \beta_t &\leq (\beta_0(1 + 4\gamma^2 k c_{\max}))(1 + 1/k^{1+\epsilon})^{2^t} \\ &\leq \left((1 - 4/k^{1+\epsilon})(1 + 1/k^{1+\epsilon})(1 + 1/k^{1+\epsilon})\right)^{2^t} \\ &\leq \left(1 - 1/k^{1+\epsilon}\right)^{2^t} \end{aligned}$$

This follows because $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$. Note that $(1 - 1/k^{1+\epsilon})^{2^t} \leq 0.1$ for $t = 2 \log k$ and hence we stay in this regime for at most $2 \log k$ steps.

2. $\sqrt{\gamma\eta} \leq \beta_t \leq 0.1$:

For notational convenience, we restart t from 0 in this stage. Because $\gamma c_{\max} \leq \gamma\eta \leq \beta_t^2$ in this regime and $3\gamma k \beta_t^2 c_{\max} \leq 0.1\beta_t^2$ as $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$, we can write-

$$\begin{aligned} \beta_{t+1} &= (\gamma c_{\max} + \beta_t^2 + 4\gamma k c_{\max} \beta_t^2)(1 + 1/k^{1+\epsilon}) \\ &\leq (\beta_t^2 + \beta_t^2 + 0.1\beta_t^2)(1 + 1/k^{1+\epsilon}) \\ &\leq 2.5\beta_t^2 \end{aligned}$$

We claim that $\beta_t < \sqrt{\gamma\eta}$ for $t = O(\log \log(\gamma\eta)^{-1})$. To verify, note that-

$$\begin{aligned} \beta_t &\leq (2.5(1 + O(\log^{-2} k))\beta_{t_1})^{2^t} \\ &\leq (0.25)^{2^t} \end{aligned}$$

Note that $(0.25)^{2^t} \leq \sqrt{\gamma\eta}$ for $t = O(\log \log(\gamma\eta)^{-1})$ and hence we stay in this stage for at most $O(\log \log(\gamma\eta)^{-1})$ steps. As $\eta^{-1} = O(d)$, this stage continues for at most $O(\log \log d)$ steps.

3. Note that in the next step, $\beta_t \leq (\gamma c_{\max} + 1.1\gamma\eta)(1 + 1/k^{1+\epsilon}) \leq 3\gamma\eta$. This is again because $3\gamma^2 k \beta_t^2 \eta \leq 0.1\beta_t^2$ and $\beta_t \leq \sqrt{\gamma\eta}$ at the end of the previous stage.

□

Therefore $\beta_t \leq 3\gamma\eta$ for some $t = O(\log \log d + \log k)$. By Lemma 6, $|\hat{a}_{i,t} - c_{i,1}| \leq 18\gamma^2\eta^2, i \neq 1$. Hence $|\hat{a}_{i,t}| \leq 2\eta$. By Lemma 2, the error at convergence satisfies $\|A_m - \hat{A}_m\|_2^2 \leq 10\gamma k \eta^2$ and the estimate of the weight \bar{w}_m satisfies $|1 - \frac{\bar{w}_m}{w_m}| \leq O(\eta)$.

Hence we have shown that if the first $(m-1)$ factors have converged to $X_i = A_i + \hat{\Delta}_i$ where $\|\hat{\Delta}_i\|_2 \leq 10\gamma k/d^2, \forall i < m$ then the m th factor converges to $X_m = A_m + \hat{\Delta}_m$ where $\|\hat{\Delta}_m\|_2 \leq 10\gamma k/d^2$ in $O(\log k + \log \log d)$ steps with probability at least $(1 - \frac{\log^5 k}{k^{1+\epsilon}})$. This proves the induction hypothesis.

We can now do a union bound to argue that each factor converges with ℓ_2 error at most $O(\gamma k/d^2)$ in $O(\log k + \log \log d)$ with overall failure probability at most $\tilde{O}(1/k^{-\epsilon}), \epsilon > 0$. □

E. Proof of additional Lemmas

In this section, we will prove the initialization condition which we used at several points in the proof of convergence of the tensor power method and Orth-ALS updates. We also provide the proof for a few Lemmas whose proofs were omitted earlier.

