Relative Error Tensor Low Rank Approximation

Zhao Song∗
zhaos@utexas.edu
UT-Austin

David P. Woodruff
dpwoodru@us.ibm.com
IBM Almaden

Peilin Zhong†
peilin.zhong@columbia.edu
Columbia University

Abstract

We consider relative error low rank approximation of tensors with respect to the Frobenius norm. Namely, given an order-$q$ tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_q}$, output a rank-$k$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT}$, where $\text{OPT} = \inf_{\text{rank-} k' A'} \|A - A'\|_F^2$. Despite much success on obtaining relative error low rank approximations for matrices, no such results were known for tensors for arbitrary $(1 + \epsilon)$-approximations. One structural issue is that there may be no rank-$k$ tensor $A_k$ achieving the above infimum. Another, computational issue, is that an efficient relative error low rank approximation algorithm for tensors would allow one to compute the rank of a tensor, which is NP-hard. We bypass these two issues via (1) bicriteria and (2) parameterized complexity solutions:

1. We give an algorithm which outputs a rank $k' = O((k/\epsilon)^{q-1})$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT}$ in $\text{nnz}(A) + n \cdot \text{poly}(k/\epsilon)$ time in the real RAM model, whenever either $A_k$ exists or $\text{OPT} > 0$. Here $\text{nnz}(A)$ denotes the number of non-zero entries in $A$. If both $A_k$ does not exist and $\text{OPT} = 0$, then $B$ instead satisfies $\|A - B\|_F^2 < \gamma$, where $\gamma$ is any positive, arbitrarily small function of $n$.

2. We give an algorithm for any $\delta > 0$ which outputs a rank $k$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT}$ and runs in $(\text{nnz}(A) + n \cdot \text{poly}(k/\epsilon) + \exp(k^3/\epsilon)) \cdot n^{1/\delta}$ time in the unit cost RAM model, whenever $\text{OPT} > 2^{-O(n^{\delta})}$ and there is a rank-$k$ tensor $B = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$ for which $\|A - B\|_F^2 \leq (1 + \epsilon/2)\text{OPT}$ and $\|u_i\|_2, \|v_i\|_2, \|w_i\|_2 \leq 2^{O(n^{\delta})}$. If $\text{OPT} \leq 2^{-\Omega(n^{\delta})}$, then $B$ instead satisfies $\|A - B\|_F^2 \leq 2^{-\Omega(n^{\delta})}$.

Our first result is polynomial time, and in fact input sparsity time, in $n, k,$ and $1/\epsilon$, for any $k \geq 1$ and any $0 < \epsilon < 1$, while our second result is fixed parameter tractable in $k$ and $1/\epsilon$. For outputting a rank-$k$ tensor, or even a bicriteria solution with rank-$Ck$ for a certain constant $C > 1$, we show a $2^{\Omega(k^{1-\omega(1)})}$ time lower bound under the Exponential Time Hypothesis.

Our results are based on an “iterative existential argument”, and also give the first relative error low rank approximations for tensors for a large number of error measures for which nothing was known. In particular, we give the first relative error approximation algorithms on tensors for: column row and tube subset selection, entrywise $\ell_p$-low rank approximation for $1 \leq p < 2$, low rank approximation with respect to sum of Euclidean norms of faces or tubes, weighted low rank approximation, and low rank approximation in distributed and streaming models. We also obtain several new results for matrices, such as $\text{nnz}(A)$-time CUR decompositions, improving the previous $\text{nnz}(A) \log n$-time CUR decompositions, which may be of independent interest.

∗Work done while visiting IBM Almaden, and supported in part by UTCS TAship (CS361 Spring 17 Introduction to Computer Security).
†Supported in part by Simons Foundation, and NSF CCF-1617955.
Contents

1 Introduction ............................................. 4
  1.1 Our Results ........................................... 6
  1.2 Our Techniques ....................................... 9
  1.3 Other Low Rank Approximation Algorithms Following Our Framework .......... 11
  1.4 Comparison to [BCV14] ................................ 16
  1.5 An Algorithm and a Roadmap ......................... 16

A Notation .................................................. 17

B Preliminaries ............................................ 19
  B.1 Subspace Embeddings and Approximate Matrix Product ....................... 20
  B.2 Tensor CURT decomposition ................................ 20
  B.3 Polynomial system verifier ................................ 24
  B.4 Lower bound on the cost of a polynomial system ............................. 25
  B.5 Frobenius norm and $\ell_2$ relaxation .................................. 25
  B.6 CountSketch and Gaussian transforms .................................... 26
  B.7 Cauchy and $p$-stable transforms ..................................... 27
  B.8 Leverage scores .......................................... 28
  B.9 Lewis weights ............................................ 28
  B.10 TENSORSKETCH ......................................... 30

C Frobenius Norm for Arbitrary Tensors ..................... 31
  C.1 $(1 + \epsilon)$-approximate low-rank approximation ......................... 31
  C.2 Input sparsity reduction .................................... 35
  C.3 Tensor multiple regression ................................... 37
  C.4 Bicriteria algorithms ....................................... 38
    C.4.1 Solving a small regression problem ................................ 38
    C.4.2 Algorithm I ............................................. 40
    C.4.3 poly($k$)-approximation to multiple regression ...................... 44
    C.4.4 Algorithm II ............................................. 46
  C.5 Generalized matrix row subset selection .................................. 47
  C.6 Column, row, and tube subset selection, $(1 + \epsilon)$-approximation .......... 51
  C.7 CURT decomposition, $(1 + \epsilon)$-approximation ........................ 53
    C.7.1 Properties of leverage score sampling and BSS sampling ............ 53
    C.7.2 Row sampling for linear regression .............................. 54
    C.7.3 Leverage scores for multiple regression ............................ 56
    C.7.4 Sampling columns according to leverage scores implicitly, improving polynomial running time to nearly linear running time ............ 58
    C.7.5 Input sparsity time algorithm .................................. 61
    C.7.6 Optimal sample complexity algorithm .............................. 63
  C.8 Face-based selection and decomposition ............................... 64
    C.8.1 Column-row, column-tube, row-tube face subset selection .......... 64
    C.8.2 CURT decomposition ....................................... 67
  C.9 Solving small problems ....................................... 69
  C.10 Extension to general $q$-th order tensors .............................. 70
    C.10.1 Fast sampling of columns according to leverage scores, implicitly ... 70
1 Introduction

Low rank approximation of matrices is one of the most well-studied problems in randomized numerical linear algebra. Given an $n \times d$ matrix $A$ with real-valued entries, we want to output a rank-$k$ matrix $B$ for which $\|A - B\|$ is small, under a given norm. While this problem can be solved exactly using the singular value decomposition for some norms like the spectral and Frobenius norms, the time complexity is still $\min(nd^{\omega-1}, dn^{\omega-1})$, where $\omega \approx 2.376$ is the exponent of matrix multiplication [Str69, CW87, Wil12]. This time complexity is prohibitive when $n$ and $d$ are large. By now there are a number of approximation algorithms for this problem, with the Frobenius norm $^1$ being one of the most common error measures. Initial solutions [FKV04, AM07] to this problem were based on sampling and achieved additive error in terms of $\epsilon \|A\|_F$, where $\epsilon > 0$ is an approximation parameter, which can be arbitrarily larger than the optimal cost $\text{OPT} = \min_{\text{rank-k}} B \|A - B\|_F^2$. Since then a number of solutions based on the technique of oblivious sketching [Sar06, CW13, MM13, NN13] as well as sampling based on non-uniform distributions [DMM06b, DMM06a, DMM08, DMIMW12], have been proposed which achieve the stronger notion of relative error; namely, which output a rank-$k$ matrix $B$ for which $\|A - B\|_F \leq (1 + \epsilon) \text{OPT}$ with high probability. It is now known how to output a factorization of such a $B = U \cdot V$, where $U$ is $n \times k$ and $V$ is $k \times d$, in $\text{nnz}(A) + (n + d) \text{poly}(k/\epsilon)$ time [CW13, MM13, NN13]. Such an algorithm is optimal, up to the poly($k/\epsilon$) factor, as any algorithm achieving relative error must read almost all of the entries.

Tensors are often more useful than matrices for capturing higher order relations in data. Computing low rank factorizations of approximations of tensors is the primary task of interest in a number of applications, such as in psychology [Kro83], chemometrics [Paa00, SBG04], neuroscience [AAB+07, KB09, CLK+15], computational biology [CV15, SC15], natural language processing [CYYM14, LZBJ14, LZMB15, BNR+15], computer vision [VT02, WA03, SH05, HPS05, HD08, AFdLGTL09, PLY10, LFC], computer graphics [VT04, WWS], signal processing [Wes94, DLDM98, Com09, CMDL], cryptography [FS99, Sch12, KYFD15, SHW], reinforcement learning, community detection, multi-armed bandit, ranking models, neural network, Gaussian mixture models and Latent Dirichlet allocation [MR05, AFH], cryptography [FS99, Sch12, KYFD15, SHW] data mining [KS08, RST10, KABO10, Mor11], machine learning applications such as learning hidden Markov models, reinforcement learning, community detection, multi-armed bandit, ranking models, neural network, Gaussian mixture models and Latent Dirichlet allocation [MR05, AFH+12, HK13, ALB13, ABSV14, AGH+14, AGHK14, BCV14, JO14a, GHK15, PBLJ15, JSA15, ALA16, AGMR16, ZSJ+17], programming languages [RTP16], signal processing [Wes94, DLM98, Com09, CMDL+15], and other applications [YCS11, LMWY13, OS14, ZCZJ14, STLS14, YCS16, RNSS16].

Despite the success for matrices, the situation for order-$q$ tensors for $q > 2$ is much less understood. There are a number of works based on alternating minimization [CC70, Har70, FMPS13, FT15, ZG01, BS15] gradient descent or Newton methods [ES09, ZG01], methods based on the Higher-order SVD (HOSVD) [LMV00a] which provably incur $\Omega(\sqrt{n})$-inapproximability for Frobenius norm error [LMV00b], the power method or orthogonal iteration method [LMV00b], additive error guarantees in terms of the flattened (unfolded) tensor rather than the original tensor [MMD08], tensor trains [Ose11], the tree Tucker decomposition [OT09], or methods specialized to orthogonal tensors [KM11, AGH+14, MHG15, WTSA15, WA16, SWZ16]. There are also a number of works on the problem of tensor completion, that is, recovering a low rank tensor from missing entries [WM01, AKDM10, TSHK11, LMWY13, MHWG14, JO14b, BM16]. There is also another line of work using the sum of squares (SOS) technique to study tensor problems [BKS15, GM15, HSS15, HSS16, MSS16, PS17, SS17], other recent work on tensor PCA [All12b, All12a, RM14, JMZ15, ADGM16, ZX17], and work applying smoothed analysis to tensor decomposition [BCM14]. Several previous works also consider more robust norms than

\footnote{Recall the Frobenius norm $\|A\|_F$ of a matrix $A$ is $(\sum_{i=1}^n \sum_{j=1}^d A_{i,j}^2)^{1/2}$.}
the Frobenius norm for tensors, e.g., the $R_1$ norm ($\ell_1$-$\ell_2$-$\ell_2$ norm in our work) [HD08], $\ell_1$-PCA [PLY10], entry-wise $\ell_1$ regularization [GGH14], M-estimator loss [YFS16], weighted approximation [Paa97, TK11, LRHG13], tensor-CUR [OST08, MMD08, CC10, FMMN11, FT15], or robust tensor PCA [GQ14, LFC+16, CLZ17].

Some of the above works, such as ones based on the tensor power method or alternating minimization, require incoherence or orthogonality assumptions. Others, such as those based on the simultaneous SVD, require an assumption on the minimum singular value. See the monograph of Moitra [Moi14] for further discussion. Unlike the situation for matrices, there is no work for tensors that is able to achieve the following natural relative error guarantee: given a $q$-th order tensor $A \in \mathbb{R}^{n \otimes q}$ and an arbitrary accuracy parameter $\epsilon > 0$, output a rank-$k$ tensor $B$ for which

$$\|A - B\|_F^2 \leq (1 + \epsilon) \text{OPT},$$

(1)

where $\text{OPT} = \inf_{\text{rank-}k} B\|A - B'\|_F^2$, and where recall the rank of a tensor $B$ is the minimal integer $k$ for which $B$ can be expressed as $\sum_{i=1}^k u_i \otimes v_i \otimes w_i$. A third order tensor, for example, has rank which is an integer in $\{0, 1, 2, \ldots, n^2\}$. We note that [BCV14] is able to achieve a relative error 5-approximation for third order tensors, and an $O(q)$-approximation for $q$-th order tensors, though it cannot achieve a $(1 + \epsilon)$-approximation. We compare our work to [BCV14] in Section 1.4 below.

For notational simplicity, we will start by assuming third order tensors with all dimensions of equal size, but we extend all of our main theorems below to tensors of any constant order $q > 3$ and dimensions of different sizes.

The first caveat regarding (1) for tensors is that an optimal rank-$k$ solution may not even exist! This is a well-known problem for tensors (see, e.g., [KHL89, Paa00, KDS08, Ste06, Ste08] and more details in section 4 of [DSL08]), for which for any rank-$k$ tensor $B$, there always exists another rank-$k$ tensor $B'$ for which $\|A - B'\|_F^2 < \|A - B\|_F^2$. If $\text{OPT} = 0$, then in this case for any rank-$k$ tensor $B$, necessarily $\|A - B\|_F^2 > 0$, and so (1) cannot be satisfied. This fact was known to algebraic geometers as early as the 19th century, which they refer to as the fact that the locus of $r$-th secant planes to a Segre variety may not define a (closed) algebraic variety [DSL08, Lan12]. It is also known as the phenomenon underlying the concept of border rank$^2$[Bin80, Bin86, BCS97, Knu98, Lan06]. In this case it is natural to allow the algorithm to output an arbitrarily small $\gamma > 0$ amount of additive error. Note that unlike several additive error algorithms for matrices, the additive error here can in fact be an arbitrarily small positive function of $n$. If, however, $\text{OPT} > 0$, then for any $\epsilon > 0$, there exists a rank-$k$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon) \text{OPT}$, and in this case we should still require the algorithm to output a relative-error solution. If an optimal rank-$k$ solution $B$ exists, then as for matrices, it is natural to require the algorithm to output a relative-error solution.

Besides the above definitional issue, a central reason that (1) has not been achieved is that computing the rank of a third order tensor is well-known to be NP-hard [Hås90, HL13]. Thus, if one had such a polynomial time procedure for solving the problem above, one could determine the rank of $A$ by running the procedure on each $k \in \{0, 1, 2, \ldots, n^2\}$, and check for the first value of $k$ for which $\|A - B\|_F^2 = 0$, thus determining the rank of $A$. However, it is unclear if approximating the tensor rank is hard. This question will also be answered in this work.

The main question which we address is how to define a meaningful notion of (1) for the case of tensors and whether it is possible to obtain provably efficient algorithms which achieve this guarantee, without any assumptions on the tensor itself. Besides (1), there are many other notions of relative error for low rank approximation of matrices for which provable guarantees for tensors are unknown, such as tensor CURT, $R_1$ norm, and the weighted and $\ell_1$ norms mentioned above. Our goal is to provide a general technique to obtain algorithms for many of these variants as well.

\footnote{https://en.wikipedia.org/wiki/Tensor_rank_decomposition#Border_rank}
1.1 Our Results

To state our results, we first consider the case when a rank-$k$ solution $A_k$ exists, that is, there exists a rank-$k$ tensor $A_k$ for which $\|A - A_k\|_F^2 = \text{OPT}$.

We first give a poly$(n, k, 1/\epsilon)$-time $(1 + \epsilon)$-relative error approximation algorithm for any $0 < \epsilon < 1$ and any $k \geq 1$, but allow the output tensor $B$ to be of rank $O((k/\epsilon)^q)$ (for general $q$-order tensors, the output rank is $O((k/\epsilon)^q-1)$, whereas we measure the cost of $B$ with respect to rank-$k$ tensors. Formally, $\|A - B\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$. In fact, our algorithm can be implemented in $\text{nnz}(A) + n \cdot \text{poly}(k/\epsilon)$ time in the real-RAM model, where $\text{nnz}(A)$ is the number of non-zero entries of $A$. Such an algorithm is optimal for any relative error algorithm, even bicriteria ones.

If $A_k$ does not exist, then our output $B$ instead satisfies $\|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT} + \gamma$, where $\gamma$ is an arbitrarily small additive error. Since $\gamma$ is arbitrarily small, $(1 + \epsilon)\text{OPT} + \gamma$ is still a relative error whenever $\text{OPT} > 0$. Our theorem is as follows.

**Theorem 1.1** (A Version of Theorem C.9, bicriteria). Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, if $A_k$ exists then there is a randomized algorithm running in $\text{nnz}(A) + n \cdot \text{poly}(k/\epsilon)$ time which outputs a (factorization of a) rank-$O(k^2/\epsilon^2)$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$. If $A_k$ does not exist, then the algorithm outputs a rank-$O(k^2/\epsilon^2)$ tensor $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT} + \gamma$, where $\gamma > 0$ is an arbitrarily small positive function of $n$. In both cases, the success probability is at least $2/3$.

One of the main applications of matrix low rank approximation is parameter reduction, as one can store the matrix using fewer parameters in factored form or more quickly multiply by the matrix if given in factored form, as well as remove directions that correspond to noise. In such applications, it is not essential that the low rank approximation have rank exactly $k$, since one still has a significant parameter reduction with a matrix of slightly larger rank. This same motivation applies to tensor low rank approximation; we obtain both space and time savings by representing a tensor in factored form, and in such applications bicriteria applications suffice. Moreover, the extremely efficient $\text{nnz}(A) + n \cdot \text{poly}(k/\epsilon)$ time algorithm we obtain may outweigh the need for outputting a tensor of rank exactly $k$. Bicriteria algorithms are common for coping with hardness; see e.g., results on robust low rank approximation of matrices [DV07, FFSS07, CW15a], sparse recovery [CKPS16], clustering [MMSW15, HT16], and approximation algorithms more generally.

We note that there are other applications, such as unique tensor decomposition in the method of moments, see, e.g., [BCV14], where one may have a hard rank constraint of $k$ for the output. However, in such applications the so-called Tucker decomposition is still a useful dimensionality-reduction analogue of the SVD and our techniques for proving Theorem 1.1 can also be used for obtaining Tucker decompositions, see Section I.

We next consider the case when the rank parameter $k$ is small, and we try to obtain rank-$k$ solutions which are efficient for small values of $k$. As before, we first suppose that $A_k$ exists.

If $A_k = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$ and the norms $\|u_i\|_2, \|v_i\|_2, \text{and } \|w_i\|_2$ are bounded by $2^{\text{poly}(n)}$, we can return a rank-$k$ solution $B$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2 + 2^{-\text{poly}(n)}$, in $f(k, 1/\epsilon) \cdot \text{poly}(n)$ time in the standard unit cost RAM model with words of size $O(\log n)$ bits. Thus, our algorithm is fixed parameter tractable in $k$ and $1/\epsilon$, and in fact remains polynomial time for any values of $k$ and $1/\epsilon$ for which $k^2/\epsilon = O(\log n)$. This is motivated by a number of low rank approximation applications in which $k$ is typically small. The additive error of $2^{-\text{poly}(n)}$ is only needed in order to write down our solution $B$ in the unit cost RAM model, since in general the entries of $B$ may be irrational, even if the entries of $A$ are specified by poly$(n)$ bits. If instead we only want to output an approximation to the value $\|A - A_k\|_F^2$, then we can output a number $Z$ for which $\text{OPT} \leq Z \leq (1 + \epsilon)\text{OPT}$, that is, we do not incur additive error.
When \( A_k \) does not exist, there still exists a rank-\( k \) tensor \( \tilde{A} \) for which \( \| A - \tilde{A} \|_F^2 \leq \text{OPT} + \gamma \). We require there exists such a \( \tilde{A} \) for which if \( \tilde{A} = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i \), then the norms \( \| u_i \|_2, \| v_i \|_2, \) and \( \| w_i \|_2 \) are bounded by \( 2^{\text{poly}(n)} \).

The assumption in the previous two paragraphs that the factors of \( A_k \) and of \( \tilde{A} \) have norm bounded by \( 2^{\text{poly}(n)} \) is necessary in certain cases, e.g., if \( \text{OPT} = 0 \) and we are to write down the factors in \( \text{poly}(n) \) time. An abridged version of our theorem is as follows.

**Theorem 1.2 (Combination of Theorem C.1 and C.2, rank-\( k \)).** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( \delta > 0 \), if \( A_k = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i \) exists and each of \( \| u_i \|_2, \| v_i \|_2, \) and \( \| w_i \|_2 \) is bounded by \( 2^{\text{poly}(n)} \), then there is a randomized algorithm running in \( O(mnz(A) + n \text{ poly}(k, 1/\epsilon) + 2^{O(k^2/\epsilon)} \cdot n^\delta \) time in the unit cost RAM model with words of size \( O(\log n) \) bits\(^3\), which outputs a (factorization of a) rank-\( k \) tensor \( B \) for which \( \| A - B \|_F^2 \leq (1 + \epsilon) \| A - A_k \|_F^2 + 2^{-O(n^\delta)} \). Further, we can output a number \( Z \) for which \( \text{OPT} \leq Z \leq (1 + \epsilon) \text{OPT} \) in the same amount of time. When \( A_k \) does not exist, if there exists a rank-\( k \) tensor \( \bar{A} \) for which \( \| A - \bar{A} \|^2_F \leq \text{OPT} + 2^{-O(n^\delta)} \) and \( \bar{A} = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i \) is such that the norms \( \| u_i \|_2, \| v_i \|_2, \) and \( \| w_i \|_2 \) are bounded by \( 2^{O(n^\delta)} \), then we can output a (factorization of a) rank-\( k \) tensor \( \tilde{A} \) for which \( \| A - \tilde{A} \|_F^2 \leq (1 + \epsilon) \text{OPT} + 2^{-O(n^\delta)} \).

Our techniques for proving Theorem 1.1 and Theorem 1.2 open up avenues for many other problems in linear algebra on tensors. We now define the problems and state our results for them.

There is a long line of research on matrix column subset selection and CUR decomposition [DMM08, BMD09, DR10, BDM11, FEGK13, BW14, WS15, ABF+16, SWZ17] under operator, Frobenius, and entry-wise \( \ell_1 \) norm. It is natural to consider tensor column subset selection or tensor-CURT\(^4\), however most previous works either give error bounds in terms of the tensor flattenings [DMM08], assume the original tensor has certain properties [OST08, FT15, TM17], consider the problems in linear algebra on tensors. We now define the problems and state our results for them.

There is a long line of research on matrix column subset selection and CUR decomposition [DMM08, BMD09, DR10, BDM11, FEGK13, BW14, WS15, ABF+16, SWZ17] under operator, Frobenius, and entry-wise \( \ell_1 \) norm. It is natural to consider tensor column subset selection or tensor-CURT\(^4\), however most previous works either give error bounds in terms of the tensor flattenings [DMM08], assume the original tensor has certain properties [OST08, FT15, TM17], consider the exact case which assumes the tensor has low rank [CC10], or only fit a high dimensional cross-shape to the tensor rather than to all of its entries [FMMN11]. Such works are not able to provide a \((1 + \epsilon)\)-approximation guarantee as in the matrix case without assumptions. We consider tensor column, row, and tube subset selection, with the goal being to find three matrices: a subset \( C \in \mathbb{R}^{n \times c} \) of columns of \( A \), a subset \( R \in \mathbb{R}^{n \times r} \) of rows of \( A \), and a subset \( T \in \mathbb{R}^{n \times t} \) of tubes of \( A \), such that there exists a tensor \( U \in \mathbb{R}^{c \times r \times t} \) for which

\[
\| U(C, R, T) - A \|_\xi \leq \alpha \| A_k - A \|_\xi + \gamma,
\]

where \( \gamma = 0 \) if \( A_k \) exists and \( \gamma = 2^{-\text{poly}(n)} \) otherwise, \( \alpha > 1 \) is the approximation ratio, \( \xi \) is either Frobenius norm or Entry-wise \( \ell_1 \) norm, and \( U(C, R, T) = \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l \). In tensor CURT decomposition, we also want to output \( U \).

We provide a (nearly) input sparsity time algorithm for this, together with an alternative input sparsity time algorithm which chooses slightly larger factors \( C, R, \) and \( T \).

To do this, we combine Theorem 1.1 with the following theorem which, given a factorization of a rank-\( k \) tensor \( B \), obtains \( C, U, \) and \( T \) in terms of it:

**Theorem 1.3 (Combination of Theorem C.40 and C.41, \( ||| \cdot |||_F \) norm, CURT decomposition).** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), let \( k \geq 1 \), and let \( U_B, V_B, W_B \in \mathbb{R}^{n \times k} \) be given. There is an algorithm running in \( O(mnz(A) \log n) + \widetilde{O}(n^2) \text{ poly}(k, 1/\epsilon) \) time (respectively, \( O(mnz(A)) + n \text{ poly}(k, 1/\epsilon) \) time) which outputs a subset \( \hat{C} \in \mathbb{R}^{n \times c} \) of columns of \( A \), a subset \( \hat{R} \in \mathbb{R}^{n \times r} \) of rows of \( A \), a subset \( \hat{T} \in \mathbb{R}^{n \times t} \) of tubes of \( A \), together with a tensor \( \hat{U} \in \mathbb{R}^{c \times r \times t} \) with rank(\( \hat{U} \)) = \( k \) such that \( c = r = t = O(k/\epsilon) \) (respectively, \( c = r = t = O(k \log k + k/\epsilon) \)), and \( \| U(C, R, T) - A \|_F^2 \leq (1 + \epsilon) \| U_B \otimes V_B \otimes W_B - A \|_F^2 \) holds with probability at least 9/10.

---

\(^3\)The entries of \( A \) are assumed to fit in \( n^w \) words.

\(^4\)T denotes the tube which is the column in 3rd dimension of tensor.
Combining Theorems 1.2 and 1.3 (with \( B \) being a \((1 + O(\epsilon))\)-approximation to \( A \)) we achieve Equation (2) with \( \alpha = (1 + \epsilon) \) and \( \xi = F \) with the optimal number of columns, rows, tubes, and rank of \( U \) (we mention our matching lower bound later), though the running time has an \( 2^{O(k^2/\epsilon)} \) term in it. We note that instead combining Theorem 1.1 and Theorem 1.3 gives a bicriteria result.

Our results for asymmetric norms can be extended to tensor \( \| \cdot \|_{\ell_p} \) norm low-rank approximation, as well as results for asymmetric tensor norms, which are natural extensions of the matrix \( \ell_1-\ell_2 \) norm. Here, for a tensor \( A, \| A \|_v = \sum_i (\sum_{j,k} (A_{i,j,k})^2)^{\frac{1}{2}} \) and \( \| A \|_u = \sum_{i,j} (\sum_k (A_{i,j,k})^2)^{\frac{1}{2}} \).

**Theorem 1.4** (Combination of Theorem D.14 (\( \| \cdot \|_1 \)-norm), Theorem E.9 (\( \| \cdot \|_p \)-norm, \( p \in (0,1) \)) Theorem F.23 (\( \| \cdot \|_{\ell_{p}-\ell_{q}} \)-norm or \( \ell_{1}-\ell_{2} \)-norm), Theorem E.9 (\( \| \cdot \|_{\ell_{p}-\ell_{q}} \)-norm or \( \ell_{1}-\ell_{2} \)-norm)). Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), let \( r = \tilde{O}(k^2) \). If \( A_k \) exists then there is an algorithm which runs in \( \text{nnz}(A) \cdot t + \tilde{O}(n) \text{poly}(k) \) time and outputs a (factorization of a) rank-\( r \) tensor \( B \) for which \( \| B - A \|_{\ell_2} \leq \text{poly}(k, \log n) \cdot \| A_k - A \|_{\ell_2} \) holds. If \( A_k \) does not exist, we have \( \| B - A \|_{\ell_2} \leq \text{poly}(k, \log n) \cdot \text{OPT} + \gamma \), where \( \gamma \) is an arbitrarily small positive function of \( n \). The success probability is at least \( 9/10 \). For \( \xi = v, t = O(1) \); for \( \xi = u, t = O(n) \).

As in the case of Frobenius norm, we can get rank-\( k \) and CURT algorithms for the above norms. Our results for asymmetric norms can be extended to \( \ell_p-\ell_2-\ell_2, \ell_p-\ell_p-\ell_2 \), and families of M-estimators.

We also obtain the following result for weighted tensor low-rank approximation.

**Theorem 1.5** (Informal Version of Theorem G.5, weighted). Suppose we are given a third order tensor \( A \in \mathbb{R}^{n \times n \times n} \), as well as a tensor \( W \in \mathbb{R}^{n \times n \times n} \) with \( r \) distinct rows and \( r \) distinct columns. Suppose there is a rank-\( k \) tensor \( A' \in \mathbb{R}^{n \times n \times n} \) for which \( \| W \circ (A' - A) \|_F^2 = \text{OPT} \) and one can write \( A' = \sum_{i=1}^k u_i \otimes v_i \otimes w_i \) for \( \| u_i \|_2, \| v_i \|_2, \| w_i \|_2 \) bounded by \( 2^{\delta} \). Then there is an algorithm running in \( (\text{nnz}(A) + \text{nnz}(W) + n2^{O(v^2k^2/\epsilon)}) \cdot n^\delta \) time and outputting \( n \times k \) matrices \( U_1, U_2, U_3 \) for which \( \| W \circ (U_1 \otimes U_2 \otimes U_3 - A) \|_F^2 \leq (1 + \epsilon) \text{OPT} \) with probability at least \( 2/3 \).

We next strengthen Håstad’s NP-hardness to show that even approximating tensor rank is hard (we note at the time of Håstad’s NP-hardness, there was no PCP theorem available; nevertheless we need to do additional work here):

**Theorem 1.6** (Informal Version of Theorem H.42). Let \( q \geq 3 \). Unless the Exponential Time Hypothesis (ETH) fails, there is an absolute constant \( c_0 > 1 \) for which distinguishing if a tensor in \( \mathbb{R}^{n^q} \) has rank at most \( k \), or at least \( c_0 \cdot k \), requires \( 2^{k^{1-o(1)}} \) time, for a constant \( \delta > 0 \).

Under random-ETH [Fei02, GL04, RSW16], an average case hardness assumption for 3SAT, we can replace the \( k^{1-o(1)} \) in the exponent above with a \( k \). We also obtain hardness in terms of \( \epsilon \):

**Theorem 1.7** (Informal Version of Corollary H.22). Let \( q \geq 3 \). Unless ETH fails, there is no algorithm running in \( 2^{2^{O(1/\epsilon q)}} \) time which, given a tensor \( A \in \mathbb{R}^{n^q} \), outputs a rank-1 tensor \( B \) for which \( \| A - B \|_F^2 \leq (1 + \epsilon) \text{OPT} \).

As a side result worth stating, our analysis improves the best matrix CUR decomposition algorithm under Frobenius norm [BW14], providing the first optimal \( \text{nnz}(A) \)-time algorithm:

**Theorem 1.8** (Informal Version of Theorem C.48, Matrix CUR decomposition). There is an algorithm, which given a matrix \( A \in \mathbb{R}^{n \times d} \) and an integer \( k \geq 1 \), runs in \( O(\text{nnz}(A)) + (n + d) \text{poly}(k, 1/\epsilon) \) time and outputs three matrices: \( C \in \mathbb{R}^{n \times c} \) containing \( c \) columns of \( A, R \in \mathbb{R}^{r \times d} \) containing \( r \) rows of \( A, \) and \( U \in \mathbb{R}^{c \times r} \) with rank \( (U) = k \) for which \( r = c = O(k/\epsilon) \) and \( \| \text{CUR} - A \|_F^2 \leq (1 + \epsilon) \min_{\text{rank}-k} \| A_k - A \|_F^2 \), holds with probability at least \( 9/10 \).
1.2 Our Techniques

Many of our proofs, in particular those for Theorem 1.1 and Theorem 1.2, are based on what we call an “iterative existential proof”, which we then turn into an algorithm in two different ways depending if we are proving Theorem 1.1 or Theorem 1.2.

Henceforth, we assume $A_k$ exists; otherwise replace $A_k$ with a suitably good tensor $\tilde{A}$ in what follows. Since $A_k = \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i^5$, we can create three $n \times k$ matrices $U^*$, $V^*$, and $W^*$ whose columns are the vectors $U_i^*$, $V_i^*$, and $W_i^*$, respectively. Now we consider the three different flattenings (or unfoldings) of $A_k$, which express $A_k$ as an $n \times n^2$ matrix. Namely, by thinking of $A_k$ as the sum of outer products, we can write the three flattenings of $A_k$ as $U^* \cdot Z_1$, $V^* \cdot Z_2$, and $W^* \cdot Z_3$, where the rows of $Z_1$ are $\text{vec}(V_i^* \otimes W_i^*)^6$ (For simplicity, we write $Z_1 = (V^{*\top} \otimes W^{*\top})^7$), the rows of $Z_2$ are $\text{vec}(U_i^* \otimes W_i^*)$, and the rows of $Z_3$ are $\text{vec}(U_i^* \otimes V_i^*)$, for $i \in [k] = \{1, 2, \ldots, k\}$. Letting the three corresponding flattenings of the input tensor $A$ be $A_1$, $A_2$, and $A_3$, by the symmetry of the Frobenius norm, we have $\|A - B\|_F^2 = \|A_1 - U^* Z_1\|_F^2 = \|A_2 - V^* Z_2\|_F^2 = \|A_3 - W^* Z_3\|_F^2$.

Let us consider the hypothetical regression problem $\min_U \|A_1 - U Z_1\|_F^2$. Note that we do not know $Z_1$, but we will not need to. Let $r = O(k/\epsilon)$, and suppose $S_1$ is an $r \times r$ matrix of i.i.d. normal random variables with mean 0 and variance $1/r$, denoted $N(0, 1/r)$. Then by standard results for regression (see, e.g., [Woo14] for a survey), if $\hat{U}$ is the minimizer to the smaller regression problem $\hat{U} = \arg\min_U \|U Z_1 - A_1 S_1\|_F^2$, then

$$\|A_1 - \hat{U} Z_1\|_F^2 \leq (1 + \epsilon)\min_U \|A_1 - U Z_1\|_F^2.$$  

Moreover, $\hat{U} = A_1 S_1 (Z_1 S_1)^\dagger$. Although we do not know $Z_1$, this implies $\hat{U}$ is in the column span of $A_1 S_1$, which we do know, since we can flatten $A$ to compute $A_1$, and then compute $A_1 S_1$. Thus, this hypothetical regression argument gives us an existential statement - there exists a good rank-$k$ matrix $\hat{U}$ in the column span of $A_1 S_1$. We could similarly define $\hat{V} = A_2 S_2 (Z_2 S_2)^\dagger$ and $\hat{W} = A_3 S_3 (Z_3 S_3)^\dagger$ as solutions to the analogous regression problems for the other two flattenings of $A$, which are in the column spans of $A_2 S_2$ and $A_3 S_3$, respectively. Given $A_1 S_1, A_2 S_2, A_3 S_3$, which we know, we could hope there is a good rank-$k$ tensor in the span of the rank-1 tensors

$$\{(A_1 S_1)_a \otimes (A_2 S_2)_b \otimes (A_3 S_3)_c\}_{a,b,c \in [r]}.$$  

However, an immediate issue arises. First, note that our hypothetical regression problem guarantees that $\|A_1 - \hat{U} Z_1\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$, and therefore since the rows of $Z_1$ are of the special form $\text{vec}(V_i^* \otimes W_i^*)$, we can perform a “retensorization” to create a rank-$k$ tensor $B = \sum_i \hat{U}_i \otimes V_i^* \otimes W_i^*$ from the matrix $\hat{U} Z_1$ for which $\|A - B\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$. While we do not know $\hat{U}$, since it is in the column span of $A_1 S_1$, it implies that $B$ is in the span of the rank-1 tensors $\{(A_1 S_1)_a \otimes V_b^* \otimes W_c^*\}_{a \in [r], b,c \in [k]}$. Analogously, we have that there is a good rank-$k$ tensor $B$ in the span of the rank-1 tensors $\{(U_a^* \otimes (A_2 S_2)_b \otimes W_c^*)\}_{a,c \in [k], b \in [r]}$, and a good rank-$k$ tensor $B$ in the span of the rank-1 tensors $\{(U_a^* \otimes V_b^* \otimes (A_3 S_3)_c\}_{a,b \in [k], c \in [r]}$. However, we do not know $U^*$ or $V^*$, and it is not clear there is a rank-$k$ tensor $B$ for which simultaneously its first factors are in the column span of $A_1 S_1$, its second factors are in the column span of $A_2 S_2$, and its third factors are in the column span of $A_3 S_3$, i.e., whether there is a good rank-$k$ tensor $B$ in the span of rank-1 tensors in (4).

We fix this by an iterative argument. Namely, we first compute $A_1 S_1$, and write $\hat{U} = A_1 S_1 (Z_1 S_1)^\dagger$. We now redefine $Z_2$ with respect to $\hat{U}$, so the rows of $Z_2$ are $\text{vec}(\hat{U}_i \otimes W_i^*)$ for $i \in [k]$, and consider

5For simplicity, we define $U \otimes V \otimes W = \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i$, where $U_i$ is the $i$-th column of $U$.

6vec($V_i^* \otimes W_i^*$) denotes a row vector that has length $n_1 n_2$ where $V_i^*$ has length $n_1$ and $W_i^*$ has length $n_2$.

7$(V^{*\top} \otimes W^{*\top})$ denotes a $k \times n_1 n_2$ matrix where the $i$-th row is vec($V_i^* \otimes W_i^*$), where length $n_1$ vector $V_i^*$ is the $i$-th column of $n_1 \times k$ matrix $V^*$, and length $n_2$ vector $W_i^*$ is the $i$-th column of $n_2 \times k$ matrix $W^*$, $V_i \in [k]$. 

9
the regression problem $\min_V \| A_2 - VZ_2 \|_F^2$. While we do not know $Z_2$, if $S_2$ is an $n^2 \times r$ matrix of
i.i.d. Gaussians, we again have the statement that $\hat{V} = A_2S_2(Z_2S_2)^\dagger$ satisfies

$$
\| A_2 - \hat{V}Z_2 \|_F^2 \leq (1 + \epsilon)\min_V \| A_2 - VZ_2 \|_F^2
$$
by the regression guarantee with Gaussians

$$
\leq (1 + \epsilon)\| A_2 - V^*Z_2 \|_F^2
$$
since $V^*$ is no better than the minimizer $V$

$$
= (1 + \epsilon)\| A_1 - \hat{U}Z_1 \|_F^2
$$
by retensorizing and flattening along a different dimension

$$
\leq (1 + \epsilon)^2\min_U \| A_1 - UZ_1 \|_F^2
$$
by (3)

$$
= (1 + \epsilon)^2\| A - A_k \|_F^2
$$
by definition of $Z_1$.

Now we can retensorize $\hat{V}Z_2$ to obtain a rank-$k$ tensor $B$ for which $\| A - B \|_F^2 \leq (1 + \epsilon)^2\| A - A_k \|_F^2$, Note that since the columns of $\hat{V}$ are in the span of $A_2S_2$, and the rows of $Z_2$ are $\text{vec}(U_i \otimes W_i^\top)$ for $i \in [k]$, where the columns of $\hat{U}$ are in the span of $A_1S_1$, it follows that $B$ is in the span of rank-1 tensors $\{(A_1S_1)_a \otimes (A_2S_2)_b \otimes \hat{V}_c \}_{a,b,c \in [r], c \in [k]}$.

Suppose we now redefine $Z_3$ so that it is now an $r^2 \times n^2$ matrix with rows $\text{vec}((A_1S_1)_a \otimes (A_2S_2)_b)$ for all pairs $a,b \in [r]$, and consider the regression problem $\min_V \| A_3 - WZ_3 \|_F^2$. Now observe that since we know $Z_3$, and since we can form $A_3$ by flattening $A$, we can solve for $W \in \mathbb{R}^{n \times r^2}$ in polynomial time by solving a regression problem. Retensorizing $WZ_3$ to a tensor $B$, it follows that we have found a rank-$r^2 = O(k^2/\epsilon^2)$ tensor $B$ for which $\| A - B \|_F^2 \leq (1 + \epsilon)^2\| A - A_k \|_F^2 = (1 + O(\epsilon))\| A - A_k \|_F^2$, and the result follows by adjusting $\epsilon$ by a constant factor.

To obtain the $\text{nnz}(A) + n\text{poly}(k/\epsilon)$ running time guarantee of Theorem 1.1, while we can replace $S_1$ and $S_2$ with compositions of a sparse CountSketch matrix and a Gaussian matrix (see chapter 2 of [Woo14] for a survey), enabling us to compute $A_1S_1$ and $A_2S_2$ in $\text{nnz}(A) + n\text{poly}(k/\epsilon)$ time, we still need to solve the regression problem $\min_V \| A_3 - WZ_3 \|_F^2$ quickly, and note that we cannot even write down $Z_3$ without spending $r^2n^2$ time. Here we use a different random matrix $S_3$ called TensorSketch, which was introduced in [Pag13, PP13], but for which we will need the stronger properties of a subspace embedding and approximate matrix product shown to hold for it in [ANW14]. Given the latter properties, we can instead solve the regression problem $\min_V \| A_3 - WZ_3 \|_F^2$, and importantly $A_3S_3$ and $Z_3S_3$ can be computed in $\text{nnz}(A) + n\text{poly}(k/\epsilon)$ time. Finally, this small problem can be solved in $n\text{poly}(k/\epsilon)$ time.

If we want to output a rank-$k$ solution as in Theorem 1.2, then we need to introduce indeterminates at several places in the preceding argument and run a generic polynomial optimization procedure which runs in time exponential in the number of indeterminates. Namely, we write $\tilde{U}$ as $A_1S_1X_1$, where $X_1$ is an $r \times k$ matrix of indeterminates, we write $\tilde{V}$ as $A_2S_2X_2$, where $X_2$ is an $r \times k$ matrix of indeterminates, and we write $\tilde{W}$ as $A_3S_3X_3$, where $X_3$ is an $r \times k$ matrix of indeterminates. When executing the above iterative argument, we let the rows of $Z_1$ be the vectors $\text{vec}(V_i^\star \otimes W_i^\star)$, the rows of $Z_2$ be the vectors $\text{vec}(U_i \otimes V_i^\star)$, and the rows of $Z_3$ be the vectors $\text{vec}(\tilde{U}_i \otimes \tilde{V}_i)$. Then $\tilde{U}$ is a $(1 + \epsilon)$-approximate minimizer to $\min_{V_i} \| A_1 - UZ_1 \|_F$, while $\tilde{V}$ is a $(1 + \epsilon)$-approximate minimizer to $\min_{V_i} \| A_2 - VZ_2 \|_F$, while $\tilde{W}$ is a $(1 + \epsilon)$-approximate minimizer to $\min_{V_i} \| A_3 - WZ_3 \|_F$.

Note that by assigning $X_1 = (Z_1S_1)^\dagger$, $X_2 = (Z_2S_2)^\dagger$, and $X_3 = (Z_3S_3)^\dagger$, it follows that the rank-$k$ tensor $B = \sum_{i=1}^k (A_1S_1X_1)_a \otimes (A_2S_2X_2)_b \otimes (A_3S_3X_3)_c$, satisfies $\| A - B \|_F^2 \leq (1 + \epsilon)^3\| A - A_k \|_F^2$, as desired. Note that here the rows of $Z_2$ are a function of $X_1$, while the rows of $Z_3$ are a function of both $X_1$ and $X_2$. What is important for us though is that it suffices to minimize the degree-6 polynomial $\sum_{a,b,c \in [n]}(\sum_{i=1}^k (A_1S_1X_1)_a,i \cdot (A_2S_2X_2)_b,i \cdot (A_3S_3X_3)_c,i - A_{a,b,c})^2$, over the $3rk = O(k^2/\epsilon)$ indeterminates $X_1, X_2, X_3$, since we know there exists an assignment to $X_1, X_2, X_3$ providing a $(1 + O(\epsilon))$-approximate solution, and any solution $X_1, X_2, X_3$ found by minimizing the above polynomial will be no worse than that solution. This polynomial can be minimized up to additive $2 - \text{poly}(n)$ additive error in $\text{poly}(n)$ time [Ren92a, BPR96] assuming the entries of $U^\star, V^\star$, and $W^\star$
are bounded by $2^{\text{poly}(n)}$, as assumed in Theorem 1.2. Similar arguments can be made for obtaining a relative error approximation to the actual value $\text{OPT}$ as well as handling the case when $A_k$ does not exist.

To optimize the running time to $\text{nnz}(A)$, we can choose CountSketch matrices $T_1, T_2, T_3$ of $t = \text{poly}(k, 1/\epsilon) \times n$ dimensions and reapply the above iterative argument. Then it suffices to minimize this small size degree-6 polynomial
\[
\sum_{a,b,c} (\sum_{i=1}^{k} (T_1 A_i S_1 X_1)_{a,i} \cdot (T_2 A_2 S_2 X_2)_{b,i} \cdot (T_3 A_3 S_3 X_3)_{c,i} - (A(T_1, T_2, T_3))_{a,b,c})^2,
\]
over the $3rk = O(k^2/\epsilon)$ indeterminates $X_1, X_2, X_3$. Outputting $A_1 S_1 X_1, A_2 S_2 X_2, A_3 S_3 X_3$ then provides a $(1 + \epsilon)$-approximate solution.

Our iterative existential argument provides a general framework for obtaining low rank approximation results for tensors for many other error measures as well.

### 1.3 Other Low Rank Approximation Algorithms Following Our Framework

**Column, row, tube subset selection, and CURT decomposition.** In tensor column, row, tube subset selection, the goal is to find three matrices: a subset $C$ of columns of $A$, a subset $R$ of rows of $A$, and a subset $T$ of tubes of $A$, such that there exists a small tensor $U$ for which $\|U(C, R, T) - A\|^2_F \leq (1 + \epsilon) \text{OPT}$. We first choose two Gaussian matrices $S_1$ and $S_2$ with $s_1 = s_2 = O(k/\epsilon)$ columns, and form a matrix $Z_3' \in \mathbb{R}^{(s_1 s_2) \times n^2}$ with $(i, j)$-th row equal to the vectorization of $(A_1 S_1)_i \otimes (A_2 S_2)_j$. Motivated by the regression problem $\min_U \|A_3 - WZ_3'\|_F$, we sample $d_3 = O(s_1 s_2/\epsilon)$ columns from $A_3$ and let $D_3$ denote this selection matrix. There are a few ways to do the sampling depending on the tradeoff between the number of columns and running time, which we describe below. Proceeding iteratively, we write down $Z_3'$ by setting its $(i, j)$-th row to the vectorization of $(A_1 S_1)_i \otimes (A_3 D_3)_j$. Then we choose two Gaussian matrices $S_1$ and $S_2$ with $s_1 = s_2 = O(k/\epsilon)$ columns, and form a matrix $Z_3'' \in \mathbb{R}^{(s_1 s_2) \times n^2}$ with $(i, j)$-th row equal to the vectorization of $(A_2 D_2)_i \otimes (A_3 D_3)_j$. We obtain $C = A_1 D_1$, $R = A_2 D_2$ and $T = A_3 D_3$. For the sampling steps, we can use a generalized matrix column subset selection technique, which extends a column subset selection technique of [BW14] in the context of CUR decompositions to the case when $C$ is not necessarily a subset of the input. This gives $O(\text{nnz}(A) \log n) + \tilde{O}(n^2) \text{poly}(k, 1/\epsilon)$ time. Alternatively, we can use a technique we developed called tensor leverage score sampling described below, yielding $O(\text{nnz}(A)) + n \text{poly}(k, 1/\epsilon)$ time.

A body of work in the matrix case has focused on finding the best possible number of columns and rows of a CUR decomposition, and we can ask the same question for tensors. It turns out that if one is given the factorization $\sum_{i=1}^{k} (U_B)_i \otimes (V_B)_i \otimes (W_B)_i$ of a rank-$k$ tensor $B \in \mathbb{R}^{n \times n \times n}$ with $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$, then one can find a set $C$ of $O(k/\epsilon)$ columns, a set $R$ of $O(k/\epsilon)$ rows, and a set $T$ of $O(k/\epsilon)$ tubes of $A$, together with a rank-$k$ tensor $U$ for which $\|U(C, R, T) - A\|^2_F \leq (1 + \epsilon) \|A - B\|^2_F$. This is based on an iterative argument, where the initial sampling (which needs to be our generalized matrix column subset selection rather than tensor leverage score sampling to achieve optimal bounds) is done with respect to $V_B^T \otimes W_B^T$, and then an iterative argument is carried out. Since we show a matching lower bound on the number of columns, rows, tubes and rank of $U$, these parameters are tight. The algorithm is efficient if one is given a rank-$k$ tensor $B$ which is a $(1 + O(\epsilon))$-approximation to $A$; if not then one can use Theorem C.2 and this step will be exponential time in $k$. If one just wants $O(k \log k + k/\epsilon)$ columns, rows, and tubes, then one can achieve $O(\text{nnz}(A)) + n \text{poly}(k, 1/\epsilon)$ time, if one is given $B$.

**Column-row, row-tube, tube-column face subset selection, and CURT decomposition.** In tensor column-row, row-tube, tube-column face subset selection, the goal is to find three tensors: a subset $C \in \mathbb{R}^{c \times n \times n}$ of row-tube faces of $A$, a subset $R \in \mathbb{R}^{n \times r \times n}$ of tube-column faces of $A$, and a subset $T \in \mathbb{R}^{n \times n \times t}$ of column-row faces of $A$, such that there exists a tensor $U \in \mathbb{R}^{l \times c \times n \times r \times n}$
with small rank for which \( \|U(T_1, C_2, R_3) - A\|_F^2 \leq (1 + \epsilon) \) OPT, where \( T_1 \in \mathbb{R}^{n \times tn} \) denotes the matrix obtained by flattening the tensor \( T \) along the first dimension, \( C_2 \in \mathbb{R}^{n \times cn} \) denotes the matrix obtained by flattening the tensor \( C \) along the second dimension, and \( R_3 \in \mathbb{R}^{n \times rn} \) denotes the matrix obtained by flattening the tensor \( T \) along the third dimension.

We solve this problem by first choosing two Gaussian matrices \( S_1 \) and \( S_2 \) with \( s_1 = s_2 = O(k/\epsilon) \) columns, and then forming matrix \( U_3 \in \mathbb{R}^{n \times s_1 s_2} \) with \((i, j)\)-th column equal to \((A_1 S_1)_i\), as well as matrix \( V_3 \in \mathbb{R}^{n \times s_1 s_2} \) with \((i, j)\)-th column equal to \((A_2 S_2)_j\). Inspired by the regression problem \( \min_{W \in \mathbb{R}^{n \times s_1 s_2}} \|V_3 \cdot (W^T \circ U_3) - A_2\|_F \), we sample \( d_3 = O(s_1 s_2/\epsilon) \) rows from \( A_2 \) and let \( D_3 \in \mathbb{R}^{n \times s_1 s_2} \) denote this selection matrix. In other words, \( D_3 \) selects \( d_3 \) tube-column faces from the original tensor \( A \). Thus, we obtain a small regression problem: \( \min_W \|D_3 V_3 \cdot (W^T \circ U_3^T) - D_3 A_2\|_F \). By retensorizing the objective function, we obtain the problem \( \min_W \|U_3 \cdot (D_3 V_3) \circ W - A(I,D_3,I)\|_F \). Flattening the objective function along the third dimension, we obtain \( \min_W \|W \cdot (U_3^T \circ (D_3 V_3)^T) - (A(I,D_3,I))_3\|_F \) which has optimal solution \((A(I,D_3,I))_3 (U_3^T \circ (D_3 V_3)^T)\). Let \( W' \) denote \((A(I,D_3,I))_3\). In the next step, we fix \( W = \exp(W_3^T) \odot (D_3 V_3)\) and consider the objective function \( \min_W \|U_2 \cdot (V_3^T \circ W'_2) - A_1\|_F \). Applying a similar argument, we obtain \( V' = (A(D_2,I,I))_2 \) and \( U' = (A(I,I,D_1))_1 \). Let \( C \) denote \((A(D_2,I,I))_2 \), \( R \) denote \((A(D_3,I,I))_3 \), and \( T \) denote \((A(I,I,D_1))_1 \). Overall, this algorithm selects \( \text{poly}(k,1/\epsilon) \) faces from each dimension.

Similar to our column-based CURT decomposition, our face-based CURT decomposition has the property that if one is given the factorization \( \sum_i \sum_j A_B \odot (V_B) \odot (W_B) \) of a rank-\( K \) tensor \( B \in \mathbb{R}^{n \times s_1 s_2} \) with \( U_B, V_B, W_B \in \mathbb{R}^{n \times k} \) which is a \((1 + O(\epsilon))\)-approximation to \( A \), then one can find a set \( C \) of \( o(k/\epsilon) \) row-tube faces, a set \( R \) of \( \text{poly}(k/\epsilon) \) tube-column faces, and a set \( T \) of \( \text{poly}(k/\epsilon) \) column-row faces of \( A \), together with a rank-\( k \) tensor \( U \) for which \( \|U(T_1, C_2, R_3) - A\|_F^2 \leq (1 + \epsilon) \) OPT.

**Tensor multiple regression and tensor leverage score sampling.** In the above we need to consider standard problems for matrices in the context of tensors. Suppose we are given a matrix \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and a matrix \( B = (V^T \odot W^T) \in \mathbb{R}^{k \times n_2 n_3} \) with rows \((V_i \odot W_i)\) for an \( n_2 \times k \) matrix \( V \) and \( n_3 \times k \) matrix \( W \). Using TensorSketch [Pag13, PP13, ANW14] one can solve multiple regression \( \min_{B} \|UB - A\|_F \) without forming \( B \) in \( \mathcal{O}(n_2 n_3 \text{poly}(k,1/\epsilon)) \) time, rather than the naïve \( \mathcal{O}(n_2 n_3 \text{poly}(k,1/\epsilon)) \) time. However, this does not immediately help us if we would like to sample columns of such a matrix \( B \) proportional to its leverage scores. Even if we apply TensorSketch to compute a \( k \times k \) change of basis matrix \( R \in \mathcal{O}(n_2 n_3 \text{poly}(k,\log(n_2 n_3))) \) time, for which the leverage scores of \( B \) are (up to a constant factor) the squared column norms of \( R^{-1} B \), there are still \( n_2 n_3 \) leverage scores and we cannot write them all down! Nevertheless, we show we can still sample by them by using that the matrix of interest is formed via a tensor product, which can be rewritten as a matrix multiplication which we never need to explicitly materialize. In more detail, for the \( i \)-th row \( e_i R^{-1} \) of \( R^{-1} \), we create a matrix \( V^{ei} \) by scaling each of the columns of \( V^T \) entrywise by the entries of \( e_i \). The squared norms of \( e_i R^{-1} B \) are exactly the squared entries of \((V^{ei}) W^T \). We cannot compute this matrix product, but we can first sample a column of it proportional to its squared norm and then sample an entry in that column proportional to its square. To sample a column, we compute \( G(V^{ei}) W^T \) for a Gaussian matrix \( G \) with \( O(\log n_3) \) rows by computing \( G \cdot V^{ei} \), then computing \((G \cdot V^{ei}) \cdot W^T \), which is \( O(n_2 n_3 \text{poly}(k,\log(n_2 n_3))) \) total time. After sampling a column, we compute the column exactly and sample a squared entry. We do this for each \( i \in [k] \), first sampling an \( i \) proportional to \( \|GV^{ei} W^T\|_F^2 \), then running the above scheme on that \( i \). The \( \text{poly}(\log n) \) factor in the running time can be replaced by \( \text{poly}(k) \) if one wants to avoid a \( \text{poly}(\log n) \) dependence in the running time.
**Entry-wise $\ell_1$ low-rank approximation.** We consider the problem of entrywise $\ell_1$-low rank approximation of an $n \times n \times n$ tensor $A$, namely, the problem of finding a rank-$k$ tensor $B$ for which 

$$\|A - B\|_1 \leq \text{poly}(k, \log n) \text{ OPT},$$

where $\text{OPT} = \inf_{\text{rank-}k \ B} \|A - B\|_1$, and where for a tensor $A$, 

$$\|A\|_1 = \sum_{i,j,k} |A_{i,j,k}|.$$ 

Our iterative existential argument can be applied in much the same way as for the Frobenius norm. We iteratively flatten $A$ along each of its three dimensions, obtaining $A_1$, $A_2$, and $A_3$ as above, and iteratively build a good rank-$k$ solution $B$ of the form $(A_1S_1X_1) \otimes (A_2S_2X_2) \otimes (A_3S_3X_3)$, where now the $S_i$ are matrices of i.i.d. Cauchy random variables or sparse matrices of Cauchy random variables and the $X_i$ are $O(k \log k) \times k$ matrices of indeterminates. For a matrix $C$ and a matrix $S$ of i.i.d. Cauchy random variables with $k$ columns, it is known [SWZ17] that the column span of $CS$ contains a poly($k \log n$)-approximate rank-$k$ space with respect to the entrywise $\ell_1$-norm for $C$. In the case of tensors, we must perform an iterative flattening and retensorizing argument to guarantee there exists a tensor $B$ of the form above. Also, if we insist on outputting a rank-$k$ solution as opposed to a bicriteria solution, $\|A_1S_1X_1\|_1 \otimes (A_2S_2X_2) \otimes (A_3S_3X_3) - A\|_1$ is not a polynomial of the $X_i$, and if we introduce sign variables for the $n^3$ absolute values, the running time of the polynomial solver will be $2^\# \text{ of variables} = 2^\Omega(n^3)$. We perform additional dimensionality reduction by Lewis weight sampling [CP15] from the flattenings to reduce the problem size to poly($k$). This small problem still has $\tilde{O}(k^3)$ sign variables, and to obtain a $2^{\tilde{O}(k^2)}$ running time we relax the reduced problem to a Frobenius norm problem, mildly increasing the approximation factor by another poly($k$) factor.

Combining the iterative existential argument with techniques in [SWZ17], we also obtain an $\ell_1$ CURT decomposition algorithm (which is similar to the Frobenius norm result in Theorem 1.3), which can find $\tilde{O}(k)$ columns, $O(k)$ rows, $\tilde{O}(k)$ tubes, and a tensor $U$. Our algorithm starts from a given factorization of a rank-$k$ tensor $B = U_B \otimes V_B \otimes W_B$ found above. We compute a sampling and rescaling diagonal matrix $D_1$ according to the Lewis weights of matrix $B_1 = (V_B^T \otimes W_B^T)$, where $D_1$ has $\tilde{O}(k)$ nonzero entries. Then we iteratively construct $B_2$, $B_2 \otimes B_3$ and $B_3$. Finally we have $C = A_1D_1$ (selecting $\tilde{O}(k)$ columns from A), $R = A_2D_2$ (selecting $\tilde{O}(k)$ rows from A), $T = A_3D_3$ (selecting $\tilde{O}(k)$ tubes from A) and tensor $U = ((B_1D_1)^\dagger) \otimes ((B_2D_2)^\dagger) \otimes ((B_3D_3)^\dagger)$.

We have similar results for entry-wise $\ell_p$, $1 \leq p < 2$, via analogous techniques.

**$\ell_1$-$\ell_2$-$\ell_2$ low-rank approximation (sum of Euclidean norms of faces).** For an $n \times n \times n$ tensor $A$, in the $\ell_1$-$\ell_2$-$\ell_2$ low rank approximation problem we seek a rank-$k$ tensor $B$ for which $\|A - B\|_v \leq \text{poly}(k, \log n) \text{ OPT}$, where $\text{OPT} = \inf_{\text{rank-}k \ B} \|A - B\|_v$ and where $\|A\|_v = \sum_i (\sum_{j,k} (A_{i,j,k})^2)^{\frac{1}{2}}$ for a tensor $A$. This norm is asymmetric, i.e., not invariant under permutations to its coordinates, and we cannot flatten the tensor along each of its dimensions while preserving its cost. Instead, we embed the problem to a new problem with a symmetric norm. Once we have a symmetric norm, we apply an iterative existential argument. We choose an oblivious sketching matrix (the $M$-Sketch in [CW15b]) $S \in \mathbb{R}^{s \times n}$ with $s = \text{poly}(k, \log n)$, and reduce the original problem to $\|S(A - B)\|_v$, by losing a small approximation factor. Because $s$ is small, we can then turn the $\ell_1$ part of the problem to the $\ell_2$ by losing another $\sqrt{s}$ in the approximation, so that now the problem is a Frobenius norm problem. We then apply our iterative existential argument to the problem $\|S((\sum_{i=1}^k U^*_i \otimes (A_2S_2X_2)_i \otimes (A_3S_3X_3)_i - A))\|_F$ where $U^*$ is a fixed matrix and $\bar{A} = SA$, and output a bicriteria solution.

**$\ell_1$-$\ell_1$-$\ell_2$ low-rank approximation (sum of Euclidean norms of tubes).** For an $n \times n \times n$ tensor $A$, in the $\ell_1$-$\ell_1$-$\ell_2$ low rank approximation problem we seek a rank-$k$ tensor $B$ for which $\|A - B\|_u \leq \text{poly}(k, \log n) \text{ OPT}$, where $\text{OPT} = \inf_{\text{rank-}k \ B} \|A - B\|_u$ and $\|A\|_u = \sum_{i,j,k} (\sum_k (A_{i,j,k})^2)^{\frac{1}{2}}$. The main difficulty in this problem is that the norm is asymmetric, and we cannot flatten the tensor along all
three dimensions. To reduce the problem to a problem with a symmetric norm, we choose random Gaussian matrices \( S \in \mathbb{R}^{n \times s} \) with \( s = O(n) \). By Dvoretzky’s theorem [Dvo61], for all tensors \( A \), \( \|AS\|_1 \approx \|A\|_u \), which reduces our problem to \( \min_{\text{rank-}k} B \|(A - B)S\|_1 \). Via an iterative existential argument, we obtain a generalized version of entrywise-\( k \) low rank approximation, \( \|(\tilde{A}_1S_1X_1) \otimes (A_2S_2X_2) \otimes (A_3S_3X_3) - A)S\|_1 \), where \( \tilde{A} = AS \) is an \( n \times n \times s \) size tensor. Finally, we can either use a polynomial system solver to obtain a rank-\( k \) solution, or output a bicriteria solution.

**Weighted low-rank approximation.** We also consider weighted low rank approximation. Given an \( n \times n \times n \) tensor \( A \) and an \( n \times n \times n \) tensor \( W \) of weights, we want to find a rank-\( k \) tensor \( B \) for which \( \|W \circ (A - B)\|_F \leq (1 + \epsilon) \text{OPT} \), where \( \text{OPT} = \min_{\text{rank-}k} B \|W \circ (A - B)\|_F \), and where for a tensor \( A \), \( \|W \circ A\|_F = (\sum_{i,j,k} W_{i,j,k} A_{i,j,k})^{\frac{1}{2}} \). We provide two algorithms based on different assumptions on the weight tensor \( W \). The first algorithm assumes that \( W \) has \( r \) distinct faces on each of its three dimensions. We flatten \( A \) and \( W \) along each of its three dimensions, obtaining \( A_1, A_2, A_3 \) and \( W_1, W_2, W_3 \). Because each \( W_i \) has \( r \) distinct rows, combining the “guess a sketch” technique from [RSW16] with our iterative argument, we can express matrices \( U_1, U_2, \) and \( U_3 \) in terms of \( O(rk^2/\epsilon) \) total indeterminates and for which a solution to the objective function \( \|W \circ (\sum_{i=1}^k (U_1)_{i} \otimes (U_2)_{i} \otimes (U_3)_{i} - A)\|_F \), together with \( O(r) \) side constraints, gives a \((1 + \epsilon)\)-approximation. We can solve the latter problem in \( \text{poly}(n) \cdot 2^{O(rk^2/\epsilon)} \) time. Our second algorithm assumes \( W \) has \( r \) distinct faces in two dimensions. Via a pigeonhole argument, the third dimension will have at most \( 2^{O(r)} \) distinct faces. We again use \( O(rk^2/\epsilon) \) variables to express \( U_1 \) and \( U_2 \), but now express \( U_3 \) in terms of these variables, which is necessary since \( W_3 \) could have an exponential number of distinct rows, ultimately causing too many variables needed to express \( U_3 \) directly. We again arrive at the objective function \( \|W \circ (\sum_{i=1}^k (U_1)_{i} \otimes (U_2)_{i} \otimes (U_3)_{i} - A)\|_F^2 \), but now have \( 2^{O(r)} \) side constraints, coming from the fact that \( U_3 \) is a rational function of the variables created for \( U_1 \) and \( U_2 \) and we need to clear denominators. Ultimately, the running time is \( 2^{O(rk^2/\epsilon)} \).

**Computational Hardness.** Our \( 2^{dk^{1-o(1)}} \) time hardness for \( c \)-approximation in Theorem H.42 is shown via a reduction from approximating MAX-3SAT to approximating MAX-E3SAT, where the latter problem has the property that each clause in the satisfiability instance has exactly 3 literals (in MAX-3SAT some clauses may have 2 literals). Then, a reduction [Tre01] from approximating MAX-E3SAT to approximating MAX-E3SAT(B) is performed, for a constant \( B \) which provides an upper bound on the number of clauses each literal can occur in. Given an instance \( \phi \) to MAX-E3SAT(B), we create a 3rd order tensor \( T \) as Håstad does using \( \phi \) [Hås90]. While Håstad’s reduction guarantees that the rank of \( T \) is at most \( r \) if \( \phi \) is satisfiable, and at least \( r + 1 \) otherwise, we can show that if \( \phi \) is not satisfiable then its rank is at least the minimal size of a set of variables which is guaranteed to intersect every unsatisfied clause in any unsatisfiable assignment. Since if \( \phi \) is not satisfiable, there are at least a linear fraction of clauses in \( \phi \) that are unsatisfied under any assignment by the inapproximability of MAX-E3SAT(B), and since each literal occurs in at most \( B \) clauses for a constant \( B \), it follows that the rank of \( T \) when \( \phi \) is not satisfiable is at least \( c_0 r \) for a constant \( c_0 > 1 \). Further, under ETH, our reduction implies one cannot approximate MAX-E3SAT(B), and thus approximate the rank of a tensor up to a factor \( c_0 \), in less than \( 2^{dk^{1-o(1)}} \) time. We need the near-linear size reduction of MAX-3SAT to MAX-E3SAT of [MR10] to get our strongest result.

The \( 2^{O(1/\epsilon^{1/4})} \) time hardness for \((1 + \epsilon)\)-approximation for rank-1 tensors in Theorem H.21 strengthens the NP-hardness for rank-1 tensor computation in Section 7 of [HL13], where instead of assuming the NP-hardness of the Clique problem, we assume ETH. Also, the proof in [HL13] did not explicitly bound the approximation error; we do this for a \( \text{poly}(1/\epsilon) \)-sized tensor (which can be
Algorithm 1 Main Meta-Algorithm

1: procedure TENSORLOWRANKAPPROXBICRITERIA($A, n, k, \epsilon$)  \& Theorem 1.1
2: \hspace{1em} Choose sketching matrices $S_2, S_3$ (Composition of Gaussian and CountSketch.)
3: \hspace{1em} Choose sketching matrices $T_2, T_3$ (CountSketch.)
4: \hspace{1em} Compute $T_2 A S_2, T_3 A S_3$.
5: \hspace{1em} Construct $\tilde{V}$ by setting $(i, j)$-th column to be $(A S_2)_i$.
6: \hspace{1em} Construct $\tilde{W}$ by setting $(i, j)$-th column to be $(A S_3)_j$.
7: \hspace{1em} Construct matrix $B$ by setting $(i, j)$-th row of $B$ is vectorization of $(T_2 A S_2)_i \otimes (T_3 A S_3)_j$.
8: \hspace{1em} Solve $\min_U \|UB - (A(I, T_2, T_3))_1\|_F^2$.
9: \hspace{1em} return $\tilde{U}, \tilde{V},$ and $\tilde{W}$.
10: end procedure

11: procedure TENSORLOWRANKAPPROX($A, n, k, \epsilon$)  \& Theorem 1.2
12: \hspace{1em} Choose sketching matrices $S_1, S_2, S_3$ (Composition of Gaussian and CountSketch.)
13: \hspace{1em} Choose sketching matrices $T_1, T_2, T_3$ (CountSketch.)
14: \hspace{1em} Compute $T_1 A S_1, T_2 A S_2, T_3 A S_3$.
15: \hspace{1em} Solve $\min_{X_1, X_2, X_3} \| (T_1 A S_1 X_1) \otimes (T_2 A S_2 X_2) \otimes (T_3 A S_3 X_3) - A(T_1, T_2, T_3) \|_F^2$.
16: \hspace{1em} return $A_1 S_1 X_1, A_2 S_2 X_2, A_3 S_3 X_3$.
17: end procedure

padded with 0s to a poly($n$)-sized tensor to rule out $(1 + \epsilon)$-approximation in $2^{\Omega(1/\epsilon^{1/4})}$ time.

The same hard instance above shows, assuming ETH, that $2^{\Omega(1/\epsilon^{1/2})}$ time is necessary for $(1 + \epsilon)$-approximation to the spectral norm of a symmetric rank-1 tensor (see Section H.2 and Section H.3).

Assuming ETH, the $2^{1/\epsilon^{1-o(1)}}$-hardness [SWZ17] for matrix $\ell_1$-low rank approximation gives the same hardness for tensor entry-wise $\ell_1$ and $\ell_1-\ell_2$ low rank approximation. Also, under ETH, we strengthen the NP-hardness in [CW15a] to a $2^{1/\epsilon^{o(1)}}$-hardness for $\ell_1-\ell_2$ low rank approximation of a matrix, which gives the same hardness for tensor $\ell_1-\ell_2$ low rank approximation.

**Hard Instance.** We extend the previous matrix CUR hard instance [BW14] to 3rd order tensors by planting multiple rotations of the hard instance for matrices into a tensor. We show C must select $\Omega(k/\epsilon)$ columns from $A$, $R$ must select $\Omega(k/\epsilon)$ rows from $A$, and $T$ must select $\Omega(k/\epsilon)$ tubes from $A$. Also the tensor $U$ must have rank at least $k$. This generalizes to $q$-th order tensors.

**Optimal matrix CUR decomposition.** We also improve the $\nnz(A) \log n + (n+d) \log(n,k,1/\epsilon)$ running time of [BW14] for CUR decomposition of $A \in \mathbb{R}^{n \times d}$ to $\nnz(A) + (n+d) \log(k,1/\epsilon)$, while selecting the optimal number of columns, rows, and a rank-$k$ matrix $U$. Using [CW13, MM13, NN13], we find a matrix $\tilde{U}$ with $k$ orthonormal columns in $\nnz(A) + n \log(k,\epsilon)$ time for which $\min_{V} \|\tilde{U}V - A\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$. Let $s_1 = \tilde{O}(k/\epsilon^2)$ and $S_1 \in \mathbb{R}^{s_1 \times n}$ be a sampling/rescaling matrix by the leverage scores of $\tilde{U}$. By strengthening the affine embedding analysis of [CW13] to leverage score sampling (the analysis of [CW13] gives a weaker analysis for affine embeddings using leverage scores which does not allow approximation in the sketch space to translate to approximation in the original space), with probability at least 0.99, for all $X'$ which satisfy $\|S_1 \tilde{U}X' - S_1 A\|_F^2 \leq (1 + \epsilon')\|\tilde{U}X' - A\|_F^2$, we have $\|\tilde{U}X' - A\|_F^2 \leq (1 + \epsilon)\min_{X} \|\tilde{U}X - S_1 A\|_F^2$, where $\epsilon' = 0.001\epsilon$. Applying our generalized row subset selection procedure, we can find $Y, R$ for which $\|S_1 \tilde{U}YR - S_1 A\|_F^2 \leq (1 + \epsilon')\min_{X} \|\tilde{U}X - S_1 A\|_F^2$, where $R$ contains $O(k/\epsilon') = O(k/\epsilon)$ rescaled rows of $S_1 A$. A key point is that rescaled rows of $S_1 A$ are also rescaled rows of $A$. Then, $\|\tilde{U}YR - A\|_F^2 \leq (1 + \epsilon)\min_{X} \|\tilde{U}X - A\|_F^2$. Finding $Y, R$ can be done in $d \log(s_1/\epsilon) = d \log(k/\epsilon)$ time. Now set
\( \hat{V} = YR \). We can choose \( S_2 \) to be a sampling/rescaling matrix, and then find \( C, Z \) for which 
\[
\| C \hat{V} S_2 - A S_2 \|_F^2 \leq (1 + \epsilon') \min \| X \hat{V} S_2 - A S_2 \|_F^2
\]
in a similar way, where \( C \) contains \( O(k/\epsilon) \) rescaled columns of \( A S_2 \), and thus also of \( A \). We thus have 
\[
\| C \hat{V} R - A \|_F^2 \leq (1 + O(\epsilon)) \| A - A_k \|_F^2.
\]

**Distributed and streaming settings.** Since our algorithms use linear sketches, they are implementable in distributed and streaming models. We use random variables with limited independence to succinctly store the sketching matrices [CW13, KVW14, KN14, Woo14, SWZ17].

**Extension to other notions of tensor rank.** This paper focuses on the standard CP rank, or canonical rank, of a tensor. As mentioned, due to border rank issues, the best rank-\( k \) solution does not exist in certain cases. There are other notions of tensor rank considered in some applications which do not suffer from this problem, e.g., the Tucker rank [KC07, PC08, MH09, ZW13, YC14], and the train rank [Ose11, OTZ11, ZWZ16, PTBD16]). We also show observe that our techniques can be applied to these notions of rank.

**1.4 Comparison to [BCV14]**

In [BCV14], the authors show for a third order \( n_1 \times n_2 \times n_3 \) tensor \( A \) how to find a rank-\( k \) tensor \( B \) for which 
\[
\| A - B \|_F^2 \leq 5 \text{OPT} \text{ in } \text{poly}(n_1 n_2 n_3) \exp(\text{poly}(k)) \text{ time.}
\]

They generalize this to \( q \)-th order tensors to find a rank-\( k \) tensor \( B \) for which 
\[
\| A - B \|_F^2 = O(q) \text{OPT} \text{ in } \text{poly}(n_1 n_2 \cdots n_q) \exp(\text{poly}(qk)) \text{ time.}
\]

In contrast, we obtain a rank-\( k \) tensor \( B \) for which 
\[
\| A - B \|_F^2 \leq (1 + \epsilon) \text{OPT} \text{ in } \text{nnz}(A) + n \cdot \text{poly}(k/\epsilon) + \exp((k^2/\epsilon) \text{poly}(q)) \text{ time for every order } q.
\]

Thus, we obtain a \((1+\epsilon)\) instead of an \(O(q)\) approximation. The \(O(q)\) approximation in [BCV14] seems inherent since the authors apply triangle inequality \( q \) times, each time losing a constant factor. This seems necessary since their argument is based on the span of the top \( k \) principal components in the SVD in each flattening separately containing a good space to project onto for a given mode. In contrast, our iterative existential argument chooses the space to project onto in successive modes adaptively as a function of spaces chosen for previous modes, and thus we obtain a \((1+\epsilon)\) instead of an \(O(q)\)-approximation, which becomes a \((1+\epsilon)\)-approximation after replacing \( \epsilon \) with \( \epsilon/q \). Also, importantly, our algorithm runs in \( \text{nnz}(A) + n \cdot \text{poly}(k/\epsilon) + \exp((k^2/\epsilon) \text{poly}(q)) \) time and there are multiple hurdles we overcome to achieve this, as described in Section 1.2 above.

**1.5 An Algorithm and a Roadmap**

**Roadmap** Section A introduces notation and definitions. Section B includes several useful tools. We provide our Frobenius norm low rank approximation algorithms in Section C. Section C.10 extends our results to general \( q \)-th order tensors. Section D has our results for entry-wise \( \ell_1 \) norm low rank approximation. Section E has our results for entry-wise \( \ell_p \) norm low rank approximation. Section G has our results for weighted low rank approximation. Section F has our results for asymmetric norm low rank approximation algorithms. We present our hardness results in Section H and Section I. Section J and Section K extend the results to distributed and streaming settings. Section L extends our techniques from tensor rank to other notions of tensor rank including tensor Tucker rank and tensor train rank.
A Notation

For an \( n \in \mathbb{N}^+ \), let \([n]\) denote the set \( \{1, 2, \ldots, n\}\).

For any function \( f \), we define \( \tilde{O}(f) \) to be \( f \cdot \log^{O(1)}(f) \). In addition to \( O(\cdot) \) notation, for two functions \( f, g \), we use the shorthand \( f \lesssim g \) (resp. \( \gtrsim \)) to indicate that \( f \leq Cg \) (resp. \( \geq \)) for an absolute constant \( C \). We use \( f \approx g \) to mean \( cf \leq g \leq Cf \) for constants \( c, C \).

For a matrix \( A \), we use \( \|A\|_2 \) to denote the spectral norm of \( A \). For a tensor \( A \), let \( \|A\| \) and \( \|A\|_2 \) (which we sometimes use interchangeably) denote the spectral norm of tensor \( A \),

\[
\|A\| = \sup_{x,y,z \neq 0} \frac{|A(x,y,z)|}{\|x\| \cdot \|y\| \cdot \|z\|}.
\]

Let \( \|A\|_F \) denote the Frobenius norm of a matrix/tensor \( A \), i.e., \( \|A\|_F \) is the square root of sum of squares of all the entries of \( A \). For \( 1 \leq p < 2 \), we use \( \|A\|_p \) to denote the entry-wise \( \ell_p \)-norm of a matrix/tensor \( A \), i.e., \( \|A\|_p \) is the \( p \)-th root of the sum of \( p \)-th powers of the absolute values of the entries of \( A \). \( \|A\|_1 \) will be an important special case of \( \|A\|_p \), which corresponds to the sum of absolute values of all of the entries.

Let \( \text{nnz}(A) \) denote the number of nonzero entries of \( A \). Let \( \det(A) \) denote the determinant of a square matrix \( A \). Let \( A^\top \) denote the transpose of \( A \). Let \( A^{-1} \) denote the inverse of a full rank square matrix.

For a 3rd order tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), its \( x \)-mode fibers are called column fibers \((x = 1)\), row fibers \((x = 2)\) and tube fibers \((x = 3)\). For tensor \( A \), we use \( A_{*,j,l} \) to denote its \((j,l)\)-th column, we use \( A_{i,*,l} \) to denote its \((i,l)\)-th row, and we use \( A_{i,j,*} \) to denote its \((i,j)\)-th tube.

A tensor \( A \) is symmetric if and only if for any \( i, j, k \), \( A_{i,j,k} = A_{i,k,j} = A_{j,i,k} = A_{j,k,i} = A_{k,i,j} = A_{k,j,i} \).

For a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), we use \( \top \) to denote rotation (3 dimensional transpose) so that \( A^\top \in \mathbb{R}^{n_3 \times n_1 \times n_2} \). For a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and matrix \( B \in \mathbb{R}^{n_3 \times k} \), we define the tensor-matrix dot product to be \( A \cdot B \in \mathbb{R}^{n_1 \times n_2 \times k} \).
We use \( \otimes \) to denote outer product, \( \circ \) to denote entrywise product, and \( \cdot \) to denote dot product. Given two column vectors \( u, v \in \mathbb{R}^n \), let \( u \otimes v \in \mathbb{R}^{n \times n} \) and \( (u \otimes v)_{i,j} = u_i \cdot v_j \), \( u^\top v = \sum_{i=1}^{n} u_i v_i \in \mathbb{R} \) and \( (u \circ v)_i = u_i v_i \).

**Definition A.1** (\( \otimes \) product for vectors). Given \( q \) vectors \( u_1 \in \mathbb{R}^{n_1}, u_2 \in \mathbb{R}^{n_2}, \ldots, u_q \in \mathbb{R}^{n_q} \), we use \( u_1 \otimes u_2 \otimes \cdots \otimes u_q \) to denote an \( n_1 \times n_2 \times \cdots \times n_q \) tensor such that, for each \( (j_1,j_2,\ldots,j_q) \in [n_1] \times [n_2] \times \cdots \times [n_q] \),

\[
(u_1 \otimes u_2 \otimes \cdots \otimes u_q)_{j_1,j_2,\ldots,j_q} = (u_1)_{j_1}(u_2)_{j_2} \cdots (u_q)_{j_q},
\]

where \( (u_i)_{j_i} \) denotes the \( j_i \)-th entry of vector \( u_i \).

**Definition A.2** (vec(), convert tensor into a vector). Given a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_q} \), let \( \text{vec}(A) \in \mathbb{R}^{1 \times \prod_{l=1}^q n_l} \) be a row vector, such that the \( t \)-th entry of \( \text{vec}(A) \) is \( A_{j_1,j_2,\ldots,j_q} \) where \( t = (j_1 - 1) \prod_{l=2}^q n_l + (j_2 - 1) \prod_{l=3}^q n_l + \cdots + (j_{q-1} - 1) n_q + j_q \).

For example if \( u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, v = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \) then \( \text{vec}(u \otimes v) = \begin{bmatrix} 3 & 4 & 5 & 6 & 8 & 10 \end{bmatrix} \).

**Definition A.3** (\( \otimes \) product for matrices). Given \( q \) matrices \( U_1 \in \mathbb{R}^{n_1 \times k}, U_2 \in \mathbb{R}^{n_2 \times k}, \ldots, U_q \in \mathbb{R}^{n_q \times k} \), we use \( U_1 \otimes U_2 \otimes \cdots \otimes U_q \) to denote an \( n_1 \times n_2 \times \cdots \times n_q \) tensor which can be written as,

\[
U_1 \otimes U_2 \otimes \cdots \otimes U_q = \sum_{i=1}^{k} (U_1)_{i} \otimes (U_2)_{i} \otimes \cdots \otimes (U_q)_{i} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_q},
\]

where \( (U_j)_i \) denotes the \( i \)-th column of matrix \( U_j \in \mathbb{R}^{n_j \times k} \).

**Definition A.4** (\( \otimes \) product for matrices). Given \( q \) matrices \( U_1 \in \mathbb{R}^{k \times n_1}, U_2 \in \mathbb{R}^{k \times n_2}, \ldots, U_q \in \mathbb{R}^{k \times n_q} \), we use \( U_1 \odot U_2 \odot \cdots \odot U_q \) to denote a \( k \times \prod_{j=1}^q n_j \) matrix where the \( i \)-th row of \( U_1 \odot U_2 \odot \cdots \odot U_q \) is the vectorization of \( (U_1)^i \odot (U_2)^i \odot \cdots \odot (U_q)^i \), i.e.,

\[
U_1 \odot U_2 \odot \cdots \odot U_q = \begin{bmatrix}
\text{vec}((U_1)^1 \odot (U_2)^1 \odot \cdots \odot (U_q)^1) \\
\text{vec}((U_1)^2 \odot (U_2)^2 \odot \cdots \odot (U_q)^2) \\
\vdots \\
\text{vec}((U_1)^k \odot (U_2)^k \odot \cdots \odot (U_q)^k)
\end{bmatrix} \in \mathbb{R}^{k \times \prod_{j=1}^q n_j}.
\]

where \( (U_j)^i \in \mathbb{R}^{n_j} \) denotes the \( i \)-th row of matrix \( U_j \in \mathbb{R}^{k \times n_j} \).
Definition A.5 (Flattening vs unflattening/retensorizing). Suppose we are given three matrices \( U \in \mathbb{R}^{n_1 \times k}, V \in \mathbb{R}^{n_2 \times k}, W \in \mathbb{R}^{n_3 \times k} \). Let tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) denote \( U \otimes V \otimes W \). Let \( A_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) denote a matrix obtained by flattening tensor \( A \) along the 1st dimension. Then \( A_1 = U \cdot B \), where \( B = V^\top \otimes W^\top \in \mathbb{R}^{k \times n_1 n_2 n_3} \) denotes the matrix for which the \( i \)-th row is \( \text{vec}(V_i \otimes W_i), \forall i \in [k] \). We let the “flattening” be the operation that obtains \( A_1 \) by \( A \). Given \( A_1 = U \cdot B \), we can obtain tensor \( A \) by unflattening/retensorizing \( A_1 \). We let “retensorization” be the operation that obtains \( A \) from \( A_1 \). Similarly, let \( A_2 \in \mathbb{R}^{n_2 \times n_1 n_3} \) denote a matrix obtained by flattening tensor \( A \) along the 2nd dimension, so \( A_2 = V \cdot C \), where \( C = W^\top \otimes U^\top \in \mathbb{R}^{k \times n_1 n_2} \) denotes the matrix for which the \( i \)-th row is \( \text{vec}(W_i \otimes U_i), \forall i \in [k] \). Let \( A_3 \in \mathbb{R}^{n_3 \times n_1 n_2} \) denote a matrix obtained by flattening tensor \( A \) along the 3rd dimension. Then, \( A_3 = W \cdot D \), where \( D = U^\top \otimes V^\top \in \mathbb{R}^{k \times n_1 n_2} \) denotes the matrix for which the \( i \)-th row is \( \text{vec}(U_i \otimes V_i), \forall i \in [k] \).

Definition A.6 (\( \cdot,\cdot,\cdot \) operator for tensors and matrices). Given tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and three matrices \( B_1 \in \mathbb{R}^{n_1 \times d_1}, B_2 \in \mathbb{R}^{n_2 \times d_2}, B_3 \in \mathbb{R}^{n_3 \times d_3} \), we define tensors \( A(B_1,I,I) \in \mathbb{R}^{d_1 \times n_2 \times n_3}, A(I,B_2,I) \in \mathbb{R}^{n_1 \times d_2 \times n_3}, A(I,I,B_3) \in \mathbb{R}^{n_1 \times n_2 \times d_3}, A(B_1,B_2,I) \in \mathbb{R}^{d_1 \times d_2 \times n_3}, A(B_1,B_2,B_3) \in \mathbb{R}^{d_1 \times d_2 \times d_3} \) as follows,

\[
A(B_1,I,I)_{i,j,l} = \sum_{i',1}^{n_1} A_{i',j}(B_1)_{i',i}, \quad \forall (i,j,l) \in [d_1] \times [n_2] \times [n_3]
\]

\[
A(I,B_2,I)_{i,j,l} = \sum_{j',1}^{n_2} A_{i,j'}(B_2)_{j',j}, \quad \forall (i,j,l) \in [n_1] \times [d_2] \times [n_3]
\]

\[
A(I,I,B_3)_{i,j,l} = \sum_{l',1}^{n_3} A_{i,j,l'}(B_3)_{l',l}, \quad \forall (i,j,l) \in [n_1] \times [n_2] \times [d_3]
\]

\[
A(B_1,B_2,I)_{i,j,l} = \sum_{i',1}^{n_1} \sum_{j',1}^{n_2} A_{i',j',i}(B_1)_{i',i}(B_2)_{j',j}, \quad \forall (i,j,l) \in [d_1] \times [d_2] \times [n_3]
\]

\[
A(B_1,B_2,B_3)_{i,j,l} = \sum_{i',1}^{n_1} \sum_{j',1}^{n_2} \sum_{l',1}^{n_3} A_{i',j',l'}(B_1)_{i',i}(B_2)_{j',j}(B_3)_{l',l}, \quad \forall (i,j,l) \in [d_1] \times [d_2] \times [d_3]
\]

Note that \( B_1^\top A = A(B_1,I,I), AB_3 = A(I,I,B_3) \) and \( B_1^\top AB_3 = A(B_1,I,B_3) \). In our paper, if \( \forall i \in [3], B_i \) is either a rectangular matrix or a symmetric matrix, then we sometimes use \( A(B_1,B_2,B_3) \) to denote \( A(B_1^\top,B_2^\top,B_3^\top) \) for simplicity. Similar to the \( \langle \cdot,\cdot,\cdot \rangle \) operator on 3rd order tensors, we can define the \( \langle \cdot,\cdot,\cdot \rangle \) operator on higher order tensors. For the matrix case, \( \min_{\text{rank} \rightarrow k} \| A - A' \|^2_F \) always exists. However, this is not true for tensors [DSL08]. For convenience, we redefine the notation of OPT and min.

Definition A.7. Given tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3}, k > 0 \), if \( \min_{\text{rank} \rightarrow k} \| A - A' \|^2_F \) does not exist, then we define \( \text{OPT} = \inf_{\text{rank} \rightarrow k} \| A - A' \|^2_F + \gamma \) for sufficiently small \( \gamma > 0 \), which can be an arbitrarily small positive function of \( n \). We let \( \min_{\text{rank} \rightarrow k} \| A - A' \|^2_F \) be the value of OPT, and we let \( \arg \min_{\text{rank} \rightarrow k} \| A - A' \|^2_F \) be a rank \( -k \) tensor \( A_k \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) which satisfies \( \| A - A_k \|^2_F = \text{OPT} \).

B Preliminaries

Section B.1 provides the definitions for Subspace Embeddings and Approximate Matrix Product. We introduce the definition for Tensor-CURT decomposition in Section B.2. Section B.3 presents...
Figure 3: A 3rd order tensor contains $n^2$ columns, $n^2$ rows, and $n^2$ tubes.

a tool which we call a “polynomial system verifier”. Section B.4 introduces a tool which is able to
determine the minimum nonzero value of the absolute value of a polynomial evaluated on a set,
provided the polynomial is never equal to 0 on that set. Section B.5 shows how to relax an $\ell_p$ problem
to an $\ell_2$ problem. We provide definitions for CountSketch and Gaussian transforms in Section B.6.
We present Cauchy and $p$-stable transforms in Section B.7. We introduce leverage scores and Lewis
weights in Section B.8 and Section B.9. Finally, we explain an extension of CountSketch, which is
called TensorSketch in Section B.10.

### B.1 Subspace Embeddings and Approximate Matrix Product

**Definition B.1** (Subspace Embedding). A $(1 \pm \epsilon)$ $\ell_2$-subspace embedding for the column space of
an $n \times d$ matrix $A$ is a matrix $S$ for which for all $x \in \mathbb{R}^d$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$.

**Definition B.2** (Approximate Matrix Product). Let $0 < \epsilon < 1$ be a given approximation parameter.
Given matrices $A$ and $B$, where $A$ and $B$ each have $n$ rows, the goal is to output a matrix $C$ so
that $\|A^TB - C\|_F \leq \epsilon\|A\|_F\|B\|_F$. Typically $C$ has the form $A^TS^SB$, for a random matrix $S$ with
a small number of rows. See, e.g., Lemma 32 of [CW13] for a number of example matrices $S$ with
$O(\epsilon^{-2})$ rows for which this property holds.

### B.2 Tensor CURT decomposition

We first review matrix CUR decompositions:

**Definition B.3** (Matrix CUR, exact). Given a matrix $A \in \mathbb{R}^{n \times d}$, we choose $C \in \mathbb{R}^{n \times c}$ to be a
subset of columns of $A$ and $R \in \mathbb{R}^{r \times n}$ to be a subset of rows of $A$. If there exists a matrix $U \in \mathbb{R}^{c \times r}$
such that $A$ can be written as,

$$CUR = A,$$

then we say $C, U, R$ is matrix $A$’s CUR decomposition.
Figure 4: A third order tensor has three types of faces: the column-row faces, the column-tube faces, and the row-tube faces.
Definition B.4 (Matrix CUR, approximate). Given a matrix $A \in \mathbb{R}^{n \times d}$, a parameter $k \geq 1$, an approximation ratio $\alpha > 1$, and a norm $\| \cdot \|_\xi$, we choose $C \in \mathbb{R}^{n \times c}$ to be a subset of columns of $A$ and $R \in \mathbb{R}^{r \times n}$ to be a subset of rows of $A$. Then if there exists a matrix $U \in \mathbb{R}^{c \times r}$ such that
\[
\|CUR - A\|_\xi \leq \alpha \min_{\text{rank}-k} \|A_k - A\|_\xi,
\]
where $\| \cdot \|_\xi$ can be operator norm, Frobenius norm or Entry-wise $\ell_1$ norm, we say that $C, U, R$ is matrix $A$’s approximate CUR decomposition, and sometimes just refer to this as a CUR decomposition.

Definition B.5 ([Bou11]). Given matrix $A \in \mathbb{R}^{m \times n}$, integer $k$, and matrix $C \in \mathbb{R}^{m \times r}$ with $r > k$, we define the matrix $\Pi^k_{C,k}(A) \in \mathbb{R}^{m \times n}$ to be the best approximation to $A$ (under the $\xi$-norm) within the column space of $C$ of rank at most $k$; so, $\Pi^k_{C,k}(A) \in \mathbb{R}^{m \times n}$ minimizes the residual $\|A - \hat{A}\|_\xi$, over all $\hat{A} \in \mathbb{R}^{m \times n}$ in the column space of $C$ of rank at most $k$.

We define the following notion of tensor-CURT decomposition.

Definition B.6 (Tensor CURT, exact). Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we choose three sets of pair of coordinates $S_1 \subseteq [n_2] \times [n_3], S_2 \subseteq [n_1] \times [n_3], S_3 \subseteq [n_1] \times [n_2]$. We define $c = |S_1|$, $r = |S_2|$ and $t = |S_3|$. Let $C \in \mathbb{R}^{n_1 \times c}$ denote a subset of columns of $A$, $R \in \mathbb{R}^{n_2 \times r}$ denote a subset of rows of $A$, and $T \in \mathbb{R}^{n_3 \times t}$ denote a subset of tubes of $A$. If there exists a tensor $U \in \mathbb{R}^{c \times r \times t}$ such that $A$ can be written as
\[
(((U \cdot T^\top) \cdot R^\top) \cdot C^\top)^\top = A,
\]
or equivalently,
\[
U(C, R, T) = A,
\]
or equivalently,
\[
\forall (i, j, l) \in [n_1] \times [n_2] \times [n_3], A_{i,j,l} = \sum_{u_1=1}^{c} \sum_{u_2=1}^{r} \sum_{u_3=1}^{t} U_{u_1,u_2,u_3} C_{i,u_1} R_{j,u_2} T_{l,u_3},
\]
then we say $C, U, R, T$ is tensor $A$’s CURT decomposition.

Definition B.7 (Tensor CURT, approximate). Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, for some $k \geq 1$, for some approximation $\alpha > 1$, for some norm $\| \cdot \|_\xi$, we choose three sets of pair of coordinates $S_1 \subseteq [n_2] \times [n_3], S_2 \subseteq [n_1] \times [n_3], S_3 \subseteq [n_1] \times [n_2]$. We define $c = |S_1|$, $r = |S_2|$ and $t = |S_3|$. Let $C \in \mathbb{R}^{n_1 \times c}$ denote a subset of columns of $A$, $R \in \mathbb{R}^{n_2 \times r}$ denote a subset of rows of $A$, and $T \in \mathbb{R}^{n_3 \times t}$ denote a subset of tubes of $A$. If there exists a tensor $U \in \mathbb{R}^{c \times r \times t}$ such that
\[
\|U(C, R, T) - A\|_\xi \leq \alpha \min_{\text{rank}-k} \|A_k - A\|_\xi,
\]
where $\| \cdot \|_\xi$ is operator norm, Frobenius norm or Entry-wise $\ell_1$ norm, then we refer to $C, U, R, T$ as an approximate CUR decomposition of $A$, and sometimes just refer to this as a CUR decomposition of $A$.

Recently, [TM17] studied a very different face-based tensor-CURT decomposition, which selects faces from tensors rather than columns. To achieve their results, [TM17] need to make several incoherence assumptions on the original tensor. Their sample complexity depends on $\log n$, and they only sample two of the three dimensions. We will provide more general face-based tensor CURT decompositions.
Figure 5: Column subset selection, row subset selection and tube subset selection.

Definition B.8 (Tensor (face-based) CURT, exact). Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we choose three sets of coordinates $S_1 \subseteq [n_1], S_2 \subseteq [n_2], S_3 \subseteq [n_3]$. We define $c = |S_1|, r = |S_2|$ and $t = |S_3|$. Let $C \in \mathbb{R}^{c \times n_2 \times n_3}$ denote a subset of row-tube faces of $A$, $R \in \mathbb{R}^{n_1 \times r \times n_3}$ denote a subset of column-tube faces of $A$, and $T \in \mathbb{R}^{n_1 \times n_2 \times t}$ denote a subset of column-row faces of $A$. Let $C_2 \in \mathbb{R}^{n_2 \times cn_3}$
denote the matrix obtained by flattening the tensor $C$ along the second dimension. Let $R_3 \in \mathbb{R}^{n_3 \times r n_1}$ denote the matrix obtained by flattening the tensor $R$ along the third dimension. Let $T_1 \in \mathbb{R}^{n_1 \times t n_2}$ denote the matrix obtained by flattening the tensor $T$ along the first dimension. If there exists a tensor $U \in \mathbb{R}^{t n_2 \times c n_3 \times r n_1}$ such that $A$ can be written as

$$
\sum_{i=1}^{t n_2} \sum_{j=1}^{c n_3} \sum_{l=1}^{r n_1} U_{i,j,l}(T_1)_{i} \otimes (C_2)_{j} \otimes (R_3)_{l} = A,
$$

or equivalently,

$$
U(T_1, C_2, R_3) = A,
$$

then we say $C, U, R, T$ is tensor $A$’s (face-based) CURT decomposition.

**Definition B.9** (Tensor (face-based) CURT, approximate). Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, for some $k \geq 1$, for some approximation $\alpha > 1$, for some norm $\|\cdot\|_\xi$, we choose three sets of coordinates $S_1 \subseteq [n_1], S_2 \subseteq [n_2], S_3 \subseteq [n_3]$. We define $c = |S_1|$, $r = |S_2|$ and $t = |S_3|$. Let $C \in \mathbb{R}^{c \times n_2 \times n_3}$ denote a subset of row-tube faces of $A$, $R \in \mathbb{R}^{n_1 \times r \times n_3}$ denote a subset of column-tube faces of $A$, and $T \in \mathbb{R}^{n_1 \times n_2 \times t}$ denote a subset of column-row faces of $A$. Let $C_2 \in \mathbb{R}^{r n_2 \times c n_3}$ denote the matrix obtained by flattening the tensor $C$ along the second dimension. Let $R_3 \in \mathbb{R}^{n_3 \times r n_1}$ denote the matrix obtained by flattening the tensor $R$ along the third dimension. Let $T_1 \in \mathbb{R}^{n_1 \times t n_2}$ denote the matrix obtained by flattening the tensor $T$ along the first dimension. If there exists a tensor $U \in \mathbb{R}^{t n_2 \times c n_3 \times r n_1}$ such that

$$
\|U(T_1, C_2, R_3) - A\|_\xi \leq \alpha \min_{\text{rank}-k A_k} \|A_k - A\|_\xi,
$$

where $\|\cdot\|_\xi$ is operator norm, Frobenius norm or Entry-wise $\ell_1$ norm, then we refer to $C, U, R, T$ as an approximate CUR decomposition of $A$, and sometimes just refer to this as a (face-based) CURT decomposition of $A$.

**B.3 Polynomial system verifier**

We use the polynomial system verifiers independently developed by Renegar [Ren92a, Ren92b] and Basu et al. [BPR96].

**Theorem B.10** (Decision Problem [Ren92a, Ren92b, BPR96]). Given a real polynomial system $P(x_1, x_2, \ldots, x_v)$ having $v$ variables and $m$ polynomial constraints $f_i(x_1, x_2, \ldots, x_v) \Delta_0, \forall i \in [m]$, where $\Delta_i$ is any of the “standard relations”: $\{>, \geq, =, \neq, \leq, <\}$, let $d$ denote the maximum degree of all the polynomial constraints and let $H$ denote the maximum bitsize of the coefficients of all the polynomial constraints. Then in

$$(md)^{O(v) \text{poly}(H)},$$

time one can determine if there exists a solution to the polynomial system $P$.

Recently, this technique has been used to solve a number of low-rank approximation and matrix factorization problems [AGKM12, Moi13, CW15a, BDL16, RSW16, SWZ17].
Figure 6: An example tensor CURT decomposition.

B.4 Lower bound on the cost of a polynomial system

An important result we use is the following lower bound on the minimum value attained by a polynomial restricted to a compact connected component of a basic closed semi-algebraic subset of $\mathbb{R}^v$.

**Theorem B.11** ([JPT13]). Let $T = \{ x \in \mathbb{R}^v | f_1(x) \geq 0, \ldots, f_\ell(x) \geq 0, f_{\ell+1}(x) = 0, \ldots, f_m(x) = 0 \}$ be defined by polynomials $f_1, \ldots, f_m \in \mathbb{Z}[x_1, \ldots, x_v]$ with $n \geq 2$, degrees bounded by an even integer $d$, and coefficients of absolute value at most $H$, and let $C$ be a compact connected (in the topological sense) component of $T$. Let $g \in \mathbb{Z}[x_1, \ldots, x_v]$ be a polynomial of degree at most $d$ and coefficients of absolute value bounded by $H$. Then, the minimum value that $g$ takes over $C$ satisfies that if it is not zero, then its absolute value is greater than or equal to

$$2^{\ell-v/2} \tilde{H} d^v - v^2 d^v,$$

where $\tilde{H} = \max\{H, 2v + 2m\}$.

While the above theorem involves notions from topology, we shall apply it in an elementary way. Namely, in our setting $T$ will be bounded and so every connected component, which is by definition closed, will also be bounded and therefore compact. As the connected components partition $T$ the theorem will just be applied to give a global minimum value of $g$ on $T$ provided that it is non-zero.

B.5 Frobenius norm and $\ell_2$ relaxation

**Theorem B.12** (Generalized rank-constrained matrix approximations, Theorem 2 in [FT07]). Given matrices $A \in \mathbb{R}^{n \times d}$, $B \in \mathbb{R}^{n \times p}$, and $C \in \mathbb{R}^{q \times d}$, let the SVD of $B$ be $B = UB \Sigma_B V_B^T$ and the SVD of $C$ be $C = UC \Sigma_C V_C^T$. Then,

$$B^T(U_B U_B^T AV_C V_C^T)_k C^T = \arg \min_{\text{rank } k \ X \in \mathbb{R}^{p \times q}} \| A - BXC \|_F,$$

where $(U_B U_B^T AV_C V_C^T)_k \in \mathbb{R}^{p \times q}$ is of rank at most $k$ and denotes the best rank-$k$ approximation to $U_B U_B^T AV_C V_C^T \in \mathbb{R}^{p \times d}$ in Frobenius norm.
Claim B.13 (ℓ2 relaxation of ℓp-regression). Let \( p \in [1, 2) \). For any matrix \( A \in \mathbb{R}^{n \times d} \) and \( b \in \mathbb{R}^n \), define \( x^* = \arg \min_{x \in \mathbb{R}^d} \|Ax - b\|_p \) and \( x' = \arg \min_{x \in \mathbb{R}^d} \|Ax - b\|_2 \). Then,
\[
\|Ax^* - b\|_p \leq \|Ax' - b\|_p \leq n^{1/p - 1/2} \cdot \|Ax^* - b\|_p.
\]

Claim B.14 ((Matrix) Frobenius norm relaxation of ℓp-low rank approximation). Let \( p \in [1, 2) \) and for any matrix \( A \in \mathbb{R}^{n \times d} \), define \( A^* = \arg \min_{\text{rank} - k B \in \mathbb{R}^{n \times d}} \|B - A\|_p \) and \( A' = \arg \min_{\text{rank} - k B \in \mathbb{R}^{n \times d}} \|B - A\|_F \). Then
\[
\|A^* - A\|_p \leq \|A' - A\|_p \leq (nd)^{1/p - 1/2} \|A^* - A\|_p.
\]

Claim B.15 ((Tensor) Frobenius norm relaxation of ℓp-low rank approximation). Let \( p \in [1, 2) \) and for any matrix \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), define
\[
A^* = \arg \min_{\text{rank} - k B \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \|B - A\|_p
\]
and
\[
A' = \arg \min_{\text{rank} - k B \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \|B - A\|_F.
\]
Then
\[
\|A^* - A\|_p \leq \|A' - A\|_p \leq (n_1 n_2 n_3)^{1/p - 1/2} \|A^* - A\|_p.
\]

B.6 CountSketch and Gaussian transforms

Definition B.16 (Sparse embedding matrix or CountSketch transform). A CountSketch transform is defined to be \( \Pi = \sigma \cdot \Phi D \in \mathbb{R}^{m \times n} \). Here, \( \sigma \) is a scalar, \( D \) is an \( n \times n \) random diagonal matrix with each diagonal entry independently chosen to be +1 or −1 with equal probability, and \( \Phi \in \{0, 1\}^{m \times n} \) is an \( m \times n \) binary matrix with \( \Phi_{h(i), j} = 1 \) and all remaining entries 0, where \( h : [n] \to [m] \) is a random map such that for each \( i \in [n] \), \( h(i) = j \) with probability \( 1/m \) for each \( j \in [m] \). For any matrix \( A \in \mathbb{R}^{n \times d} \), \( \Pi A \) can be computed in \( O(\text{nnz}(A)) \) time. For any tensor \( A \in \mathbb{R}^{n \times d_1 \times d_2} \), \( \Pi A \) can be computed in \( O(\text{nnz}(A)) \) time. Let \( \Pi_1, \Pi_2, \Pi_3 \) denote three CountSketch transforms. For any tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), \( A(\Pi_1, \Pi_2, \Pi_3) \) can be computed in \( O(\text{nnz}(A)) \) time.

If the above scalar \( \sigma \) is not specified in the context, we assume the scalar \( \sigma \) to be 1.

Definition B.17 (Gaussian matrix or Gaussian transform). Let \( S = \sigma \cdot G \in \mathbb{R}^{m \times n} \) where \( \sigma \) is a scalar, and each entry of \( G \in \mathbb{R}^{m \times n} \) is chosen independently from the standard Gaussian distribution. For any matrix \( A \in \mathbb{R}^{n \times d} \), \( SA \) can be computed in \( O(m \cdot \text{nnz}(A)) \) time. For any tensor \( A \in \mathbb{R}^{n \times d_1 \times d_2} \), \( SA \) can be computed in \( O(m \cdot \text{nnz}(A)) \) time.

If the above scalar \( \sigma \) is not specified in the context, we assume the scalar \( \sigma \) to be \( 1/\sqrt{m} \). In most places, we can combine CountSketch and Gaussian transforms to achieve the following:

Definition B.18 (CountSketch + Gaussian transform). Let \( S' = S \Pi \), where \( \Pi \in \mathbb{R}^{t \times n} \) is the CountSketch transform (defined in Definition B.16) and \( S \in \mathbb{R}^{m \times t} \) is the Gaussian transform (defined in Definition B.17). For any matrix \( A \in \mathbb{R}^{n \times d} \), \( S'A \) can be computed in \( O(\text{nnz}(A) + dtm^\omega - 2) \) time, where \( \omega \) is the matrix multiplication exponent.
Lemma B.19 (Affine Embedding - Theorem 39 in [CW13]). Given matrices $A \in \mathbb{R}^{n \times r}, B \in \mathbb{R}^{n \times d}$, and rank($A$) = $k$, let $m = \text{poly}(k/\epsilon)$, $S \in \mathbb{R}^{m \times n}$ be a sparse embedding matrix (Definition B.16) with scalar $\sigma = 1$. Then with probability at least 0.999, $\forall X \in \mathbb{R}^{r \times d}$, we have

$$(1 - \epsilon) \cdot \|AX - B\|_F^2 \leq \|S(AX - B)\|_F^2 \leq (1 + \epsilon)\|AX - B\|_F^2.$$ 

Lemma B.20 (see, e.g., Lemma 10 in version 1 of [BWZ16]8). Let $m = \Omega(k/\epsilon)$, $S = \frac{1}{\sqrt{m}} \cdot G$, where $G \in \mathbb{R}^{m \times n}$ is a random matrix where each entry is an i.i.d Gaussian $N(0,1)$. Then with probability at least 0.998, $S$ satisfies (1 ± 1/8) Subspace Embedding (Definition B.1) for any fixed matrix $C \in \mathbb{R}^{n \times k}$, and it also satisfies $\mathcal{O}(\sqrt{\epsilon/k})$ Approximate Matrix Product (Definition B.2) for any fixed matrix $A$ and $B$ which has the same number of rows.

Lemma B.21 (see, e.g., Lemma 11 in version 1 of [BWZ16]8). Let $m = \Omega(k^2 + k/\epsilon)$, $\Pi \in \mathbb{R}^{m \times n}$, where $\Pi$ is a sparse embedding matrix (Definition B.16) with scalar $\sigma = 1$, then with probability at least 0.998, $S$ satisfies (1 ± 1/8) Subspace Embedding (Definition B.1) for any fixed matrix $C \in \mathbb{R}^{n \times k}$, and it also satisfies $\mathcal{O}(\sqrt{\epsilon/k})$ Approximate Matrix Product (Definition B.2) for any fixed matrix $A$ and $B$ which has the same number of rows.

Lemma B.22 (see, e.g., Lemma 12 in version 1 of [BWZ16]8). Let $m_2 = \Omega(k^2 + k/\epsilon)$, $\Pi \in \mathbb{R}^{m_2 \times n}$, where $\Pi$ is a sparse embedding matrix (Definition B.16) with scalar $\sigma = 1$. Let $m_1 = \Omega(k/\epsilon)$, $S = \frac{1}{\sqrt{m_1}} \cdot G$, where $G \in \mathbb{R}^{m_1 \times m_2}$ is a random matrix where each entry is an i.i.d Gaussian $N(0,1)$. Let $S' = \Pi S$. Then with probability at least 0.99, $S'$ is a (1 ± 1/3) Subspace Embedding (Definition B.1) for any fixed matrix $C \in \mathbb{R}^{n \times k}$, and it also satisfies $\mathcal{O}(\sqrt{\epsilon/k})$ Approximate Matrix Product (Definition B.2) for any fixed matrix $A$ and $B$ which have the same number of rows.

Theorem B.23 (Theorem 36 in [CW13]). Given $A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{n \times d}$, suppose $S \in \mathbb{R}^{m \times n}$ is such that $S$ is a (1 ± 1/\sqrt{2}) Subspace Embedding for $A$, and satisfies $\mathcal{O}(\sqrt{\epsilon/k})$ Approximate Matrix Product for matrices $A$ and $C$ where $C$ has $n$ rows, where $C$ depends on $A$ and $B$. If

$$\hat{X} = \arg\min_{X \in \mathbb{R}^{k \times d}} \|SAX - SB\|_F^2,$$

then

$$\|A\hat{X} - B\|_F^2 \leq (1 + \epsilon) \min_{X \in \mathbb{R}^{k \times d}} \|AX - B\|_F^2.$$ 

B.7 Cauchy and $p$-stable transforms

Definition B.24 (Dense Cauchy transform). Let $S = \sigma \cdot C \in \mathbb{R}^{m \times n}$ where $\sigma$ is a scalar, and each entry of $C \in \mathbb{R}^{m \times n}$ is chosen independently from the standard Cauchy distribution. For any matrix $A \in \mathbb{R}^{n \times d}$, $SA$ can be computed in $O(m \cdot \text{nnz}(A))$ time.

Definition B.25 (Sparse Cauchy transform). Let $\Pi = \sigma \cdot SC \in \mathbb{R}^{m \times n}$, where $\sigma$ is a scalar, $S \in \mathbb{R}^{m \times n}$ has each column chosen independently and uniformly from the $m$ standard basis vectors of $\mathbb{R}^m$, and $C \in \mathbb{R}^{n \times n}$ is a diagonal matrix with diagonals chosen independently from the standard Cauchy distribution. For any matrix $A \in \mathbb{R}^{n \times d}$, $\Pi A$ can be computed in $O(\text{nnz}(A))$ time. For any tensor $A \in \mathbb{R}^{n \times d_1 \times d_2}$, $\Pi A$ can be computed in $O(\text{nnz}(A))$ time. Let $\Pi_1 \in \mathbb{R}^{m_1 \times m_1}, \Pi_2 \in \mathbb{R}^{m_2 \times m_2}, \Pi_3 \in \mathbb{R}^{m_3 \times m_3}$ denote three sparse Cauchy transforms. For any tensor $A \in \mathbb{R}^{n_1 \times m_2 \times n_3}, A(\Pi_1, \Pi_2, \Pi_3) \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ can be computed in $O(\text{nnz}(A))$ time.

Definition B.26 (Dense p-stable transform). Let \( p \in (1, 2) \). Let \( S = \sigma \cdot C \in \mathbb{R}^{m \times n} \), where \( \sigma \) is a scalar, and each entry of \( C \in \mathbb{R}^{m \times n} \) is chosen independently from the standard p-stable distribution. For any matrix \( A \in \mathbb{R}^{n \times d} \), \( SA \) can be computed in \( O(m \text{nnz}(A)) \) time.

Definition B.27 (Sparse p-stable transform). Let \( p \in (1, 2) \). Let \( \Pi = \sigma \cdot SC \in \mathbb{R}^{m \times n} \), where \( \sigma \) is a scalar, \( S \in \mathbb{R}^{m \times n} \) has each column chosen independently and uniformly from the \( m \) standard basis vectors of \( \mathbb{R}^m \), and \( C \in \mathbb{R}^{n \times n} \) is a diagonal matrix with diagonals chosen independently from the standard p-stable distribution. For any matrix \( A \in \mathbb{R}^{n \times d} \), \( \Pi A \) can be computed in \( O(\text{nnz}(A)) \) time. For any tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), \( \Pi \) \( A \) can be computed in \( O(\text{nnz}(A)) \) time. Let \( \Pi_1 \in \mathbb{R}^{m_1 \times n_1}, \Pi_2 \in \mathbb{R}^{m_2 \times n_2}, \Pi_3 \in \mathbb{R}^{m_3 \times n_3} \) denote three sparse p-stable transforms. For any tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), \( A(\Pi_1, \Pi_2, \Pi_3) \in \mathbb{R}^{m_1 \times m_2 \times m_3} \) can be computed in \( O(\text{nnz}(A)) \) time.

B.8 Leverage scores

Definition B.28 (Leverage scores). Let \( U \in \mathbb{R}^{n \times k} \) have orthonormal columns, and let \( p_i = u_i^2 / k \), where \( u_i^2 = \|e_i^\top U\|_2^2 \) is the \( i \)-th leverage score of \( U \).

Definition B.29 (Leverage score sampling). Given \( A \in \mathbb{R}^{n \times d} \) with rank \( k \), let \( U \in \mathbb{R}^{n \times k} \) be an orthonormal basis of the column space of \( A \), and for each i let \( p_i \) be the squared row norm of the \( i \)-th row of \( U \), i.e., the \( i \)-th leverage score. Let \( k \cdot p_i \) denote the \( i \)-th leverage score of \( U \) scaled by \( k \). Let \( \beta > 0 \) be a constant and \( q = (q_1, \ldots, q_n) \) denote a distribution such that, for each \( i \in [n] \), \( q_i \geq \beta p_i \). Let \( s \) be a parameter. Construct an \( n \times s \) sampling matrix \( B \) and an \( s \times s \) rescaling matrix \( D \) as follows. Initially, \( B = 0^{n \times s} \) and \( D = 0^{s \times s} \). For each column \( j \) of \( B \), \( D \), independently, and with replacement, pick a row index \( i \in [n] \) with probability \( q_i \), and set \( B_{i,j} = 1 \) and \( D_{j,i} = 1 / \sqrt{q_i} \). We denote this procedure LEVERAGE SCORE SAMPLING according to the matrix \( A \).

B.9 Lewis weights

We follow the exposition of Lewis weights from [CP15].

Definition B.30. For a matrix \( A \), let \( a_i \) denote the \( i \)-th row of \( A \), where \( a_i = (A^i)^\top \) is a column vector. The statistical leverage score of a row \( a_i \) is

\[
\tau_i(A) \overset{\text{def}}{=} a_i^\top (A^\top A)^{-1} a_i = \| (A^\top A)^{-1/2} a_i \|_2^2.
\]

For a matrix \( A \) and norm \( p \), the \( \ell_p \) Lewis weights \( w \) are the unique weights such that for each row \( i \) we have

\[
w_i = \tau_i(W^{1/2 - 1/p} A).
\]

or equivalently,

\[
a_i^\top (A^\top W^{1-2/p} A)^{-1} a_i = w_i^{2/p}.
\]

Lemma B.31 (Lemma 2.4 of [CP15] and Lemma 7 of [CLM+15]). Given a matrix \( A \in \mathbb{R}^{n \times d} \), \( n \geq d \), for any constant \( C > 0, 4 > p \geq 1 \), there is an algorithm which can compute \( C \)-approximate \( \ell_p \) Lewis weights for every row \( i \) of \( A \) in \( O((\text{nnz}(A) + d^p \log d) \log n) \) time, where \( \omega < 2.373 \) is the matrix multiplication exponent [Str69, CW87, Wil12].
Lemma B.32 (Theorem 7.1 of [CP15]). Given matrix $A \in \mathbb{R}^{n \times d}$ ($n \geq d$) with $\ell_p$ ($4 > p \geq 1$) Lewis weights $w_i$, for any set of sampling probabilities $p_i$, $\sum_i p_i = N$,

$$p_i \geq f(d,p)w_i,$$

if $S \in \mathbb{R}^{N \times n}$ has each row chosen independently as the $i$th standard basis vector, multiplied by $1/p_i^{1/p}$, with probability $p_i/N$. Then, overall with probability at least 0.999,

$$\forall x \in \mathbb{R}^d, \frac{1}{2}\|Ax\|_p^p \leq \|SAx\|_p^p \leq 2\|Ax\|_p^p.$$

Furthermore, if $p = 1$, $N = O(d\log d)$. If $1 < p < 2$, $N = O(d \log d \log \log d)$. If $2 \leq p < 4$, $N = O(d^{p/2} \log d)$.

Lemma B.33. Given matrix $A \in \mathbb{R}^{n \times d}$ ($n \geq d$), there is an algorithm to compute a diagonal matrix $D = SS_1$ with $N$ nonzero entries in $O(n \text{poly}(d))$ time such that, with probability at least 0.999, for all $x \in \mathbb{R}^d$

$$\forall x \in \mathbb{R}^d, \frac{1}{10}\|DAx\|_p^p \leq \|Ax\|_p^p \leq 10\|DAx\|_p^p,$$

where $S, S_1$ are two sampling/rescaling matrices. Furthermore, if $p = 1$, then $N = O(d \log d)$. If $1 < p < 2$, then $N = O(d \log d \log \log d)$. If $2 \leq p < 4$, then $N = O(d^{p/2} \log d)$.

Given a matrix $A \in \mathbb{R}^{n \times d}$ ($n \geq d$), by Lemma B.32 and Lemma B.31, we can compute a sampling/rescaling matrix $S$ in $O((mnz(A) + d^k \log d \log n) \log n)$ time with $\tilde{O}(d)$ nonzero entries such that

$$\forall x \in \mathbb{R}^d, \frac{1}{2}\|Ax\|_p^p \leq \|SAx\|_p^p \leq 2\|Ax\|_p^p.$$

Sometimes, poly($d$) is much smaller than $\log n$. In this case, we are also able to compute such a sampling/rescaling matrix $S$ in $n \text{poly}(d)$ time in an alternative way.

To do so, we run one of the input sparsity $\ell_p$ embedding algorithms (see e.g., [MM13]) to compute a well conditioned basis $U$ of the column span of $A$ in $n \text{poly}(d/e)$ time. By sampling according to the well conditioned basis (see e.g. [Cla05, DDH+09, Woo14]), we can compute a sampling/rescaling matrix $S_1$ such that $(1 - \epsilon)\|Ax\|_p^p \leq \|S_1Ax\|_p^p \leq (1 + \epsilon)\|Ax\|_p^p$ where $\epsilon \in (0,1)$ is an arbitrary constant. Notice that $S_1$ has $\text{poly}(d/e)$ nonzero entries, and thus $S_1A$ has size $\text{poly}(d/e)$. Next, we apply Lewis weight sampling according to $S_1A$, and we obtain a sampling/rescaling matrix $S$ for which

$$\forall x \in \mathbb{R}^d, (1 - \frac{1}{3})\|S_1Ax\|_p^p \leq \|SS_1Ax\|_p^p \leq (1 + \frac{1}{3})\|S_1Ax\|_p^p.$$

This implies that

$$\forall x \in \mathbb{R}^d, \frac{1}{2}\|Ax\|_p^p \leq \|SS_1Ax\|_p^p \leq 2\|Ax\|_p^p.$$

Note that $SS_1$ is still a sampling/rescaling matrix according to $A$, and the number of non-zero entries is $\tilde{O}(d)$. The total running time is thus $n \text{poly}(d/e)$, as desired.
B.10 TensorSketch

Let $\phi(v_1, v_2, \cdots, v_q)$ denote the function that maps $q$ vectors $(u_i \in \mathbb{R}^{n_i})$ to the $\prod_{i=1}^{q} n_i$-dimensional vector formed by $v_1 \otimes v_2 \otimes \cdots \otimes u_q$.

We first give the definition of TensorSketch. Similar definitions can be found in previous work [Pag13, PP13, ANW14, WTSA15].

**Definition B.34 (TensorSketch [Pag13]).** Given $q$ points $v_1, v_2, \cdots, v_q$ where for each $i \in [q], v_i \in \mathbb{R}^{n_i}$, let $m$ be the target dimension. The TensorSketch transform is specified using $q$ 3-wise independent hash functions $h_1, \cdots, h_q$, where for each $i \in [q], h_i : [n_i] \rightarrow [m]$, as well as $q$ 4-wise independent sign functions $s_1, \cdots, s_q$, where for each $i \in [q], s_i : [n_i] \rightarrow \{-1, +1\}$.

TensorSketch applied to $v_1, \cdots, v_q$ is then CountSketch applied to $\phi(v_1, \cdots, v_q)$ with hash function $H : [\prod_{i=1}^{q} n_i] \rightarrow [m]$ and sign functions $S : [\prod_{i=1}^{q} n_i] \rightarrow \{-1, +1\}$ defined as follows:

$$H(i_1, \cdots, i_q) = h_1(i_1) + h_2(i_2) + \cdots + h_q(i_q) \pmod{m},$$

and

$$S(i_1, \cdots, i_q) = s_1(i_1) \cdot s_2(i_2) \cdots s_q(i_q).$$

Using the Fast Fourier Transform, TensorSketch$(v_1, \cdots, v_q)$ can be computed in $O(\sum_{i=1}^{q} (\text{nnz}(v_i) + m \log m))$ time.

Note that Theorem 1 in [ANW14] only defines $\phi(v) = v \otimes v \otimes \cdots \otimes v$. Here we state a stronger version of Theorem 1 than in [ANW14], though the proofs are identical; a formal derivation can be found in [DW17].

**Theorem B.35** (Generalized version of Theorem 1 in [ANW14]). Let $S$ be the $(\prod_{i=1}^{q} n_i) \times m$ matrix such that TensorSketch $(v_1, v_2, \cdots, v_q)$ is $\phi(v_1, v_2, \cdots, v_q)S$ for a randomly selected TensorSketch. The matrix $S$ satisfies the following two properties.

Property I (Approximate Matrix Product). Let $A$ and $B$ be matrices with $\prod_{i=1}^{q} n_i$ rows. For $m \geq \left(2 + 3^q\right)/(\epsilon^2 \delta)$, we have

$$\Pr[\|A^T S S^T B - A^T B\|_F^2 \leq \epsilon^2 \|A\|_F^2 \|B\|_F^2] \geq 1 - \delta.$$

Property II (Subspace Embedding). Consider a fixed $k$-dimensional subspace $V$. If $m \geq k^2 (2 + 3^q)/(\epsilon^2 \delta)$, then with probability at least $1 - \delta$, $\|xS\|_2 = (1 \pm \epsilon)\|x\|_2$ simultaneously for all $x \in V$. 

30
C Frobenius Norm for Arbitrary Tensors

Section C.1 presents a Frobenius norm tensor low-rank approximation algorithm with \((1 + \epsilon)\)-approximation ratio. Section C.2 introduces a tool which is able to reduce the size of the objective function from \(n^3\) to \(\text{poly}(k, 1/\epsilon)\). Section C.3 introduces a new problem called tensor multiple regression. Section C.4 presents several bicriteria algorithms. Section C.5 introduces a powerful tool which we call generalized matrix row subset selection. Section C.6 presents an algorithm that is able to select a batch of columns, rows and tubes from a given tensor, and those samples are also able to form a low-rank solution. Section C.7 presents several useful tools for tensor problems, and also two \((1 + \epsilon)\)-approximation CUR decomposition algorithms: one has the optimal sample complexity, and the other has the optimal running time. Section C.9 shows how to solve the problem if the size of the objective function is small. Section C.10 extends several techniques from 3rd order tensors to general \(q\)-th order tensors, for any \(q \geq 3\). Finally, in Section C.11 we also provide a new matrix CUR decomposition algorithm, which is faster than [BW14].

For simplicity of presentation, we assume \(A_k\) exists in theorems (e.g., Theorem C.1) which concern outputting a rank-\(k\) solution, as well as the theorems (e.g., Theorem C.7, Theorem C.8, Theorem C.13) which concern outputting a bicriteria solution (the output rank is larger than \(k\)). For each of the bicriteria theorems, we can obtain a more detailed version when \(A_k\) does not exist, like Theorem 1.1 in Section 1 (by instead considering a tensor sufficiently close to \(A_k\) in objective function value). Note that the theorems for column, row, tube subset selection Theorem C.20 and Theorem C.21 also belong to this first category. In the second category, for each of the rank-\(k\) theorems we can obtain a more detailed version handling all cases, even when \(A_k\) does not exist, like Theorem 1.2 in Section 1 (by instead considering a tensor sufficiently close to \(A_k\) in objective function value).

Several other tensor results or tools (e.g., Theorem C.4, Lemma C.3, Theorem C.40, Theorem C.41, Theorem C.14, Theorem C.46) that we build in this section do not belong to the above two categories. It means those results do not depend on whether \(A_k\) exists or not and whether OPT is zero or not.

C.1 \((1 + \epsilon)\)-approximate low-rank approximation

**Algorithm 2** Frobenius Norm Low-rank Approximation

```plaintext
1: procedure FLOWRANKAPPROX(A, n, k, \epsilon)  \hspace{1em} \triangleright \text{Theorem C.1}
2: \hspace{1em} s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow O(k/\epsilon).
3: \hspace{1em} \text{Choose sketching matrices } S_1 \in \mathbb{R}^{n^2 \times s_1}, S_2 \in \mathbb{R}^{n^2 \times s_2}, S_3 \in \mathbb{R}^{n^2 \times s_3}. \hspace{1em} \triangleright \text{Definition B.18}
4: \hspace{1em} \text{Compute } A_i S_i, \forall i \in [3].
5: \hspace{1em} Y_1, Y_2, Y_3, C \leftarrow \text{FINPUTSPARSITYREDUCTION}(A, A_1 S_1, A_2 S_2, A_3 S_3, n, s_1, s_2, s_3, k, \epsilon). \hspace{1em} \triangleright \text{Algorithm 3}
6: \hspace{1em} \text{Create variables for } X_i \in \mathbb{R}^{s_i \times k}, \forall i \in [3].
7: \hspace{1em} \text{Run polynomial system verifier for } \| (Y_1 X_1) \otimes (Y_2 X_2) \otimes (Y_3 X_3) - C \|^2_F.
8: \hspace{1em} \text{return } A_1 S_1 X_1, A_2 S_2 X_2, \text{ and } A_3 S_3 X_3.
9: end procedure
```

**Theorem C.1.** Given a 3rd order tensor \(A \in \mathbb{R}^{n \times n \times n}\), for any \(k \geq 1, \epsilon \in (0, 1)\), there exists an algorithm which takes \(O(\text{nnz}(A)) + n \text{ poly}(k, 1/\epsilon) + 2^{O(k^2/\epsilon)}\) time and outputs three matrices
Let obtain the following optimization problem,

\[
\sum_{i=1}^{k} U_i \otimes V_i \otimes W_i - A \leq (1 + \epsilon) \min_{\text{rank-}k A_k} \|A_k - A\|_F^2.
\]

holds with probability 9/10.

Proof. Given any tensor \(A \in \mathbb{R}^{n_1 \times n_2 \times n_3}\), we define three matrices \(A_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3}\), \(A_2 \in \mathbb{R}^{n_2 \times n_3 \times n_1}\), \(A_3 \in \mathbb{R}^{n_3 \times n_1 \times n_2}\) such that, for any \(i \in [n_1], j \in [n_2], l \in [n_3]\),

\[
A_{i,j,l} = (A_1)_{i,(j-1)-n_3+l} = (A_2)_{j,(l-1)-n_1+i} = (A_3)_{l,(i-1)n_2+j}.
\]

We define OPT as

\[
\text{OPT} = \min_{\text{rank-}k A'} \|A' - A\|_F^2.
\]

Suppose the optimal \(A_k = U^* \otimes V^* \otimes W^*\). We fix \(V^* \in \mathbb{R}^{n_k}\) and \(W^* \in \mathbb{R}^{n_k}\). We use \(V_1^*, V_2^*, \ldots, V_k^*\) to denote the columns of \(V^*\) and \(W_1^*, W_2^*, \ldots, W_k^*\) to denote the columns of \(W^*\).

We consider the following optimization problem,

\[
\min_{U_1, \ldots, U_k \in \mathbb{R}^n} \left\| \sum_{i=1}^{k} U_i \otimes V_i^* \otimes W_i - A \right\|_F^2,
\]

which is equivalent to

\[
\min_{U_1, \ldots, U_k \in \mathbb{R}^n} \left\| \begin{bmatrix} U_1 & U_2 & \cdots & U_k \end{bmatrix} \begin{bmatrix} V_1^* \otimes W_1^* \\ V_2^* \otimes W_2^* \\ \vdots \\ V_k^* \otimes W_k^* \end{bmatrix} - A \right\|_F^2.
\]

We use matrix \(Z_1\) to denote

\[
\begin{bmatrix}
\text{vec}(V_1^* \otimes W_1^*) \\
\text{vec}(V_2^* \otimes W_2^*) \\
\vdots \\
\text{vec}(V_k^* \otimes W_k^*)
\end{bmatrix} \in \mathbb{R}^{k \times n^2}
\]

and matrix \(U\) to denote \(\begin{bmatrix} U_1 & U_2 & \cdots & U_k \end{bmatrix}\).

Then we can obtain the following equivalent objective function,

\[
\min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2.
\]

Notice that \(\min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2 = \text{OPT}\), since \(A_k = U^*Z_1\).

Let \(S_1^\top \in \mathbb{R}^{s_1 \times n^2}\) be a sketching matrix defined in Definition B.18, where \(s_1 = O(k/\epsilon)\). We obtain the following optimization problem,

\[
\min_{U \in \mathbb{R}^{n \times k}} \|UZ_1S_1 - A_1S_1\|_F^2.
\]

Let \(\hat{U} \in \mathbb{R}^{n \times k}\) denote the optimal solution to the above optimization problem. Then \(\hat{U} = A_1S_1(Z_1S_1)^\top\). By Lemma B.22 and Theorem B.23, we have

\[
\|\hat{U}Z_1 - A_1\|_F^2 \leq (1 + \epsilon) \min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2 = (1 + \epsilon) \text{OPT},
\]

32
which implies
\[
\left\| \sum_{i=1}^{k} \hat{U}_i \otimes V_i^* \otimes W_i^* - A \right\|_F^2 \leq (1 + \epsilon) \text{OPT}.
\]

To write down \( \hat{U}_1, \ldots, \hat{U}_k \), we use the given matrix \( A_1 \), and we create \( s_1 \times k \) variables for matrix \((Z_1S_1)^\dagger\).

As our second step, we fix \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and we convert tensor \( A \) into matrix \( A_2 \).

Let matrix \( Z_2 \) denote
\[
\begin{bmatrix}
\text{vec}(\hat{U}_1 \otimes W_1^*) \\
\text{vec}(\hat{U}_2 \otimes W_2^*) \\
\vdots \\
\text{vec}(\hat{U}_k \otimes W_k^*)
\end{bmatrix}.
\]

We consider the following objective function,
\[
\min_{V \in \mathbb{R}^{n \times k}} \| VZ_2 - A_2 \|_F^2,
\]
for which the optimal cost is at most \((1 + \epsilon) \text{OPT}\).

Let \( S_2^\top \in \mathbb{R}^{s_2 \times n^2} \) be a sketching matrix defined in Definition B.18, where \( s_2 = O(k/\epsilon) \). We sketch \( S_2 \) on the right of the objective function to obtain the new objective function,
\[
\min_{V \in \mathbb{R}^{n \times k}} \| VZ_2S_2 - A_2S_2 \|_F^2.
\]

Let \( \hat{V} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{V} = A_2S_2(Z_2S_2)^\dagger \). By Lemma B.22 and Theorem B.23, we have,
\[
\| \hat{V}Z_2 - A_2 \|_F^2 \leq (1 + \epsilon) \min_{V \in \mathbb{R}^{n \times k}} \| VZ_2 - A_2 \|_F^2 \leq (1 + \epsilon)^2 \text{OPT},
\]
which implies
\[
\left\| \sum_{i=1}^{k} \hat{U}_i \otimes \hat{V}_i \otimes W_i^* - A \right\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.
\]

To write down \( \hat{V}_1, \ldots, \hat{V}_k \), we need to use the given matrix \( A_2 \in \mathbb{R}^{n^2 \times n} \), and we need to create \( s_2 \times k \) variables for matrix \((Z_2S_2)^\dagger\).

As our third step, we fix the matrices \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( \hat{V} \in \mathbb{R}^{n \times k} \). We convert tensor \( A \in \mathbb{R}^{n^2 \times n^2} \)

into matrix \( A_3 \in \mathbb{R}^{n^2 \times n} \). Let matrix \( Z_3 \) denote
\[
\begin{bmatrix}
\text{vec}(\hat{U}_1 \otimes \hat{V}_1) \\
\text{vec}(\hat{U}_2 \otimes \hat{V}_2) \\
\vdots \\
\text{vec}(\hat{U}_k \otimes \hat{V}_k)
\end{bmatrix}.
\]

We consider the following objective function,
\[
\min_{W \in \mathbb{R}^{n \times k}} \| WZ_3 - A_3 \|_F^2,
\]
which has optimal cost at most \((1 + \epsilon)^2 \text{OPT}\).

Let \( S_3^\top \in \mathbb{R}^{s_3 \times n^2} \) be a sketching matrix defined in Definition B.18, where \( s_3 = O(k/\epsilon) \). We sketch \( S_3 \) on the right of the objective function to obtain a new objective function,
\[
\min_{W \in \mathbb{R}^{n \times k}} \| WZ_3S_3 - A_3S_3 \|_F^2.
\]
Thus, we have

\[ \|\hat{W}Z_3 - A_3\|_F^2 \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}. \]

Thus, we have

\[ \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{3} (A_1S_1X_1)_i \otimes (A_2S_2X_2)_i \otimes (A_3S_3X_3)_i - A \right\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}. \]

Let \( V_1, V_2 = A_2S_2, V_3 = A_3S_3 \), we then apply Lemma C.3, and we obtain \( \hat{V}_1, \hat{V}_2, \hat{V}_3, C \). We then apply Theorem C.45. Correctness follows by rescaling \( \epsilon \) by a constant factor.

**Running time.** Due to Definition B.18, the running time of line 4 is \( O(\text{nnz}(A)) + n \text{poly}(k) \). The running time of line 5 is shown by Lemma C.3, and the running time of line 7 is shown by Theorem C.45.

**Theorem C.2.** Suppose we are given a 3rd order \( n \times n \times n \) tensor \( A \) such that each entry can be written using \( n^\delta \) bits, where \( \delta > 0 \) is a given, value which can be arbitrarily small (e.g., we could have \( n^\delta = O(\log n) \)). Define \( \text{OPT} = \inf_{\text{rank} \rightarrow k} \|A_k - A\|_F^2 \). For any \( k \geq 1 \), and for any \( 0 < \epsilon < 1 \), define \( n^{\delta'} = O(n^\delta 2^{O(k^2/\epsilon)}) \). (I) If \( \text{OPT} > 0 \), and there exists a rank-\( k \) \( A_k = U^* \otimes V^* \otimes W^* \) tensor, with size \( n \times n \times n \), such that \( \|A_k - A\|_F^2 = \text{OPT} \), and \( \max(\|U^*\|_F, \|V^*\|_F, \|W^*\|_F) \leq 2^{O(n^{\delta'})} \), then there exists an algorithm that takes \( (\text{nnz}(A) + n \text{poly}(k, 1/\epsilon) + 2^{O(k^2/\epsilon)})n^\delta \) time in the unit cost RAM model with word size \( O(\log n) \) bits\(^9\) and outputs three \( n \times k \) matrices \( U, V, W \) such that

\[ \|U \otimes V \otimes W - A\|_F^2 \leq (1 + \epsilon) \text{OPT} \quad (5) \]

holds with probability \( 9/10 \), and each entry of each of \( U, V, W \) fits in \( n^{\delta'} \) bits.

(II) If \( \text{OPT} > 0 \), and \( A_k \) does not exist, and there exist three \( n \times k \) matrices \( U', V', W' \) for which \( \max(\|U'\|_F, \|V'\|_F, \|W'\|_F) \leq 2^{O(n^{\delta'})} \) and \( \|U' \otimes V' \otimes W' - A\|_F^2 \leq (1 + \epsilon/2) \text{OPT} \), then we can find \( U, V, W \) such that (5) holds.

(III) If \( \text{OPT} = 0 \) and \( A_k \) does exist, and there exists a solution \( U^*, V^*, W^* \) such that each entry can be written by \( n^{\delta'} \) bits, then we can obtain (5).

(IV) If \( \text{OPT} = 0 \), and there exist three \( n \times k \) matrices \( U, V, W \) such that \( \max(\|U\|_F, \|V\|_F, \|W\|_F) \leq 2^{O(n^{\delta'})} \) and

\[ \|U \otimes V \otimes W - A\|_F^2 \leq (1 + \epsilon) \text{OPT} + 2^{-\Omega(n^{\delta'})} = 2^{-\Omega(n^{\delta'})}, \quad (6) \]

then we can output \( U, V, W \) such that (6) holds.

Further if \( A_k \) exists, we can output a number \( Z \) for which \( \text{OPT} \leq Z \leq (1 + \epsilon) \text{OPT} \). For all the cases above, the algorithm runs in the same time as (I) and succeeds with probability at least \( 9/10 \).

**Proof.** This follows by the discussion in Section 1, Theorem C.1 and Theorem C.45 in Section C.9.

Part (I) Suppose \( \delta > 0 \) and \( A_k = U^* \otimes V^* \otimes W^* \) exists and each of \( \|U^*\|_F, \|V^*\|_F \), and \( \|W^*\|_F \) is bounded by \( 2^{O(n^{\delta'})} \). Assume the computation model is the unit cost RAM model with words of size \( O(\log n) \) bits, and allow each number of the input tensor \( A \) to be written using \( n^\delta \) bits. For the

\(^9\)The entries of \( A \) are assumed to fit in \( n^{\delta'} \) words.
case when OPT is nonzero, using the proof of Theorem C.1 and Theorems C.45, B.11, there exists a lower bound on the cost OPT, which is at least $2^{-O(n^\delta)2O(k^2/\epsilon)}$. We can round each entry of matrices $U^*, V^*, W^*$ to be an integer expressed using $O(n^\delta)$ bits to obtain $U', V', W'$. Using the triangle inequality and our lower bound on OPT, it follows that $U', V', W'$ provide a $(1+\epsilon)$-approximation.

Thus, applying Theorem C.1 by fixing $U', V', W'$ and using Theorem C.45 at the end, we can output three matrices $U, V, W$, where each entry can be written using $n^\delta$ bits, so that we satisfy $\|U \otimes V \otimes W - A\|_F^2 \leq (1+\epsilon)\text{OPT}$.

For the running time, since each entry of the input is bounded by $n^\delta$ bits, due to Theorem C.1, we need $(\text{nnz}(A) + n \text{poly}(k/\epsilon)) \cdot n^\delta$ time to reduce the size of the problem to $\text{poly}(k/\epsilon)$ size (with each number represented using $O(n^\delta)$ bits). According to Theorem C.45, the running time of using a polynomial system verifier to get the solution is $2^{O(k^2/\epsilon)nO(\delta')} = 2^{O(k^2/\epsilon)nO(\delta)}$ time. Thus the total running time is $(\text{nnz}(A) + n \text{poly}(k/\epsilon))n^\delta + 2^{O(k^2/\epsilon)}nO(\delta)$.

Part (II) is similar to Part (I). Part (III) is trivial to prove since there exists a solution which can be written down in the bit model, so we obtain a $(1+\epsilon)$-approximation. Part (IV) is also very similar to Part (II).

\[ \Box \]

### C.2 Input sparsity reduction

**Algorithm 3** Reducing the Size of the Objective Function from $\text{poly}(n)$ to $\text{poly}(k)$

1: `procedure` FINPUTSPARSITYREDUCTION($A, V_1, V_2, V_3, n, b_1, b_2, b_3, k, \epsilon$) \> Lemma C.3
2: \hspace{1em} $c_1 \leftarrow c_2 \leftarrow c_3 \leftarrow \text{poly}(k, 1/\epsilon)$.
3: \hspace{1em} Choose sparse embedding matrices $T_1 \in \mathbb{R}^{c_1 \times n}, T_2 \in \mathbb{R}^{c_2 \times n}, T_3 \in \mathbb{R}^{c_3 \times n}$. \> Definition B.16
4: \hspace{1em} $\hat{V}_i \leftarrow T_i V_i \in \mathbb{R}^{c_i \times b_i}, \forall i \in [3]$.
5: \hspace{1em} $C \leftarrow A(T_1, T_2, T_3) \in \mathbb{R}^{c_1 \times c_2 \times c_3}$.
6: \hspace{1em} `return` $\hat{V}_1, \hat{V}_2, \hat{V}_3$ and $C$.
7: `end procedure`

**Lemma C.3.** Let poly($k, 1/\epsilon$) $\geq b_1 b_2 b_3 \geq k$. Given a tensor $A \in \mathbb{R}^{n \times n \times n}$ and three matrices $V_1 \in \mathbb{R}^{n \times b_1}, V_2 \in \mathbb{R}^{n \times b_2}$, and $V_3 \in \mathbb{R}^{n \times b_3}$, there exists an algorithm that takes $O(\text{nnz}(A) + \text{nnz}(V_1) + \text{nnz}(V_2) + \text{nnz}(V_3)) = O(\text{nnz}(A) + n \text{poly}(k/\epsilon))$ time and outputs a tensor $C \in \mathbb{R}^{c_1 \times c_2 \times c_3}$ and three matrices $\hat{V}_1 \in \mathbb{R}^{c_1 \times b_1}, \hat{V}_2 \in \mathbb{R}^{c_2 \times b_2}$ and $\hat{V}_3 \in \mathbb{R}^{c_3 \times b_3}$ with $c_1 = c_2 = c_3 = \text{poly}(k, 1/\epsilon)$, such that with probability at least 0.99, for all $\alpha > 0, X_1, X_2, X_3 \in \mathbb{R}^{b_1 \times k}, X_2' \in \mathbb{R}^{b_2 \times k}, X_3' \in \mathbb{R}^{b_3 \times k}$ satisfy that,

\[
\left\| \sum_{i=1}^{k} (\hat{V}_1 X_1^i) \otimes (\hat{V}_2 X_2^i) \otimes (\hat{V}_3 X_3^i) - C \right\|_F^2 \leq \alpha \left\| \sum_{i=1}^{k} (V_1 X_1^i) \otimes (V_2 X_2^i) \otimes (V_3 X_3^i) - C \right\|_F^2,
\]

then,

\[
\left\| \sum_{i=1}^{k} (V_1 X_1^i) \otimes (V_2 X_2^i) \otimes (V_3 X_3^i) - A \right\|_F^2 \leq (1+\epsilon)\alpha \left\| \sum_{i=1}^{k} (V_1 X_1^i) \otimes (V_2 X_2^i) \otimes (V_3 X_3^i) - A \right\|_F^2.
\]

**Proof.** Let $X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}$. First, we define $Z_1 = ((V_2 X_2)^T \otimes (V_3 X_3)^T) \in \mathbb{R}^{k \times n^2}$. (Note that, for each $i \in [k]$, the $i$-th row of matrix $Z_1$ is vec($((V_2 X_2_i) \otimes (V_3 X_3_i))$.) Then, by
Then, by flattening, we have
\[
\left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2 = \left\| V_1 X_1 \cdot Z - A \right\|_F^2.
\]
We choose a sparse embedding matrix (Definition B.16) \( T_1 \in \mathbb{R}^{c_1 \times n} \) with \( c_1 = \text{poly}(k, 1/\epsilon) \) rows. Since \( V_1 \) has \( b_1 \leq \text{poly}(k/\epsilon) \) columns, according to Lemma B.19 with probability 0.999, for all \( X_1 \in \mathbb{R}^{b_1 \times k} \), \( Z \in \mathbb{R}^{k \times n^2} \),
\[
(1 - \epsilon)\|V_1 X_1 Z - A_1\|_F^2 \leq \|T_1 V_1 X_1 Z - T_1 A_1\|_F^2 \leq (1 + \epsilon)\|V_1 X_1 Z - A_1\|_F^2.
\]
Therefore, we have
\[
\|T_1 V_1 X_1 \cdot Z_1 - T_1 A_1\|_F^2 = (1 \pm \epsilon) \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2.
\]
Second, we unflatten matrix \( T_1 A_1 \in \mathbb{R}^{c_1 \times n^2} \) to obtain a tensor \( A' \in \mathbb{R}^{c_1 \times n \times n} \). Then we flatten \( A' \) along the second direction to obtain \( A_2 \in \mathbb{R}^{n \times c_1} \). We define \( Z_2 = (T_1 V_1 X_1)^T \otimes (V_3 X_3)^T \in \mathbb{R}^{k \times c_1 n} \). Then, by flattening,
\[
\|V_2 X_2 \cdot Z_2 - A_2\|_F^2 = \|T_1 V_1 X_1 \cdot Z_1 - T_1 A_1\|_F^2
\]
\[
= (1 \pm \epsilon) \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2.
\]
We choose a sparse embedding matrix (Definition B.16) \( T_2 \in \mathbb{R}^{c_2 \times n} \) with \( c_2 = \text{poly}(k, 1/\epsilon) \) rows. Then according to Lemma B.19 with probability 0.999, for all \( X_2 \in \mathbb{R}^{b_2 \times k} \), \( Z \in \mathbb{R}^{k \times c_1 n} \),
\[
(1 - \epsilon)\|V_2 X_2 Z - A_2\|_F^2 \leq \|T_2 V_2 X_2 Z - T_2 A_2\|_F^2 \leq (1 + \epsilon)\|V_2 X_2 Z - A_2\|_F^2.
\]
Therefore, we have
\[
\|T_2 V_2 X_2 \cdot Z_2 - T_2 A_2\|_F^2 = (1 \pm \epsilon)\|V_2 X_2 \cdot Z_2 - A_2\|_F^2
\]
\[
= (1 \pm \epsilon)^2 \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2.
\]
Third, we unflatten matrix \( T_2 A_2 \in \mathbb{R}^{c_2 \times c_1 n} \) to obtain a tensor \( A''(= A(T_1, T_2, I)) \in \mathbb{R}^{c_1 \times c_2 \times n} \). Then we flatten tensor \( A'' \) along the last direction (the third direction) to obtain matrix \( A_3 \in \mathbb{R}^{n \times c_1 c_2} \). We define \( Z_3 = (T_1 V_1 X_1)^T \otimes (T_2 V_2 X_2)^T \in \mathbb{R}^{k \times c_1 c_2} \). Then, by flattening, we have
\[
\|V_3 X_3 \cdot Z_3 - A_3\|_F^2 = \|T_2 V_2 X_2 \cdot Z_2 - T_2 A_2\|_F^2
\]
\[
= (1 \pm \epsilon)^2 \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2.
\]
We choose a sparse embedding matrix (Definition B.16) \( T_3 \in \mathbb{R}^{c_3 \times n} \) with \( c_3 = \text{poly}(k, 1/\epsilon) \) rows. Then according to Lemma B.19 with probability 0.999, for all \( X_3 \in \mathbb{R}^{b_3 \times k} \), \( Z \in \mathbb{R}^{k \times c_1 c_2} \),
\[
(1 - \epsilon)\|V_3 X_3 Z - A_3\|_F^2 \leq \|T_3 V_3 X_3 Z - T_3 A_3\|_F^2 \leq (1 + \epsilon)\|V_3 X_3 Z - A_3\|_F^2.
\]
Therefore, we have
\[ \|T_3 V_3 X_3 \cdot Z_3 - T_3 A_3\|^2_F = (1 + \epsilon)^3 \left\| \sum_{i=1}^{k} (V_1 X_1_i) \otimes (V_2 X_2_i) \otimes (V_3 X_3_i) - A \right\|^2_F. \]

Note that
\[ \|T_3 V_3 X_3 \cdot Z_3 - T_3 A_3\|^2_F = \left\| \sum_{i=1}^{k} (T_1 V_1 X_1_i) \otimes (T_2 V_2 X_2_i) \otimes (T_3 V_3 X_3_i) - A(T_1, T_2, T_3) \right\|^2_F, \]
and thus, we have \( \forall X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k} \)
\[ \left\| \sum_{i=1}^{k} (T_1 V_1 X_1_i) \otimes (T_2 V_2 X_2_i) \otimes (T_3 V_3 X_3_i) - A(T_1, T_2, T_3) \right\|^2_F = (1 + \epsilon)^3 \left\| \sum_{i=1}^{k} (V_1 X_1_i) \otimes (V_2 X_2_i) \otimes (V_3 X_3_i) - A \right\|^2_F. \]

Let \( \hat{V}_i \) denote \( T_i V_i \), for each \( i \in [3] \). Let \( C \in \mathbb{R}^{c_1 \times c_2 \times c_3} \) denote \( A(T_1, T_2, T_3) \). For \( \alpha > 1 \), if
\[ \left\| \sum_{i=1}^{k} (\hat{V}_1 X_1_i') \otimes (\hat{V}_2 X_2_i') \otimes (\hat{V}_3 X_3_i') - C \right\|^2_F \leq \alpha \left\| \sum_{i=1}^{k} (\hat{V}_1 X_1_i) \otimes (\hat{V}_2 X_2_i) \otimes (\hat{V}_3 X_3_i) - C \right\|^2_F, \]
then
\[ \left\| \sum_{i=1}^{k} (V_1 X_1_i') \otimes (V_2 X_2_i') \otimes (V_3 X_3_i') - C \right\|^2_F \leq \frac{1}{(1 - \epsilon)^3} \left\| \sum_{i=1}^{k} (\hat{V}_1 X_1_i) \otimes (\hat{V}_2 X_2_i) \otimes (\hat{V}_3 X_3_i) - C \right\|^2_F \leq \frac{1}{(1 - \epsilon)^3} \alpha \left\| \sum_{i=1}^{k} (\hat{V}_1 X_1_i) \otimes (\hat{V}_2 X_2_i) \otimes (\hat{V}_3 X_3_i) - C \right\|^2_F \leq \frac{(1 + \epsilon)^3}{(1 - \epsilon)^3} \left\| \sum_{i=1}^{k} (V_1 X_1_i) \otimes (V_2 X_2_i) \otimes (V_3 X_3_i) - C \right\|^2_F. \]

By rescaling \( \epsilon \) by a constant, we complete the proof of correctness.

**Running time.** According to Section B.6, for each \( i \in [3] \), \( T_i V_i \) can be computed in \( O(nnz(V_i)) \) time, and \( A(T_1, T_2, T_3) \) can be computed in \( O(nnz(A)) \) time.

By the analysis above, the proof is complete. \( \square \)

### C.3 Tensor multiple regression

**Theorem C.4.** Given matrices \( A \in \mathbb{R}^{d \times n^2}, U, V \in \mathbb{R}^{n \times k} \), let \( B \in \mathbb{R}^{k \times n^2} \) denote \( U^\top \otimes V^\top \). There exists an algorithm that takes \( O(nnz(A) + nnz(U) + nnz(V) + d \text{poly}(k, 1/\epsilon)) \) time and outputs a matrix \( W' \in \mathbb{R}^{d \times k} \) such that,
\[ \|W'B - A\|_F \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{d \times k}} \|WB - A\|_F. \]
Algorithm 4 Procrustes Norm Tensor Multiple Regression

1: procedure FTENSORMULTIPLEREGRESSION($A, U, V, d, n, k$) \Comment{Theorem C.4}
2: \hspace{1em} $s \leftarrow O(k^2 + k/\epsilon)$.
3: \hspace{1em} Choose $S \in \mathbb{R}^{n^2 \times s}$ to be a TensorSketch. \Comment{Definition B.34}
4: \hspace{1em} Compute $A \cdot S$.
5: \hspace{1em} Compute $B \cdot S$.
6: \hspace{1em} $W \leftarrow (AS)(BS)$\textsuperscript{T}
7: \hspace{1em} return $W$.
8: end procedure

Proof. We choose a TensorSketch (Definition B.34) $S \in \mathbb{R}^{n^2 \times s}$ to reduce the problem to a smaller problem,
\[
\min_{W \in \mathbb{R}^{d \times k}} \|WBS - AS\|_F^2.
\]
Let $W'$ denote the optimal solution to the above problem. Following a similar proof to that in Section C.7.3, if $S$ is a $(1\pm1/2)$-subspace embedding and satisfies $\sqrt{\epsilon/k}$-approximate matrix product, then $W'$ provides a $(1 + \epsilon)$-approximation to the original problem. By Theorem B.35, we have $s = O(k^2 + k/\epsilon)$.

Running time. According to Definition B.34, $BS$ can be computed in $O(nnz(U) + nnz(V)) + poly(k/\epsilon)$ time. Notice that each row of $S$ has exactly 1 nonzero entry, thus $AS$ can be computed in $O(nnz(A))$ time. Since $BS \in \mathbb{R}^{k \times s}$ and $AS \in \mathbb{R}^{d \times s}$, $\min_{W \in \mathbb{R}^{d \times k}} \|WBS - AS\|_F^2$ can be solved in $d \cdot poly(sk) = d \cdot poly(k/\epsilon)$ time. \qed

C.4 Bicriteria algorithms

C.4.1 Solving a small regression problem

Lemma C.5. Given tensor $A \in \mathbb{R}^{n \times n \times n}$ and three matrices $U \in \mathbb{R}^{n \times s_1}$, $V \in \mathbb{R}^{n \times s_2}$ and $W \in \mathbb{R}^{n \times s_3}$, there exists an algorithm that takes $O(nnz(A) + n \cdot poly(s_1, s_2, s_3, 1/\epsilon))$ time and outputs $\alpha' \in \mathbb{R}^{s_1 \times s_2 \times s_3}$ such that
\[
\left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha'_{i,j,l} \cdot U_i \otimes V_j \otimes W_l - A \right\|_F^2 \leq (1 + \epsilon) \min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot U_i \otimes V_j \otimes W_l - A \right\|_F^2.
\]

holds with probability at least .99.

Proof. We define $\tilde{b} \in \mathbb{R}^n$ to be the vector where the $i + (j-1)n + (l-1)n^2$-th entry of $\tilde{b}$ is $A_{i,j,l}$. We define $\tilde{A} \in \mathbb{R}^{n \times s_1 s_2 s_3}$ to be the matrix where the $(i + (j-1)n + (l-1)n^2, i' + (j'-1)s_2 + (l'-1)s_2 s_3)$ entry is $U_{i',i} \cdot V_{j',j} \cdot W_{l',l}$. This problem is equivalent to a linear regression problem,
\[
\min_{x \in \mathbb{R}^{s_1 s_2 s_3}} \|\tilde{A}x - \tilde{b}\|_2^2,
\]
where $\tilde{A} \in \mathbb{R}^{n \times s_1 s_2 s_3}$, $\tilde{b} \in \mathbb{R}^n$. Thus, it can be solved fairly quickly using recent work [CW13, MM13, NN13]. However, the running time of this naïve is $\Omega(n^3)$, since we have to write down each entry of $A$. In the next few paragraphs, we show how to improve the running time to $nnz(A) + n \cdot poly(s_1, s_2, s_3)$.
Since \( \alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3} \), \( \alpha \) can be always written as \( \alpha = X_1 \otimes X_2 \otimes X_3 \), where \( X_1 \in \mathbb{R}^{s_1 \times s_2 \times s_3} \), \( X_2 \in \mathbb{R}^{s_2 \times s_1 \times s_3} \), \( X_3 \in \mathbb{R}^{s_1 \times s_2 \times s_3} \), we have

\[
\begin{align*}
\min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} & \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot U_i \otimes V_j \otimes W_l - A \right\|^2_F = \\
\min_{X_1 \in \mathbb{R}^{s_1 \times s_2 \times s_3}, X_2 \in \mathbb{R}^{s_2 \times s_1 \times s_3}, X_3 \in \mathbb{R}^{s_1 \times s_2 \times s_3}} & \left\| (UX_1) \otimes (VX_2) \otimes (WX_3) - A \right\|^2_F.
\end{align*}
\]

By Lemma C.3, we can reduce the problem size \( n \times n \times n \) to a smaller problem that has size \( t_1 \times t_2 \times t_3 \),

\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{s_1} (T_1 UX_1)_i \otimes (T_2 VX_2)_i \otimes (T_3 WX_3)_i - A(T_1, T_2, T_3) \right\|^2_F,
\]

where \( T_1 \in \mathbb{R}^{t_1 \times n} \), \( T_2 \in \mathbb{R}^{t_2 \times n} \), \( T_3 \in \mathbb{R}^{t_3 \times n} \), \( t_1 = t_2 = t_3 = \text{poly}(s_1 s_2 s_3 / \epsilon) \). Notice that

\[
\begin{align*}
\min_{X_1, X_2, X_3} & \left\| \sum_{i=1}^{s_1} (T_1 UX_1)_i \otimes (T_2 VX_2)_i \otimes (T_3 WX_3)_i - A(T_1, T_2, T_3) \right\|^2_F \\
= & \min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot (T_1 U)_i \otimes (T_2 V)_j \otimes (T_3 W)_l - A(T_1, T_2, T_3) \right\|^2_F.
\end{align*}
\]

Let

\[
\alpha' = \arg \min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot (T_1 U)_i \otimes (T_2 V)_j \otimes (T_3 W)_l - A(T_1, T_2, T_3) \right\|^2_F,
\]

then we have

\[
\left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha'_{i,j,l} \cdot U_i \otimes V_j \otimes W_l - A \right\|^2_F \leq (1 + \epsilon) \min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot U_i \otimes V_j \otimes W_l - A \right\|^2_F.
\]

Again, according to Lemma C.3, the total running time is then \( O(\text{nnz}(A) + n \text{poly}(s_1, s_2, s_3, 1/\epsilon)) \).

**Lemma C.6.** Given tensor \( A \in \mathbb{R}^{n \times n \times n} \), and two matrices \( U \in \mathbb{R}^{n \times s} \), \( V \in \mathbb{R}^{n \times s} \) with rank \( (U) = r_1 \), rank \((V) = r_2 \), let \( T_1 \in \mathbb{R}^{t_1 \times n} \), \( T_2 \in \mathbb{R}^{t_2 \times n} \) be two sparse embedding matrices (Definition B.16) with \( t_1 = \text{poly}(r_1/\epsilon) \), \( t_2 = \text{poly}(r_2/\epsilon) \). Then with probability at least 0.99, \( \forall X \in \mathbb{R}^{n \times s} \),

\[
(1 - \epsilon)\|U \otimes V \otimes X - A\|_F^2 \leq \|T_1 U \otimes T_2 V \otimes X - A(T_1, T_2, I)\|_F^2 \leq (1 + \epsilon)\|U \otimes V \otimes X - A\|_F^2.
\]

**Proof.** Let \( X \in \mathbb{R}^{n \times s} \). We define \( Z_1 = (V^\top \otimes X^\top) \in \mathbb{R}^{s \times n^2} \). We choose a sparse embedding matrix (Definition B.16) \( T_1 \in \mathbb{R}^{t_1 \times n} \) with \( t_1 = \text{poly}(r_1/\epsilon) \) rows. According to Lemma B.19 with probability 0.999, for all \( Z \in \mathbb{R}^{s \times n^2} \),

\[
(1 - \epsilon)\|UZ - A_1\|_F^2 \leq \|T_1 UZ - T_1 A_1\|_F^2 \leq (1 + \epsilon)\|T_1 UZ - A_1\|_F^2.
\]
It means that

\[(1 - \epsilon)\|UZ_1 - A_1\|_F^2 \leq \|T_1UZ_1 - T_1A_1\|_F^2 \leq (1 + \epsilon)\|UZ_1 - A_1\|_F^2.\]

Second, we unflatten matrix \(T_1A_1 \in \mathbb{R}^{t_1 \times n^2}\) to obtain a tensor \(A' \in \mathbb{R}^{t_1 \times n \times n}\). Then we flatten \(A'\) along the second direction to obtain \(A'_2 \in \mathbb{R}^{n \times t_1 n}\). We define \(Z_2 = ((T_1U)^T \odot X^T) \in \mathbb{R}^{n \times t_1 n}\). Then, by flattening,

\[\|V \cdot Z_2 - A'_2\|_F^2 = \|T_1U \cdot Z_1 - T_1A_1\|_F^2 = (1 \pm \epsilon)\|U \otimes V \otimes X - A\|_F^2.\]

We choose a sparse embedding matrix (Definition B.16) \(T_2 \in \mathbb{R}^{t_2 \times n}\) with \(t_2 = \text{poly}(r_2/\epsilon)\) rows. Then according to Lemma B.19 with probability 0.999, for all \(Z \in \mathbb{R}^{s \times t_1 n}\),

\[(1 - \epsilon)\|VZ - A_2\|_F^2 \leq \|T_2VZ - T_2A_2\|_F^2 \leq (1 + \epsilon)\|VZ - A'_2\|_F^2.\]

Thus,

\[\|T_2V \cdot Z_2 - T_2A_2\|_F^2 = (1 \pm \epsilon)^2\|U \otimes V \otimes X - A\|_F^2.\]

After rescaling \(\epsilon\) by a constant, with probability at least 0.99, \(\forall X \in \mathbb{R}^{n \times s}\),

\[(1 - \epsilon)\|U \otimes V \otimes X - A\|_F^2 \leq \|T_1U \otimes T_2V \otimes X - A(T_1T_2I)\|_F^2 \leq (1 + \epsilon)\|U \otimes V \otimes X - A\|_F^2.\]

\[\square\]

### C.4.2 Algorithm I

We start with a slightly unoptimized bicriteria low rank approximation algorithm.

**Algorithm 5** Frobenius Norm Bicriteria Low Rank Approximation Algorithm, rank-\(O(k^3/\epsilon^3)\)

1: procedure FTENSORLOWRANKBICRITERIACUBICRANK\((A, n, k)\) \hspace{1cm} \(\triangleright \) Theorem C.7
2: \(s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \text{O}(k/\epsilon)\).
3: \(t_1 \leftarrow t_2 \leftarrow t_3 \leftarrow \text{poly}(k/\epsilon)\).
4: Choose \(S_i \in \mathbb{R}^{n^2 \times s_i}\) to be a Sketching matrix, \(\forall i \in [3]\). \hspace{1cm} \(\triangleright \) Definition B.18
5: Choose \(T_i \in \mathbb{R}^{t_i \times n}\) to be a Sketching matrix, \(\forall i \in [3]\). \hspace{1cm} \(\triangleright \) Definition B.16
6: Compute \(U \leftarrow T_1 \cdot (A_1 \cdot S_1), V \leftarrow T_2 \cdot (A_2 \cdot S_2), W \leftarrow T_3 \cdot (A_3 \cdot S_3)\).
7: Compute \(C \leftarrow A(T_1, T_2, T_3)\).
8: \(X \leftarrow \text{FTENSORREGRESSION}(C, U, V, W, t_1, s_1, t_2, s_2, t_3, s_3)\). \hspace{1cm} \(\triangleright \) Linear regression
9: return \(X(A_1S_1, A_2S_2, A_3S_3)\).
10: end procedure

**Theorem C.7.** Given a 3rd order tensor \(A \in \mathbb{R}^{n \times n \times n}\), for any \(k \geq 1, \epsilon \in (0, 1)\), let \(r = O(k^3/\epsilon^3)\). There exists an algorithm that takes \(O(\text{nnz}(A) + n \cdot \text{poly}(k, 1/\epsilon))\) time and outputs three matrices \(U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{n \times r}, W \in \mathbb{R}^{n \times r}\) such that

\[\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}_k A_k} \|A_k - A\|_F^2.\]

holds with probability 9/10.
Theorem C.8. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1, \epsilon \in (0, 1)$, let $r = O(k^2/\epsilon^2)$. There exists an algorithm that takes $O(n \text{nnz}(A) + n \text{poly}(k, 1/\epsilon))$ time and outputs three matrices $U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{n \times r}, W \in \mathbb{R}^{n \times r}$ such that

$$\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank} - k A_k} \| A_k - A \|_F^2$$

holds with probability $9/10$.

Note that there are two different ways to implement algorithm $\text{FTensorLowRankBicriteriaQuadraticRank}$. We present the proofs for both of them here.

Approach I.
Proof. Let $\text{OPT} = \min_{\text{rank} - k < A_k} \|A_k - A\|_F^2$. According to Theorem C.1, we know that there exists a sketching matrix $S_3 \in \mathbb{R}^{n^2 \times s_3}$ where $s_3 = O(k/\epsilon)$, such that

$$\min_{X_1 \in \mathbb{R}^{s_1 \times k}, X_2 \in \mathbb{R}^{s_2 \times k}, X_3 \in \mathbb{R}^{s_3 \times k}} \left\| \sum_{l=1}^{k} (A_1 S_1 X_1)_l \otimes (A_2 S_2 X_2)_l \otimes (A_3 S_3 X_3)_l - A \right\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

Now we fix an $l$ and we have:

$$(A_1 S_1 X_1)_l \otimes (A_2 S_2 X_2)_l \otimes (A_3 S_3 X_3)_l$$

$$= \left( \sum_{i=1}^{s_1} (A_1 S_1)_i (X_1)_{i,l} \right) \otimes \left( \sum_{j=1}^{s_2} (A_2 S_2)_j (X_2)_{j,l} \right) \otimes (A_3 S_3 X_3)_l$$

$$= \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} (A_1 S_1)_i \otimes (A_2 S_2)_j \otimes (A_3 S_3 X_3)_l (X_1)_{i,l} (X_2)_{j,l}$$

Thus, we have

$$\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} (A_1 S_1)_i \otimes (A_2 S_2)_j \otimes \left( \sum_{l=1}^{k} (A_3 S_3 X_3)_l (X_1)_{i,l} (X_2)_{j,l} \right) - A \right\|_F^2 \leq (1 + \epsilon) \text{OPT}. \quad (7)$$

We use matrices $A_1 S_1 \in \mathbb{R}^{n \times s_1}$ and $A_2 S_2 \in \mathbb{R}^{n \times s_2}$ to construct a matrix $B \in \mathbb{R}^{s_1 s_2 \times n^2}$ in the following way: each row of $B$ is the vector corresponding to the matrix generated by the $\otimes$ product between one column vector in $A_1 S_1$ and the other column vector in $A_2 S_2$, i.e.,

$$B_{i+(j-1)s_1} = \text{vec}((A_1 S_1)_i \otimes (A_2 S_2)_j), \forall i \in [s_1], j \in [s_2], \quad (8)$$

where $(A_1 S_1)_i$ denotes the $i$-th column of $A_1 S_1$ and $(A_2 S_2)_j$ denote the $j$-th column of $A_2 S_2$.

We create matrix $\hat{U} \in \mathbb{R}^{n \times s_1 s_2}$ by copying matrix $A_1 S_1$ $s_2$ times, i.e.,

$$\hat{U} = \begin{bmatrix} A_1 S_1 & A_1 S_1 & \cdots & A_1 S_1 \end{bmatrix}. \quad (9)$$

We create matrix $\hat{V} \in \mathbb{R}^{n \times s_1 s_2}$ by copying the $i$-th column of $A_2 S_2$ a total of $s_1$ times, into columns $(i-1)s_1, \cdots, is_1$ of $\hat{V}$, for each $i \in [s_2]$, i.e.,

$$\hat{V} = \begin{bmatrix} (A_2 S_2)_1 & \cdots & (A_2 S_2)_1 & (A_2 S_2)_2 & \cdots & (A_2 S_2)_2 & \cdots & (A_2 S_2)_s_2 & \cdots & (A_2 S_2)_s_2 \end{bmatrix}. \quad (10)$$

Thus, we can use $\hat{U}$ and $\hat{V}$ to represent $B$,

$$B = (\hat{U}^T \otimes \hat{V}^T) \in \mathbb{R}^{s_1 s_2 \times n^2}. $$

According to Equation (7), we have:

$$\min_{W \in \mathbb{R}^{n \times s_1 s_2}} \|WB - A_3\|_F^2 \leq (1 + \epsilon) \text{OPT}. $$

Next, we want to find matrix $W \in \mathbb{R}^{n \times s_1 s_2}$ by solving the following optimization problem,

$$\min_{W \in \mathbb{R}^{n \times s_1 s_2}} \|WB - A_3\|_F^2. $$
Note that $B$ has size $s_1s_2 \times n^2$. Naïvely writing down $B$ already requires $\Omega(n^2)$ time. In order to achieve nearly linear time in $n$, we cannot write down $B$. We choose $S_3 \in \mathbb{R}^{n_1n_2 \times s_3}$ to be a TENSORSKETCH (Definition B.34). In order to solve multiple regression, we need to set $s_3 = O((s_1s_2)^2 + (s_1s_2)/\epsilon)$. Let $\hat{W}$ denote the optimal solution to $\|WBS_3 - A_3S_3\|^2_F$. Then $\hat{W} = (A_3S_3)(BS_3)\dagger$. Since each row of $S_3$ has exactly 1 nonzero entry, $A_3S_3$ can be computed in $O(\text{nnz}(A))$ time. Since $B = (\hat{U}^\top \odot \hat{V}^\top)$, according to Definition B.34, $BS_3$ can be computed in $n\ poly(s_1s_2/\epsilon) = n\ poly(k/\epsilon)$ time. By Theorem C.4, we have

$$\|\hat{W}B - A_3\|^2_F \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times 1s_2}} \|WB - A_3\|^2_F.$$ 

Thus, we have

$$\|\hat{U} \odot \hat{V} \odot \hat{W} - A\|^2_F \leq (1 + \epsilon) \text{OPT}.$$ 

According to Definition B.18, $A_1S_1, A_2S_2$ can be computed in $O(\text{nnz}(A) + \text{poly}(k/\epsilon))$ time. The total running time is thus $O(\text{nnz}(A) + \text{poly}(k/\epsilon))$. □

**Approach II.**

**Proof.** Let $\text{OPT} = \min_{\text{rank} - k} \|A_k - A\|^2_F$. Choose sketching matrices (Definition B.18) $S_1 \in \mathbb{R}^{n_2 \times s_1}$, $S_2 \in \mathbb{R}^{n_2 \times s_2}$, $S_3 \in \mathbb{R}^{n_2 \times s_3}$, and sketching matrices (Definition B.16) $T_1 \in \mathbb{R}^{t_1 \times n}$ and $T_2 \in \mathbb{R}^{t_2 \times n}$ with $s_1 = s_2 = s_3 = O(k/\epsilon), t_1 = t_2 = \text{poly}(k/\epsilon)$. We create matrix $\hat{U} \in \mathbb{R}^{n \times 1s_2}$ by copying matrix $A_1S_1$ $s_2$ times, i.e.,

$$\hat{U} = [A_1S_1 \ A_1S_1 \ \cdots \ A_1S_1].$$ 

We create matrix $\hat{V} \in \mathbb{R}^{n \times 1s_2}$ by copying the $i$-th column of $A_2S_2$ a total of $s_1$ times, into columns $(i - 1)s_1, \cdots, is_1$ of $\hat{V}$, for each $i \in [s_2]$, i.e.,

$$\hat{V} = [(A_2S_2)_1 \ \cdots \ (A_2S_2)_1 \ (A_2S_2)_2 \ \cdots \ (A_2S_2)_2 \ \cdots \ (A_2S_2)_s_2 \ \cdots \ (A_2S_2)_s_2].$$ 

As we proved in Approach I, we have

$$\min_{X \in \mathbb{R}^{n \times 1s_2}} \|\hat{U} \odot \hat{V} \odot X - A\|^2_F \leq (1 + \epsilon) \text{OPT}.$$ 

Let $B = ((T_1\hat{U})^\top \odot (T_2\hat{V})^\top) \in \mathbb{R}^{s_1s_2 \times t_1t_2}$, and flatten $A(T_1, T_2, I)$ along the third direction to obtain $C_3 \in \mathbb{R}^{n \times t_1t_2}$. Let

$$\hat{W} = \arg \min_{X \in \mathbb{R}^{n \times 1s_2}} ||T_1\hat{U} \odot T_2\hat{V} \odot X - A(T_1, T_2, I)||^2_F = \arg \min_{X \in \mathbb{R}^{n \times 1s_2}} ||XB - C_3||^2_F.$$ 

Let

$$W^* = \arg \min_{X \in \mathbb{R}^{n \times 1s_2}} ||\hat{U} \odot \hat{V} \odot X - A||^2_F.$$ 

43
According to Lemma C.6,
\[ \|\hat{U} \otimes \hat{V} \otimes \hat{W} - A\|_F^2 \leq \frac{1}{1 - \epsilon} \|T_1 \hat{U} \otimes T_2 \hat{V} \otimes \hat{W} - A(T_1, T_2, I)\|_F^2 \]
\[ \leq \frac{1}{1 - \epsilon} \|T_1 \hat{U} \otimes T_2 \hat{V} \otimes W^* - A(T_1, T_2, I)\|_F^2 \]
\[ \leq \frac{1}{1 - \epsilon} + \frac{1}{1 - \epsilon} \|\hat{U} \otimes \hat{V} \otimes W^* - A\|_F^2 \]
\[ \leq \left( \frac{1 + \epsilon}{1 - \epsilon} \right)^2 \text{OPT}. \]

According to Definition B.18, \( A_1 S_1, A_2 S_2 \) can be computed in \( O(\text{nnz}(A) + \text{poly}(k/\epsilon)) \) time. The total running time is thus \( O(\text{nnz}(A) + \text{poly}(k/\epsilon)) \). Since \( T_1, T_2 \) are sparse embedding matrices, \( T_1 \hat{U}, T_2 \hat{V} \) can be computed in \( O(\text{nnz}(A) + \text{poly}(k/\epsilon)) \) time. The total running time is in \( O(\text{nnz}(A) + \text{poly}(k/\epsilon)) \).