Lemma 1. *If $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$ for some $\epsilon > 0$, then with probability at least $(1 - \frac{\log^5 k}{k^{1+\epsilon}}), \frac{|w_i a_{i,0}|}{\max_i |w_i a_{i,0}|} \leq 1 - 5/k^{1+\epsilon} \forall i \neq \arg \max_i |w_i a_{i,0}|$.*

Proof. Without loss of generality, assume $\arg \max_i |w_i a_{i,0}| = 1$. We will first express all factors in terms of a particular choice of orthonormal basis vectors $\{v_i\}, i \in [k]$. $v_1 = A_1$, and v_i is unit vector along the projection of A_i orthogonal to $\{A_j\}, j < i$. In terms of this basis, $A_1 = (1, 0, \dots, 0)$, let $A_2 = (x_{1,2}, u_{2,2}, 0 \dots, 0)$ and in general $A_i = (x_{i,1}, x_{i,2}, \dots, x_{i,i-1}, u_{i,i}, 0, \dots, 0)$. We will show that $|x_{i,j}| \leq O(c_{\max})$ for all valid i, j i.e. for all $j < i, i \in [k]$.

We claim that $|x_{i,j}| \leq c_{\max}(1 + j c_{\max})$ for all valid i, j . We prove this via induction on j . It is clear that $x_{i,1} \leq c_{\max}(1 + c_{\max})$ for all valid i as $\langle A_i, A_1 \rangle \leq c_{\max}, i \neq 1$. The induction step is that $x_{i,j} \leq c_{\max}(1 + p c_{\max})$ for all valid i and $j \leq p$. We show that this implies that $x_{i,p+1} \leq c_{\max}(1 + (p+1)c_{\max})$ for all valid i . Note that $|\langle A_i, A_{p+1} \rangle| \leq c_{\max} \forall i \geq p$

therefore,

$$\begin{aligned}
 |u_{p+1,p+1}x_{i,p+1}| &\leq c_{\max} + \sum_{i=1}^p c_{\max}^2 (1 + ic_{\max})^2 \\
 &\leq c_{\max} + c_{\max}^2 \sum_{i=1}^p (1 + 4ic_{\max}) \\
 &\leq c_{\max} + pc_{\max}^2 + 4p^2 c_{\max}^3
 \end{aligned}$$

From the induction hypothesis, $|u_{p+1,p+1}| \geq 1 - 2kc_{\max}^2$. This is because $|x_{i,j}| \leq c_{\max}(1 + jc_{\max}) \leq 2c_{\max} \implies \sum_j x_{i,j}^2 \leq 4c_{\max}^2 \implies |u_{p+1,p+1}| \geq 1 - 2kc_{\max}^2$. Hence,

$$\begin{aligned}
 |x_{i,p+1}| &\leq (c_{\max} + pc_{\max}^2 + 4p^2 c_{\max}^3)(1 - 2kc_{\max}^2)^{-1} \\
 &\leq (c_{\max} + pc_{\max}^2 + 4p^2 c_{\max}^3)(1 + 4kc_{\max}^2) \\
 &\leq c_{\max} + pc_{\max}^2 + 4k^2 c_{\max}^3 + 4kc_{\max}^3 + 4k^2 c_{\max}^4 + 16k^3 c_{\max}^5 \\
 &\leq c_{\max}(1 + (p+1)c_{\max})
 \end{aligned}$$

Therefore $|x_{i,j}| \leq c_{\max}(1 + jc_{\max}) \leq 2c_{\max}$ for all valid i, j . Let the random initialization be $(t_1/r, t_2/r, \dots, t_k/r)$ where $t_i \sim N(0, 1/d)$ and $r = \sum_i t_i^2$. Let $u_i = w_i t_i$. By Lemma 14 with probability at least $\left(1 - \frac{10 \log^4 k}{k^{1+\epsilon}}\right)$, $\left|\frac{w_i t_i}{w_1 t_1}\right| \leq 1 - 10/k^{1+\epsilon}$, $\epsilon > 0 \forall i \neq 1$. We claim that $\left|\frac{w_i a_{i,0}}{w_1 a_{1,0}}\right| \leq 1 - 5/k^{1+\epsilon}$, $\epsilon > 0 \forall i \neq 1$ whenever $\left|\frac{t_i}{t_1}\right| \leq 1 - 10/k^{1+\epsilon}$, $\epsilon > 0 \forall i \neq 1$. This follows because-