**Theorem C.9.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \) and any \( 0 < \epsilon < 1 \), if \( A_k \) exists then there is a randomized algorithm running in \( \text{nnz}(A) + \text{poly}(k/\epsilon) \) time which outputs a rank-\( O(k^2/\epsilon^2) \) tensor \( B \) for which \( \|A - B\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2 \). If \( A_k \) does not exist, then the algorithm outputs a rank-\( O(k^2/\epsilon^2) \) tensor \( B \) for which \( \|A - B\|_F^2 \leq (1 + \epsilon)\text{OPT} + \gamma \), where \( \gamma \) is an arbitrarily small positive function of \( n \). In both cases, the algorithm succeeds with probability at least 9/10.

**Proof.** If \( A_k \) exists, then the proof directly follows the proof of Theorem C.1 and Theorem C.8. If \( A_k \) does not exist, then for any \( \gamma > 0 \), there exist \( U^* \in \mathbb{R}^{n \times k}, V^* \in \mathbb{R}^{n \times k}, W^* \in \mathbb{R}^{n \times k} \) such that
\[ \|U^* \otimes V^* \otimes W^* - A\|_F^2 \leq \inf_{\text{rank} - k \ A'} \|A - A'\|_F^2 + \frac{1}{10} \gamma. \]
Then we just regard \( U^* \otimes V^* \otimes W^* \) as the “best” rank \( k \) approximation to \( A \), and follow the same argument as in the proof of Theorem C.1 and the proof of Theorem C.8. We can finally output a tensor \( B \in \mathbb{R}^{n \times n \times n} \) with rank-\( O(k^2/\epsilon^2) \) such that
\[ \|B - A\|_F^2 \leq (1 + \epsilon)\|U^* \otimes V^* \otimes W^* - A\|_F^2 \]
\[ \leq (1 + \epsilon) \left( \inf_{\text{rank} - k \ A'} \|A - A'\|_F^2 + \frac{1}{10} \gamma \right) \]
\[ \leq (1 + \epsilon) \inf_{\text{rank} - k \ A'} \|A - A'\|_F^2 + \gamma \]
where the first inequality follows by the proof of Theorem C.1 and the proof of theorem C.8. The second inequality follows by our choice of \( U^*, V^*, W^* \). The third inequality follows since \( 1 + \epsilon < 2 \) and \( \gamma > 0 \).

**C.4.3** \( \text{poly}(k) \)-approximation to multiple regression

**Lemma C.10** ((1.4) and (1.9) in [RV09]). Let \( s \geq k \). Let \( U \in \mathbb{R}^{n \times k} \) denote a matrix that has orthonormal columns, and \( S \in \mathbb{R}^{s \times n} \) denote an i.i.d. \( N(0, 1/s) \) Gaussian matrix. Then \( SU \) is also an \( s \times k \) i.i.d. Gaussian matrix with each entry draw from \( N(0, 1/s) \), and furthermore, we have with arbitrarily large constant probability,
\[ \sigma_{\text{max}}(SU) = O(1) \quad \text{and} \quad \sigma_{\text{min}}(SU) = \Omega(1/\sqrt{s}). \]
Proof. Note that \( \sqrt{s} - \sqrt{k} - 1 = \frac{s-k-1}{\sqrt{s}+\sqrt{k}-1} = \Omega(1/\sqrt{s}) \).

Lemma C.11. Given matrices \( A \in \mathbb{R}^{n \times k} \), \( B \in \mathbb{R}^{n \times d} \), let \( S \in \mathbb{R}^{s \times n} \) denote a standard Gaussian \( N(0, 1) \) matrix with \( s = k \). Let \( X^* = \min_{X \in \mathbb{R}^{k \times d}} \|AX - B\|_F \). Let \( X' = \min_{X \in \mathbb{R}^{k \times d}} \|SAX - SB\|_F \). Then, we have that

\[
\|AX' - B\|_F \leq O(\sqrt{k})\|AX^* - B\|_F,
\]

holds with probability at least 0.99.

Proof. Let \( X^* \in \mathbb{R}^{k \times d} \) denote the optimal solution such that

\[
\|AX^* - B\|_F = \min_{X \in \mathbb{R}^{k \times d}} \|AX - B\|_F.
\]

Consider a standard Gaussian matrix \( S \in \mathbb{R}^{k \times n} \) scaled by \( 1/\sqrt{k} \) with exactly \( k \) rows. Then for any \( X \in \mathbb{R}^{k \times d} \), by the triangle inequality, we have

\[
\|SAX - SB\|_F \leq \|SAX - SAX^*\|_F + \|SAX^* - SB\|_F,
\]

and

\[
\|SAX - SB\|_F \geq \|SAX - SAX^*\|_F - \|SAX^* - SB\|_F.
\]

We first show how to bound \( \|SAX - SAX^*\|_F \), and then show how to bound \( \|SAX^* - SB\|_F \).

Note that Lemma C.10 implies the following result,

Claim C.12. For any \( X \in \mathbb{R}^{k \times d} \), with probability 0.999, we have

\[
\frac{1}{\sqrt{k}}\|AX - AX^*\|_F \lesssim \|SAX - SAX^*\|_F \lesssim \|AX - AX^*\|_F.
\]

Proof. First, we can write \( A = UR \in \mathbb{R}^{n \times k} \) where \( U \in \mathbb{R}^{n \times k} \) has orthonormal columns and \( R \in \mathbb{R}^{k \times k} \). It gives,

\[
\|SAX - SAX^*\|_F = \|SU(RX - RX^*)\|_F.
\]

Second, applying Lemma C.10 to \( SU \in \mathbb{R}^{s \times k} \) completes the proof.

Using Markov’s inequality, for any fixed matrix \( AX^* - B \), choosing a Gaussian matrix \( S - B \), we have that

\[
\|SAX^* - SB\|_F = O(\|AX^* - B\|_F^2)
\]

holds with probability at least 0.999. This is equivalent to

\[
\|SAX^* - SB\|_F = O(\|AX^* - B\|_F), \quad (11)
\]

holding with probability at least 0.999.
Let $X' = \arg\min_{X \in \mathbb{R}^{k \times d}} \|SAX - SB\|_F$. Putting it all together, we have
\[
\|AX' - B\|_F \\
\leq \|AX' - AX^*\|_F + \|AX^* - B\|_F \\
\leq O(\sqrt{k})\|SAX' - SAX^*\|_F + \|AX^* - B\|_F \\
\leq O(\sqrt{k})\|SAX' - SB\|_F + O(\sqrt{k})\|SAX^* - SB\|_F + \|AX^* - B\|_F \\
\leq O(\sqrt{k})\|SAX^* - SB\|_F + O(\sqrt{k})\|SAX^* - B\|_F + \|AX^* - B\|_F \\
\leq O(\sqrt{k})\|AX^* - B\|_F.
\]
by triangle inequality
by Claim C.12
by triangle inequality
by definition of $X'$
by Equation (11)

\[\square\]

### C.4.4 Algorithm II

**Theorem C.13.** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, let $r = k^2$. There exists an algorithm which takes $O(\text{nnz}(A)k) + n\text{ poly}(k)$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times r}$ such that,
\[
\left\| \sum_{i=1}^r U_i \otimes V_i \otimes W_i - A \right\|_F \leq \text{poly}(k) \min_{\|A' - A\|_F} \text{rank}_{-k} A' - A\|_F
\]
holds with probability 9/10.

**Proof.** Let $\text{OPT} = \min_{\|A' - A\|_F} \|A' - A\|_F$, we fix $V^* \in \mathbb{R}^{n \times k}, W^* \in \mathbb{R}^{n \times k}$ to be the optimal solution of the original problem. We use $Z_1 = (V^* \otimes W^* \otimes) \in \mathbb{R}^{k \times n^2}$ to denote the matrix where the $i$-th row is the vectorization of $V_i^* \otimes W_i^*$. Let $A_1 \in \mathbb{R}^{n \times n^2}$ denote the matrix obtained by flattening tensor $A \in \mathbb{R}^{n \times n \times n}$ along the first direction. Then, we have
\[
\min_U \|UZ_1 - A_1\|_F \leq \text{OPT}.
\]
Choosing an $N(0, 1/k)$ Gaussian sketching matrix $S_1 \in \mathbb{R}^{n^2 \times s_1}$ with $s_1 = k$, we can obtain the smaller problem,
\[
\min_{U \in \mathbb{R}^{n \times k}} \|UZ_1S_1 - A_1S_1\|_F.
\]
Define $\hat{U} = A_1S_1(Z_1S_1)^\dagger$. Define $\alpha = O(\sqrt{k})$. By Lemma C.11, we have
\[
\|\hat{U}Z_1 - A_1\|_F \leq \alpha \text{OPT}.
\]
Second, we fix $\hat{U}$ and $W^*$. Define $Z_2, A_2$ similarly as above. Choosing an $N(0, 1/k)$ Gaussian sketching matrix $S_2 \in \mathbb{R}^{n^2 \times s_2}$ with $s_2 = k$, we can obtain another smaller problem,
\[
\min_{V \in \mathbb{R}^{n \times k}} \|VZ_2S_2 - A_2S_2\|_F.
\]
Define $\hat{V} = A_2S_2(Z_2S_2)^\dagger$. By Lemma C.11 again, we have
\[
\|\hat{V}Z_2 - A_2\|_F \leq \alpha^2 \text{OPT}.
\]

46
Thus, we now have
\[
\min_{X_1, X_2, W} \|A_1 S_1 X_1 \otimes A_2 S_2 X_2 \otimes W - A\|_F \leq \alpha^2 \text{OPT}
\]

We use a similar idea as in the proof of Theorem C.8. We create matrix \(\tilde{U} \in \mathbb{R}^{n \times s_1 s_2}\) by copying matrix \(A_1 S_1\) \(s_2\) times, i.e.,
\[
\tilde{U} = [A_1 S_1 \quad A_1 S_1 \quad \ldots \quad A_1 S_1].
\]

We create matrix \(\tilde{V} \in \mathbb{R}^{n \times s_1 s_2}\) by copying the \(i\)-th column of \(A_2 S_2\) a total of \(s_1\) times, into columns \((i-1)s_1, \ldots, is_1\) of \(\tilde{V}\), for each \(i \in [s_2]\), i.e.,
\[
\tilde{V} = [(A_2 S_2)_1 \quad \ldots \quad (A_2 S_2)_1 \quad (A_2 S_2)_2 \quad \ldots \quad (A_2 S_2)_2 \quad \ldots \quad (A_2 S_2)_s_2].
\]

We have
\[
\min_{X \in \mathbb{R}^{n \times s_1 s_2}} \|\tilde{U} \otimes \tilde{V} \otimes X - A\|_F \leq \alpha^2 \text{OPT}.
\]

Choose \(T_i \in \mathbb{R}^{t_i \times n}\) to be a sparse embedding matrix (Definition B.16) with \(t_i = \text{poly}(k/\epsilon)\), for each \(i \in [2]\). By applying Lemma C.6, we have, if \(W'\) satisfies,
\[
\|T_1 \tilde{U} \otimes T_2 \tilde{V} \otimes W' - A(T_1, T_2, I)\|_F = \min_{X \in \mathbb{R}^{n \times s_1 s_2}} \|T_1 \tilde{U} \otimes T_2 \tilde{V} \otimes X - A(T_1, T_2, I)\|_F
\]
then,
\[
\|\tilde{U} \otimes \tilde{V} \otimes W' - A\|_F \leq (1 + \epsilon) \min_{X \in \mathbb{R}^{n \times s_1 s_2}} \|\tilde{U} \otimes \tilde{V} \otimes X - A\|_F \leq (1 + \epsilon) \alpha^2 \text{OPT}.
\]

Thus, we only need to solve
\[
\min_{X \in \mathbb{R}^{n \times s_1 s_2}} \|T_1 \tilde{U} \otimes T_2 \tilde{V} \otimes X - A(T_1, T_2, I)\|_F,
\]
which is similar to the proof of Theorem C.8. Therefore, we complete the proof of correctness. For the running time, \(A_1 S_1, A_2 S_2\) can be computed in \(O(\text{nnz}(A)k)\) time, \(T_1 \tilde{U}, T_2 \tilde{V}\) can be computed in \(n \text{ poly}(k)\) time. The final regression problem can be computed in \(n \text{ poly}(k)\) running time. \(\square\)

### C.5 Generalized matrix row subset selection

Note that in this section, the notation \(\pi_{C, k}^\xi\) is given in Definition B.5.

**Theorem C.14.** Given matrices \(A \in \mathbb{R}^{n \times m}\) and \(C \in \mathbb{R}^{n \times k}\), there exists an algorithm which takes \(O(\text{nnz}(A) \log n) + (m + n) \text{ poly}(k, 1/\epsilon)\) time and outputs a diagonal matrix \(D \in \mathbb{R}^{n \times n}\) with \(d = O(k/\epsilon)\) nonzeros (or equivalently a matrix \(R\) that contains \(d = O(k/\epsilon)\) rescaled rows of \(A\)) and a matrix \(U \in \mathbb{R}^{k \times d}\) such that
\[
\|CU DA - A\|_F^2 \leq (1 + \epsilon) \min_{X \in \mathbb{R}^{k \times m}} \|CX - A\|_F^2
\]
holds with probability \(.99\).
Algorithm 7 Generalized Matrix Row Subset Selection: Constructing $R$ with $r = O(k + k/\epsilon)$ Rows and a rank-$k$ $U \in \mathbb{R}^{k \times r}$

1: procedure GENERALIZEDMATRIXROWSUBSETSELECTION($A, C, n, m, k, \epsilon$) \Comment{Theorem C.14}
2: \hspace{1em} $Y, \Phi, \Delta \leftarrow$ APPROXSUBSPACEVD($A, C, k$). \Comment{Claim C.16 and Lemma 3.12 in [BW14]}
3: \hspace{1em} $B \leftarrow Y \Delta$. \Comment{$Z_2, D \in \mathbb{R}^{m \times k}$, $Z_2^T Z_2 = I_k$, $D \in \mathbb{R}^{k \times k}$}
4: \hspace{1em} $Z_2, D \leftarrow$ QR($B$). \Comment{$Z_2 \in \mathbb{R}^{m \times k}$, $Z_2^T Z_2 = I_k$, $D \in \mathbb{R}^{k \times k}$}
5: \hspace{1em} $h_2 \leftarrow 8k \ln(20k)$. \Comment{$h \in \mathbb{R}$}
6: \hspace{1em} $\Omega_2, D_2 \leftarrow$ RANDSAMPLING($Z_2, h_2, 1$) \Comment{Definition 3.6 in [BW14]}
7: \hspace{1em} $M_2 \leftarrow Z_2^T \Omega_2 D_2 \in \mathbb{R}^{k \times h_2}$. \Comment{$M_2 \in \mathbb{R}^{k \times h_2}$}
8: \hspace{1em} $U_{M_2}, \Sigma_{M_2}, V_{M_2}^T \leftarrow$ SVD($M_2$). \Comment{rank($M_2$) = $k$ and $V_{M_2} \in \mathbb{R}^{h_2 \times k}$}
9: \hspace{1em} $r_1 \leftarrow 4k$. \Comment{$r_1 \in \mathbb{R}$}
10: \hspace{1em} $S_2 \leftarrow$ BSSSAMPLINGSPARSE($V_{M_2}, (A^T - A^T Z_2 Z_2^T) \Omega_2 D_2^T, r_1, 0.5$) \Comment{Lemma 4.3 in [BW14]}
11: \hspace{1em} $R_1 \leftarrow (A^T \Omega_2 D_2 S_2)^T \in \mathbb{R}^{r_1 \times n}$ containing rescaled rows from $A$. \Comment{In [BW14], $R_1 \leftarrow (A^T \Omega_2 D_2 S_2)^T \in \mathbb{R}^{r_1 \times n}$ containing rescaled rows from $A$.}
12: \hspace{1em} $r_2 \leftarrow 4820k/\epsilon$. \Comment{$r_2 \in \mathbb{R}$}
13: \hspace{1em} $R_2 \leftarrow$ ADAPTIVEROWSSPARSE($A, Z_2, R_1, r_2$) \Comment{Lemma 4.5 in [BW14]}
14: \hspace{1em} $R \leftarrow [R_1^T, R_2^T]^T$, \Comment{$R \in \mathbb{R}^{(r_1+r_2) \times n}$ containing $r = 4k + 4820k/\epsilon$ rescaled rows of $A$.}
15: \hspace{1em} Choose $W \in \mathbb{R}^{\xi \times m}$ to be a randomly chosen sparse subspace embedding with $\xi = \Omega(k^2 \epsilon^{-2})$. \Comment{In [BW14], $W \in \mathbb{R}^{\xi \times m}$ to be a randomly chosen sparse subspace embedding with $\xi = \Omega(k^2 \epsilon^{-2})$.}
16: \hspace{1em} $U \leftarrow \Phi^{-1} \Delta D^{-1} (WC \Phi^{-1} \Delta D^{-1})^T \omega_{AR} = \Phi^{-1} \Delta D^{-1} (WC)^T \omega_{AR}$. \Comment{$U \in \mathbb{R}^{m \times k}$}
17: \hspace{1em} return $R, U$.
18: end procedure

Proof. This follows by combining Lemma C.17 and C.18. Let $U, R$ denote the output of procedure GENERALIMATRIXROWSUBSETSELECTION,

$$\|A - CUR\|_F^2 \leq (1 + \epsilon)\|A - Z_2 Z_2^T A R_1^T R_2\|_F^2 \leq (1 + \epsilon)(1 + 60\epsilon)\|A - \Pi_{C,k}^F(A)\|_F^2 \leq (1 + 130\epsilon)\|A - \Pi_{C,k}^F(A)\|_F^2.$$  

Because $R$ is a subset of rows of $A$ and $R$ has size $O(k/\epsilon) \times m$, there must exist a diagonal matrix $D \in \mathbb{R}^{n \times n}$ with $O(k/\epsilon)$ nonzeros such that $R = DA$. This completes the proof.

\[ \square \]

Corollary C.15 (A slightly different version of Theorem C.14, faster running time, and small input matrix). Given matrices $A \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{n \times k}$, if $\min(m, n) = \text{poly}(k, 1/\epsilon)$, then there exists an algorithm which takes $O(\text{nnz}(A)) + (m + n)\text{poly}(k, 1/\epsilon)$ time and outputs a diagonal matrix $D \in \mathbb{R}^{n \times n}$ with $d = O(k/\epsilon)$ nonzeros (or equivalently a matrix $R$ that contains $d = O(k/\epsilon)$ rescaled rows of $A$) and a matrix $U \in \mathbb{R}^{k \times d}$ such that

$$\|CUDA - A\|_F^2 \leq (1 + \epsilon) \min_{X \in \mathbb{R}^{k \times m}} \|CX - A\|_F^2$$

holds with probability .99.

Proof. The log $n$ factor comes from the adaptive sampling where we need to choose a Gaussian matrix with $O(\log n)$ rows and compute $SA$. If $A$ has $\text{poly}(k, 1/\epsilon)$ columns, it is sufficient to choose $S$ to be a CountSketch matrix with $\text{poly}(k, 1/\epsilon)$ rows. Then, we do not need a log $n$ factor in the running time. If $S$ has $\text{poly}(k, 1/\epsilon)$ rows, then we no longer need the matrix $S$.  

\[ \square \]
Claim C.16. Given matrices $A \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{m \times c}$, let $Y \in \mathbb{R}^{m \times c}, \Phi \in \mathbb{R}^{c \times c}$ and $\Delta \in \mathbb{R}^{c \times k}$ denote the output of procedure $\text{APPROXSUBSPACESVD}(A, C, k, \epsilon)$. Then with probability .99, we have,

$$\|A - Y \Delta \Delta^\top Y^\top A\|_F^2 \leq (1 + 30\epsilon)\|A - \Pi^F_{C,k}(A)\|_F^2.$$  

Proof. This follows by Lemma 3.12 in [BW14].

Lemma C.17. The matrices $R$ and $Z_2$ in procedure $\text{GENERALIZEDMATRIXROWSUBSETSELECTION}$ (Algorithm 7) satisfy with probability at least $0.17 - 2/n$,

$$\|A - Z_2Z_2^\top AR^\dagger R\|_F^2 \leq \|A - \Pi^F_{C,k}(A)\|_F^2 + 60\epsilon\|A - \Pi^F_{C,k}(A)\|_F^2.$$  

Proof. We can show,

$$\|A - Z_2Z_2^\top A\|_F^2 + \frac{30\epsilon}{4820}\|A - AR^\dagger_1 R_1\|_F^2$$

$$= \|A - BB^\dagger A\|_F^2 + \frac{30\epsilon}{4820}\|A - AR^\dagger_1 R_1\|_F^2$$

$$\leq \|A - BB^\dagger A\|_F^2 + 30\epsilon\|A - A_k\|_F^2$$

$$\leq \|A - Y \Delta \Delta^\top Y A\|_F^2 + 30\epsilon\|A - \Pi^F_{C,k}(A)\|_F^2$$

$$\leq (1 + 30\epsilon)\|A - \Pi^F_{C,k}(A)\|_F^2 + 30\epsilon\|A - \Pi^F_{C,k}(A)\|_F^2,$$

where the first step follows by the fact that $Z_2Z_2^\top = Z_2DD^{-1}Z_2^\top = (Z_2D)(Z_2D)^\dagger = BB^\dagger$, the second step follows by $\|A - AR^\dagger_1 R_1\|_F \leq 4820\|A - A_k\|_F^2$, the third step follows by $B = Y \Delta$ and $B^\dagger = (Y \Delta)^\dagger = \Delta^\dagger Y^\top = \Delta^\top Y^\top$, and the last step follows by Claim C.16.

Lemma C.18. The matrices $C, U$ and $R$ in procedure $\text{GENERALIZEDMATRIXROWSUBSETSELECTION}$ (Algorithm 7) satisfy that

$$\|A - CUR\|_F^2 \leq (1 + \epsilon)\|A - Z_2Z_2^\top AR^\dagger R\|_F^2$$

with probability at least .99.

Proof. Let $U_R, \Sigma_R, V_R$ denote the SVD of $R$. Then $V_RV_R^\top = R^\dagger R$.

We define $Y^\star$ to be the optimal solution of

$$\min_{X \in \mathbb{R}^{k \times r}} \|WAV_RV_R^\top - WC\Phi^{-1}\Delta D^{-1} Y R\|_F^2.$$  

We define $\hat{X}^\star$ to be $Y^\star R \in \mathbb{R}^{k \times n}$, which is also equivalent to defining $\hat{X}^\star$ to be the optimal solution of

$$\min_{X \in \mathbb{R}^{k \times n}} \|WAV_RV_R^\top - WC\Phi^{-1}\Delta D^{-1} X\|_F^2.$$  

Furthermore, it implies $\hat{X}^\star = (WC\Phi^{-1}\Delta D^{-1})^\dagger WAV_RV_R^\dagger$.

We also define $X^\star$ to be the optimal solution of

$$\min_{X \in \mathbb{R}^{k \times n}} \|AV_RV_R^\top - C\Phi^{-1}\Delta D^{-1} X\|_F^2,$$
which implies that,

\[ X^* = (C\Phi^{-1}\Delta D^{-1})^\dagger AV_RV_R^T = Z_2^\top AV_RV_R^T. \]

Now, we start to prove an upper bound on \( \|A - CUR\|_F^2 \),

\[
\|A - CUR\|_F^2 = \|A - C\Phi^{-1}\Delta D^{-1}Y^* R\|_F^2 \quad \text{by definition of } U
\]
\[
= \|A - C\Phi^{-1}\Delta D^{-1}\hat{X}^*\|_F^2 \quad \text{by } \hat{X}^* = Y^* R
\]
\[
= \|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}\hat{X}^* + A - AV_RV_R^T\|_F^2
\]
\[
= \left(\frac{\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}\hat{X}^*\|_F^2}{\alpha} + \frac{\|A - AV_RV_R^T\|_F^2}{\beta}\right)
\]
\[
(12)
\]

where the last step follows by \( \hat{X}^* = MV_R^\top, A - AV_RV_R^T = A(I - V_RV_R^T) \) and the Pythagorean theorem. We show how to upper bound the term \( \alpha \),

\[
\alpha \leq (1 + \epsilon)\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2 \quad \text{by Lemma C.19}
\]
\[
= \epsilon\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2 + \|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2
\]
\[
= \epsilon\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2 + \|A - AV_RV_R^T\|_F^2
\]
\[
(13)
\]

By the Pythagorean theorem and the definition of \( Z_2 \) (which means \( Z_2 = C\Phi^{-1}\Delta D^{-1} \)), we have,

\[
\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}(Z_2^\top AR^\dagger R)\|_F^2 + \beta
\]
\[
= \|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}(Z_2^\top AR^\dagger R)\|_F^2 + \|A - AV_RV_R^T\|_F^2
\]
\[
= \|A - Z_2Z_2^\top AR^\dagger R\|_F^2
\]
\[
(14)
\]

Combining Equations (12), (13) and (14) together, we obtain,

\[
\|A - CUR\|_F^2 \leq \epsilon\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2 + \|A - Z_2Z_2^\top AR^\dagger R\|_F^2.
\]

We want to show \( \|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2 \leq \|A - Z_2Z_2^\top AR^\dagger R\|_F^2 \),

\[
\|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}X^*\|_F^2
\]
\[
= \|AV_RV_R^T - C\Phi^{-1}\Delta D^{-1}Z_2^\top AV_RV_R^T\|_F^2 \quad \text{by } X^* = Z_2^\top AV_RV_R^T
\]
\[
\leq \|A - C\Phi^{-1}\Delta D^{-1}Z_2^\top A\|_F^2 \quad \text{by properties of projections}
\]
\[
\leq \|A - C\Phi^{-1}\Delta D^{-1}Z_2^\top AR^\dagger R\|_F^2 \quad \text{by properties of projections}
\]
\[
= \|A - Z_2Z_2^\top AR^\dagger R\|_F^2 \quad \text{by } Z_2 = C\Phi^{-1}\Delta D^{-1}
\]

This completes the proof. \( \square \)

**Lemma C.19 ([CW13]).** Let \( A \in \mathbb{R}^{n \times d} \) have rank \( \rho \) and \( B \in \mathbb{R}^{n \times r} \). Let \( W \in \mathbb{R}^{r \times n} \) be a randomly chosen sparse subspace embedding with \( r = \Omega(\rho^2 \epsilon^{-2}) \). Let \( \bar{X}^* = \arg \min_{X \in \mathbb{R}^{d \times r}} \|WAX - WB\|_F^2 \) and let

\[
X^* = \arg \min_{X \in \mathbb{R}^{d \times r}} \|AX - B\|_F^2.
\]

Then with probability at least .99,

\[
\|A\bar{X}^* - B\|_F^2 \leq (1 + \epsilon)\|AX^* - B\|_F^2.
\]
Algorithm 8: Frobenius Norm Tensor Column, Row and Tube Subset Selection, Polynomial Time

1: procedure FCRTSelection($A, n, k, \epsilon$) \hfill $\triangleright$ Theorem C.20
2: \hspace{1em} $s_1 \leftarrow s_2 \leftarrow O(k/\epsilon)$.
3: \hspace{1em} Choose a Gaussian matrix $S_1$ with $s_1$ columns. \hfill $\triangleright$ Definition B.18
4: \hspace{1em} Choose a Gaussian matrix $S_2$ with $s_2$ columns. \hfill $\triangleright$ Definition B.18
5: \hspace{1em} Form matrix $Z_3'$ by setting the $(i,j)$-th row to be the vectorization of $(A_1S_1)_i \otimes (A_2S_2)_j$.
6: \hspace{1em} $D_3 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_3^\top, (Z_3')^\top, n^2, n, s_1s_2, \epsilon)$. \hfill $\triangleright$ Algorithm 9
7: \hspace{1em} Let $d_3$ denote the number of nonzero entries in $D_3$. \hfill $d_3 = O(s_1s_2/\epsilon)$
8: \hspace{1em} Form matrix $Z_2'$ by setting the $(i,j)$-th row to be the vectorization of $(A_1S_1)_i \otimes (A_3S_3)_j$.
9: \hspace{1em} $D_2 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_2^\top, (Z_2')^\top, n^2, n, s_1d_3, \epsilon)$.
10: \hspace{1em} Let $d_2$ denote the number of nonzero entries in $D_2$. \hfill $d_2 = O(s_1d_3/\epsilon)$
11: \hspace{1em} Form matrix $Z_1'$ by setting the $(i,j)$-th row to be the vectorization of $(A_2D_2)_i \otimes (A_3D_3)_j$.
12: \hspace{1em} $D_1 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_1^\top, (Z_1')^\top, n^2, n, d_2d_3, \epsilon)$.
13: \hspace{1em} Let $d_1$ denote the number of nonzero entries in $D_1$. \hfill $d_1 = O(d_2d_3/\epsilon)$
14: \hspace{1em} $C \leftarrow A_1D_1$, $R \leftarrow A_2D_2$ and $T \leftarrow A_3D_3$.
15: \hspace{1em} return $C$, $R$ and $T$.
16: end procedure

C.6 Column, row, and tube subset selection, $(1+\epsilon)$-approximation

We provide two bicriteria CURT results in this Section. We first present a warm-up result. That result (Theorem C.20) does not output tensor $U$ and only guarantees that there is a rank-poly$(k/\epsilon)$ tensor $U$. Then we show the second result (Theorem C.21), our second result is able to output tensor $U$. The $U$ has rank poly$(k/\epsilon)$, but not $k$.

**Theorem C.20.** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(\text{nz}(A)) + n \text{poly}(k, 1/\epsilon)$ time and outputs three matrices: $C \in \mathbb{R}^{n \times c}$, a subset of columns of $A$, $R \in \mathbb{R}^{n \times r}$ a subset of rows of $A$, and $T \in \mathbb{R}^{n \times t}$, a subset of tubes of $A$ where $c = r = t = \text{poly}(k, 1/\epsilon)$, and there exists a tensor $U \in \mathbb{R}^{c \times r \times t}$ such that

$$\|((U \cdot T^\top)^\top \cdot R^\top)^\top \cdot C^\top - A\|_F^2 \leq (1+\epsilon) \min_{\text{rank} - k} \|A_k - A\|_F^2,$$

or equivalently,

$$\left\| \sum_{j = 1}^{c} \sum_{i = 1}^{r} \sum_{l = 1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_F^2 \leq (1+\epsilon) \min_{\text{rank} - k} \|A_k - A\|_F^2,$$

holds with probability $9/10$.

**Proof.** We mainly analyze Algorithm 8 and it is easy to extend to Algorithm 9.

We fix $V^* \in \mathbb{R}^{n \times k}$ and $W^* \in \mathbb{R}^{n \times k}$. We define $Z_1 \in \mathbb{R}^{k \times n^2}$ where the $i$-th row of $Z_1$ is the vector $V_i \otimes W_i$. Choose sketching (Gaussian) matrix $S_1 \in \mathbb{R}^{n^2 \times s_1}$ (Definition B.18), and let $\hat{U} = A_1S_1(Z_1S_1)^\top \in \mathbb{R}^{n \times k}$. Following a similar argument as in the previous theorem, we have

$$\|\hat{U}Z_1 - A_1\|_F^2 \leq (1+\epsilon) \text{OPT}.$$
We fix $\tilde{U}$ and $W^*$. We define $Z_2 \in \mathbb{R}^{k \times n^2}$ where the $i$-th row of $Z_2$ is the vector $\tilde{U}_i \otimes W^*_i$. Choose sketching (Gaussian) matrix $S_2 \in \mathbb{R}^{n^2 \times s_2}$ (Definition B.18), and let $\hat{V} = A_2 S_2 (Z_2 S_2)^\dagger \in \mathbb{R}^{n \times k}$. Following a similar argument as in the previous theorem, we have

$$\|\hat{V} Z_2 - A_2\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.$$  

We fix $\tilde{U}$ and $\hat{V}$. Note that $\tilde{U} = A_1 S_1 (Z_1 S_1)^\dagger$ and $\hat{V} = A_2 S_2 (Z_2 S_2)^\dagger$. We define $Z_3 \in \mathbb{R}^{k \times n^2}$ such that the $i$-th row of $Z_3$ is the vector $\tilde{U}_i \otimes \hat{V}_i$. Let $z_3 = s_1 \cdot s_2$. We define $Z'_3 \in \mathbb{R}^{z_3 \times n^2}$ such that, $\forall i \in [s_1], \forall j \in [s_2]$, the $i + (j-1)s_1$-th row of $Z'_3$ is the vector $(A_1 S_1)_i \otimes (A_2 S_2)_j$. We consider the following objective function,

$$\min_{W \in \mathbb{R}^{n \times k},X \in \mathbb{R}^{k \times 3}} \|WXZ'_3 - A_3\|_F^2 \leq \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.$$  

Using Theorem C.14, we can find a diagonal matrix $D_3 \in \mathbb{R}^{n^2 \times n^2}$ with $d_3 = O(z_3/\epsilon) = O(k^2/\epsilon^3)$ nonzero entries such that

$$\min_{X \in \mathbb{R}^{d_3 \times 3}} \|A_3 D_3 X Z'_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.$$  

In the following, we abuse notation and let $A_3 D_3 \in \mathbb{R}^{n \times d_3}$ by deleting zero columns. Let $W'$ denote $A_3 D_3 \in \mathbb{R}^{n \times d_3}$. Then,

$$\min_{X \in \mathbb{R}^{d_3 \times 3}} \|W' X Z'_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.$$  

We fix $\tilde{U}$ and $W'$. Let $z_2 = s_1 \cdot d_3$. We define $Z'_2 \in \mathbb{R}^{z_2 \times n^2}$ such that, $\forall i \in [s_1], \forall j \in [d_3]$, the $i + (j-1)s_1$-th row of $Z'_2$ is the vector $(A_1 S_1)_i \otimes (A_3 D_3)_j$. Using Theorem C.14, we can find a diagonal matrix $D_2 \in \mathbb{R}^{n^2 \times n^2}$ with $d_2 = O(z_2/\epsilon) = O(s_1 d_3/\epsilon) = O(k^3/\epsilon^5)$ nonzero entries such that

$$\min_{X \in \mathbb{R}^{d_2 \times z_2}} \|A_2 D_2 X Z'_2 - A_2\|_F^2 \leq (1 + \epsilon)^4 \text{OPT}.$$  

Let $V'$ denote $A_2 D_2$. Then,

$$\min_{X \in \mathbb{R}^{d_2 \times z_2}} \|V' X Z'_2 - A_2\|_F^2 \leq (1 + \epsilon)^4 \text{OPT}.$$  

We fix $V'$ and $W'$. Let $z_1 = d_2 \cdot d_3$. We define $Z'_1 \in \mathbb{R}^{z_1 \times n^2}$ such that, $\forall i \in [d_2], \forall j \in [d_3]$, the $i + (j-1)s_1$-th row of $Z'_1$ is the vector $(A_2 D_2)_i \otimes (A_3 D_3)_j$. Using Theorem C.14, we can find a diagonal matrix $D_1 \in \mathbb{R}^{n^2 \times n^2}$ with $d_1 = O(z_1/\epsilon) = O(d_2 d_3/\epsilon) = O(k^5/\epsilon^9)$ nonzero entries such that

$$\min_{X \in \mathbb{R}^{d_1 \times z_1}} \|A_1 D_1 X Z'_1 - A_1\|_F^2 \leq (1 + \epsilon)^5 \text{OPT}.$$  

Let $U'$ denote $A_1 D_1$. Then,

$$\min_{X \in \mathbb{R}^{d_1 \times z_1}} \|U' X Z'_1 - A_1\|_F^2 \leq (1 + \epsilon)^5 \text{OPT}.$$  

Putting $U', V', W'$ all together, we complete the proof.

All the above analysis gives the running time $O(\text{nnz}(A)) \log n + n^2 \text{poly}(\log n, k, 1/\epsilon)$. To improve the running time, we need to use Algorithm 9, the similar analysis will go through, the running time will be improved to $O(\text{nnz}(A) + n \text{poly}(k, 1/\epsilon))$, but the sample complexity of $c, r, k$ will be slightly worse (poly log factors).
Algorithm 9: Frobenius Norm Tensor Column, Row and Tube Subset Selection, Input Sparsity Time

1: procedure FCRTSelection($A, n, k, \epsilon$) \Comment{Theorem C.20}
2: \hspace{1em} $s_1 \leftarrow s_2 \leftarrow O(k/\epsilon)$. 
3: \hspace{1em} $\epsilon_0 \leftarrow 0.001$.
4: \hspace{1em} Choose a Gaussian matrix $S_1$ with $s_1$ columns. \Comment{Definition B.18}
5: \hspace{1em} Choose a Gaussian matrix $S_2$ with $s_2$ columns. \Comment{Definition B.18}
6: \hspace{1em} Form matrix $B_1$ by setting $(i,j)$-th column to be $(A_1S_1)_i$. 
7: \hspace{1em} Form matrix $B_2$ by setting $(i,j)$-th column to be $(A_2S_2)_j$. \Comment{$Z'_3 = B_1^T \odot B_2^T$}
8: \hspace{1em} $d_3 \leftarrow O(s_1s_2 \log(s_1s_2) + (s_1s_2/\epsilon))$.
9: \hspace{1em} $D_3 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(B_1^T, B_2^T, n, n, s_1s_2, \epsilon_0, d_1)$. \Comment{Algorithm 15}
10: \hspace{1em} Form matrix $B_1$ by setting $(i,j)$-th column to be $(A_1S_1)_i$. 
11: \hspace{1em} Form matrix $B_3$ by setting $(i,j)$-th column to be $(A_3D_3)_j$. \Comment{$Z'_2 = B_1^T \odot B_3^T$}
12: \hspace{1em} $d_2 \leftarrow O(s_1d_3 \log(s_1d_3) + (s_1d_3/\epsilon))$.
13: \hspace{1em} $D_2 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(B_1^T, B_3^T, n, n, s_1d_3, \epsilon_0, d_2)$.
14: \hspace{1em} Form matrix $B_2$ by setting $(i,j)$-th column to be $(A_2D_2)_i$. 
15: \hspace{1em} Form matrix $B_3$ by setting $(i,j)$-th column to be $(A_3D_3)_j$. \Comment{$Z'_1 = B_2^T \odot B_3^T$}
16: \hspace{1em} $d_1 \leftarrow O(d_2d_3 \log(d_2d_3) + (d_2d_3/\epsilon))$.
17: \hspace{1em} $D_1 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(B_2^T, B_3^T, n, n, d_2d_3, \epsilon_0, d_1)$.
18: \hspace{1em} $C \leftarrow A_1D_1, R \leftarrow A_2D_2$ and $T \leftarrow A_3D_3$.

Theorem C.21. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(\text{nnz}(A) + n\text{poly}(k,1/\epsilon))$ time and outputs three matrices: $C \in \mathbb{R}^{n \times c}$, a subset of columns of $A$, $R \in \mathbb{R}^{n \times r}$ a subset of rows of $A$, and $T \in \mathbb{R}^{n \times t}$, a subset of tubes of $A$, together with a tensor $U \in \mathbb{R}^{c \times r \times t}$ with rank($U$) = $k'$ where $c = r = t = \text{poly}(k,1/\epsilon)$ and $k' = \text{poly}(k,1/\epsilon)$ such that

$$\|U(C, R, T) - A\|^2_F \leq (1 + \epsilon) \min_{\text{rank} - k} \|A - A\|^2_F,$$

or equivalently,

$$\left| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_{i} \odot R_{j} \odot T_l - A \right|^2_F \leq (1 + \epsilon) \min_{\text{rank} - k} \|A - A\|^2_F.$$

holds with probability $9/10$.

Proof. The proof follows by combining Theorem 1.1 and Theorem 1.3 directly. \qed

C.7 CURT decomposition, $(1 + \epsilon)$-approximation

C.7.1 Properties of leverage score sampling and BSS sampling

Notice that, the BSS algorithm is a deterministic procedure developed in [BSS12] for selecting rows from a matrix $A \in \mathbb{R}^{n \times d}$ (with $\|A\|_2 \leq 1$ and $\|A\|_F^2 \leq k$) using a selection matrix $S$ so that

$$\|A^T S^T S A - A^T A\|_2 \leq \epsilon.$$
The algorithm runs in \( \text{poly}(n, d, 1/\epsilon) \) time. Using the ideas from [BW14] and [CEM+15], we are able to reduce the number of nonzero entries from \( O(\epsilon^{-2}k \log k) \) to \( O(\epsilon^{-2}k) \), and also improve the running time to input sparsity.

**Lemma C.22** (Leverage score preserves subspace embedding - Theorem 2.11 in [Woo14]). Given a rank-\( k \) matrix \( A \in \mathbb{R}^{n \times d} \), via leverage score sampling, we can obtain a diagonal matrix \( D \) with \( m \) nonzero entries such that, letting \( B = DA \), if \( m = O(\epsilon^{-2}k \log k) \), then, with probability at least 0.999, for all \( x \in \mathbb{R}^d \),

\[
(1 - \epsilon) \|Ax\|_2 \leq \|Bx\|_2 \leq (1 + \epsilon) \|Ax\|_2
\]

**Proof.** We choose a sparse embedding matrix (Definition B.16) \( \Pi \in \mathbb{R}^{d \times s} \) with \( s = \text{poly}(k/\epsilon) \). With probability at least 0.999, \( \Pi^\top \) is a subspace embedding of \( A^\top \). Thus, \( \text{rank}(A\Pi) = \text{rank}(A) \). Also, the leverage scores of \( A\Pi \) are the same as those of \( A \). Thus, we can compute the leverage scores of \( A\Pi \). The running time of computing \( A\Pi \) is \( O(\text{nnz}(A)) \). Thus the total running time is \( O(\text{nnz}(A) + n \text{poly}(k, 1/\epsilon)) \). \( \square \)

**Lemma C.23.** Given a rank-\( k \) matrix \( A \in \mathbb{R}^{n \times d} \), there exists an algorithm that runs in \( O(\text{nnz}(A) + n \text{ poly}(k, 1/\epsilon)) \) time and outputs a matrix \( B \) containing \( O(\epsilon^{-2}k \log k) \) re-weighted rows of \( A \), such that, with probability at least 0.999, for all \( x \in \mathbb{R}^d \),

\[
(1 - \epsilon) \|Ax\|_2 \leq \|Bx\|_2 \leq (1 + \epsilon) \|Ax\|_2
\]

**Proof.** Using Lemma C.23, we can obtain \( B \). Then we apply a sparse subspace embedding matrix \( \Pi \) on the right of \( B \). At the end, we run the BSS algorithm on \( B\Pi \) and we are able to output \( O(\epsilon^{-2}k) \) re-weighted rows of \( B\Pi \). Using these rows, we are able to determine \( O(\epsilon^{-2}k) \) re-weighted rows of \( A \). \( \square \)

**C.7.2 Row sampling for linear regression**

**Theorem C.26** (Theorem 5 in [CNW15]). We are given \( A \in \mathbb{R}^{n \times d} \) with \( \|A\|_2 \leq 1 \) and \( \|A\|_F^2 \leq k \), and an \( \epsilon \in (0, 1) \). There exists a diagonal matrix \( S \) with \( O(k/\epsilon^2) \) nonzero entries such that

\[
\|(SA)^\top SA - A^\top A\|_2 \leq \epsilon.
\]
**Corollary C.27.** Given a rank-\(k\) matrix \(A \in \mathbb{R}^{n \times d}\), vector \(b \in \mathbb{R}^n\), and parameter \(\epsilon > 0\), let \(U \in \mathbb{R}^{n \times (k+1)}\) denote an orthonormal basis of \([A, b] \). Let \(S \in \mathbb{R}^{n \times n}\) denote a sampling and rescaling diagonal matrix according to Leverage score sampling and sparse BSS sampling of \(U\) with \(m\) nonzero entries. If \(m = O(k)\), then \(S\) is a \((1 \pm 1/2)\) subspace embedding for \(U\); if \(m = O(k/\epsilon)\), then \(S\) satisfies \(\sqrt{\epsilon}\)-operator norm approximate matrix product for \(U\).

**Proof.** This follows by Lemma C.22, Lemma C.24 and Theorem C.26.

**Lemma C.28 ([NW14]).** Given \(A \in \mathbb{R}^{n \times d}\) and \(b \in \mathbb{R}^n\), let \(S \in \mathbb{R}^{n \times n}\) denote a sampling and rescaling diagonal matrix. Let \(x^*\) denote \(\arg \min_x \|Ax - b\|_2^2\) and \(x'\) denote \(\arg \min_x \|SAx - Sb\|_2^2\). If \(S\) is a \((1 \pm 1/2)\) subspace embedding for the column span of \(A\), and \(\epsilon' (= \sqrt{\epsilon})\)-operator norm approximate matrix product for \(U\) adjoined with \(b - Ax^*\), then, with probability at least .999,

\[
\|Ax' - b\|_2^2 \leq (1 + \epsilon)\|Ax^* - b\|_2^2.
\]

**Proof.** We define \(\text{OPT} = \min_b \|Ax - b\|_2\). We define \(x' = \arg \min_x \|SAx - Sb\|_2^2\) and \(x^* = \arg \min_x \|Ax - b\|_2^2\). Let \(w = b - Ax^*\). Let \(U\) denote an orthonormal basis of \(A\). We can write \(Ax' - Ax^* = U\beta\). Then, we have,

\[
\|Ax' - b\|_2^2 = \|Ax' - Ax^* + AA^\dagger b - b\|_2^2 \\
= \|U\beta + (UU^\top - I)b\|_2^2 \\
= \|Ax^* - Ax'\|_2^2 + \|Ax^* - b\|_2^2 \\
= \|U\beta\|_2^2 + \text{OPT}^2 \\
= \beta\|_2^2 + \text{OPT}^2.
\]

If \(S\) is a \((1 \pm 1/2)\) subspace embedding for \(U\), then we can show

\[
\|\beta\|_2 - \|U^\top S^\top SU\beta\|_2 \leq \|\beta - U^\top S^\top SU\beta\|_2 \\
\leq \|SU\beta\|_2 \cdot \|\beta\|_2 \\
\leq \frac{1}{2} \|\beta\|_2.
\]

Thus, we obtain

\[
\|U^\top S^\top SU\beta\|_2 \geq \|\beta\|_2/2.
\]

Next, we can show

\[
\|U^\top S^\top SU\beta\|_2 = \|U^\top S^\top S(Ax' - Ax^*)\|_2^2 \\
= \|U^\top S^\top S(A(SA)^\dagger Sb - Ax^*)\|_2^2 \\
= \|U^\top S^\top S(b - Ax^*)\|_2^2 \\
= \|U^\top S^\top Sw\|_2^2.
\]

by \(x' = (SA)^\dagger Sb\),

by \(SA(SA)^\dagger = I\),

by \(w = b - Ax^*\).
We define \( U' = [U \ w/\|w\|_2] \). We define \( X \) and \( y \) to satisfy \( U = U'X \) and \( w = U'y \). Then, we have

\[
\|U^T S^T Sw\|_2 = \|U^T S^T Sw - U^T w\|_2 \\
= \|X^T U'^T S^T SU'y - X^T U'^T U'y\|_2 \\
= \|X^T (U'^T S^T SU' - I)y\|_2 \\
\leq \|X\|_2 \cdot \|U'^T S^T SU' - I\|_2 \cdot \|y\|_2 \\
\leq \epsilon'\|X\|_2 \|w\|_2 \\
= \epsilon'\text{OPT},
\]

by \( \|U\|_2 = 1 \) and \( \|w\|_2 = \text{OPT} \)

where the fifth inequality follows since \( S \) satisfies \( \epsilon' \)-operator norm approximate matrix product for the column span of \( U \) adjoined with \( w \).

Putting it all together, we have

\[
\|Ax' - b\|_2^2 = \|Ax' - b\|_2^2 + \|Ax' - Ax'\|_2^2 \\
= \text{OPT}^2 + \|\beta\|_2^2 \\
\leq \text{OPT}^2 + 4\|U^T S^T Sw\|_2^2 \\
\leq \text{OPT}^2 + 4(\epsilon'\text{OPT})^2 \\
\leq (1 + \epsilon)\text{OPT}^2.
\]

by \( \epsilon' = \frac{1}{2}\sqrt{\epsilon} \).

Finally, note that \( S \) satisfies \( \epsilon' \)-operator norm approximate matrix product for \( U \) adjoined with \( w \) if it is a \((1 \pm \epsilon')\)-subspace embedding for \( U \) adjoined with \( w \), which holds using BSS sampling by Theorem 5 of [CNW15] with \( O(d/\epsilon) \) samples.

\[\square\]

C.7.3 Leverage scores for multiple regression

Lemma C.29 (see, e.g., Lemma 32 in [CW13] among other places). Given matrix \( A \in \mathbb{R}^{n \times d} \) with orthonormal columns, and parameter \( \epsilon > 0 \), if \( S \in \mathbb{R}^{n \times n} \) is a sampling and rescaling diagonal matrix according to the leverage scores of \( A \) where the number of nonzero entries is \( t = O(1/\epsilon^2) \), then, for any \( B \in \mathbb{R}^{n \times m} \), we have

\[
\|A^T S^T SB - A^T B\|_F^2 < \epsilon^2\|A\|_F^2\|B\|_F^2,
\]

holds with probability at least 0.9999.

Corollary C.30. Given matrix \( A \in \mathbb{R}^{n \times d} \) with orthonormal columns, and parameter \( \epsilon > 0 \), if \( S \in \mathbb{R}^{n \times n} \) is a sampling and rescaling diagonal matrix according to the leverage scores of \( A \) with \( m \) nonzero entries, then if \( m = O(d \log d) \), then \( S \) is a \((1 \pm 1/2)\) subspace embedding for \( A \). If \( m = O(d/\epsilon) \), then \( S \) satisfies \( \sqrt{\epsilon/d} \)-Frobenius norm approximate matrix product for \( A \).

Proof. This follows by Lemma C.22 and Lemma C.29.

\[\square\]

Lemma C.31 ([NW14]). Given \( A \in \mathbb{R}^{n \times d} \) and \( B \in \mathbb{R}^{n \times m} \), let \( S \in \mathbb{R}^{n \times n} \) denote a sampling and rescaling matrix according to \( A \). Let \( X^* \) denote \( \arg \min_X \|AX - B\|_F^2 \) and \( X' \) denote \( \arg \min_X \|SAX -
Next, we can show

Thus, we obtain

holds with probability at least 0.999.

Proof. We define $\text{OPT} = \min_X \|AX - B\|_F$. Let $A = USV^T$ denote the SVD of $A$. Since $A$ has rank $k$, $U$ and $V$ have $k$ columns. We can write $A(X' - X^*) = U\beta$. Then, we have

$$
\|AX' - B\|_F^2 = \|AX' - AX^* + AX^* + AA^\dagger B - B\|_F^2
= \|U\beta + (UU^\top - I)B\|_F^2.
= \|AX^* - AX'\|_F^2 + \|AX^* - B\|_F^2
= \|U\beta\|_F^2 + \|B\|_F^2
= \|\beta\|_F^2 + \text{OPT}^2.
$$

(15)

If $S$ is a $(1 \pm 1/2)$ subspace embedding for $U$, then we can show,

$$
\|\beta\|_F - \|U^\top S^\top SSU\beta\|_F
\leq \|\beta - U^\top S^\top SU\beta\|_F
= \|(I - U^\top S^\top SU)\beta\|_F
\leq \|(I - U^\top S^\top SU)\|_2 \cdot \|\beta\|_F
\leq \frac{1}{2} \|\beta\|_F.
$$

(16)

Thus, we obtain

$$
\|U^\top S^\top SU\beta\|_F \geq \|\beta\|_F/2.
$$

Next, we can show

$$
\|U^\top S^\top SU\beta\|_F
= \|U^\top S^\top S(AX' - AX^*)\|_F
= \|U^\top S^\top S(A(SA)^\dagger Sb - AX^*)\|_F
= \|U^\top S^\top S(B - AX^*)\|_F.
$$

(16)

Then, we can show

$$
\|U^\top S^\top S(B - AX^*)\|_F \leq \epsilon' \|U^\top\|_F \|B - AX^*\|_F
= \epsilon' \sqrt{d} \text{OPT}.
$$

(17)

Putting it all together, we have

$$
\|AX' - B\|_F^2 = \|AX^* - B\|_F^2 + \|AX^* - AX'\|_F^2
= \text{OPT}^2 + \|\beta\|_F^2
\leq \text{OPT}^2 + 4 \|U^\top S^\top Sw\|_F^2
\leq \text{OPT}^2 + 4(\epsilon' \sqrt{d} \text{OPT})^2
\leq (1 + \epsilon) \text{OPT}^2.
$$

by $\epsilon' = \frac{1}{2} \sqrt{\epsilon/d}$
C.7.4 Sampling columns according to leverage scores implicitly, improving polynomial running time to nearly linear running time

This section explains an algorithm that is able to sample from the leverage scores from the \( \odot \) product of two matrices \( U, V \) without explicitly writing down \( U \odot V \). To build this algorithm we combine TensorSketch, some ideas from [DMIMW12] and some ideas from [AKO11, MW10]. Finally, we are able to improve the running time of sampling columns according to leverage scores from \( \Omega(n^2) \) to \( \tilde{O}(n) \). Given two matrices \( U, V \in \mathbb{R}^{k \times n} \), we define \( A \in \mathbb{R}^{k \times n_{12}} \) to be the matrix where the \( i \)-th row of \( A \) is the vectorization of \( U_i \odot V^i \), \( \forall i \in [k] \). Naively, in order to sample \( O(\text{poly}(k, 1/\epsilon)) \) rows from \( A^T \) according to leverage scores, we need to write down \( n^2 \) leverage scores. This approach will take at least \( \Omega(n^2) \) running time. In the rest of this section, we will explain how to do it in \( O(n \cdot \text{poly}(\log n, k, 1/\epsilon)) \) time. In Section C.10.1, we will explain how to extend this idea from 3rd order tensors to general \( q \)-th order tensors and remove the \( \text{poly}(\log n) \) factor from running time, i.e., obtain \( O(n \cdot \text{poly}(k, 1/\epsilon)) \) time.

**Lemma C.32.** Given two matrices \( U \in \mathbb{R}^{k \times n_1} \) and \( V \in \mathbb{R}^{k \times n_2} \), there exists an algorithm that takes \( O((n_1 + n_2) \cdot \text{poly}(\log(n_1 n_2), k) \cdot R_{\text{samples}}) \) time and samples \( R_{\text{samples}} \) columns of \( U \odot V \in \mathbb{R}^{k \times n_{12}} \) according to the leverage scores of \( R^{-1}(U \odot V) \), where \( R \) is the \( R \) of a QR factorization.

**Proof.** We choose \( \Pi \in \mathbb{R}^{n_1 n_{12} \times k} \) to be a TensorSketch. Then, according to Section B.10, we can compute \( R^{-1} \) in \( n \cdot \text{poly}(\log n, k, 1/\epsilon) \) time, where \( R \) is the \( R \) in a QR-factorization. We want to sample columns from \( U \odot V \) according to the square of the \( \ell_2 \)-norms of each column of \( R^{-1}(U \odot V) \). However, explicitly writing down the matrix \( R^{-1}(U \odot V) \) takes \( k n_1 n_2 \) time, and the number of columns is already \( n_1 n_2 \). The goal is to sample columns from \( R^{-1}(U \odot V) \) without explicitly computing the square of the \( \ell_2 \)-norm of each column.

The first simple observation is that the following two sampling procedures are equivalent in terms of the column samples of a matrix that they take. (1) We sample a single entry from the matrix \( R^{-1}(U \odot V) \) proportional to its squared value. (2) We sample a column from the matrix \( R^{-1}(U \odot V) \) proportional to its squared \( \ell_2 \)-norm. Let the \( (i, j_1, j_2) \)-th entry denote the entry in the \( i \)-th row and the \( (j_1 - 1)n_2 + j_2 \)-th column. We can show, for a particular column \( (j_1 - 1)n_2 + j_2 \),

\[
\Pr[\text{sample an entry from the } (j_1 - 1)n_2 + j_2 \text{ th column of a matrix}] = \sum_{i=1}^{k} \Pr[\text{sample the } (i, j_1, j_2) \text{-th entry of matrix}]
\]

\[
= \sum_{i=1}^{k} \frac{|(R^{-1}(U \odot V))_{i,(j_1-1)n_2+j_2}|^2}{\|R^{-1}(U \odot V)\|_F^2}
\]

\[
= \frac{\|(R^{-1}(U \odot V))_{(j_1-1)n_2+j_2}\|^2}{\|R^{-1}(U \odot V)\|_F^2}
\]

\[
= \Pr[\text{sample the } (j_1 - 1)n_2 + j_2 \text{ th column of matrix}]. \tag{18}
\]

Thus, it is sufficient to show how to sample a single entry from matrix \( R^{-1}(U \odot V) \) proportional to its squared value without writing down all of the entries of a \( k \times n_1 n_2 \) matrix.

We choose a Gaussian matrix \( G_1 \in \mathbb{R}^{n_1 \times k} \) with \( g_1 = O(\epsilon^{-2} \log(n_1 n_2)) \). By Claim C.33 we can reduce the length of each column vector of matrix \( R^{-1}U \odot V \) from \( k \) to \( g_1 \) while preserving the squared \( \ell_2 \)-norm of all columns simultaneously. Thus, we obtain a new matrix \( GR^{-1}(U \odot V) \in \mathbb{R}^{g_1 \times n_1 n_2} \), and sampling from this new matrix is equivalent to sampling from the original matrix \( R^{-1}(U \odot V) \).

In the following paragraphs, we explain a sampling procedure (also described in Procedure FASTTENSORLEVERAGESCORE in Algorithm 10) which contains three sampling steps. The first
Algorithm 10 Fast Tensor Leverage Score Sampling

1: procedure FASTTENSOREVERAGESCORE($U, V, n_1, n_2, k, \epsilon, R_{\text{samples}}$) \Comment{Lemma C.32}
2: \hspace{1em} $s_1 \leftarrow \text{poly}(k, 1/\epsilon)$. \\
3: \hspace{1em} $g_1 \leftarrow g_2 \leftarrow g_3 \leftarrow O(\epsilon^{-2} \log(n_1n_2))$. \\
4: \hspace{1em} Choose $\Pi \in \mathbb{R}^{n_1n_2 \times 1}$ to be a $\text{TENSORSKETCH}$. \Comment{Definition B.34}
5: \hspace{1em} Compute $R^{-1} \in \mathbb{R}^{k \times k}$ by using $(U \odot V)\Pi$. \Comment{$U \in \mathbb{R}^{k \times n_1}, V \in \mathbb{R}^{k \times n_2}$}
6: \hspace{1em} Choose $G_1 \in \mathbb{R}^{g_1 \times k}$ to be a Gaussian sketching matrix.
7: \hspace{1em} for $i = 1 \rightarrow g_1$ do \\
8: \hspace{2em} $w \leftarrow (G^i R^{-1})^\top$. \Comment{$G^i$ denotes the $i$-th row of $G$}
9: \hspace{2em} for $j = 1 \rightarrow [n_1]$ do \\
10: \hspace{3em} $U^i_j \leftarrow w \odot U_j, \forall j \in [n_1]$. \Comment{$U_j$ denotes the $j$-th column of $U \in \mathbb{R}^{k \times n_1}$}
11: \hspace{2em} end for
12: \hspace{1em} end for
13: \hspace{1em} Choose $G_{2,i} \in \mathbb{R}^{g_2 \times n_1}$ to be a Gaussian sketching matrix.
14: \hspace{1em} for $i = 1 \rightarrow g_1$ do \\
15: \hspace{2em} $\alpha_i \leftarrow ||(G_{2,i}U^\top)V||_F^2$. \\
16: \hspace{1em} Choose $G_{3,i} \in \mathbb{R}^{g_3 \times n_1}$ to be a Gaussian sketching matrix.
17: \hspace{1em} for $j_2 = 1 \rightarrow n_2$ do \\
18: \hspace{2em} $\beta_{i,j_2} \leftarrow \|G_{3,i}(U^\top)V_{j_2}\|_2^2$. \\
19: \hspace{2em} end for
20: \hspace{1em} end for
21: \hspace{1em} $S \leftarrow \emptyset$.
22: \hspace{1em} for $r = 1 \rightarrow R_{\text{samples}}$ do \\
23: \hspace{2em} Sample $i$ from $[g_1]$ with probability $\alpha_i / \sum_{i'=1}^{g_1} \alpha_{i'}$. \\
24: \hspace{2em} Sample $j_2$ from $[n_2]$ with probability $\beta_{i,j_2} / \sum_{j_2'=1}^{n_2} \beta_{i,j_2'}$. \\
25: \hspace{2em} for $j_1 = 1 \rightarrow n_1$ do \\
26: \hspace{3em} $\gamma_{j_1} \leftarrow ((U^\top)_{j_1}V_{j_2})^2$. \\
27: \hspace{2em} end for \\
28: \hspace{2em} Sample $j_1$ from $[n_1]$ with probability $\gamma_{j_1} / \sum_{j_1'=1}^{n_1} \gamma_{j_1'}$. \\
29: \hspace{2em} $S \leftarrow S \cup (j_1, j_2)$. \\
30: \hspace{2em} end for
31: \hspace{1em} Convert $S$ into a diagonal matrix $D$ with at most $R_{\text{samples}}$ nonzero entries.
32: \hspace{1em} return $D$. \Comment{Diagonal matrix $D \in \mathbb{R}^{n_1n_2 \times n_1n_2}$}
33: end procedure

step is sampling $i$ from $[g_1]$, the second step is sampling $j_2$ from $[n_2]$, and the last step is sampling $j_1$ from $[n_1]$. For each $j_1 \in [n_1]$, let $U_{j_1}$ denote the $j_1$-th column of $U$. For each $i \in [g_1]$, let $G^i$ denote the $i$-th row of matrix $G_1 \in \mathbb{R}^{g_1 \times k}$, let $U^i \in \mathbb{R}^{k \times n_1}$ denote a matrix where the $j_1$-th column is $(G^i R^{-1})^\top \odot U_{j_1}$, $\forall j \in [n_1]$. Then, using Claim C.37, we have that $(G^i R^{-1}) \odot (U \odot V) \in \mathbb{R}^{n_1n_2}$ is a vector where the entry in the $(j_1 - 1)n_2 + j_2$-th coordinate is the entry in the $j_1$-th row and $j_2$-th column of matrix $(U^\top V) \in \mathbb{R}^{n_1 \times n_2}$. Further, the squared $\ell_2$-norm of vector $(G^i R^{-1}) \odot (U \odot V)$ is equal to the squared Frobenius norm of matrix $(U^\top V)$. Thus, sampling $i$ proportional to the squared $\ell_2$-norm of vector $(G^i R^{-1}) \odot (U \odot V)$ is equivalent to sampling $i$ proportional to the squared Frobenius norm of matrix $(U^\top V)$. Na"ively, computing the Frobenius norm of an $n_1 \times n_2$ matrix requires $O(n_1n_2)$ time. However, we can choose a Gaussian matrix $G_{2,i} \in \mathbb{R}^{g_2 \times n_1}$ to sample
according to the value \( \| (G_{2,i}U^{\top})V \|_F^2 \), which can be computed in \( O((n_1 + n_2)g_2k) \) time. By claim C.35, \( \| (G_{2,i}U^{\top})V \|_F^2 \approx \| (U^{\top}V) \|_F^2 \) with high probability. So far, we have finished the first step of the sampling procedure.

For the second step of the sampling procedure, we need to sample \( j_2 \) from \( [n_2] \). To do that, we need to compute the squared \( \ell_2 \)-norm of each column of \( U^{\top}V \in \mathbb{R}^{n_1 \times n_2} \). This can be done by choosing another Gaussian matrix \( G_{3,i} \in \mathbb{R}^{g_1 \times n_1} \). For all \( j_2 \in [n_2] \), by Claim C.36, we have \( \| G_{3,i}U^{\top}V_{j_2} \|_2 \approx \| U^{\top}V_{j_2} \|_2 \). Also, for \( j_2 \in [n_2] \), \( \| G_{3,i}U^{\top}V_{j_2} \|_2 \) can be computed in nearly linear in \( n_1 + n_2 \) time.

For the third step of the sampling procedure, we need to sample \( j_1 \) from \( [n_1] \). Since we already have \( i \) and \( j_2 \) from the previous two steps, we can directly compute \( \| (U^{\top}V)_{i}V_{j_2} \|^2 \), for all \( j_1 \). This only takes \( O(n_1k) \) time.

Overall, the running time is \( O((n_1 + n_2) \cdot \text{poly}(\log(n_1n_2), k, 1/\epsilon)) \). Because our estimates are accurate enough, our sampling probabilities are also good approximations to the leverage score sampling probabilities. Putting it all together, we complete the proof.

**Claim C.33.** Given matrix \( R^{-1}(U \odot V) \in \mathbb{R}^{k \times n_1n_2} \), let \( G_1 \in \mathbb{R}^{g_1 \times k} \) denote a Gaussian matrix with \( g_1 = (\epsilon^{-2} \log(n_1n_2)) \). Then with probability at least \( 1 - 1/\text{poly}(n_1n_2) \), we have: for all \( j \in [n_1n_2] \),

\[
(1 - \epsilon)\| R^{-1}(U \odot V)_{j} \|^2 \leq \| G_1 R^{-1}(U \odot V)_{j} \|^2 \leq (1 + \epsilon)\| R^{-1}(U \odot V)_{j} \|^2.
\]

**Proof.** This follows by the Johnson-Lindenstrauss Lemma.

**Claim C.34.** For a fixed \( i \in [g_1] \), let \( G_{2,i} \in \mathbb{R}^{g_2 \times n_1} \) denote a Gaussian matrix with \( g_2 = O(\epsilon^{-2} \log(n_1n_2)) \). Then with probability at least \( 1 - 1/\text{poly}(n_1n_2) \), we have: for all \( j_2 \in [n_2] \),

\[
(1 - \epsilon)\| U^{\top}V_{j_2} \|^2 \leq \| (G_{2,i}U^{\top})V_{j_2} \|^2 \leq (1 + \epsilon)\| U^{\top}V_{j_2} \|^2.
\]

By taking the union bound over all \( i \in [g_1] \), we obtain a stronger claim,

**Claim C.35.** With probability at least \( 1 - 1/\text{poly}(n_1n_2) \), we have: for all \( i \in [g_1] \), for all \( j_2 \in [n_2] \),

\[
(1 - \epsilon)\| U^{\top}V_{j_2} \|^2 \leq \| (G_{2,i}U^{\top})V_{j_2} \|^2 \leq (1 + \epsilon)\| U^{\top}V_{j_2} \|^2.
\]

Similarly, if we choose \( G_{3,i} \) to be a Gaussian matrix, we can obtain the same result as for \( G_{2,i} \):

**Claim C.36.** With probability at least \( 1 - 1/\text{poly}(n_1n_2) \), we have: for all \( i \in [g_1] \), for all \( j_2 \in [n_2] \),

\[
(1 - \epsilon)\| U^{\top}V_{j_2} \|^2 \leq \| (G_{3,i}U^{\top})V_{j_2} \|^2 \leq (1 + \epsilon)\| U^{\top}V_{j_2} \|^2.
\]

**Claim C.37.** For any \( i \in [g_1] \), \( j_1 \in [n_1] \), \( j_2 \in [n_2] \), let \( G_1^i \) denote the \( i \)-th row of matrix \( G_1 \in \mathbb{R}^{g_1 \times k} \). Let \( (U \odot V)_{(j_1-1)n_2+j_2} \) denote the \( (j_1 - 1)n_2 + j_2 \)-th column of matrix \( \mathbb{R}^{k \times n_1n_2} \). Let \( (U^{\top})_{j_1} \) denote the \( j_1 \)-th row of matrix \( U^{\top} \in \mathbb{R}^{n_1 \times k} \). Let \( V_{j_2} \) denote the \( j_2 \)-th column of matrix \( V \in \mathbb{R}^{k \times n_2} \). Then, we have

\[
G_1^i R^{-1}(U \odot V)_{(j_1-1)n_2+j_2} = (U^{\top})_{j_1} V_{j_2}.
\]

**Proof.** This follows by,

\[
G_1^i R^{-1}(U \odot V)_{(j_1-1)n_2+j_2} = G_1^i R^{-1}(U_{j_1} \circ V_{j_2}) = (G_1^i R^{-1} \circ (U_{j_1}^{\top}) V_{j_2}) = (U^{\top})_{j_1} V_{j_2}.
\]
Lemma C.38. Given \( A \in \mathbb{R}^{n \times n^2} \), \( V, W \in \mathbb{R}^{k \times n} \), for any \( \epsilon > 0 \), there exists an algorithm that runs in \( O(n \cdot \text{poly}(k, 1/\epsilon)) \) time and outputs a diagonal matrix \( D \in \mathbb{R}^{n^2 \times n^2} \) with \( m = O(k \log k + k/\epsilon) \) nonzero entries such that,

\[
\| \hat{U}(V \odot W) - A \|_F^2 \leq (1 + \epsilon) \min_{U \in \mathbb{R}^{n \times k}} \| U(V \odot W) - A \|_F^2,
\]

holds with probability at least 0.999, where \( \hat{U} \) denotes the optimal solution to \( \min_U \| U(V \odot W)D - AD \|_F^2 \).

Proof. This follows by combining Theorem C.46, Corollary C.30, and Lemma C.31.

Remark C.39. Replacing Theorem C.46 (Algorithm 15) by Lemma C.32 (Algorithm 10), we can obtain a slightly different version of Lemma C.38 with \( n \cdot \text{poly}(\log n, k, 1/\epsilon) \) running time, where the dependence on \( k \) is better.

C.7.5 Input sparsity time algorithm

Algorithm 11 Frobenius Norm CURT Decomposition Algorithm, Input Sparsity Time and Nearly Optimal Number of Samples

\[
\begin{align*}
1: & \textbf{procedure} \text{FCURTInputSparsity}(A, U_B, V_B, W_B, n, k, \epsilon) \quad \triangleright \text{Theorem C.40} \\
2: & d_1 \leftarrow d_2 \leftarrow d_3 \leftarrow O(k \log k + k/\epsilon). \\
3: & \epsilon_0 \leftarrow 0.01. \\
4: & \text{Form } B_1 = V_B^T \odot W_B^T \in \mathbb{R}^{k \times n^2}. \\
5: & D_1 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(V_B^T, W_B^T, n, n, k, \epsilon_0, d_1). \quad \triangleright \text{Algorithm 15} \\
6: & \text{Form } \hat{U} = A_1 D_1 (B_1 D_1)^{\dagger} \in \mathbb{R}^{n \times k}. \\
7: & \text{Form } B_2 = \hat{U}^T \odot W_B^T \in \mathbb{R}^{k \times n^2}. \\
8: & D_2 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(\hat{U}^T, W_B^T, n, n, k, \epsilon_0, d_2). \\
9: & \text{Form } \hat{V} = A_2 D_2 (B_2 D_2)^{\dagger} \in \mathbb{R}^{n \times k}. \\
10: & \text{Form } B_3 = \hat{V}^T \odot \hat{V}^T \in \mathbb{R}^{k \times n^2}. \\
11: & D_3 \leftarrow \text{FastTensorLeverageScoreGeneralOrder}(\hat{U}^T, \hat{V}^T, n, n, k, \epsilon_0, d_3). \\
12: & C \leftarrow A_1 D_1, R \leftarrow A_2 D_2, T \leftarrow A_3 D_3. \\
13: & U \leftarrow \sum_{i=1}^{k} ((B_1 D_1)^{\dagger})_i \otimes ((B_2 D_2)^{\dagger})_i \otimes ((B_3 D_3)^{\dagger})_i. \\
14: & \text{return } C, R, T \text{ and } U. \\
15: \textbf{end procedure}
\]

Theorem C.40. Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), let \( k \geq 1 \), and let \( U_B, V_B, W_B \in \mathbb{R}^{n \times k} \) denote a rank-k, \( \alpha \)-approximation to \( A \). Then there exists an algorithm which takes \( O(mn \alpha(A) + n \cdot \text{poly}(k, 1/\epsilon)) \) time and outputs three matrices \( C \in \mathbb{R}^{n \times n \times n} \) with columns from \( A \), \( R \in \mathbb{R}^{n \times r} \) with rows from \( A \), \( T \in \mathbb{R}^{n \times t} \) with tubes from \( A \), and a tensor \( U \in \mathbb{R}^{n \times r \times t} \) with rank \( (U) = k \) such that \( c = r = t = O(k \log k + k/\epsilon) \), and

\[
\left\| \sum_{i=1}^{k} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_F^2 \leq (1 + \epsilon) \alpha \min_{\text{rank-k } A'} \| A' - A \|_F^2
\]

holds with probability 9/10.
Proof. We define
\[
\text{OPT} := \min_{\text{rank-}k A'} \|A' - A\|_F^2.
\]
We already have three matrices \(U_B \in \mathbb{R}^{n \times k}, V_B \in \mathbb{R}^{n \times k}\) and \(W_B \in \mathbb{R}^{n \times k}\) and these three matrices provide a rank-\(k\), \(\alpha\)-approximation to \(A\), i.e.,
\[
\left\| \sum_{i=1}^{k} (U_B)_{i} \otimes (V_B)_{i} \otimes (W_B)_{i} - A \right\|_F^2 \leq \alpha \\text{OPT}.
\] (19)

Let \(B_1 = V_B^T \odot W_B^T \in \mathbb{R}^{k \times n^2}\) denote the matrix where the \(i\)-th row is the vectorization of \((V_B)_{i} \otimes (W_B)_{i}\). Let \(D_1 \in \mathbb{R}^{n^2 \times n^2}\) be a sampling and rescaling matrix corresponding to sampling by the leverage scores of \(B_1^T\); there are \(d_1\) nonzero entries on the diagonal of \(D_1\). Let \(A_i \in \mathbb{R}^{n \times n^2}\) denote the matrix obtained by flattening \(A\) along the \(i\)-th direction, for each \(i \in [3]\).

Define \(U^* \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{U \in \mathbb{R}^{n \times k}} \|UB_1 - A_1\|_F^2, \hat{U} = A_1D_1(B_1D_1)^T \in \mathbb{R}^{n \times k}\), and \(V_0 \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{V \in \mathbb{R}^{n \times k}} \|V \cdot (\hat{U}^T \odot W_B^T) - A_2\|_F^2\). Due to Lemma C.38, if \(d_1 = O(k \log k + k/\epsilon)\) then with constant probability, we have
\[
\|\hat{U}B_1 - A_1\|_F^2 \leq \alpha D_1 \|U^*B_1 - A_1\|_F^2.
\] (20)

Recall that \((\hat{U}^T \odot W_B^T) \in \mathbb{R}^{k \times n^2}\) denotes the matrix where the \(i\)-th row is the vectorization of \(\hat{U}_i \otimes (W_B)_i\), \(\forall i \in [k]\). Now, we can show,
\[
\|V_0 \cdot (\hat{U}^T \odot W_B^T) - A_2\|_F^2 \leq \|\hat{U}B_1 - A_1\|_F^2 \leq \alpha D_1 \|U^*B_1 - A_1\|_F^2 
\] by \(V_0 = \arg\min_{V \in \mathbb{R}^{n \times k}} \|V \cdot (\hat{U}^T \odot W_B^T) - A_2\|_F^2\) by Equation (20)
\[
\leq \alpha D_1 \|UB_1 - A_1\|_F^2 \leq \alpha D_1 \alpha \text{OPT}.
\] by Equation (19) (21)

We define \(B_2 = \hat{U}^T \odot W_B^T\). Let \(D_2 \in \mathbb{R}^{n^2 \times n^2}\) be a sampling and rescaling matrix corresponding to the leverage scores of \(B_2^T\). Suppose there are \(d_2\) nonzero entries on the diagonal of \(D_2\).

Define \(V^* \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{V \in \mathbb{R}^{n \times k}} \|VB_2 - A_2\|_F^2, \hat{V} = A_2D_2(B_2D_2)^T \in \mathbb{R}^{n \times k}, W_0 \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{W \in \mathbb{R}^{n \times k}} \|W \cdot (\hat{U}^T \odot \hat{V}^T) - A_3\|_F^2\), and \(V^*\) to be the optimal solution to \(\min_{V \in \mathbb{R}^{n \times k}} \|VB_2D_2 - A_2D_2\|_F^2\).

Due to Lemma C.38, with constant probability, we have
\[
\|\hat{V}B_2 - A_2\|_F^2 \leq \alpha D_2 \|V^*B_2 - A_2\|_F^2.
\] (22)

Recall that \((\hat{U}^T \odot \hat{V}^T) \in \mathbb{R}^{k \times n^2}\) denotes the matrix where the \(i\)-th row is the vectorization of \(\hat{U}_i \otimes \hat{V}_i\), \(\forall i \in [k]\). Now, we can show,
\[
\|W_0 \cdot (\hat{U}^T \odot \hat{V}^T) - A_3\|_F^2 \leq \|\hat{V}B_2 - A_2\|_F^2 \leq \alpha D_2 \|V^*B_2 - A_2\|_F^2 \leq \alpha D_2 \|V_0B_2 - A_2\|_F^2 
\] by \(W_0 = \arg\min_{W \in \mathbb{R}^{n \times k}} \|W \cdot (\hat{U}^T \odot \hat{V}^T) - A_3\|_F^2\) by Equation (22)
\[
\leq \alpha D_2 \|V_0B_2 - A_2\|_F^2 \leq \alpha D_2 \alpha \text{OPT}.
\] by Equation (21) (23)
We define $B_3 = \hat{U}^\top \circ \hat{V}^\top$. Let $D_3 \in \mathbb{R}^{n^2 \times n^2}$ denote a sampling and rescaling matrix corresponding to sampling by the leverage scores of $B_3^\top$. Suppose there are $d_3$ nonzero entries on the diagonal of $D_3$.

Define $W^* \in \mathbb{R}^{n \times k}$ to be the optimal solution to $\min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_F^2$, $\tilde{W} = A_3 D_3 (B_3 D_3)^\dagger \in \mathbb{R}^{n \times k}$, and $W'$ to be the optimal solution to $\min_{W \in \mathbb{R}^{n \times k}} \|WB_3 D_3 - A_3 D_3\|_F^2$.

Due to Lemma C.38 with constant probability, we have

$$\|\tilde{W} B_3 - A_3\|_F^2 \leq \alpha_{D_3} \|W^* B_3 - A_3\|_F^2. \tag{24}$$

Now we can show,

$$\|\tilde{W} B_3 - A_3\|_F^2 \leq \alpha_{D_3} \|W^* B_3 - A_3\|_F^2, \quad \text{by Equation (24)}$$
$$\leq \alpha_{D_3} \|W_0 B_3 - A_3\|_F^2, \quad \text{by } W^* = \arg \min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_F^2$$
$$\leq \alpha_{D_3} \alpha_{D_2} \alpha_{D_1} \alpha \text{OPT}, \quad \text{by Equation (23)}$$

This implies,

$$\left\| \sum_{i=1}^k \hat{U}_i \otimes \hat{V}_i \otimes \hat{W}_i - A \right\|_F^2 \leq O(1) \alpha \text{OPT}^2.$$

where $\hat{U} = A_1 D_1 (B_1 D_1)^\dagger$, $\hat{V} = A_2 D_2 (B_2 D_2)^\dagger$, $\hat{W} = A_3 D_3 (B_3 D_3)^\dagger$.

By Lemma C.38, we need to set $d_1 = d_2 = d_3 = O(k \log k + k/\epsilon)$. Note that $B_1 = (V_B^\top \circ W_B^\top)$. Thus $D_1$ can be found in $n \cdot \text{poly}(k, 1/\epsilon)$ time. Because $D_1$ has a small number of nonzero entries on the diagonal, we can compute $B_1 D_1$ quickly without explicitly writing down $B_1$. Also $A_1 D_1$ can be computed in $\text{nnz}(A)$ time. Using $(A_1 D_1)$ and $(B_1 D_1)$, we can compute $\hat{U}$ in $n \text{poly}(k, 1/\epsilon)$ time. In a similar way, we can compute $B_2$, $B_2$, $B_3$, and $D_3$. Since tensor $U$ is constructed based on three $\text{poly}(k, 1/\epsilon)$ size matrices, $(B_1 D_1)^\dagger$, $(B_2 D_2)^\dagger$, and $(B_3 D_3)^\dagger$, the overall running time is $O(\text{nnz}(A) + n \text{poly}(k, 1/\epsilon))$.

C.7.6 Optimal sample complexity algorithm

**Theorem C.41.** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $k \geq 1$, and let $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$ denote a rank-$k$, $\alpha$-approximation to $A$. Then there exists an algorithm which takes $O(\text{nnz}(A) \log n + n^2 \text{poly}(\log n, k, 1/\epsilon))$ time and outputs three matrices: $C \in \mathbb{R}^{n \times \ell}$ with columns from $A$, $R \in \mathbb{R}^{n \times r}$ with rows from $A$, $T \in \mathbb{R}^{n \times t}$ with tubes from $A$, and a tensor $U \in \mathbb{R}^{c \times r \times t}$ with rank($U$) = $k$ such that $c = r = t = O(k/\epsilon)$, and

$$\left\| \sum_{i=1}^c \sum_{j=1}^r \sum_{l=1}^t U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_F^2 \leq (1 + \epsilon) \alpha \min_{\text{rank} - k} \|A' - A\|_F^2$$

holds with probability 9/10.

**Proof.** The proof is almost the same as the proof of Theorem C.40. The only difference is that instead of using Theorem C.38, we use Theorem C.14. \qed
Algorithm 12 Frobenius Norm CURT Decomposition Algorithm, Optimal Sample Complexity

1: procedure FCURTOPTIMALSAMPLES($A, U_B, V_B, W_B, n, k$) \texttt{\textsuperscript{\textcopyright} Theorem C.41}
2: \quad $d_1 \leftarrow d_2 \leftarrow d_3 \leftarrow O(k/\epsilon)$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
3: \quad Form $B_1 = V_B^\top \odot W_B^\top \in \mathbb{R}^{k \times n}$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
4: \quad $D_1 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_1^\top, B_1^\top, n^2, n, k, \epsilon)$. \texttt{\textsuperscript{\textcopyright} Algorithm 7}
5: \quad Let $d_1$ denote the number of nonzero entries in $D_1$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
6: \quad Form $\hat{U} = A_1 D_1 (B_1 D_1)^\dagger \in \mathbb{R}^{n \times k}$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
7: \quad Form $B_2 = \hat{U}^\top \odot W_B^\top \in \mathbb{R}^{k \times n^2}$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
8: \quad $D_2 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_2^\top, B_2^\top, n^2, n, k, \epsilon)$. \texttt{\textsuperscript{\textcopyright} Algorithm 7}
9: \quad Let $d_2$ denote the number of nonzero entries in $D_2$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
10: \quad Form $\hat{V} = A_2 D_2 (B_2 D_2)^\dagger \in \mathbb{R}^{n \times k}$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
11: \quad Form $B_3 = \hat{V}^\top \odot V_B^\top \in \mathbb{R}^{k \times n^2}$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
12: \quad $D_3 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_3^\top, B_3^\top, n^2, n, k, \epsilon)$. \texttt{\textsuperscript{\textcopyright} Algorithm 7}
13: \quad $d_3$ denote the number of nonzero entries in $D_3$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
14: \quad $C \leftarrow A_1 D_1, R \leftarrow A_2 D_2, T \leftarrow A_3 D_3$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
15: \quad $U \leftarrow \sum_{i=1}^k ((B_1 D_1)^\dagger_i \otimes (B_2 D_2)^\dagger_i \otimes (B_3 D_3)^\dagger_i)$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
16: \quad return $C, R, T$ and $U$. \texttt{\textsuperscript{\textcopyright} Theorem C.41}
17: end procedure

C.8 Face-based selection and decomposition

Previously we provided column-based tensor CURT algorithms, which are algorithms that can select a subset of columns from each of the three dimensions. Here we provide two face-based tensor CURT decomposition algorithms. The first algorithm runs in polynomial time and is a bicriteria algorithm (the number of samples is poly($k/\epsilon$)). The second algorithm needs to start with a rank-$k$ $(1 + O(\epsilon))$-approximate solution, which we then show how to combine with our previous algorithm. Both of our algorithms are able to select a subset of column-row faces, a subset of row-tube faces and a subset of column-tube faces. The second algorithm is able to output $U$, but the first algorithm is not.

C.8.1 Column-row, column-tube, row-tube face subset selection

Theorem C.42. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(n \text{nnz}(A)) \log n + n^2 \text{poly}(\log n, k, 1/\epsilon)$ time and outputs three tensors: a subset $C \in \mathbb{R}^{c \times n \times n}$ of row-tube faces of $A$, a subset $R \in \mathbb{R}^{n \times n \times n}$ of column-tube faces of $A$, and a subset $T \in \mathbb{R}^{n \times n \times t}$ of column-row faces of $A$, where $c = r = t = \text{poly}(k, 1/\epsilon)$, and for which there exists a tensor $U \in \mathbb{R}^{tn \times n \times n}$ for which

$$\|U(T_1, C_2, R_3) - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}-k A'} \|A' - A\|_F^2,$$

or equivalently,

$$\left\| \sum_{i=1}^{tn} \sum_{j=1}^{cn} \sum_{l=1}^{rn} U_{i,j,l} \cdot (T_1)_{i} \otimes (C_2)_{j} \otimes (R_3)_{l} - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}-k A'} \|A' - A\|_F^2.$$

Proof. We fix $V^* \in \mathbb{R}^{n \times k}$ and $W^* \in \mathbb{R}^{n \times k}$. We define $Z_1 \in \mathbb{R}^{k \times n^2}$ where the $i$-th row of $Z_1$ is the vector $V_i \otimes W_i$. Choose a sketching (Gaussian) matrix $S_1 \in \mathbb{R}^{n \times s_1}$ (Definition B.18), and let
We fix a sketching (Gaussian) matrix $U$. Following a similar argument as in the previous theorem, we have

$\|\tilde{U}Z_1 - A_1\|_F^2 \leq (1 + \epsilon)\text{OPT}.$

We fix $\tilde{U}$ and $W^*$. We define $Z_2 \in \mathbb{R}^{k \times n^2}$ where the $i$-th row of $Z_2$ is the vector $\tilde{U}_i \odot W_i$. Choose a sketching (Gaussian) matrix $S_2 \in \mathbb{R}^{n^2 \times n_2}$ (Definition B.18), and let $\tilde{V} = A_2S_2(Z_2S_2)^\dagger \in \mathbb{R}^{n \times k}$. Following a similar argument as in the previous theorem, we have

$\|\tilde{V}Z_2 - A_2\|_F^2 \leq (1 + \epsilon)^2\text{OPT}.$

We fix $\tilde{U}$ and $\tilde{V}$. Note that $\tilde{U} = A_1S_1(Z_1S_1)^\dagger$ and $\tilde{V} = A_2S_2(Z_2S_2)^\dagger$. We define $Z_3 \in \mathbb{R}^{k \times n^2}$ such that the $i$-th row of $Z_3$ is the vector $\tilde{U}_i \odot \tilde{V}_i$. Let $z_3 = s_1 \odot s_2$. We define $Z_3' \in \mathbb{R}^{k \times n^2}$ such that, $\forall i \in [s_1], \forall j \in [s_2], i + (j - 1)s_1$-th row of $Z_3'$ is the vector $(A_1S_1)_i \odot (A_2S_2)_j$.

We define $U_3 \in \mathbb{R}^{n \times z_3}$ to be the matrix where the $i + (j - 1)s_1$-th column is $(A_1S_1)_i$ and $V_3 \in \mathbb{R}^{n \times z_3}$ to be the matrix where the $i + (j - 1)s_1$-th column is $(A_2S_2)_j$. Then $Z_3' = (U_3^\top \odot V_3^\top)$.

We first have,

$$\min_{W \in \mathbb{R}^{n \times k}, X \in \mathbb{R}^{k \times z_3}} \|WXZ_3' - A_3\|_F^2 \leq \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2 \leq (1 + \epsilon)^2\text{OPT}.$$ 

Now consider the following objective function,

$$\min_{W \in \mathbb{R}^{n \times z_3}} \|V_3 \cdot (W^\top \odot U_3^\top) - A_2\|_F^2.$$

Let $D_3$ denote a sampling and rescaling diagonal matrix according to $V_1 \in \mathbb{R}^{n \times z_3}$, let $d_3$ denote the number of nonzero entries of $D_3$. Then we have

$$\min_{W \in \mathbb{R}^{n \times z_3}} \|D_3V_3 \cdot (W^\top \odot U_3^\top) - D_3A_2\|_F^2 = \min_{W \in \mathbb{R}^{n \times z_3}} \|U_3 \odot (D_3V_3) \odot W - A(I, D_3, I)\|_F^2 = \min_{W \in \mathbb{R}^{n \times z_3}} \|W \cdot (U_3^\top \odot (D_3V_3)^\top) - (A(I, D_3, I))_3\|_F^2,$$
Let $Z_3$ denote $U_3^T (D_3 V_3)^T \in \mathbb{R}^{z_3 \times n d_3}$ and \( W' = (A(I, D_3, I))^3 \in \mathbb{R}^{n \times n d_3} \). Using Theorem C.14, we can find a diagonal matrix $D_3 \in \mathbb{R}^{n^2 \times n^2}$ with $d_3 = O(z_3/\epsilon) = O(k^2/\epsilon^3)$ nonzero entries such that

$$
\|U_3 \otimes V_3 \otimes (W' Z_3^\dagger) - A\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.
$$

We define $U_2 = U_3 \in \mathbb{R}^{n \times z_2}$ with $z_2 = z_3$. We define $W_2 = W' Z_3^\dagger \in \mathbb{R}^{n \times z_2}$ with $z_2 = z_3$. We consider,

$$
\min_{V \in \mathbb{R}^{n \times z_2}} \|U_2 \cdot (V^T \otimes W_2^T) - A_1\|_F^2.
$$

Let $D_2$ denote a sampling and rescaling matrix according to $U_2$, and let $d_2$ denote the number of nonzero entries of $D_2$. Then, we have

$$
\min_{V \in \mathbb{R}^{n \times z_2}} \|D_2 U_2 \cdot (V^T \otimes W_2^T) - D_2 A_1\|_F^2.
$$

where the first equality follows by retensorizing the objective function, and the second equality follows by flattening the tensor along the third dimension.

Let $Z_2$ denote $(W_2^T \otimes (D_2 U_2)^T) \in \mathbb{R}^{z_2 \times n d_2}$ and $V' = (A(D_2, I, I))^2 \in \mathbb{R}^{n \times n d_2}$. Using Theorem C.14, we can find a diagonal matrix $D_2 \in \mathbb{R}^{n^2 \times n^2}$ with $d_2 = O(z_2/\epsilon) = O(z_2/\epsilon)$ nonzero entries such that

$$
\|U_2 \otimes (V' Z_2^\dagger) \otimes W_2 - A\|_F^2 \leq (1 + \epsilon)^4 \text{OPT}.
$$

We define $W_1 = W_2 \in \mathbb{R}^{D \times z_1}$ with $z_1 = z_2$, and define $V_1 = (V' Z_2^\dagger) \in \mathbb{R}^{n \times z_1}$ with $z_1 = z_2$.

Let $D_1$ denote a sampling and rescaling matrix according to $W_1$, and let $d_1$ denote the number of nonzero entries of $D_1$. Then we have

$$
\min_{V \in \mathbb{R}^{n \times z_1}} \|D_1 W_1 \cdot (U^T \otimes V_1^T) - D_1 A_3\|_F^2.
$$

where the first equality follows by unflattening the objective function, and second equality follows by flattening the tensor along the first dimension.

Let $Z_1$ denote $(V_1^T \otimes (D_1 W_1)^T) \in \mathbb{R}^{z_1 \times n d_1}$, and $U' = A(I, I, D_1) \in \mathbb{R}^{n \times n d_1}$. Using Theorem C.14, we can find a diagonal matrix $D_1 \in \mathbb{R}^{n^2 \times n^2}$ with $d_1 = O(z_1/\epsilon) = O(z_1/\epsilon)$ nonzero entries such that

$$
\|(U' Z_1^\dagger) \otimes (V_1) \otimes W_1 - A\|_F^2 \leq (1 + \epsilon)^5 \text{OPT},
$$

which means,

$$
\|(U' Z_1^\dagger) \otimes (V' Z_2^\dagger) \otimes (W' Z_3^\dagger) - A\|_F^2 \leq (1 + \epsilon)^5 \text{OPT}.
$$

Putting $U', V', W'$ together completes the proof.
Corollary C.43. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(\text{nnz}(A)) + n^2 \text{poly}(k,1/\epsilon)$ time and outputs three tensors: a subset $C \in \mathbb{R}^{c \times n \times n}$ of row-tube faces of $A$, a subset $R \in \mathbb{R}^{n \times r \times n}$ of column-tube faces of $A$, and a subset $T \in \mathbb{R}^{n \times n \times t}$ of column-row faces of $A$, where $c = r = t = \text{poly}(k,1/\epsilon)$, so that there exists a tensor $U \in \mathbb{R}^{n \times n \times c} \times \mathbb{R}^{n \times n \times r} \times \mathbb{R}^{n \times n \times t}$ for which

$$
\|U(T_1, C_2, R_3) - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank} - k A'} \|A' - A\|_F^2,
$$

or equivalently,

$$
\left\| \sum_{i=1}^{tn} \sum_{j=1}^{cn} \sum_{l=1}^{rn} U_{i,j,l} \cdot (T_1)_i \otimes (C_2)_j \otimes (R_3)_l - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank} - k A'} \|A' - A\|_F^2.
$$

Proof. If we allow a $\text{poly}(k/\epsilon)$ factor increase in running time and a $\text{poly}(k/\epsilon)$ factor increase in the number of faces selected, then instead of using generalized row subset selection, which has running time depending on $\log n$, we can use the technique in Section C.11 to avoid the $\log n$ factor. \qed

C.8.2 CURT decomposition

Algorithm 14 Frobenius Norm (Face-based) CURT Decomposition Algorithm, Optimal Sample Complexity

1: procedure FFACERCURTDECOMPOSITION($A, U_B, V_B, W_B, n, k$) \hfill \triangleright \ \text{Theorem C.44}
2: \hspace{1em} $D_1 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_3, W_B, n, n^2, k, \epsilon)$. \hfill \triangleright \ \text{Algorithm 7,}
3: \hspace{1em} the number of nonzero entries is $d_1 = O(k/\epsilon)$
4: \hspace{1em} Form $Z_1 = V_B^\top \odot (D_1 W_B)^\top$.
5: \hspace{1em} Form $\widehat{U} = (A(I, I, D_1))_1 Z_1^\dagger \in \mathbb{R}^{n \times k}$.
6: \hspace{1em} $D_2 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_1, \widehat{U}, n, n^2, k, \epsilon)$. \hfill \triangleright \ \text{The number of}
7: \hspace{1em} nonzero entries is $d_2 = O(k/\epsilon)$
8: \hspace{1em} Form $Z_2 = (W_B^\top \odot (D_2 \widehat{U}))$.
9: \hspace{1em} Form $\widehat{V} = (A(D_2, I, I))_2 Z_2^\dagger \in \mathbb{R}^{n \times k}$.
10: \hspace{1em} $D_3 \leftarrow \text{GENERALIZEDMATRIXROWSUBSETSELECTION}(A_2, \widehat{V}, n, n^2, k, \epsilon)$. \hfill \triangleright \ \text{The number of}
11: \hspace{1em} nonzero entries is $d_3 = O(k/\epsilon)$
12: \hspace{1em} Form $Z_3 = \widehat{U}^\top \odot (D_3 \widehat{V})^\top$.
13: \hspace{1em} Form $\widehat{W} = (A(I, D_3, I))_3 (Z_3)^\dagger \in \mathbb{R}^{n \times k}$.
14: \hspace{1em} $T \leftarrow A(I, I, D_1), C \leftarrow A(D_2, I, I), R \leftarrow A(I, D_3, I)$.
15: \hspace{1em} $U \leftarrow \sum_{i=1}^{k} ((Z_1)^\dagger)_i \odot ((Z_2)^\dagger)_i \odot ((Z_3)^\dagger)_i$.
17: end procedure

Theorem C.44. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $k \geq 1$, and let $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$ denote a rank-$k$, $\alpha$-approximation to $A$. Then there exists an algorithm which takes $O(\text{nnz}(A)) \log n + n^2 \text{poly}(\log n, k, 1/\epsilon)$ time and outputs three tensors: $C \in \mathbb{R}^{c \times n \times n}$ with row-tube faces from $A$, $R \in \mathbb{R}^{n \times r \times n}$ with column-tube faces from $A$, $T \in \mathbb{R}^{n \times n \times t}$ with column-row faces from $A$, and a (factorization of a) tensor $U \in \mathbb{R}^{n \times c \times n \times r \times n \times t}$ with rank($U$) = $k$ for which $c = r = t = O(k/\epsilon)$ and

$$
\|U(T_1, C_2, R_3) - A\|_F^2 \leq (1 + \epsilon) \alpha \min_{\text{rank} - k A'} \|A' - A\|_F^2,
$$

67
or equivalently,

\[
\| \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{r} U_{i,j,l} \cdot (T_1)_i \otimes (C_2)_j \otimes (R_3)_l - A \|_F^2 \leq (1 + \epsilon) \alpha \min_{\operatorname{rank}-k A'} \| A' - A \|_F^2
\]

holds with probability 9/10.

Proof. We already have three matrices \( U_B \in \mathbb{R}^{n \times k}, V_B \in \mathbb{R}^{n \times k} \) and \( W_B \in \mathbb{R}^{n \times k} \) and these three matrices provide a rank-\( k \), \( \alpha \)-approximation to \( A \), i.e.,

\[
\| U_B \otimes V_B \otimes W_B - A \|_F^2 \leq \alpha \min_{\operatorname{OPT}} \| A' - A \|_F^2.
\]

We can consider the following problem,

\[
\min_{U \in \mathbb{R}^{n \times k}} \| W_B \cdot (U^\top \otimes V_B^\top) - A_3 \|_F^2.
\]

Let \( D_1 \) denote a sampling and rescaling diagonal matrix according to \( W_B \), and let \( d_1 \) denote the number of nonzero entries of \( D_1 \). Then we have

\[
\min_{U \in \mathbb{R}^{n \times k}} \| (D_1 W_B) \cdot (U^\top \otimes V_B^\top) - D_1 A_3 \|_F^2
\]

\[
= \min_{U \in \mathbb{R}^{n \times k}} \| U \otimes V_B \otimes D_1 W_B - A(I, I, D_1) \|_F^2
\]

\[
= \min_{U \in \mathbb{R}^{n \times k}} \| U \cdot (V_B^\top \otimes (D_1 W_B)^\top) - (A(I, I, D_1))_1 \|_F^2,
\]

where the first equality follows by retensorizing the objective function, and the second equality follows by flattening the tensor along the first dimension. Let \( Z_1 \) denote \( V_B^\top \otimes (D_1 W_B)^\top \in \mathbb{R}^{k \times nd_1} \), and define \( \hat{U} = (A(I, I, D_1))_1 Z_1^\top \in \mathbb{R}^{n \times k} \). Then we have

\[
\| \hat{U} \otimes V_B \otimes W_B - A \|_F^2 \leq (1 + \epsilon) \alpha \operatorname{OPT}.
\]

In the second step, we fix \( \hat{U} \) and \( W_B \), and consider the following objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \| \hat{U} \cdot (V^\top \otimes W_B) - A_1 \|_F^2.
\]

Let \( D_2 \) denote a sampling and rescaling matrix according to \( \hat{U} \), and let \( d_2 \) denote the number of nonzero entries of \( D_2 \). Then we have

\[
\min_{V \in \mathbb{R}^{n \times k}} \| (D_2 \hat{U}) \cdot (V^\top \otimes W_B^\top) - D_2 A_1 \|_F^2
\]

\[
= \min_{V \in \mathbb{R}^{n \times k}} \| (D_2 \hat{U}) \otimes V \otimes W_B - A(D_2, I, I) \|_F^2
\]

\[
= \min_{V \in \mathbb{R}^{n \times k}} \| V \cdot (W_B^\top \otimes (D_2 \hat{U})^\top) - (A(D_2, I, I))_2 \|_F^2,
\]

where the first equality follows by unflattening the objective function, and the second equality follows by flattening the tensor along the second dimension. Let \( Z_2 \) denote \( W_B^\top \otimes (D_2 \hat{U})^\top \in \mathbb{R}^{k \times nd_2} \), and define \( \hat{V} = (A(D_2, I, I))_2 Z_2^\top \in \mathbb{R}^{n \times k} \). Then we have

\[
\| \hat{U} \otimes \hat{V} \otimes W_B - A \|_F^2 \leq (1 + \epsilon)^2 \alpha \operatorname{OPT}.
\]

68
In the third step, we fix $\hat{U}$ and $\hat{V}$, and consider the following objective function,

$$\min_{W \in \mathbb{R}^{n \times k}} \| \hat{V} \cdot (W \odot \hat{U}) - A_2 \|_F^2.$$ 

Let $D_3$ denote a sampling and rescaling matrix according to $\hat{V}$, and let $d_3$ denote the number of nonzero entries of $D_3$. Then we have,

$$\min_{W \in \mathbb{R}^{n \times k}} \| (D_3 \hat{V}) \cdot (W^\top \odot \hat{U}^\top) - D_3 A_2 \|_F^2 = \min_{W \in \mathbb{R}^{n \times k}} \| \hat{U} \otimes (D_3 \hat{V}) \otimes W - A(I, D_3, I) \|_F^2 = \min_{W \in \mathbb{R}^{n \times k}} \| W \cdot (\hat{U}^\top \odot (D_3 \hat{V})^\top) - (A(I, D_3, I))_3 \|_F^2,$$

where the first equality follows by retensorizing the objective function, and the second equality follows by flattening the tensor along the third dimension. Let $Z_3$ denote $(\hat{U}^\top \odot (D_3 \hat{V})^\top) \in \mathbb{R}^{d_3 \times n \times d_3}$, and define $\hat{W} = (A(I, D_3, I))_3 (Z_3)^\dagger$. Putting it all together, we have,

$$\| \hat{U} \otimes \hat{V} \otimes \hat{W} - A \|_F^2 \leq (1 + \epsilon)^3 \alpha \text{OPT}.$$ 

This implies

$$\| (A(I, I, D_1))_1 Z_1^\dagger \otimes (A(D_2, I, I))_2 Z_2^\dagger \otimes (A(I, D_3, I))_3 Z_3^\dagger - A \|_F^2 \leq (1 + \epsilon)^3 \alpha \text{OPT}.$$

\[ \square \]

### C.9 Solving small problems

**Theorem C.45.** Let $\max \{t_1, d_1\} \leq n$. Given a $t_1 \times t_2 \times t_3$ tensor $A$ and three matrices: a $t_1 \times d_1$ matrix $T_1$, a $t_2 \times d_2$ matrix $T_2$, and a $t_3 \times d_3$ matrix $T_3$, if for any $\delta > 0$ there exists a solution to

$$\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^k (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i - A \right\|_F^2 := \text{OPT},$$

and each entry of $X_i$ can be expressed using $O(n^\delta)$ bits, then there exists an algorithm that takes $n^{O(\delta)} 2^{O(d_1 k + d_2 k + d_3 k)}$ time and outputs three matrices: $\hat{X}_1, \hat{X}_2,$ and $\hat{X}_3$ such that $\| (T_1 \hat{X}_1) \otimes (T_2 \hat{X}_2) \otimes (T_3 \hat{X}_3) - A \|_F^2 = \text{OPT}.$

**Proof.** For each $i \in [3]$, we can create $t_i \times d_i$ variables to represent matrix $X_i$. Let $x$ denote this list of variables. Let $B$ denote the tensor $\sum_{i=1}^k (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i$ and let $B_{i,j,l}(x)$ denote an entry of tensor $B$ (which can be thought of as a polynomial written in terms of $x$). Then we can write the following objective function,

$$\min_x \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} (B_{i,j,l}(x) - A_{i,j,l})^2.$$ 

We slightly modify the above objective function to obtain a new objective function,

$$\min_{x, \sigma} \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} (B_{i,j,l}(x) - A_{i,j,l})^2,$$

s.t. $\|x\|_2^2 \leq 2^{O(n^\delta)},$
where the last constraint is unharmful, because there exists a solution that can be written using $O(n^δ)$ bits. Note that the number of inequality constraints in the above system is $O(1)$, the degree is $O(1)$, and the number of variables is $v = (d_1 k + d_2 k + d_3 k)$. Thus by Theorem B.11, the minimum nonzero cost is at least

$$(2^{O(n^δ)}) - 2^{O(v)}.$$ 

It is clear that the upper bound on the cost is at most $2^{O(n^δ)}$. Thus the number of binary search steps is at most $\log(2^{O(n^δ)})/2^{O(v)}$. In each step of the binary search, we need to choose a cost between the lower bound and the upper bound, and write down the polynomial system,

$$\sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} (B_{i,j,l}(x) - A_{i,j,l})^2 \leq C,$$

$$\|x\|_2^2 \leq 2^{O(n^δ)}.$$

Using Theorem B.10, we can determine if there exists a solution to the above polynomial system. Since the number of variables is $v$, and the degree is $O(1)$, the number of inequality constraints is $O(1)$. Thus, the running time is

$$\text{poly}(\text{bitsize}) \cdot (\# \text{constraints} \cdot \text{degree}) \cdot \# \text{variables} = n^{O(δ)}2^{O(v)}.$$

\[
\square
\]

### C.10 Extension to general $q$-th order tensors

This section provides the details for our extensions from 3rd order tensors to general $q$-th order tensors. In most practical applications, the order $q$ is a constant. Thus, to simplify the analysis, we use $O_q(\cdot)$ to hide dependencies on $q$.

#### C.10.1 Fast sampling of columns according to leverage scores, implicitly

This section explains an algorithm that is able to sample from the leverage scores from the tensor product of $q$ matrices $U_1, U_2, \ldots, U_q$ without explicitly writing down $U_1 \odot U_2 \odot \cdots \odot U_q$. To build this algorithm we combine TensorSketch, some ideas from [DMIMW12], and some techniques from [AKO11, MW10]. Finally, we improve the running time for sampling columns according to the leverage scores from poly$(n)$ to $O(n)$. Given $q$ matrices $U_1, U_2, \ldots, U_q$, with each such matrix $U_i$ having size $k \times n_i$, we define $A \in \mathbb{R}^{k \times \prod_{i=1}^{q} n_i}$ to be the matrix where the $i$-th row of $A$ is the vectorization of $U_{i_1} \odot U_{i_2} \odot \cdots \odot U_{i_q}$, for all $i \in [k]$. Naively, in order to sample poly$(k, 1/\epsilon)$ rows from $A$ according to the leverage scores, we need to write down $\prod_{i=1}^{q} n_i$ leverage scores. This approach will take at least $\prod_{i=1}^{q} n_i$ running time. In the remainder of this section, we will explain how to do it in $O_q(n \cdot \text{poly}(k, 1/\epsilon))$ time for any constant $p$, and $\max_{i \in [q]} n_i \leq n$.

**Theorem C.46.** Given $q$ matrices $U_1 \in \mathbb{R}^{k \times n_1}$, $U_2 \in \mathbb{R}^{k \times n_2}$, $\ldots$, $U_q \in \mathbb{R}^{k \times n_q}$, let $\max_i n_i \leq n$. There exists an algorithm that takes $O_q(n \cdot \text{poly}(k, 1/\epsilon) \cdot R_{\text{samples}})$ time and samples $R_{\text{samples}}$ columns of $U_1 \odot U_2 \odot \cdots \odot U_q \in \mathbb{R}^{k \times \prod_{i=1}^{q} n_i}$ according to the leverage scores of $U_1 \odot U_2 \odot \cdots \odot U_q$.

**Proof.** Let $\max_i n_i \leq n$. First, choosing $\Pi_0$ to be a TensorSketch, we can compute $R^{-1}$ in $O_q(n \text{poly}(k, 1/\epsilon))$ time, where $R$ is the $R$ in a QR-factorization. We want to sample columns from $U_1 \odot U_2 \odot \cdots \odot U_q$ according to the square of the $\ell_2$-norm of each column of $R^{-1}(U_1 \odot U_2 \odot \cdots \odot U_q)$. The solution is to use

$$\sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} (B_{i,j,l}(x) - A_{i,j,l})^2 \leq C,$$

$$\|x\|_2^2 \leq 2^{O(n^δ)}.$$
**Algorithm 15** Fast Tensor Leverage Score Sampling, for General $q$-th Order

1: **procedure** FastTensorLeverageScoreGeneralOrder($\{U_i\}_{i \in [q]}, \{n_i\}_{i \in [q]}, k, \epsilon, R_{\text{samples}}$)  
   ▷ Theorem C.46

2: $s_1 \leftarrow \text{poly}(k, 1/\epsilon)$.

3: Choose $\Pi_0, \Pi_1 \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_q \times s_1}$ to each be a TensorSketch.  
   ▷ Definition B.34

4: Compute $R^{-1} \in \mathbb{R}^{k \times k}$ by using $(U_1 \odot U_2 \odot \cdots \odot U_q) \Pi_0$.  
   ▷ $U_i \in \mathbb{R}^{k \times n_i}, \forall i \in [q]$ 

5: $V_0 \leftarrow R^{-1}, n_0 \leftarrow k$.

6: **for** $i = 1 \rightarrow [n_0]$ **do**

7: \hspace{1em} $\alpha_i \leftarrow \| (V_0)^i ((U_1 \odot U_2 \odot \cdots \odot U_q) \Pi_1) \|^2_2$.

8: **end for**

9: **for** $r = 1 \rightarrow R_{\text{samples}}$ **do**

10: \hspace{1em} Sample $j_0$ from $[n_0]$ with probability $\alpha_i / \sum_{i'=1}^{n_q} \alpha_{i'}$.

11: \hspace{1em} **for** $l = 1 \rightarrow q - 1$ **do**

12: \hspace{2em} $s_{l+1} \leftarrow O_q(\text{poly}(k, 1/\epsilon))$.

13: \hspace{2em} Choose $\Pi_{l+1} \in \mathbb{R}^{n_{l+1} \times n_q \times s_{l+1}}$ to be a TensorSketch.

14: \hspace{2em} **for** $j_l = 1 \rightarrow [n_l]$ **do**  
   \hspace{3em} $\triangledown_j (V_l)^i (U_l+1 \odot \cdots \odot U_q) \Pi_{l+1}) \|^2_2$.  
   ▷ Form $V_l \in \mathbb{R}^{n_l \times k}$

15: \hspace{2em} **end for**

16: \hspace{1em} **for** $i = 1 \rightarrow n_q$ **do**

17: \hspace{2em} $\beta_i \leftarrow \| (V_q)^i (U_{l+1} \odot \cdots \odot U_q) \Pi_{l+1}) \|^2_2$.

18: \hspace{2em} **end for**

19: \hspace{1em} Sample $j_t$ from $[n_l]$ with probability $\beta_i / \sum_{i'=1}^{n_q} \beta_{i'}$.

20: **end for**

21: **for** $i = 1 \rightarrow n_q$ **do**

22: \hspace{2em} $\beta_i \leftarrow \| (V_{q-1})^{j_q-1} (U_q) \|^2_2$.

23: **end for**

24: **for** $q = 1 \rightarrow R_{\text{samples}}$ **do**

25: \hspace{2em} Sample $j_q$ from $[n_q]$ with probability $\beta_i / \sum_{i'=1}^{n_q} \beta_{i'}$.

26: \hspace{2em} $S \leftarrow S \cup (j_1, \cdots, j_q)$.

27: **end for**

28: Convert $S$ into a diagonal matrix $D$ with at most $R_{\text{samples}}$ nonzero entries.

29: **return** $D$.  
   ▷ Diagonal matrix $D \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_q \times n_1 \times n_2 \cdots n_q}$

30: **end procedure**

The issue is the number of columns of this matrix is already $\prod_{i=1}^{q} n_i$. The goal is to sample columns from $R^{-1}(U_1 \odot U_2 \odot \cdots \odot U_q)$ without explicitly computing the square of the $\ell_2$-norm of each column.

Similarly as in the proof of Lemma C.32, we have the observation that the following two sampling procedures are equivalent in terms of sampling a column of a matrix: (1) We sample a single entry from matrix $R^{-1}(U_1 \odot U_2 \odot \cdots \odot U_q)$ proportional to its squared value, (2) We sample a column from matrix $R^{-1}(U_1 \odot U_2 \odot \cdots \odot U_q)$ proportional to its squared $\ell_2$-norm. Let the $(i, j_1, j_2, \cdots, j_q)$-th entry denote the entry in the $i$-th row and the $j$-th column, where

$$j = \sum_{l=1}^{q-1} (j_l - 1) \prod_{t=l+1}^{q} n_t + j_q.$$

Similarly to Equation (18), we can show, for a particular column $j$,

$$\Pr[\text{we sample an entry from the } j\text{-th column of matrix}] = \Pr[\text{we sample the } j\text{-th column of a matrix}].$$
Thus, it is sufficient to show how to sample a single entry from matrix $R^{-1}(U_1 \odot U_2 \odot \cdots \odot U_q)$ proportional to its squared value without writing down all the entries of the $k \times \prod_{i=1}^q n_i$ matrix.

Let $V_0$ denote $R^{-1}$. Let $n_0$ denote the number of rows of $V_0$.

In the next few paragraphs, we describe a sampling procedure (procedure FastTensorLeverageScoreGeneralOrder in Algorithm 15) which first samples $\hat{j}_0$ from $[n_0]$, then samples $\hat{j}_1$ from $[n_1]$, $\cdots$, and at the end samples $\hat{j}_q$ from $[n_q]$.

In the first step, we want to sample $\hat{j}_0$ from $[n_0]$ proportional to the squared $\ell_2$-norm of that row. To do this efficiently, we choose $\Pi_1 \in \mathbb{R}^{n_1 \times 1}$ to be a TensorSketch to sketch on the right of $V_0(U_1 \odot U_2 \odot \cdots \odot U_q)$. By Section B.10, as long as $s_1 = O_q(\poly(k, 1/\epsilon))$, then $\Pi_1$ is a $(1 \pm \epsilon)$-subspace embedding matrix. Thus with probability $1 - 1/\Omega(q)$, for all $i \in [n_0]$,

$$\|(V_0)^i((U_1 \odot U_2 \odot \cdots \odot U_q)\Pi_1)\|^2_2 = (1 \pm \epsilon)\|(V_0)^i((U_1 \odot U_2 \odot \cdots \odot U_q))\|^2_2,$$

which means we can sample $\hat{j}_0$ from $[n_0]$ in $O_q(n \poly(k, 1/\epsilon))$ time.

In the second step, we have already obtained $\hat{j}_0$. Using that row of $V_0$ with $U_1$, we can form a new matrix $V_1 \in \mathbb{R}^{n_1 \times k}$ in the following sense,

$$(V_1)^i = (V_0)_{\hat{j}_0} \circ (U_1)^T, \forall i \in [n_1],$$

where $(V_1)^i$ denotes the $i$-th row of matrix $V_1$, $(V_0)_{\hat{j}_0}$ denotes the $\hat{j}_0$-th row of $V_0$ and $(U_1)_i$ is the $i$-th column of $U_1$. Another important observation is, the entry in the $(j_1, j_2, \cdots, j_q)$-th coordinate of vector $(V_0)_{\hat{j}_0}(U_1 \odot U_2 \odot \cdots \odot U_q)$ is the same as the entry in the $j_1$-th row and $(j_2, \cdots, j_q)$-th column of matrix $V_1(U_2 \odot U_3 \odot \cdots \odot U_q)$. Thus, sampling $j_1$ is equivalent to sampling $j_1$ from the new matrix $V_1(U_2 \odot U_3 \odot \cdots \odot U_q)$ proportional to the squared $\ell_2$-norm of that row. We still have the computational issue that the length of the row vector is very long. To deal with this, we can choose $\Pi_2 \in \mathbb{R}^{n_2 \times s_2}$ to be a TensorSketch to multiply on the right of $V_1(U_2 \odot U_3 \odot \cdots \odot U_q)$.

By Section B.10, as long as $s_2 = O_q(\poly(k, 1/\epsilon))$, then $\Pi_2$ is a $(1 \pm \epsilon)$-subspace embedding matrix. Thus with probability $1 - 1/\Omega(q)$, for all $i \in [n_1]$,

$$\|(V_1)^i((U_2 \odot \cdots \odot U_q)\Pi_2)\|^2_2 = (1 \pm \epsilon)\|(V_1)^i((U_2 \odot \cdots \odot U_q))\|^2_2,$$

which means we can sample $\hat{j}_1$ from $[n_1]$ in $O_q(n \poly(k, 1/\epsilon))$ time.

We repeat the above procedure until we obtain each of $\hat{j}_0, \hat{j}_1, \cdots, \hat{j}_q$. Note that the last one, $\hat{j}_q$, is easier, since the length of the vector is already small enough, and so we do not need to use TensorSketch for it.

By Section B.10, the time for multiplying by TensorSketch is $O_q(n \poly(k, 1/\epsilon))$. Setting $\epsilon$ to be a small constant, and taking a union bound over $O(q)$ events completes the proof.

**Lemma C.47.** Given $A \in \mathbb{R}^{n_0 \times \prod_{i=1}^q n_i}$, $U_1, U_2, \cdots, U_q \in \mathbb{R}^{k \times n}$, for any $\epsilon > 0$, there exists an algorithm that runs in $O(n \cdot \poly(k, 1/\epsilon))$ time and outputs a diagonal matrix $D \in \mathbb{R}^{\prod_{i=1}^q n_i \times n_i}$ with $m = O(k \log k + k/\epsilon)$ nonzero entries such that,

$$\|\hat{U}(U_1 \odot U_2 \odot \cdots \odot U_q) - A\|_F^2 \leq (1 + \epsilon) \min_{U \in \mathbb{R}^{n \times k}} \|U(U_1 \odot U_2 \odot \cdots \odot U_q) - A\|_F^2,$$

holds with probability at least 0.999, where $\hat{U}$ denotes the optimal solution of

$$\min_{U \in \mathbb{R}^{n_0 \times k}} \|U(U_1 \odot U_2 \odot \cdots \odot U_q) - AD\|_F^2.$$

**Proof.** This follows by combining Theorem C.46, Corollary C.30, and Lemma C.31. \qed
Algorithm 16 General $q$-th Order Iterative Existential Proof

1: procedure GENERALITERATIVEEXISTENTIALPROOF($A, n, k, q, \epsilon$) \Comment{Section C.10.2}
2: Fix $U^*_1, U^*_2, \cdots, U^*_q \in \mathbb{R}^{n \times k}$.
3: for $i = 1 \rightarrow q$ do
4:   Choose sketching matrix $S_i \in \mathbb{R}^{n^{q-1} \times s_i}$ with $s_i = O_q(k/\epsilon)$.
5:   Define $Z_i \in \mathbb{R}^{k \times n^{q-1}}$ to be $\bigcirc \hat{U}_i^\top \circ \circ U^{*\top}$.
6:   Let $A_i$ denote the matrix obtained by flattening tensor $A$ along the $i$-th dimension.
7:   Define $\hat{U}_i$ to be $A_i S_i (Z_i S_i)^\dagger$.
8: end for
9: return $\hat{U}_1, \hat{U}_2, \cdots, \hat{U}_q$.
10: end procedure

C.10.2 General iterative existential proof

Given a $q$-th order tensor $A \in \mathbb{R}^{n \times n \times \cdots \times n}$, we fix $U^*_1, U^*_2, \cdots, U^*_q \in \mathbb{R}^{n \times k}$ to be the best rank-$k$ solution (if it does not exist, then we replace it by a good approximation, as discussed). We define $\text{OPT} = \|U^*_1 \otimes U^*_2 \otimes \cdots \otimes U^*_q - A\|_F^2$. Our iterative proof works as follows. We first obtain the objective function,

$$\min_{U_1 \in \mathbb{R}^{n \times k}} \|U_1 \cdot Z_1 - A_1\|_F^2 \leq \text{OPT},$$

where $A_1$ is a matrix obtained by flattening tensor $A$ along the first dimension, $Z_1 = (U_2^{*\top} \otimes U_3^{*\top} \otimes \cdots \otimes U_q^{*\top})$ denotes a $k \times n^{q-1}$ matrix. Choosing $S_1 \in \mathbb{R}^{n^{q-1} \times s_1}$ to be a Gaussian sketching matrix with $s_1 = O(k/\epsilon)$, we obtain a smaller problem,

$$\min_{U_1 \in \mathbb{R}^{n \times k}} \|U_1 \cdot Z_1 S_1 - A_1 S_1\|_F^2.$$

We define $\hat{U}_1$ to be $A_1 S_1 (Z_1 S_1)^\dagger \in \mathbb{R}^{n \times k}$, which gives,

$$\|\hat{U}_1 \cdot Z_1 - A_1\|_F^2 \leq (1 + \epsilon) \text{OPT}.$$

After retensorizing the above, we have,

$$\|\hat{U}_1 \otimes U_2^* \otimes \cdots \otimes U_q^* - A\|_F^2 \leq (1 + \epsilon) \text{OPT}.$$

In the second round, we fix $\hat{U}_1, U_3^*, \cdots, U_q^* \in \mathbb{R}^{n \times k}$, and choose $S_2 \in \mathbb{R}^{n^{q-1} \times s_2}$ to be a Gaussian sketching matrix with $s_2 = O(k/\epsilon)$. We define $Z_2 \in \mathbb{R}^{k \times n^{q-1}}$ to be $(\hat{U}_1^\top \otimes U_3^{*\top} \otimes \cdots \otimes U_q^{*\top})$. We define $\hat{U}_2$ to be $A_2 S_2 (Z_2 S_2)^\dagger \in \mathbb{R}^{n \times k}$. Then, we have

$$\|\hat{U}_1 \otimes \hat{U}_2 \otimes U_3^* \otimes \cdots \otimes U_q^* - A\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.$$

We repeat the above process, where in the $i$-th round we fix $\hat{U}_1, \cdots, \hat{U}_{i-1}, U_{i+1}^*, \cdots, U_q^* \in \mathbb{R}^{n \times k}$, and choose $S_i \in \mathbb{R}^{n^{q-1} \times s_i}$ to be a Gaussian sketching matrix with $s_i = O(k/\epsilon)$. We define $Z_i \in \mathbb{R}^{k \times n^{q-1}}$ to be $(\hat{U}_1^\top \otimes \cdots \otimes \hat{U}_{i-1}^\top \otimes U_{i+1}^{*\top} \otimes \cdots \otimes U_q^{*\top})$. We define $\hat{U}_i$ to be $A_i S_i (Z_i S_i)^\dagger \in \mathbb{R}^{n \times k}$. Then, we have

$$\|\hat{U}_1 \otimes \cdots \otimes \hat{U}_{i-1} \otimes \hat{U}_i \otimes U_{i+1}^* \otimes \cdots \otimes U_q^* - A\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.$$
At the end of the $q$-th round, we have

$$\|\hat{U}_1 \otimes \cdots \otimes \hat{U}_q - A\|_F^2 \leq (1 + \epsilon)^q \text{OPT}. $$

Replacing $\epsilon = \epsilon'/(2q)$, we obtain

$$\|\hat{U}_1 \otimes \cdots \otimes \hat{U}_q - A\|_F^2 \leq (1 + \epsilon') \text{OPT}. $$

where for all $i \in [q]$, $s_i = O(kq/\epsilon') = \text{O}_q(k/\epsilon')$.

### C.10.3 General input sparsity reduction

This section shows how to extend the input sparsity reduction from third order tensors to general $q$-th order tensors. Given a tensor $A \in \mathbb{R}^{n \times n \times \cdots \times n}$ and $q$ matrices, for each $i \in [q]$, matrix $V_i$ has size $V_i \in \mathbb{R}^{n \times b_i}$, with $b_i \leq \text{poly}(k, 1/\epsilon)$. We choose a batch of sparse embedding matrices $T_i \in \mathbb{R}^{t_i \times n}$. Define $V_i = T_i V_i$, and $C = A(T_1, T_2, \cdots, T_q)$. Thus we have with probability $99/100$, for any $\alpha \geq 0$, for all $\{X_i, X_i' \in \mathbb{R}^{b_i \times k}\}_{i \in [q]}$, if

$$\|\hat{V}_1 X_1' \otimes \hat{V}_2 X_2' \otimes \cdots \otimes \hat{V}_q X_q' - C\|_F^2 \leq \alpha \|\hat{V}_1 X_1 \otimes \hat{V}_2 X_2 \otimes \cdots \otimes \hat{V}_q X_q - C\|_F^2,$$

then

$$\|V_1 X_1 \otimes V_2 X_2 \otimes \cdots \otimes V_q X_q - A\|_F^2 \leq (1 + \epsilon)\alpha \|V_1 X_1 \otimes V_2 X_2 \otimes \cdots \otimes V_q X_q - A\|_F^2,$$

where $t_i = \text{O}_q(\text{poly}(b_i, 1/\epsilon))$.

#### Algorithm 17 General $q$-th Order Input Sparsity Reduction

1: procedure GENERALINPUTSPARSITYREDUCTION($A, \{V_i\}_{i \in [q]}, n, k, q, \epsilon$) ▷ Section C.10.3
2: for $i = 1 \rightarrow q$
3:     Choose sketching matrix $T_i \in \mathbb{R}^{t_i \times n}$ with $t_i = \text{poly}(k, q, 1/\epsilon)$.
4:     $\hat{V}_i \leftarrow T_i V_i$.
5: end for
6: $C \leftarrow A(T_1, T_2, \cdots, T_q)$.
7: return $\{\hat{V}_i\}_{i \in [q]}, C$.
8: end procedure

### C.10.4 Bicriteria algorithm

This section explains how to extend the bicriteria algorithm from third order tensors (Section C.4) to general $q$-th order tensors. Given any $q$-th order tensor $A \in \mathbb{R}^{n \times n \times \cdots \times n}$, we can output a rank-$r$ tensor (or equivalently $q$ matrices $U_1, U_2, \cdots, U_q \in \mathbb{R}^{n \times r}$) such that,

$$\|U_1 \otimes U_2 \otimes \cdots \otimes U_q - A\|_F^2 \leq (1 + \epsilon) \text{OPT},$$

where $r = \text{O}_q((k/\epsilon)^{q-1})$ and the algorithm takes $\text{O}_q(n \text{nnz}(A) + n \cdot \text{poly}(k, 1/\epsilon))$. 

74
Algorithm 18 General \(q\)-th Order Bicriteria Algorithm

1: procedure \textsc{GeneralBicriteriaAlgorithm}(\(A, n, k, q, \epsilon\)) \hfill \textsection C.10.4
2: \hspace{1em} for \(i = 2 \rightarrow q\) do
3: \hspace{2em} Choose sketching matrix \(S_i \in \mathbb{R}^{n^{q-1} \times s_i}\) with \(s_i = O(kq/\epsilon)\).
4: \hspace{2em} Choose sketching matrix \(T_i \in \mathbb{R}^{t_i \times n}\) with \(t_i = \text{poly}(k, q, 1/\epsilon)\).
5: \hspace{2em} Form matrix \(\hat{U}_i\) by setting \((j_2, j_3, \ldots, j_q)\)-th column to be \((A_i S_i)_{j_i}\).
6: \hspace{1em} end for
7: Solve \(\min_{U_1} \|U_1 B - (A(I, T_2, \ldots, T_q))_1\|_F^2\).
8: return \(\{\hat{U}_i\}_{i \in [q]}\).
9: end procedure

C.10.5 CURT decomposition

This section extends the tensor CURT algorithm from 3rd order tensors (Section C.7) to general \(q\)-th order tensors. Given a \(q\)-th order tensor \(A \in \mathbb{R}^{n \times n \times \cdots \times n}\) and a batch of matrices \(U_1, U_2, \ldots, U_q \in \mathbb{R}^{n \times k}\), we iteratively apply the proof in Theorem C.40 (or Theorem C.41) \(q\) times. Then for each \(i \in [q]\), we are able to select \(d_i\) columns from the \(i\)-th dimension of tensor \(A\) (let \(C_i\) denote those columns) and also find a tensor \(\hat{U} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_q}\) such that,

\[
\|U(C_1, C_2, \ldots, C_q) - A\|_F^2 \leq (1 + \epsilon)\|U_1 \otimes U_2 \otimes \cdots \otimes U_q - A\|_F^2,
\]

where either \(d_i = O_q(k \log k + k/\epsilon)\) (similar to Theorem C.40) or \(d_i = O_q(k/\epsilon)\) (similar to Theorem C.41).

Algorithm 19 General \(q\)-th Order CURT Decomposition

1: procedure \textsc{GeneralCURTDecomposition}(\(A, \{U_i\}_{i \in [q]}, n, k, q, \epsilon\)) \hfill \textsection C.10.5
2: \hspace{1em} for \(i = 1 \rightarrow q\) do
3: \hspace{2em} Form \(B_i = \otimes_{j < i} \hat{U}_j^T \otimes \otimes_{j > i} U_j^T \in \mathbb{R}^{k \times n^{q-1}}\).
4: \hspace{2em} if \hspace{0.5em} fast = \text{true} \hspace{0.5em} then \hspace{1em} \text{\hfill \textsection Optimal running time}
5: \hspace{3em} \(\epsilon_0 \leftarrow 0.01\).
6: \hspace{3em} \(d_i \leftarrow O_q(k \log k + k/\epsilon)\).
7: \hspace{3em} \(D_i \leftarrow \text{\textsc{FastTensorLeverageScoreGeneralOrder}}\) \((\{\hat{U}_j\}_{j < i}; \{U_j\}_{j > i}; n, k, \epsilon_0, d_i)\). \hfill \textsection Algorithm 15
8: \hspace{2em} else \hspace{1em} \text{\hfill \textsection Optimal sample complexity}
9: \hspace{3em} \(\epsilon_0 \leftarrow O_q(\epsilon)\).
10: \hspace{3em} \(D_i \leftarrow \text{\textsc{GeneralizedMatrixRowSubsetSelection}}\) \((A_i^T; B_i^T; n^{q-1}, n, k, \epsilon_0)\). \hfill \textsection Algorithm C.5, \(d_i = O_q(k/\epsilon)\).
11: \hspace{2em} end if
12: \hspace{2em} \(\hat{U}_i \leftarrow A_i D_i (B_i D_i)^\dagger\).
13: \hspace{2em} \(C_i \leftarrow A_i D_i\).
14: \hspace{1em} end for
15: \(U \leftarrow (B_1 D_1)^\dagger \otimes (B_2 D_2)^\dagger \otimes \cdots \otimes (B_q D_q)^\dagger\).
16: return \(\{C_i\}_{i \in [q]}; U\).
17: end procedure
C.11 Matrix CUR decomposition

There is a long line of research on matrix CUR decomposition under operator, Frobenius or recently, entry-wise $\ell_1$ norm [DMM08, BMD09, DR10, BDM11, BW14, SWZ17]. We provide the first algorithm that runs in $\text{nnz}(A)$ time, which improves the previous best matrix CUR decomposition algorithm under Frobenius norm [BW14].

C.11.1 Algorithm

Algorithm 20 Optimal Matrix CUR Decomposition Algorithm

\begin{algorithm}
\begin{algorithmic}[1]
\Procedure{OptimalMatrixCUR}{A, n, k, $\epsilon$} \Comment{Theorem C.48}
\State \textbf{procedure} \textbf{OptimalMatrixCUR}(A, n, k, $\epsilon$)
\State $\epsilon' \leftarrow 0.1\epsilon$, $\epsilon'' \leftarrow 0.001\epsilon$.
\State $\hat{U} \leftarrow \text{SparseSVD}(A, k, \epsilon')$. \Comment{$\hat{U} \in \mathbb{R}^{n \times k}$}
\State Choose $S_1 \in \mathbb{R}^{n \times n}$ to be a sampling and rescaling diagonal matrix according to the leverage scores of $\hat{U}$ with $s_1 = O(\epsilon^{-2}k \log k)$ nonzero entries.
\State $R, Y \leftarrow \text{GeneralizedMatrixRowSubsetSelection}(S_1 A, S_1 \hat{U}, s_1, n, k, \epsilon'')$. \Comment{$\hat{U}$}
\State \textbf{Algorithm 7}, $R \in \mathbb{R}^{r \times n}, Y \in \mathbb{R}^{k \times r}$ and $r = O(k/\epsilon)$
\State $\hat{V} \leftarrow Y R \in \mathbb{R}^{k \times n}$.
\State Choose $S_2 \in \mathbb{R}^{n \times n}$ to be a sampling and rescaling diagonal matrix according to the leverage scores of $\hat{V}^T \in \mathbb{R}^{n \times k}$ with $s_2 = O(\epsilon^{-2}k \log k)$ nonzero entries.
\State $C^T, Z^T \leftarrow \text{GeneralizedMatrixRowSubsetSelection }((A S_2)^T, (\hat{V} S_2)^T, s_2, n, k, \epsilon'')$. \Comment{$\hat{U}$}
\State \textbf{Algorithm 7}, $C \in \mathbb{R}^{n \times c}, Z \in \mathbb{R}^{c \times k}$, and $c = O(k/\epsilon)$
\State $U \leftarrow Z Y$. \Comment{$U \in \mathbb{R}^{c \times r}$ and rank$(U) = k$}
\State \Return $C, U, R$.
\EndProcedure
\end{algorithmic}
\end{algorithm}

Theorem C.48. Given matrix $A \in \mathbb{R}^{n \times n}$, for any $k \geq 1$ and $\epsilon \in (0, 1)$, there exists an algorithm that takes $O(\text{nnz}(A) + n \text{ poly}(k, 1/\epsilon))$ time and outputs three matrices $C \in \mathbb{R}^{n \times c}$ with $c$ columns from $A$, $R \in \mathbb{R}^{r \times n}$ with $r$ rows from $A$, and $U \in \mathbb{R}^{c \times r}$ with rank$(U) = k$ such that $r = c = O(k/\epsilon)$ and,

$$\|\text{CUR} - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank} - k A_k} \|A_k - A\|_F^2,$$

holds with probability at least 9/10.

Proof. We define

$$\text{OPT} = \min_{\text{rank} - k A_k} \|A_k - A\|_F^2.$$

We first compute $\hat{U} \in \mathbb{R}^{n \times k}$ by using the result of [CW13], so that $\hat{U}$ satisfies:

$$\min_{X \in \mathbb{R}^{k \times n}} \|\hat{U} X - A\|_F^2 \leq (1 + \epsilon) \text{OPT}. \quad (25)$$

This step can be done in $O(\text{nnz}(A) + n \text{ poly}(k, 1/\epsilon))$ time.

We choose $S_1 \in \mathbb{R}^{n \times n}$ to be a sampling and rescaling diagonal matrix according to the leverage scores of $\hat{U}$, where here $s_1 = O(\epsilon^{-2}k \log k)$ is the number of samples. This step also can be done in $O(n \text{ poly}(k, 1/\epsilon))$ time.

76
We run GeneralizedMatrixRowSubsetSelection (Algorithm 7) on matrices $S_1A$ and $S_1\hat{U}$. Then we obtain two new matrices $R$ and $Y$, where $R$ contains $r = O(k/\epsilon)$ rows of $S_1A$ and $Y$ has size $k \times r$. According to Theorem C.14 and Corollary C.15, this step takes $n \operatorname{poly}(k, 1/\epsilon)$ time.

We construct $\hat{V} = YR$, and choose $S_2^\top$ to be another sampling and rescaling diagonal matrix according to the leverage scores of $\hat{V}^\top$ with $s_2 = O(\epsilon^{-2}k\log k)$ nonzero entries. As in the case of constructing $S_1$, this step can be done in $O(n \operatorname{poly}(k, 1/\epsilon))$ time.

We run GeneralizedMatrixRowSubsetSelection (Algorithm 7) on matrices $(AS_2)^\top$ and $(\hat{V}S_2)^\top$. Then we can obtain two new matrices $C^\top$ and $Y^\top$, where $C^\top$ contains $c = O(k/\epsilon)$ rows of $(AS_2)^\top$ and $Z^\top$ has size $k \times c$. According to Theorem C.14 and Corollary C.15, this step takes $n \operatorname{poly}(k, 1/\epsilon)$ time.

Thus, overall the running time is $O(n \operatorname{nnz}(A) + n \operatorname{poly}(k, 1/\epsilon))$.

**Correctness.** Let

$$X^* = \arg \min_{X \in \mathbb{R}^{n \times k}} \|X\hat{V} - A\|_F^2.$$ 

According to Corollary C.15,

$$\|CZ\hat{V}S_2 - AS_2\|_F^2 \leq (1 + \epsilon'') \min_{X \in \mathbb{R}^{n \times k}} \|X\hat{V}S_2 - AS_2\|_F^2 \leq (1 + \epsilon'') \|X^*\hat{V}S_2 - AS_2\|_F^2.$$ 

According to Theorem C.52, $\epsilon'' = 0.001\epsilon'$,

$$\|CZ\hat{V} - A\|_F^2 \leq (1 + \epsilon') \|X^*\hat{V} - A\|_F^2.$$ 

Let

$$\tilde{X} = \arg \min_{X \in \mathbb{R}^{k \times n}} \|\hat{U}X - A\|_F^2.$$ 

According to Corollary C.15,

$$\|S_1\tilde{U}YR - S_1A\|_F^2 \leq (1 + \epsilon'') \min_{X \in \mathbb{R}^{k \times n}} \|S_1\tilde{U}X - S_1A\|_F^2 \leq (1 + \epsilon'') \|S_1\tilde{U}\tilde{X} - S_1A\|_F^2.$$ 

According to Theorem C.52, since $\epsilon'' = 0.001\epsilon'$,

$$\|\tilde{U}YR - A\|_F^2 \leq (1 + \epsilon') \|\tilde{U}\tilde{X} - A\|_F^2.$$ 

Then, we can conclude

$$\|CUR - A\|_F^2 = \|CYR - A\|_F^2 = \|CZ\hat{V} - A\|_F^2 \leq (1 + \epsilon') \|X\hat{V} - A\|_F^2 \leq (1 + \epsilon') \|\hat{U}\hat{V} - A\|_F^2 \leq (1 + \epsilon')^2 \min_{X \in \mathbb{R}^{k \times n}} \|\hat{U}X - A\|_F^2 \leq (1 + \epsilon')^3 \operatorname{OPT} \leq (1 + \epsilon) \operatorname{OPT}.$$
The first equality follows since \( U = ZY \). The second equality follows since \( YR = \hat{V} \). The first inequality follows by Equation (26). The third inequality follows by Equation (27). The fourth inequality follows by Equation (25). The last inequality follows since \( \epsilon' = 0.1\epsilon \).

Notice that \( C \) has \( O(k/\epsilon) \) reweighted columns of \( AS_2 \), and \( AS_2 \) is a subset of reweighted columns of \( A \), so \( C \) has \( O(k/\epsilon) \) reweighted columns of \( A \). Similarly, we can prove that \( R \) has \( O(k/\epsilon) \) reweighted rows of \( A \). Thus, \( CUR \) is a CUR decomposition of \( A \).

\[ \square \]

C.11.2 Stronger property achieved by leverage scores

Claim C.49. Given matrix \( A \in \mathbb{R}^{n \times m} \), for any distribution \( p = (p_1, p_2, \ldots, p_n) \) define random variable \( X \) such that \( X = \|A_i\|^2/p_i \) with probability \( p_i \), where \( A_i \) is the \( i \)-th row of matrix \( A \). Then take \( m \) independent samples \( X^1, X^2, \ldots, X^m \), and let \( Y = \frac{1}{m} \sum_{j=1}^{m} X^j \). We have

\[ \Pr[Y \leq 100\|A\|_F^2] \geq .99. \]

Proof. We can compute the expectation of \( X^j \), for any \( j \in [m] \),

\[ E[X^j] = \sum_{i=1}^{n} \frac{\|A_i\|^2}{p_i} \cdot p_i = \|A\|_F^2. \]

Then \( E[Y] = \frac{1}{m} \sum_{j=1}^{m} E[X^j] = \|A\|_F^2 \). Using Markov’s inequality, we have

\[ \Pr[Y \geq 2\|A\|_F^2] \leq .01. \]

\[ \square \]

Theorem C.50 (The leverage score case of Theorem 39 in [CW13]). Let \( A \in \mathbb{R}^{n \times k} \), \( B \in \mathbb{R}^{n \times d} \). Let \( S \in \mathbb{R}^{n \times n} \) denote a sampling and rescaling diagonal matrix according to the leverage scores of \( A \). If the event occurs that \( S \) satisfies \((\epsilon/\sqrt{k})\)-Frobenius norm approximate matrix product for \( A \), and also \( S \) is a \((1 + \epsilon)\)-subspace embedding for \( A \), then let \( X^* \) be the optimal solution of \( \min_X \|AX - B\|_F^2 \), and \( \tilde{B} = AX^* - B \). Then, for all \( X \in \mathbb{R}^{k \times d} \),

\[ (1 - 2\epsilon)\|AX - B\|_F^2 \leq \|S(AX - B)\|_F^2 + \|\tilde{B}\|_F^2 - \|S\tilde{B}\|_F^2 \leq (1 + 2\epsilon)\|AX - B\|_F^2. \]

Furthermore, if \( S \) has \( m = O(\epsilon^{-2}k \log(k)) \) nonzero entries, the above event happens with probability at least 0.99.

Note that Theorem 39 in [CW13] is stated in a way that holds for general sketching matrices. However, we are only interested in the case when \( S \) is a sampling and rescaling diagonal matrix according to the leverage scores. For completeness, we provide the full proof of the leverage score case with certain parameters.

Proof. Suppose \( S \) is a sampling and rescaling diagonal matrix according to the leverage scores of \( A \), and it has \( m = O(\epsilon^{-2}k \log k) \) nonzero entries. Then, according to Lemma C.22, \( S \) is a \((1 + \epsilon)\)-subspace embedding for \( A \) with probability at least 0.999, and according to Lemma C.29, \( S \) satisfies \((\epsilon/\sqrt{k})\)-Frobenius norm approximate matrix product for \( A \) with probability at least 0.999.

Let \( U \in \mathbb{R}^{n \times k} \) denote an orthonormal basis of the column span of \( A \). Then the leverage scores of \( U \) are the same as the leverage scores of \( A \). Furthermore, for any \( X \in \mathbb{R}^{k \times d} \), there is a matrix \( Y \) such that \( AX = UY \), and vice versa, so we can now assume \( A \) has \( k \) orthonormal columns.

78
Then,
\[
\|S(AX - B)\|_F^2 - \|S\tilde{B}\|_F^2 = \|SA(X - X^*) + S(AX^* - B)\|_F^2 - \|S\tilde{B}\|_F^2 = \|SA(X - X^*)\|_F^2 + \|S(AX^* - B)\|_F^2 + 2\text{tr}\left((X - X^*)^\top A^\top S^\top S(AX^* - B)\right) - \|S\tilde{B}\|_F^2
\]
(28)

The second equality follows using \(\|C + D\|_F^2 = \|C\|_F^2 + \|D\|_F^2 + 2\text{tr}(C^\top D)\). The third equality follows from \(\tilde{B} = AX^* - B\). Now, let us first upper bound the term \(\alpha\) in Equation (28):
\[
\|SA(X - X^*)\|_F^2 + 2\text{tr}\left((X - X^*)^\top A^\top S^\top S\tilde{B}\right)
\leq (1 + \epsilon)\|A(X - X^*)\|_F^2 + 2\|X - X^*\|_F\|A^\top S^\top S\tilde{B}\|_F
\leq (1 + \epsilon)\|A(X - X^*)\|_F^2 + 2(\epsilon/\sqrt{k}) \cdot \|X - X^*\|_F\|A\|_F\|\tilde{B}\|_F
\leq (1 + \epsilon)\|A(X - X^*)\|_F^2 + 2\epsilon\|A(X - X^*)\|_F\|\tilde{B}\|_F.
\]
The first inequality follows since \(S\) is a \((1 + \epsilon)\) subspace embedding of \(A\), and \(\text{tr}(C^\top D) \leq \|C\|_F\|D\|_F\). The second inequality follows since \(S\) satisfies \((\epsilon/\sqrt{k})\)-Frobenius norm approximate matrix product for \(A\). The last inequality follows using that \(\|A\|_F \leq \sqrt{k}\) since \(A\) only has \(k\) orthonormal columns.

Now, let us lower bound the term \(\alpha\) in Equation (28):
\[
\|SA(X - X^*)\|_F^2 + 2\text{tr}\left((X - X^*)^\top A^\top S^\top S\tilde{B}\right)
\geq (1 - \epsilon)\|A(X - X^*)\|_F^2 - 2\|X - X^*\|_F\|A^\top S^\top S\tilde{B}\|_F
\geq (1 - \epsilon)\|A(X - X^*)\|_F^2 - 2(\epsilon/\sqrt{k}) \cdot \|X - X^*\|_F\|A\|_F\|\tilde{B}\|_F
\geq (1 - \epsilon)\|A(X - X^*)\|_F^2 - 2\epsilon\|A(X - X^*)\|_F\|\tilde{B}\|_F.
\]
The first inequality follows since \(S\) is a \((1 + \epsilon)\) subspace embedding of \(A\), and \(\text{tr}(C^\top D) \geq -\|C\|_F\|D\|_F\). The second inequality follows since \(S\) satisfies \((\epsilon/\sqrt{k})\)-Frobenius norm approximate matrix product for \(A\). The last inequality follows using that \(\|A\|_F \leq \sqrt{k}\) since \(A\) only has \(k\) orthonormal columns.

Therefore,
\[
(1 - \epsilon)\|A(X - X^*)\|_F^2 - 2\epsilon\|A(X - X^*)\|_F\|\tilde{B}\|_F \leq \|S(AX - B)\|_F^2 - \|S\tilde{B}\|_F^2
\]
(29)

and
\[
(1 + \epsilon)\|A(X - X^*)\|_F^2 + 2\epsilon\|A(X - X^*)\|_F\|\tilde{B}\|_F \geq \|S(AX - B)\|_F^2 - \|S\tilde{B}\|_F^2.
\]
(30)

Notice that \(\tilde{B} = AX^* - B = AA^\top B - B = (AA^\top - I)B\), so according to the Pythagorean theorem, we have
\[
\|AX - B\|_F^2 = \|A(X - X^*)\|_F^2 + \|\tilde{B}\|_F^2,
\]
which means that
\[
\|A(X - X^*)\|_F^2 = \|AX - B\|_F^2 - \|\tilde{B}\|_F^2.
\]
(31)
Using Equation (31), we can rewrite and lower bound the LHS of Equation (29),
\[
(1 - \epsilon)\|A(X - X^*)\|_F^2 - 2\epsilon\|A(X - X^*)\|_F \|\tilde{B}\|_F \\
= \|A(X - X^*)\|_F^2 - \epsilon \left(\|A(X - X^*)\|_F^2 + 2\|A(X - X^*)\|_F \|\tilde{B}\|_F\right) \\
= \|AX - B\|_F^2 - \|\tilde{B}\|_F^2 - \epsilon \left(\|A(X - X^*)\|_F^2 + \|\tilde{B}\|_F^2\right) \\
\geq \|AX - B\|_F^2 - \|\tilde{B}\|_F^2 - \epsilon \left(\|A(X - X^*)\|_F^2 + \|\tilde{B}\|_F^2\right) \\
= (1 - 2\epsilon)\|AX - B\|_F^2 - \|\tilde{B}\|_F^2. \tag{32}
\]
The second step follows by Equation (31). The first inequality follows using \(a^2 + 2ab < (a + b)^2\). The second inequality follows using \((a + b)^2 \leq 2(a^2 + b^2)\). The last equality follows using \(\|A(X - X^*)\|_F^2 + \|\tilde{B}\|_F^2 = \|AX - B\|_F^2\). Similarly, using Equation (31), we can rewrite and upper bound the LHS of Equation (30)
\[
(1 + \epsilon)\|A(X - X^*)\|_F^2 + 2\epsilon\|A(X - X^*)\|_F \|\tilde{B}\|_F \leq (1 + 2\epsilon)\|AX - B\|_F^2 - \|\tilde{B}\|_F^2. \tag{33}
\]
Combining Equations (29),(32),(30),(33), we conclude that
\[
(1 - 2\epsilon)\|AX - B\|_F^2 - \|\tilde{B}\|_F^2 \leq \|S(AX - B)\|_F^2 - \|S\tilde{B}\|_F^2 \leq (1 + 2\epsilon)\|AX - B\|_F^2 - \|\tilde{B}\|_F^2.
\]

\[\square\]

**Theorem C.51.** Let \(A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{n \times d}\), and \(\frac{1}{2} > \epsilon > 0\). Let \(X^*\) be the optimal solution to \(\min_X \|AX - B\|_F^2\), and \(\tilde{B} \equiv AX^* - B\). Let \(S \in \mathbb{R}^{n \times n}\) denote a sketching matrix which satisfies the following:

1. \(\|S\tilde{B}\|_F^2 \leq 100 \cdot \|\tilde{B}\|_F^2\),
2. for all \(X \in \mathbb{R}^{k \times d}\),
   \[
   (1 - \epsilon)\|AX - B\|_F^2 \leq \|S(AX - B)\|_F^2 + \|\tilde{B}\|_F^2 - \|S\tilde{B}\|_F^2 \leq (1 + \epsilon)\|AX - B\|_F^2.
   \]

Then, for all \(X_1, X_2 \in \mathbb{R}^{k \times d}\) satisfying
\[
\|SAX_1 - SB\|_F^2 \leq \left(1 + \frac{\epsilon}{100}\right) \cdot \|SAX_2 - SB\|_F^2,
\]
we have
\[
\|AX_1 - B\|_F^2 \leq (1 + 5\epsilon) \cdot \|AX_2 - B\|_F^2.
\]

**Proof.** Let \(A, B, S, \epsilon\) be the same as in the statement of the theorem, and suppose \(S\) satisfies those two conditions. Let \(X_1, X_2 \in \mathbb{R}^{k \times d}\) satisfy
\[
\|SAX_1 - SB\|_F^2 \leq \left(1 + \frac{\epsilon}{100}\right) \|SAX_2 - SB\|_F^2.
\]
We have
\[
\|AX_1 - B\|_F^2
\leq \frac{1}{1 - \epsilon} \left( \|S(AX_1 - B)\|_F^2 + \|	ilde{B}\|_F^2 - \|SB\|_F^2 \right)
\leq \frac{1}{1 - \epsilon} \left( (1 + \frac{\epsilon}{100}) \cdot \|S(AX_2 - B)\|_F^2 + \|	ilde{B}\|_F^2 - \|SB\|_F^2 \right)
= \frac{1}{1 - \epsilon} \left( (1 + \frac{\epsilon}{100}) \cdot \left( \|S(AX_2 - B)\|_F^2 + \|	ilde{B}\|_F^2 - \|SB\|_F^2 \right) - \frac{\epsilon}{100} \cdot \left( \|	ilde{B}\|_F^2 - \|SB\|_F^2 \right) \right)
\leq \frac{1}{1 - \epsilon} \cdot (1 + \frac{\epsilon}{100}) \cdot \|AX_2 - B\|_F^2 - \frac{1}{1 - \epsilon} \cdot \frac{\epsilon}{100} \cdot \left( \|	ilde{B}\|_F^2 - \|SB\|_F^2 \right)
\leq (1 + 3\epsilon)\|AX_2 - B\|_F^2 + \frac{1}{1 - \epsilon} \cdot \frac{\epsilon}{100} \cdot \|SB\|_F^2.
\]

The first inequality follows since $S$ satisfies the second condition. The second inequality follows by the relationship between $X_1$ and $X_2$. The third inequality follows since $S$ satisfies the second condition. The fifth inequality follows using that $\epsilon < \frac{1}{2}$ and that $S$ satisfies the first condition. The last inequality follows using that $\|\tilde{B}\|_F^2 = \|AX^* - B\|_F^2 \leq \|AX_2 - B\|_F^2$. \qed

**Theorem C.52.** Let $A \in \mathbb{R}^{n \times k}$, $B \in \mathbb{R}^{n \times d}$, and $\frac{1}{2} > \epsilon > 0$. Let $S \in \mathbb{R}^{n \times n}$ denote a sampling and rescaling diagonal matrix according to the leverage scores of $A$. If $S$ has at least $m = O(k \log(k)/\epsilon^2)$ nonzero entries, then with probability at least 0.98, for all $X_1, X_2 \in \mathbb{R}^{k \times d}$ satisfying
\[
\|SAX_1 - SB\|_F^2 \leq (1 + \frac{\epsilon}{500}) \cdot \|SAX_2 - SB\|_F^2,
\]
we have
\[
\|AX_1 - B\|_F^2 \leq (1 + \epsilon) \cdot \|AX_2 - B\|_F^2.
\]

**Proof.** The proof directly follows by Claim C.49, Theorem C.50 and Theorem C.51. Because of Claim C.49, $S$ satisfies the first condition in the statement of Theorem C.51 with probability at least 0.99. According to Theorem C.50, $S$ satisfies the second condition in the statement of Theorem C.51 with probability at least 0.99. Thus, with probability 0.98, by Theorem C.51, we complete the proof. \qed
D Entry-wise $\ell_1$ Norm for Arbitrary Tensors

In this section, we provide several different algorithms for tensor $\ell_1$-low rank approximation. Section D.1 provides some useful facts and definitions. Section D.2 presents several existence results. Section D.3 describes a tool that is able to reduce the size of the objective function from poly$(n)$ to poly$(k)$. Section D.4 discusses the case when the problem size is small. Section D.5 provides several bicriteria algorithms. Section D.6 summarises a batch of algorithms. Section D.7 provides an algorithm for $\ell_1$ norm CUR decomposition.

Notice that if the rank $- k$ solution does not exist, then every bicriteria algorithm in Section D.5 can be stated in a form similar to Theorem 1.1, and every algorithm which can output a rank $- k$ solution in Section D.6 can be stated in a form similar to Theorem 1.2. See Section 1 for more details.

D.1 Facts

We present a method that is able to reduce the entry-wise $\ell_1$-norm objective function to the Frobenius norm objective function.

**Fact D.1.** Given a 3rd order tensor $C \in \mathbb{R}^{c_1 \times c_2 \times c_3}$, three matrices $V_1 \in \mathbb{R}^{c_1 \times b_1}$, $V_2 \in \mathbb{R}^{c_2 \times b_2}$, $V_3 \in \mathbb{R}^{c_3 \times b_3}$, for any $k \in [1, \min_i b_i]$, if $X_1' \in \mathbb{R}^{b_1 \times k}$, $X_2' \in \mathbb{R}^{b_2 \times k}$, $X_3' \in \mathbb{R}^{b_3 \times k}$ satisfies that,

$$\|(V_1' X_1') \otimes (V_2' X_2') \otimes (V_3' X_3') - C\|_F \leq \alpha \min_{X_1, X_2, X_3} \|(V_1 X_1) \otimes (V_2 X_2) \otimes (V_3 X_3) - C\|_F,$$

then

$$\|(V_1' X_1') \otimes (V_2' X_2') \otimes (V_3' X_3') - C\|_1 \leq \alpha \sqrt{c_1 c_2 c_3} \min_{X_1, X_2, X_3} \|(V_1 X_1) \otimes (V_2 X_2) \otimes (V_3 X_3) - C\|_1.$$

We extend Lemma C.15 in [SWZ17] to tensors:

**Fact D.2.** Given tensor $A \in \mathbb{R}^{n \times n \times n}$, let $\text{OPT} = \min_{\rank - k} A_k$. For any $r \geq k$, if rank-$r$ tensor $B \in \mathbb{R}^{n \times n \times n}$ is an $f$-approximation to $A$, i.e.,

$$\|B - A\|_1 \leq f \cdot \text{OPT},$$

and $U, V, W \in \mathbb{R}^{n \times k}$ is a $g$-approximation to $B$, i.e.,

$$\|U \otimes V \otimes W - B\|_1 \leq g \cdot \min_{\rank - k} B_k \|B_k - B\|_1,$$

then,

$$\|U \otimes V \otimes W - A\|_1 \leq g f \cdot \text{OPT}.$$

**Proof.** We define $\tilde{U}, \tilde{V}, \tilde{W} \in \mathbb{R}^{n \times k}$ to be three matrices, such that

$$\|\tilde{U} \otimes \tilde{V} \otimes \tilde{W} - B\|_1 \leq g \min_{\rank - k} B_k \|B_k - B\|_1,$$

and also define,

$$\hat{U}, \hat{V}, \hat{W} = \arg \min_{U, V, W \in \mathbb{R}^{n \times k}} \|U \otimes V \otimes W - B\|_1 \text{ and } U^*, V^*, W^* = \arg \min_{U, V, W \in \mathbb{R}^{n \times k}} \|U \otimes V \otimes W - A\|_1.$$

82
It is obvious that,
\[ \|\tilde{U} \otimes \tilde{V} \otimes \tilde{W} - B\|_1 \leq \|U^* \otimes V^* \otimes W^* - B\|_1. \] (34)

Then,
\[
\begin{align*}
\|\tilde{U} \otimes \tilde{V} \otimes \tilde{W} - A\|_1 & \leq \|\tilde{U} \otimes \tilde{V} \otimes \tilde{W} - B\|_1 + \|B - A\|_1 & \text{by the triangle inequality} \\
& \leq g\|\tilde{U} \otimes \tilde{V} \otimes \tilde{W} - B\|_1 + \|B - A\|_1 & \text{by definition} \\
& \leq g\|U^* \otimes V^* \otimes W^* - B\|_1 + \|B - A\|_1 & \text{by Equation (34)} \\
& \leq g\|U^* \otimes V^* \otimes W^* - A\|_1 + g\|B - A\|_1 + \|B - A\|_1 & \text{by the triangle inequality} \\
= g\text{OPT} + (g + 1)\|B - A\|_1 & \text{by definition of OPT} \\
\leq g\text{OPT} + (g + 1)f \cdot \text{OPT} & \text{since } B \text{ is an } f \text{-approximation to } A \\
\leq gf \text{OPT}. & \text{ } \square
\end{align*}
\]

This completes the proof.

Using the above fact, we are able to optimize our approximation ratio.

D.2 Existence results

Definition D.3 ($\ell_1$ multiple regression cost preserving sketch - Definition D.5 in [SWZ17]). Given matrices $U \in \mathbb{R}^{n \times r}, A \in \mathbb{R}^{n \times d}$, let $S \in \mathbb{R}^{m \times n}$. If $\forall \beta \geq 1, \tilde{V} \in \mathbb{R}^{r \times d}$ which satisfy
\[
\|SU\tilde{V} - SA\|_1 \leq \beta \cdot \min_{V \in \mathbb{R}^{r \times d}} \|SUV - SA\|_1,
\]
it holds that
\[
\|U\tilde{V} - A\|_1 \leq \beta \cdot c \cdot \min_{V \in \mathbb{R}^{r \times d}} \|UV - A\|_1,
\]
then $S$ provides a $c$-$\ell_1$-multiple-regression-cost-preserving-sketch for $(U, A)$.

Theorem D.4. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exist three matrices $S_1 \in \mathbb{R}^{n \times s_1}, S_2 \in \mathbb{R}^{n \times s_2}, S_3 \in \mathbb{R}^{n \times s_3}$ such that
\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^k (A_1 S_1 X_1)_i \otimes (A_2 S_2 X_2)_i \otimes (A_3 S_3 X_3)_i - A \right\|_1 \leq \alpha \min_{\text{rank}-k \ A_k \in \mathbb{R}^{n \times n \times n}} \|A_k - A\|_1,
\]
holds with probability 99/100.

(I). Using a dense Cauchy transform,
$s_1 = s_2 = s_3 = \tilde{O}(k), \alpha = \tilde{O}(k^{1.5}) \log^3 n$.

(II). Using a sparse Cauchy transform,
$s_1 = s_2 = s_3 = \tilde{O}(k^5), \alpha = \tilde{O}(k^{13.5}) \log^3 n$.

(III). Guessing Lewis weights,
$s_1 = s_2 = s_3 = \tilde{O}(k), \alpha = \tilde{O}(k^{1.5})$. 

83
Proof. We use OPT to denote
\[
\text{OPT} := \min_{\text{rank}-k} \| A_k - A \|_1.
\]

Given a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), we define three matrices \( A_1 \in \mathbb{R}^{n_1 \times n_2 n_3}, A_2 \in \mathbb{R}^{n_2 \times n_3 n_1}, A_3 \in \mathbb{R}^{n_3 \times n_1 n_2} \) such that, for any \( i \in [n_1], j \in [n_2], l \in [n_3] \),
\[
A_{i,j,l} = (A_1)_{i,(j-1)n_3 + l} = (A_2)_{j,(l-1)n_1 + i} = (A_3)_{i,(l-1)n_2 + j}.
\]

We fix \( V^* \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and use \( V_1^*, V_2^*, \cdots, V_k^* \) to denote the columns of \( V^* \) and \( W_1^*, W_2^*, \cdots, W_k^* \) to denote the columns of \( W^* \).

We consider the following optimization problem,
\[
\min_{U_1, \cdots, U_k \in \mathbb{R}^n} \left\| \sum_{i=1}^k U_i \odot V_i^* \odot W_i^* - A \right\|_1,
\]
which is equivalent to
\[
\min_{U_1, \cdots, U_k \in \mathbb{R}^n} \left\| \begin{bmatrix} U_1 & U_2 & \cdots & U_k \end{bmatrix} \begin{bmatrix} V_1^* \odot W_1^* \\ V_2^* \odot W_2^* \\ \vdots \\ V_k^* \odot W_k^* \end{bmatrix} - A \right\|_1.
\]

We use matrix \( Z_1 \) to denote \( V^\top \odot W^\top \in \mathbb{R}^{k \times n^2} \) and matrix \( U \) to denote \( [U_1 \ U_2 \ \cdots \ U_k] \). Then we can obtain the following equivalent objective function,
\[
\min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 - A_1 \|_1.
\]

Choose an \( \ell_1 \) multiple regression cost preserving sketch \( S_1 \in \mathbb{R}^{n \times s_1} \) for \( (Z_1^\top, A_1^\top) \). We can obtain the optimization problem,
\[
\min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 S_1 - A_1 S_1 \|_1 = \min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^n \| U^i Z_1 S_1 - (A_1 S_1)^i \|_1,
\]
where \( U^i \) denotes the \( i \)-th row of matrix \( U \in \mathbb{R}^{n \times k} \) and \( (A_1 S_1)^i \) denotes the \( i \)-th row of matrix \( A_1 S_1 \). Instead of solving it under the \( \ell_1 \)-norm, we consider the \( \ell_2 \)-norm relaxation,
\[
\min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 S_1 - A_1 S_1 \|_F^2 = \min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^n \| U^i Z_1 S_1 - (A_1 S_1)^i \|_2^2.
\]

Let \( \hat{U} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above optimization problem. Then, \( \hat{U} = A_1 S_1 (Z_1 S_1)^\dagger \). We plug \( \hat{U} \) into the objective function under the \( \ell_1 \)-norm. According to Claim B.13, we have,
\[
\| \hat{U} Z_1 S_1 - A_1 S_1 \|_1 = \sum_{i=1}^n \| \hat{U}^i Z_1 S_1 - (A_1 S_1)^i \|_1 \leq \sqrt{s_1} \min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 S_1 - A_1 S_1 \|_1.
\]

Since \( S_1 \in \mathbb{R}^{n \times s_1} \) satisfies Definition D.3, we have
\[
\| \hat{U} Z_1 - A_1 \|_1 \leq \alpha \min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 - A_1 \|_1 = \alpha \text{OPT},
\]
84
where \( \alpha = \sqrt{s_1} \beta \) and \( \beta \) (see Definition D.3) is a parameter which depends on which kind of sketching matrix we actually choose. It implies

\[
\| \hat{U} \otimes V^* \otimes W^* - A \|_1 \leq \alpha \text{OPT}.
\]

As a second step, we fix \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and convert tensor \( A \) into matrix \( A_2 \). Let matrix \( Z_2 \) denote \( \hat{U}^\top \otimes W^* \). We consider the following objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 - A_2 \|_1,
\]

and the optimal cost of it is at most \( \alpha \text{OPT} \).

Choose an \( \ell_1 \) multiple regression cost preserving sketch \( S_2 \in \mathbb{R}^{n^2 \times s_2} \) for \( (Z_2^\top, A_2^\top) \), and sketch on the right of the objective function to obtain this new objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 S_2 - A_2 S_2 \|_1 = \min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^n \| V^i Z_2 S_2 - (A_2 S_2)^i \|_1,
\]

where \( V^i \) denotes the \( i \)-th row of matrix \( V \) and \( (A_2 S_2)^i \) denotes the \( i \)-th row of matrix \( A_2 S_2 \). Instead of solving this under the \( \ell_1 \)-norm, we consider the \( \ell_2 \)-norm relaxation,

\[
\min_{U \in \mathbb{R}^{n \times k}} \| V Z_2 S_2 - A_2 S_2 \|_2 = \min_{U \in \mathbb{R}^{n \times k}} \| V^i (Z_2 S_2) - (A_2 S_2)^i \|_2.
\]

Let \( \hat{V} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{V} = A_2 S_2 (Z_2 S_2)^\top \). By properties of the sketching matrix \( S_2 \in \mathbb{R}^{n^2 \times s_2} \), we have,

\[
\| \hat{V} Z_2 - A_2 \|_1 \leq \alpha \min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 - A_2 \|_1 \leq \alpha^2 \text{OPT},
\]

which implies

\[
\| \hat{U} \otimes \hat{V} \otimes W^* - A \|_1 \leq \alpha^2 \text{OPT}.
\]

As a third step, we fix the matrices \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( \hat{V} \in \mathbb{R}^{n \times k} \). We can convert tensor \( A \in \mathbb{R}^{n \times n \times n} \) into matrix \( A_3 \in \mathbb{R}^{n^2 \times n} \). Let matrix \( Z_3 \) denote \( \hat{U}^\top \otimes \hat{V}^\top \in \mathbb{R}^{k \times n^2} \). We consider the following objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \| W Z_3 - A_3 \|_1,
\]

and the optimal cost of it is at most \( \alpha^2 \text{OPT} \).

Choose an \( \ell_1 \) multiple regression cost preserving sketch \( S_3 \in \mathbb{R}^{n^2 \times s_3} \) for \( (Z_3^\top, A_3^\top) \) and sketch on the right of the objective function to obtain the new objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \| W Z_3 S_3 - A_3 S_3 \|_1.
\]

Let \( \hat{W} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{W} = A_3 S_3 (Z_3 S_3)^\top \). By properties of sketching matrix \( S_3 \in \mathbb{R}^{n^2 \times s_3} \), we have,

\[
\| \hat{W} Z_3 - A_3 \|_1 \leq \alpha \min_{W \in \mathbb{R}^{n \times k}} \| W Z_3 - A_3 \|_1 \leq \alpha^3 \text{OPT}.
\]
Thus, we obtain,
\[
\min_{X_1 \in \mathbb{R}^{s_1 \times k}, X_2 \in \mathbb{R}^{s_2 \times k}, X_3 \in \mathbb{R}^{s_3 \times k}} \left\| \sum_{i=1}^{k} (A_1 S_1 X_1)_i \otimes (A_2 S_2 X_2)_i \otimes (A_3 S_3 X_3)_i - A \right\|_1 \leq \alpha^3 \text{OPT}.
\]

Proof of (I) By Theorem C.1 in [SWZ17], we can use dense Cauchy transforms for \( S_1, S_2, S_3 \), and then \( s_1 = s_2 = s_3 = O(k \log k) \) and \( \alpha = O(\sqrt{k \log k}) \).

Proof of (II) By Theorem C.1 in [SWZ17], we can use sparse Cauchy transforms for \( S_1, S_2, S_3 \), and then \( s_1 = s_2 = s_3 = O(k^5 \log^5 k) \) and \( \alpha = O(k^{4.5} \log^{4.5} k \log n) \).

Proof of (III) By Theorem C.1 in [SWZ17], we can sample by Lewis weights. Then \( S_1, S_2, S_3 \in \mathbb{R}^{n^2 \times n^2} \) are diagonal matrices, and each of them has \( O(k \log k) \) nonzero rows. This gives \( \alpha = O(\sqrt{k \log k}) \).

D.3 Polynomial in \( k \) size reduction

Definition D.5 (Definition D.1 in [SWZ17]). Given a matrix \( M \in \mathbb{R}^{n \times d} \), if matrix \( S \in \mathbb{R}^{m \times n} \) satisfies
\[
\|SM\|_1 \leq \beta \|M\|_1,
\]
then \( S \) has at most \( \beta \) dilation on \( M \).

Definition D.6 (Definition D.2 in [SWZ17]). Given a matrix \( U \in \mathbb{R}^{n \times k} \), if matrix \( S \in \mathbb{R}^{m \times n} \) satisfies
\[
\forall x \in \mathbb{R}^k, \|SUx\|_1 \geq \frac{1}{\beta} \|Ux\|_1,
\]
then \( S \) has at most \( \beta \) contraction on \( U \).

Theorem D.7. Given a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and three matrices \( V_1 \in \mathbb{R}^{n_1 \times b_1}, V_2 \in \mathbb{R}^{n_2 \times b_2}, V_3 \in \mathbb{R}^{n_3 \times b_3} \), let \( X_1^* \in \mathbb{R}^{b_1 \times k}, X_2^* \in \mathbb{R}^{b_2 \times k}, X_3^* \in \mathbb{R}^{b_3 \times k} \) satisfies
\[
X_1^*, X_2^*, X_3^* = \arg \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|V_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - A\|_1.
\]
Let \( S \in \mathbb{R}^{m \times n} \) have at most \( \beta_1 \geq 1 \) dilation on \( V_1 X_1^* \cdot ((V_2 X_2^*)^\top \otimes (V_3 X_3^*)^\top) - A_1 \) and \( S \) have at most \( \beta_2 \geq 1 \) contraction on \( V_1 \). If \( \tilde{X}_1 \in \mathbb{R}^{b_1 \times k}, \tilde{X}_2 \in \mathbb{R}^{b_2 \times k}, \tilde{X}_3 \in \mathbb{R}^{b_3 \times k} \) satisfies
\[
\|SV_1 \tilde{X}_1 \otimes V_2 \tilde{X}_2 \otimes V_3 \tilde{X}_3 - SA\|_1 \leq \beta \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|SV_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - SA\|_1,
\]
where \( \beta \geq 1 \), then
\[
\|V_1 \tilde{X}_1 \otimes V_2 \tilde{X}_2 \otimes V_3 \tilde{X}_3 - A\|_1 \leq \beta_1 \beta_2 \beta \min_{X_1, X_2, X_3} \|V_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - A\|_1.
\]

The proof idea is similar to [SWZ17].