$$\begin{aligned}
 \frac{w_i a_{i,0}}{w_1 a_{1,0}} &= \frac{w_i t_i u_{i,i} + \sum_{j < i} x_{i,j} w_j t_j \frac{w_i}{w_j}}{w_1 t_1} \\
 \implies \left|\frac{w_i a_{i,0}}{w_1 a_{1,0}}\right| &\leq \left|\frac{w_i t_i}{w_1 t_1}\right| + \sum_{j < i} 2c_{\max} \left|\frac{w_i t_i}{w_1 t_1}\right| \gamma \\
 &\leq 1 - 10/k^{1+\epsilon} + 2\gamma k c_{\max} \\
 &\leq 1 - 10/k^{1+\epsilon} + 1/k^{1+\epsilon} \\
 &\leq 1 - 5/k^{1+\epsilon}
 \end{aligned}$$

□

Lemma 14. Let $u_i \sim N(0, w_i^2)$, $i \in [k]$ be independent Gaussian random variables. For $\log^4 k \leq h \leq k^2$, with probability at least $\left(1 - \frac{\log^4 k}{h}\right)$, $\frac{|w_i u_i|}{\max_i |w_i u_i|} \leq 1 - 1/h$ for all $i \neq \arg \max_i |w_i u_i|$.

Proof. We refer to the pdf of u_i by $f_i(x)$. Without loss of generality, assume $\arg \max_i |w_i u_i| = 1$. As we are only interested in the ratio of the absolute value of random variables $\{u_i\}$, we will assume without loss of generality that the standard deviations or the weights w_i have been scaled such that $w_i \geq 1$. We will use the following tail bound on the standard Gaussian random variable x (refer to (Duembgen, 2010))-

$$\frac{e^{-t^2/2}}{\sqrt{2\pi}} \frac{4}{\sqrt{4+t^2}+t} \leq \mathbb{P}[|x| > t] \leq \frac{2e^{-t^2/2}}{t\sqrt{2\pi}} \quad (\text{E.1})$$

Let κ be some variable which satisfies the following relation-

$$1 - \frac{3 \log k}{k} \leq \frac{\sum_{i=1}^k \mathbb{P}[|u_i| \leq \kappa]}{k} \leq 1 - \frac{2 \log k}{k} \quad (\text{E.2})$$

Let $m = \max_i u_i$. As the u_i are independent,

$$\mathbb{P}[m > \kappa] = 1 - \prod_i \mathbb{P}[|u_i| \leq \kappa]$$

By the AM-GM inequality-

$$\begin{aligned} \prod_i \mathbb{P}[|u_i| \leq \kappa] &\leq \left(\frac{\sum_{i=1}^k \mathbb{P}[|u_i| \leq \kappa]}{k} \right)^k \\ &\leq \left(1 - \frac{2 \log k}{k} \right)^k \leq \frac{1}{k^2} \end{aligned}$$

Hence with failure probability at most $1/k^2$ the maximum is at least κ .

Instead of drawing k samples from the k distributions corresponding to the k factors, we first draw the maximum m from the distribution of the maximum of the k samples, and then draw the remaining samples conditioned on the maximum being m . We have shown that $m > \kappa$ with high probability. We condition on the maximum $m > \kappa$. We now show that with high probability no sample lies in the range $[m(1 - 1/h), m]$, given that the maximum is at least κ . After drawing the maximum from its distribution, we will draw samples from the distributions corresponding to *all* the k factors even though one of the factors would already be the maximum m . Clearly this can only increase the probability of a sample lying in the interval $[m(1 - 1/h), m]$, and as we only want an upper bound this is fine. Let the conditional pdf of the i th random variable u_i conditioned on the maximum being m be $g_{i|m}(x)$. Conditioned on the maximum being m , all remaining samples are at most m and hence $g_{i|m}(x) = f_i(x)/\mathbb{P}[|u_i| \leq m]$ for all $x \leq m$ and is 0 otherwise. We will now upper bound $1/\mathbb{P}[|u_i| \leq m]$. We rely on the following observation about the distribution of a standard Normal random variable x -

$$\mathbb{P}[|x| \leq t] \geq \begin{cases} 0.5t & t \in [0, 1] \\ 0.5 & t > 1 \end{cases}$$