Proof. Let \( A, V_1, V_2, V_3, S, X_1^*, X_2^*, X_3^*, \beta_1, \beta_2 \) be the same as stated in the theorem. Let \( \tilde{X}_1 \in \mathbb{R}^{b_1 \times k}, \tilde{X}_2 \in \mathbb{R}^{b_2 \times k}, \tilde{X}_3 \in \mathbb{R}^{b_3 \times k} \) satisfy
\[
\|SV_1 \tilde{X}_1 \otimes V_2 \tilde{X}_2 \otimes V_3 \tilde{X}_3 - SA\|_1 \leq \beta \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|SV_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - SA\|_1.
\]
We have,

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_1 \\
\geq \|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SV_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3\|_1 - \|SV_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - SA\|_1 \\
\geq \frac{1}{\beta_2} \|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3\|_1 - \beta_1 \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
\geq \frac{1}{\beta_2} \|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - A\|_1 - \frac{1}{\beta_2} \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
- \beta_1 \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
= \frac{1}{\beta_2} \|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - A\|_1 - (\frac{1}{\beta_2} + \beta_1) \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1. 
\]

(35)

The first and the third inequality follow by the triangle inequalities. The second inequality follows using that

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SV_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3\|_1 \\
= \left\|SV_1 \left(\hat{X}_1 - X^*_1\right) \cdot \left((V_2 \hat{X}_2 - X^*_2)\right)^T \otimes \left((V_3 \hat{X}_3 - X^*_3)\right)^T\right\|_1 \\
\geq \frac{1}{\beta_2} \left\|V_1 \left(\hat{X}_1 - X^*_1\right) \cdot \left((V_2 \hat{X}_2 - X^*_2)\right)^T \otimes \left((V_3 \hat{X}_3 - X^*_3)\right)^T\right\|_1 \\
\geq \frac{1}{\beta_2} \|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3\|_1,
\]

and

\[
\|SV_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - SA\|_1 \\
= \|S(V_1 X^*_1 \cdot ((V_2 X^*_2)^T \otimes (V_3 X^*_3)^T) - A)\|_1 \\
\leq \|V_1 X^*_1 \cdot ((V_2 X^*_2)^T \otimes (V_3 X^*_3)^T) - A\|_1 \\
= \beta_1 \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1. 
\]

(36)

Then, we have

\[
\|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - A\|_1 \\
\leq \beta_2 \|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_1 + \left(1 + \beta_1 \beta_2\right) \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
\leq \beta_2 \beta \|SV_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - SA\|_1 + \left(1 + \beta_1 \beta_2\right) \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
\leq \beta_1 \beta_2 \beta \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 + \left(1 + \beta_1 \beta_2\right) \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1 \\
\leq \beta(1 + 2 \beta_1 \beta_2) \|V_1 X^*_1 \otimes V_2 X^*_2 \otimes V_3 X^*_3 - A\|_1.
\]

The first inequality follows by Equation (35). The second inequality follows by

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_1 \leq \beta \min_{X_1, X_2, X_3} \|SV_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - SA\|_1.
\]

The third inequality follows by Equation (36). The final inequality follows using that \(\beta \geq 1\). \qed

**Lemma D.8.** Let \(\min(b_1, b_2, b_3) \geq k\). Given three matrices \(V_1 \in \mathbb{R}^{n \times b_1}\), \(V_2 \in \mathbb{R}^{n \times b_2}\), and \(V_3 \in \mathbb{R}^{n \times b_3}\), there exists an algorithm that takes \(O(mnz(A)) + n \text{poly}(b_1, b_2, b_3)\) time and outputs a tensor.
Algorithm 21 Reducing the Size of the Objective Function to $\text{poly}(k)$

1: procedure L1POLYKSIZEREDUCTION($A, V_1, V_2, V_3, n, b_1, b_2, b_3, k$) \Comment*{Lemma D.8}
2:   for $i = 1 \rightarrow 3$ do
3:       $c_i \leftarrow \widehat{O}(b_i)$.
4:       Choose sampling and rescaling matrices $T_i \in \mathbb{R}^{c_i \times n}$ according to the Lewis weights of $V_i$.
5:       $\hat{V}_i \leftarrow T_iV_i \in \mathbb{R}^{c_i \times b_i}$.
6:   end for
7:   $C \leftarrow A(T_1, T_2, T_3) \in \mathbb{R}^{c_1 \times c_2 \times c_3}$.
8: return $\hat{V}_1, \hat{V}_2, \hat{V}_3$ and $C$.
9: end procedure

$C \in \mathbb{R}^{c_1 \times c_2 \times c_3}$ and three matrices $\hat{V}_1 \in \mathbb{R}^{c_1 \times b_1}$, $\hat{V}_2 \in \mathbb{R}^{c_2 \times b_2}$ and $\hat{V}_3 \in \mathbb{R}^{c_3 \times b_3}$ with $c_1 = c_2 = c_3 = \text{poly}(b_1, b_2, b_3)$, such that with probability 0.99, for any $\alpha \geq 1$, if $X'_1, X'_2, X'_3$ satisfy that,

$$\left\| \sum_{i=1}^{k} (\hat{V}_1 X'_1)_i \otimes (\hat{V}_2 X'_2)_i \otimes (\hat{V}_3 X'_3)_i - C \right\|_1 \leq \alpha \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - C \right\|_1,$$

then,

$$\left\| \sum_{i=1}^{k} (V_1 X'_1)_i \otimes (V_2 X'_2)_i \otimes (V_3 X'_3)_i - A \right\|_1 \leq \alpha \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_1.$$

Proof. For simplicity, we define $\text{OPT}$ to be

$$\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_1.$$

Let $T_1 \in \mathbb{R}^{c_1 \times n}$ sample according to the Lewis weights of $V_1 \in \mathbb{R}^{n \times b_1}$, where $c_1 = \widehat{O}(b_1)$. Let $T_2 \in \mathbb{R}^{c_2 \times n}$ sample according to the Lewis weights of $V_2 \in \mathbb{R}^{n \times b_2}$, where $c_2 = \widehat{O}(b_2)$. Let $T_3 \in \mathbb{R}^{c_3 \times n}$ sample according to the Lewis weights of $V_3 \in \mathbb{R}^{n \times b_3}$, where $c_3 = \widehat{O}(b_3)$.

For any $\alpha \geq 1$, let $X'_1 \in \mathbb{R}^{b_1 \times k}$, $X'_2 \in \mathbb{R}^{b_2 \times k}$, $X'_3 \in \mathbb{R}^{b_3 \times k}$ satisfy

$$\|T_1 V_1 X'_1 \otimes T_2 V_2 X'_2 \otimes T_3 V_3 X'_3 - A(T_1, T_2, T_3)\|_1 \leq \alpha \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|T_1 V_1 X_1 \otimes T_2 V_2 X_2 \otimes T_3 V_3 X_3 - A(T_1, T_2, T_3)\|_1.$$

First, we regard $T_1$ as the sketching matrix for the remainder. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have

$$\|V_1 X'_1 \otimes T_2 V_2 X'_2 \otimes T_3 V_3 X'_3 - A(I, T_2, T_3)\|_1 \leq \alpha \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|V_1 X_1 \otimes T_2 V_2 X_2 \otimes T_3 V_3 X_3 - A(I, T_2, T_3)\|_1.$$

Second, we regard $T_2$ as a sketching matrix for $V_1 X_1 \otimes V_2 X_2 \otimes T_3 V_3 X_3 - A(I, I, T_3)$. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have

$$\|V_1 X'_1 \otimes V_2 X'_2 \otimes T_3 V_3 X'_3 - A(I, I, T_3)\|_1 \leq \alpha \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|V_1 X_1 \otimes V_2 X_2 \otimes T_3 V_3 X_3 - A(I, I, T_3)\|_1.$$
Let $\beta$ be a sampling/rescaling matrix according to the Lewis weights of $U$.

### Proof

We first look at $X^* = \arg\min_{X\in\mathbb{R}^{n \times s}} \|U \otimes V \otimes X - A\|_1$. Then according to Lemma D.11 in [SWZ17], $T$ has at most constant dilation (Definition D.5) on $U \cdot (V^T \circ (X^*)^T) - A_1$, and has at most constant contraction (Definition D.6) on $U$. We first look at

$$
\|TU \otimes V \otimes X^* - TA_1\|_1 \\
= \|TU \cdot (V^T \circ (X^*)^T) - TA_1\|_1 \\
\geq \|TU \cdot ((V^T \circ (X^*)^T) - (V^T \circ (X^*)^T))\|_1 - \|TU \cdot (V^T \circ (X^*)^T) - TA_1\|_1 \\
\geq \frac{1}{\beta_2} \|U \cdot ((V^T \circ (X^*)^T) - A_1\|_1 - \frac{1}{\beta_2} \|U \cdot (V^T \circ (X^*)^T) - A_1\|_1,
$$

where $\beta_1 \geq 1, \beta_2 \geq 1$ are two constants. Then we have:

$$
\|U \otimes V \otimes X^* - A\|_1 \\
\leq \beta_2 \|TU \cdot (V^T \circ (X^*)^T) - TA_1\|_1 + (1 + \beta_1\beta_2) \|U \cdot (V^T \circ (X^*)^T) - A_1\|_1 \\
\leq \alpha \beta_2 \|TU \cdot (V^T \circ (X^*)^T) - TA_1\|_1 + (1 + \beta_1\beta_2) \|U \cdot (V^T \circ (X^*)^T) - A_1\|_1 \\
\leq \alpha \beta_1 \beta_2 \|U \cdot (V^T \circ (X^*)^T) - A_1\|_1 + (1 + \beta_1\beta_2) \|U \cdot (V^T \circ (X^*)^T) - A_1\|_1 \\
\leq \alpha \|U \otimes V \otimes X^* - A\|_1.
$$

### Corollary D.10

Given tensor $A \in \mathbb{R}^{n \times n \times n}$, and two matrices $U \in \mathbb{R}^{n \times s}, V \in \mathbb{R}^{n \times s}$ with $\text{rank}(U) = r_1, \text{rank}(V) = r_2$, let $T_1 \in \mathbb{R}^{t_1 \times n}$ be a sampling/rescaling matrix according to the Lewis weights of $U$, and let $T_2 \in \mathbb{R}^{t_2 \times n}$ be a sampling/rescaling matrix according to the Lewis weights of $V$ with $t_1 = \tilde{O}(r_1), t_2 = \tilde{O}(r_2)$. Then with probability at least 0.99, for all $X' \in \mathbb{R}^{n \times s}, \alpha \geq 1$ which satisfy

$$
\|T_1U \otimes T_2V \otimes X' - A(T_1, T_2, I)\|_1 \leq \alpha \cdot \min_{X \in \mathbb{R}^{n \times s}} \|T_1U \otimes T_2V \otimes X - A(T_1, T_2, I)\|_1,
$$

it holds that

$$
\|U \otimes V \otimes X' - A\|_1 \leq \alpha \cdot \min_{X \in \mathbb{R}^{n \times s}} \|U \otimes V \otimes X - A\|_1.
$$
Proof. We apply Lemma D.9 twice: if
\[ \|T_1 U \otimes T_2 V \otimes X' - A(T_1, T_2, I)\|_1 \leq \alpha \cdot \min_{X \in \mathbb{R}^{n \times s}} \|T_1 U \otimes T_2 V \otimes X - A(T_1, T_2, I)\|_1, \]
then
\[ \|U \otimes T_2 V \otimes X' - A(I, T_2, I)\|_1 \leq \alpha \cdot \min_{X \in \mathbb{R}^{n \times s}} \|U \otimes T_2 V \otimes X - A(I, T_2, I)\|_1. \]
Then, we have
\[ \|U \otimes V \otimes X' - A\|_1 \leq \alpha \cdot \min_{X \in \mathbb{R}^{n \times s}} \|U \otimes V \otimes X - A\|_1. \]

\[
\square
\]

D.4 Solving small problems

Theorem D.11. Let \( \max_i \{t_i, d_i\} \leq n \). Given a \( t_1 \times t_2 \times t_3 \) tensor \( A \) and three matrices: a \( t_1 \times d_1 \) matrix \( T_1 \), a \( t_2 \times d_2 \) matrix \( T_2 \), and a \( t_3 \times d_3 \) matrix \( T_3 \), if for \( \delta > 0 \) there exists a solution to
\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i - A \right\|_1 := \text{OPT},
\]
such that each entry of \( X_i \) can be expressed using \( O(n^\delta) \) bits, then there exists an algorithm that takes \( n^{O(\delta)} \cdot 2^{O(d_1 k + d_2 k + d_3 k)} \) time and outputs three matrices: \( \hat{X}_1 \), \( \hat{X}_2 \), and \( \hat{X}_3 \) such that \( \|(T_1 \hat{X}_1) \otimes (T_2 \hat{X}_2) \otimes (T_3 \hat{X}_3) - A\|_1 = \text{OPT} \).

Proof. For each \( i \in [3] \), we can create \( t_i \times d_i \) variables to represent matrix \( X_i \). Let \( x \) denote the list of these variables. Let \( B \) denote tensor \( \sum_{i=1}^{k} (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i \). Then we can write the following objective function,
\[
\min_{x} \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} |B_{i,j,l}(x) - A_{i,j,l}|.
\]
To remove the \(| \cdot |\), we create \( t_1 t_2 t_3 \) extra variables \( \sigma_{i,j,l} \). Then we obtain the objective function:
\[
\min_{x, \sigma} \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} \sigma_{i,j,l} (B_{i,j,l}(x) - A_{i,j,l})
\]
s.t. \( \sigma_{i,j,l}^2 = 1 \),
\[
\sigma_{i,j,l} (B_{i,j,l}(x) - A_{i,j,l}) \geq 0,
\]
\[
\|x\|_2^2 + \|\sigma\|_2^2 \leq 2^{O(n^\delta)}
\]
where the last constraint is unharful, because there exists a solution that can be written using \( O(n^\delta) \) bits. Note that the number of inequality constraints in the above system is \( O(t_1 t_2 t_3) \), the degree is \( O(1) \), and the number of variables is \( v = (t_1 t_2 t_3 + d_1 k + d_2 k + d_3 k) \). Thus by Theorem B.11, we know that the minimum nonzero cost is at least
\[
(2^{O(n^\delta)}) - 2^{O(v)}.
\]

90
It is immediate that the upper bound on cost is at most $2^{O(n^\delta)}$, and thus the number of binary search steps is at most $\log(2^{O(n^\delta)}2^{O(v)})$. In each step of the binary search, we need to choose a cost $C$ between the lower bound and the upper bound, and write down the polynomial system,

$$\sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \sum_{l=1}^{t_3} \sigma_{i,j,l}(B_{i,j,l}(x) - A_{i,j,l}) \leq C,$$

$$\sigma_{i,j,l}^2 = 1,$$

$$\sigma_{i,j,l}(B_{i,j,l}(x) - A_{i,j,l}) \geq 0,$$

$$\|x\|_2^2 + \|\sigma\|_2^2 \leq 2^{O(n^\delta)}.$$

Using Theorem B.10, we can determine if there exists a solution to the above polynomial system. Since the number of variables is $v$, and the degree is $O(1)$, the number of inequality constraints is $t_1t_2t_3$. Thus, the running time is

$$\text{poly}(\text{bitsize}) \cdot (\# \text{ constraints} \cdot \text{degree}) \cdot \# \text{variables} = n^{O(\delta)}2^{O(v)}.$$

D.5 Bicriteria algorithms

We present several bicriteria algorithms with different tradeoffs. We first present an algorithm that runs in nearly linear time and outputs a solution with rank $\tilde{O}(k^3)$ in Theorem D.12. Then we show an algorithm that runs in $\text{nnz}(A)$ time but outputs a solution with rank $\text{poly}(k)$ in Theorem D.13. Then we explain an idea which is able to decrease the cubic rank to quadratic rank, and thus we can obtain Theorem D.14 and Theorem D.15.

D.5.1 Input sparsity time

Algorithm 22 $\ell_1$-Low Rank Approximation, Bicriteria Algorithm, rank-$\tilde{O}(k^3)$, Nearly Input Sparsity Time

1: procedure $\text{L1BICRITERIAALGORITHM}(A, n, k)$ \Comment{Theorem D.12}
2: \quad $s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k)$.\Comment{Part (I) of Theorem D.2}
3: \quad For each $i \in [3]$, choose $S_i \in \mathbb{R}^{n^2 \times s_i}$ to be a dense Cauchy transform.\Comment{Algorithm 21}
4: \quad Compute $A_1 \cdot S_1, A_2 \cdot S_2, A_3 \cdot S_3$.\Comment{Part (I) of Theorem D.2}
5: \quad $Y_1, Y_2, Y_3, C \leftarrow \text{L1POLYKSIZEREDUCTION}(A, A_1 S_1, A_2 S_2, A_3 S_3, n, s_1, s_2, s_3, k)$\Comment{Algorithm 21}
6: \quad Form objective function

$$\min_{X \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} X_{i,j,l}(Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_1.$$  

7: \quad Run $\ell_1$-regression solver to find $X$.\Comment{Theorem D.12}
8: \quad return $A_1 S_1, A_2 S_2, A_3 S_3$ and $X$.\Comment{Theorem D.12}
9: end procedure
There exists an algorithm which takes \( \text{nnz}(A) \cdot \tilde{O}(k) + O(n) \text{poly}(k) + \text{poly}(k) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times r} \) such that

\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_1 \leq \tilde{O}(k^{3/2}) \log^3 n \min_{\text{rank} - k} \left\| A_k - A \right\|_1
\]

holds with probability 9/10.

**Proof.** We first choose three dense Cauchy transforms \( S_i \in \mathbb{R}^{n^2 \times s_i} \). According to Section B.7, for each \( i \in [3] \), \( A_i S_i \) can be computed in \( \text{nnz}(A) \cdot \tilde{O}(k) \) time. Then we apply Lemma D.8 (Algorithm 21). We obtain three matrices \( Y_1, Y_2, Y_3 \) and a tensor \( C \). Note that for each \( i \in [3] \), \( Y_i \) can be computed in \( n \text{poly}(k) \) time. Because \( C = A(T_1, T_2, T_3) \) and \( T_1, T_2, T_3 \in \mathbb{R}^{n \times \tilde{O}(k)} \) are three sampling and rescaling matrices, \( C \) can be computed in \( \text{nnz}(A) + \tilde{O}(k^3) \) time. At the end, we just need to run an \( \ell_1 \)-regression solver to find the solution to the problem,

\[
\min_{X \in \mathbb{R}^{n \times s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} X_{i,j,l} (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_1,
\]

where \((Y_1)_i\) denotes the \( i \)-th column of matrix \( Y_1 \). Since the size of the above problem is only \( \text{poly}(k) \), this can be solved in \( \text{poly}(k) \) time. \( \Box \)

**Algorithm 23** \( \ell_1 \)-Low Rank Approximation, Bicriteria Algorithm, rank-poly(\( k \)), Input Sparsity Time

1: \textbf{procedure} L1BICRITERIAlGORITHM\((A, n, k)\) \hspace{1cm} \( \triangleright \) Theorem D.13
2: \hspace{0.5cm} \( s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k^{5}) \).
3: \hspace{0.5cm} For each \( i \in [3] \), choose \( S_i \in \mathbb{R}^{n^2 \times s_i} \) to be a sparse Cauchy transform. \( \triangleright \) Part (II) of Theorem D.4
4: \hspace{0.5cm} Compute \( A_1 \cdot S_1, A_2 \cdot S_2, A_3 \cdot S_3 \).
5: \hspace{0.5cm} \( Y_1, Y_2, Y_3, C \leftarrow \text{L1PolyKSizeReduction}(A, A_1 S_1, A_2 S_2, A_3 S_3, n, s_1, s_2, s_3, k) \) \hspace{0.5cm} \( \triangleright \) Algorithm 21
6: \hspace{0.5cm} Form objective function

\[
\min_{X \in \mathbb{R}^{n \times s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} X_{i,j,l} (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_1.
\]

7: \hspace{0.5cm} Run \( \ell_1 \)-regression solver to find \( X \).
8: \hspace{0.5cm} \textbf{return} \( A_1 S_1, A_2 S_2, A_3 S_3 \) and \( X \).
9: \textbf{end procedure}

**Theorem D.13.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1, \epsilon \in (0, 1) \), let \( r = \tilde{O}(k^{15}) \). There exists an algorithm that takes \( \text{nnz}(A) + O(n) \text{poly}(k) + \text{poly}(k) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times r} \) such that

\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank} - k} \left\| A_k - A \right\|_1
\]

holds with probability 9/10.
There exists an algorithm which takes \( n_{\text{nnz}}(A) \) time. According to Section B.7, for each \( i \in [3] \), \( A_iS_i \) can be computed in \( O(n_{\text{nnz}}(A)) \) time. Then we apply Lemma D.8 (Algorithm 21), and can obtain three matrices \( Y_1, Y_2, Y_3 \) and a tensor \( C \). Note that for each \( i \in [3] \), \( Y_i \) can be computed in \( O(n) \, \text{poly}(k) \) time. Because \( C = A(T_1, T_2, T_3) \) and \( T_1, T_2, T_3 \in \mathbb{R}^{n \times \tilde{O}(k)} \) are three sampling and rescaling matrices, \( C \) can be computed in \( n_{\text{nnz}}(A) + \tilde{O}(k^3) \) time. At the end, we just need to run an \( \ell_1 \)-regression solver to find the solution to the problem,

\[
\min_{X \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} X_{i,j,l} (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_1 ,
\]

where \((Y_1)_i\) denotes the \( i \)-th column of matrix \( Y_1 \). Since the size of the above problem is only \( \text{poly}(k) \), it can be solved in \( \text{poly}(k) \) time.

\[ \square \]

D.5.2 Improving cubic rank to quadratic rank

**Algorithm 24** \( \ell_1 \)-Low Rank Approximation, Bicriteria Algorithm, rank-\( \tilde{O}(k^2) \), Nearly Input Sparsity Time

1: \textbf{procedure} L1BICRITERIAALGORITHM(\( A, n, k \)) \hfill \triangleright \text{Theorem D.14}
2: \hspace{1em} \( s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k) \).
3: \hspace{1em} For each \( i \in [3] \), choose \( S_i \in \mathbb{R}^{n_2 \times s_i} \) to be a dense Cauchy transform. \hfill \triangleright \text{Part (I) of Theorem D.2}
4: \hspace{1em} Compute \( A_1 \cdot S_1, A_2 \cdot S_2 \).
5: \hspace{1em} For each \( i \in [2] \), choose \( T_i \) to be a sampling and rescaling diagonal matrix according to the Lewis weights of \( A_iS_i \), with \( t_i = \tilde{O}(k) \) nonzero entries.
6: \hspace{1em} \( C \leftarrow A(T_1, T_2, I) \).
7: \hspace{1em} \( B^{l+(j-1)s_1} \leftarrow \text{vec}((T_1A_1S_1)_i \otimes (T_2A_2S_2)_j), \forall i \in [s_1], j \in [s_2] \).
8: \hspace{1em} Form objective function \( \min_{W} \|WB - C_3\|_1 \)
9: \hspace{1em} Run \( \ell_1 \)-regression solver to find \( \hat{W} \).
10: \hspace{1em} Construct \( \hat{U} \) by using \( A_1S_1 \) according to Equation (38).
11: \hspace{1em} Construct \( \hat{V} \) by using \( A_2S_2 \) according to Equation (39).
12: \hspace{1em} \textbf{return} \( \hat{U}, \hat{V}, \hat{W} \).
13: \textbf{end procedure}

**Theorem D.14.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1, \epsilon \in (0, 1) \), let \( r = \tilde{O}(k^2) \). There exists an algorithm which takes \( n_{\text{nnz}}(A) \cdot \tilde{O}(k) + O(n) \, \text{poly}(k) + \text{poly}(k) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times r} \) such that

\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_1 \leq \tilde{O}(k^{3/2}) \log^3 n \, \min_{\text{rank-}k} \|A_k - A\|_1
\]

holds with probability \( 9/10 \).

**Proof.** Let \( \text{OPT} = \min_{A_k \in \mathbb{R}^{n \times n \times n}} \|A_k - A\|_1 \). We first choose three dense Cauchy transforms \( S_i \in \mathbb{R}^{n_2 \times s_i} \), \( \forall i \in [3] \). According to Section B.7, for each \( i \in [3] \), \( A_iS_i \) can be computed in \( n_{\text{nnz}}(A) \cdot \tilde{O}(k) \) time. Then we choose \( T_i \) to be a sampling and rescaling diagonal matrix according to the Lewis weights of \( A_iS_i, \forall i \in [2] \).
According to Theorem D.4, we have
\[
\begin{align*}
\min_{X_1 \in \mathbb{R}^{s_1 \times k}, X_2 \in \mathbb{R}^{s_2 \times k}, X_3 \in \mathbb{R}^{s_3 \times k}} \left\| \sum_{l=1}^{k} (A_1 S_1 X_1)_l \otimes (A_2 S_2 X_2)_l \otimes (A_3 S_3 X_3)_l - A \right\|_1 
\leq \tilde{O}(k^{1.5}) \log^3 n \text{ OPT}
\end{align*}
\]

Now we fix an \(l\) and we have:
\[
\begin{align*}
(A_1 S_1 X_1)_l \otimes (A_2 S_2 X_2)_l \otimes (A_3 S_3 X_3)_l \\
= \left( \sum_{i=1}^{s_1} (A_1 S_1)_i (X_1)_i,l \right) \otimes \left( \sum_{j=1}^{s_2} (A_2 S_2)_j (X_2)_j,l \right) \otimes (A_3 S_3 X_3)_l \\
= \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} (A_1 S_1)_i \otimes (A_2 S_2)_j \otimes (A_3 S_3 X_3)_l (X_1)_i,l (X_2)_j,l
\end{align*}
\]

Thus, we have
\[
\begin{align*}
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} (A_1 S_1)_i \otimes (A_2 S_2)_j \otimes \left( \sum_{l=1}^{k} (A_3 S_3 X_3)_l (X_1)_i,l (X_2)_j,l \right) - A \right\|_1 
\leq \tilde{O}(k^{1.5}) \log^3 n \text{ OPT}.
\end{align*}
\] (37)

We create matrix \( \hat{U} \in \mathbb{R}^{n \times s_1 s_2} \) by copying matrix \( A_1 S_1 \) \( s_2 \) times, i.e.,
\[
\hat{U} = \begin{bmatrix} A_1 S_1 & A_1 S_1 & \cdots & A_1 S_1 \end{bmatrix}.
\] (38)

We create matrix \( \hat{V} \in \mathbb{R}^{n \times s_1 s_2} \) by copying the \( i \)-th column of \( A_2 S_2 \) a total of \( s_1 \) times into the columns \((i-1)s_1, \cdots, is_1\) of \( \hat{V} \), for each \( i \in [s_2] \), i.e.,
\[
\hat{V} = \begin{bmatrix} (A_2 S_2)_1 & \cdots & (A_2 S_2)_1 & (A_2 S_2)_2 & \cdots & (A_2 S_2)_2 & \cdots & (A_2 S_2)_s_2 & \cdots & (A_2 S_2)_{s_2} \end{bmatrix}.
\] (39)

According to Equation (37), we have:
\[
\min_{W \in \mathbb{R}^{n \times s_1 s_2}} \| \hat{U} \hat{V} \otimes W - A \|_1 \leq \tilde{O}(k^{1.5}) \log^3 n \cdot \text{ OPT}.
\]

Let
\[
\hat{W} = \arg \min_{W \in \mathbb{R}^{n \times s_1 s_2}} \| T_1 \hat{U} \otimes T_2 \hat{V} \otimes W - A(T_1, T_2, I) \|_1.
\]

Due to Corollary D.10, we have
\[
\| \hat{U} \hat{V} \hat{W} - A \|_1 \leq \tilde{O}(k^{1.5}) \log^3 n \cdot \text{ OPT}.
\]

Putting it all together, we have that \( \hat{U}, \hat{V}, \hat{W} \) gives a rank-\( \tilde{O}(k^2) \) bicriteria algorithm to the original problem.

\[\square\]

**Theorem D.15.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), \( \epsilon \in (0, 1) \), let \( r = \tilde{O}(k^{10}) \). There exists an algorithm which takes \( \text{nnz}(A) + O(n) \text{ poly}(k) + \text{poly}(k) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times r} \) such that
\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank}-k A_k} \| A_k - A \|_1
\]
holds with probability 9/10.
Algorithm 25 $\ell_1$-Low Rank Approximation, Bicriteria Algorithm, rank-poly$(k)$, Input Sparsity Time

1: procedure L1BICRITERIALGORITHM($A, n, k$) \Comment{Theorem D.15}
2: \hspace{1em} $s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \widetilde{O}(k^3)$. \Comment{Part (II) of Theorem D.2}
3: \hspace{1em} For each $i \in [3]$, choose $S_i \in \mathbb{R}^{n^2 \times s_i}$ to be a sparse Cauchy transform.
4: \hspace{1em} Compute $A_1 \cdot S_1$, $A_2 \cdot S_2$.
5: \hspace{1em} For each $i \in [2]$, choose $T_i$ to be a sampling and rescaling diagonal matrix according to the Lewis weights of $A_i S_i$, with $t_i = \widetilde{O}(k)$ nonzero entries.
6: \hspace{1em} $C \leftarrow A(T_1, T_2, I)$.
7: \hspace{1em} $B^{i+(i-1)s_1} \leftarrow \text{vec}((T_1 A_1 S_1)_i \otimes (T_2 A_2 S_2)_j), \forall i \in [s_1], j \in [s_2]$.
8: \hspace{1em} Form objective function $\min_W \|WB - C_3\|_1$.
9: \hspace{1em} Run $\ell_1$-regression solver to find $W$.
10: \hspace{1em} Construct $\hat{U}$ by using $A_1 S_1$ according to Equation (38).
11: \hspace{1em} Construct $\hat{V}$ by using $A_2 S_2$ according to Equation (39).
12: \hspace{1em} return $\hat{U}, \hat{V}, \hat{W}$.
13: end procedure

Proof. The proof is similar to the proof of Theorem D.14. The only difference is that instead of choosing dense Cauchy matrices $S_1, S_2$, we choose sparse Cauchy matrices. \hfill \Box

Notice that if we firstly apply a sparse Cauchy transform, we can reduce the rank of the matrix to poly$(k)$. Then we apply a dense Cauchy transform and can further reduce the dimension while only incurring another poly$(k)$ factor in the approximation ratio. By combining a sparse Cauchy transform and a dense Cauchy transform, we can improve the running time from $\text{nnz}(A) \cdot \widetilde{O}(k)$ to $\text{nnz}(A)$.

Corollary D.16. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, $\epsilon \in (0, 1)$, let $r = \widetilde{O}(k^2)$. There exists an algorithm which takes $\text{nnz}(A) + O(n) \text{poly}(k) + \text{poly}(k)$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times r}$ such that

$$
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank} - k A_k} \|A_k - A\|_1
$$

holds with probability $9/10$.

D.6 Algorithms

In this section, we show two different algorithms by using different kinds of sketches. One is shown in Theorem D.17 which gives a fast running time. Another one is shown in Theorem D.19 which gives the best approximation ratio.

D.6.1 Input sparsity time algorithm

Theorem D.17. Given a 3rd tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm that takes $\text{nnz}(A) \cdot \widetilde{O}(k) + O(n) \text{poly}(k) + 2\widetilde{O}(k^2)$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that,

$$
\|U \otimes V \otimes W - A\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank} - k A} \|A' - A\|_1.
$$

95
Algorithm 26 \ell_1\text{-Low Rank Approximation, Bicriteria Algorithm, rank-\(\tilde{O}(k^2)\), Input Sparsity Time}

1: \textbf{procedure} L1BIKRITRIAALGORITHM\((A, n, k)\) \hspace{1cm} \triangleright \text{Corollary D.16}
2: \hspace{1cm} \(s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k)\).
3: \hspace{1cm} For each \(i \in [3]\), choose \(S_i \in \mathbb{R}^{n^2 \times s_i}\) to be the composition of a sparse Cauchy transform and a dense Cauchy transform. \hspace{1cm} \triangleright \text{Part (I,II) of Theorem D.2}
4: \hspace{1cm} Compute \(A_1 \cdot S_1, A_2 \cdot S_2\).
5: \hspace{1cm} For each \(i \in [2]\), choose \(T_i\) to be a sampling and rescaling diagonal matrix according to the Lewis weights of \(A_i S_i\), with \(t_i = \tilde{O}(k)\) nonzero entries.
6: \hspace{1cm} \(C \leftarrow A(T_1, T_2, I)\).
7: \hspace{1cm} \(B^{i+ (j-1)s_1} \leftarrow \text{vec}(T_i A_1 S_1)_i \otimes (T_j A_2 S_2)_j, \forall i \in [s_1], j \in [s_2]\).
8: \hspace{1cm} Form objective function \(\min_B \|WB - C_3\|_1\).
9: \hspace{1cm} Run \(\ell_1\)-regression solver to find \(\tilde{W}\).
10: \hspace{1cm} Construct \(\tilde{U}\) by using \(A_1 S_1\) according to Equation (38).
11: \hspace{1cm} Construct \(\tilde{V}\) by using \(A_2 S_2\) according to Equation (39).
12: \hspace{1cm} \textbf{return} \(\tilde{U}, \tilde{V}, \tilde{W}\).
13: \textbf{end procedure}

Algorithm 27 \ell_1\text{-Low Rank Approximation, Input sparsity Time Algorithm}

1: \textbf{procedure} L1TENSORLOWRANKAPPROXINPUTSPARSITY\((A, n, k)\) \hspace{1cm} \triangleright \text{Theorem D.17}
2: \hspace{1cm} \(s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k^5)\).
3: \hspace{1cm} Choose \(S_i \in \mathbb{R}^{n^2 \times s_i}\) to be a dense Cauchy transform, \(\forall i \in [3]\). \hspace{1cm} \triangleright \text{Part (I) of Theorem D.4}
4: \hspace{1cm} Compute \(A_1 \cdot S_1, A_2 \cdot S_2\), and \(A_3 \cdot S_3\).
5: \hspace{1cm} \(Y_1, Y_2, Y_3, C \leftarrow L1POLYKSIZEREDUCTION\((A, A_1 S_1, A_2 S_2, A_3 S_3, n, s_1, s_2, s_3, k)\). \hspace{1cm} \triangleright \text{Algorithm 21}
6: \hspace{1cm} Create variables \(s_1 \times k + s_2 \times k + s_3 \times k\) variables for each entry of \(X_1, X_2, X_3\).
7: \hspace{1cm} Form objective function \(\|(Y_1 X_1) \otimes (Y_2 X_2) \otimes (Y_3 X_3) - C\|_F^2\).
8: \hspace{1cm} Run polynomial system verifier.
9: \hspace{1cm} \textbf{return} \(A_1 S_1 X_1, A_2 S_2 X_2, A_3 S_3 X_3\).
10: \textbf{end procedure}

holds with probability at least \(9/10\).

\textbf{Proof.} First, we apply part (II) of Theorem D.4. Then \(A_i S_i\) can be computed in \(O(\text{nnz}(A))\) time. Second, we use Lemma D.8 to reduce the size of the objective function from \(O(n^3)\) to poly\((k)\) in \(n\ \text{poly}(k)\) time by only losing a constant factor in approximation ratio. Third, we use Claim B.15 to relax the objective function from entry-wise \(\ell_1\)-norm to Frobenius norm, and this step causes us to lose some other poly\((k)\) factors in approximation ratio. As a last step, we use Theorem C.45 to solve the Frobenius norm objective function. \(\square\)

Notice again that if we first apply a sparse Cauchy transform, we can reduce the rank of the matrix to poly\((k)\). Then as before we can apply a dense Cauchy transform to further reduce the dimension while only incurring another poly\((k)\) factor in the approximation ratio. By combining a sparse Cauchy transform and a dense Cauchy transform, we can improve the running time from nnz\((A) \cdot \tilde{O}(k)\) to nnz\((A)\), while losing some additional poly\((k)\) factors in approximation ratio.
Theorem D.20. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm that takes $\text{nnz}(A) + O(n) \text{poly}(k) + 2^{O(k^2)}$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that,

$$\|U \otimes V \otimes W - A\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank} - k \ A'} \|A' - A\|_1.$$ 

holds with probability at least 9/10.

D.6.2 $\tilde{O}(k^{3/2})$-approximation algorithm

Algorithm 28 $\ell_1$-Low Rank Approximation Algorithm, $\tilde{O}(k^{3/2})$-approximation

```plaintext
1: procedure L1TensorLowRankApproxK(A, n, k) \triangleright\text{Theorem D.19}
2: \quad s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k).
3: \quad \text{Guess diagonal matrices } S_i \in \mathbb{R}^{n^2 \times s_i} \text{ with } s_i \text{ nonzero entries, } \forall i \in [3]. \triangleright\text{Part (III) of Theorem D.4}
4: \quad Y_1, Y_2, Y_3, C \leftarrow L1PolyKSizeReduction(A, A_1S_1, A_2S_2, A_3S_3, n, s_1, s_2, s_3, k). \triangleright\text{Algorithm 21}
5: \quad \text{Create } s_1 \times k + s_2 \times k + s_3 \times k \text{ variables for each entry of } X_1, X_2, X_3.
6: \quad \text{Form objective function } \|\langle Y_1X_1 \rangle \otimes \langle Y_2X_2 \rangle \otimes \langle Y_3X_3 \rangle - C\|_1.
7: \quad \text{Run polynomial system verifier.}
8: \quad \text{return } U, V, W.
9: end procedure
```

Theorem D.19. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm that takes $n^{\tilde{O}(k)}2^{\tilde{O}(k^3)}$ time and output three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that,

$$\|U \otimes V \otimes W - A\|_1 \leq \tilde{O}(k^{3/2}) \min_{\text{rank} - k \ A'} \|A' - A\|_1.$$ 

holds with probability at least 9/10.

Proof. First, we apply part (III) of Theorem D.4. Then, guessing $S_i$ requires $n^{\tilde{O}(k)}$ time. Second, we use Lemma D.8 to reduce the size of the objective from $O(n^3)$ to $\text{poly}(k)$ in polynomial time while only losing a constant factor in approximation ratio. Third, we use Theorem D.11 to solve the entry-wise $\ell_1$-norm objective function directly. \hfill \square

D.7 CURT decomposition

Theorem D.20. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $k \geq 1$, let $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$ denote a rank-$k$, $\alpha$-approximation to $A$. Then there exists an algorithm which takes $O(\text{nnz}(A) + O(n^2)\text{poly}(k)$ time and outputs three matrices: $C \in \mathbb{R}^{n \times c}$ with columns from $A$, $R \in \mathbb{R}^{n \times r}$ with rows from $A$, $T \in \mathbb{R}^{n \times t}$ with tubes from $A$, and a tensor $U \in \mathbb{R}^{c \times r \times t}$ with rank($U$) = $k$ such that $c = r = t = O(k \log k)$, and

$$\left\| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_1 \leq \tilde{O}(k^{1.5})\alpha \min_{\text{rank} - k \ A'} \|A' - A\|_1$$

holds with probability 9/10.
Theorem D.20

\[ \text{OPT} := \min_{\text{rank} = k} A, \| A' - A \|_1. \]

We already have three matrices \( U_B \in \mathbb{R}^{n \times k}, V_B \in \mathbb{R}^{n \times k} \) and \( W_B \in \mathbb{R}^{n \times k} \) and these three matrices provide a rank-\( k \), \( \alpha \) approximation to \( A \), i.e.,

\[
\left\| \sum_{i=1}^k (U_B)_i \otimes (V_B)_i \otimes (W_B)_i - A \right\|_1 \leq \alpha \text{ OPT} \tag{40}
\]

Let \( B_1 = V_B^T \otimes W_B^T \in \mathbb{R}^{k \times n^2} \) denote the matrix where the \( i \)-th row is the vectorization of \( (V_B)_i \otimes (W_B)_i \). By Section B.3, we can compute \( D_1 \in \mathbb{R}^{n^2 \times n^2} \) which is a sampling and rescaling matrix corresponding to the Lewis weights of \( B_1^T \) in \( O(n^2 \text{poly}(k)) \) time, and there are \( d_1 = O(k \log k) \) nonzero entries on the diagonal of \( D_1 \). Let \( A_i \in \mathbb{R}^{n \times n} \) denote the matrix obtained by flattening \( A \) along the \( i \)-th direction, for each \( i \in [3] \).

Define \( U^* \in \mathbb{R}^{n \times k} \) to be the optimal solution to \( \min_{U \in \mathbb{R}^{n \times k}} \| UB_1 - A_1 \|_1, \hat{U} = A_1 D_1 (B_1 D_1)^\dagger \in \mathbb{R}^{n \times k}, \) \( V_0 \in \mathbb{R}^{n \times k} \) to be the optimal solution to \( \min_{V \in \mathbb{R}^{n \times k}} \| V \cdot (\hat{U^T} \otimes W_B^T) - A_2 \|_1 \), and \( U' \) to be the optimal solution to \( \min_{U \in \mathbb{R}^{n \times k}} \| UB_1 D_1 - A_1 D_1 \|_1 \).

By Claim B.13, we have

\[
\| \hat{U} B_1 D_1 - A_1 D_1 \|_1 \leq \sqrt{d_1} \| U' B_1 D_1 - A_1 D_1 \|_1
\]

Due to Lemma D.11 and Lemma D.8 (in [SWZ17]) with constant probability, we have

\[
\| \hat{U} B_1 - A_1 \|_1 \leq \sqrt{d_1} \alpha_{D_1} \| U^* B_1 - A_1 \|_1, \tag{41}
\]

where \( \alpha_{D_1} = O(1) \).
Recall that $(\hat{U}^T \circ W_B^T) \in \mathbb{R}^{k \times n^2}$ denotes the matrix where the $i$-th row is the vectorization of $\hat{U}_i \otimes (W_B)_i, \forall i \in [k]$. Now, we can show,

$$\|V_0 \cdot (\hat{U}^T \circ W_B^T) - A_2\|_1 \leq \|\hat{UB}_1 - A_1\|_1 \quad \text{by } V_0 = \arg\min_{V \in \mathbb{R}^{n \times k}} \|V \cdot (\hat{U}^T \circ W_B^T) - A_2\|_1$$

$$\leq \sqrt{d_1}\|U^*B_1 - A_1\|_1 \quad \text{by Equation (41)}$$

$$\leq \sqrt{d_1}\|UB_1 - A_1\|_1 \quad \text{by } U^* = \arg\min_{U \in \mathbb{R}^{n \times k}} \|UB_1 - A_1\|_1$$

$$\leq O(\sqrt{d_1})\alpha \text{OPT} \quad \text{by Equation (40)} \quad (42)$$

We define $B_2 = \hat{U}^T \circ W_B^T$. We can compute $D_2 \in \mathbb{R}^{n^2 \times n^2}$ which is a sampling and rescaling matrix corresponding to the Lewis weights of $B_2^\top$ in $O(n^2 \text{poly}(k))$ time, and there are $d_2 = O(k \log k)$ nonzero entries on the diagonal of $D_2$.

Define $V^* \in \mathbb{R}^{n \times k}$ to be the optimal solution of $\min_{V \in \mathbb{R}^{n \times k}} \|VB_2 - A_2\|_1$, $\hat{V} = A_2D_2(B_2D_2)^\top \in \mathbb{R}^{n \times k}$, $W_0 \in \mathbb{R}^{n \times k}$ to be the optimal solution of $\min_{W \in \mathbb{R}^{n \times k}}\|W \cdot (\hat{U}^T \circ \hat{V}^T) - A_3\|_1$, and $V'$ to be the optimal solution of $\min_{V \in \mathbb{R}^{n \times k}}\|V'B_2 - A_2D_2\|_1$.

By Claim B.13, we have

$$\|\hat{V}B_2 - A_2\|_1 \leq \sqrt{d_2}\|V'B_2 - A_2D_2\|_1.$$ 

Due to Lemma D.11 and Lemma D.8 in [SWZ17]) with constant probability, we have

$$\|\hat{V}B_2 - A_2\|_1 \leq \sqrt{d_2}\|V^*B_2 - A_2\|_1,$$ 

(43)

where $\alpha_{D_2} = O(1)$.

Recall that $(\hat{U}^T \circ \hat{V}^T) \in \mathbb{R}^{k \times n^2}$ denotes the matrix for which the $i$-th row is the vectorization of $\hat{U}_i \otimes \hat{V}_i, \forall i \in [k]$. Now, we can show,

$$\|W_0 \cdot (\hat{U}^T \circ \hat{V}^T) - A_3\|_1 \leq \|\hat{V}B_2 - A_2\|_1 \quad \text{by } W_0 = \arg\min_{W \in \mathbb{R}^{n \times k}} \|W \cdot (\hat{U}^T \circ \hat{V}^T) - A_3\|_1$$

$$\leq \sqrt{d_2}\|V^*B_2 - A_2\|_1 \quad \text{by Equation (43)}$$

$$\leq \sqrt{d_2}\|V_0B_2 - A_2\|_1 \quad \text{by } V^* = \arg\min_{V \in \mathbb{R}^{n \times k}} \|V'B_2 - A_2\|_1$$

$$\leq O(\sqrt{d_1d_2})\alpha \text{OPT} \quad \text{by Equation (42)} \quad (44)$$

We define $B_3 = \hat{U}^T \circ \hat{V}^T$. We can compute $D_3 \in \mathbb{R}^{n^2 \times n^2}$ which is a sampling and rescaling matrix corresponding to the Lewis weights of $B_3^\top$ in $O(n^2 \text{poly}(k))$ time, and there are $d_3 = O(k \log k)$ nonzero entries on the diagonal of $D_3$.

Define $W^* \in \mathbb{R}^{n \times k}$ to be the optimal solution of $\min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_1$, $\hat{W} = A_3D_3(B_3D_3)^\top \in \mathbb{R}^{n \times k}$, and $W'$ to be the optimal solution of $\min_{W \in \mathbb{R}^{n \times k}}\|WB_3D_3 - A_3D_3\|_1$.

By Claim B.13, we have

$$\|\hat{W}B_3D_3 - A_3D_3\|_1 \leq \sqrt{d_3}\|W'B_3D_3 - A_3D_3\|_1.$$ 

Due to Lemma D.11 and Lemma D.8 in [SWZ17]) with constant probability, we have

$$\|\hat{W}B_3 - A_3\|_1 \leq \sqrt{d_3}\|W^*B_3 - A_3\|_1,$$ 

(45)
where $\alpha_{D3} = O(1)$. Now we can show,
\[
\|\hat{W}B_3 - A_3\|_1 \lesssim \sqrt{d_3}\|W^*B_3 - A_3\|_1, \quad \text{by Equation (45)}
\]
\[
\leq \sqrt{d_3}\|W_0B_3 - A_3\|_1, \quad \text{by } W^* = \arg \min_{W \in \mathbb{R}^{n \times k}}\|WB_3 - A_3\|_1
\]
\[
\leq O(\sqrt{d_1d_2d_3})\alpha \text{ OPT} \quad \text{by Equation (44)}
\]

Thus, it implies,
\[
\left\| \sum_{i=1}^{k} \hat{U}_i \otimes \hat{V}_i \otimes \hat{W}_i - A \right\|_1 \leq \text{poly}(k, \log n) \text{ OPT}.
\]

where $\hat{U} = A_1D_1(B_1D_1)^\dagger$, $\hat{V} = A_2D_2(B_2D_2)^\dagger$, $\hat{W} = A_3D_3(B_3D_3)^\dagger$.

\begin{algorithm}
\caption{$\ell_1$-CURT decomposition algorithm}
\begin{algorithmic}[1]
\Procedure{L1CURT$^+$}{$(A,n,k)$} \Comment{Theorem D.21}
\State $U_B, V_B, W_B \leftarrow \text{L1LowRankApproximation}(A,n,k)$. \Comment{Corollary D.18}
\State $C, R, T, U \leftarrow \text{L1CURT}(A,U_B,V_B,W_B,n,k)$. \Comment{Algorithm 29}
\State \textbf{return } $C$, $R$, $T$ and $U$.
\EndProcedure
\end{algorithmic}
\end{algorithm}

\textbf{Theorem D.21.} Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(\text{nnz}(A)) + O(n^2) \text{ poly}(k) + 2^{\tilde{O}(k^2)}$ time and outputs three matrices $C \in \mathbb{R}^{n \times c}$ with columns from $A$, $R \in \mathbb{R}^{n \times r}$ with rows from $A$, $T \in \mathbb{R}^{n \times t}$ with tubes from $A$, and a tensor $U \in \mathbb{R}^{c \times r \times t}$ with rank($U$) = $k$ such that $c = r = t = O(k \log k)$, and
\[
\left\| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_1 \leq \text{poly}(k, \log n) \min_{\text{rank} - k} \|A' - A\|_1,
\]
holds with probability $9/10$.

\textbf{Proof.} This follows by combining Corollary D.18 and Theorem D.20. \hfill \square
E Entry-wise $\ell_p$ Norm for Arbitrary Tensors, $1 < p < 2$

There is a long line of research dealing with $\ell_p$ norm-related problems [DDH+09, MM13, CDMI+13, CP15, BCKY16, YCRM16, BBC+17].

In this section, we provide several different algorithms for tensor $\ell_p$-low rank approximation. Section E.1 formally states the $\ell_p$ version of Theorem C.1 in [SWZ17]. Section E.2 presents several existence results. Section E.3 describes a tool that is able to reduce the size of the objective function from $\text{poly}(n)$ to $\text{poly}(k)$. Section E.4 discusses the case when the problem size is small. Section E.5 provides several bicriteria algorithms. Section E.6 summarizes a batch of algorithms. Section E.7 provides an algorithm for $\ell_p$ norm CURT decomposition.

Notice that if the rank-$k$ solution does not exist, then every bicriteria algorithm in Section E.5 can be stated in the form as Theorem 1.1, and every algorithm which can output a rank-$k$ solution in Section E.6 can be stated in the form as Theorem 1.2. See Section 1 for more details.

E.1 Existence results for matrix case

**Theorem E.1 ([SWZ17]).** Let $1 \leq p < 2$. Given $V \in \mathbb{R}^{k \times n}, A \in \mathbb{R}^{d \times n}$. Let $S \in \mathbb{R}^{n \times s}$ be a proper random sketching matrix. Let

$$\tilde{U} = \arg \min_{U \in \mathbb{R}^{d \times k}} \|UVS - AS\|_F^2,$$

i.e.,

$$\tilde{U} = AS(VS)^\dagger.$$

Then with probability at least 0.999,

$$\|\tilde{U}V - A\|_p^p \leq \alpha \cdot \min_{U \in \mathbb{R}^{d \times k}} \|UV - A\|_p^p,$$

(1). $S$ denotes a dense $p$-stable transform,

$s = O(k), \alpha = O(k^{1-p/2}) \log d.$

(II). $S$ denotes a sparse $p$-stable transform,

$s = O(k^5), \alpha = O(k^{5-p/2+2/p}) \log d.$

(III). $S^\top$ denotes a sampling/rescaling matrix according to the $\ell_p$ Lewis weights of $V^\top$,

$s = O(k), \alpha = O(k^{1-p/2}).$

We give the proof for completeness.

**Proof.** Let $S \in \mathbb{R}^{n \times s}$ be a sketching matrix which satisfies the property (*): $\forall c \geq 1, \tilde{U} \in \mathbb{R}^{d \times k}$ which satisfy

$$\|\tilde{U}VS - AS\|_p^p \leq c \cdot \min_{U \in \mathbb{R}^{d \times k}} \|UVS - AS\|_p^p,$$

we have

$$\|\tilde{U}V - A\|_p^p \leq c \beta_s \cdot \min_{U \in \mathbb{R}^{d \times k}} \|UV - A\|_p^p,$$

where $\beta_s \geq 1$ only depends on the sketching matrix $S$. Let

$$\forall i \in [d], (\tilde{U}^i)^\top = \arg \min_{x \in \mathbb{R}^k} \|x^\top VS - A^i S\|_2^2,$$
Due to Lemma E.8 and Lemma E.11 of [SWZ17], we have:

\[ \hat{U} = AS(VS)^\dagger. \]

Let

\[ \hat{U} = \arg \min_{U \in \mathbb{R}^{d \times k}} \|UVS - AS\|_p^p. \]

Then, we have:

\[
\|UVS - AS\|_p^p
\leq \sum_{i=1}^{d} (s^{1/p - 1/2} \|\hat{U}^iVS - A^iS\|_2^p)^p
\leq \sum_{i=1}^{d} (s^{1/p - 1/2} \|\hat{U}^iVS - A^iS\|_2)^p
\leq \sum_{i=1}^{d} (s^{1/p - 1/2} \|\hat{U}^iVS\|_p)^p
\leq s^{1-p/2} \|\hat{U}VS - AS\|_p^p.
\]

The first inequality follows using \( \forall x \in \mathbb{R}^s, \|x\|_p \leq s^{1/p - 1/2} \|x\|_2 \) since \( p < 2 \). The third inequality follows using \( \forall x \in \mathbb{R}^s, \|x\|_2 \leq \|x\|_p \) since \( p < 2 \). Thus, according to the property (*) of \( S \),

\[ \|\hat{U}V - A\|_p^p \leq s^{1-p/2} \beta_S \min_{U \in \mathbb{R}^{d \times k}} \|UV - A\|_p^p. \]

Due to Lemma E.8 and Lemma E.11 of [SWZ17], we have:

for (I), \( s = O(k), \beta_S = O(\log d), \alpha = s^{1-p/2} \beta_S = O(k^{1-p/2}) \log d \),

for (II), \( s = O(k^5), \beta_S = O(k^{2/p} \log d), \alpha = s^{1-p/2} \beta_S = O(k^{5 - 5p/2 + 2/p}) \log d \),

for (III), \( s = O(k), \beta_S = O(1), \alpha = s^{1-p/2} \beta_S = O(k^{1-p/2}) \).

\[ \square \]

### E.2 Existence results

**Theorem E.2.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), there exist three matrices \( S_1 \in \mathbb{R}^{n^2 \times s_1}, S_2 \in \mathbb{R}^{n^2 \times s_2}, S_3 \in \mathbb{R}^{n^2 \times s_3} \) such that

\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (A_1S_1X_1)_i \otimes (A_2S_2X_2)_i \otimes (A_3S_3X_3)_i - A \right\|_p^p \leq \alpha \min_{\text{rank} - k} \min_{A_k \in \mathbb{R}^{n \times n \times n}} \|A_k - A\|_p^p,
\]

holds with probability 99/100.

(I). Using a dense \( p \)-stable transform, \( s_1 = s_2 = s_3 = O(k), \alpha = O(k^{5 - 1.5p}) \log^3 n \).

(II). Using a sparse \( p \)-stable transform, \( s_1 = s_2 = s_3 = O(1), \alpha = O(k^{15 - 7.5p + 6/p}) \log^3 n \).

(III). Guessing Lewis weights, \( s_1 = s_2 = s_3 = O(k), \alpha = O(k^{5 - 1.5p}) \).
Proof. We use $\text{OPT}$ to denote

$$\text{OPT} := \min_{\text{rank} - k} \sum_{k} \| A_k - A \|^p_p.$$ 

Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define three matrices $A_1 \in \mathbb{R}^{n_1 \times n_2 n_3}, A_2 \in \mathbb{R}^{n_2 \times n_3 n_1}, A_3 \in \mathbb{R}^{n_3 n_1 \times n_2}$ such that, for any $i \in [n_1], j \in [n_2], l \in [n_3]$,

$$A_{i,j,l} = (A_1)_{i,(j-1)n_3+l} = (A_2)_{j,(i-1)n_1+l} = (A_3)_{l,(i-1)n_2+j}.$$ 

We fix $V^* \in \mathbb{R}^{n \times k}$ and $W^* \in \mathbb{R}^{n \times k}$, and use $V_1^*, V_2^*, \ldots, V^*_k$ to denote the columns of $V^*$ and $W_1^*, W_2^*, \ldots, W^*_k$ to denote the columns of $W^*$.

We consider the following optimization problem,

$$\min_{U_1, \ldots, U_k \in \mathbb{R}^n} \left\| \sum_{i=1}^k U_i \otimes V_i^* \otimes W^*_i - A \right\|_p,$$

which is equivalent to

$$\min_{U_1, \ldots, U_k \in \mathbb{R}^n} \left\| \begin{bmatrix} U_1 & U_2 & \cdots & U_k \end{bmatrix} \begin{bmatrix} V_1^* \otimes W_1^* \\ V_2^* \otimes W_2^* \\ \vdots \\ V_k^* \otimes W_k^* \end{bmatrix} - A \right\|_p.$$

We use matrix $Z_1$ to denote $V^*^T \otimes W^*^T \in \mathbb{R}^{k \times n^2}$ and matrix $U$ to denote $[U_1 \ U_2 \ \cdots \ U_k]$. Then we can obtain the following equivalent objective function,

$$\min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 - A_1 \|^p_p.$$

Choose a sketching matrix (a dense $p$-stable, a sparse $p$-stable or an $\ell_p$ Lewis weight sampling/rescaling matrix to $Z_1) \ S_1 \in \mathbb{R}^{n^2 \times s_1}$. We can obtain the optimization problem,

$$\min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 S_1 - A_1 S_1 \|^p_p = \min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^n \| U^i Z_1 S_1 - (A_1 S_1)^i \|^p_p,$$

where $U^i$ denotes the $i$-th row of matrix $U \in \mathbb{R}^{n \times k}$ and $(A_1 S_1)^i$ denotes the $i$-th row of matrix $A_1 S_1$. Instead of solving it under the $\ell_p$-norm, we consider the $\ell_2$-norm relaxation,

$$\min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 S_1 - A_1 S_1 \|^2_F = \min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^n \| U^i Z_1 S_1 - (A_1 S_1)^i \|^2_F.$$

Let $\widehat{U} \in \mathbb{R}^{n \times k}$ denote the optimal solution of the above optimization problem. Then, $\widehat{U} = A_1 S_1 (Z_1 S_1)^i$. We plug $\widehat{U}$ into the objective function under the $\ell_p$-norm. By choosing $s_1$ and by the properties of sketching matrices (a dense $p$-stable, a sparse $p$-stable or an $\ell_p$ Lewis weight sampling/rescaling matrix to $Z_1) \ S_1 \in \mathbb{R}^{n^2 \times s_1}$, we have

$$\| \widehat{U} Z_1 - A_1 \|^p_p \leq \alpha \min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 - A_1 \|^p_p = \alpha \text{OPT}.$$ 

This implies

$$\| \widehat{U} \otimes V^* \otimes W^* - A \|^p_p \leq \alpha \text{OPT}.$$
As a second step, we fix \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and convert tensor \( A \) into matrix \( A_2 \). Let matrix \( Z_2 \) denote \( \hat{U}^\top \odot W^* \). We consider the following objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \|VZ_2 - A_2\|_p^p,
\]

and the optimal cost of it is at most \( \alpha \text{ OPT} \).

We choose a sketching matrix (a dense \( p \)-stable, a sparse \( p \)-stable or an \( \ell_p \) Lewis weight sampling/rescaling matrix to \( Z_2 \)) \( S_2 \in \mathbb{R}^{n^2 \times s_2} \) and sketch on the right of the objective function to obtain the new objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \|VZ_2S_2 - A_2S_2\|_p^p = \min_{V \in \mathbb{R}^{n \times k}} \sum_{i=1}^{n} \|V^iZ_2S_2 - (A_2S_2)^i\|_p^p,
\]

where \( V^i \) denotes the \( i \)-th row of matrix \( V \) and \( (A_2S_2)^i \) denotes the \( i \)-th row of matrix \( A_2S_2 \). Instead of solving this under the \( \ell_p \)-norm, we consider the \( \ell_2 \)-norm relaxation,

\[
\min_{V \in \mathbb{R}^{n \times k}} \|VZ_2S_2 - A_2S_2\|_2^2 = \min_{V \in \mathbb{R}^{n \times k}} \sum_{i=1}^{n} \|V^i(Z_2S_2) - (A_2S_2)^i\|_2^2.
\]

Let \( \hat{V} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{V} = A_2S_2(\hat{Z}_2S_2) \). By properties of sketching matrix \( S_2 \in \mathbb{R}^{n^2 \times s_2} \), we have,

\[
\|\hat{V}Z_2 - A_2\|_p^p \leq \alpha \min_{V \in \mathbb{R}^{n \times k}} \|VZ_2 - A_2\|_p^p \leq \alpha^2 \text{ OPT},
\]

which implies

\[
\|\hat{U} \odot \hat{V} \odot W^* - A\|_p^p \leq \alpha^2 \text{ OPT},
\]

As a third step, we fix the matrices \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( \hat{V} \in \mathbb{R}^{n \times k} \). We can convert tensor \( A \in \mathbb{R}^{n \times n \times n} \) into matrix \( A_3 \in \mathbb{R}^{n^2 \times n} \). Let matrix \( Z_3 \) denote \( \hat{U}^\top \odot \hat{V}^\top \in \mathbb{R}^{k \times n^2} \). We consider the following objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_p^p,
\]

and the optimal cost of it is at most \( \alpha^2 \text{ OPT} \).

We choose sketching matrix (a dense \( p \)-stable, a sparse \( p \)-stable or an \( \ell_p \) Lewis weight sampling/rescaling matrix to \( Z_3 \)) \( S_3 \in \mathbb{R}^{n^2 \times s_3} \) and sketch on the right of the objective function to obtain the new objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3S_3 - A_3S_3\|_p^p.
\]

Instead of solving this under the \( \ell_p \)-norm, we consider the \( \ell_2 \)-norm relaxation,

\[
\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3S_3 - A_3S_3\|_2^2 = \min_{W \in \mathbb{R}^{n \times k}} \sum_{i=1}^{n} \|W^i(Z_3S_3) - (A_3S_3)^i\|_2^2.
\]

Let \( \hat{W} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{W} = A_3S_3(\hat{Z}_3S_3) \). By properties of sketching matrix \( S_3 \in \mathbb{R}^{n^2 \times s_3} \), we have,

\[
\|\hat{W}Z_3 - A_3\|_p^p \leq \alpha \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_p^p \leq \alpha^3 \text{ OPT}.
\]
Thus, we obtain,

\[
\min_{X_1 \in \mathbb{R}^{s_1 \times k}, X_2 \in \mathbb{R}^{s_2 \times k}, X_3 \in \mathbb{R}^{s_3 \times k}} \left\| \sum_{i=1}^{k} (A_1 S_1 X_1)_i \otimes (A_2 S_2 X_2)_i \otimes (A_3 S_3 X_3)_i - A \right\|_p \leq \alpha^3 \text{OPT}.
\]

According to Theorem E.1, we let \( s = s_1 = s_2 = s_3 \) and take the corresponding \( \alpha \). We can directly get the results for (I), (II) and (III). \( \square \)

### E.3 Polynomial in \( k \) size reduction

**Definition E.3** (Definition E.1 in [SWZ17]). Given a matrix \( M \in \mathbb{R}^{n \times d} \), if matrix \( S \in \mathbb{R}^{m \times n} \) satisfies

\[
\|SM\|_p^p \leq \beta \|M\|_p^p,
\]

then \( S \) has at most \( \beta \) dilation on \( M \) in the \( \ell_p \) case.

**Definition E.4** (Definition E.2 in [SWZ17]). Given a matrix \( U \in \mathbb{R}^{n \times k} \), if matrix \( S \in \mathbb{R}^{m \times n} \) satisfies

\[
\forall x \in \mathbb{R}^k, \|SUx\|_p^p \geq \frac{1}{\beta} \|Ux\|_p^p,
\]

then \( S \) has at most \( \beta \) contraction on \( U \) in the \( \ell_p \) case.

**Theorem E.5.** Given a tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and three matrices \( V_1 \in \mathbb{R}^{n_1 \times b_1}, V_2 \in \mathbb{R}^{n_2 \times b_2}, V_3 \in \mathbb{R}^{n_3 \times b_3} \), let \( X_1^* \in \mathbb{R}^{b_1 \times k}, X_2^* \in \mathbb{R}^{b_2 \times k}, X_3^* \in \mathbb{R}^{b_3 \times k} \) satisfy

\[
X_1^*, X_2^*, X_3^* = \arg\min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|V_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - A\|_p^p.
\]

Let \( S \in \mathbb{R}^{m \times n} \) have at most \( \beta_1 \geq 1 \) dilation on \( V_1 X_1^* \cdot ((V_2 X_2^*)^T \otimes (V_3 X_3^*)^T) - A_1 \) and have at most \( \beta_2 \geq 1 \) contraction on \( V_1 \) in the \( \ell_p \) case. If \( \hat{X}_1 \in \mathbb{R}^{b_1 \times k}, \hat{X}_2 \in \mathbb{R}^{b_2 \times k}, \hat{X}_3 \in \mathbb{R}^{b_3 \times k} \) satisfy

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_p^p \leq \beta \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|SV_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - SA\|_p^p,
\]

where \( \beta \geq 1 \), then

\[
\|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - A\|_p^p \leq \beta_1 \beta_2 \beta \min_{X_1, X_2, X_3} \|V_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - A\|_p^p.
\]

The proof is essentially the same as the proof of Theorem D.7:

**Proof.** Let \( A, V_1, V_2, S, X_1^*, X_2^*, X_3^*, \beta_1, \beta_2 \) be as stated in the theorem. Let \( \hat{X}_1 \in \mathbb{R}^{b_1 \times k}, \hat{X}_2 \in \mathbb{R}^{b_2 \times k}, \hat{X}_3 \in \mathbb{R}^{b_3 \times k} \) satisfy

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_p^p \leq \beta \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \|SV_1 X_1 \otimes V_2 X_2 \otimes V_3 X_3 - SA\|_p^p.
\]

Similar to the proof of Theorem D.7, we have,

\[
\|SV_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - SA\|_p^p = 2^{2-2p} \frac{1}{\beta_2} \|V_1 \hat{X}_1 \otimes V_2 \hat{X}_2 \otimes V_3 \hat{X}_3 - A\|_p^p - (2^{1-p} \frac{1}{\beta_2} + \beta_1) \|V_1 X_1^* \otimes V_2 X_2^* \otimes V_3 X_3^* - A\|_p^p.
\]
Let \( \{ (X_i, c_i, \beta_i) \}_{i=1}^k \) be any sampling according to the \( \ell_p \) Lewis weights of \( V_1 \in \mathbb{R}^{n \times b_1} \), where \( c_1 = b_1 \). Let \( T_3 \in \mathbb{R}^{c_2 \times n} \) be sampling according to the \( \ell_p \) Lewis weights of \( V_2 \in \mathbb{R}^{n \times b_2} \), where \( c_2 = b_2 \). Let \( T_3 \in \mathbb{R}^{c_3 \times n} \) be sampling according to the \( \ell_p \) Lewis weights of \( V_3 \in \mathbb{R}^{n \times b_3} \), where \( c_3 = b_3 \).

For any \( \alpha \geq 1 \), let \( X'_1 \in \mathbb{R}^{b_1 \times k}, X'_2 \in \mathbb{R}^{b_2 \times k}, X'_3 \in \mathbb{R}^{b_3 \times k} \) satisfy

\[
\| T_1 V_1 X'_1 \odot T_2 V_2 X'_2 \odot T_3 V_3 X'_3 - A(T_1, T_2, T_3) \|_p^p \leq \alpha \min_{X_1, X_2, X_3} \| T_1 V_1 X_1 \odot T_2 V_2 X_2 \odot T_3 V_3 X_3 - A(T_1, T_2, T_3) \|_p^p.
\]

First, we regard \( T_1 \) as the sketching matrix for the remainder. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have

\[
\| V_1 X'_1 \odot T_2 V_2 X'_2 \odot T_3 V_3 X'_3 - A(I, T_2, T_3) \|_p^p \leq \alpha \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \| V_1 X_1 \odot T_2 V_2 X_2 \odot T_3 V_3 X_3 - A(I, T_2, T_3) \|_p^p.
\]

Second, we regard \( T_2 \) as the sketching matrix for \( V_1 X_1 \odot V_2 X_2 \odot T_3 V_3 X_3 - A(I, I, T_3) \). Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have

\[
\| V_1 X'_1 \odot V_2 X'_2 \odot T_3 V_3 X'_3 - A(I, I, T_3) \|_p^p \leq \alpha \min_{X_1 \in \mathbb{R}^{b_1 \times k}, X_2 \in \mathbb{R}^{b_2 \times k}, X_3 \in \mathbb{R}^{b_3 \times k}} \| V_1 X_1 \odot V_2 X_2 \odot T_3 V_3 X_3 - A(I, I, T_3) \|_p^p.
\]
Third, we regard $T_3$ as the sketching matrix for $V_1X_1 \otimes V_2X_2 \otimes V_3X_3 - A$. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have
\[
\left\| \sum_{i=1}^{k} (V_1X_1')_i \otimes (V_2X_2')_i \otimes (V_3X_3')_i - A \right\|_p \approx \alpha \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1X_1)_i \otimes (V_2X_2)_i \otimes (V_3X_3)_i - A \right\|_p.
\]

\[\square\]

### E.4 Solving small problems

Combining Section B.5 in [SWZ17] and the proof of Theorem D.4, for any $p = a/b$ with $a, b$ are integers, we can obtain the $\ell_p$ version of Theorem D.4.

### E.5 Bicriteria algorithm

We present several bicriteria algorithms with different tradeoffs. We first present an algorithm that runs in nearly linear time and outputs a solution with rank $\widetilde{O}(k^3)$ in Theorem E.7. Then we show an algorithm that runs in $\text{nnz}(A)$ time but outputs a solution with rank $\text{poly}(k)$ in Theorem E.8. Then we explain an idea which is able to decrease the cubic rank to quadratic, and thus we can obtain Theorem E.9.

**Theorem E.7.** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, let $r = \widetilde{O}(k^3)$. There exists an algorithm which takes $\text{nnz}(A) \cdot \widetilde{O}(k) + n \text{poly}(k)$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times r}$ such that
\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_p \leq \widetilde{O}(k^{3-p/2}) \log^3 n \min_{\text{rank} - k} \| A_k - A \|_p
\]
holds with probability 9/10.

**Proof.** We first choose three dense Cauchy transforms $S_i \in \mathbb{R}^{n^2 \times s_i}$. According to Section B.7, for each $i \in [3]$, $A_i S_i$ can be computed in $\text{nnz}(A) \cdot \widetilde{O}(k)$ time. Then we apply Lemma E.6. We obtain three matrices $Y_i = T_i A_i S_i$, $i = 1, 2, 3$. Then we apply Lemma D.11 and the proof of Theorem D.7. We have
\[
\text{nnz}(A) \cdot \widetilde{O}(k^3) + n \text{poly}(k) = \text{nnz}(A) \cdot \widetilde{O}(k^3) + n \text{poly}(k) + \text{poly}(k).
\]

We regard $T_3$ as the sketching matrix for $V_1X_1 \otimes V_2X_2 \otimes V_3X_3 - A$. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have
\[
\left\| \sum_{i=1}^{k} (V_1X_1')_i \otimes (V_2X_2')_i \otimes (V_3X_3')_i - A \right\|_p \approx \alpha \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1X_1)_i \otimes (V_2X_2)_i \otimes (V_3X_3)_i - A \right\|_p.
\]

We first choose three dense Cauchy transforms $S_i \in \mathbb{R}^{n^2 \times s_i}$. According to Section B.7, for each $i \in [3]$, $A_i S_i$ can be computed in $\text{nnz}(A) \cdot \widetilde{O}(k)$ time. Then we apply Lemma E.6. We obtain three matrices $Y_i = T_i A_i S_i$, $i = 1, 2, 3$. Then we apply Lemma D.11 and the proof of Theorem D.7. We have
\[
\text{nnz}(A) \cdot \widetilde{O}(k^3) + n \text{poly}(k) + \text{poly}(k) = \text{nnz}(A) \cdot \widetilde{O}(k^3) + n \text{poly}(k) + \text{poly}(k).
\]

We regard $T_3$ as the sketching matrix for $V_1X_1 \otimes V_2X_2 \otimes V_3X_3 - A$. Then by Lemma D.11 in [SWZ17] and Theorem D.7, we have
\[
\left\| \sum_{i=1}^{k} (V_1X_1')_i \otimes (V_2X_2')_i \otimes (V_3X_3')_i - A \right\|_p \approx \alpha \min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (V_1X_1)_i \otimes (V_2X_2)_i \otimes (V_3X_3)_i - A \right\|_p.
\]
**Algorithm 31** \(\ell_p\)-Low Rank Approximation, Bicriteria Algorithm, rank-\(\tilde{O}(k^2)\), Input Sparsity Time

1: \textbf{procedure} \textsc{LpBicriteriaAlgorithm}(A, n, k) \hfill \triangleright \text{Corollary E.10}
2: \hspace{1em} \text{Let } s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k).
3: \hspace{1em} For each \(i \in [3]\), choose \(S_i \in \mathbb{R}^{n^2 \times s_i}\) to be the composition of a sparse \(p\)-stable transform and a dense \(p\)-stable transform. \hfill \triangleright \text{Part (I,II) of Theorem E.2}
4: \hspace{1em} Compute \(A_1 \cdot S_1, A_2 \cdot S_2\).
5: \hspace{1em} For each \(i \in [2]\), choose \(T_i\) to be a sampling and rescaling diagonal matrix according to the Lewis weights of \(A_i S_i\), with \(t_i = \tilde{O}(k)\) nonzero entries.
6: \hspace{1em} \(C \leftarrow A(T_1, T_2, I)\).
7: \hspace{1em} \(B^{i+(j-1)s_1} \leftarrow \text{vec}( (T_1 A_1 S_1)_i \otimes (T_2 A_2 S_2)_j), \forall i \in [s_1], j \in [s_2].\)
8: \hspace{1em} Form objective function \(\min_W \|WB - C\|_1\).
9: \hspace{1em} Run \(\ell_p\)-regression solver to find \(\hat{W}\).
10: \hspace{1em} Construct \(\hat{U}\) by copying \((A_1 S_1)_i\) to the \((i, j)\)-th column of \(\hat{U}\).
11: \hspace{1em} Construct \(\hat{V}\) by copying \((A_2 S_2)_j\) to the \((i, j)\)-th column of \(\hat{V}\).
12: \textbf{return} \(\hat{U}, \hat{V}, \hat{W}\).
13: \textbf{end procedure}

As for \(\ell_1\), notice that if we first apply a sparse Cauchy transform, we can reduce the rank of the matrix to \(\text{poly}(k)\). Then we can apply a dense Cauchy transform and further reduce the dimension, while only incurring another \(\text{poly}(k)\) factor in the approximation ratio. By combining sparse \(p\)-stable and dense \(p\)-stable transforms, we can improve the running time from \(\text{nnz}(A) \cdot \tilde{O}(k^2)\) to be \(\text{nnz}(A)\) by losing some additional \(\text{poly}(k)\) factors in the approximation ratio.

\[\text{Proof.}\] We first choose three sparse \(p\)-stable transforms \(S_i \in \mathbb{R}^{n^2 \times s_i}\). According to Section B.7, for each \(i \in [3]\), \(A_i S_i\) can be computed in \(O(\text{nnz}(A))\) time. Then we apply Lemma E.6, and can obtain three matrices \(Y_1 = T_1 A_1 S_1, Y_2 = T_2 A_2 S_2, Y_3 = T_3 A_3 S_3\) and a tensor \(C = A(T_1, T_2, T_3)\). Note that for each \(i \in [3]\), \(Y_i\) can be computed in \(n \text{ poly}(k)\) time. Because \(C = A(T_1, T_2, T_3)\) and \(T_1, T_2, T_3 \in \mathbb{R}^{n \times O(k)}\) are three sampling and rescaling matrices, \(C\) can be computed in \(\text{nnz}(A) + \tilde{O}(k^3)\) time. At the end, we just need to run an \(\ell_p\)-regression solver to find the solution to the problem,

\[
\min_{X \in \mathbb{R}^{n^2 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} X_{i,j,l} (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_p^p,
\]

where \((Y_1)_i\) denotes the \(i\)-th column of matrix \(Y_1\). Since the size of the above problem is only \(\text{poly}(k)\), it can be solved in \(\text{poly}(k)\) time.

**Theorem E.9.** Given a 3rd order tensor \(A \in \mathbb{R}^{n \times n \times n}\), for any \(k \geq 1\), \(\epsilon \in (0, 1)\), let \(r = \tilde{O}(k^2)\). There exists an algorithm which takes \(\text{nnz}(A) \cdot \tilde{O}(k) + n \text{ poly}(k) + \text{poly}(k)\) time and outputs three matrices \(U, V, W \in \mathbb{R}^{n \times r}\) such that

\[
\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_p^p \leq \tilde{O}(k^{3-1.5p}) \log^3 n \min_{\text{rank} - k} \|A_k - A\|_p^p
\]

holds with probability \(9/10\).

**Proof.** The proof is similar to Theorem D.14.
Corollary E.10. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, $\epsilon \in (0,1)$, let $r = \tilde{O}(k^2)$. There exists an algorithm which takes $\text{nnz}(A) + n \text{poly}(k) + \text{poly}(k)$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times r}$ such that

$$\left\| \sum_{i=1}^{r} U_i \otimes V_i \otimes W_i - A \right\|_p \leq \text{poly}(k, \log n) \min_{\text{rank}-k} \|A_k - A\|_p$$

holds with probability at least 9/10.

E.6 Algorithms

In this section, we show two different algorithms by using different kind of sketches. One is shown in Theorem E.11 which gives a fast running time. Another one is shown in Theorem E.12 which gives the best approximation ratio.

Theorem E.11. Given a 3rd tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm which takes $O(\text{nnz}(A)) + n \text{poly}(k) + 2^{\tilde{O}(k^2)}$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that,

$$\|U \otimes V \otimes W - A\|_p \leq \text{poly}(k, \log n) \min_{\text{rank}-k} \|A' - A\|_p$$

holds with probability at least 9/10.

Proof. First, we apply part (II) of Theorem E.2. Then, guessing $S_i$ requires $n \tilde{O}(k)$ time. Second, we use Lemma E.6 to reduce the size of the objective function from $O(n^3)$ to $\text{poly}(k)$ in $n \text{poly}(k)$ time by only losing a constant factor in approximation ratio. Third, we use Claim B.15 to relax the objective function from entry-wise $\ell_p$-norm to Frobenius norm, and this step causes us to lose some other $\text{poly}(k)$ factors in approximation ratio. As a last step, we use Theorem C.45 to solve the Frobenius norm objective function.

Theorem E.12. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, there exists an algorithm that takes $n \tilde{O}(k)2^{\tilde{O}(k^2)}$ time and output three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that,

$$\|U \otimes V \otimes W - A\|_p \leq \tilde{O}(k^{3-1.5p}) \min_{\text{rank}-k} \|A' - A\|_p$$

holds with probability at least 9/10.

Proof. First, we apply part (III) of Theorem E.2. Then, guessing $S_i$ requires $n \tilde{O}(k)$ time. Second, we use Lemma E.6 to reduce the size of the objective function from $O(n^3)$ to $\text{poly}(k)$ in polynomial time while only losing a constant factor in approximation ratio. Third, we solve the small optimization problem.

E.7 CURT decomposition

Theorem E.13. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $k \geq 1$, and let $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$ denote a rank-$k$, $\alpha$-approximation to $A$. Then there exists an algorithm which takes $O(\text{nnz}(A)) + O(n^2)\text{poly}(k)$ time and outputs three matrices $C \in \mathbb{R}^{n \times c}$ with columns from $A$, $R \in \mathbb{R}^{n \times r}$ with rows from $A$, $T \in \mathbb{R}^{r \times t}$ with tubes from $A$, and a tensor $U \in \mathbb{R}^{c \times r \times t}$ with rank($U$) = $k$ such that $c = r = t = \text{O}(k \log k \log \log k)$, and

$$\left\| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_p \leq \tilde{O}(k^{3-1.5p}) \alpha \min_{\text{rank}-k} \|A' - A\|_p$$

holds with probability 9/10.
Proof. We define 

\[
\text{OPT} := \min_{\text{rank } - k A} \|A' - A\|_p^p.
\]

We already have three matrices \(U_B \in \mathbb{R}^{n \times k}, V_B \in \mathbb{R}^{n \times k}\) and \(W_B \in \mathbb{R}^{n \times k}\) and these three matrices provide a rank-k, \(\alpha\) approximation to \(A\), i.e.,

\[
\left\| \sum_{i=1}^k (U_B)_i \otimes (V_B)_i \otimes (W_B)_i - A \right\|_p^p \leq \alpha \text{OPT}.
\] (46)

Let \(B_1 = V_B^T \otimes W_B^T \in \mathbb{R}^{k \times n^2}\) denote the matrix where the \(i\)-th row is the vectorization of 
\((V_B)_i \otimes (W_B)_i\). By Section B.3 in [SWZ17], we can compute \(D_1 \in \mathbb{R}^{n^2 \times n^2}\) which is a sampling and rescaling matrix corresponding to the Lewis weights of \(B_1^T\) in \(O(n^2 \text{ poly}(k))\) time, and there are \(d_1 \approx O(k \log k \log \log k)\) nonzero entries on the diagonal of \(D_1\). Let \(A_i \in \mathbb{R}^{n \times n^2}\) denote the matrix obtained by flattening \(A\) along the \(i\)-th direction, for each \(i \in [k]\).

Define \(U^* \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{U \in \mathbb{R}^{n \times k}} \|UB_1 - A_1\|_p^p\), \(\hat{U} = A_1D_1(B_1D_1)\dagger \in \mathbb{R}^{n \times k}\), \(V_0 \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{V \in \mathbb{R}^{n \times k}} \|V \cdot (\hat{U}^T \otimes W_B^T) - A_2\|_p^p\), and \(U'\) to be the optimal solution to \(\min_{U \in \mathbb{R}^{n \times k}} \|UB_1D_1 - A_1D_1\|_p^p\).

By Claim B.13, we have

\[
\|\hat{U}B_1D_1 - A_1D_1\|_p^p \leq d_1^{1-p/2}\|U'B_1D_1 - A_1D_1\|_p^p.
\]

Due to Lemma E.11 and Lemma E.8 in [SWZ17], with constant probability, we have

\[
\|\hat{U}B_1 - A_1\|_p^p \leq d_1^{1-p/2}\alpha_{D_1}\|U^*B_1 - A_1\|_p^p,
\] (47)

where \(\alpha_{D_1} = O(1)\).

Recall that \((\hat{U}^T \otimes W_B^T) \in \mathbb{R}^{k \times n^2}\) denotes the matrix where the \(i\)-th row is the vectorization of \(\hat{U}_i \otimes (W_B)_i, \forall i \in [k]\). Now, we can show,

\[
\|V_0 \cdot (\hat{U}^T \otimes W_B^T) - A_2\|_p^p \leq \|\hat{U}B_1 - A_1\|_p^p \quad \text{by } V_0 = \arg \min_{V \in \mathbb{R}^{n \times k}} \|V \cdot (\hat{U}^T \otimes W_B^T) - A_2\|_p^p
\]

\[
\lesssim d_1^{1-p/2}\|U^*B_1 - A_1\|_p^p \quad \text{by Equation (47)}
\]

\[
\leq d_1^{1-p/2}\|UB_1 - A_1\|_p^p \quad \text{by } U^* = \arg \min_{U \in \mathbb{R}^{n \times k}} \|UB_1 - A_1\|_p^p
\]

\[
\leq O(d_1^{1-p/2})\alpha \text{OPT}.
\] (48)

We define \(B_2 = \hat{U}^T \otimes W_B^T\). We can compute \(D_2 \in \mathbb{R}^{n^2 \times n^2}\) which is a sampling and rescaling matrix corresponding to the \(\ell_p\) Lewis weights of \(B_2^T\) in \(O(n^2 \text{ poly}(k))\) time, and there are \(d_2 \approx O(k \log k \log \log k)\) nonzero entries on the diagonal of \(D_2\).

Define \(V^* \in \mathbb{R}^{n \times k}\) to be the optimal solution of \(\min_{V \in \mathbb{R}^{n \times k}} \|VB_2 - A_2\|_p^p\), \(\hat{V} = A_2D_2(B_2D_2)\dagger \in \mathbb{R}^{n \times k}\), \(W_0 \in \mathbb{R}^{n \times k}\) to be the optimal solution of \(\min_{W \in \mathbb{R}^{n \times k}} \|W \cdot (\hat{U}^T \otimes \hat{V}^T) - A_3\|_p^p\), and \(V'\) to be the optimal solution of \(\min_{V \in \mathbb{R}^{n \times k}} \|VB_2D_2 - A_2D_2\|_p^p\).

By Claim B.13, we have

\[
\|\hat{V}B_2D_2 - A_2D_2\|_p^p \leq d_2^{1-p/2}\|V'B_2D_2 - A_2D_2\|_p^p.
\]
Due to Lemma E.11 and Lemma E.8 in [SWZ17], with constant probability, we have
\[
\|\tilde{V}B_2 - A_2\|_p^p \leq d_2^{1-p/2}\alpha_D \|V^*B_2 - A_2\|_p^p,
\]
(49)
where \(\alpha_D = O(1)\).