The bound for $t \in [0, 1]$ follows from the concavity of the Gaussian cumulative distribution function for $t > 0$, the bound for $t > 1$ is easily verified. Using this, we can write

$$\mathbb{P}[|u_i| \leq m] \geq 0.5 \min \left\{ \frac{m}{w_i}, 1 \right\}$$

We will now find an upper bound on $f_i(m(1 - 1/k))$, the pdf of the samples at $m(1 - 1/h)$. Let $t_i = \frac{m}{w_i}$. Using Eq. E.1-

$$\begin{aligned} f_i(m) &\leq \frac{(\sqrt{4 + t_i^2} + t_i) \mathbb{P}[|u_i| \geq m]}{4w_i} \\ \implies g_{i|m}(m) &\leq \frac{(2t_i + 2) \mathbb{P}[|u_i| \geq \kappa]}{4w_i \mathbb{P}[|u_i| \leq m]} \\ &\leq \frac{(2t_i + 2) \mathbb{P}[|u_i| \geq \kappa]}{2w_i \min \left\{ \frac{m}{w_i}, 1 \right\}} \\ &\leq \frac{(t_i + 1) \mathbb{P}[|u_i| \geq \kappa]}{\min\{m, w_i\}} \\ \implies \sum_{i: t_i \leq \log k} g_{i|m}(m) &\leq \sum_{i: t_i \leq \log k} \frac{(t_i + 1) \mathbb{P}[|u_i| \geq \kappa]}{m / \log k} \\ &\leq \frac{2 \log^2 k \sum_i \mathbb{P}[|u_i| \geq \kappa]}{m} \\ &\leq 6 \log^3 k / m \end{aligned} \tag{E.3}$$

where we used Eq. E.2 in the last step. We will now relate $g_{i|m}(m)$ and $g_{i|m}(m(1 - 1/h))$. We can write,

$$\begin{aligned}
 g_{i|m}(m(1 - 1/h)) &= \frac{\exp\left(-\frac{m^2}{2w_i^2}(1 - 2/h + 1/k^{2+2\epsilon})\right)}{\sqrt{2\pi w_i} \mathbb{P}[|u_i| \leq m]} \\
 &\leq \frac{\exp\left(-t_i^2(1 - 2/h)/2\right)}{\sqrt{2\pi w_i} \mathbb{P}[|u_i| \leq m]} \\
 &= \left(\frac{e^{-t_i^2/2}}{\sqrt{2\pi w_i} \mathbb{P}[|u_i| \leq m]}\right) e^{t_i^2/h} \\
 &= g_{i|m}(m) e^{t_i^2/h} \\
 \implies \sum_{i:t_i \leq \log k} g_{i|m}(m(1 - 1/h)) &\leq \sum_{i:t_i \leq \log k} g_{i|m}(m) e^{t_i^2/h} \\
 &\leq 20 \log^3 k/m
 \end{aligned}$$

where we used Eq. E.3 in the last step. For all $i : t_i > \log k$, we can write,

$$\begin{aligned}
 g_{i|m}(m(1 - 1/h)) &\leq \frac{e^{-t_i^2(1/2 - 1/h)}}{\sqrt{2\pi w_i} \mathbb{P}[|u_i| \leq m]} \\
 &\leq \frac{2e^{-t_i^2/3}}{\sqrt{2\pi} \min\{m, w_i\}} \\
 &\leq \frac{2e^{-t_i^2/3}}{\sqrt{2\pi m}/t_i} \\
 &\leq \frac{2t_i e^{-t_i^2/3}}{\sqrt{2\pi m}} \leq 1/(k^2 m)
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 \sum_{i:t_i > \log k} g_{i|m}(m(1 - 1/h)) &\leq 1/(km) \\
 \implies \sum_i g_{i|m}(m(1 - 1/h)) &\leq 25 \log^3 k/m
 \end{aligned}$$

Hence the probability of a sample lying in the interval $[m(1 - 1/h), m]$ can be bounded by-

$$\begin{aligned}
 \mathbb{P}[\cup(u_i \in [m(1 - 1/h), m])] &\leq \sum_i \mathbb{P}[u_i \in [m(1 - 1/h), m]] \\
 &\leq \sum_i g_{i|m}(m(1 - 1/h)) \frac{m}{h} \\
 &= \frac{m}{h} \sum_i g_{i|m}(m(1 - 1/h)) \\
 &\leq \frac{25 \log^3 k}{h}
 \end{aligned}$$

Hence with probability at least $(1 - \frac{1}{k^2})(1 - \frac{25 \log^3 k}{h}) = 1 - \frac{\log^4 k}{h}$ the maximum is greater than κ and there are no samples in the interval $[m(1 - 1/h), m]$. \square

Lemma 11. For any connected Eulerian graph G , let N be the number of nodes and M be the number of edges. Consider any decomposition of G into a edge-disjoint set of p cycles. Then, $N \leq M - p + 1$. Moreover, if G has four edge-disjoint paths between a pair of nodes then $N \leq M - p$.

Proof. Consider any decomposition of G into a disjoint set of p cycles $C_1 \cup C_2 \cdots \cup C_p$. We will consider the number of unique nodes in $C_1 \cup C_2 \cdots \cup C_t$ for $t \leq p$. Let $N(C_1 \cup C_2 \cdots \cup C_t)$ be the number of unique nodes in $C_1 \cup C_2 \cdots \cup C_t$. Similarly, let $M(C_1 \cup C_2 \cdots \cup C_t)$ be the number of edges in $C_1 \cup C_2 \cdots \cup C_t$. Note that $N(C_1) \leq M(C_1)$. There must be a cycle in $C_2 \cdots \cup C_p$ with at least one node common to C_1 as G is connected. Assume C_2 has this property. Then $N(C_1 \cup C_2) \leq M(C_1) + M(C_2) - 1$ as C_1 and C_2 have one node in common. Repeating this argument, at any stage when we have selected t cycles, there must be a cycle which has not been selected yet but has a node common to the selected cycles. Hence $N(C_1 \cup C_2 \cdots \cup C_p) \leq \sum_1^p M(C_i) - p + 1$.

To prove the second part of the Lemma, we claim that if $N = M - p + 1$ for some decomposition \mathcal{C} of G into a disjoint set of p cycles, then there cannot be more than two disjoint paths between any pair of nodes. By our previous argument, if $N = M - p + 1$ then for any union \mathcal{S} of connected cycles, any cycle not in \mathcal{S} can have at most one node common with the nodes in \mathcal{S} . Note that the number of disjoint set of paths between any pair u and v must be even as the graph is Eulerian. Assume for the sake of contradiction that there are at least four disjoint paths between two nodes u and v . Consider any set of cycles \mathcal{S} in \mathcal{C} which cover any two of the disjoint paths. Say that P is some path which is not covered by \mathcal{S} . Then, there must exist nodes s and t such that s and t are present in \mathcal{C} but a segment of the path P is not present in \mathcal{C} . We claim that this implies that for some union \mathcal{S}' of cycles such that there is some cycle having two nodes common with \mathcal{S}' . To verify this, we simply add cycles to \mathcal{S} to grow our subgraph from node s till it reaches node t . At some point, there must be a cycle with two nodes common to the cycles already selected, because we have to reach the node t which has already been included. \square

Lemma 2. Let $\gamma k c_{\max} \leq 1/k^{1+\epsilon}$. Without loss of generality assume convergence to the first factor A_1 . Define $\hat{a}_{i,t} = |\frac{a_{i,t}}{a_{1,t}}|$ - the ratio of the correlation of the i th and 1st factor with the iterate at time t . If $\hat{a}_{i,t} \leq 2\eta \forall i \neq 1$, then $\|A_1 - \hat{A}_1\|_2^2 \leq 10\gamma k \eta^2$ in the subsequent iteration. Also, if $\|A_1 - \hat{A}_1\|_2^2 \leq O(\gamma k \eta^2)$ the relative error in the estimation of the weight w_1 is at most $O(\eta)$.