Recall that \((\hat{U}^\top \odot \hat{V}^\top) \in \mathbb{R}^{k \times n^2}\) denotes the matrix for which the \(i\)-th row is the vectorization of \(\hat{U}_i \odot \hat{V}_i\), \(\forall i \in [k]\). Now, we can show,
\[
\|W_0 \cdot (\hat{U}^\top \odot \hat{V}^\top) - A_3\|_p^p \leq \|\hat{V}B_2 - A_2\|_p^p \quad \text{by } W_0 = \arg \min_{W \in \mathbb{R}^{n \times k}} \|W \cdot (\hat{U}^\top \odot \hat{V}^\top) - A_3\|_p^p
\]
\[
\leq d_2^{1-p/2}\|V^*B_2 - A_2\|_p^p \quad \text{by Equation (49)}
\]
\[
\leq d_2^{1-p/2}\|V_0B_2 - A_2\|_p^p \quad \text{by } V^* = \arg \min_{V \in \mathbb{R}^{n \times k}} \|VB_2 - A_2\|_p^p
\]
\[
\leq O((d_1d_2)^{1-p/2})\alpha \text{OPT} \quad \text{by Equation (48)}
\]
(50)

We define \(B_3 = \hat{U}^\top \odot \hat{V}^\top\). We can compute \(D_3 \in \mathbb{R}^{n^2 \times n^2}\) which is a sampling and rescaling matrix corresponding to the \(\ell_p\) Lewis weights of \(B_3^\top\) in \(O(n^2 \text{poly}(k))\) time, and there are \(d_3 = O(k \log k \log \log k)\) nonzero entries on the diagonal of \(D_3\).

Define \(W^* \in \mathbb{R}^{n \times k}\) to be the optimal solution to \(\min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_p^p\), \(\hat{W} = A_3D_3(B_3D_3)^\top \in \mathbb{R}^{n \times k}\), and \(\hat{W}^\top\) to be the optimal solution to \(\min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_p^p\).

By Claim B.13, we have
\[
\|\hat{W}B_3D_3 - A_3D_3\|_p^p \leq d_3^{1-p/2}\|\hat{W}'B_3D_3 - A_3D_3\|_p^p.
\]

Due to Lemma E.11 and Lemma E.8 in [SWZ17], with constant probability, we have
\[
\|\hat{W}B_3 - A_3\|_p^p \leq d_3^{1-p/2}\alpha_{D_3}\|W^*B_3 - A_3\|_p^p,
\]
(51)
where \(\alpha_{D_3} = O(1)\). Now we can show,
\[
\|\hat{W}B_3 - A_3\|_p^p \leq d_3^{1-p/2}\|W^*B_3 - A_3\|_p^p \quad \text{by Equation (51)}
\]
\[
\leq d_3^{1-p/2}\|W_0B_3 - A_3\|_p^p \quad \text{by } W^* = \arg \min_{W \in \mathbb{R}^{n \times k}} \|WB_3 - A_3\|_p^p
\]
\[
\leq O((d_1d_2d_3)^{1-p/2})\alpha \text{OPT} \quad \text{by Equation (50)}
\]

Thus, it implies,
\[
\left\| \sum_{i=1}^k \hat{U}_i \odot \hat{V}_i \odot \hat{W}_i - A \right\|_p^p \leq \text{poly}(k, \log n) \text{OPT}.
\]

where \(\hat{U} = A_1D_1(B_1D_1)^\top\), \(\hat{V} = A_2D_2(B_2D_2)^\top\), \(\hat{W} = A_3D_3(B_3D_3)^\top\).

\[
\square
\]
F  Robust Subspace Approximation (Asymmetric Norms for Arbitrary Tensors)

Recently, [CW15b] and [CW15a] study the linear regression problem and low-rank approximation problem under M-Estimator loss functions. In this section, we extend the matrix version of the low rank approximation problem to tensors, i.e., in particular focusing on tensor low-rank approximation under M-Estimator norms. Note that M-Estimators are very different from Frobenius norm and Entry-wise $\ell_1$ norm, which are symmetric norms. Namely, flattening the tensor objective function along any of the dimensions does not change the cost if the norm is Frobenius or Entry-wise $\ell_1$-norm. However, for M-Estimator norms, we cannot flatten the tensor along all three dimensions. This property makes the tensor low-rank approximation problem under M-Estimator norms more difficult. This section can be split into two independent parts. Section F.2 studies the $\ell_1$-$\ell_2$-$\ell_2$ norm setting, and Section F.3 studies the $\ell_1$-$\ell_1$-$\ell_2$ norm setting.

F.1 Preliminaries

**Definition F.1** (Nice functions for M-Estimators, $\mathcal{M}_2$, $\mathcal{L}_p$, [CW15a]). We say an M-Estimator is nice if $M(x) = M(-x)$, $M(0) = 0$, $M$ is non-decreasing in $|x|$, there is a constant $C_M > 0$ and a constant $p \geq 1$ so that for all $a,b \in \mathbb{R}_{>0}$ with $a \geq b$, we have

$$C_M \frac{|a|}{|b|} \leq \frac{M(a)}{M(b)} \leq \left( \frac{a}{b} \right)^p,$$

and also that $M(x)^{\frac{1}{p}}$ is subadditive, that is, $M(x+y)^{\frac{1}{p}} \leq M(x)^{\frac{1}{p}} + M(y)^{\frac{1}{p}}$.

Let $\mathcal{M}_2$ denote the set of such nice M-estimators, for $p = 2$. Let $\mathcal{L}_p$ denote M-Estimators with $M(x) = |x|^p$ and $p \in [1,2)$.

F.2 $\ell_1$-Frobenius (a.k.a $\ell_1$-$\ell_2$-$\ell_2$) norm

Section F.2.1 presents basic definitions and facts for the $\ell_1$-$\ell_2$-$\ell_2$ norm setting. Section F.2.2 introduces some useful tools. Section F.2.3 presents the “no dilation” and “no contraction” bounds, which are the key ideas for reducing the problem to a “generalized” Frobenius norm low rank approximation problem. Finally, we provide our algorithms in Section F.2.6.

F.2.1 Definitions

We first give the definition for the $v$-norm of a tensor, and then give the definition of the $v$-norm for a matrix and a weighted version of the $v$-norm for a matrix.

**Definition F.2** (Tensor $v$-norm). For an $n \times n \times n$ tensor $A$, we define the $v$-norm of $A$, denoted $\|A\|_v$, to be

$$\left( \sum_{i=1}^{n} M(\|A_{i,*,*}\|_F) \right)^{1/p},$$

where $A_{i,*,*}$ is the $i$-th face of $A$ (along the 1st direction), and $p$ is a parameter associated with the function $M()$, which defines a nice M-Estimator.
Definition F.3 (Matrix \(v\)-norm). For an \(n \times d\) matrix \(A\), we define the \(v\)-norm of \(A\), denoted \(\|A\|_v\), to be
\[
\sum_{i=1}^{n} M(\|A_{i,*}\|_2)^{1/p},
\]
where \(A_{i,*}\) is the \(i\)-th row of \(A\), and \(p\) is a parameter associated with the function \(M()\), which defines a nice \(M\)-Estimator.

Definition F.4. Given matrix \(A \in \mathbb{R}^{n \times d}\), let \(A_{i,*}\) denote the \(i\)-th row of \(A\). Let \(T_S \subset [n]\) denote the indices \(i\) such that \(e_i\) is chosen for \(S\). Using a probability vector \(q\) and a sampling and rescaling matrix \(S \in \mathbb{R}^{n \times n}\) from \(q\), we will estimate \(\|A\|_v\) using \(S\) and a re-weighted version, \(\|S \cdot \|_{v,w'}\) of \(\|\cdot\|_v\), with
\[
\|S A\|_{v,w'} = \left(\sum_{i \in T_S} w'_i M(\|A_{i,*}\|_2)\right)^{1/p},
\]
where \(w'_i = w_i/q_i\). Since \(w'\) is generally understood, we will usually just write \(\|SA\|_v\). We will also need an “entrywise row-weighted” version:
\[
\|SA\| = \left(\sum_{i \in T_S} \frac{w_i}{q_i} \|A_{i,*}\|_M\right)^{1/p} = \left(\sum_{i \in T_S, j \in [d]} \frac{w_j}{q_i} M(A_{i,j})\right)^{1/p},
\]
where \(A_{i,j}\) denotes the entry in the \(i\)-th row and \(j\)-th column of \(A\).

Fact F.5. For \(p = 1\), for any two matrices \(A\) and \(B\), we have \(\|A + B\|_v \leq \|A\|_v + \|B\|_v\). For any two tensors \(A\) and \(B\), we have \(\|A + B\|_v \leq \|A\|_v + \|B\|_v\).

F.2.2 Sampling and rescaling sketches

Note that Lemmas 42 and 44 in [CW15a] are stronger than stated. In particular, we do not need to assume \(X\) is a square matrix. For any \(m \geq z\), if \(X \in \mathbb{R}^{d \times m}\), then we have the same result.

Lemma F.6 (Lemma 42 in [CW15a]). Let \(\rho > 0\) and integer \(z > 0\). For sampling matrix \(S\), suppose for a given \(y \in \mathbb{R}^d\) with failure probability \(\delta\) it holds that \(\|SAy\|_M = (1 \pm 1/10)\|Ay\|_M\). There is \(K_1 = O(z^2/C_M)\) so that with failure probability \(\delta(K_N/C_M)(1+p)^d\), for a constant \(K_N\), any rank-\(z\) matrix \(X \in \mathbb{R}^{d \times m}\) has the property that if \(\|AX\|_v \geq K_1 \rho\), then \(\|SAX\|_v \geq \rho\), and that if \(\|AX\|_v \leq \rho/K_1\), then \(\|SAX\|_v \leq \rho\).

Lemma F.7 (Lemma 44 in [CW15a]). Let \(\delta, \rho > 0\) and integer \(z > 0\). Given matrix \(A \in \mathbb{R}^{n \times d}\), there exists a sampling and rescaling matrix \(S \in \mathbb{R}^{n \times n}\) with \(r = O(\gamma(A,M,\delta)\epsilon^{-\epsilon}z^2 \log(z/\epsilon) \log(1/\delta))\) nonzero entries such that, with probability at least \(1 - \delta\), for any rank-\(z\) matrix \(X \in \mathbb{R}^{d \times m}\), we have either
\[
\|SAX\|_v \geq \rho,
\]
or
\[
(1 - \epsilon)\|AX\|_v - \epsilon \rho \leq \|SAX\|_v \leq (1 + \epsilon)\|AX\|_v + \epsilon \rho.
\]
Lemma F.8 (Lemma 43 in [CW15a]). For $r > 0$, let $\hat{r} = r/\gamma(A, M, w)$, and let $q \in \mathbb{R}^n$ have
\[ q_i = \min\{1, \hat{r} \gamma_i(A, M, w)\}. \]

Let $S$ be a sampling and rescaling matrix generated using $q$, with weights as usual $w'_i = w_i/q_i$. Let $W \in \mathbb{R}^{d \times z}$, and $\delta > 0$. There is an absolute constant $C$ so that for $\hat{r} \geq Cz \log(1/\delta)/\epsilon^2$, with probability at least $1 - \delta$, we have
\[ (1 - \epsilon)\|AW\|_{v, w} \leq \|SAW\|_{v, w} \leq (1 + \epsilon)\|AW\|_{v, w}. \]

F.2.3 No dilation and no contraction

Lemma F.9. Given matrices $A \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times d}$, let $V^* = \arg\min\limits_{\text{rank}-k} \|UV - A\|_v$. If $S \in \mathbb{R}^{s \times n}$ has at most $c_1$-dilation on $UV^* - A$, i.e.,
\[ \|S(UV^* - A)\|_v \leq c_1\|UV^* - A\|_v, \]
and it has at most $c_2$-contraction on $U$, i.e.,
\[ \forall x \in \mathbb{R}^d, \|SUx\|_v \geq \frac{1}{c_2}\|Ux\|_v, \]
then $S$ has at most $(c_2, c_1 + \frac{1}{c_2})$-contraction on $(U, A)$, i.e.,
\[ \forall \text{rank}-k V \in \mathbb{R}^{d \times m}, \|SUV - SA\|_v \geq \frac{1}{c_2}\|UV - A\|_v - (c_1 + \frac{1}{c_2})\|UV^* - A\|_v. \]

Proof. Let $A \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times d}$ and $S \in \mathbb{R}^{s \times n}$ be the same as that described in the lemma. Let $(V - V^*)_j$ denote the $j$-th column of $V - V^*$. Then $\forall \text{rank}-k V \in \mathbb{R}^{d \times m}$,
\begin{align*}
\|SUV - SA\|_v &\geq \|SUV - SUV^*\|_v - \|SUV^* - SA\|_v \\
&\geq \|SUV - SUV^*\|_v - c_1\|UV^* - A\|_v \\
&= \|SU(V - V^*)\|_v - c_1\|UV^* - A\|_v \\
&= \sum\limits_{j=1}^{m}\|SU(V - V^*)_j\|_v - c_1\|UV^* - A\|_v \\
&\geq \sum\limits_{j=1}^{m}\frac{1}{c_2}\|U(V - V^*)_j\|_v - c_1\|UV^* - A\|_v \\
&= \frac{1}{c_2}\|UV - UV^*\|_v - c_1\|UV^* - A\|_v \\
&\geq \frac{1}{c_2}\|UV - A\|_v - \frac{1}{c_2}\|UV^* - A\|_v - c_1\|UV^* - A\|_v \\
&= \frac{1}{c_2}\|UV - A\|_v - \left(\frac{1}{c_2} + c_1\right)\|UV^* - A\|_v, \\
\end{align*}
where the first inequality follows by the triangle inequality, the second inequality follows since $S$ has at most $c_1$ dilation on $UV^* - A$, the third inequality follows since $S$ has at most $c_2$ contraction on $U$, and the fourth inequality follows by the triangle inequality. \qed

114
Claim F.10. Given matrix $A \in \mathbb{R}^{n \times m}$, for any distribution $p = (p_1, p_2, \cdots, p_n)$ define random variable $X$ such that $X = \|A_i\|_2/p_i$ with probability $p_i$ where $A_i$ is the $i$-th row of matrix $A$. Then take $m$ independent samples $X^1, X^2, \cdots, X^m$, and let $Y = \frac{1}{m} \sum_{j=1}^{m} X^j$. We have

$$
\Pr[Y \leq 1000 \|A\|_v] \geq .999.
$$

Proof. We can compute the expectation of $X^j$, for any $j \in [m]$,

$$
\mathbb{E}[X^j] = \sum_{i=1}^{n} \frac{\|A_i\|_2}{p_i} = \|A\|_v.
$$

Then $\mathbb{E}[Y] = \frac{1}{m} \sum_{j=1}^{m} \mathbb{E}[X^j] = \|A\|_v$. Using Markov’s inequality, we have

$$
\Pr[Y \geq \|A\|_v] \leq .001.
$$

Lemma F.11. For any fixed $U^* \in \mathbb{R}^{n \times d}$ and rank-$k$ $V^* \in \mathbb{R}^{d \times m}$ with $d = \text{poly}(k)$, there exists an algorithm that takes $\text{poly}(n,d)$ time to compute a sampling and rescaling diagonal matrix $S \in \mathbb{R}^{n \times n}$ with $s = \text{poly}(k)$ nonzero entries such that, with probability at least .999, we have: for all rank-$k$ $V \in \mathbb{R}^{d \times m}$,

$$
\|U^*V^* - U^*V\|_v \lesssim \|SU^*V^* - SU^*V\|_v \lesssim \|U^*V^* - U^*V\|_v.
$$

Lemma F.12 (No dilation). Given matrices $A \in \mathbb{R}^{n \times m}, U^* \in \mathbb{R}^{n \times d}$ with $d = \text{poly}(k)$, define $V^* \in \mathbb{R}^{d \times m}$ to be the optimal solution $\min_{\text{rank-}k \ V \in \mathbb{R}^{d \times m}} \|U^*V^* - A\|_v$. Choose a sampling and rescaling diagonal matrix $S \in \mathbb{R}^{n \times n}$ with $s = \text{poly}(k)$ according to Lemma F.8. Then with probability at least .99, we have: for all rank-$k$ $V \in \mathbb{R}^{d \times m}$,

$$
\|SU^*V^* - SA\|_v \lesssim \|U^*V^* - U^*V\|_v + O(1)\|U^*V^* - A\|_v \lesssim \|U^*V - A\|_v.
$$

Proof. Using Claim F.10 and Lemma F.11, we have with probability at least .99, for all rank-$k$ $V \in \mathbb{R}^{d \times m}$,

$$
\begin{align*}
\|SU^*V - SA\|_v &\leq \|SU^*V - SU^*V\|_v + \|SU^*V^* - SA\|_v \\
&\lesssim \|SU^*V - SU^*V\|_v + O(1)\|U^*V^* - A\|_v \\
&\lesssim \|U^*V - U^*V^*\|_v + O(1)\|U^*V^* - A\|_v \\
&\lesssim \|U^*V - A\|_v + O(1)\|U^*V^* - A\|_v \\
&\lesssim \|U^*V - A\|_v.
\end{align*}
$$

Lemma F.13 (No contraction). Given matrices $A \in \mathbb{R}^{n \times m}, U^* \in \mathbb{R}^{n \times d}$ with $d = \text{poly}(k)$, define $V^* \in \mathbb{R}^{d \times m}$ to be the optimal solution $\min_{\text{rank-}k \ V \in \mathbb{R}^{d \times m}} \|U^*V - A\|_v$. Choose a sampling and rescaling diagonal matrix $S \in \mathbb{R}^{n \times n}$ with $s = \text{poly}(k)$ according to Lemma F.8. Then with probability at least .99, we have: for all rank-$k$ $V \in \mathbb{R}^{d \times m}$,

$$
\|U^*V - A\|_v \lesssim \|SU^*V - SA\|_v + O(1)\|U^*V^* - A\|_v.
$$

Proof. This follows by Lemma F.9, Claim F.10 and Lemma F.12.
F.2.4 Oblivious sketches, MSKETCH

In this section, we recall a concept called M-sketches for M-estimators which is defined in [CW15b]. M-sketch is an oblivious sketch for matrices.

**Theorem F.14** (Theorem 3.1 in [CW15b]). Let OPT denote \( \min_{x \in \mathbb{R}^d} \| Ax - b \|_G \). There is an algorithm that in \( O(\text{nnz}(A)) + \text{poly}(d \log n) \) time, with constant probability finds \( x' \) such that \( \| Ax' - b \|_G \leq O(1) \text{OPT} \).

**Definition F.15** (M-Estimator sketches or MSKETCH [CW15b]). Given parameters \( N, n, m, b > 1 \), define \( h_{\text{max}} = \lfloor \log_b(n/m) \rfloor \), \( \beta = (b - b^{-h_{\text{max}}})/(b - 1) \) and \( s = N h_{\text{max}} \). For each \( p \in [n] \), \( \sigma_p, g_p, h_p \) are generated (independently) in the following way,

\[
\sigma_p \leftarrow \pm 1, \quad g_p \in [N], \quad h_p \leftarrow t,
\]

chosen with equal probability, chosen with equal probability, chosen with probability \( 1/(\beta^t) \) for \( t \in \{0, 1, \ldots, h_{\text{max}}\} \).

For each \( p \in [n] \), we define \( j_p = g_p + N h_p \). Let \( w \in \mathbb{R}^s \) denote the scaling vector such that, for each \( j \in [s] \),

\[
w_j = \begin{cases} \beta b^{h_p}, & \text{if there exists } p \in [n] \text{ s.t. } j = j_p, \\ 0, & \text{otherwise.} \end{cases}
\]

Let \( \overline{S} \in \mathbb{R}^{N h_{\text{max}} \times n} \) be such that, for each \( j \in [s], \) for each \( p \in [n] \),

\[
\overline{S}_{j,p} = \begin{cases} \sigma_p, & \text{if } j = g_p + N \cdot h_p, \\ 0, & \text{otherwise.} \end{cases}
\]

Let \( D_w \) denote the diagonal matrix where the \( i \)-th entry on the diagonal is the \( i \)-th entry of \( w \). Let \( S = D_w \overline{S} \). We say \((\overline{S}, w)\) or \( S \) is an MSKETCH.

**Definition F.16** (Tensor \( \| \cdot \|_{v,w} \)-norm). For a tensor \( A \in \mathbb{R}^{d \times n_1 \times n_2} \) and a vector \( w \in \), we define

\[
\| A \|_{v,w} = \sum_{i=1}^{d} w_i \| A_{i,,} \|_F.
\]

Let \((\overline{S}, w)\) denote an MSKETCH, and let \( S = D_w \overline{S} \). If \( v \) corresponds to a scale-invariant M-Estimator, then for any three matrices \( U, V, W \), we have the following,

\[
\| (\overline{S} U) \otimes V \otimes W \|_{v,w} = \| (D_w \overline{S} U) \otimes V \otimes W \|_v = \| (SU) \otimes V \otimes W \|_v.
\]

**Fact F.17.** For a tensor \( A \in \mathbb{R}^{n \times n \times n} \), let \( S \in \mathbb{R}^{s \times n} \) denote an MSKETCH (defined in F.15) with \( s = \text{poly}(k, \log n) \). Then \( S A \) can be computed in \( O(\text{nnz}(A)) \) time.

**Lemma F.18.** For any fixed \( U^* \in \mathbb{R}^{n \times d} \) and rank-\( k \) \( V^* \in \mathbb{R}^{d \times m} \) with \( d = \text{poly}(k) \), let \( S \in \mathbb{R}^{s \times n} \) denote an MSKETCH (defined in Definition F.15) with \( s = \text{poly}(k, \log n) \) rows. Then with probability at least 0.999, we have: for all rank-\( k \) \( V \in \mathbb{R}^{d \times m} \),

\[
\| U^* V^* - U^* V \|_v \lesssim \| S U^* V^* - S U^* V \|_v \lesssim \| U^* V^* - U^* V \|_v.
\]
Lemma F.19 (No dilation, Theorem 3.4 in [CW15b]). Given matrices $A \in \mathbb{R}^{n \times m}$, $U^* \in \mathbb{R}^{n \times d}$ with $d = \text{poly}(k)$, define $V^* \in \mathbb{R}^{d \times m}$ to be the optimal solution to $\min_{\text{rank}-k V \in \mathbb{R}^{d \times m}} \| U^* V - A \|_v$. Choose an MSketch $S \in \mathbb{R}^{s \times n}$ with $s = \text{poly}(k, \log n)$ according to Definition F.15. Then with probability at least .99, we have: for all rank-$k V \in \mathbb{R}^{d \times m}$,
\[
\| SU^* V - SA \|_v \lesssim \| U^* V^* - U^* V \|_v + O(1) \| U^* V^* - A \|_v \lesssim \| U^* V - A \|_v.
\]

Lemma F.20 (No contraction). Given matrices $A \in \mathbb{R}^{n \times m}$, $U^* \in \mathbb{R}^{n \times d}$ with $d = \text{poly}(k)$, define $V^* \in \mathbb{R}^{d \times m}$ to be the optimal solution to $\min_{\text{rank}-k V \in \mathbb{R}^{d \times m}} \| U^* V - A \|_v$. Choose an MSketch $S \in \mathbb{R}^{s \times n}$ with $s = \text{poly}(k, \log n)$ according to Definition F.15. Then with probability at least .99, we have: for all rank-$k V \in \mathbb{R}^{d \times m}$,
\[
\| U^* V - A \|_v \lesssim \| SU^* V - SA \|_v + O(1) \| U^* V^* - A \|_v.
\]

F.2.5 Running time analysis

Lemma F.21. Given a tensor $A \in \mathbb{R}^{n \times d \times d}$, let $S \in \mathbb{R}^{s \times n}$ denote an MSketch with $s$ rows. Let $SA$ denote a tensor that has size $s \times d \times d$. For each $i \in \{2, 3\}$, let $(SA)_i \in \mathbb{R}^{d \times ds}$ denote a matrix obtained by flattening tensor $SA$ along the $i$-th dimension. For each $i \in \{2, 3\}$, let $S_i \in \mathbb{R}^{ds \times s_i}$ denote a CountSketch transform with $s_i$ columns. For each $i \in \{2, 3\}$, let $T_i \in \mathbb{R}^{s_i \times d}$ denote a CountSketch transform with $t_i$ rows. Then
(I) For each $i \in \{2, 3\}$, $(SA)_i S_i$ can be computed in $O(\text{nnz}(A))$ time.
(II) For each $i \in \{2, 3\}$, $T_i (SA)_i S_i$ can be computed in $O(\text{nnz}(A))$ time.

Proof. Proof of Part (I). First note that $(SA)_2 S_2$ has size $n \times S_2$. Thus for each $i \in [d], j \in [s_2]$, we have,
\[
((SA)_2 S_2)_{i,j} = \sum_{x'=1}^{ds} ((SA)_2)_{i,x'} (S_2)_{x',j} \quad \text{by } (SA)_2 \in \mathbb{R}^{d \times ds}, S_2 \in \mathbb{R}^{ds \times s_2}
\]
\[
= \sum_{y=1}^{d} \sum_{z=1}^{s} ((SA)_2)_{i,(y-1)s+z} (S_2)_{(y-1)s+z,j} \quad \text{by unflattening}
\]
\[
= \sum_{y=1}^{d} \sum_{z=1}^{s} (SA)_{z,i,y} (S_2)_{(y-1)s+z,j} \quad \text{by unflattening}
\]
\[
= \sum_{y=1}^{d} \sum_{z=1}^{s} \left( \sum_{x=1}^{n} S_{z,x} A_{x,i,y} \right) (S_2)_{(y-1)s+z,j}
\]
\[
= \sum_{y=1}^{d} \sum_{z=1}^{s} S_{z,x} \cdot A_{x,i,y} \cdot (S_2)_{(y-1)s+z,j}.
\]

For each nonzero entry $A_{x,i,y}$, there is only one $z$ such that $S_{z,x}$ is nonzero. Thus there is only one $j$ such that $(S_2)_{(y-1)s+z,j}$ is nonzero. It means that $A_{x,i,y}$ can only affect one entry of $((SA)_2 S_2)_{i,j}$. Thus, $(SA)_2 S_2$ can be computed in $O(\text{nnz}(A))$ time. Similarly, we can compute $(SA)_3 S_3$ in $O(\text{nnz}(A))$ time.

117
Proof of Part (II). Note that $T_2(SA)_2S_2$ has size $t_2 \times s_2$. Thus for each $i \in [t_2], j \in [s_2]$, we have,

\[
(T_2(SA)_2S_2)_{i,j} = \sum_{x=1}^{d} \sum_{y=1}^{ds} (T_2)_{i,x}((SA)_2)_{x,y'}(S_2)_{y',j} \\
= \sum_{x=1}^{d} \sum_{y=1}^{s} \sum_{z=1}^{s} (T_2)_{i,x}(SA)_{z,x,y}(S_2)_{(y-1)s+z,j} \\
= \sum_{x=1}^{d} \sum_{y=1}^{s} \sum_{z=1}^{s} (T_2)_{i,x} \left( \sum_{u=1}^{n} S_{z,w}A_{w,x,y} \right) (S_2)_{(y-1)s+z,j} \\
= \sum_{x=1}^{d} \sum_{y=1}^{s} \sum_{z=1}^{s} \sum_{u=1}^{n} (T_2)_{i,x} \cdot S_{z,w} \cdot A_{w,x,y} \cdot (S_2)_{(y-1)s+z,j}.
\]

For each nonzero entry $A_{w,x,y}$, there is only one $z$ such that $S_{z,w}$ is nonzero. There is only one $i$ such that $(T_2)_{i,x}$ is nonzero. Since there is only one $z$ to make $S_{z,w}$ nonzero, there is only one $j$, such that $(S_2)_{(y-1)s+z,j}$ is nonzero. Thus, $T_2(SA)_2S_2$ can be computed in $O(nnz(A))$ time. Similarly, we can compute $T_3(SA)_3S_3$ in $O(nnz(A))$ time.

\]

F.2.6 Algorithms

We first give a “warm-up” algorithm in Theorem F.22 by using a sampling and rescaling matrix. Then we improve the running time to be polynomial in all the parameters by using an oblivious sketch, and thus we obtain Theorem F.23.

Algorithm 32 $\ell_1$-Frobenius($\ell_1$-$\ell_2$-$\ell_2$) Low-rank Approximation Algorithm, poly($k$)-approximation

1: procedure L122TENSORLOWRANKAPPROX($A, n, k$) \triangledown Theorem F.22
2: $\epsilon \leftarrow 0.1$.
3: $s \leftarrow \text{poly}(k, 1/\epsilon)$.
4: Guess a sampling and rescaling matrix $S \in \mathbb{R}^{s \times n}$.
5: $s_2 \leftarrow s_3 \leftarrow O(k/\epsilon)$.
6: $r \leftarrow s_2 s_3$.
7: Choose sketching matrices $S_2 \in \mathbb{R}^{n \times s_2}, S_3 \in \mathbb{R}^{n \times s_3}$.
8: Compute $(SA)_2S_2, (SA)_3S_3$.
9: Form $\tilde{V} \in \mathbb{R}^{n \times r}$ by repeating $(SA)_2S_2, s_3$ times according to Equation (59).
10: Form $\tilde{W} \in \mathbb{R}^{n \times r}$ by repeating $(SA)_3S_3, s_2$ times according to Equation (60).
11: Form objective function $\min_{U \in \mathbb{R}^{n \times r}} \|U : (\tilde{V}^\top \circ \tilde{W}^\top) - A_1\|_F$.
12: Use a linear regression solver to find a solution $\tilde{U}$.
13: Take the best solution found over all guesses.
14: return $\tilde{U}, \tilde{V}, \tilde{W}$.
15: end procedure

\]

Theorem F.22. Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$, let $r = O(k^2)$. There exists
an algorithm which takes $n^{\text{poly}(k)}$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times k}$ such that
\[
\|U \otimes V \otimes W - A\|_v \leq \text{poly}(k) \min_{\text{rank } k} \|A' - A\|_v,
\]
holds with probability at least $9/10$.

Proof. We define OPT as follows,
\[
\text{OPT} = \min_{U, V, W \in \mathbb{R}^{n \times k}} \|U \otimes V \otimes W - A\|_v = \min_{U, V, W \in \mathbb{R}^{n \times k}} \left\| \sum_{i=1}^k U_i \otimes V_i \otimes W_i - A \right\|_v.
\]
Let $A_1 \in \mathbb{R}^{n \times n^2}$ denote the matrix obtained by flattening tensor $A$ along the 1st dimension. Let $U^* \in \mathbb{R}^{n \times k}$ denote the optimal solution. We fix $U^* \in \mathbb{R}^{n \times k}$, and consider this objective function,
\[
\min_{V, W \in \mathbb{R}^{n \times k}} \|U^* \otimes V \otimes W - A\|_v \equiv \min_{V, W \in \mathbb{R}^{n \times k}} \left\| U^* \cdot (V^T \otimes W^T) - A_1 \right\|_v,
\] (52)
which has cost at most OPT, and where $V^T \otimes W^T \in \mathbb{R}^{k \times n^2}$ denotes the matrix for which the $i$-th row is a vectorization of $V_i \otimes W_i, \forall i \in [k]$. (Note that $V_i \in \mathbb{R}^n$ is the $i$-th column of matrix $V \in \mathbb{R}^{n \times k}$). Choose a sampling and rescaling diagonal matrix $S \in \mathbb{R}^{n \times n}$ according to $U^*$, which has $s = \text{poly}(k)$ non-zero entries. Using $S$ to sketch on the left of the objective function when $U^*$ is fixed (Equation (52)), we obtain a smaller problem,
\[
\min_{V, W \in \mathbb{R}^{n \times k}} \|(SU^*) \otimes V \otimes W - SA\|_v \equiv \min_{V, W \in \mathbb{R}^{n \times k}} \left\| SU^* \cdot (V^T \otimes W^T) - SA_1 \right\|_v.
\] (53)
Let $V', W'$ denote the optimal solution to the above problem, i.e.,
\[
V', W' = \text{arg min}_{V, W \in \mathbb{R}^{n \times k}} \|(SU^*) \otimes V \otimes W - SA\|_v.
\]
Then using properties (no dilation Lemma F.12 and no contraction Lemma F.13) of $S$, we have
\[
\|U^* \otimes V' \otimes W' - A\|_v \leq \alpha \text{OPT},
\]
where $\alpha$ is an approximation ratio determined by $S$.

By definition of $\| \cdot \|_v$ and $\| \cdot \|_2 \leq \| \cdot \|_1 \leq \sqrt{\text{dim}} \| \cdot \|_2$, we can rewrite Equation (53) in the following way,
\[
\|(SU^*) \otimes V \otimes W - SA\|_v
= \sum_{i=1}^s \left( \sum_{j=1}^n \sum_{l=1}^n \left( ((SU^*) \otimes V \otimes W)_{i,j,l} - (SA)_{i,j,l} \right)^2 \right)^{\frac{1}{2}}
\leq \sqrt{s} \sum_{i=1}^s \sum_{j=1}^n \sum_{l=1}^n \left( ((SU^*) \otimes V \otimes W)_{i,j,l} - (SA)_{i,j,l} \right)^{\frac{1}{2}}
= \sqrt{s} \|(SU^*) \otimes V \otimes W - SA\|_F.
\] (54)
Given the above properties of $S$ and Equation (54), for any $\beta \geq 1$, let $V'', W''$ denote a $\beta$-approximate solution of $\min_{V, W \in \mathbb{R}^{n \times k}} \|(SU^*) \otimes V \otimes W - SA\|_F$, i.e.,

119
\[(SU^*) \otimes V'' \otimes W'' - SA\|_F \leq \beta \cdot \min_{V,W} \|(SU^*) \otimes V \otimes W - SA\|_F. \quad (55)\]

Then,
\[\|U^* \otimes V'' \otimes W'' - A\|_F \leq \sqrt{s} \alpha \beta \cdot \text{OPT}. \quad (56)\]

In the next few paragraphs we will focus on solving Equation (55). We start by fixing \(W^* \in \mathbb{R}^{n \times k}\) to be the optimal solution of
\[\min_{V,W} \|(SU^*) \otimes V \otimes W - SA\|_F. \]

We use \((SA)_2 \in \mathbb{R}^{n \times ns}\) to denote the matrix obtained by flattening the tensor \(SA \in \mathbb{R}^{s \times n \times n}\) along the second direction. We use \(Z_2 = (SU^*)^\top \otimes (W^*)^\top \in \mathbb{R}^{k \times ns}\) to denote the matrix where the \(i\)-th row is the vectorization of \((SU^*)_i \otimes W^*_i\). We can consider the following objective function,
\[\min_{V} \|VZ_2 - (SA)_2\|_F.\]

Choosing a sketching matrix \(S_2 \in \mathbb{R}^{n \times s_2}\) with \(s_2 = O(k/\epsilon)\) gives a smaller problem,
\[\min_{V} \|VZ_2S_2 - (SA)_2S_2\|_F. \]

Letting \(\hat{V} = (SA)_2S_2(Z_2S_2)^\top \in \mathbb{R}^{n \times k}\), then
\[
\|\hat{V}Z_2 - (SA)_2\|_F \leq (1 + \epsilon) \min_{V} \|VZ_2 - (SA)_2\|_F
\]
\[
= (1 + \epsilon) \min_{V} \|V((SU^*)^\top \otimes (W^*)^\top) - (SA)_2\|_F
\]
\[
= (1 + \epsilon) \min_{V} \|(SU^*) \otimes V \otimes W^* - SA\|_F \quad \text{by unflattening}
\]
\[
= (1 + \epsilon) \min_{V,W} \|(SU^*) \otimes V \otimes W - SA\|_F. \quad \text{by definition of } W^* \quad (57)
\]

We define \(D_2 \in \mathbb{R}^{n^2 \times n^2}\) to be a diagonal matrix obtained by copying the \(n \times n\) identity matrix \(s\) times on \(n\) diagonal blocks of \(D_2\). Then it has \(ns\) nonzero entries. Thus, \(D_2\) also can be thought of as a matrix that has size \(n^2 \times ns\).

We can think of \((SA)_2S_2 \in \mathbb{R}^{n \times s_2}\) as follows,
\[
(SA)_2S_2 = (A(S,I,I))_2S_2
\]
\[
= \underbrace{A_2 \cdot D_2}_{n \times n^2} \cdot \underbrace{S_2}_{n^2 \times n^2} \cdot \underbrace{S_2}_{ns \times s_2}
\]
\[
= A_2 \cdot \begin{bmatrix} c_{2,1} I_n \newline c_{2,2} I_n \newline \vdots \newline c_{2,n} I_n \end{bmatrix} \cdot S_2
\]
where \(I_n\) is an \(n \times n\) identity matrix, \(c_{2,i} \geq 0\) for each \(i \in [n]\), and the number of nonzero \(c_{2,i}\) is \(s\).

For the last step, we fix \(SU^*\) and \(\hat{V}\). We use \((SA)_3 \in \mathbb{R}^{n \times ns}\) to denote the matrix obtained by flattening the tensor \(SA \in \mathbb{R}^{s \times n \times n}\) along the third direction. We use \(Z_3 = (SU^*)^\top \otimes \hat{V}^\top \in \mathbb{R}^{k \times ns}\).
to denote the matrix where the $i$-th row is the vectorization of $(SU^*)_i \otimes \widehat{V}_i$. We can consider the following objective function,

$$\min_{W \in \mathbb{R}^{n \times k}} \| WZ_3 - (SA)_3 \|_F.$$ 

Choosing a sketching matrix $3 \in \mathbb{R}^{n \times n}$ with $s_3 = O(k/\epsilon)$ gives a smaller problem,

$$\min_{W \in \mathbb{R}^{n \times k}} \| WZ_3S_3 - (SA)_3S_3 \|_F.$$ 

Let $\widehat{W} = (SA)_3S_3(Z_3S_3)^\dagger \in \mathbb{R}^{n \times k}$. Then

$$\| \widehat{W}Z_3 - (SA)_3 \|_F \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \| WZ_3 - (SA)_3 \|_F \quad \text{by property of } S_3$$

$$= (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \| W((SU^*)^\top \otimes \widehat{V}^\top) - (SA)_3 \|_F \quad \text{by definition } Z_3$$

$$= (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \| (SU^*) \otimes \widehat{V} \otimes W - SA \|_F \quad \text{by unflattening}$$

$$\leq (1 + \epsilon)^2 \|(SU^*) \otimes V \otimes W - SA \|_F \quad \text{by Equation (57)}$$

We define $3 \in \mathbb{R}^{n^2 \times n^2}$ to be a diagonal matrix formed by copying the $n \times n$ identity matrix $s$ times on $n$ diagonal blocks of $3$. Then it has $ns$ nonzero entries. Thus, $3$ also can be thought of as a matrix that has size $n^2 \times ns$ and $3$ is uniquely determined by $S$.

Similarly as to the 2nd dimension, for the 3rd dimension, we can think of $(SA)_3S_3$ as follows,

$$(SA)_3S_3 = (A(S, I, I))_3S_3$$

$$= \frac{\mathbf{A}_3}{n \times n^2} \cdot \frac{\mathbf{D}_3}{n^2 \times n^2} \cdot \frac{\mathbf{S}_3}{ns \times ns} \quad \text{by } D_3 \text{ can be thought of as having size } n^2 \times ns$$

$$= \mathbf{A}_3 \cdot \begin{bmatrix} c_{3,1}I_n \\ c_{3,2}I_n \\ \vdots \\ c_{3,n}I_n \end{bmatrix} \cdot \mathbf{S}_3$$

where $I_n$ is an $n \times n$ identity matrix, $c_{3,i} \geq 0$ for each $i \in [n]$ and the number of nonzero $c_{3,i}$ is $s$.

Overall, we have proved that,

$$\min_{X_2, X_3} \|(SU^*) \otimes (A_2D_2S_2X_2) \otimes (A_3D_3S_3X_3) - SA \|_F \leq (1 + \epsilon)^2 \|(SU^*) \otimes V \otimes W - SA \|_F, \quad (58)$$

where diagonal matrix $D_2 \in \mathbb{R}^{n^2 \times n^2}$ (with $ns$ nonzero entries) and $3 \in \mathbb{R}^{n^2 \times n^2}$ (with $ns$ nonzero entries) are uniquely determined by diagonal matrix $S \in \mathbb{R}^{n \times n}$ ($s$ nonzero entries). Let $X'_2$ and $X'_3$ denote the optimal solution to the above problem (Equation (58)). Let $V'' = (A_2D_2S_2X'_2) \in \mathbb{R}^{n \times k}$ and $W'' = (A_3D_3S_3X'_3) \in \mathbb{R}^{n \times k}$. Then we have

$$\|U^* \otimes V'' \otimes W'' - A\|_V \leq \sqrt{\alpha \beta \text{ OPT}}.$$ 

We construct matrix $\widehat{V} \in \mathbb{R}^{n \times s_3}$ by copying matrix $(SA)_2S_2 \in \mathbb{R}^{n \times s_2}$ $s_3$ times,

$$\widehat{V} = [(SA)_2S_2 \ (SA)_2S_2 \ \cdots \ (SA)_2S_2] \quad (59)$$
We construct matrix \( \tilde{W} \in \mathbb{R}^{n \times s_2 s_3} \) by copying the \( i \)-th column of matrix \( (SA)_3 S_3 \in \mathbb{R}^{n \times s_2} \) into \( (i-1)s_2 + 1, \ldots, is_2 \) columns of \( \tilde{W} \),
\[
\tilde{W} = [(SA)_3 S_3)_1 \cdots ((SA)_3 S_3)_1 ((SA)_3 S_3)_2 \cdots ((SA)_3 S_3)_2 \cdots ((SA)_3 S_3)_s \cdots ((SA)_3 S_3)_s]. \quad (60)
\]

Although we don’t know \( S \), we can guess all of the possibilities. For each possibility, we can find a solution \( \tilde{U} \in \mathbb{R}^{n \times s_2 s_3} \) to the following problem,

\[
\begin{align*}
&\min_{U \in \mathbb{R}^{n \times s_2 s_3}} \left\| \sum_{i=1}^{s_2} \sum_{j=1}^{s_3} U_{(i-1)s_3 + j} \otimes ((SA)_2 S_2)_i \otimes ((SA)_3 S_3)_j - A \right\|_v \\
= &\min_{U \in \mathbb{R}^{n \times s_2 s_3}} \left\| \sum_{i=1}^{s_2} \sum_{j=1}^{s_3} U_{(i-1)s_3 + j} \cdot \text{vec}((SA)_2 S_2)_i \otimes ((SA)_3 S_3)_j - A_1 \right\|_v \\
= &\min_{U \in \mathbb{R}^{n \times s_2 s_3}} \left\| U \cdot (\tilde{V}^T \circ \tilde{W}^T) - A_1 \right\|_v \\
= &\min_{U \in \mathbb{R}^{n \times s_2 s_3}} \| UZ - A_1 \|_v \\
= &\min_{U \in \mathbb{R}^{n \times s_2 s_3}} \sum_{i=1}^{s_2 s_3} \| U^i Z - A_1 \|_2,
\end{align*}
\]

where the first step follows by flattening the tensor along the 1st dimension, \( U_{(i-1)s_3 + j} \) denotes the \( (i-1)s_3 + j \)-th column of \( U \in \mathbb{R}^{n \times s_2 s_3} \), \( A_1 \in \mathbb{R}^{n \times n^2} \) denotes the matrix obtained by flattening tensor \( A \) along the 1st dimension, the second step follows since \( \tilde{V}^T \circ \tilde{W}^T \in \mathbb{R}^{s_2 s_3 \times n^2} \) is defined to be the matrix where the \( (i-1)s_3 + j \)-th row is vectorization of \( ((SA)_2 S_2)_i \otimes ((SA)_3 S_3)_j \), the fourth step follows by defining \( Z \) to be \( \tilde{V}^T \circ \tilde{W}^T \), and the last step follows by definition of \( \| \cdot \|_v \) norm. Thus, we obtain a multiple regression problem and it can be solved directly by using [CW13, NN13].

Finally, we take the best \( \tilde{U}, \tilde{V}, \tilde{W} \) over all the guesses. The entire running time is dominated by the number of guesses, which is \( n^{\text{poly}(k)} \). This completes the proof. \( \square \)

**Theorem F.23.** Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), let \( r = O(k^2) \). There exists an algorithm which takes \( O(mnz(A)) + n \text{ poly}(k, \log n) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times r} \) such that

\[
\| U \otimes V \otimes W - A \|_v \leq \text{poly}(k, \log n) \min_{\text{rank} - k} A' - A \|_v
\]

holds with probability at least \( 9/10 \).

**Proof.** We define OPT as follows,

\[
\text{OPT} = \min_{U, V, W \in \mathbb{R}^{n \times k}} \| U \otimes V \otimes W - A \|_v = \min_{U, V, W \in \mathbb{R}^{n \times k}} \left\| \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i - A \right\|_v.
\]
Algorithm 33 $\ell_1$-Frobenius($\ell_1,\ell_2,\ell_2$) Low-rank Approximation Algorithm, \( \text{poly}(k, \log n) \)-approximation

Let \( A_1 \in \mathbb{R}^{n \times n^2} \) denote the matrix obtained by flattening tensor \( A \) along the 1st dimension. Let \( U^* \in \mathbb{R}^{n \times k} \) denote the optimal solution. We fix \( U^* \in \mathbb{R}^{n \times k} \), and consider the objective function,

\[
\min_{V,W \in \mathbb{R}^{n \times k}} \| U^* \odot V \odot W - A \|_v \equiv \min_{V,W \in \mathbb{R}^{n \times k}} \left\| U^* \cdot (V^T \odot W^T) - A_1 \right\|_v ,
\]

which has cost at most \( \text{OPT} \), and where \( V^T \odot W^T \in \mathbb{R}^{k \times n^2} \) denotes the matrix for which the \( i \)-th row is a vectorization of \( V_i \odot W_i \), \( V_i \in [k] \). (Note that \( V_i \in \mathbb{R}^n \) is the \( i \)-th column of matrix \( V \in \mathbb{R}^{n \times k} \).) Choose an (oblivious) MSketch \( S \in \mathbb{R}^{s \times n} \) with \( s = \text{poly}(k, \log n) \) according to Definition F.15. Using MSketch \( S \), we sketch on the left of the objective function when \( U^* \) is fixed (Equation (61)), we obtain a smaller problem,

\[
\min_{V,W \in \mathbb{R}^{n \times k}} \|(SU^*) \odot V \odot W - SA_1\|_v \equiv \min_{V,W \in \mathbb{R}^{n \times k}} \left\| SU^* \cdot (V^T \odot W^T) - SA_1 \right\|_v .
\]

Let \( V', W' \) denote the optimal solution to the above problem, i.e.,

\[
V', W' = \arg\min_{V,W \in \mathbb{R}^{n \times k}} \|(SU^*) \odot V \odot W - SA_1\|_v .
\]

Then using properties (no dilation Lemma F.19 and no contraction Lemma F.20) of \( S \), we have

\[
\| U^* \odot V' \odot W' - A \|_v \leq \alpha \text{OPT} ,
\]

where \( \alpha \) is an approximation ratio determined by \( S \).
By definition of \( \| \cdot \|_v \) and \( \| \cdot \|_2 \leq \| \cdot \|_1 \leq \sqrt{\dim} \| \cdot \|_2 \), we can rewrite Equation (62) in the following way,

\[
\| (SU^*) \otimes V \otimes W - SA \|_v \\
= \sum_{i=1}^{s} \left( \sum_{j=1}^{n} \sum_{l=1}^{n} \left( ((SU^*) \otimes V \otimes W)_{i,j,l} - (SA)_{i,j,l} \right)^2 \right)^{\frac{1}{2}} \\
\leq \sqrt{s} \left( \sum_{i=1}^{s} \sum_{j=1}^{n} \sum_{l=1}^{n} \left( ((SU^*) \otimes V \otimes W)_{i,j,l} - (SA)_{i,j,l} \right)^2 \right)^{\frac{1}{2}} \\
= \sqrt{s} \| (SU^*) \otimes V \otimes W - SA \|_F
\]

Using the properties of \( S \) and Equation (63), for any \( \beta \geq 1 \), let \( V'', W'' \) denote a \( \beta \)-approximation solution of \( \min_{V,W \in \mathbb{R}^{n \times k}} \| (SU^*) \otimes V \otimes W - SA \|_F \), i.e.,

\[
\| (SU^*) \otimes V'' \otimes W'' - SA \|_F \leq \beta \cdot \min_{V,W \in \mathbb{R}^{n \times k}} \| (SU^*) \otimes V \otimes W - SA \|_F.
\]

Then,

\[
\| U^* \otimes V'' \otimes W'' - A \|_v \leq \sqrt{s} \alpha \beta \cdot \text{OPT}.
\]

Let \( \tilde{A} \) denote \( SA \). Choose \( S_i \in \mathbb{R}^{n \times s_i} \) to be Gaussian matrix with \( s_i = O(k/\epsilon), \forall \{2, 3\} \). By a similar proof as in Theorem F.22, we have if \( X'_2, X'_3 \) is a \( \beta \)-approximate solution to

\[
\min_{X_2, X_3} \| (SU^*) \otimes (\tilde{A}_2 S_2 X_2) \otimes (\tilde{A}_3 S_3 X_3) - SA \|_F,
\]

then,

\[
\| U^* \otimes (\tilde{A}_2 S_2 X_2) \otimes (\tilde{A}_3 S_3 X_3) - A \|_v \leq \sqrt{s} \alpha \beta.
\]

To reduce the size of the objective function from \( \text{poly}(n) \) to \( \text{poly}(k/\epsilon) \), we use perform an “input sparsity reduction” (in Lemma C.3). Note that, we do not need to use this idea to optimize the running time in Theorem F.22. The running time of Theorem F.22 is dominated by guessing sampling and rescaling matrices. (That running time is \( \gg \text{nnz}(A) \)). Choose \( T_i \in \mathbb{R}^{t_i \times n} \) to be a sparse subspace embedding matrix (CountSketch transform) with \( t_i = \text{poly}(k, 1/\epsilon), \forall \{2, 3\} \). Applying the proof of Lemma C.3 here, we obtain, if \( X'_2, X'_3 \) is a \( \beta \)-approximate solution to

\[
\min_{X_2, X_3} \| (SU^*) \otimes (T_2(SA)_2 S_2 X_2) \otimes (T_3(SA)_3 S_3 X_3) - SA \|_F,
\]

then,

\[
\| U^* \otimes ((SA)_2 S_2 X_2) \otimes ((SA)_3 S_3 X_3) - A \|_v \leq \sqrt{s} \alpha \beta.
\]

Similar to the bicriteria results in Section C.4, Equation (66) indicates that we can construct a bicriteria solution by using two matrices \((SA)_2 S_2\) and \((SA)_3 S_3\). The next question is how to obtain the final results \( \tilde{U}, \tilde{V}, \tilde{W} \). We first show how to obtain \( \tilde{U} \). Then we show to construct \( \tilde{V} \) and \( \tilde{W} \).
To obtain $\hat{U}$, we need to solve a regression problem related to two matrices $V, W$ and a tensor $A(I, T_2, T_3)$. We construct matrix $V \in \mathbb{R}^{t_2 \times s_2 s_3}$ by copying matrix $T_2(SA)_2 S_2 \in \mathbb{R}^{t_2 \times s_2}$ s_2 times,

$$V = [T_2(SA)_2 S_2 \; T_2(SA)_2 S_2 \; \cdots \; T_2(SA)_2 S_2].$$  \hfill (67)

We construct matrix $W \in \mathbb{R}^{t_3 \times s_2 s_3}$ by copying the $i$-th column of matrix $T_3(SA)_3 S_3 \in \mathbb{R}^{t_3 \times s_3}$ into $(i - 1)s_2 + 1, \cdots, is_2$ columns of $W$,

$$W = [F_1 \; \cdots \; F_1 \; F_2 \; \cdots \; F_2 \; \cdots \; F_{s_3} \; \cdots \; F_{s_3}],$$  \hfill (68)

where $F = T_3(SA)_3 S_3$.

Thus, to obtain $\hat{U} \in \mathbb{R}^{s_2 s_3}$, we just need to use a linear regression solver to solve a smaller problem,

$$\min_{U \in \mathbb{R}^{s_2 s_3}} \| U \cdot (V^T \circ W^T) - A(I, T_2, T_3) \|_F,$$

which can be solved in $O(\text{nnz}(A)) + n \text{poly}(k, \log n)$ time. We will show how to obtain $V$ and $W$.

We construct matrix $\tilde{V} \in \mathbb{R}^{n \times s_2 s_3}$ by copying matrix $(SA)_2 S_2 \in \mathbb{R}^{n \times s_2}$ s_3 times,

$$\tilde{V} = [(SA)_2 S_2 \; (SA)_2 S_2 \; \cdots \; (SA)_2 S_2].$$  \hfill (69)

We construct matrix $\tilde{W} \in \mathbb{R}^{n \times s_2 s_3}$ by copying the $i$-th column of matrix $T_3(SA)_3 S_3 \in \mathbb{R}^{n \times s_3}$ into $(i - 1)s_2 + 1, \cdots, is_2$ columns of $\tilde{W}$,

$$\tilde{W} = [F_1 \; \cdots \; F_1 \; F_2 \; \cdots \; F_2 \; \cdots \; F_{s_3} \; \cdots \; F_{s_3}],$$  \hfill (70)

where $F = (SA)_3 S_3$.

\section{\textcolor{blue}{F.3} $\ell_1-\ell_1-\ell_2$ norm}

Section F.3.1 presents some definitions and useful facts for the tensor $\ell_1-\ell_1-\ell_2$ norm. We provide some tools in Section F.3.2. Section F.3.3 presents a key idea which shows we are able to reduce the original problem to a new problem under entry-wise $\ell_1$ norm. Section F.3.4 presents several existence results. Finally, Section F.3.6 introduces several algorithms with different tradeoffs.

\subsection{\textcolor{blue}{F.3.1} Definitions}

\textbf{Definition F.24. (Tensor $u$-norm)} For an $n \times n \times n$ tensor $A$, we define the $u$-norm of $A$, denoted $\|A\|_u$, to be

$$\left( \sum_{i=1}^{n} \sum_{j=1}^{n} M(\|A_{i,j,*}\|_2) \right)^{1/p},$$

where $A_{i,j,*}$ is the $(i, j)$-th tube of $A$, and $p$ is a parameter associated with the function $M()$, which defines a nice $M$-Estimator.

\textbf{Definition F.25. (Matrix $u$-norm)} For an $n \times n$ matrix $A$, we define $u$-norm of $A$, denoted $\|A\|_u$, to be

$$\left( \sum_{i=1}^{n} M(\|A_{i,*}\|_2) \right)^{1/p},$$

where $A_{i,*}$ is the $i$-th row of $A$, and $p$ is a parameter associated with the function $M()$, which defines a nice $M$-Estimator.
Fact F.26. For $p = 1$, for any two matrices $A$ and $B$, we have $\|A + B\|_u \leq \|A\|_u + \|B\|_u$. For any two tensors $A$ and $B$, we have $\|A + B\|_u \leq \|A\|_u + \|B\|_u$.

F.3.2 Projection via Gaussians

Definition F.27. Let $p \geq 1$. Let $\ell_p^{S^{n-1}}$ be an infinite dimensional $\ell_p$ metric which consists of a coordinate for each vector $r$ in the unit sphere $S^{n-1}$. Define function $f : S^{n-1} \to \mathbb{R}$. The $\ell_1$-norm of any such $f$ is defined as follows:

$$\|f\|_1 = \left( \int_{r \in S^{n-1}} |f(r)|^p dr \right)^{1/p}.$$ 

Claim F.28. Let $f_v(r) = \langle v, r \rangle$. There exists a universal constant $\alpha_p$ such that

$$\|f_v\|_p = \alpha_p \|v\|_2.$$

Proof. We have,

$$\|f_v\|_p = \left( \int_{r \in S^{n-1}} |\langle v, r \rangle|^p dr \right)^{1/p}
= \left( \int_{\theta \in S^{n-1}} \|v\|_2^p \cdot |\cos \theta|^p d\theta \right)^{1/p}
= \|v\|_2 \left( \int_{\theta \in S^{n-1}} |\cos \theta|^p d\theta \right)^{1/p}
= \alpha_p \|v\|_2.$$

This completes the proof. \qed

Lemma F.29. Let $G \in \mathbb{R}^{k \times n}$ denote i.i.d. random Gaussian matrices with rescaling. Then for any $v \in \mathbb{R}^n$, we have

$$\Pr[(1 - \epsilon)\|v\|_2 \leq \|Gv\|_1 \leq (1 + \epsilon)\|v\|_2] \geq 1 - 2^{-\Omega(k\epsilon^2)}.$$

Proof. For each $i \in [k]$, we define $X_i = \langle v, g_i \rangle$, where $g_i \in \mathbb{R}^n$ is the $i$-th row of $G$. Then $X_i = \sum_{j=1}^n v_j g_{i,j}$ and $\mathbb{E}[|X_i|] = \alpha_p \|v\|_2$. Define $Y = \sum_{i=1}^k |X_i|$. We have $\mathbb{E}[Y] = k\alpha_1 \|v\|_2 = k\alpha_1$.

We can show

$$\Pr[Y \geq (1 + \epsilon)\alpha_1 k] = \Pr[e^{sY} \geq e^{s(1 + \epsilon)\alpha_1 k}]$$
for all $s > 0$

$$\leq \mathbb{E}[e^{sY}]/e^{s(1 + \epsilon)\alpha_1 k} \quad \text{by Markov’s inequality}$$

$$= e^{-s(1 + \epsilon)\alpha_1 k} \cdot \mathbb{E}\left[\prod_{i=1}^k e^{s|X_i|}\right] \quad \text{by } Y = \sum_{i=1}^k |X_i|$$

$$= e^{-s(1 + \epsilon)\alpha_1 k} \cdot (\mathbb{E}[e^{s|X_1|}])^k$$

It remains to bound $\mathbb{E}[e^{s|X_1|}]$. Since $X_1 \sim \mathcal{N}(0, 1)$, we have that $X_1$ has density function $e^{-t^2/2}$.
Thus, we have,

\[
E[e^{s|X_1|}] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{s|t|} \cdot e^{-t^2/2} \, dt
= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{s^2/2} \cdot e^{-(|t|-s)^2/2} \, dt
= e^{s^2/2}(\text{erf}(s/\sqrt{2}) + 1)
\leq e^{s^2/2}2 \sqrt{2/\pi} \frac{s}{1 + \exp(-4s^2/\pi)} + 1
\leq e^{s^2/2}(\sqrt{2/\pi} s + 1).
\]

Thus, we have

\[
\Pr[Y \geq (1 + \epsilon)\alpha_1 k] \leq e^{-s(1+\epsilon)k} e^{ks^2/2} (1 + s\sqrt{2/\pi})^k
= e^{-s(1+\epsilon)k} e^{ks^2/2} e^{\log(1 + s\sqrt{2/\pi})}
\leq e^{-s(1+\epsilon)k + ks^2/2 + ks} \sqrt{2/\pi}
\leq e^{-\Omega(k^2)}.
\]

Thus, we have

\[
\Pr[\gamma \geq (1 + \epsilon)\alpha_1 k] \leq e^{-s(1+\epsilon)k} e^{ks^2/2} (1 + s\sqrt{2/\pi})^k
= e^{-s(1+\epsilon)k} e^{ks^2/2} e^{\log(1 + s\sqrt{2/\pi})}
\leq e^{-s(1+\epsilon)k + ks^2/2 + ks} \sqrt{2/\pi}
\leq e^{-\Omega(k^2)}.
\]

by \(\alpha_1 \geq \sqrt{2/\pi}\) and setting \(s = \epsilon\)

\[\blacksquare\]

**Lemma F.30.** For any \(\epsilon \in (0, 1)\), let \(k = O(n/\epsilon^2)\). Let \(G \in \mathbb{R}^{k \times n}\) denote i.i.d. random Gaussian matrices with rescaling. Then for any \(v \in \mathbb{R}^n\), with probability at least \(1 - 2^{-\Omega(n/\epsilon^2)}\), we have : for all \(v \in \mathbb{R}^n\),

\[
(1 - \epsilon)\|v\|_2 \leq \|Gv\|_1 \leq (1 + \epsilon)\|v\|_2.
\]

**Proof.** Let \(S\) denote \(\{y \in \mathbb{R}^n \mid \|y\|_2 = 1\}\). We construct a \(\gamma\)-net so that for all \(y \in S\), there exists a vector \(w \in \mathcal{N}\) for which \(\|y - w\|_2 \leq \gamma\). We set \(\gamma = 1/2\).

For any unit vector \(y\), we can write

\[
y = y^0 + y^1 + y^2 + \cdots,
\]

where \(\|y^i\|_2 \leq 1/2^i\) and \(y^i\) is a scalar multiple of a vector in \(\mathcal{N}\). Thus, we have

\[
\|Gy\|_1 = \|G(y^0 + y^1 + y^2 + \cdots)\|_1
\leq \sum_{i=0}^{\infty} \|Gy^i\|_1
\leq \sum_{i=0}^{\infty} (1 + \epsilon)\|y^i\|_2
\leq \sum_{i=0}^{\infty} (1 + \epsilon)\left(\frac{1}{2^i}\right)
\leq 1 + \Theta(\epsilon).
\]

Similarly, we can lower bound \(\|Gy\|_1\) by \(1 - \Theta(\epsilon)\). By Lemma 2.2 in [Woo14], we know that for any \(\gamma \in (0, 1)\), there exists a \(\gamma\)-net \(\mathcal{N}\) of \(S\) for which \(|\mathcal{N}| \leq (1 + 4/\gamma)^n\).

\[\blacksquare\]
F.3.3 Reduction, projection to high dimension

**Lemma F.31.** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $S \in \mathbb{R}^{n \times s}$ denote a Gaussian matrix with $s = O(n/\epsilon^2)$ columns. With probability at least $1 - 2^{-\Omega(n/\epsilon^2)}$, for any $U, V, W \in \mathbb{R}^{n \times k}$, we have

$$(1 - \epsilon) \|U \otimes V \otimes W - A\|_u \leq \|(U \otimes V \otimes W)S - AS\|_1 \leq (1 + \epsilon) \|U \otimes V \otimes W - A\|_u.$$

**Proof.** By definition of the $\otimes$ product between matrices and $\cdot$ product between a tensor and a matrix, we have $(U \otimes V \otimes W)S = U \otimes V \otimes (SW) \in \mathbb{R}^{n \times n \times s}$. We use $A_{i,j,*} \in \mathbb{R}^n$ to denote the $(i,j)$-th tube (the column in the 3rd dimension) of tensor $A$. We first prove the upper bound,

$$\|(U \otimes V \otimes W)S - AS\|_1 = \sum_{i=1}^n \sum_{j=1}^n \|((U \otimes V \otimes W)_{i,j,*} - A_{i,j,*})S\|_1 \leq \sum_{i=1}^n \sum_{j=1}^n (1 + \epsilon) \|((U \otimes V \otimes W)_{i,j,*} - A_{i,j,*})\|_2 = (1 + \epsilon) \|U \otimes V \otimes W - A\|_u,$$

where the first step follows by definition of tensor $\|\cdot\|_u$ norm, the second step follows by Lemma F.30, and the last step follows by tensor entry-wise $\ell_1$ norm. Similarly, we can prove the lower bound,

$$\|(U \otimes V \otimes W)S - AS\|_1 \geq \sum_{i=1}^n \sum_{j=1}^n (1 - \epsilon) \|((U \otimes V \otimes W)_{i,j,*} - A_{i,j,*})\|_2 = (1 - \epsilon) \|U \otimes V \otimes W - A\|_u.$$

This completes the proof.

**Corollary F.32.** For any $\alpha \geq 1$, if $U', V', W'$ satisfy

$$\|(U' \otimes V' \otimes W' - A)S\|_1 \leq \gamma \min_{\text{rank} - k A_k} \|(A_k - A)S\|_1,$$

then

$$\|U' \otimes V' \otimes W' - A\|_u \leq \frac{1 + \epsilon}{1 - \epsilon} \min_{\text{rank} - k A_k} \|A_k - A\|_u.$$

**Proof.** Let $\hat{U}, \hat{V}, \hat{W}$ denote the optimal solution to $\min_{\text{rank} - k A_k} \|(A_k - A)S\|_1$. Let $U^*, V^*, W^*$ denote the optimal solution to $\min_{\text{rank} - k A_k} \|A_k - A\|_u$. Then,

$$\|U' \otimes V' \otimes W' - A\|_u \leq \frac{1}{1 - \epsilon} \|(U' \otimes V' \otimes W' - A)S\|_1 \leq \gamma \frac{1}{1 - \epsilon} \|(\hat{U} \otimes \hat{V} \otimes \hat{W} - A)S\|_1 \leq \gamma \frac{1}{1 - \epsilon} \|U^* \otimes V^* \otimes W^* - A\|_u,$$

which completes the proof.
F.3.4 Existence results

**Theorem F.33 (Existence results).** Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$ and a matrix $S \in \mathbb{R}^{n \times \pi}$, let $\text{OPT}$ denote $\min_{\text{rank} - k \ A_k \in \mathbb{R}^{n \times n \times n}} \| (A_k - A) S \|_1$, let $\hat{A} = AS \in \mathbb{R}^{n \times n \times \pi}$. For any $k \geq 1$, there exist three matrices $S_1 \in \mathbb{R}^{n \pi \times s_1}$, $S_2 \in \mathbb{R}^{n \pi \times s_2}$, $S_3 \in \mathbb{R}^{n \times 3 s_3}$ such that

$$\min_{X_1 \in \mathbb{R}^{n \times 1 \times k}, X_2 \in \mathbb{R}^{n \times 2 \times k}, X_3 \in \mathbb{R}^{n \times 3 \times k}} \| (\hat{A}_1 S_1 X_1) \otimes (\hat{A}_2 S_2 X_2) \otimes (\hat{A}_3 S_3 X_3) - \hat{A} \|_1 \leq \alpha \text{OPT},$$

or equivalently,

$$\min_{X_1 \in \mathbb{R}^{n \times 1 \times k}, X_2 \in \mathbb{R}^{n \times 2 \times k}, X_3 \in \mathbb{R}^{n \times 3 \times k}} \| (\hat{A}_1 S_1 X_1) \otimes (\hat{A}_2 S_2 X_2) \otimes (A S_3 X_3) - \hat{A} \|_1 \leq \alpha \text{OPT},$$

holds with probability $99/100$.

(I). Using a dense Cauchy transform,

$s_1 = s_2 = s_3 = \tilde{O}(k), \alpha = \tilde{O}(k^{1.5}) \log^3 n$.

(II). Using a sparse Cauchy transform,

$s_1 = s_2 = s_3 = \tilde{O}(k^5), \alpha = \tilde{O}(k^{13.5}) \log^3 n$.

(III). Guessing Lewis weights,

$s_1 = s_2 = s_3 = \tilde{O}(k), \alpha = \tilde{O}(k^{1.5})$.

**Proof.** We use $\text{OPT}$ to denote the optimal cost,

$$\text{OPT} := \min_{\text{rank} - k \ A_k \in \mathbb{R}^{n \times n \times n}} \| (A_k - A) S \|_1.$$ 

We fix $V^* \in \mathbb{R}^{n \times k}$ and $W^* \in \mathbb{R}^{n \times k}$ to be the optimal solution to

$$\min_{U \in \mathbb{R}^{n \times k}} \| U \otimes V \otimes W - A S \|_1.$$

We define $Z_1 \in \mathbb{R}^{k \times n \pi}$ to be the matrix where the $i$-th row is the vectorization of $V_i^* \otimes (S W_i^*)$. We define tensor

$$\hat{A} = AS \in \mathbb{R}^{n \times n \times \pi}.$$

Then we also have $\hat{A} = A (I, I, S)$ according to the definition of the $\cdot$ product between a tensor and a matrix.

Let $\hat{A}_1 \in \mathbb{R}^{n \times n \pi}$ denote the matrix obtained by flattening tensor $\hat{A}$ along the first direction. We can consider the following optimization problem,

$$\min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 - \hat{A}_1 \|_1.$$

Choosing $S_1$ to be one of the following sketching matrices:

(I) a dense Cauchy transform,

(II) a sparse Cauchy transform,

(III) a sampling and rescaling diagonal matrix according to Lewis weights.

Let $\alpha_{s_1}$ denote the approximation ratio produced by the sketching matrix $S_1$. We use $S_1 \in \mathbb{R}^{n \pi \times s_1}$ to sketch on right of the above problem, and obtain the problem:

$$\min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 S_1 - \hat{A}_1 S_1 \|_1 = \min_{U \in \mathbb{R}^{n \times k}} \| U^i Z_1 S_1 - (\hat{A}_1 S_1)^i \|_1,$$

129.
where $U^i$ denotes the $i$-th row of matrix $U \in \mathbb{R}^{nxk}$ and $(\hat{A}_1S_1)^i$ denotes the $i$-th row of matrix $\hat{A}_1S_1$. Instead of solving it under $\ell_1$-norm, we consider the $\ell_2$-norm relaxation,

$$\min_{U \in \mathbb{R}^{nxk}} \|UZ_1S_1 - \hat{A}_1S_1\|_F^2 = \min_{U \in \mathbb{R}^{nxk}} \sum_{i=1}^{n} \|U^iZ_1S_1 - (\hat{A}_1S_1)^i\|_2^2.$$

Let $\hat{U} \in \mathbb{R}^{nxk}$ denote the optimal solution of the above optimization problem, so that $\hat{U} = \hat{A}_1S_1(Z_1S_1)^\dagger$. We plug $\hat{U}$ into the objective function under the $\ell_1$-norm. By the property of sketching matrix $S_1 \in \mathbb{R}^{n\pi x s_1}$, we have,

$$\|\hat{U}Z_1 - \hat{A}_1\|_1 \leq \alpha S_1 \min_{U \in \mathbb{R}^{nxk}} \|UZ_1 - \hat{A}_1\|_1 = \alpha S_1 \text{OPT},$$

which implies that,

$$\|\hat{U} \otimes V^* \otimes (SW^*) - \hat{A}_1\|_1 = \|((\hat{U} \otimes V^* \otimes W^*)S - \hat{A}_1\|_1 \leq \alpha S_1 \text{OPT}.$$

In the second step, we fix $\hat{U} \in \mathbb{R}^{nxk}$ and $W^* \in \mathbb{R}^{nxk}$. Let $\hat{A}_2 \in \mathbb{R}^{nxn\pi}$ denote the matrix obtained by flattening tensor $\hat{A} \in \mathbb{R}^{nxn\pi}$ along the second direction. We choose a sketching matrix $S_2 \in \mathbb{R}^{n\pi \times s_2}$. Let $Z_2 = \hat{U}^\dagger \otimes (SW^*)^\dagger \in \mathbb{R}^{k \times n\pi}$ denote the matrix where the $i$-th row is the vectorization of $\hat{U}_i \otimes (SW^*_i)$. Define $\hat{V} = A_2S_2(Z_2S_2)^\dagger$. By the properties of sketching matrix $S_2$, we have

$$\|\hat{V}Z_2 - \hat{A}_2\|_1 \leq \alpha S_2 \alpha S_1 \text{OPT},$$

In the third step, we fix $\hat{U} \in \mathbb{R}^{nxk}$ and $\hat{V} \in \mathbb{R}^{nxk}$. Let $\hat{A}_3 \in \mathbb{R}^{nxn^2}$ denote the matrix obtained by flattening tensor $\hat{A} \in \mathbb{R}^{nxn\pi}$ along the third direction. We choose a sketching matrix $S_3 \in \mathbb{R}^{n^2 \times s_3}$. Let $Z_3 \in \mathbb{R}^{k \times n^2}$ denote the matrix where the $i$-th row is the vectorization of $\hat{U}_i \otimes \hat{V}_i$. Define $W' = \hat{A}_3S_3(Z_3S_3)^\dagger \in \mathbb{R}^{nxk}$ and $\hat{W} = A_3S_3(Z_3S_3)^\dagger \in \mathbb{R}^{nxk}$. Then we have,

$$W' = \hat{A}_3S_3(Z_3S_3)^\dagger = (A(I, I, S))_3S_3(Z_3S_3)^\dagger = (S^T A_3)_3S_3(Z_3S_3)^\dagger = S^T \hat{W}.$$

By properties of sketching matrix $S_3$, we have

$$\|W'Z_3 - \hat{A}_3\|_1 \leq \alpha S_4 \alpha S_2 \alpha S_1 \text{OPT}.$$

Replacing $W'$ by $S^T \hat{W}$, we obtain,

$$\|W'Z_3 - \hat{A}_3\|_1 = \|S^T \hat{W}Z_3 - \hat{A}_3\|_1 = \|S^T \hat{W}Z_3 - S^T A_3\|_1 = \|(\hat{U} \otimes \hat{V} \otimes \hat{W} - A)S\|_1.$$

Thus, we have

$$\min_{X_1 \in \mathbb{R}^{k \times n}, X_2 \in \mathbb{R}^{k \times n}, X_3 \in \mathbb{R}^{k \times n}} \|\hat{A}_1S_1X_1 \otimes (\hat{A}_2S_2X_2) \otimes (\hat{A}_3S_3X_3) - \hat{A}\|_1 \leq \alpha S_1 \alpha S_2 \alpha S_3 \text{OPT}.$$
F.3.5 Running time analysis

Fact F.34. Given tensor $A \in \mathbb{R}^{n \times n \times n}$ and a matrix $B \in \mathbb{R}^{n \times d}$ with $d = O(n)$, let $AB$ denote an $n \times n \times d$ size tensor. For each $i \in [3]$, let $(AB)_i$ denote a matrix obtained by flattening tensor $AB$ along the $i$-th dimension, then

$$(AB)_1 \in \mathbb{R}^{n \times nd}, (AB)_2 \in \mathbb{R}^{n \times nd}, (AB)_3 \in \mathbb{R}^{d \times n^2}.$$  

For each $i \in [3]$, let $S_i \in \mathbb{R}^{n \times s_i}$ denote a sparse Cauchy transform, $T_i \in \mathbb{R}^{t_i \times n}$. Then we have,

(I) If $T_1$ denotes a sparse Cauchy transform or a sampling and rescaling matrix according to the Lewis weights, $T_1(AB)_1S_1$ can be computed in $O(nnz(A)d)$ time. Otherwise, it can be computed in $O(nnz(A)d + n s_1 t_1)$.

(II) If $T_2$ denotes a sparse Cauchy transform or a sampling and rescaling matrix according to the Lewis weights, $T_2(AB)_2S_2$ can be computed in $O(nnz(A)d)$ time. Otherwise, it can be computed in $O(nnz(A)d + n s_2 t_2)$.

(III) If $T_3$ denotes a sparse Cauchy transform or a sampling and rescaling matrix according to the Lewis weights, $T_3(AB)_3S_3$ can be computed in $O(nnz(A)d)$ time. Otherwise, it can be computed in $O(nnz(A)d + d s_3 t_3)$.

Proof. Part (I). Note that $T_1(AB)_1S_1 \in \mathbb{R}^{t_1 \times s_1}$ and $(AB)_1 \in \mathbb{R}^{n \times nd}$, for each $i \in [t_1], j \in [s_1],$

$$
(T_1(AB)_1S_1)_{i,j} = \sum_{x=1}^{n} \sum_{y=1}^{nd} (T_1)_{i,x} ((AB)_1)_{x,y} (S_1)_{y,j}
$$

$$
= \sum_{x=1}^{n} \sum_{y=1}^{n} \sum_{z=1}^{d} (T_1)_{i,x} ((AB)_1)_{x,(y-1)d+z} (S_1)_{(y-1)d+z,j}
$$

$$
= \sum_{x=1}^{n} \sum_{y=1}^{n} \sum_{z=1}^{d} (T_1)_{i,x} (AB)_{x,y,z} (S_1)_{(y-1)d+z,j}
$$

$$
= \sum_{x=1}^{n} \sum_{y=1}^{n} \sum_{z=1}^{d} (T_1)_{i,x} \sum_{u=1}^{d} (A_{x,y,w} B_{w,z})(S_1)_{(y-1)d+z,j}
$$

$$
= \sum_{x=1}^{n} \sum_{y=1}^{n} \sum_{z=1}^{d} (T_1)_{i,x} A_{x,y,w} \sum_{z=1}^{d} B_{w,z}(S_1)_{(y-1)d+z,j}.
$$

We look at a non-zero entry $A_{x,y,w}$ and the entry $B_{w,z}$. If $T_1$ denotes a sparse Cauchy transform or a sampling and rescaling matrix according to the Lewis weights, then there is at most one pair $(i, j)$ such that $(T_1)_{i,x} A_{x,y,w} B_{w,z} (S_1)_{(y-1)d+z,j}$ is non-zero. Therefore, computing $T_1(AB)_1S_1$ only needs $nnz(A)d$ time. If $T_1$ is not in the above case, since $S_1$ is sparse, we can compute $(AB)_1S_1$ in $nnz(A)d$ time by a similar argument. Then, we can compute $T_1(AB)_1S_1$ in $nt_1s_1$ time.

Part (II). It is as the same as Part (I).
Part (III). Note that \( T_3(AB)_3S_3 \in \mathbb{R}^{t_3 \times s_3} \) and \((AB)_3 \in \mathbb{R}^{d \times n^2} \). For each \( i \in [t_3], j \in [s_3] \),
\[
(T_3(AB)_3S_3)_{i,j} = \sum_{x=1}^{d} \sum_{y=1}^{n} (T_3)_{i,x} ((AB)_3x,y(S_3)y,j)
\]
\[
= \sum_{x=1}^{d} \sum_{y=1}^{n} \sum_{z=1}^{n} (T_3)_{i,x} ((AB)_3x,y,z(S_3)(y-1)n+z,j)
\]
\[
= \sum_{x=1}^{d} \sum_{y=1}^{n} \sum_{z=1}^{n} (T_3)_{i,x} (AB)y,z,x(S_3)(y-1)n+z,j
\]
\[
= \sum_{x=1}^{d} \sum_{y=1}^{n} \sum_{z=1}^{n} (T_3)_{i,x} \sum_{w=1}^{n} A_y,z,w B_{w,x}(S_3)(y-1)n+z,j
\]

Similar to Part (I), if \( T_1 \) denotes a sparse Cauchy transform or a sampling and rescaling matrix according to the Lewis weights, computing \( T_3(AB)_3S_3 \) only needs \( \text{nnz}(A)d \) time. Otherwise, it needs \( dt_3s_3 + \text{nnz}(A)d \) running time.

\[ \square \]

F.3.6 Algorithms

Algorithm 34 \( \ell_1-\ell_1-\ell_2 \)-Low Rank Approximation algorithm, input sparsity time

1: procedure L112TENSORLOWRANKAPPROXINPUTSPARSITY\((A,n,k)\) \> Theorem F.35
2: \( \pi \leftarrow O(n) \).
3: \( s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \tilde{O}(k^5) \).
4: Choose \( S \in \mathbb{R}^{n \times \pi} \) to be a Gaussian matrix.
5: Choose \( S_1 \in \mathbb{R}^{n \pi \times s_1} \) to be a sparse Cauchy transform. \> Part (II) of Theorem F.33
6: Choose \( S_2 \in \mathbb{R}^{n \pi \times s_2} \) to be a sparse Cauchy transform.
7: Choose \( S_3 \in \mathbb{R}^{n \pi \times s_3} \) to be a sparse Cauchy transform.
8: Form \( \hat{A} = AS \).
9: Compute \( \hat{A}_1S_1, \hat{A}_2S_2, \) and \( \hat{A}_3S_3 \).
10: \( Y_1, Y_2, Y_3, C \leftarrow \text{L1POLYKSIZEREDUCTION}(\hat{A}, \hat{A}_1S_1, \hat{A}_2S_2, \hat{A}_3S_3, n, n, \pi, s_1, s_2, s_3, k) \) \> Algorithm 21
11: Create \( s_1k + s_2k + s_3k \) variables for each entry of \( X_1, X_2, X_3 \).
12: Form objective function \( \| (Y_1X_1) \otimes (Y_2X_2) \otimes (Y_3X_3) - C \|_F^2 \).
13: Run polynomial system verifier.
14: return \( A_1S_1X_1, A_2S_2X_2, A_3S_3X_3 \)
15: end procedure

Theorem F.35. Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), there exists an algorithm which takes \( O(\text{nnz}(A)n) + \tilde{O}(n) \text{ poly}(k) + n2\tilde{O}(k^2) \) time and outputs three matrices \( U, V, W \in \mathbb{R}^{n \times k} \) such that,
\[
\|U \otimes V \otimes W - A\|_u \leq \text{poly}(k, \log n) \min_{\text{rank} - k A'} \| A' - A \|_u,
\]
holds with probability at least \( 9/10 \).
Proof. We first choose a Gaussian matrix \( S \in \mathbb{R}^{n \times \pi} \) with \( \pi = O(n) \). By applying Corollary \( \text{F.32} \), we can reduce the original problem to a “generalized” \( \ell_1 \) low rank approximation problem. Next, we use the existence results (Theorem \( \text{F.33} \)) and polynomial in \( k \) size reduction (Lemma \( \text{D.8} \)). At the end, we relax the \( \ell_1 \)-norm objective function to a Frobenius norm objective function (Fact \( \text{D.1} \)). \( \square \)

Algorithm 35 \( \ell_1-\ell_2\)-Low Rank Approximation Algorithm, \( \widetilde{O}(k^{2/3}) \)

1: procedure L112TENSORLOWRANKAPPROXK\((A,n,k)\) \( \triangleright \) Theorem \( \text{F.36} \)
2: \( \pi \leftarrow O(n) \).
3: \( s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow \widetilde{O}(k) \).
4: Choose \( S \in \mathbb{R}^{n \times \pi} \) to be a Gaussian matrix.
5: Guess a diagonal matrix \( S_1 \in \mathbb{R}^{n \pi \times s_1} \) with \( s_1 \) nonzero entries. \( \triangleright \) Part (III) of Theorem \( \text{F.33} \)
6: Guess a diagonal matrix \( S_2 \in \mathbb{R}^{n \pi \times s_2} \) with \( s_2 \) nonzero entries.
7: Guess a diagonal matrix \( S_3 \in \mathbb{R}^{n \pi \times s_3} \) with \( s_3 \) nonzero entries.
8: Form \( \hat{A} = \hat{A}S \).
9: Compute \( \hat{A}_1S_1, \hat{A}_2S_2, \) and \( \hat{A}_3S_3 \)
10: \( Y_1,Y_2,Y_3,C \leftarrow \text{L1POLYKSIZEREDUCTION}(\hat{A},\hat{A}_1S_1,\hat{A}_2S_2,\hat{A}_3S_3,n,n,\pi,s_1,s_2,s_3,k) \) \( \triangleright \) Algorithm 21
11: Create \( s_1k + s_2k + s_3k \) variables for each entry of \( X_1, X_2, X_3 \).
12: Form objective function \( \| (Y_1X_1) \otimes (Y_2X_2) \otimes (Y_3X_3) - C \|_1 \).
13: Run polynomial system verifier.
14: return \( A_1S_1X_1, A_2S_2X_2, A_3S_3X_3 \)
15: end procedure

Theorem \( \text{F.36} \). Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), there exists an algorithm which takes \( n^{\widetilde{O}(k)}2^{\widetilde{O}(k^3)} \) time and outputs three matrices \( U,V,W \in \mathbb{R}^{n \times k} \) such that,

\[
\| U \otimes V \otimes W - A \|_u \leq O(k^{3/2}) \min_{\text{rank}-k} A' \| A' - A \|_u,
\]

holds with probability at least \( 9/10 \).

Proof. We first choose a Gaussian matrix \( S \in \mathbb{R}^{n \times \pi} \) with \( \pi = O(n) \). By applying Corollary \( \text{F.32} \), we can reduce the original problem to a “generalized” \( \ell_1 \) low rank approximation problem. Next, we use the existence results (Theorem \( \text{F.33} \)) and polynomial in \( k \) size reduction (Lemma \( \text{D.8} \)). At the end, we solve an entry-wise \( \ell_1 \) norm objective function directly. \( \square \)

Theorem \( \text{F.37} \). Given a 3rd order tensor \( A \in \mathbb{R}^{n \times n \times n} \), for any \( k \geq 1 \), let \( r = \widetilde{O}(k^2) \). There is an algorithm which takes \( O(\text{nnz}(A)n) + \widetilde{O}(n) \text{ poly}(k) \) time and outputs three matrices \( U,V,W \in \mathbb{R}^{n \times r} \) such that

\[
\| U \otimes V \otimes W - A \|_u \leq \text{poly}(\log n,k) \min_{\text{rank}-k} \| A_k - A \|_u,
\]

holds with probability at least \( 9/10 \).

Proof. We first choose a Gaussian matrix \( S \in \mathbb{R}^{n \times \pi} \) with \( \pi = O(n) \). By applying Corollary \( \text{F.32} \), we can reduce the original problem to a “generalized” \( \ell_1 \) low rank approximation problem. Next, we use the existence results (Theorem \( \text{F.33} \)) and polynomial in \( k \) size reduction (Lemma \( \text{D.8} \)). At the end, we solve an entry-wise \( \ell_1 \) norm objective function directly. \( \square \)
Algorithm 36 $\ell_1$-$\ell_1$-$\ell_2$-Low Rank Approximation Algorithm, Bicriteria Algorithm

1: procedure L112TensorLowRankApproxBicriteria($A, n, k$) \hfill \triangleright \text{Theorem F.37}
2: \quad $\pi \leftarrow O(n)$. 
3: \quad $s_2 \leftarrow s_3 \leftarrow \widetilde{O}(k^5)$. 
4: \quad $t_2 \leftarrow t_3 \leftarrow \widetilde{O}(k)$. 
5: \quad $r \leftarrow s_2 s_3$.
6: \quad Choose $S \in \mathbb{R}^{n \times \pi}$ to be a Gaussian matrix.
7: \quad Form $\hat{A} = AS \in \mathbb{R}^{n \times n \times \pi}$. 
8: \quad Choose a sketching matrix $S_2 \in \mathbb{R}^{n \times s_2}$ with $s_2$ nonzero entries (Sparse Cauchy transform), for each $i \in \{2, 3\}$. \hfill \triangleright \text{Part (II) of Theorem F.33}
9: \quad Choose a sampling and rescaling diagonal matrix $D_i$ according to the Lewis weights of $\hat{A}_iS_i$ with $t_i$ nonzero entries, for each $i \in \{2, 3\}$. 
10: \quad Form $\hat{V} \in \mathbb{R}^{n \times r}$ by setting the $(i, j)$-th column to be $(\hat{A}_2S_2)_i$. 
11: \quad Form $\hat{W} \in \mathbb{R}^{n \times r}$ by setting the $(i, j)$-th column to be $(\hat{A}_3S_3)_j$. 
12: \quad Form matrix $B \in \mathbb{R}^{r \times t_2 t_3}$ by setting the $(i, j)$-th column to be the vectorization of $(T_2\hat{A}_2S_2)_i \otimes (T_3\hat{A}_3S_3)_j$.
13: \quad Solve $\min_{U} \|U \cdot B - (\hat{A}(I, T_2, T_3))_1\|_1$. 
14: \quad return $\hat{U}, \hat{V}, \hat{W}$.
15: end procedure

G Weighted Frobenius Norm for Arbitrary Tensors

This section presents several tensor algorithms for the weighted case. For notational purposes, instead of using $U, V, W$ to denote the ground truth factorization, we use $U_1, U_2, U_3$ to denote the ground truth factorization. We use $A$ to denote the input tensor, and $W$ to denote the tensor of weights. Combining our new tensor techniques with existing weighted low rank approximation algorithms [RSW16] allows us to obtain several interesting new results. We provide some necessary definitions and facts in Section G.1. Section G.2 provides an algorithm when $W$ has at most $r$ distinct faces in each dimension. Section G.3 studies relationships between $r$ distinct faces and $r$ distinct columns. Finally, we provides an algorithm with a similar running time but weaker assumption, where $W$ has at most $r$ distinct columns and $r$ distinct rows in Section G.4. The result in Theorem G.2 is fairly similar to Theorem G.5, except for the running time. We only put a very detailed discussion in the statement of Theorem G.5. Note that Theorem G.2 also has other versions which are similar to the Frobenius norm rank-$k$ algorithms described in Section 1. For simplicity of presentation, we only present one clean and simple version (which assumes $A_k$ exists and has factor norms which are not too large).

G.1 Definitions and Facts

For a matrix $A \in \mathbb{R}^{n \times m}$ and a weight matrix $W \in \mathbb{R}^{n \times m}$, we define $\|W \circ A\|_F$ as follows,

$$\|W \circ A\|_F = \left( \sum_{i=1}^{n} \sum_{j=1}^{m} W_{i,j}^2 A_{i,j}^2 \right)^{\frac{1}{2}}.$$
For a tensor $A \in \mathbb{R}^{n \times n \times n}$ and a weight tensor $W \in \mathbb{R}^{n \times n \times n}$, we define $\|W \circ A\|_F$ as follows,
\[
\|W \circ A\|_F = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{n} W_{i,j,l}^2 A_{i,j,l}^2 \right)^{\frac{1}{2}}.
\]

For three matrices $A \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times m}$ and a weight matrix $W$, from one perspective, we have
\[
\|(UV - A) \circ W\|_F^2 = \sum_{i=1}^{n} \|(U^i V - A^i) \circ W^i\|_2^2 = \sum_{i=1}^{n} \|(U^i V - A^i) D_{W^i}\|_2^2,
\]
where $W^i$ denote the $i$-th row of matrix $W$, and $D_{W^i} \in \mathbb{R}^{m \times m}$ denotes a diagonal matrix where the $j$-th entry on diagonal is the $j$-th entry of vector $W^i$. From another perspective, we have
\[
\|(UV - A) \circ W\|_F^2 = \sum_{j=1}^{m} \|(U V_j - A_j) \circ W_j\|_2^2 = \sum_{j=1}^{m} \|(U V_j - A_j) D_{W_j}\|_2^2,
\]
where $W_j$ denotes the $j$-th column of matrix $W$, and $D_{W_j} \in \mathbb{R}^{n \times n}$ denotes a diagonal matrix where the $i$-th entry on the diagonal is the $i$-th entry of vector $W_j$.

One of the key tools we use in this section is,

**Lemma G.1 (Cramer’s rule).** Let $R$ be an $n \times n$ invertible matrix. Then, for each $i \in [n], j \in [n],$
\[
(R^{-1})_{i,j} = \det(R_{-,j}^{-1}) / \det(R),
\]
where $R_{-,j}$ is the matrix $R$ with the $i$-th row and the $j$-th column removed.

### G.2 $r$ distinct faces in each dimension

Notice that in the matrix case, it is sufficient to assume that $\|A'\|_F$ is upper bounded [RSW16]. Once we have that $\|A'\|_F$ is bounded, without loss of generality, we can assume that $U_1^*$ is an orthonormal basis[CW15a, RSW16]. If $U_1^*$ is not an orthonormal basis, then let $U_1^* R$ denote a QR factorization of $U_1^*$, and then write $U_3^* = R U_3^*$. However, in the case of tensors we have to assume that each factor $\|U_i^*\|_F$ is upper bounded due to border rank issues (see, e.g., [DSL08]).

**Theorem G.2.** Given a 3rd order $n \times n \times n$ tensor $A$ and an $n \times n \times n$ tensor $W$ of weights with $r$ distinct faces in each of the three dimensions for which each entry can be written using $O(n^\delta)$ bits, for $\delta > 0$, define $\text{OPT} = \inf_{\text{rank-}k} A_k \|W \circ (A_k - A)\|_F^2$. Let $k \geq 1$ be an integer and let $0 < \epsilon < 1$.

If $\text{OPT} > 0$, and there exists a rank-$k$ $A_k = U_1^* \otimes U_2^* \otimes U_3^*$ tensor (with size $n \times n \times n$) such that $\|W \circ (A_k - A)\|_F^2 = \text{OPT}$, and $\max_{i \in [3]} \|U_i^*\|_F \leq 2^{O(n^\delta)}$, then there exists an algorithm that takes $(\text{nnz}(A) + \text{nnz}(W) + n^{O(k^2 / \epsilon)}) n^{O(\delta)}$ time in the unit cost RAM model with words of size $O(\log n)$ bits\footnote{The entries of $A$ and $W$ are assumed to fit in $n^\delta$ words.} and outputs three $n \times k$ matrices $U_1, U_2, U_3$ such that
\[
\|W \circ (U_1 \otimes U_2 \otimes U_3 - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}
\]
holds with probability $9/10$.\footnote{The entries of $A$ and $W$ are assumed to fit in $n^\delta$ words.}
Algorithm 37 Weighted Tensor Low-rank Approximation Algorithm when the Weighted Tensor has $r$ Distinct Faces in Each of the Three Dimensions.

```
procedure WEIGHTEDRDISTINCTFACESIN3DIMENSIONS($A, W, n, r, k, \epsilon$)  \triangleright Theorem G.2

for $j = 1 \rightarrow 3$
do
  $s_j \leftarrow O(k/\epsilon)$.
  Choose a sketching matrix $S_j \in \mathbb{R}^{n^2 \times s_j}$.
  for $i = 1 \rightarrow r$
do
    Create $k \times s_1$ variables for matrix $P_{i,j} \in \mathbb{R}^{k \times s_j}$.
  end for

  for $i = 1 \rightarrow n$
do
    Write down $(\tilde{U}_j)^i = A_j^\dag D W_i^j S_j P_i^{j\dag}(P_i^{j\dag}P_i^{j\dag})^{-1}$.
  end for

Form $\|W \circ (\tilde{U}_1 \otimes \tilde{U}_2 \otimes \tilde{U}_3 - A)\|_F^2$.
Run polynomial system verifier.
return $U_1, U_2, U_3$
end procedure
```

Proof. Note that $W$ has $r$ distinct columns, rows, and tubes. Hence, each of the matrices $W_1, W_2, W_3 \in \mathbb{R}^{n \times n^2}$ has at most $r$ distinct columns, and at most $r$ distinct rows. Let $U_1^r, U_2^r, U_3^r \in \mathbb{R}^{n \times k}$ denote the matrices satisfying $\|W \circ (U_1^r \otimes U_2^r \otimes U_3^r - A)\|_F^2 = \text{OPT}$. We fix $U_2^r$ and $U_3^r$, and consider a flattening of the tensor along the first dimension,

$$
\min_{U_1^r \in \mathbb{R}^{n \times k}} \|(U_1 Z_1 - A_1) \circ W_1\|_F^2 = \text{OPT},
$$

where matrix $Z_1 = U_2^{r\dag} \otimes U_3^{r\dag}$ has size $k \times n^2$ and for each $i \in [k]$ the $i$-th row of $Z_1$ is $\text{vec}((U_2^r)_i \otimes (U_3^r)_i)$. For each $i \in [n]$, let $W_1^i$ denote the $i$-th row of $n \times n^2$ matrix $W_1$. For each $i \in [n]$, let $D_{W_1^i}$ denote the diagonal matrix of size $n^2 \times n^2$, where each diagonal entry is from the vector $W_1^i \in \mathbb{R}^{n^2}$. Without loss of generality, we can assume the first $r$ rows of $W_1$ are distinct. We can rewrite the objective function along the first dimension as a sum of multiple regression problems. For any $n \times k$ matrix $U_1$,

$$
\|(U_1 Z_1 - A_1) \circ W_1\|_F^2 = \sum_{i=1}^{n} \|U_1^i Z_1 D_{W_1^i} - A_1^i D_{W_1^i}\|_2^2. \quad (72)
$$

Based on the observation that $W_1$ has $r$ distinct rows, we can group the $n$ rows of $W_1$ into $r$ groups. We use $g_1, g_1, \cdots, g_r$ to denote $r$ sets of indices such that, for each $i \in g_{1,j}$, $W_1^i = W_1^{j\dag}$. Thus we can rewrite Equation (72),

$$
\|(U_1 Z_1 - A_1) \circ W_1\|_F^2 = \sum_{i=1}^{n} \|U_1^i Z_1 D_{W_1^i} - A_1^i D_{W_1^i}\|_2^2 \\
= \sum_{j=1}^{r} \sum_{i \in g_{1,j}} \|U_1^i Z_1 D_{W_1^i} - A_1^i D_{W_1^i}\|_2^2.
$$

We can sketch the objective function by choosing Gaussian matrices $S_1 \in \mathbb{R}^{n^2 \times s_1}$ with $s_1 = O(k/\epsilon)$.

$$
\sum_{i=1}^{n} \|U_1^i Z_1 D_{W_1^i} S_1 - A_1^i D_{W_1^i} S_1\|_2^2.
$$
Let \( \hat{U}_1 \) denote the optimal solution of the sketch problem,

\[
\hat{U}_1 = \arg\min_{U_1 \in \mathbb{R}^{n \times k}} \sum_{i=1}^{n} \|U_1 Z_1 D_{W_1} S_1 - A_i^t D_{W_i} S_1\|^2_2.
\]

By properties of \( S_1([RSW16]) \), plugging \( \hat{U} \in \mathbb{R}^{n \times k} \) into the original problem, we obtain,

\[
\sum_{i=1}^{n} \|\hat{U}_i Z_1 D_{W_i} - A_i D_{W_i}\|^2_2 \leq (1 + \epsilon) \text{OPT}.
\]

Note that \( \hat{U}_1 \in \mathbb{R}^{n \times k} \) also has the following form. For each \( i \in [n] \),

\[
\hat{U}_i^i = A_i^t D_{W_i} S_1 (Z_1 D_{W_i} S_1)^\dagger = A_i^t D_{W_i} S_1 (Z_1 D_{W_i} S_1)^\top ((Z_1 D_{W_i} S_1)(Z_1 D_{W_i} S_1)^\top)^{-1}.
\]

Note that \( W_1 \) has \( r \) distinct rows. Thus, we only have \( r \) distinct \( D_{W_i} \). This implies that there are \( r \) distinct matrices \( Z_1 D_{W_i} S_1 \in \mathbb{R}^{k \times s_1} \). Using the definition of \( g_{i,j} \), for \( j \in [r] \), for each \( i \in g_{i,j} \subset [n] \), we have

\[
\hat{U}_i \hat{U} = A_i^t D_{W_i} S_1 (Z_1 D_{W_i} S_1)^\dagger = A_i^t D_{W_i} S_1 (Z_1 D_{W_i} S_1)^\top \text{ by } W_i^i = W_j^j,
\]

which means we only need to write down \( r \) different \( Z_1 D_{W_i} S_1 \). For each \( k \times s_1 \) matrix \( Z_1 D_{W_i} S_1 \), we create \( k \times s_1 \) variables to represent it. Thus, we need to create \( r k s_1 \) variables to represent \( r \) matrices,

\[
\{Z_1 D_{W_i} S_1, Z_1 D_{W_2} S_1, \ldots, Z_1 D_{W_r} S_1\}.
\]

For simplicity, let \( P_{i,i} \in \mathbb{R}^{k \times s_1} \) denote \( Z_1 D_{W_i} S_1 \). Then we can rewrite \( \hat{U}_i \in \mathbb{R}^k \) as follows,

\[
\hat{U}_i = A_i^t D_{W_i} S_1 P_{i,i}^t (P_{i,i} P_{i,i}^t)^{-1}.
\]

If \( P_{i,i} P_{i,i}^t \in \mathbb{R}^{k \times k} \) has rank \( k \), then we can use Cramer’s rule (Lemma G.1) to write down the inverse of \( P_{i,i} P_{i,i}^t \). However, vector \( W_i^i \) could have many zero entries. Then the rank of \( P_{i,i} P_{i,i}^t \) can be smaller than \( k \). There are two different ways to solve this issue.

One way is by using the argument from [RSW16], which allows us to assume that \( P_{i,i} P_{i,i}^t \in \mathbb{R}^{k \times k} \) has rank \( k \).

The other way is straightforward: we can guess the rank. There are \( k \) possibilities. Let \( t_i \leq k \) denote the rank of \( P_{i,i} \). Then we need to figure out a maximal linearly independent subset of rows of \( P_{i,i} \). There are \( 2^{O(k)} \) possibilities. Next, we need to figure out a maximal linearly independent subset of columns of \( P_{i,i} \). We can also guess all the possibilities, which is at most \( 2^{O(k)} \). Because we have \( r \) different \( P_{i,i} \), the total number of guesses we have is at most \( 2^{O(rk)} \). Thus, we can write down \( (P_{i,i} P_{i,i}^t)^{-1} \) according to Cramer’s rule.

After \( \hat{U}_1 \) is obtained, we will fix \( \hat{U}_1 \) and \( U_3 \) in the next round. We consider the flattening of the tensor along the second direction,

\[
\min_{U_2 \in \mathbb{R}^{n \times k}} \| (U_2 Z_2 - A_2) \circ W_2 \|^2_F,
\]
where \( n \times n^2 \) matrix \( A_2 \) is obtained by flattening tensor \( A \) along the second dimension, \( k \times n^2 \) matrix \( Z_2 \) denotes \( U_1^\top \odot U_3^\top \), and \( n \times n^2 \) matrix \( W_2 \) is obtained by flattening tensor \( W \) along the second dimension. For each \( i \in [n] \), let \( W_2^i \) denote the \( i \)-th row of \( n \times n^2 \) matrix \( W_2 \). For each \( i \in [n] \), let \( D_{W_2^i} \) denote the diagonal matrix which has size \( n^2 \times n^2 \) and for which each entry is from vector \( W_2^i \in \mathbb{R}^{n^2} \). Without loss of generality, we can assume the first \( r \) rows of \( W_2 \) are distinct. We can rewrite the objective function along the second dimension as a sum of multiple regression problems. For any \( n \times k \) matrix \( U_2 \),

\[
\|(U_2 Z_2 - A_2) \circ W_2\|_F^2 = \sum_{i=1}^{n} \|U_2^i Z_2 D_{W_2^i} - A_2^i D_{W_2^i}\|^2_2.
\]  

(73)

Based on the observation that \( W_2 \) has \( r \) distinct rows, we can group the \( n \) rows of \( W_2 \) into \( r \) groups. We use \( g_{2,1}, g_{2,2}, \ldots, g_{2,r} \) to denote \( r \) sets of indices such that, for each \( i \in g_{2,j} \), \( W_2^i = W_2^j \). Thus we obtain,

\[
\|(U_2 Z_2 - A_2) \circ W_2\|_F^2 = \sum_{i=1}^{n} \|U_2^i Z_2 D_{W_2^i} - A_2^i D_{W_2^i}\|^2_2
\]

\[= \sum_{j=1}^{r} \sum_{i \in g_{2,j}} \|U_2^i Z_2 D_{W_2^i} - A_2^i D_{W_2^i}\|^2_2.
\]

We can sketch the objective function by choosing a Gaussian sketch \( S_2 \in \mathbb{R}^{n^2 \times s_2} \) with \( s_2 = O(k/\epsilon) \). Let \( \hat{U}_2 \) denote the optimal solution to the sketch problem. Then \( \hat{U}_2 \) has the form, for each \( i \in [n] \),

\[
\hat{U}_2^i = A_2^i D_{W_2^i} S_2 (Z_2 D_{W_2^i} S_2)^\dagger.
\]

Similarly as before, we only need to write down \( r \) different matrices \( Z_2 D_{W_2^i} S_1 \), and for each of them, create \( k \times s_2 \) variables. Let \( P_{2,i} \in \mathbb{R}^{k \times s_2} \) denote \( Z_2 D_{W_2^i} S_2 \). By our guessing argument, we can obtain \( \hat{U}_2 \).

In the last round, we fix \( \hat{U}_1 \) and \( \hat{U}_2 \). We then write down \( \hat{U}_3 \). Overall, by creating \( l = O(rk^2/\epsilon) \) variables, we have rational polynomials \( \hat{U}_1(x), \hat{U}_2(x), \hat{U}_3(x) \). Putting it all together, we can write this objective function,

\[
\min_{x \in \mathbb{R}^l} \|((\hat{U}_1(x) \otimes \hat{U}_2(x) \otimes \hat{U}_3(x) - A) \circ W\|_F^2.
\]

s.t. \( h_{1,i}(x) \neq 0, \forall i \in [r] \).
\( h_{2,i}(x) \neq 0, \forall i \in [r] \).
\( h_{3,i}(x) \neq 0, \forall i \in [r] \).

where \( h_{1,i}(x) \) denotes the denominator polynomial related to a full rank sub-block of \( P_{1,i}(x) \). By a perturbation argument in Section 4 in [RSW16], we know that the \( h_{1,i}(x) \) are nonzero. By a similar argument as in Section 5 in [RSW16], we can show a lower bound on the cost of the denominator polynomial \( h_{1,i}(x) \). Thus we can create new bounded variables \( x_{l+1}, \ldots, x_{3r+1} \) to rewrite the objective function,
\[
\min_{x \in \mathbb{R}^{l+3r}} \frac{q(x)}{p(x)}.
\]
\[
\text{s.t. } h_{1,i}(x)x_{l+i} = 0, \forall i \in [r].
\]
\[
h_{2,i}(x)x_{l+r+i} = 0, \forall i \in [r].
\]
\[
h_{3,i}(x)x_{l+2r+i} = 0, \forall i \in [r].
\]
\[
p(x) = \prod_{i=1}^{r} h_{1,i}^2(x)h_{2,i}^2(x)h_{3,i}^2(x)
\]

Note that the degree of the above system is \(\text{poly}(kr)\) and all the equality constraints can be merged into one single constraint. Thus, the number of constraints is \(O(1)\). The number of variables is \(O(rk^2/\epsilon)\).

Using Theorem B.11 and a similar argument from Section 5 of [RSW16], we have that the minimum nonzero cost is at least \(2^{-n^2 \tilde{O}(rk^2/\epsilon)}\). Combining the binary search explained in Section C (similar techniques also can be found in Section 6 of [RSW16]) with the lower bound we obtained, we can find the solution for the original problem in time,
\[
(nnz(A) + nnz(W) + n2^{\tilde{O}(rk^2/\epsilon)})n^{O(\delta)}.
\]

G.3 \(r\) distinct columns, rows and tubes

Lemma G.3. Let \(W \in \mathbb{R}^{n\times n\times n}\) denote a tensor that has \(r\) distinct columns and \(r\) distinct rows, then \(W\) has

(I) \(r\) distinct column-tube faces.

(II) \(r\) distinct row-tube faces.

Proof. Proof of Part (I). Without loss of generality, we consider the first (which is the bottom one) column-row face. Assume it has \(r\) distinct rows and \(r\) distinct columns. We can re-order all the column-tube faces to make sure that all the \(n\) columns in the bottom face have been split into \(r\) continuous disjoint groups \(C_i\), e.g., \(\{C_1, C_2, \cdots, C_r\} = [n]\). Next, we can re-order all the row-tube faces to make sure that all the \(n\) rows in the bottom face have been split into \(r\) continuous disjoint groups \(R_i\), e.g., \(\{R_1, R_2, \cdots, R_r\} = [n]\). Thus, the new bottom face can be regarded as \(r \times r\) groups, and the number in each position of the same group is the same.

Suppose that the tensor has \(r + 1\) distinct column-tube faces. By the pigeonhole principle there exist two different column-tube faces belonging to the same group \(C_i\), for some \(i \in [r]\). Note that these two column-tube faces are the same by looking at the bottom (column-row) face. Since they are distinct faces, there must exist one row vector \(v\) which is not in the bottom (column-row) face, and it has a different value in coordinates belong to group \(C_i\). Note that, considering the bottom face, for each row vector, it has the same value over coordinates belonging to group \(C_i\). But \(v\) has different values in coordinates belong to group \(C_i\). Also, note that the bottom (column-row) face also has \(r\) distinct rows, and \(v\) is not one of them. This means there are at least \(r + 1\) distinct rows, which contradicts that there are \(r\) distinct rows in total. Thus, there are at most \(r\) distinct column-tube faces.

Proof of Part (II). It is similar to Part (I).
Figure 7: Let $W$ denote a tensor that has columns(red), rows(green) and tubes(blue). For each $i \in [3]$, let $W_i$ denote the matrix obtained by flattening tensor $W$ along the $i$-th dimension.

Figure 8: Each face $W_{*,*,i}$ is a column-row face. $W_{*,*,1}$ is the bottom column-row face. $r = 3$. The blue blocks represent column-tube faces, the red blocks represent column-tube faces.

**Corollary G.4.** Let $W \in \mathbb{R}^{n \times n \times n}$ denote a tensor that has $r$ distinct columns, $r$ distinct rows, and $r$ distinct tubes. Then $W$ has $r$ distinct column-tube faces, $r$ distinct row-tube faces, and $r$ distinct column-row faces.

**Proof.** This follows by applying Lemma G.3 twice. \hfill $\square$
Thus, we obtain the same result as in Theorem G.2 by changing the assumption from \( r \) distinct faces in each dimension to \( r \) distinct columns, \( r \) distinct rows and \( r \) distinct tubes.

### G.4 \( r \) distinct columns and rows

The main difference between Theorem G.2 and Theorem G.5 is the running time. The first one takes \( 2^{O(r^2/k^2/\epsilon)} \) time and the second one is slightly longer, \( 2^{O(r^2k^2/\epsilon)} \). By Lemma G.3, \( r \) distinct columns in two dimensions implies \( r \) distinct faces in two of the three kinds of faces. Thus, the following theorem also holds for \( r \) distinct columns in two dimensions.

**Algorithm 38** Weighted Tensor Low-rank Approximation Algorithm when the Weighted Tensor has \( r \) Distinct Faces in Each of the Two Dimensions.

```plaintext
procedure WEIGHTEDRDISTINCTFACESIN2DIMENSIONS(A, W, n, r, k, \( \epsilon \)) ⊢ Theorem G.5
  for \( j = 1 \rightarrow 3 \) do
    \( s_j \leftarrow O(k/\epsilon) \).
    Choose a sketching matrix \( S_j \in \mathbb{R}^{n^2 \times s_j} \).
    if \( j \neq 3 \) then
      for \( i = 1 \rightarrow r \) do
        Create \( k \times s_1 \) variables for matrix \( P_{i,j} \in \mathbb{R}^{k \times s_j} \).
      end for
    end if
    for \( i = 1 \rightarrow n \) do
      Write down \( (\hat{U}_j)^i = A_{i,j}D_{W_{i,j}}S_jP_{j,i}^T(P_{j,i}P_{j,i}^T)^{-1} \).
    end for
  end for
  Form \( \| W \circ (\hat{U}_1 \otimes \hat{U}_2 \otimes \hat{U}_3 - A) \|_F^2 \).
  Run polynomial system verifier.
  return \( \hat{U}_1, \hat{U}_2, \hat{U}_3 \)
end procedure
```

**Theorem G.5.** Given a 3rd order \( n \times n \times n \) tensor \( A \) and an \( n \times n \times n \) tensor \( W \) of weights with \( r \) distinct faces in each dimension (out of three dimensions) such that each entry can be written using \( O(n^\delta) \) bits for some \( \delta > 0 \), define \( \text{OPT} = \inf_{\text{rank} - k} \| W \circ (A_k - A) \|_F^2 \). For any \( k \geq 1 \) and any \( 0 < \epsilon < 1 \).

(I) If \( \text{OPT} > 0 \), and there exists a rank-\( k \) \( A_k = U_1^k \otimes U_2^k \otimes U_3^k \) tensor (with size \( n \times n \times n \)) such that \( \| W \circ (A_k - A) \|_F^2 = \text{OPT} \), and \( \max_{i \in [3]} \| U_i^k \|_F \leq 2^{O(n^\delta)} \), then there exists an algorithm that takes \( (mnz(A) + mnz(W)) + n2^{O(r^2k^2/\epsilon)}n^{O(\delta)} \) time in the unit cost RAM model with words of size \( O(\log n) \) bits\(^{11}\) and outputs three \( n \times k \) matrices \( U_1, U_2, U_3 \) such that

\[
\| W \circ (U_1 \otimes U_2 \otimes U_3 - A) \|_F^2 \leq (1 + \epsilon) \text{OPT}
\]  

holds with probability \( 9/10 \).

(II) If \( \text{OPT} > 0 \), \( A_k \) does not exist, and there exist three \( n \times k \) matrices \( U_1', U_2', U_3' \) where each entry can be written using \( O(n^\delta) \) bits and \( \| W \circ (U_1' \otimes U_2' \otimes U_3' - A) \|_F^2 \leq (1 + \epsilon/2) \text{OPT} \), then we can find \( U, V, W \) such that (74) holds.

\(^{11}\)The entries of \( A \) and \( W \) are assumed to fit in \( n^\delta \) words.
(III) If $OPT = 0$, $A_k$ exists, and there exists a solution $U_1^*, U_2^*, U_3^*$ such that each entry of the matrix can be written using $O(n^\delta)$ bits, then we can obtain (74).

(IV) If $OPT = 0$, and there exist three $n \times k$ matrices $U_1, U_2, U_3$ such that $\max_{i \in [3]} \|U_i^*\|_F \leq 2^{O(n^\delta)}$ and

$$\|W \circ (U_1 \otimes U_2 \otimes U_3 - A)\|_F^2 \leq (1 + \epsilon) \text{OPT} + 2^{-\Omega(n^\delta)},$$

then we can output $U_1, U_2, U_3$ such that (75) holds.

(V) Further if $A_k$ exists, we can output a number $Z$ for which $\text{OPT} \leq Z \leq (1 + \epsilon) \text{OPT}$. For all the cases, the algorithm succeeds with probability at least $9/10$.

Proof. By Lemma G.3, we have $W$ has $r$ distinct column-tube faces and $r$ distinct row-tube faces. By Claim G.7, we know that $W$ has $R = 2^{O(r \log r)}$ distinct column-row faces.

We use the same approach as in the proof of Theorem G.2 (which is also similar to Section 8 of [RSW16]) to create variables, write down the polynomial systems and add not equal constraints. Instead of having $3r$ distinct denominators as in the proof of Theorem G.2, we have $2r + R$.

We create $l = O(\frac{rk^2}{\epsilon})$ variables for $\{Z_1D_{W_1}S_1, Z_1D_{W_2}S_1, \ldots, Z_1D_{W_3}S_1\}$. Then we can write down $\hat{U}_1$ with $r$ distinct denominators $g_i(x)$. Each $g_i(x)$ is non-zero in an optimal solution using the perturbation argument in Section 4 in [RSW16]. We create new variables $x_{2l+i}$ to remove the denominators $g_i(x), \forall i \in [r]$. Then the entries of $\hat{U}_1$ are polynomials as opposed to rational functions.

We create $l = O(\frac{rk^2}{\epsilon})$ variables for $\{Z_2D_{W_1}S_2, Z_2D_{W_2}S_2, \ldots, Z_2D_{W_3}S_2\}$. Then we can write down $\hat{U}_2$ with $r$ distinct denominators $g_{r+i}(x)$. Each $g_{r+i}(x)$ is non-zero in an optimal solution using the perturbation argument in Section 4 in [RSW16]. We create new variables $x_{2l+r+i}$ to remove the denominators $g_{r+i}(x), \forall i \in [r]$. Then the entries of $\hat{U}_2$ are polynomials as opposed to rational functions.

Using $\hat{U}_1$ and $\hat{U}_2$ we can express $\hat{U}_3$ with $R$ distinct denominators $f_j(x)$, which are also non-zero by using the perturbation argument in Section 4 in [RSW16], and using that $W_3$ has at most this number of distinct rows. Finally we can write the following optimization problem,

$$\min_{x \in \mathbb{R}^{2l+2r}} \frac{p(x)}{q(x)} \quad \text{s.t.} \quad g_i(x)x_{2l+i} - 1 = 0, \forall i \in [r]$$
$$g_{r+i}(x)x_{2l+r+i} - 1 = 0, \forall i \in [r]$$
$$f_j^2(x) \neq 0, \forall j \in [R]$$
$$q(x) = \prod_{j=1}^{R} f_j^2(x)$$

We then determine if there exists a solution to the above semi-algebraic set in time

$$(\text{poly}(k, r)R)^{O(\frac{rk^2}{\epsilon})} = 2^{\tilde{O}(\frac{rk^2}{\epsilon})}.$$ 

Using similar techniques from Section 5 of [RSW16], we can show a lower bound on the cost similar to Section 8.3 of [RSW16], namely, the minimum nonzero cost is at least

$$2^{-\frac{n}{\delta}2^{\tilde{O}(\frac{kr^2}{\epsilon})}}.$$
Figure 9: Each face $W_{s,s,i}$ is a column-row face. $W_{s,s,1}$ is the bottom column-row face. $r = 3$. The blue blocks represent $|C_3|$ column-tube faces. The green blocks represent $|R_3|$ row-tube faces. In each column-row face, the intersection between blue faces and green faces is a size $|R_3| \times |C_3|$ block, and all the entries in this block are the same.

Combining the binary search explained in Section C (a similar techniques also can be found in Section 6 of [RSW16]) with the lower bound we obtained, we can find a solution for the original problem in time

$$(\text{nnz}(A) + \text{nnz}(W) + n2^{O(r^2k^2/\epsilon)})n^{O(\delta)}.$$

Remark G.6. Note that the running time for the Frobenius norm and for the $\ell_1$ norm are of the form $\text{poly}(n) + \exp(\text{poly}(k/\epsilon))$ rather than $\text{poly}(n) \cdot \exp(k/\epsilon)$. The reason is, we can use an input sparsity reduction to reduce the size of the objective function from $\text{poly}(n)$ to $\text{poly}(k)$.

Claim G.7. Let $W \in \mathbb{R}$ denote a third order tensor that has $r$ distinct columns and $r$ distinct rows. Then it has $2^{O(r \log r)}$ distinct column-row faces.
Proof. By similar arguments as in the proof of Lemma G.3, the bottom (column-row) face can be split into \( r \) groups \( C_1, C_2, \ldots, C_r \) based on \( r \) columns, and split into \( r \) groups \( R_1, R_2, \ldots, R_r \) based on rows. Thus, the bottom (column-row) face can be regarded as having \( r \times r \) groups, and the number in each position of the same group is the same.

We can assume that all the \( r^2 \) blocks in the bottom column-row face have the same size. Otherwise, we can expand the tensor to the situation that all the \( r^2 \) blocks have the same size. Because this small tensor is a sub-tensor of the big tensor, if the big tensor has at most \( t \) distinct column-row faces, then the small tensor has at most \( t \) distinct column-row faces.

By Lemma G.3, we know that the tensor \( W \) has at most \( r \) distinct column-tube faces and row-tube faces. Because it has \( r \) distinct column-tube faces, then all the faces belonging to coordinates in \( C_r \) are the same. Thus, all the columns belonging to \( C_r \) and in the second column-row face are the same. Similarly, we have that all the rows belonging to \( R_r \) and in the second column-row face are the same. Thus we have that all the entries in block \( C_R \cup R_r \) and in the second column-row faces are the same. Further, we can conclude, for every column-row face, for every \( C_i \cup R_j \) block, all the entries in the same block are the same.

The next observation is, if there exist \( r^2 + 1 \) different values in the tensor, then there exist either \( r \) distinct columns or \( r \) distinct rows. Indeed, otherwise since we have \( r \) distinct columns, each column has at most \( r \) distinct entries given our bound on the number of distinct rows. Thus, the \( r \) distinct columns could have at most \( r^2 \) distinct entries in total, a contradiction.

For each column-row face, there are at most \( r^2 \) blocks, and the value in each block can have at most \( r^2 \) possibilities. Thus, overall we have at most \( (r^2)^r = 2^{O(r^2 \log r)} \) column-row faces.

By using different argument, we can improve the above bound. Note that we already show in each column-row face of a tensor, it has \( r^2 \) blocks, and all the values in each block have to be the same. Since we have \( r \) distinct rows, we can fix the those \( r \) distinct rows. If we copy row \( v \) into one row of \( R_i \), then we have to copy row \( v \) into every row of \( R_i \). This is because if \( R_i \) contains two distinct rows, then there must exist a block \( C_j \) for which the entries in block \( R_i \cup C_j \) are not all the same. Thus, for each row group, all the rows in that group are the same.

Now, for each column-row face, consider the leftmost \( r \) blocks, \( R_1 \cup C_1, R_2 \cup C_1, \ldots, R_r \cup C_1 \). There are at most \( r \) possible values in each block, because we have \( r \) distinct rows in total. Overall the total number of possibilities for the leftmost \( r \) blocks is at most \( (r)^r = 2^{O(r \log r)} \). Once the leftmost \( r \) blocks are determined, the remaining \( r(r-1) \) are also determined. This completes the proof.

Also, notice that there is an example that has \( 2^{\Omega(r \log r)} \) distinct column-row faces. For the bottom column-row faces, there are \( r \times r \) blocks for which all the blocks have the same size, the blocks on the diagonal have all 1s, and all the other blocks contain 0s everywhere. For the later column-row faces, we can arbitrarily permute this block diagonal matrix, and the total number of possibilities is \( \Omega(r!) \geq 2^{\Omega(r \log r)} \).
H  Hardness

We first provide definitions and results for some fundamental problems in Section H.1. Section H.2 presents our hardness result for the symmetric tensor eigenvalue problem. Section H.3 presents our hardness results for symmetric tensor singular value problems, computing tensor spectral norm, and rank-1 approximation. We improve Håstad’s NP-hardness\cite{Hastad90} result for tensor rank in Section H.4. We also show a better hardness result for robust subspace approximation in Section H.5. Finally, we discuss several other tensor hardness results that are implied by matrix hardness results in Section H.6.

H.1 Definitions

We first provide the definitions for $3\text{SAT}$, ETH, MAX-$3\text{SAT}$, MAX-E$3\text{SAT}$ and then state some fundamental results related to those definitions.

**Definition H.1** ($3\text{SAT}$ problem). Given $n$ variables and $m$ clauses in a conjunctive normal form CNF formula with the size of each clause at most 3, the goal is to decide whether there exists an assignment to the $n$ Boolean variables to make the CNF formula be satisfied.

**Hypothesis H.2** (Exponential Time Hypothesis (ETH) \cite{IPZ98}). There is a $\delta > 0$ such that the $3\text{SAT}$ problem defined in Definition H.1 cannot be solved in $O(2^{\delta n})$ time.

**Definition H.3** (MAX-$3\text{SAT}$). Given $n$ variables and $m$ clauses, a conjunctive normal form CNF formula with the size of each clause at most 3, the goal is to find an assignment that satisfies the largest number of clauses.

We use MAX-E$3\text{SAT}$ to denote the version of MAX-$3\text{SAT}$ where each clause contains exactly 3 literals.

**Theorem H.4** (\cite{Hastad01}). For every $\delta > 0$, it is NP-hard to distinguish a satisfiable instance of MAX-E$3\text{SAT}$ from an instance where at most a $7/8 + \delta$ fraction of the clauses can be simultaneously satisfied.

**Theorem H.5** (\cite{Hastad01, MR10}). Assume ETH holds. For every $\delta > 0$, there is no $2^{o(n^{1-o(1)})}$ time algorithm to distinguish a satisfiable instance of MAX-E$3\text{SAT}$ from an instance where at most a fraction $7/8 + \delta$ of the clauses can be simultaneously satisfied.

We use MAX-E$3\text{SAT}(B)$ to denote the restricted special case of MAX-$3\text{SAT}$ where every variable occurs in at most $B$ clauses. Håstad \cite{Hastad00} proved that the problem is approximable to within a factor $7/8 + 1/(64B)$ in polynomial time, and that it is hard to approximate within a factor $7/8 + 1/(\log B)^{\Omega(1)}$. In 2001, Trevisan improved the hardness result,

**Theorem H.6** (\cite{Trevisan01}). Unless RP $=$ NP, there is no polynomial time $(7/8 + 5/\sqrt{B})$-approximate algorithm for MAX-E$3\text{SAT}(B)$.

**Theorem H.7** (\cite{Hastad01, Trevisan01, MR10}). Unless ETH fails, there is no $2^{o(n^{1-o(1)})}$ time $(7/8 + 5/\sqrt{B})$-approximate algorithm for MAX-E$3\text{SAT}(B)$.

**Theorem H.8** (\cite{LMS11}). Unless ETH fails, there is no $2^{o(n)}$ time algorithm for the Independent Set problem.

**Definition H.9** (MAX-CUT decision problem). Given a positive integer $c^*$ and an unweighted graph $G = (V, E)$ where $V$ is the set of vertices of $G$ and $E$ is the set of edges of $G$, the goal is to determine whether there is a cut of $G$ that has at least $c^*$ edges.
Note that Feige’s original assumption\cite{feige2002random} states that there is no polynomial time algorithm for the problem in Assumption H.10. We do not know of any better algorithm for the problem in Assumption H.10 and have consulted several experts\cite{personal} about the assumption who do not know a counterexample to it.

**Assumption H.10** (Random Exponential Time Hypothesis). Let $c > \ln 2$ be a constant. Consider a random 3SAT formula on $n$ variables in which each clause has 3 literals, and in which each of the $8n^3$ clauses is picked independently with probability $c/n^2$. Then any algorithm which always outputs 1 when the random formula is satisfiable, and outputs 0 with probability at least 1/2 when the random formula is unsatisfiable, must run in $2^{c'n}$ time on some input, where $c' > 0$ is an absolute constant.

The 4SAT-version of the above random-ETH assumption has been used in \cite{GL04} and \cite{RSW16} (Assumption 1.3).

**H.2 Symmetric tensor eigenvalue**

**Definition H.11** (Tensor Eigenvalue \cite{HL13}). An eigenvector of a tensor $A \in \mathbb{R}^{n \times n \times n}$ is a nonzero vector $x \in \mathbb{R}^n$ such that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j,k} x_i x_j = \lambda x_k, \forall k \in [n]$$

for some $\lambda \in \mathbb{R}$, which is called an eigenvalue of $A$.

**Theorem H.12** (\cite{N03}). Let $G = (V, E)$ on $v$ vertices have stability number (the size of a maximum independent set) $\alpha(G)$. Let $n = v + \frac{v(v-1)}{2}$ and $S^{n-1} = \{(x, y) \in \mathbb{R}^v \times \mathbb{R}^{v(v-1)/2} : \|x\|_2^2 + \|y\|_2^2 = 1\}$. Then,

$$\sqrt{1 - \frac{1}{\alpha(G)}} \geq 3\sqrt{\frac{3}{2}} \max_{(x, y) \in S^{n-1}} \sum_{i<j, (i, j) \notin E} x_i x_j y_{i,j}.$$ 

For any graph $G(V, E)$, we can construct a symmetric tensor $A \in \mathbb{R}^{n \times n \times n}$. For any $1 \leq i < j < k \leq v$, let

$$A_{i,j,k} = \begin{cases} 1 & 1 \leq i < j \leq v, k = v + \phi(i, j), (i, j) \notin E, \\ 0 & \text{otherwise,} \end{cases}$$

where $\phi(i, j) = (i-1)v - i(i-1)/2 + j - i$ is a lexicographical enumeration of the $v(v-1)/2$ pairs $i < j$. For the other cases $i < k < j$, $\cdots$, $k < j < i$, we set

$$A_{i,j,k} = A_{i,k,j} = A_{j,i,k} = A_{j,k,i} = A_{k,i,j} = A_{k,j,i}.$$ 

If two or more indices are equal, we set $A_{i,j,k} = 0$. Thus tensor $T$ has the following property,

$$A(z, z, z) = 6 \sum_{i<j, (i,j) \notin E} x_i x_j y_{i,j},$$

where $z = (x, y) \in \mathbb{R}^n$.

\footnote{Personal communication with Russell Impagliazzo and Ryan Williams.}
Thus, we have
\[ \lambda = \max_{z \in S^{n-1}} A(z, z, z) = \max_{(x, y) \in S^{n-1}} 6 \sum_{i<j, (i, j) \notin E} x_i x_j y_{i,j}. \]

Furthermore, \( \lambda \) is the maximum eigenvalue of \( A \).

**Theorem H.13.** Unless ETH fails, there is no \( 2^{o(\sqrt{n})} \) time to approximate the largest eigenvalue of an \( n \)-dimensional symmetric tensor within \( (1 \pm \Theta(1/n)) \) relative error.

**Proof.** The additive error is at least
\[ \sqrt{1 - 1/v} - \sqrt{1 - 1/(v - 1)} \geq \frac{1}{v - 1} - 1/v \geq 1/v^2. \]

Thus, the relative error is \( (1 \pm \Theta(1/v^2)) \). By the definition of \( n \), we know \( n = \Theta(v^2) \). Assuming ETH, there is no \( 2^{o(v)} \) time algorithm to compute the clique number of \( G \). Because the clique number of \( G \) is \( \alpha(G) \), there is no \( 2^{o(v)} \) time algorithm to compute \( \alpha(G) \). Furthermore, there is no \( 2^{o(v)} \) time algorithm to approximate the maximum eigenvalue within \( (1 \pm \Theta(1/v^2)) \) relative error. Thus, we complete the proof.

**Corollary H.14.** Unless ETH fails, there is no polynomial running time algorithm to approximate the largest eigenvalue of an \( n \)-dimensional tensor within \( (1 \pm \Theta(1/\log^{2+\gamma}(n))) \) relative error, where \( \gamma > 0 \) is an arbitrarily small constant.

**Proof.** We can apply a padding argument here. According to Theorem H.13, there is a \( d \)-dimensional tensor such that there is no \( 2^{o(\sqrt{d})} \) time algorithm that can give a \( (1 + \Theta(1/d)) \) relative error approximation. If we pad 0s everywhere to extend the size of the tensor to \( n = 2^{d(1-\gamma')/2} \), where \( \gamma' > 0 \) is a sufficiently small constant, then \( \text{poly}(n) = 2^{o(\sqrt{d})} \), so \( d = \log^{2+O(\gamma')}(n) \). Thus, it means that there is no polynomial running time algorithm which can output a \( (1 + 1/(\log^{2+\gamma})) \)-relative approximation to the tensor which has size \( n \).

---

**H.3 Symmetric tensor singular value, spectral norm and rank-1 approximation**

[HL13] defines two kinds of singular values of a tensor. In this paper, we only consider the following kind:

**Definition H.15 (\( \ell_2 \) singular value in [HL13]).** Given a 3rd order tensor \( A \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), the number \( \sigma \in \mathbb{R} \) is called a singular value and the nonzero \( u \in \mathbb{R}^{n_1}, v \in \mathbb{R}^{n_2}, w \in \mathbb{R}^{n_3} \) are called singular vectors of \( A \) if
\[
\sum_{j=1}^{n_2} \sum_{k=1}^{n_3} A_{i,j,k} v_j w_k = \sigma u_i, \forall i \in [n_1] \\
\sum_{i=1}^{n_1} \sum_{k=1}^{n_3} A_{i,j,k} u_i w_k = \sigma v_j, \forall j \in [n_2] \\
\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A_{i,j,k} u_i v_j = \sigma w_k, \forall k \in [n_3].
\]
Definition H.16 (Spectral norm [HL13]). The spectral norm of a tensor $A$ is:

$$\|A\|_2 = \sup_{x,y,z \neq 0} \frac{|A(x,y,z)|}{\|x\|_2 \|y\|_2 \|z\|_2}.$$  

Notice that the spectral norm is the absolute value of either the maximum value of $\frac{A(x,y,z)}{\|x\|_2 \|y\|_2 \|z\|_2}$ or the minimum value of it. Thus, it is an $\ell_2$-singular value of $A$. Furthermore, it is the maximum $\ell_2$-singular value of $A$.

Theorem H.17 ([Ban38]). Let $A \in \mathbb{R}^{n \times n \times n}$ be a symmetric 3rd order tensor. Then,

$$\|A\|_2 = \sup_{x,y,z \neq 0} \frac{A(x,y,z)}{\|x\|_2 \|y\|_2 \|z\|_2} = \sup_{x \neq 0} \frac{|A(x,x,x)|}{\|x\|_2^2}.$$  

It means that if a tensor is symmetric, then its largest eigenvalue is the same as its largest singular value and its spectral norm. Then, by combining with Theorem H.13, we have the following corollary:

Corollary H.18. Unless ETH fails,

1. There is no $2^{o(\sqrt{n})}$ time algorithm to approximate the largest singular value of an $n$-dimensional symmetric tensor within $(1 + \Theta(1/n))$ relative-error.
2. There is no $2^{o(\sqrt{n})}$ time algorithm to approximate the spectral norm of an $n$-dimensional symmetric tensor within $(1 + \Theta(1/n))$ relative-error.

By Corollary H.14, we have:

Corollary H.19. Unless ETH fails,

1. There is no polynomial time algorithm to approximate the largest singular value of an $n$-dimensional tensor within $(1 + \Theta(1/\log^{2+\gamma}(n)))$ relative-error, where $\gamma > 0$ is an arbitrarily small constant.
2. There is no polynomial time algorithm to approximate the spectral norm of an $n$-dimensional tensor within $(1+\Theta(1/\log^{2+\gamma}(n)))$ relative-error, where $\gamma > 0$ is an arbitrarily small constant.

Now, let us consider Frobenius norm rank-1 approximation.

Theorem H.20 ([Ban38]). Let $A \in \mathbb{R}^{n \times n \times n}$ be a symmetric 3rd order tensor. Then,

$$\min_{\sigma \geq 0, \|u\|_2 = \|v\|_2 = \|w\|_2 = 1} \|A - \sigma u \otimes v \otimes w\|_F = \min_{\lambda \geq 0, \|v\|_2 = 1} \|A - \lambda v \otimes v \otimes v\|_F.$$  

Furthermore, the optimal $\sigma$ and $\lambda$ may be chosen to be equal.

Notice that

$$\|A - \sigma u \otimes v \otimes w\|_F^2 = \|A\|_F^2 - 2\sigma A(u,v,w) + \sigma^2 \|u \otimes v \otimes w\|_F^2.$$  

Then, if $\|u\|_2 = \|v\|_2 = \|w\|_2 = 1$, we have:

$$\|A - \sigma u \otimes v \otimes w\|_F^2 = \|A\|_F^2 - 2\sigma A(u,v,w) + \sigma^2.$$  

When $A(u,v,w) = \sigma$, then the above is minimized.

Thus, we have:

$$\min_{\sigma \geq 0, \|u\|_2 = \|v\|_2 = \|w\|_2 = 1} \|A - \sigma u \otimes v \otimes w\|_F^2 + \|A\|_2^2 = \|A\|_F^2.$$  

It is sufficient to prove the following theorem:
Theorem H.21. Given $A \in \mathbb{R}^{n \times n \times n}$, unless ETH fails, there is no $2^{o(\sqrt{n})}$ time algorithm to compute $u', v', w' \in \mathbb{R}^n$ such that

$$\|A - u' \otimes v' \otimes w'\|_F^2 \leq (1 + \epsilon) \min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2,$$

where $\epsilon = O(1/n^2)$.

Proof. Let $A \in \mathbb{R}^{n \times n \times n}$ be the same hard instance mentioned in Theorem H.12. Notice that each entry of $A$ is either 0 or 1. Thus, $\min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2 \leq \|A\|_F^2$. Notice that Theorem H.12 also implies that it is hard to distinguish the two cases $\|A\|_2 \leq 2\sqrt{2/3} \cdot \sqrt{1 - 1/(c + 1)}$ or $\|A\|_2 \geq 2\sqrt{2/3} \cdot \sqrt{1 - 1/(c + 1)}$ where $c$ is an integer which is no greater than $\sqrt{n}$. So the difference between $(2\sqrt{2/3} \cdot \sqrt{1 - 1/(c + 1)})^2$ and $(2\sqrt{2/3} \cdot \sqrt{1 - 1/(c + 1)})^2$ is at least $\Theta(1/n)$. Since $\|A\|_F^2$ is at most $n$ (see construction of $A$ in the proof of Lemma H.12), $\Theta(1/n)$ is an $\epsilon = O(1/n^2)$ fraction of $\min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2$. Because

$$\min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2 + \|A\|_F^2 = \|A\|_F^2,$$

if we have a $2^{o(\sqrt{n})}$ time algorithm to compute $u', v', w' \in \mathbb{R}^n$ such that

$$\|A - u' \otimes v' \otimes w'\|_F^2 \leq (1 + \epsilon) \min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2$$

for $\epsilon = O(1/n^2)$, it will contradict the fact that we cannot distinguish whether $\|A\|_2 \leq 2\sqrt{2/3} \cdot \sqrt{1 - 1/c}$ or $\|A\|_2 \geq 2\sqrt{2/3} \cdot \sqrt{1 - 1/(c + 1)}$.

Corollary H.22. Given $A \in \mathbb{R}^{n \times n \times n}$, unless ETH fails, for any $\epsilon$ for which $\frac{1}{2} \geq \epsilon \geq c/n^2$ where $c$ is any constant, there is no $2^{o(\epsilon^{-1/4})}$ time algorithm to compute $u', v', w' \in \mathbb{R}^n$ such that

$$\|A - u' \otimes v' \otimes w'\|_F^2 \leq (1 + \epsilon) \min_{u, v, w \in \mathbb{R}^n} \|A - u \otimes v \otimes w\|_F^2.$$

Proof. If $\epsilon = \Omega(1/n^2)$, it means that $n = \Omega(1/\sqrt{\epsilon})$. Then, we can construct a hard instance $B$ with size $m \times m \times m$ where $m = \Theta(1/\sqrt{\epsilon})$, and we can put $B$ into $A$, and let $A$ have zero entries elsewhere. Since $B$ is hard, i.e., there is no $2^{o(m^{-1/2})} = 2^{o(\epsilon^{-1/4})}$ running time to compute a rank-1 approximation to $B$, this means there is no $2^{o(\epsilon^{-1/4})}$ running time algorithm to find an approximate rank-1 approximation to $A$.

Corollary H.23. Unless ETH fails, there is no polynomial time algorithm to approximate the best rank-1 approximation of an $n$-dimensional tensor within $(1 + \Theta(1/\log 2^{3+\gamma}(n)))$ relative-error, where $\gamma > 0$ is an arbitrarily small constant.

Proof. We can apply a padding argument here. According to Theorem H.21, there is a $d$-dimensional tensor such that there is no $2^{o(\sqrt{d})}$ time algorithm which can give a $(1 + \Theta(1/d^4))$ relative approximation. Then, if we pad with 0s everywhere to extend the size of the tensor to $n = 2^{d(1-\gamma')/2}$ where $\gamma' > 0$ is a sufficiently small constant, then poly$(n) = 2^{o(\sqrt{d})}$, and $d^4 = \log^{2+O(\gamma')}(n)$. Thus, it means that there is no polynomial time algorithm which can output a $(1+1/(\log^{2+\gamma}))$-relative error approximation to the tensor which has size $n$.

H.4 Tensor rank is hard to approximate

This section presents the hardness result for approximating tensor rank under ETH. According to our new result, we notice that not only deciding the tensor rank is a hard problem, but also approximating the tensor rank is a hard problem. This therefore strengthens Håstad’s NP-Hadness [Hås90] for computing tensor rank.
Figure 10: Cover number. For a 3SAT instance with \( n \) variables and \( m \) clauses, we can draw a 
bipartite graph which has \( n \) nodes on the left and \( m \) nodes on the right. Each node (blue) on the 
left corresponds to a variable \( x_i \), each node (green) on the right corresponds to a clause \( C_j \). If either 
\( x_i \) or \( \overline{x}_i \) belongs to clause \( C_j \), then we draw a line between these two nodes. Consider an input 
string \( y \in \{0, 1\}^7 \). There exists some unsatisfied clauses with respect to this input string \( y \). For 
example, let \( C_1, C_2 \) and \( C_3 \) denote those unsatisfied clauses. We want to pick a smallest set of 
nodes on the left partition of the graph to guarantee that for each unsatisfied clause in the right 
partition, there exists a node on the left to cover it. The cover number is defined to be the smallest 
such number over all possible input strings.

H.4.1 Cover number

Before getting into the details of the reduction, we provide a definition of an important concept 
called the “cover number” and discuss the cover number for the MAX-E3SAT(B) problem.

Definition H.24 (Cover number). For any 3SAT instance \( S \) with \( n \) variables and \( m \) clauses, we 
are allowed to assign one of three values \( \{0, 1, \ast\} \) to each variable. For each clause, if one of the 
literals outputs true, then the clause outputs true. For each clause, if the corresponding variable of 
one of the literals is assigned to \( \ast \), then the clause outputs true. We say \( y \in \{0, 1\}^n \) is a string, and 
\( z \in \{0, 1, \ast\}^n \) is a star string. For an instance \( S \), if there exists a string \( y \in \{0, 1\}^n \) that causes all 
the clauses to be true, then we say that \( S \) is satisfiable, otherwise it is unsatisfiable. For an instance 
\( S \), let \( Z_S \) denote the set of star strings which cause all of the clauses of \( S \) to be true. For each star 
string \( z \in \{0, 1, \ast\}^n \), let \( \text{star}(z) \) denote the number of \( \ast \)s in the star-string \( z \). We define the “cover 
number” of instance \( S \) to be

\[
\text{cover-number}(S) = \min_{z \in Z_S} \text{star}(z).
\]

Notice that for a satisfiable 3SAT instance \( S \), the cover number \( p \) is 0. Also, for any unsatisfiable 
3SAT instance \( S \), the cover number \( p \) is at least 1. This is because for any input string, there exists 
at least one clause which cannot be satisfied. To fix that clause, we have to assign \( \ast \) to a variable
belonging to that clause. (Assigning * to a variable can be regarded as assigning both 0 and 1 to a variable)

**Lemma H.25.** Let $S$ denote a MAX-E3SAT($B$) instance with $n$ variables and $m$ clauses and $S$ suppose $S$ is at most $7/8 + A$ satisfiable, where $A \in (0, 1/8)$. Then the cover number of $S$ is at least $(1/8 - A)m/B$.

**Proof.** For any input string $y \in \{0, 1\}^n$, there exists at least $(1/8 - A)m$ clauses which are not satisfied. Since each variable appears in at most $B$ clauses, we need to assign * to at least $(1/8 - A)m/B$ variables. Thus, the cover number of $S$ is at least $(1/8 - A)m/B$.

We say $x_1, x_2, \ldots, x_n$ are variables and $x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_n, \overline{x}_n$ are literals.

**Definition H.26.** For a list of clauses $C$ and a set of variables $P$, if for each clause, there exists at least one literal such that the corresponding variable of that literal belongs to $P$, then we say $P$ covers $L$.

**H.4.2 Properties of 3SAT instances**

**Fact H.27.** For any 3SAT instance $S$ with $n$ variables and $m = \Theta(n)$ clauses, let $c > 0$ denote a constant. If $S$ is $(1 - c)m$ satisfiable, then let $y \in \{0, 1\}^n$ denote a string for which $S$ has the smallest number of unsatisfiable clauses. Let $T$ denote the set of unsatisfiable clauses and let $b$ denote the number of variables in $T$. Then $\Omega((cm)^{1/3}) \leq b \leq O(cm)$.

**Proof.** Note that in $S$, there is no duplicate clause. Let $T$ denote the set of unsatisfiable clauses by assigning string $y$ to $S$. First, we can show that any two literals $x_i, \overline{x}_i$ cannot belong to $T$ at the same time. If $x_i$ and $\overline{x}_i$ belong to the same clause, then that clause must be an “always” satisfiable clause. If $x_i$ and $\overline{x}_i$ belong to different clauses, then one of the clauses must be satisfiable. This contradicts the fact that that clause belongs to $T$. Thus, we can assume that literals $x_1, x_2, \ldots, x_b$ belong to $T$.

There are two extreme cases: one is that each clause only contains three literals and each literal appears in exactly one clause in $T$. Then $b = 3cm$. The other case is that each clause contains 3 literals, and each literal appears in as many clauses as possible. Then $\left(\frac{3}{2}\right)^m = cm$, which gives $b = \Theta((cm)^{1/3})$. □

**Lemma H.28.** For a random 3SAT instance, with probability $1 - 2^{-\Omega(\log n \log \log n)}$ there is no literal appearing in at least $\log n$ clauses.

**Proof.** By the property of random 3SAT, for any literal $x$ and any clause $C$, the probability that $x$ appears in $C$ is $\frac{3}{2m}$, i.e., $\Pr[x \in C] = \frac{3}{2m} = \Theta(1/n)$. Let $p$ denote this probability. For any literal $x$,
the probability of $x$ appearing in at least $\log n$ clauses (out of $m$ clauses) is

$$\Pr[ x \text{ appearing in } \geq \log n \text{ clauses } ]$$

$$= \sum_{i=\log n}^{m} \binom{m}{i} p^i (1-p)^{m-i}$$

$$= \sum_{i=\log n}^{m/2} \binom{m}{i} p^i (1-p)^{m-i} + \sum_{i=m/2}^{m} \binom{m}{i} p^i (1-p)^{m-i}$$

$$\leq \sum_{i=\log n}^{m/2} (e m / i)^i p^i + \sum_{i=m/2}^{m} \binom{m}{i} p^i$$

$$\leq (\Theta(1/\log n))^{\log n} + 2 \cdot (2e)^{m/2} \cdot \Theta(1/n)^{m/2}$$

$$\leq 2^{-\Omega(\log n \cdot \log \log n)}$$

Taking a union bound over all the literals, we complete the proof,

$$\Pr[ \# x \text{ appearing in } \geq \log n \text{ clauses } ] \geq 1 - 2^{-\Omega(\log n \cdot \log \log n)}.$$

□

**Lemma H.29.** For a sufficiently large constant $c' > 0$ and a constant $c > 0$, for any random 3SAT instance which has $n$ variables and $m = c'n$ clauses, suppose it is $(1-c)m$ satisfiable. Then with probability $1 - 2^{-\Omega(\log n \cdot \log \log n)}$, for all input strings $y$, among the unsatisfied clauses, each literal appears in $O(\log n)$ places.

**Proof.** This follows by Lemma H.28. □

Next, we show how to reduce the $O(\log n)$ to $O(1)$.

**Lemma H.30.** For a sufficiently large constant $c$, for any random 3SAT instance that has $n$ variables and $m = cn$ clauses, for any constant $B \geq 1, b \in (0, 1)$, with probability at least $1 - \frac{3m}{Bn}$, there exist at least $(1-b)m$ clauses such that each variable (in these $(1-b)m$ clauses) only appears in at most $B$ clauses (out of these $(1-b)m$ clauses).

**Proof.** For each $i \in [m]$, we use $z_i$ to denote the indicator variable such that it is 1, if for each variable in the $i$th clause, it appears in at most $a$ clauses. Let $B \in [1, \infty)$ denote a sufficiently large constant, which we will decide upon later.

For each variable $x$, the probability of it appearing in the $i$-th clause is $\frac{3}{n}$. Then we have

$$\mathbb{E}[ \# \text{ clauses that contain x } ] = \sum_{i=1}^{m} \mathbb{E}[i\text{-th clause contains x } ] = \frac{3m}{n}$$

By Markov’s inequality,

$$\Pr[ \# \text{ clauses that contain x } \geq a ] \leq \mathbb{E}[ \# \text{ clauses that contain x } ] / B = \frac{3m}{Bn}$$
By a union bound, we can compute $E[z_i]$, 

$$E[z_i] = \Pr[z_i = 1] \geq 1 - 3 \Pr[\text{one variable in } i\text{-th clause appearing } \geq B \text{ clauses }] \geq 1 - \frac{9m}{Bn}.$$  

Furthermore, we have 

$$E[z] = E[\sum_{i=1}^{m} z_i] = \sum_{i=1}^{m} E[z_i] \geq (1 - \frac{9m}{Bn})m.$$  

Note that $z \leq m$. Thus $E[z] \leq m$. Let $b \in (0,1)$ denote a sufficiently small constant. We can show 

$$\Pr[m - z \geq bm] \leq \frac{E[m - z]}{bm} \leq \frac{m - E[z]}{bm} \leq \frac{m - (1 - \frac{9m}{Bn})m}{bm} = \frac{9m}{Bbm}. $$

This implies that with probability at least $1 - \frac{9m}{Bbm}$, we have $m - z \leq bm$. Notice that in random-ETH, $m = cn$ for a constant $c$. Thus, by choosing a sufficiently large constant $B$ (which is a function of $c,b$), we can obtain arbitrarily large constant success probability.

**H.4.3 Reduction**

We reduce 3SAT to tensor rank by following the same construction in [Has90]. To obtain a stronger hardness result, we use the property that each variable only appears in at most $B$ (some constant) clauses and that the cover number of an unsatisfiable 3SAT instance is large. Note that both MAX-E3SAT(B) instances and random-ETH instances have that property. Also each MAX-E3SAT(B) is also a 3SAT instance. Thus if the reduction holds for 3SAT, it also holds for MAX-E3SAT(B), and similarly for random-ETH.

Recall the definition of 3SAT: 3SAT is the problem of given a Boolean formula of $n$ variables in CNF form with at most 3 variables in each of the $m$ clauses, is it possible to find a satisfying assignment to the formula? We say $x_1, x_2, \cdots, x_n$ are variables and $x_1, \bar{x}_1, x_2, \bar{x}_2, \cdots, x_n, \bar{x}_n$ are literals. We transform this to the problem of computing the rank of a tensor of size $n_1 \times n_2 \times n_3$ where $n_1 = 2 + n + 2m, n_2 = 3n$ and $n_3 = 3n + m$. $T$ has the following $n_3$ column-row faces, where each of the faces is an $n_1 \times n_2$ matrix,

- $n$ variable matrices $V_i \in \mathbb{R}^{n_1 \times n_2}$. It has a 1 in positions $(1, 2i - 1)$ and $(2, 2i)$ while all other elements are 0.

- $n$ help matrices $S_i \in \mathbb{R}^{n_1 \times n_2}$. It has a 1 position in $(1, 2n + i)$ and is 0 otherwise.

- $n$ help matrices $M_i \in \mathbb{R}^{n_1 \times n_2}$. It has a 1 in positions $(1, 2i - 1), (2 + i, 2i)$ and $(2 + i, 2n + i)$ and is 0 otherwise.

153
There are $3n + m$ column-row faces, $V_i, \forall i \in [n]$, $S_i, \forall i \in [n]$, $M_i, \forall i \in [n]$, $C_l, \forall l \in [m]$. In face $C_l$, each $u_{l,j}$ is either $x_i$ or $\overline{x}_i$ where $x_i = e_{2i - 1}$ and $\overline{x}_i = e_{2i - 1} + e_{2i}$.

- $m$ clause matrices $C_l \in \mathbb{R}^{n_1 \times n_2}$. Suppose the clause $c_l$ contains the literals $u_{l,1}, u_{l,2}$ and $u_{l,3}$. For each $j \in [3]$, $u_{l,j} \in \{x_1, x_2, \ldots, x_n, \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\}$. Note that $x_i, \overline{x}_i$ are the literals of the 3SAT formula. We can also think of $x_i, \overline{x}_i$ as length $3n$ vectors. Let $x_i$ denote the vector that has a 1 in position $2i - 1$, i.e., $x_i = e_{2i - 1}$. Let $\overline{x}_i$ denote the vector that has a 1 in positions $2i - 1$ and $2i$, $\overline{x}_i = e_{2i - 1} + e_{2i}$.

- Row 1 is the vector $u_{l,1} \in \mathbb{R}^{3n}$,
- Row $2 + n + 2l - 1$ is the vector $u_{l,1} - u_{l,2} \in \mathbb{R}^{3n}$,
- Row $2 + n + 2l$ is the vector $u_{l,1} - u_{l,3} \in \mathbb{R}^{3n}$.  

![Figure 11](image_url)
First, we can obtain Lemma H.31 which follows by Lemma 2 in [Hås90]. For completeness, we provide a proof.

**Lemma H.31.** If the formula is satisfiable, then the constructed tensor has rank at most $4n + 2m$.

**Proof.** We will construct $4n + 2m$ rank-1 matrices $V_i^{(1)}, V_i^{(2)}, S_i^{(1)}, M_i^{(1)}, C_l^{(1)}$ and $C_l^{(2)}$. Then the goal is to show that for each matrix in the set

$$\{V_1, V_2, \ldots, V_n, S_1, S_2, \ldots, S_m, M_1, M_2, \ldots, M_n, C_1, C_2, \ldots, C_m\},$$

it can be written as a linear combination of these constructed matrices.

- Matrices $V_i^{(1)}$ and $V_i^{(2)}$. $V_i^{(1)}$ has the first row equal to $x_i$ iff $\alpha_i = 1$ and otherwise $\overline{x}_i$. All the other rows are 0. We set $V_i^{(2)} = V_i - V_i^{(1)}$.

- Matrices $S_i^{(1)}$. $S_i^{(1)} = S_i$.

- Matrices $M_i^{(1)}$.

$$M_i^{(1)} = \begin{cases} M_i - V_i^{(1)} & \text{if } \alpha_i = 1 \\ M_i - V_i^{(1)} - S_i & \text{if } \alpha_i = 0 \end{cases}$$

- Matrices $C_l^{(1)}$ and $C_l^{(2)}$. Let $x_i = \alpha_i$ be the assignment that makes the clause $c_l$ true. Then $C_l - V_i^{(1)}$ has rank 2, since either it has just two nonzero rows (in the case where $x_i$ is the first variable in the clause) or it has three nonzero rows of which two are equal. In both cases we just need two additional rank 1 matrices.

Once the 3SAT instance $S$ is unsatisfiable, then its cover number is at least 1. For each unsatisfiable 3SAT instance $S$ with cover number $p$, we can show that the constructed tensor has rank at most $4n + 2m + O(p)$ and also has rank at least $4n + 2m + \Omega(p)$. We first prove an upper bound.

**Lemma H.32.** For a 3SAT instance $S$, let $y \in \{0, 1\}$ denote a string such that $S(y)$ has a set $L$ that contains unsatisfiable clauses. Let $p$ denote the smallest number of variables that cover all clauses in $L$. Then the constructed tensor $T$ has rank at most $4n + 2m + p$.

**Proof.** Let $y$ denote a length-$n$ Boolean string $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Based on the assignment $y$, all the clauses of $S$ can be split into two sets: $L$ contains all the unsatisfied clauses and $\overline{T}$ contains all the satisfied clauses. We use set $P$ to denote a set of variables that covers all the clauses in set $L$. Let $p = |P|$. We will construct $4n + 2m + p$ rank-1 matrices $V_i^{(1)}, V_i^{(2)}, S_i^{(1)}, M_i^{(1)}, \forall i \in [n], C_l^{(1)}, C_l^{(2)}, \forall l \in [m], \text{and } V_j^{(3)}, \forall j \in P$. Then the goal is to show that the $V_i, S_i, M_i$ and $C_l$ can be written as linear combinations of these constructed matrices.

- Matrices $V_i^{(1)}$ and $V_i^{(2)}$. $V_i^{(1)}$ has first row equal to $x_i$ iff $\alpha_i = 1$ and otherwise $\overline{x}_i$. All the other rows are 0. We set $V_i^{(2)} = V_i - V_i^{(1)}$.

- Matrices $V_j^{(3)}$. For each $j \in P$, $V_j^{(3)}$ has the first row equal to $x_i$ iff $\alpha_i = 0$ and otherwise $\overline{x}_i$.

- Matrices $S_i^{(1)}$. $S_i^{(1)} = S_i$.  

155
Figure 12: Two possibilities for $V_i^{(1)}, \forall i \in [n], V_i^{(2)}, \forall i \in [n], M_i^{(1)}, \forall i \in [n]$.

- Matrices $M_i^{(1)}$.

$$M_i^{(1)} = \begin{cases} 
M_i - V_i^{(1)} & \text{if } \alpha_i = 1 \\
M_i - V_i^{(1)} - S_i & \text{if } \alpha_i = 0 
\end{cases}$$

- Matrices $C_l^{(1)}$ and $C_l^{(2)}$.

- For each $l \notin L$, clause $c_l$ is satisfied according to assignment $y$. Let $x_i = \alpha_i$ be the assignment that makes the clause $c_l$ true. Then $C_l - V_i^{(1)}$ has rank 2, since either it has just two nonzero rows (in the case where $x_i$ is the first variables in the clause) or it has three nonzero rows of which two are equal. In both cases we just need two additional rank 1 matrices.

- For each $l \in L$. It means clause $c_l$ is unsatisfied according to assignment $y$. Let $x_{j_1} = \alpha_{j_1}, x_{j_2} = \alpha_{j_2}, x_{j_3} = \alpha_{j_3}$ be an assignment that makes the clause $c_l$ false. In other words, one of $j_1, j_2, j_3$ must be $P$ according to the definition that $P$ covers $L$. Then matrix $C_l - V_{j_1}^{(3)}$ has rank 2, since either it has just two nonzero rows (in the case where $x_{j_1}$ is the first variables in the clause) or it has three nonzero rows of which two are equal. In both cases we just need two additional rank 1 matrices.
We finish the proof by taking the $P$ that has the smallest size.

Further, we have:

**Corollary H.33.** For a 3SAT instance $S$, let $p$ denote the cover number of $S$, then the constructed tensor $T$ has rank at most $4n + 2m + p$.

**Proof.** This follows by applying Lemma H.32 to all the input strings and the definition of cover number (Definition H.24).

We can split the tensor $T \in \mathbb{R}^{(2+n)\times 3n \times (3n+m)}$ into two sub-tensors, one is $T_1 \in \mathbb{R}^{2 \times 3n \times (3n+m)}$ (that contains the first two row-tube faces of $T$ and linear combination of the remaining $2m$ row-tube faces of $T$), and the other is $T_2 \in \mathbb{R}^{(n+2m)\times 3n \times (3n+m)}$ (that contains the next $n+2m$ row-tube faces of $T$). We first analyze the rank of $T_1$ and then analyze the rank of $T_2$.

**Claim H.34.** The rank of $T_2$ is $n + 2m$.

**Proof.** According to Figure 11, the nonzero rows are distributed in $n + m$ fully separated sub-tensors. It is obvious that the rank of each one of those $n$ sub-tensors is 1, and the rank of each of those $m$ sub-tensors is 2. Thus, overall, the rank $T_2$ is $n + 2m$.

To make sure $\text{rank}(T) = \text{rank}(T_1) + \text{rank}(T_2)$, the $T_1 \in \mathbb{R}^{2 \times 3n \times (3n+m)}$ can be described as the following $3n + m$ column-row faces, and each of the faces is a $2 \times 3n$ matrix.

- Matrices $\tilde{V}_i, \forall i \in [n]$. The two rows are from the first two rows of $V_i$ in Figure 11, i.e., the first row is $e_{2i-1}$ and the second row is $e_{2i}$.
- Matrices $\tilde{S}_i, \forall i \in [n]$. The two rows are from the first two rows of $S_i$ in Figure 11, i.e., the first row is $e_{2n+i}$ and the second row is zero everywhere else.
- Matrices $\tilde{M}_i, \forall i \in [n]$. The first row is $e_{2i-1} + \beta_{i,1}(e_{2i} + e_{2n+i})$, while the second row is $\beta_{i,2}(e_{2i} + e_{2n+i})$.
- Matrices $\tilde{C}_i, \forall i \in [m]$. The first row is $(1 + \gamma_{i,1} + \gamma_{i,2})u_{i,1} - \gamma_{i,1}u_{i,2} - \gamma_{i,2}u_{i,3}$ and the second is $(\gamma_{i,3} + \gamma_{i,4})u_{i,1} - \gamma_{i,3}u_{i,2} - \gamma_{i,4}u_{i,3}$.
Figure 14: There are $n + p$ matrices $A_i \in \mathbb{R}^{2 \times (2n+p)}, \forall i \in [n+p]$ and $2n + p$ matrices $B_i \in \mathbb{R}^{2 \times (n+p)}, \forall i \in [2n+p]$. Tensor $A$ and tensor $B$ represent the same tensor, and for each $i \in [n+p], j \in [2], l \in [2n+p]$, $(A_i)_{j,l} = (B_l)_{j,i}$.

where for each $i \in [3n]$, we use vector $e_i$ to denote a length $3n$ vector such that it only has a 1 in position $i$ and 0 otherwise. $\beta, \gamma$ are variables. The goal is to show a lower bound for,

$$\text{rank}(T_1).$$

**Lemma H.35.** Let $P$ denote the set $\{i \mid \text{the second row of matrix } \tilde{M}_i \text{ is nonzero, } \forall i \in [n]\}$. Then the rank of $T_1$ is at least $3n + |P|$.

**Proof.** We define $p = |P|$. Without loss of generality, we assume that for each $i \in [p]$, the second row of matrix $\tilde{M}_i$ is nonzero.

Notice that matrices $\tilde{V}_i, \tilde{S}_i, \tilde{M}_i$ have size $2 \times 3n$, but we only focus on the first $2n + p$ columns. Thus, we have $n + p$ column-row faces (from the 3rd dimension) $A_j \in \mathbb{R}^{2 \times (2n+p)}$,

- $A_{ij}, 1 \leq j \leq n$, $A_{ij}$ is the first $2n + p$ columns of $\tilde{V}_j - \sum_{i=1}^{n} \alpha_{ij} \tilde{S}_i \in \mathbb{R}^{2 \times 3n}$, where $\alpha_{ij}$ are some coefficients.

- $A_{n+j}, 1 \leq j \leq p$, $A_{ij}$ is the first $2n + p$ columns of $\tilde{M}_j - \sum_{i=1}^{n} \alpha_{i,n+j} \tilde{S}_i \in \mathbb{R}^{2 \times 3n}$, where $\alpha_{ij}$ are some coefficients.

Consider the first $2n + p$ column-tube faces (from 2nd dimension), $B_j, \forall j \in [2n+p]$, of $T_1$. Notice that these matrices have size $2 \times (n+p)$.

- $B_{2i-1}, 1 \leq i \leq p$, it has a 1 in positions $(1, i)$ and $(1, n+i)$.

- $B_{2i}, 1 \leq i \leq p$, it has $\beta_{i,1}$ in position $(1, n+i), 1$ in position $(2, i)$ and $\beta_{i,2}$ in position $(2, n+i)$.

- $B_{2i-1}, p+1 \leq i \leq n$, it has 1 in position $(1, i)$. 

158
• \( B_{2i}, \ p + 1 \leq i \leq n \), it has 1 in position \((2, i)\).

• \( B_{2n+i}, \ 1 \leq i \leq p \), the first row is unknown, the second row has \( \beta_i, 2 \) in position in \((2, n + i)\).

It is obvious that the first \( 2n \) matrices are linearly independent, thus the rank is at least \( 2n \). We choose the first \( 2n \) matrices as our basis. For \( B_{2n+1} \), we try to write it as a linear combination of the first \( 2n \) matrices \( \{B_i\}_{i \in [2n]} \). Consider the second row of \( B_{2n+1} \). The first \( n \) positions are all 0. The matrices \( B_{2i} \) all have disjoint support for the second row of the first \( n \) columns. Thus, the matrices \( B_{2i} \) should not be used. Consider the second row of \( B_{2l-1}, \forall i \in [n] \). None of them has a nonzero value in position \( n+1 \). Thus \( B_{2n+1} \) cannot be written as a linear combination of the first \( 2n \) matrices. Thus, we can show for any \( i \in [p] \), \( B_{2n+i} \) cannot be written as a linear combination of matrices \( \{B_i\}_{i \in [2n]} \). Consider the \( p \) matrices \( \{B_{2n+i}\}_{i \in [p]} \). Each of them has a different nonzero position in the second row. Thus these matrices are all linearly independent. Putting it all together, we know that the rank of matrices \( \{B_i\}_{i \in [2n+p]} \) is at least \( 2n + p \).

Next, we consider another special case when \( \beta_i, 2 = 0 \), for all \( i \in [n] \). If we subtract \( \beta_i, 1 \) times \( \tilde{S}_i \) from \( \tilde{M}_i \) and leave the other column-row faces (from the 3rd dimension) as they are, and we make all column-tube faces (from the 2nd dimension) for \( j > 2n \) identically 0, then all other choices do not change the first \( 2n \) column-tube faces (from the 2nd dimension) and make some other column-tube faces (from the 2nd dimension) nonzero. Such a choice could clearly only increase the rank of \( T \). Thus, we obtain,

\[
\text{rank}(T) = 2n + 2m + \min \text{rank}(T_3),
\]

where \( T_3 \) is a tensor of size \( 2 \times 2n \times (2n + m) \) given by the following column-row faces (from 3rd dimension) \( A_i, \forall i \in [2n + m] \) and each matrix has size \( 2 \times 2n \) (shown in Figure 15).

- \( A_i, \ i \in [n] \), the first \( 2n \) columns of \( \tilde{V}_i \).
- \( A_{n+i}, \ i \in [n] \), the first \( 2n \) columns of \( \tilde{M}_i \). The first row is \( e_{2i-1} + \beta_i, 1 e_{2i} \), and the second row is 0.
- \( A_{2n+l}, \ l \in [m] \), the first \( 2n \) columns of \( \tilde{C}_l \). The first row is \( (1 + \gamma_{l,1} + \gamma_{l,2})u_{l,1} - \gamma_{l,1}u_{l,2} - \gamma_{l,2}u_{l,3} \), and the second row is \( (\gamma_{l,3} + \gamma_{l,4})u_{l,1} - \gamma_{l,3}u_{l,2} - \gamma_{l,4}u_{l,3} \).

We can show

Lemma H.36. Let \( p \) denote the cover number of the 3SAT instance. \( T_3 \) has rank at least \( 2n + \Omega(p) \).

Proof. First, we can show that all matrices \( A_{n+i} - A_i \) and \( A_{n+i} \) (for all \( i \in [n] \)) are in the expansion of tensor \( T_3 \). Thus, the rank of \( T_3 \) is at least \( 2n \).

We need the following claim:

Claim H.37. For any \( l \in [m] \), if \( A_{2n+l} \) can be written as a linear combination of \( \{A_{n+i} - A_i\}_{i \in [n]} \) and \( \{A_{n+i}\}_{i \in [n]} \), then the second row of \( A_{2n+l} \) is 0, and the first row of one of the \( A_{n+i} \) is \( u_i \), where \( u_i \) is one of the literals appearing in clause \( \ell_i \).

Proof. We prove this for the second row first. For each \( l \in [m] \), we consider the possibility of using all matrices \( A_{n+i} - A_i \) and \( A_{n+i} \) to express matrix \( A_{2n+l} \). If the second row of \( A_{2n+l} \) is nonzero, then it must have a nonzero entry in an odd position. But there is no nonzero in an odd position of the second row of any of matrices \( A_{n+i} - A_i \) and \( A_{n+i} \).
Figure 15: For any $i \in [n]$, $\beta_{i,1} \in \mathbb{R}$, for any $l \in [m]$, $\gamma_{l,1}, \gamma_{l,2} \in \mathbb{R}$, for any $l \in [m]$, if the first literal of clause $l$ is $x_j$, then row vector $u_{l,1} = e_{2i-1} + e_{2i} \in \mathbb{R}^{2n}$; if the first literal of clause $l$ is $\overline{x}_j$, then row vector $u_{l,1} = e_{2i-1} - e_{2i} \in \mathbb{R}^{2n}$.

For the first row. It is obvious that the first row of $A_{2n+l}$ must have at least one nonzero position, for any $\gamma_{l,1}, \gamma_{l,2}$. Let $u_j$ be a literal belonging to the variable $x_i$ which appears in the first row of $A_{2n+l}$ with a nonzero coefficient. Since only $A_{n+i}$ of all the other $A_{n+s}, \forall s \in [n]$ matrices has nonzero elements in either of the positions $(1, 2i-1)$ or $(1, 2i)$, then $A_{n+i}$ must be used to cancel these elements. Thus, the first row of $A_{n+i}$ must be a multiple of $u_j$ and since the element in position $(1, 2i-1)$ of $A_{n+i}$ is 1, this multiple must be 1.

Note that matrices $A_i, \forall i \in [n]$ have the property that, for any matrix in $\{A_{n+1}, \ldots, A_{2n+m}\}$, it cannot be written as the linear combination of matrices $A_i, \forall i \in [n]$. Let $A \in \mathbb{R}^{(n+m) \times 2n}$ denote a matrix that consists of the first rows of $\{A_{n+1}, \ldots, A_{2n+m}\}$. According to the property of matrices $A_i, \forall i \in [n]$, and that the rank of a tensor is always greater than or equal to the rank of any sub-tensor, we know that

$$\text{rank}(T_3) \geq n + \min \text{rank}(\tilde{A}).$$

**Claim H.38.** For a 3SAT instance $S$, for any input string $y \in \{0, 1\}^n$, set $\beta_{s,1}$ to be the entry-wise flipping of $y$, (I) if the clause $l$ is satisfied, then the $(n+l)$-th row of $\tilde{A} \in \mathbb{R}^{(n+m) \times 2n}$ can be written as a linear combination of the first $n$ rows of $\tilde{A}$, (II) if the clause $l$ is unsatisfied, then the $(n+l)$-th row of $\tilde{A}$ cannot be written as a linear combination of the first $n$ rows of $\tilde{A}$.

**Proof.** Part (I), consider a clause $l$ which is satisfied with input string $y$. Then there must exist a variable $x_i$ belonging to clause $l$ (either literal $x_i$ or literal $\overline{x}_i$) and one of the following holds: if $x_i$ belongs to clause $l$, then $\alpha_i = 1$; if $\overline{x}_i$ belongs to clause $l$, then $\alpha_i = 0$. Suppose clause $l$ contains literal $x_i$. The other case can be proved in a similar way. We consider the $(n+l)$-th row. One of the following assignments $(0,0), (-1,0), (0,-1)$ to $\gamma_{l,1}, \gamma_{l,1}$ is going to set the $(n+l)$-th row of $\tilde{A}$ to be vector $e_{2i-1}$. We consider the $i$-th row of $\tilde{A}$. Since we set $\alpha_i = 1$, then we set $\beta_{i,1} = 0$, it follows that the $i$-th row of $\tilde{A}$ becomes $e_{2i-1}$. Therefore, the $(n+l)$-th row of $\tilde{A}$ can be written as a linear combination of the $\tilde{A}$.

Part (II), consider a clause $l$ which is unsatisfied with input string $y$. Suppose that clause contains three literals $x_{i_1}, x_{i_2}, x_{i_3}$ (the other seven possibilities can be proved in a similar way). Then for input string $y$, we have $\alpha_{i_1} = 0$, $\alpha_{i_2} = 0$ and $\alpha_{i_3} = 0$, otherwise this clause $l$ is satisfied. Consider $i_1$-th row of $\tilde{A}$. It becomes $e_{2i_1-1} + e_{2i_1}$. Similarly for the $i_2$-th row and $i_3$-th row. Consider the $(n+l)$-th row. We can observe that all of positions $2i_1, 2i_2, 2i_3$ must be 0. Any
linear combination formed by the $i_1, i_2, i_3$-th row of $\tilde{A}$ must have one nonzero in one of positions $2i_1, 2i_2, 2i_3$. However, if we consider the $(n + l)$-th row of $\tilde{A}$, one of the positions $2i_1, 2i_2, 2i_3$ must be 0. Also, the remaining $n - 3$ of the first $n$ rows of $\tilde{A}$ also have 0 in positions $2i_1, 2i_2, 2i_3$. Thus, we can show that the $(n + l)$-th row of $\tilde{A}$ cannot be written as a linear combination of the first $n$ rows. Similarly, for the other seven cases.

Note that in order to make sure as many as possible rows in $n + 1, \ldots, n + m$ can be written as linear combinations of the first $n$ rows of $\tilde{A}$, the $\beta_{i, 1}$ should be set to either 0 or 1. Also each possibility of input string $y$ is corresponding to a choice of $\beta_{i, 1}$. According to the above Claim H.38, let $l_0$ denote the smallest number of unsatisfied clauses over the choices of all the $2^n$ input strings. Then over all choices of $\beta, \gamma$, there must exist at least $l_0$ rows of $A_{n+1}, \ldots, A_{n+m}$, such that each of those rows cannot be written as the linear combination of the first $n$ rows.

**Claim H.39.** Let $\tilde{A} \in \mathbb{R}^{(n+m)\times 2n}$ denote a matrix that consists of the first rows of $A_{n+i}, \forall i \in [n]$ and $A_{n+l}, \forall l \in [m]$. Let $p$ denote the cover number of $3\text{SAT}$ instance. Then $\min \text{rank}(\tilde{A}) \geq n + \Omega(p)$.

**Proof.** For any choices of $\{\beta_{i, 1}\}_{i \in [n]}$, there must exist a set of rows out of the next $m$ rows such that, each of those rows cannot be written as a linear combination of the first $n$ rows. Let $L$ denote the set of those rows. Let $t$ denote the maximum size set of disjoint rows from $L$. Since those $t$ rows in $L$ all have disjoint support, they are always linearly independent. Thus the rank is at least $n + t$.