Proof. Consider any step τ of the power iterations at the end of which $|\hat{a}_{i,\tau}| \leq 2\eta \forall i \neq 1$. Let the first (largest) factor have true correlation $a_{1,\tau}$ with the iterate at this time step. Consider the next tensor power method update. From the update formula, the result $Z_{\tau+1}$ is-

$$Z_{\tau+1} = \frac{\sum_{i=1}^k w_i \hat{w}_i a_{1,\tau}^2 \hat{a}_{i,\tau}^2 A_i}{\|\sum_{i=1}^k w_i \hat{w}_i a_{1,\tau}^2 \hat{a}_{i,\tau}^2 A_i\|_2}$$

Let $\kappa = \frac{w_1 a_{1,\tau}^2}{\|\sum_{i=1}^k w_i \hat{w}_i a_{1,\tau}^2 \hat{a}_{i,\tau}^2 A_i\|_2}$. Hence the estimate at the end of the m th iteration is-

$$\begin{aligned} X_{\tau+1} &= \kappa \sum_{i=1}^k \hat{w}_i \hat{a}_{i,\tau}^2 A_i \\ &= \kappa (A_1 + \sum_{i \neq 1} \hat{w}_i \hat{a}_{i,\tau}^2 A_i) \end{aligned}$$

Denote $\sum_{i \neq 1} \hat{w}_i \hat{a}_{i,\tau}^2 A_i = \hat{\Delta}_1$. Note that $\|\hat{\Delta}_1\|_2 \leq 4\gamma k \eta^2$ as $|\hat{a}_{i,\tau}| \leq 2\eta \implies \hat{w}_i \hat{a}_{i,\tau}^2 \leq 4\gamma \eta^2$ and the factors A_i have unit norm. As $\|X_{\tau+1}\|_2 = 1, \kappa = 1/\|A_1 + \hat{\Delta}_1\|_2$. From the triangle inequality,

$$\begin{aligned} 1 - \|\hat{\Delta}_1\|_2 &\leq \|A_1 + \hat{\Delta}_1\|_2 \leq 1 + \|\hat{\Delta}_1\|_2 \\ \implies 1 - 4\gamma k \eta^2 &\leq \|A_1 + \hat{\Delta}_1\|_2 \leq 1 + 4\gamma k \eta^2 \\ \implies \frac{1}{1 + 4\gamma k \eta^2} &\leq \kappa \leq \frac{1}{1 - 4\gamma k \eta^2} \\ \implies 1 - 4\gamma k \eta^2 &\leq \kappa \leq 1 + 5\gamma k \eta^2 \end{aligned}$$

We can now write the error $\|A_1 - X_{\tau+1}\|_2$ as-

$$\begin{aligned}
 \|A_1 - X_{\tau+1}\|_2 &= \|A_1 - \kappa(A_1 + \hat{\Delta}_1)\|_2 \\
 &= \|A_1(1 - \kappa) + \kappa\hat{\Delta}_1\|_2 \\
 &\leq \|A_1(1 - \kappa)\|_2 + \kappa\|\hat{\Delta}_1\|_2 \\
 &\leq 5\gamma k\eta^2 + 4\gamma k\eta^2(1 + 5\gamma k\eta^2) \\
 &\leq 10\gamma k\eta^2
 \end{aligned}$$

We also show that the error in estimating the weight w_1 of factor A_1 is small once we have good estimate of the factor.

$$\begin{aligned}
 \bar{w}_1 &= \sum_i w_i \langle \hat{A}_1, A_i \rangle^3 \\
 &= \sum_i w_i \langle A_1 + \hat{\Delta}_1, A_i \rangle^3 \\
 \implies |w_1 - \bar{w}_1| &\leq \left| w_1 \langle A_1 + \hat{\Delta}_1, A_1 \rangle^3 - w_1 \right| + \left| \sum_{i \neq 1} w_i \langle A_1 + \hat{\Delta}_1, A_i \rangle^3 \right| \\
 &\leq 3w_1\eta + 8 \sum_i w_i \eta^3 \\
 &\leq 3w_1\eta + 8w_i k \eta^3 \leq 4w_1\eta \\
 \implies \left| 1 - \frac{\bar{w}_1}{w_1} \right| &\leq O(\eta)
 \end{aligned} \tag{E.4}$$

where Eq. E.4 follows as $|A_i^T \hat{\Delta}_1| \leq \|\hat{\Delta}_1\|_2 \leq \eta$. □