Note that each row corresponds to a unique clause and each clause corresponds to a unique row. We can just pick an arbitrary clause $l$ in $L$, then remove the clauses that are using the same literal as clause $l$ from $L$. Because each variable occurs in at most $B$ clauses, we only need to remove at most $3B$ clauses from $L$. We repeat the procedure until there is no clause $L$. The corresponding rows of all the clauses we picked have disjoint supports, thus we can show a lower bound for $t$,

$$t \geq |L|/(3B) \geq l_0/(3B) \geq p/(9B) \gtrsim p,$$

where the second step follows by $|L| \geq l_0$, the third step follows $3l_0 \geq p$, and the last step follows by $B$ is some constant.

Thus, putting it all together, we complete the proof. 

Now, we consider a general case when there are $q$ different $i \in [n]$ satisfying that $\beta_{i, 2} \neq 0$. Similar to tensor $T_3$, we can obtain $T_4$ such that,

$$\text{rank}(T) = 2n + 2m + \min \text{rank}(T_4)$$

where $T_4$ is a tensor of size $2 \times 2n \times (2n + m)$ given by the following column-row faces (from 3rd dimension) $A_i, \forall i \in [2n + m]$ and each matrix has size $2 \times 2n$ (shown in Figure 16).

- $A_i, i \in [n]$, the first $2n$ columns of $V_i$.
- $A_{n+i}, i \in [q]$, the first $2n$ columns of $M_i$. The first row is $e_{2i-1} + \beta_{i, 1}e_{2i}$, and the second row is $\beta_{i, 2}e_{2i}$.
- $A_{n+i}, i \in [q+1, \ldots, n]$, the first $2n$ columns of $M_i$. The first row is $e_{2i-1} + \beta_{i, 1}e_{2i}$, and the second row is 0.
- $A_{2n+l}, l \in [m]$, the first $2n$ columns of $C_l$. The first row is $(1 + \gamma_{l, 1} + \gamma_{l, 2})u_{l, 1} - \gamma_{l, 1}u_{l, 2} - \gamma_{l, 2}u_{l, 3}$, and the second row is $(\gamma_{l, 3} + \gamma_{l, 4})u_{l, 1} - \gamma_{l, 3}u_{l, 2} - \gamma_{l, 4}u_{l, 3}$.
Figure 16: For any $i \in [n]$, $\beta_{i,1} \in \mathbb{R}$. For any $i \in [q]$, $\beta_{i,2} \in \mathbb{R}$. For any $l \in [m]$, $\gamma_{l,1}, \gamma_{l,2} \in \mathbb{R}$. For any $l \in [m]$, if the first literal of clause $l$ is $x_j$, then row vector $u_{l,1} = e_{2i-1} \in \mathbb{R}^{2n}$; if the first literal of clause $l$ is $\overline{x}_j$, then row vector $u_{l,1} = e_{2i-1} + e_{2i} \in \mathbb{R}^{2n}$.

Note that modifying $q$ entries (from Figure 15 to Figure 16) of a tensor can only decrease the rank by $q$, thus we obtain

**Lemma H.40.** Let $q$ denote the number of $i$ such that $\beta_{i,2} \neq 0$, and let $p$ denote the cover number of the 3SAT instance. Then $T_4$ has rank at least $2n + \Omega(p) - q$.

Combining the two perspectives we have

**Lemma H.41.** Let $p$ denote the cover number of an unsatisfiable 3SAT instance. Then the tensor has rank at least $4n + 2m + \Omega(p)$.

**Proof.** Let $q$ denote the $q$ in Figure 16. From one perspective, we know that the tensor has rank at least $4n + 2m + \Omega(p) - q$. From another perspective, we know that the tensor has rank at least $4n + 2m + q$. Combining them together, we obtain the rank is at least $4n + 2m + \Omega(p)/2$, which is still $4n + 2m + \Omega(p)$.

**Theorem H.42.** Unless ETH fails, there is a $\delta > 0$ and an absolute constant $c_0 > 1$ such that the following holds. For the problem of deciding if the rank of a $q$-th order tensor, $q \geq 3$, with each dimension $n$, is at most $k$ or at least $c_0 k$, there is no $2^{\delta k^{1-o(1)}}$ time algorithm.

**Proof.** The reduction can be split into three parts. The first part reduces the MAX-3SAT problem to the MAX-E3SAT problem by [MR10]. For each MAX-3SAT instance with size $n$, the corresponding MAX-E3SAT instance has size $n^{1+o(1)}$. The second part is by reducing the MAX-E3SAT problem to MAX-E3SAT(B) by [Tre01]. For each MAX-E3SAT instance with size $n$, the corresponding MAX-E3SAT(B) instance has size $\Theta(n)$ when $B$ is a constant. The third part is by reducing the MAX-E3SAT(B) problem to the tensor problem. Combining Theorem H.7, Lemma H.25 with this reduction, we complete the proof.

**Theorem H.43.** Unless random-ETH fails, there is an absolute constant $c_0 > 1$ for which any deterministic algorithm for deciding if the rank of a $q$-th order tensor is at most $k$ or at least $c_0 k$, requires $2^{\Omega(k)}$ time.

**Proof.** This follows by combining the reduction with random-ETH and Lemma H.30.

---

13The first two parts are accomplished by personal communication with Dana Moshkovitz and Govind Ramnarayan.
Note that, if $\text{BPP} = \text{P}$ then it also holds for randomized algorithms which succeed with probability $2/3$.

Indeed, we know that any deterministic algorithm requires $2^\Omega(n)$ running time on tensors that have size $n \times n \times n$. Let $g(n)$ denote a fixed function of $n$, and $g(n) = o(n)$. We change the original tensor from size $n \times n \times n$ to $2^g(n) \times 2^g(n) \times 2^g(n)$ by adding zero entries. Then the number of entries in the new tensor is $2^{3g(n)}$ and the deterministic algorithm still requires $2^\Omega(n)$ running time on this new tensor. Assume there is a randomized algorithm that runs in $2^{cg(n)}$ time, for some constant $c > 3$. Then considering the size of this new tensor, the deterministic algorithm is a super-polynomial time algorithm, but the randomized algorithm is a polynomial time algorithm. Thus, by assuming $\text{BPP} = \text{P}$, we can rule out randomized algorithms, which means Theorem H.43 also holds for randomized algorithms which succeed with probability $2/3$.

We provide some some motivation for the $\text{BPP} = \text{P}$ assumption: this is a standard conjecture in complexity theory, as it is implied by the existence of strong pseudorandom generators or if any problem in deterministic exponential time has exponential size circuits [IW97].

### H.5 Hardness result for robust subspace approximation

This section improves the previous hardness for subspace approximation [CW15a] from $1 \pm 1/\text{poly}(d)$ to $1 \pm 1/\text{poly}(\log d)$. (Note that, we provide the algorithmic results for this problem in Section F.)

**Lemma H.44 ([Dem14]).** For any graph $G$ with $n$ nodes, $m$ edges, for which the maximum degree in graph $G$ is $d$, there exists a $d$-regular graph $G'$ with $2nd - 2m$ nodes such that the clique size of $G'$ is the same as the clique size of $G$.

**Proof.** First we create $d$ copies of the original graph $G$. For each $i \in [n]$, let $v_{i,1}, v_{i,2}, \ldots, v_{i,d}$ denote the set of nodes in $G'$ that are corresponding to $v_i$ in $G$. Let $d_{v_i}$ denote the degree of node $v_i$ in graph $G$. In graph $G'$, we create $d - d_{v_i}$ new nodes $v'_{i,1}, v'_{i,2}, \ldots, v'_{i,d_{v_i}}$ and connect each of them to all of the $v_1, v_2, \ldots, v_d$. Therefore, 1. For each $i \in [n], j \in [d_{v_i}]$, node $v'_{i,j}$ has degree $d$. 2. For each $i \in [n], j \in [d]$, node $v_{i,j}$ has degree $d_{v_i}$ (from the original graph), and $d - d_{v_i}$ degree (from the edges to all the $v'_{i,1}, v'_{i,2}, \ldots, v'_{i,d_{v_i}}$). Thus, we proved the graph $G$ is $d$-regular.

The number of nodes in the new graph $G'$ is,

$$nd + \sum_{i=1}^{n} (d - d_{v_i}) = 2nd - \sum_{i=1}^{n} d_{v_i} = 2nd - 2m.$$

It remains to show the clique size is the same in graph $G$ and $G'$. Since we can always reorder the indices for all the nodes, without loss of generality, let us assume the the first $k$ nodes $v_1, v_2, \ldots, v_k$ forms a $k$-clique that has the largest size. It is obvious that the clique size $k'$ in graph $G'$ is at least $k$, since we make $k$ copies of the original graph and do not delete any edges and nodes. Then we just need to show $k' \leq k$. By the property of the construction, the node in one copy does not connect to a node in any other copy. Consider the new nodes we created. For each node $v'_{i,j}$, consider the neighbors of this node. None of them share a edge. Combining the above two properties gives $k' \leq k$. Thus, we finish the proof.

**Theorem H.45** (Theorem 2.6 in [GJS76]). Any $n$ variable $m$ clauses 3SAT instance can be reduced to a graph $G$ with $24m$ vertices, which is an instance of $10m$-independent set. Furthermore $G$ is a 3-regular graph.

We give the proof for completeness here.
There is a constant \(0 < c < 1\), such that for any \(\epsilon > 0\), there is no \(O(2^{n^{1-\epsilon}})\) time algorithm which can solve \(k\)-clique for an \(n\)-vertex \((n-3)\)-regular graph where \(k = cn\) unless ETH fails.
there exists $\epsilon > 0$ such that we have an algorithm with running time $O(2^{(24m)^{1-\epsilon}})$ which can solve $10m$-clique for a $24m - 3$ regular graph with $24m$ vertices, then we can solve the 3SAT problem in $O(2^{n^{1-\epsilon'}})$ time, where $\epsilon' = \Theta(\epsilon)$. Thus, it contradicts ETH.

**Definition H.47.** Let $V$ be a $k$-dimensional subspace of $\mathbb{R}^d$, represented as the column span of a $d \times k$ matrix with orthonormal columns. We abuse notation and let $V$ be both the subspace and the corresponding matrix. For a set $Q$ of points, let

$$c(Q, V) = \sum_{q \in Q} d(q, V)^p = \sum_{q \in Q} \|q^\top (I -VV^\top)\|^p = \sum_{q \in Q} (\|q\|^2 - \|q^\top V\|^2)^{p/2},$$

be the sum of $p$-th powers of distances of points in $Q$, i.e., $\|Q - QVV^\top\|_v$, with associated $M(x) = |x|^p$.

**Lemma H.48.** For any $k \in [d]$, the $k$-dimensional subspaces $V$ which minimize $c(E, V)$ are exactly the $\binom{d}{k}$ subspaces formed by taking the span of $k$ distinct standard unit vectors $e_i, i \in [d]$. The cost of any such $V$ is $d - k$.

**Theorem H.49.** Given a set $Q$ of poly($d$) points in $\mathbb{R}^d$, for a sufficiently small $\epsilon = 1/\text{poly}(d)$, it is $\text{NP}$-hard to output a $k$-dimensional subspace $V$ of $\mathbb{R}^d$ for which $c(Q, V) \leq (1 + \epsilon)c(Q, V^*)$, where $V^*$ is the $k$-dimensional subspace minimizing the expression $c(Q, V)$, that is $c(Q, V) \geq c(Q, V^*)$ for all $k$-dimensional subspaces $V$.

**Theorem H.50.** For a sufficiently small $\epsilon = 1/\text{poly}(\log(d))$, there exist $1 \leq k \leq d$, unless ETH fails, there is no algorithm that can output a $k$-dimensional subspace $V$ of $\mathbb{R}^d$ for which $c(Q, V) \leq (1 + \epsilon)c(Q, V^*)$, where $V^*$ is the $k$-dimensional subspace minimizing the expression $c(Q, V)$, that is $c(Q, V) \geq c(Q, V^*)$ for all $k$-dimensional subspaces $V$.

**Proof.** The reduction is from the clique problem of $d$-vertices $(d - 3)$-regular graph. We construct the hard instance in the same way as in [CW15a]. Given a $d$-vertices $(d - 3)$-regular graph graph $G$, let $B_1 = d^\alpha, B_2 = d^\beta$ where $\beta > \alpha \geq 1$ are two sufficiently large constants. Let $c$ be such that

$$(1 - 1/B_1)^2 + c^2/B_1 = 1.$$
We construct a $d \times d$ matrix $A$ as the following: $\forall i \in [d]$, let $A_{i,i} = 1 - 1/B_1$ and $\forall i \neq j, A_{i,j} = A_{j,i} = c/\sqrt{B_1}$ if $(i, j)$ is an edge in $G$, and $A_{i,j} = A_{j,i} = 0$ otherwise. Let us construct $A' \in \mathbb{R}^{2d \times d}$ as follows:

$$A' = \begin{bmatrix} A \\ B_2 \cdot I_d \end{bmatrix},$$

where $I_d \in \mathbb{R}^d$ is a $d \times d$ identity matrix.

**Claim H.51** (In proof of Theorem 54 in [CW15a]). Let $V' \in \mathbb{R}^{d \times k}$ satisfy that

$$c(A', V') \leq (1 + 1/d^{\gamma})c(A', V^*),$$

where $A'$ is constructed as the above corresponding to the given graph $G$, and $\gamma > 1$ is a sufficiently large constant, $V^*$ is the optimal solution which minimizes $c(A', V)$. Then if $G$ has a $k$-Clique, given $V'$, there is a poly$(d)$ time algorithm which can find the clique which has size at least $k$.

Now, to apply ETH here, we only need to apply a padding argument. We can construct a matrix $A'' \in \mathbb{R}^{N \times d}$ as follows:

$$A'' = \begin{bmatrix} A' \\ A' \\ \cdots \\ A' \end{bmatrix}.$$

Basically, $A''$ contains $N/(2d)$ copies of $A'$ where $N = 2^{d-\alpha}$, and $0 < \alpha$ is a constant which can be arbitrarily small. Notice that $\forall V \in \mathbb{R}^{d \times k},$

$$c(V, A'') = \sum_{q \in A''} d(q, V)^p = N/(2d) \sum_{q \in A'} d(q, V)^p = N/(2d)c(V, A').$$

So if $V''$ gives a $(1 + 1/d^{\gamma})$ approximation to $A'$, it also gives a $(1 + 1/d^{\gamma})$ approximation to $A'$. So if we can find $V''$ in poly$(N, d)$ time, we can output a $k$-Clique of $G$ in poly$(N, d)$ time. But unless ETH fails, for a sufficiently small constant $\alpha' > 0$ there is no poly$(N, d) = O(2^{d-\alpha'})$ time algorithm that can output a $k$-Clique of $G$. It means that there is no poly$(N, d)$ time algorithm that can compute a $(1 + 1/d^{\gamma}) = (1 + 1/poly(\log(N)))$ approximation to $A''$. To make $A''$ be a square matrix, we can just pad with 0s to make the size of $A''$ be $N \times N$. Thus, we can conclude, unless ETH fails, there is no polynomial algorithm that can compute a $(1 + 1/poly(\log(N)))$ rank-$k$ subspace approximation to a point set with size $N$.

**H.6 Extending hardness from matrices to tensors**

In this section, we briefly state some hardness results which are implied by hardness for matrices. The intuition is that, if there is a hard instance for the matrix problem, then we can always construct a tensor hard instance for the tensor problem as follows: the first face of the tensor is the hard instance matrix and it has all 0s elsewhere. We can prove that the optimal tensor solution will always fit the first face and will have all 0s elsewhere. Then the optimal tensor solution gives an optimal matrix solution.
H.6.1 Entry-wise $\ell_1$ norm and $\ell_1$-$\ell_1$-$\ell_2$ norm

In the following we will show that the hardness for entry-wise $\ell_1$ norm low rank matrix approximation implies the hardness for entry-wise $\ell_1$ norm low rank tensor approximation and asymmetric tensor norm ($\ell_1$-$\ell_1$-$\ell_2$) low rank tensor approximation problems.

**Theorem H.52** (Theorem H.13 in [SWZ17]). **Unless ETH fails, for an arbitrarily small constant $\gamma > 0$, given some matrix $A \in \mathbb{R}^{n \times n}$, there is no algorithm that can compute $\hat{x}, \hat{y} \in \mathbb{R}^n$ s.t.**

\[
\|A - \hat{x}y^\top\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \min_{x,y \in \mathbb{R}^n} \|A - xy^\top\|_1,
\]

**in poly(n) time.**

We can get the hardness for tensors directly.

**Theorem H.53.** **Unless ETH fails, for an arbitrarily small constant $\gamma > 0$, given some tensor $A \in \mathbb{R}^{n \times n \times n}$,**

1. **there is no algorithm that can compute $\hat{x}, \hat{y}, \hat{z} \in \mathbb{R}^n$ s.t.**

\[
\|A - \hat{x} \otimes \hat{y} \otimes \hat{z}\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \min_{x,y,z \in \mathbb{R}^n} \|A - x \otimes y \otimes z\|_1,
\]

**in poly(n) time.**

2. **there is no algorithm can compute $\hat{x}, \hat{y}, \hat{z} \in \mathbb{R}^n$ s.t.**

\[
\|A - \hat{x} \otimes \hat{y} \otimes \hat{z}\|_u \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \min_{x,y,z \in \mathbb{R}^n} \|A - x \otimes y \otimes z\|_u,
\]

**in poly(n) time.**

**Proof.** Let matrix $\hat{A} \in \mathbb{R}^{n \times n}$ be the hard instance in Theorem H.52. We construct tensor $A \in \mathbb{R}^{n \times n \times n}$ as follows: $\forall i,j,l \in [n], l \neq 1$ we let $A_{i,j,l} = \hat{A}_{i,j}, A_{i,j,l} = 0$.

Suppose $\hat{x}, \hat{y}, \hat{z} \in \mathbb{R}^n$ satisfies

\[
\|A - \hat{x} \otimes \hat{y} \otimes \hat{z}\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \min_{x,y,z \in \mathbb{R}^n} \|A - x \otimes y \otimes z\|_1.
\]

Then letting $z' = (1,0,0,\cdots,0)^\top$, we have

\[
\|A - \hat{x} \otimes \hat{y} \otimes z'\|_1 \leq \|A - \hat{x} \otimes \hat{y} \otimes \hat{z}\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \min_{x,y,z \in \mathbb{R}^n} \|A - x \otimes y \otimes z\|_1.
\]

The first inequality follows since $\forall i,j,l \in [n], l \neq 1$, we have $A_{i,j,l} = 0$. Let

\[
x^*, y^* = \arg \min_{x,y \in \mathbb{R}^n} \|A - xy^\top\|_1.
\]

Then

\[
\|A - \hat{x} \otimes \hat{y} \otimes z'\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \|A - \hat{x} \otimes \hat{y} \otimes \hat{z}\|_1 \leq \left(1 + \frac{1}{\log^{1+\gamma}(n)}\right) \|A - x^* \otimes y^* \otimes z'\|_1.
\]
Thus, we have

\[ \| \hat{A} - \hat{x}\hat{y}^\top \|_1 \leq \left( 1 + \frac{1}{\log^{1+\gamma}(n)} \right) \| \hat{A} - x^*(y^*)^\top \|_1. \]

Combining with Theorem H.52, we know that unless ETH fails, there is no \( \text{poly}(n) \) running time algorithm which can output

\[ \| A - \hat{A} - \hat{x}\hat{y}\hat{z}^\top \|_1 \leq \left( 1 + \frac{1}{\log^{1+\gamma}(n)} \right) \min_{x,y,z \in \mathbb{R}^n} \| A - x \otimes y \otimes z \|_1. \]

Similarly, we can prove that if \( \tilde{x}, \tilde{y}, \tilde{z} \in \mathbb{R}^n \) satisfies:

\[ \| A - \tilde{x}\tilde{y}\tilde{z}^\top \|_u \leq \left( 1 + \frac{1}{\log^{1+\gamma}(n)} \right) \min_{x,y,z \in \mathbb{R}^n} \| A - x \otimes y \otimes z \|_u, \]

then

\[ \| \hat{A} - \tilde{x}\tilde{y}^\top \|_1 \leq \left( 1 + \frac{1}{\log^{1+\gamma}(n)} \right) \| \hat{A} - x^*(y^*)^\top \|_1. \]

We complete the proof.

\[ \square \]

**Corollary H.54.** Unless ETH fails, for arbitrarily small constant \( \gamma > 0 \),

1. there is no algorithm that can compute \( (1+\epsilon) \) entry-wise \( \ell_1 \) norm rank-1 tensor approximation in \( 2^{O(1/\epsilon^{1-\gamma})} \) running time. (\( \| \cdot \|_1 \)-norm is defined in Section D)

2. there is no algorithm that can compute \( (1+\epsilon) \) \( \ell_u \)-norm rank-1 tensor approximation in \( 2^{O(1/\epsilon^{1-\gamma})} \) running time. (\( \| \cdot \|_u \)-norm is defined in Section F.3)

**H.6.2 \( \ell_1-\ell_2-\ell_2 \) norm**

**Theorem H.55.** Unless ETH fails, for arbitrarily small constant \( \gamma > 0 \), given some tensor \( A \in \mathbb{R}^{n \times n \times n} \), there is no algorithm can compute \( \hat{U}, \hat{V}, \hat{W} \in \mathbb{R}^{n \times k} \) s.t.

\[ \| A - \hat{U} \otimes \hat{V} \otimes \hat{W} \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \min_{U,V,W \in \mathbb{R}^{n \times k}} \| A - U \otimes V \otimes W \|_v, \]

in \( \text{poly}(n) \) running time. (\( \| \cdot \|_v \)-norm is defined in Section F.2)

**Proof.** Let matrix \( \hat{A} \in \mathbb{R}^{n \times n} \) be the hard instance in Theorem H.50. We construct tensor \( A \in \mathbb{R}^{n \times n \times n} \) as follows: \( \forall i, j, l \in [n], l \neq 1 \) we let \( A_{i,j,1} = \hat{A}_{i,j}, A_{i,j,l} = 0 \).

Suppose \( \hat{U}, \hat{V}, \hat{W} \in \mathbb{R}^{n \times k} \) satisfies

\[ \| A - \hat{U} \otimes \hat{V} \otimes \hat{W} \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \min_{U,V,W \in \mathbb{R}^{n \times k}} \| A - U \otimes V \otimes W \|_v. \]

Let \( W' \in \mathbb{R}^{n \times k} \) be the following:

\[
W' = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & 0
\end{bmatrix},
\]
then we have
\[ \| A - \hat{U} \otimes \hat{V} \otimes W' \|_v \leq \| A - \hat{U} \otimes \hat{V} \otimes \hat{W} \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \min_{U,V,W \in \mathbb{R}^{n \times k}} \| A - U \otimes V \otimes W \|_v. \]

The first inequality follows since \( \forall i, j, l \in [n], l \neq 1 \), we have \( A_{i,j,l} = 0 \). Let
\[ U^*, V^* = \arg \min_{U,V \in \mathbb{R}^{n \times k}} \| \hat{A} - UV^\top \|_v. \]

Then
\[ \| A - \hat{U} \otimes \hat{V} \otimes W' \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \| A - \hat{U} \otimes \hat{V} \otimes \hat{W} \|_v \]
\[ \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \| A - U^* \otimes V^* \otimes W' \|_v. \]

Thus, we have
\[ \| \hat{A} - \hat{U} \hat{V}^\top \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \| \hat{A} - U^*(V^*)^\top \|_v. \]

Combining with Theorem H.50, we know that unless ETH fails, there is no \( \text{poly}(n) \) time algorithm which can output
\[ \| A - \hat{U} \otimes \hat{V} \otimes \hat{W} \|_v \leq \left( 1 + \frac{1}{\text{poly}(\log n)} \right) \min_{U,V,W \in \mathbb{R}^{n \times k}} \| A - U \otimes V \otimes W \|_v. \]

\( \square \)
I Hard Instance

This section provides some hard instances for tensor problems.

I.1 Frobenius CURT decomposition for 3rd order tensor

In this section we will prove that a relative-error Tensor CURT is not possible unless $C$ has $\Omega(k/\epsilon)$ columns from $A$, $R$ has $\Omega(k/\epsilon)$ rows from $A$, $T$ has $\Omega(k/\epsilon)$ tubes from $A$ and $U$ has rank $\Omega(k)$.

We use a similar construction from [BW14, BDM11, DR10] and extend it to the tensor setting.

**Theorem I.1.** There exists a tensor $A \in \mathbb{R}^{n \times n \times n}$ with the following property. Consider a factorization CURT, with $C \in \mathbb{R}^{n \times C}$ containing $c$ columns of $A$, $R \in \mathbb{R}^{n \times R}$ containing $r$ rows of $A$, $T \in \mathbb{R}^{n \times T}$ containing $r$ tubes of $A$, and $U \in \mathbb{R}^{C \times R \times T}$, such that

$$
\left\| A - \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{n} U_{i,j,l} \cdot C_{i} \otimes R_{j} \otimes T_{l} \right\|_{F}^2 \leq (1 + \epsilon) \| A - A_{k} \|_{F}^2.
$$

Then, for any $\epsilon < 1$ and any $k \geq 1$,

$$
c = \Omega(k/\epsilon), \quad r = \Omega(k/\epsilon), \quad t = \Omega(k/\epsilon) \text{ and rank}(U) \geq k/3.
$$

**Proof.** For any $i \in [d]$, let $e_{i} \in \mathbb{R}^{d}$ denote the $i$-th standard basis vector. For $\alpha > 0$ and integer $d > 1$, consider the matrix $D \in \mathbb{R}^{(d+1) \times (d+1)}$,

$$
D = \begin{bmatrix}
e_{1} + \alpha e_{2} & e_{1} + \alpha e_{3} & \cdots & e_{1} + \alpha e_{d+1} \\
1 & 1 & \cdots & 1 & 0 \\
\alpha & 0 & \cdots & 0 \\
\alpha & 0 & \cdots & 0 \\
\alpha & 0 & \cdots & 0 \\
\end{bmatrix}
$$

We construct matrix $B \in \mathbb{R}^{(d+1)k/3 \times (d+1)k/3}$ by repeating matrix $D$ $k/3$ times along its main diagonal,

$$
B = \begin{bmatrix}
D & D & \cdots \\
D & D & \cdots \\
\cdots & \cdots & \cdots \\
D & D & \cdots \\
\end{bmatrix}
$$

Let $m = (d+1)k/3$. We construct a tensor $A \in \mathbb{R}^{n \times n \times n}$ with $n = 3m$ by repeating matrix $B$ three times in the following way,

$$
A_{1,j,l} = B_{j,l}, \forall j, l \in [m] \times [m] \\
A_{m+i,m+1,m+l} = B_{i,l}, \forall i, l \in [m] \times [m] \\
A_{2m+i,2m+j,2m+l} = B_{i,j}, \forall i, j \in [m] \times [m]
$$

and 0 everywhere else. We first state some useful properties for matrix $D$,

$$
D^\top D = \begin{bmatrix}1_d 1_d^\top + \alpha^2 I_d & 0 \\
0 & 0 \end{bmatrix} \in \mathbb{R}^{(d+1) \times (d+1)}
$$

170
where
\[
\begin{align*}
\sigma_i^2(D) &= d + \alpha^2, \\
\sigma_i^2(D) &= \alpha^2, \\
\sigma_{d+1}(D) &= 0.
\end{align*}
\]

By definition of matrix $B$, we can obtain the following properties,
\[
\begin{align*}
\sigma_i^2(B) &= d + \alpha^2, & \forall i &= 1, \cdots, k/3 \\
\sigma_i^2(B) &= \alpha^2, & \forall i &= k/3 + 1, \cdots, dk/3 \\
\sigma_i^2(B) &= 0, & \forall i &= dk + 1, \cdots, dk/3 + k/3
\end{align*}
\]

By definition of $A$, we can copy $B$ into three disjoint $n \times n \times n$ sub-tensors on the main diagonal of tensor $A$. Thus, we have
\[
\begin{align*}
\sigma_i^2(A) &= d + \alpha^2, & \forall i &= 1, \cdots, k \\
\sigma_i^2(A) &= \alpha^2, & \forall i &= k + 1, \cdots, dk \\
\sigma_i^2(A) &= 0, & \forall i &= dk + 1, \cdots, dk + k
\end{align*}
\]

Let $A(k)$ denote the best rank-$k$ approximation to $A$, and let $D_1$ denote the best rank-1 approximation to $D$. Using the above properties, for any $k \geq 1$, we can compute $\|A - A(k)\|_F^2$,
\[
\|A - A_k\|_F^2 = k\|D - D_1\|_F^2 = k(d - 1)\alpha^2. \tag{76}
\]

Suppose we have a CUR decomposition with $c' = o(k/\epsilon)$ columns, $r' = o(k/\epsilon)$ rows or $t' = o(k/\epsilon)$ tubes. Since the tensor is equivalent by looking through any of the 3 dimensions/directions, we just need to show why the cost will be at least $(1 + \epsilon)\|A - A_k\|_F^2$ if we choose $t = o(k/\epsilon)$ columns and $t = o(k/\epsilon)$ rows.

Let $C \in \mathbb{R}^{n \times c}$ denote the optimal solution. Then it should have the following form,
\[
C = \begin{bmatrix}
C_1 \\
C_2 \\
C_3
\end{bmatrix}
\]

where $C_1 \in \mathbb{R}^{m \times c_1}$ contains $c_1$ columns from $A_{1:m,1:m,1:m} \in \mathbb{R}^{m \times m \times m}$, $C_2 \in \mathbb{R}^{m \times c_2}$ contains $c_2$ columns from $A_{m+1:2m,m+1:2m,m+1:2m} \in \mathbb{R}^{m \times m \times m}$, $C_3 \in \mathbb{R}^{m \times c_3}$ contains $c_3$ columns from $A_{2m+1:3m,2m+1:3m,2m+1:3m} \in \mathbb{R}^{m \times m \times m}$.

Let $R \in \mathbb{R}^{n \times r}$ denote the optimal solution. Then it should have the following form,
\[
R = \begin{bmatrix}
R_1 \\
R_2 \\
R_3
\end{bmatrix}
\]

\[
\|A - A(CC^\dagger, RR^\dagger, I)\|_F^2 \geq \|B - R_1R_1^\dagger B\|_F^2 + \|B - C_2C_2^\dagger B\|_F^2 + \|B^\top - C_3C_3^\dagger B^\top\|_F^2. \tag{77}
\]

By the analysis in Proposition 4 of [DV06], we have
\[
\|B - R_1R_1^\dagger B\|_F^2 \geq (k/3)(1 + b \cdot \alpha)\|D - D_{(1)}\|_F^2. \tag{78}
\]

171
\[ \| B - C_2 C_1^\top B \|_F^2 \geq (k/3)(1 + b \cdot \alpha) \| D - D_{(1)} \|_F^2. \]  

(79)

Let \( C_3 \in \mathbb{R}^{m \times c_3} \) contain any \( c_3 \) columns from \( B^\top \). Note that \( C_3 \) contains \( c_3(\leq t) \) columns from \( B^\top \), equivalently \( C_2^\top \) contains \( c_2 \) rows from \( B \). Recall that \( B \) contains \( k \) copies of \( D \in \mathbb{R}^{(d+1)\times(d+1)} \) along its main diagonal. Even if we choose \( t \) columns of \( B^\top \), the cost is at least

\[ \| B^\top - C_3 C_4^\top B^\top \|_F^2 \geq (k/3)\| D - D_{(t)} \|_F^2 \geq (k/3)(d-t)\alpha^2. \]  

(80)

Combining Equations (76), (77), (78), (79), (80), \( \alpha = \epsilon \) gives,

\[
\frac{\| A - CC^\top A \|_F^2}{\| A - A(k) \|_F^2} \geq \frac{\| B - R_1 R_1^\top B \|_F^2 + \| B - C_2 C_2^\top B \|_F^2 + \| B^\top - C_3 C_4^\top B^\top \|_F^2}{k(d-1)\alpha^2} \]
by Eq. (77)

\[
\geq \frac{2(k/3)(1 + b\epsilon)(d-1)\epsilon^2 + (k/3)(d-t)\epsilon^2}{k(d-1)\epsilon^2} \]
by Eq. (78),(79),(80) and \( \alpha = \epsilon \)

\[
= \frac{k(d-1)\epsilon^2 + (k/3)(-t+1)\epsilon^2 + 2(k/3)b\epsilon(d-1)\epsilon^2}{k(d-1)\epsilon^2} \]

\[
= 1 + \frac{(k/3)\epsilon^2(2b\epsilon(d-1) - t + 1)}{k(d-1)\epsilon^2} \]

\[
= 1 + \frac{2b\epsilon(d-1) - t + 1}{3(d-1)} \]

\[
\geq 1 + (b/3)\epsilon \]  

by \( 2t \leq b\epsilon(d-1)/2 \)

\[
\geq 1 + \epsilon. \]

which gives a contradiction. \[ \square \]

### I.2 General Frobenius CURT decomposition for q-th order tensor

In this section, we extend the hard instance for 3rd order tensors to q-th order tensors.

**Theorem I.2.** For any constant \( q \geq 1 \), there exists a tensor \( A \in \mathbb{R}^{n \times n \times \cdots \times n} \) with the following property. Define

\[ \text{OPT} = \min_{\text{rank} - k} \{ A \in \mathbb{R}^{n \times n \times \cdots \times n} \} \| A - A_k \|_F^2. \]

Consider a q-th order factorization CURT, with \( C_1 \in \mathbb{R}^{n \times c_1} \) containing \( c_1 \) columns from the 1st dimension of \( A \), \( C_2 \in \mathbb{R}^{n \times c_2} \) containing \( c_2 \) columns from the 2nd dimension of \( A \), \( \cdots \), \( C_q \in \mathbb{R}^{n \times c_q} \) containing \( c_q \) columns from the q-th dimension of \( A \) and a tensor \( U \in \mathbb{R}^{c_1 \times c_2 \times \cdots \times c_q} \), such that

\[
\left\| A - \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_q=1}^{n} U_{i_1,i_2,\cdots,i_q} \cdot C_{1,i_1} \otimes C_{2,i_2} \otimes \cdots \otimes C_{q,i_q} \right\|_F^2 \leq (1 + \epsilon) \text{OPT}. \]

There exists a constant \( c' < 1 \) such that for any \( \epsilon < c' \) and any \( k \geq 1 \),

\[
c_1 = \Omega(k/\epsilon), \quad c_2 = \Omega(k/\epsilon), \cdots, \quad c_q = \Omega(k/\epsilon) \quad \text{and} \quad \text{rank}(U) \geq c'k. \]
Proof. We use the same matrix $D \in \mathbb{R}^{(d+1) \times (d+1)}$ as the proof of Theorem I.1. Then we can construct matrix $B \in \mathbb{R}^{(d+1)k/q \times (d+1)k/q}$ by repeating matrix $D k/q$ times along the its main diagonal,

$$B = \begin{bmatrix} D & & \\ & D & \\ & & \ddots \\ & & & D \end{bmatrix}$$

Let $m = (d + 1)/q$. We construct a tensor $A \in \mathbb{R}^{n \times n \times \cdots \times n}$ with $n = qm$ by repeating the matrix $q$ times in the following way,

$A_{[1:m],[1:m],[1:1,..,1,1]} = B,$

$A_{m+1,[m+1:2m],[m+1:2m],m+1,m+1,..,m+1,m+1} = B^\top,$

$A_{2m+1,[2m+1:3m],[2m+1:3m],2m+1,..,2m+1,2m+1} = B,$

$A_{3m+1,3m+1,3m+1,..,3m+1,3m+1} = B^\top,$

$\ldots \ldots \ldots$

$A_{(q-2)m+1,(q-2)m+1,(q-2)m+1,(q-2)m+1,\ldots,[(q-2)m+1:(q-1)m],[q-2)m+1:(q-1)m]} = B,$

$A_{[(q-1)m+1:qm],[q-1)m+1,(q-1)m+1,(q-1)m+1,\ldots,(q-1)m+1,(q-1)m+1:qm]} = B^\top,$

where there are $q/2$ $B$s and $q/2$ $B^\top$s on the right when $q$ is even, and there are $(q+1)/2$ $B$s and $(q-1)/2$ $B$s on the right when $q$ is odd. Note that this tensor $A$ is equivalent if we look through any of the $q$ dimensions/directions. Similarly as before, we have

$$\|A - A_{(k)}\|_F^2 = k\|D - D_{(1)}\|_F^2 = k(d - 1)\alpha^2.$$ 

Suppose there is a general CURT decomposition (of this $q$th order tensor), with $c_1 = c_2 = \cdots c_q = o(k/e)$ columns from each dimension. Let $C_1 \in \mathbb{R}^{n \times c_1}, C_2 \in \mathbb{R}^{n \times c_2}, \ldots, C_q \in \mathbb{R}^{n \times c_q}$ denote the optimal solution. Then the $C_i$ should have the following form,

$$C_1 = \begin{bmatrix} C_{1,1} \\ C_{1,2} \\ \vdots \\ C_{1,q} \end{bmatrix}, C_2 = \begin{bmatrix} C_{2,1} \\ C_{2,2} \\ \vdots \\ C_{2,q} \end{bmatrix}, \ldots, C_q = \begin{bmatrix} C_{q,1} \\ C_{q,2} \\ \vdots \\ C_{q,q} \end{bmatrix}$$

(In the rest of the proof, we focus on the case when $q$ is even. Similarly, we can show the same thing when $q$ is odd.) We have

$$\|A - A(C_1,C_2^\dagger,\ldots,C_q,C_q^\dagger)\|_F^2$$

$$\geq \sum_{i=1}^{q/2} \|B - C_{2i-1,2i-1C_{2i-1,2i-1}^\dagger}B\|_F^2 + \|B^\top - C_{2i,2iC_{2i,2i}^\dagger}B\|_F^2$$

$$\geq (q/2) \left((k/q)(1 + ba)\|D - D_{(1)}\|_F^2 + (k/q)(d - t)\alpha^2 \right)$$

$$= (q/2) \left((k/q)(1 + ba)(d - 1)\alpha^2 + (k/q)(d - t)\alpha^2 \right)$$

where the second inequality follows by Equations (79) and (80), and the third step follows by $\|D - D_{(1)}\|_F^2 = (d - 1)\alpha^2$. 

173
Putting it all together, we have

\[ \|A - A(C_1 C_1^T, C_2 C_2^T, \ldots, C_q C_q^T)\|_F^2 \]
\[ \|A - A(k)\|_F^2 \]
\[ \geq \frac{(q/2)((k/q)(1 + b\alpha)(d - 1)\alpha^2 + (k/q)(d - t)\alpha^2)}{k(d - 1)\alpha^2} \]
\[ = \frac{k(d - 1)\alpha^2 + (k/2)b\alpha(d - 1)\alpha^2 + (k/q)(-t + 1)\alpha^2}{k(d - 1)\alpha^2} \]
\[ = 1 + \frac{(k/2)b\alpha(d - 1)\alpha^2 + (k/q)(-t + 1)\alpha^2}{k(d - 1)\alpha^2} \]
\[ \leq 1 + \frac{(k/3)b\alpha(d - 1)\alpha^2}{k(d - 1)\alpha^2} \]
\[ = 1 + (b/3)\epsilon \quad \text{by } \epsilon = \alpha \]
\[ > 1 + \epsilon \quad \text{by } b > 3. \]

which leads to a contradiction. Similarly we can show the rank is at least \(\Omega(k)\). \qed
J Distributed Setting

Input data to large-scale machine learning and data mining tasks may be distributed across different machines. The communication cost becomes the major bottleneck of distributed protocols, and so there is a growing body of work on low rank matrix approximations in the distributed model [TD99, QOSG02, BCL05, BRB08, MBZ10, FEGK13, PMvdG+13, KVW14, BKLW14, BLS+16, BWZ16, WZ16, SWZ17] and also many other machine learning problems such as clustering, boosting, and column subset selection [BBLM14, BLG+15, ABW17]. Thus, it is natural to ask whether our algorithm can be applied in the distributed setting. This section will discuss the distributed Frobenius norm low rank tensor approximation protocol in the so-called arbitrary-partition model (see, e.g. [KVW14, BWZ16]).

In the following, we extend the definition of the arbitrary-partition model [KVW14] to fit our tensor setting.

**Definition J.1** (Arbitrary-partition model [KVW14]). There are $s$ machines, and the $i$th machine holds a tensor $A_i \in \mathbb{R}^{n \times n \times n}$ as its local data tensor. The global data tensor is implicit and is denoted as $A = \sum_{i=1}^s A_i$. Then, we say that $A$ is arbitrarily partitioned into $s$ matrices distributed in the $s$ machines. In addition, there is also a coordinator. In this model, the communication is only allowed between the machines and the coordinator. The total communication cost is the total number of words delivered between machines and the coordinator. Each word has $O(\log(sn))$ bits.

Now, let us introduce the distributed Frobenius norm low rank tensor approximation problem in the arbitrary partition model:

**Definition J.2** (Arbitrary-partition model Frobenius norm rank-$k$ tensor approximation). Tensor $A \in \mathbb{R}^{n \times n \times n}$ is arbitrarily partitioned into $s$ matrices $A_1, A_2, \ldots, A_s$ distributed in $s$ machines respectively, and $\forall i \in [s]$, each entry of $A_i$ is at most $O(\log(sn))$ bits. Given tensor $A$, $k \in \mathbb{N}_+$ and an error parameter $0 < \epsilon < 1$, the goal is to find a distributed protocol in the model of Definition J.1 such that

1. Upon termination, the protocol leaves three matrices $U^*, V^*, W^* \in \mathbb{R}^{n \times k}$ on the coordinator.

2. $U^*, V^*, W^*$ satisfies that
   \[
   \left\| \sum_{i=1}^k U_i^* \otimes V_i^* \otimes W_i^* - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}-k A'} \| A' - A \|_F^2.
   \]

3. The communication cost is as small as possible.

**Theorem J.3.** Suppose tensor $A \in \mathbb{R}^{n \times n \times n}$ is distributed in the arbitrary partition model (See Definition J.1). There is a protocol (in Algorithm 39) which solves the problem in Definition J.2 with constant success probability. In addition, the communication complexity of the protocol is $s(\text{poly}(k/\epsilon) + O(kn))$ words.

**Proof.** Correctness. The correctness is implied by Algorithm 2 and Algorithm 3 (Theorem C.1.) Notice that $A_1 = \sum_{i=1}^s A_{i,1}$, $A_2 = \sum_{i=1}^s A_{i,2}$, $A_3 = \sum_{i=1}^s A_{i,3}$, which means that

\[
Y_1 = T_1 A_1 S_1, Y_2 = T_2 A_2 S_2, Y_3 = T_3 A_3 S_3,
\]

and

\[
C = A(T_1, T_2, T_3).
\]
According to line 23,
\[
X_1^*, X_2^*, X_3^* = \arg \min_{X_1, X_2, X_3} \left\| \sum_{j=1}^{k} (Y_1 X_1)_{ij} \otimes (Y_2 X_2)_{ij} \otimes (Y_3 X_3)_{ij} - C \right\|_F.
\]

According to Lemma C.3, we have
\[
\left\| \sum_{j=1}^{k} (T_1 A_1 S_1 X_1)_{ij} \otimes (T_2 A_2 S_2 X_2)_{ij} \otimes (T_3 A_3 S_3 X_3)_{ij} - A(T_1, T_2, T_3) \right\|_F^2
\leq (1 + O(\epsilon)) \min_{X_1, X_2, X_3} \left\| \sum_{j=1}^{k} (A_1 S_1 X_1)_{ij} \otimes (A_2 S_2 X_2)_{ij} \otimes (A_3 S_3 X_3)_{ij} - A \right\|_F^2
\leq (1 + O(\epsilon)) \min_{U, V, W} \left\| \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i - A \right\|_F^2,
\]
where the last inequality follows by the proof of Theorem C.1. By scaling a constant of \( \epsilon \), we complete the proof of correctness.

**Communication complexity.** Since \( S_1, S_2, S_3 \) are \( w_1 \)-wise independent, and \( T_1, T_2, T_3 \) are \( w_2 \)-wise independent, the communication cost of sending random seeds in line 5 is \( O(s(w_1 + w_2)) \) words, where \( w_1 = O(k), w_2 = O(1) \) (see [KVW14, CW13, Woo14, KN14]). The communication cost in line 18 is \( s \cdot \text{poly}(k/\epsilon) \) words due to \( T_1 A_{i,1} S_1, T_2 A_{i,2} S_2, T_3 A_{i,3} S_3 \in \mathbb{R}^{\text{poly}(k/\epsilon) \times O(k/\epsilon)} \) and \( C_i = A_i(T_1, T_2, T_3) \in \mathbb{R}^{\text{poly}(k/\epsilon) \times \text{poly}(k/\epsilon) \times \text{poly}(k/\epsilon)} \).

Notice that, since \( \forall i \in [s] \) each entry of \( A_i \) has at most \( O(\log(sn)) \) bits, each entry of \( Y_1, Y_2, Y_3, C \) has at most \( O(\log(sn)) \) bits. Due to Theorem J.7, each entry of \( X_1^*, X_2^*, X_3^* \) has at most \( O(\log(sn)) \) bits, and the sizes of \( X_1^*, X_2^*, X_3^* \) are \( \text{poly}(k/\epsilon) \) words. Thus the communication cost in line 24 is \( s \cdot \text{poly}(k/\epsilon) \) words.

Finally, since \( \forall i \in [s], U_i^*, V_i^*, W_i^* \in \mathbb{R}^{n \times k} \), the communication here is at most \( O(skn) \) words. The total communication cost is \( s(\text{poly}(k/\epsilon) + O(kn)) \) words.

**Remark J.4.** If we slightly change the goal in Definition J.2 to the following: the coordinator does not need to output \( U_i^*, V_i^*, W_i^* \), but each machine \( i \) holds \( U_i^*, V_i^*, W_i^* \) such that \( U^* = \sum_{i=1}^{s} U_i^*, V^* = \sum_{i=1}^{s} V_i^*, \) \( W^* = \sum_{i=1}^{s} W_i^* \), then the protocol shown in Algorithm 39 does not have to do the line 28. Thus the total communication cost is at most \( s \cdot \text{poly}(k/\epsilon) \) words in this setting.

**Remark J.5.** Algorithm 39 needs exponential in \( \text{poly}(k/\epsilon) \) running time since it solves a polynomial solver in line 23. Instead of solving line 23, we can solve the following optimization problem:
\[
\alpha^* = \arg \min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_F.
\]
Since it is actually a regression problem, it only takes polynomial running time to get \( \alpha^* \). And according to Lemma C.5,
\[
\sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l}^* \cdot (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l
\]
Algorithm 39 Distributed Frobenius Norm Low Rank Approximation Protocol

1: procedure DISTRIBUTEDFNORMLOWRANKAPPROXPROTOCOL($A, \epsilon, k, s$)
2: \( A \in \mathbb{R}^{n \times n \times n} \) was arbitrarily partitioned into \( s \) matrices \( A_1, \ldots, A_s \in \mathbb{R}^{n \times n \times n} \) on \( s \) machines.
3: \( \text{Coordinator} \quad \text{Machines} \ i \)
4: \( \text{Chooses a random seed.} \)
5: \( \text{Sends it to all machines.} \)
6: \( \begin{align*}
7: s_i &\leftarrow O(k/\epsilon), \forall i \in [3]. \\
8: \text{Agree on } S_i &\in \mathbb{R}^{n^2 \times s_i}, \forall i \in [3] \\
9: \text{which are } w_1\text{-wise independent random} \\
10: N(0, 1/s_i) \text{ Gaussian matrices.} \\
11: t_i &\leftarrow \text{poly}(k/\epsilon), \forall i \in [3]. \\
12: \text{Agree on } T_i &\in \mathbb{R}^{n \times n}, \forall i \in [3] \\
13: \text{which are } w_2\text{-wise independent random} \\
14: \text{sparse embedding matrices.} \\
15: \text{Compute } Y_{i,1} &\leftarrow T_i A_{i,1} S_i, \\
16: Y_{i,2} &\leftarrow T_i A_{i,2} S_i, Y_{i,3} &\leftarrow T_i A_{i,3} S_i. \\
17: \text{Send } Y_{i,1}, Y_{i,2}, Y_{i,3} \text{ to the coordinator.} \\
18: \text{Send } C_i &\leftarrow A_i(T_1, T_2, T_3) \text{ to the coordinator.} \\
19: \end{align*} \)
20: \( \text{<-------------------->} \)
21: \( \begin{align*}
22: &\text{Compute } Y_1 \leftarrow \sum_{i=1}^{s} Y_{i,1}, Y_2 \leftarrow \sum_{i=1}^{s} Y_{i,2}, \\
23: \end{align*} \)
24: \( \text{<-------------------->} \)
25: \( \begin{align*}
26: Y_3 &\leftarrow \sum_{i=1}^{s} Y_{i,3}, C &\leftarrow \sum_{i=1}^{s} C_i. \\
27: \text{Compute } X_1^*, X_2^*, X_3^* \text{ by solving} \\
28: \min_{X_1^*, X_2^*, X_3^*} \| (Y_1 X_1) \otimes (Y_2 X_2) \otimes (Y_3 X_3) - C \|_F \\
29: \text{Send } X_1^*, X_2^*, X_3^* \text{ to machines.} \\
30: \end{align*} \)
31: \( \text{<-------------------->} \)
32: \( \begin{align*}
33: &\text{Compute } U_i^* &\leftarrow A_{i,1} S_1 X_1^*, \\
34: V_i^* &\leftarrow A_{i,2} S_2 X_2^*, W_i^* &\leftarrow A_{i,3} S_3 X_3^ *. \\
35: &\text{Send } U_i^*, V_i^*, W_i^* \text{ to the coordinator.} \\
36: \end{align*} \)
37: \( \text{<-------------------->} \)
38: \( \begin{align*}
39: &\text{ Compute } U^* \leftarrow \sum_{i=1}^{s} U_i^*, \\
40: \text{ Compute } V^* \leftarrow \sum_{i=1}^{s} V_i^*, \\
41: \text{ Compute } W^* \leftarrow \sum_{i=1}^{s} W_i^*. \\
42: \text{return } U^*, V^*, W^*. \\
43: \end{align*} \)
44: \end{procedure}

\( \text{gives a rank-} O(k^3/\epsilon^3) \text{ bicriteria solution.} \)

Further, similar to Theorem C.8, we can solve

\[ \min_{U \in \mathbb{R}^{n \times 2 \times s_3}} \| \sum_{i=1}^{s} \sum_{j=1}^{s_2} U_{i+s_1(j-1)} \otimes (Y_2)_i \otimes (Y_3)_j - C \|_F, \]

where \( C = \sum_i A_i(I, T_2, T_3) \). Thus, we can obtain a rank-\( O(k^2/\epsilon^2) \) in polynomial time.

Remark J.6. If we select sketching matrices \( S_1, S_2, S_3, T_1, T_2, T_3 \) to be random Cauchy matrices,
then we are able to compute distributed entry-wise \( \ell_1 \) norm rank-\( k \) tensor approximation (see Theorem D.17). The communication cost is still \( s(\text{poly}(k/\epsilon) + O(kn)) \) words. If we only require a bicriteria solution, then it only needs polynomial running time.

Using similar techniques as in the proof of Theorem C.45, we can obtain:

**Theorem J.7.** Let \( \max_i \{t_i, d_i\} \leq n \). Given a \( t_1 \times t_2 \times t_3 \) tensor \( A \) and three matrices: a \( t_1 \times d_1 \) matrix \( T_1 \), a \( t_2 \times d_2 \) matrix \( T_2 \), and a \( t_3 \times d_3 \) matrix \( T_3 \). For any \( \delta > 0 \), if there exists a solution to

\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i - A \right\|_F^2 := \text{OPT},
\]

and each entry of \( X_i \) can be expressed using \( O(\log n) \) bits, then there exists an algorithm that takes \( \text{poly}(\log n) \cdot 2^{O(d_1 k + d_2 k + d_3 k)} \) time and outputs three matrices: \( \hat{X}_1 \), \( \hat{X}_2 \), and \( \hat{X}_3 \) such that

\[
\| (T_1 \hat{X}_1) \otimes (T_2 \hat{X}_2) \otimes (T_3 \hat{X}_3) - A \|_F = \text{OPT}.
\]
K Streaming Setting

One of the computation models which is closely related to the distributed model of computation is the streaming model. There is a growing line of work in the streaming model. Some problems are very fundamental in the streaming model such like Heavy Hitters [LNNT16, BCI*16, BCIW16], and streaming numerical linear algebra problems [CW09]. Streaming low rank matrix approximation has been extensively studied by previous work like [CW09, KL11, GP14, Lib13, KLM*14, BWZ16, SWZ17]. In this section, we show that there is a streaming algorithm which can compute a low rank tensor approximation.

In the following, we introduce the turnstile streaming model and the turnstile streaming tensor Frobenius norm low rank approximation problem. The following gives a formal definition of the computation model we study.

Definition K.1 (Turnstile model). Initially, tensor $A \in \mathbb{R}^{n \times n \times n}$ is an all zero tensor. In the turnstile streaming model, there is a stream of update operations, and the $i$th update operation is in the form $(x_i, y_i, z_i, \delta_i)$ where $x_i, y_i, z_i \in [n]$, and $\delta_i \in \mathbb{R}$ has $O(\log n)$ bits. Each $(x_i, y_i, z_i, \delta_i)$ means that $A_{x_i y_i z_i}$ should be incremented by $\delta_i$. And each entry of $A$ has at most $O(\log n)$ bits at the end of the stream. An algorithm in this computation model is only allowed one pass over the stream. At the end of the stream, the algorithm stores a summary of $A$. The space complexity of the algorithm is the total number of words required to compute and store this summary while scanning the stream. Here, each word has at most $O(\log(n))$ bits.

The following is the formal definition of the problem.

Definition K.2 (Turnstile model Frobenius norm rank-$k$ tensor approximation). Given tensor $A \in \mathbb{R}^{n \times n \times n}$, $k \in \mathbb{N}_+$ and an error parameter $1 > \epsilon > 0$, the goal is to design an algorithm in the streaming model of Definition K.1 such that

1. Upon termination, the algorithm outputs three matrices $U^*, V^*, W^* \in \mathbb{R}^{n \times k}$.

2. $U^*, V^*, W^*$ satisfy that

$$\left\| \sum_{i=1}^{k} U_i^* \otimes V_i^* \otimes W_i^* - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}-k} \| A' - A \|_F^2.$$

3. The space complexity of the algorithm is as small as possible.

Theorem K.3. Suppose tensor $A \in \mathbb{R}^{n \times n \times n}$ is given in the turnstile streaming model (see Definition K.1), there is an streaming algorithm (in Algorithm 40) which solves the problem in Definition K.2 with constant success probability. In addition, the space complexity of the algorithm is $\text{poly}(k/\epsilon) + O(nk/\epsilon)$ words.

Proof. Correctness. Similar to the distributed protocol, the correctness of this streaming algorithm is also implied by Algorithm 2 and Algorithm 3 (Theorem C.1.) Notice that at the end of the stream $V_1 = A_1 S_1 \in \mathbb{R}^{n \times s_1}, V_2 = A_2 S_2 \in \mathbb{R}^{n \times s_2}, V_3 = A_3 S_3 \in \mathbb{R}^{n \times s_3}, C = A(T_1, T_2, T_3) \in \mathbb{R}^{t_1 \times t_2 \times t_3}$. It also means that

$$Y_1 = T_1 A_1 S_1, Y_2 = T_2 A_2 S_2, Y_3 = T_3 A_3 S_3.$$
According to line 26 of procedure TurnstileStreaming,

\[
X_1^*, X_2^*, X_3^* = \arg\min_{X_1 \in \mathbb{R}^{s_1 \times k}, X_2 \in \mathbb{R}^{s_2 \times k}, X_3 \in \mathbb{R}^{s_3 \times k}} \left\| \sum_{j=1}^{k} (Y_1 X_1)_j \otimes (Y_2 X_2)_j \otimes (Y_3 X_3)_j - C \right\|_F
\]

According to Lemma C.3, we have

\[
\left\| \sum_{j=1}^{k} (Y_1 X_1)_j \otimes (Y_2 X_2)_j \otimes (Y_3 X_3)_j - C \right\|_F^2 = \left\| \sum_{j=1}^{k} (T_1 A_1 S_1 X_1^*)_j \otimes (T_2 A_2 S_2 X_2^*)_j \otimes (T_3 A_3 S_3 X_3^*)_j - A(T_1, T_2, T_3) \right\|_F^2
\]

\[
\leq (1 + O(\epsilon)) \min_{X_1, X_2, X_3} \left\| \sum_{j=1}^{k} (A_1 S_1 X_1)_j \otimes (A_2 S_2 X_2)_j \otimes (A_3 S_3 X_3)_j - A \right\|_F^2
\]

\[
\leq (1 + O(\epsilon)) \min_{U, V, W} \left\| \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i - A \right\|_F^2,
\]

where the last inequality follows by the proof of Theorem C.1. By scaling a constant of \(\epsilon\), we complete the proof of correctness.

**Space complexity.** Since \(S_1, S_2, S_3\) are \(w_1\)-wise independent, and \(T_1, T_2, T_3\) are \(w_2\)-wise independent, the space needed to construct these sketching matrices in line 3 and line 5 of procedure TurnstileStreaming is \(O(w_1 + w_2)\) words, where \(w_1 = O(k), w_2 = O(1)\) (see [KVW14, CW13, Woo14, KN14]). The cost to maintain \(V_1, V_2, V_3\) is \(O(nk/\epsilon)\) words, and the cost to maintain \(C\) is \(poly(k/\epsilon)\) words.

Notice that, since each entry of \(A\) has at most \(O(\log(sn))\) bits, each entry of \(Y_1, Y_2, Y_3, C\) has at most \(O(\log(sn))\) bits. Due to Theorem J.7, each entry of \(X_1^*, X_2^*, X_3^*\) has at most \(O(\log(sn))\) bits, and the sizes of \(X_1^*, X_2^*, X_3^*\) are \(poly(k/\epsilon)\) words. Thus the space cost in line 26 is \(poly(k/\epsilon)\) words.

The total space cost is \(poly(k/\epsilon) + O(nk/\epsilon)\) words.

**Remark K.4.** In the Algorithm 40, for each update operation, we need \(O(k/\epsilon)\) time to maintain matrices \(V_1, V_2, V_3\), and we need \(poly(k/\epsilon)\) time to maintain tensor \(C\). Thus the update time is \(poly(k/\epsilon)\). At the end of the stream, the time to compute

\[
X_1^*, X_2^*, X_3^* = \arg\min_{X_1, X_2, X_3 \in \mathbb{R}^{O(k/\epsilon) \times k}} \left\| \sum_{j=1}^{k} (Y_1 X_1)_j \otimes (Y_2 X_2)_j \otimes (Y_3 X_3)_j - C \right\|_F,
\]

is exponential in \(poly(k/\epsilon)\) running time since it should use a polynomial system solver. Instead of computing the rank-\(k\) solution, we can solve the following:

\[
\alpha^* = \arg\min_{\alpha \in \mathbb{R}^{s_1 \times s_2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l} \cdot (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l - C \right\|_F
\]
Algorithm 40 Turnstile Frobenius Norm Low Rank Approximation Algorithm

1: procedure TURNSTILESTREAMING($k, S$)
2:   $s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow O(k/\epsilon)$.
3:   Construct sketching matrices $S_i \in \mathbb{R}^{n^2 \times s_i}, \forall i \in [3]$ where entries of $S_1, S_2, S_3$ are $w_1$-wise independent random $N(0, 1/s_i)$ Gaussian variables.
4:   $t_1 \leftarrow t_2 \leftarrow t_3 \leftarrow \text{poly}(k/\epsilon)$.
5:   Construct sparse embedding matrices $T_i \in \mathbb{R}^{t_i \times n}, \forall i \in [3]$ where entries are $w_2$-wise independent.
6:   Initialize matrices:
7:   $V \leftarrow \{0\}^{n \times s_1}, \forall i \in [3]$.
8:   $C \leftarrow \{0\}^{t_1 \times t_2 \times t_3}$
9:   for $i \in [l]$ do
10:      Receive update operation $(x_i, y_i, z_i, \delta_i)$ from the data stream $S$.
11:      for $r = 1 \rightarrow s_1$ do
12:         $(V_1)_{x_i, r} \leftarrow (V_1)_{x_i, r} + \delta_i \cdot (S_1)_{(y_i-1)n+z_i, r}$.
13:      end for
14:      for $r = 1 \rightarrow s_2$ do
15:         $(V_2)_{y_i, r} \leftarrow (V_2)_{y_i, r} + \delta_i \cdot (S_2)_{(z_i-1)n+z_i, r}$.
16:      end for
17:      for $r = 1 \rightarrow s_3$ do
18:         $(V_3)_{z_i, r} \leftarrow (V_3)_{z_i, r} + \delta_i \cdot (S_3)_{(x_i-1)n+y_i, r}$.
19:      end for
20:      for $r = 1 \rightarrow t_1, p = 1 \rightarrow t_2, q = 1 \rightarrow t_3$ do
21:         $C_{r, p, q} \leftarrow C_{r, p, q} + \delta_i \cdot (T_1)_{r, x_i} (T_2)_{p, y_i} (T_3)_{q, z_i}$.
22:      end for
23:   end for
24:   Compute $Y_1 \leftarrow T_1 V_1, Y_2 \leftarrow T_2 V_2, Y_3 \leftarrow T_3 V_3$.
25:   Compute $X_i^* \in \mathbb{R}^{s_i \times k}, \forall i \in [3]$ by solving
26:      $\min_{X_1, X_2, X_3} \| (Y_1 X_1) \otimes (Y_2 X_2) \otimes (Y_3 X_3) - C \|_F$
27:   Compute $U^* \leftarrow V_1 X_1^*, V^* \leftarrow V_2 X_2^*, W^* \leftarrow V_3 X_3^*$.
28:   return $U^*, V^*, W^*$
29: end procedure

which will then give

$$\sum_{i=1}^{s_1} \sum_{j=1}^{s_2} \sum_{l=1}^{s_3} \alpha_{i,j,l}^* \cdot (Y_1)_i \otimes (Y_2)_j \otimes (Y_3)_l$$

to be a rank-$O(k^3/\epsilon^3)$ bicriteria solution.

Further, similar to Theorem C.8, we can solve

$$\min_{U \in \mathbb{R}^{n^2 \times s_3}} \left\| \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} U_{i+s_1(j-1)} \otimes (Y_2)_i \otimes (Y_3)_j - C \right\|_F$$

where $C = \sum_i A_i (I, T_2, T_3)$. Thus, we can obtain a rank-$O(k^2/\epsilon^2)$ in polynomial time.
Remark K.5. If we choose $S_1, S_2, S_3, T_1, T_2, T_3$ to be random Cauchy matrices, then we are able to apply the entry-wise $\ell_1$ norm low rank tensor approximation algorithm (see Theorem D.17) in turnstile model.
L Extension to Other Tensor Ranks

The tensor rank studied in the previous sections is also called the CP rank or canonical rank. The tensor rank can be thought of as a direct extension of the matrix rank. We would like to point out that there are other definitions of tensor rank, e.g., the tucker rank and train rank. In this section we explain how to extend our proofs to other notions of tensor rank. Section L.1 provides the extension to tucker rank, and Section L.2 provides the extension to train rank.

L.1 Tensor Tucker rank

Tensor Tucker rank has been studied in a number of works [KC07, PC08, MH09, ZW13, YC14]. We provide the formal definition here:

**L.1.1 Definitions**

**Definition L.1** (Tucker rank). Given a third order tensor $A \in \mathbb{R}^{n \times n \times n}$, we say $A$ has tucker rank $k$ if $k$ is the smallest integer such that there exist three matrices $U, V, W \in \mathbb{R}^{n \times k}$ and a (small) tensor $C \in \mathbb{R}^{k \times k \times k}$ satisfying

$$A_{i,j,l} = \sum_{i' = 1}^{k} \sum_{j' = 1}^{k} \sum_{l' = 1}^{k} C_{i',j',l'} U_{i,i'} V_{j,j'} W_{l,l'}, \forall i, j, l \in [n] \times [n] \times [n],$$

or equivalently,

$$A = C(U, V, W).$$

**L.1.2 Algorithm**

**Algorithm 41** Probenius Norm Low (Tucker) Rank Approximation

1: procedure FLOWTUCKERRANKAPPROX($A, n, k, \epsilon$) \textit{\textsuperscript{}} Theorem L.2
2: $s_1 \leftarrow s_2 \leftarrow s_3 \leftarrow O(k/\epsilon)$. \textit{\textsuperscript{}}
3: $t_1 \leftarrow t_2 \leftarrow t_3 \leftarrow \text{poly}(k, 1/\epsilon)$. \textit{\textsuperscript{}}
4: Choose sketching matrices $S_1 \in \mathbb{R}^{n^2 \times s_1}$, $S_2 \in \mathbb{R}^{n^2 \times s_2}$, $S_3 \in \mathbb{R}^{n^2 \times s_3}$. \textit{\textsuperscript{}} Definition B.18
5: Choose sketching matrices $T_1 \in \mathbb{R}^{t_1 \times n}$, $T_2 \in \mathbb{R}^{t_2 \times n}$, $T_3 \in \mathbb{R}^{t_3 \times n}$. \textit{\textsuperscript{}}
6: Compute $A_1 S_i, \forall i \in [3]$. \textit{\textsuperscript{}}
7: Compute $T_i A_i S_i, \forall i \in [3]$. \textit{\textsuperscript{}}
8: Compute $B \leftarrow A(T_1, T_2, T_3)$. \textit{\textsuperscript{}}
9: Create variables for $X_i \in \mathbb{R}^{s_i \times k}, \forall i \in [3]$. \textit{\textsuperscript{}}
10: Create variables for $C \in \mathbb{R}^{k \times k \times k}$. \textit{\textsuperscript{}}
11: Run a polynomial system verifier for $\|C((Y_1X_1), (Y_2X_2), (Y_3X_3)) - B\|_F^2$. \textit{\textsuperscript{}}
12: return $C, A_1 S_1 X_1, A_2 S_2 X_2$, and $A_3 S_3 X_3$. \textit{\textsuperscript{}}
13: end procedure \textit{\textsuperscript{}}

**Theorem L.2**. Given a third order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1$ and $\epsilon \in (0, 1)$, there exists an algorithm which takes $O(\text{nnz}(A)) + n \text{poly}(k, 1/\epsilon) + 2^{O(k^2/\epsilon + k^3)}$ time and outputs three matrices $U, V, W \in \mathbb{R}^{n \times k}$, and a tensor $C \in \mathbb{R}^{k \times k \times k}$ for which

$$\|C(U, V, W) - A\|_F^2 \leq (1 + \epsilon) \text{min}_{\text{tucker rank } k} \|A_k - A\|_F^2$$
holds with probability 9/10.

Proof. We define OPT to be

\[ \text{OPT} = \min_{\text{tucker rank}-k} A \|A' - A\|_F^2. \]

Suppose the optimal \( A_k = C^*(U^*, V^*, W^*) \). We fix \( C^* \in \mathbb{R}^{k \times k \times k} \), \( V^* \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \). We use \( V_1^*, V_2^*, \ldots, V_k^* \) to denote the columns of \( V^* \) and \( W_1^*, W_2^*, \ldots, W_k^* \) to denote the columns of \( W^* \).

We consider the following optimization problem,

\[ \min_{U_1, \ldots, U_k \in \mathbb{R}^n} \|C^*(U, V^*, W^*) - A\|_F^2, \]

which is equivalent to

\[ \min_{U_1, \ldots, U_k \in \mathbb{R}^n} \|U \cdot C^*(I, V^*, W^*) - A\|_F^2, \]

because \( C^*(U, V^*, W^*) = U \cdot C^*(I, V^*, W^*) \) according to Definition A.6.

Recall that \( C^*(I, V^*, W^*) \) denotes a \( k \times n \times n \) matrix. Let \( (C^*(I, V^*, W^*))_1 \) denote the matrix obtained by flattening \( C^*(I, V^*, W^*) \) along the first dimension. We use matrix \( Z_1 \) to denote \( (C^*(I, V^*, W^*))_1 \in \mathbb{R}^{k \times n^2} \). Then we can obtain the following equivalent objective function,

\[ \min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2. \]

Notice that \( \min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2 = \text{OPT} \), since \( A_k = U^*Z_1 \).

Let \( S_1^T \in \mathbb{R}^{s_1 \times n^2} \) be the sketching matrix defined in Definition B.18, where \( s_1 = O(k/\epsilon) \). We obtain the following optimization problem,

\[ \min_{U \in \mathbb{R}^{n \times k}} \|UZ_1S_1 - A_1S_1\|_F^2. \]

Let \( \hat{U} \in \mathbb{R}^{n \times k} \) denote the optimal solution to the above optimization problem. Then \( \hat{U} = A_1S_1(Z_1S_1)^\dagger \). By Lemma B.22 and Theorem B.23, we have

\[ \|\hat{U}Z_1 - A_1\|_F^2 \leq (1 + \epsilon) \min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2 = (1 + \epsilon) \text{OPT}, \]

which implies

\[ \|C^*(\hat{U}, V^*, W^*) - A\|_F^2 \leq (1 + \epsilon) \text{OPT}. \]

To write down \( \hat{U}_1, \ldots, \hat{U}_k \), we use the given matrix \( A_1 \), and we create \( s_1 \times k \) variables for matrix \( (Z_1S_1)^\dagger \).

As our second step, we fix \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and we convert tensor \( A \) into matrix \( A_2 \). Let matrix \( Z_2 \) denote \( (C^*(\hat{U}, I, W^*))_2 \in \mathbb{R}^{k \times n^2} \). We consider the following objective function,

\[ \min_{V \in \mathbb{R}^{n \times k}} \|VZ_2 - A_2\|_F^2, \]

for which the optimal cost is at most \((1 + \epsilon) \text{OPT}\).
Let $S_2^T \in \mathbb{R}^{s_2 \times n^2}$ be a sketching matrix defined in Definition B.18, where $s_2 = O(k/\epsilon)$. We sketch $S_2$ on the right of the objective function to obtain a new objective function,

$$\min_{V \in \mathbb{R}^{n \times k}} \|VZ_2S_2 - A_2S_2\|_F^2.$$ 

Let $\hat{V} \in \mathbb{R}^{n \times k}$ denote the optimal solution to the above problem. Then $\hat{V} = A_2S_2(Z_2S_2)\dagger$. By Lemma B.22 and Theorem B.23, we have,

$$\|\hat{V}Z_2 - A_2\|_F^2 \leq (1 + \epsilon) \min_{V \in \mathbb{R}^{n \times k}} \|VZ_2 - A_2\|_F^2 \leq (1 + \epsilon)^2 \text{OPT},$$

which implies

$$\left\| C^*(\hat{U}, \hat{V}, W^*) - A \right\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.$$ 

To write down $\hat{V}_1, \ldots, \hat{V}_k$, we need to use the given matrix $A_2 \in \mathbb{R}^{n^2 \times n}$, and we need to create $s_2 \times k$ variables for matrix $(Z_2S_2)\dagger$.

As our third step, we fix the matrices $\hat{U} \in \mathbb{R}^{n \times k}$ and $\hat{V} \in \mathbb{R}^{n \times k}$. We convert tensor $A \in \mathbb{R}^{n \times n \times n}$ into matrix $A_3 \in \mathbb{R}^{n^2 \times n}$. Let matrix $Z_3$ denote $(C^*(\hat{U}, \hat{V}, I))_3 \in \mathbb{R}^{k \times n^2}$. We consider the following objective function,

$$\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2,$$

which has optimal cost at most $(1 + \epsilon)^2 \text{OPT}.$

Let $S_3^T \in \mathbb{R}^{s_3 \times n^2}$ be a sketching matrix defined in Definition B.18, where $s_3 = O(k/\epsilon)$. We sketch $S_3$ on the right of the objective function to obtain a new objective function,

$$\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3S_3 - A_3S_3\|_F^2.$$

Let $\hat{W} \in \mathbb{R}^{n \times k}$ denote the optimal solution of the above problem. Then $\hat{W} = A_3S_3(Z_3S_3)\dagger$. By Lemma B.22 and Theorem B.23, we have,

$$\|\hat{W}Z_3 - A_3\|_F^2 \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.$$ 

Thus, we have

$$\min_{X_1, X_2, X_3} \|C^*((A_1S_1X_1), (A_2S_2X_2), (A_3S_3X_3)) - A\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.$$ 

Let $V_1 = A_1S_1, V_2 = A_2S_2$, and $V_3 = A_3S_3$. We then apply Lemma C.3, and we obtain $\hat{V}_1, \hat{V}_2, \hat{V}_3, B$. We then apply Theorem C.45. Correctness follows by rescaling $\epsilon$ by a constant factor.

**Running time.** Due to Definition B.18, the running time of line 7 (Algorithm 41) is $O(nnz(A)) + n \text{poly}(k, 1/\epsilon)$. Due to Lemma C.3, line 7 and 8 can be executed in $nnz(A) + n \text{poly}(k, 1/\epsilon)$ time. The running time of line 11 is given by Theorem C.45. (For simplicity, we ignore the bit complexity in the running time.)
L.2 Tensor Train rank

L.2.1 Definitions

The tensor train rank has been studied in several works [Ose11, OTZ11, ZWZ16, PTBD16]. We provide the formal definition here.

**Definition L.3** (Tensor Train rank). Given a third order tensor $A \in \mathbb{R}^{n \times n \times n}$, we say $A$ has train rank $k$ if $k$ is the smallest integer such that there exist three tensors $U \in \mathbb{R}^{1 \times n \times k}$, $V \in \mathbb{R}^{k \times n \times k}$, $W \in \mathbb{R}^{k \times n \times 1}$ satisfying:

$$A_{i,j,l} = \sum_{i_1=1}^{1} \sum_{i_2=1}^{k} \sum_{i_3=1}^{k} U_{i_1,i_2} V_{i_2,i_3} W_{i_3,j,l}, \forall i, j, l \in [n] \times [n] \times [n],$$

or equivalently,

$$A_{i,j,l} = \sum_{i_2=1}^{k} \sum_{i_3=1}^{k} (U_2)_{i,j_2} (V_2)_{j_2,i_3} (W_2)_{i_3,l},$$

where $V_2 \in \mathbb{R}^{n \times k^2}$ denotes the matrix obtained by flattening the tensor $U$ along the second dimension, and $(V_2)_{i,j_2+k(i_2-1)}$ denotes the entry in the $i$-th row and $i_1+k(i_2-1)$-th column of $V_2$. We similarly define $U_2, W_2 \in \mathbb{R}^{n \times k}$.

**Algorithm 42** Frobenius Norm Low (Train) rank Approximation

1: procedure FLOWTrainRankApprox($A, n, k, \epsilon$) \hspace{1cm} \triangleright Theorem L.4
2: $s_1 \leftarrow s_3 \leftarrow O(k/\epsilon)$.
3: $s_2 \leftarrow O(k^2/\epsilon)$.
4: $t_1 \leftarrow t_2 \leftarrow t_3 \leftarrow \text{poly}(k, 1/\epsilon)$.
5: Choose sketching matrices $S_1 \in \mathbb{R}^{n^2 \times s_1}$, $S_2 \in \mathbb{R}^{n^2 \times s_2}$, $S_3 \in \mathbb{R}^{n^2 \times s_3}$. \hspace{1cm} \triangleright Definition B.18
6: Choose sketching matrices $T_1 \in \mathbb{R}^{t_1 \times n}$, $T_2 \in \mathbb{R}^{t_2 \times n}$, $T_3 \in \mathbb{R}^{t_3 \times n}$.
7: Compute $A_i S_i$, $\forall i \in [3]$.
8: Compute $T_i A_i S_i$, $\forall i \in [3]$.
9: Compute $B \leftarrow A(T_1, T_2, T_3)$.
10: Create variables for $X_1 \in \mathbb{R}^{s_1 \times k}$.
11: Create variables for $X_3 \in \mathbb{R}^{s_3 \times k}$.
12: Create variables for $X_2 \in \mathbb{R}^{s_2 \times k}$.
13: Create variables for $C \in \mathbb{R}^{k \times k \times k}$.
14: Run polynomial system verifier for $\| \sum_{i_2=1}^{k} \sum_{i_3=1}^{k} (Y_1 X_1)_{i_2} (Y_2 X_2)_{i_2+k(i_3-1)} (Y_3 X_3)_{i_3} - B \|_F^2$.\hspace{1cm} \triangleright Theorem L.4
15: return $A_1 S_1 X_1$, $A_2 S_2 X_2$, and $A_3 S_3 X_3$.
16: end procedure

L.2.2 Algorithm

**Theorem L.4.** Given a third order tensor $A \in \mathbb{R}^{n \times n \times n}$, for any $k \geq 1, \epsilon \in (0, 1)$, there exists an algorithm which takes $O(n \max(A)) + n \text{poly}(k, 1/\epsilon) + 2^{O(k^2/\epsilon)}$ time and outputs three tensors $U \in \mathbb{R}^{1 \times n \times k}$, $V \in \mathbb{R}^{k \times n \times k}$, $W \in \mathbb{R}^{k \times n \times 1}$ such that

$$\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} (U_2)_{i} \otimes (V_2)_{i+k(j-1)} \otimes (W_2)_{j} - A \right\|_F^2 \leq (1 + \epsilon) \min_{\text{rank}_{-k} A_k} \| A_k - A \|_F^2$$
holds with probability 9/10.

Proof. We define \( \text{OPT} \) as

\[
\text{OPT} = \min_{\text{train rank } - k} \| A' - A \|_F^2.
\]

Suppose the optimal

\[
A_k = \sum_{i=1}^{k} \sum_{j=1}^{k} U_i^* \otimes V_i^* \otimes W_j^*.
\]

We fix \( V^* \in \mathbb{R}^{n \times k^2} \) and \( W^* \in \mathbb{R}^{n \times k} \). We use \( V_1^*, V_2^*, \ldots, V_k^* \) to denote the columns of \( V^* \), and \( W_1^*, W_2^*, \ldots, W_k^* \) to denote the columns of \( W^* \).

We consider the following optimization problem,

\[
\min_{U \in \mathbb{R}^{n \times k}} \left\| \sum_{i=1}^{k} \sum_{j=1}^{k} U_i \otimes V_i^* \otimes W_j^* - A \right\|_F^2,
\]

which is equivalent to

\[
\min_{U \in \mathbb{R}^{n \times k}} \left\| U \cdot \begin{bmatrix}
\sum_{j=1}^{k} V_{1+k(j-1)}^* \otimes W_j^*
\sum_{j=1}^{k} V_{2+k(j-1)}^* \otimes W_j^*
\vdots
\sum_{j=1}^{k} V_{k+k(j-1)}^* \otimes W_j^*
\end{bmatrix} - A \right\|_F.
\]

Let \( A_1 \in \mathbb{R}^{n \times n^2} \) denote the matrix obtained by flattening the tensor \( A \) along the first dimension. We use matrix \( Z_1 \in \mathbb{R}^{k \times n^2} \) to denote

\[
\begin{bmatrix}
\sum_{j=1}^{k} \text{vec}(V_{1+k(j-1)}^* \otimes W_j^*) \\
\sum_{j=1}^{k} \text{vec}(V_{2+k(j-1)}^* \otimes W_j^*) \\
\vdots \\
\sum_{j=1}^{k} \text{vec}(V_{k+k(j-1)}^* \otimes W_j^*)
\end{bmatrix}.
\]

Then we can obtain the following equivalent objective function,

\[
\min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 - A_1 \|_F^2.
\]

Notice that \( \min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 - A_1 \|_F^2 = \text{OPT} \), since \( A_k = U^*Z_1 \).

Let \( S_1^T \in \mathbb{R}^{s_1 \times n^2} \) be a sketching matrix defined in Definition B.18, where \( s_1 = O(k/\epsilon) \). We obtain the following optimization problem,

\[
\min_{U \in \mathbb{R}^{n \times k}} \| UZ_1 S_1 - A_1 S_1 \|_F^2.
\]
Let \( \hat{U} \in \mathbb{R}^{n \times k} \) denote the optimal solution to the above optimization problem. Then \( \hat{U} = A_1 S_1 (Z_1 S_1)^\dagger \). By Lemma B.22 and Theorem B.23, we have

\[
\| \hat{U} Z_1 - A_1 \|_F^2 \leq (1 + \epsilon) \min_{U \in \mathbb{R}^{n \times k}} \| U Z_1 - A_1 \|_F^2 = (1 + \epsilon) \text{OPT},
\]

which implies

\[
\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} \hat{U}_i \otimes V^*_{i+k(j-1)} \otimes W^*_j - A \right\|_F^2 \leq (1 + \epsilon) \text{OPT}.
\]

To write down \( \hat{U}_1, \ldots, \hat{U}_k \), we use the given matrix \( A_1 \), and we create \( s_1 \times k \) variables for matrix \( (Z_1 S_1)^\dagger \).

As our second step, we fix \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( W^* \in \mathbb{R}^{n \times k} \), and we convert the tensor \( A \) into matrix \( A_2 \). Let matrix \( Z_2 \in \mathbb{R}^{k^2 \times n^2} \) denote the matrix where the \((i,j)\)-th row is the vectorization of \( \hat{U}_i \otimes W^*_j \). We consider the following objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 - A_2 \|_F^2,
\]

for which the optimal cost is at most \((1 + \epsilon) \text{OPT}\).

Let \( S_2^\dagger \in \mathbb{R}^{s_2 \times n^2} \) be a sketching matrix defined in Definition B.18, where \( s_2 = O(k^2 / \epsilon) \). We sketch \( S_2 \) on the right of the objective function to obtain the new objective function,

\[
\min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 S_2 - A_2 S_2 \|_F^2.
\]

Let \( \hat{V} \in \mathbb{R}^{n \times k} \) denote the optimal solution of the above problem. Then \( \hat{V} = A_2 S_2 (Z_2 S_2)^\dagger \). By Lemma B.22 and Theorem B.23, we have,

\[
\| \hat{V} Z_2 - A_2 \|_F^2 \leq (1 + \epsilon) \min_{V \in \mathbb{R}^{n \times k}} \| V Z_2 - A_2 \|_F^2 \leq (1 + \epsilon)^2 \text{OPT},
\]

which implies

\[
\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} \hat{U}_i \otimes \hat{V}_{i+k(j-1)} \otimes W^* - A \right\|_F^2 \leq (1 + \epsilon)^2 \text{OPT}.
\]

To write down \( \hat{V}_1, \ldots, \hat{V}_k \), we need to use the given matrix \( A_2 \in \mathbb{R}^{n^2 \times n} \), and we need to create \( s_2 \times k \) variables for matrix \( (Z_2 S_2)^\dagger \).

As our third step, we fix the matrices \( \hat{U} \in \mathbb{R}^{n \times k} \) and \( \hat{V} \in \mathbb{R}^{n \times k} \). We convert tensor \( A \in \mathbb{R}^{n \times n \times n} \) into matrix \( A_3 \in \mathbb{R}^{n^2 \times n} \). Let matrix \( Z_3 \in \mathbb{R}^{k \times n^2} \) denote

\[
\begin{bmatrix}
\sum_{i=1}^{k} \text{vec}(\hat{U}_i \otimes \hat{V}_{i+k(0)}) \\
\sum_{i=1}^{k} \text{vec}(\hat{U}_i \otimes \hat{V}_{i+k(1)}) \\
\vdots \\
\sum_{i=1}^{k} \text{vec}(\hat{U}_i \otimes \hat{V}_{i+k(k-1)})
\end{bmatrix}.
\]

We consider the following objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \| W Z_3 - A_3 \|_F^2,
\]

188
which has optimal cost at most \((1 + \epsilon)^2 \text{OPT}\).

Let \(S_3^T \in \mathbb{R}^{s_3 \times n^2}\) be a sketching matrix defined in Definition B.18, where \(s_3 = O(k/\epsilon)\). We sketch \(S_3\) on the right of the objective function to obtain a new objective function,

\[
\min_{W \in \mathbb{R}^{n \times k}} \|WZ_3S_3 - A_3S_3\|_F^2.
\]

Let \(\hat{W} \in \mathbb{R}^{n \times k}\) denote the optimal solution of the above problem. Then \(\hat{W} = A_3S_3(Z_3S_3)^\dagger\). By Lemma B.22 and Theorem B.23, we have,

\[
\|\hat{W}Z_3 - A_3\|_F^2 \leq (1 + \epsilon) \min_{W \in \mathbb{R}^{n \times k}} \|WZ_3 - A_3\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.
\]

Thus, we have

\[
\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} \sum_{j=1}^{k} (A_1S_1X_1)_i \otimes (A_2S_2X_2)_{i+k(j-1)} \otimes (A_3S_3X_3)_j - A \right\|_F^2 \leq (1 + \epsilon)^3 \text{OPT}.
\]

Let \(V_1 = A_1S_1, V_2 = A_2S_2,\) and \(V_3 = A_3S_3\). We then apply Lemma C.3, and we obtain \(\hat{V}_1, \hat{V}_2, \hat{V}_3, B\). We then apply Theorem C.45. Correctness follows by rescaling \(\epsilon\) by a constant factor.

**Running time.** Due to Definition B.18, the running time of line 7 (Algorithm 42) is \(O(\text{nnz}(A)) + n \text{poly}(k, 1/\epsilon)\). Due to Lemma C.3, lines 8 and 9 can be executed in \(\text{nnz}(A) + n \text{poly}(k, 1/\epsilon)\) time. The running time of \(2^{O(k^4/\epsilon)}\) comes from running Theorem C.45 (For simplicity, we ignore the bit complexity in the running time.)
M Acknowledgments

The authors would like to thank Udit Agarwal, Alexandr Andoni, Arturs Backurs, Saugata Basu, Lijie Chen, Xi Chen, Thomas Dillig, Yu Feng, Rong Ge, Daniel Hsu, Chi Jin, Ravindran Kannan, J. M. Landsberg, Qi Lei, Fu Li, Syed Mohammad Meesum, Ankur Moitra, Dana Moshkovitz, Cameron Musco, Richard Peng, Eric Price, Govind Ramnarayan, Ilya Razenshteyn, James Renegar, Rocco Servedio, Tselil Schramm, Clifford Stein, Wen Sun, Yining Wang, Zhaoran Wang, Wei Ye, Huacheng Yu, Huan Zhang, Kai Zhong, David Zuckerman for useful discussions.
References


Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. In *2014 IEEE 55th


