Efficient and Accurate Tensor Compression via Recursive Sketching

Amit Sharma*
Department of CSE,
IIT Hyderabad.
cs24resch02002@iith.ac.in

Mohammad Azhar Khan*
Department of CSE,
IIT Hyderabad.
cs24mtech12006@iith.ac.in

Rameshwar Pratap
Department of CSE,
IIT Hyderabad.
rameshwar@cse.iith.ac.in

Abstract

The computation of pairwise inner products/ Euclidean distances between high-order tensor data points is a fundamental subroutine in numerous machine learning applications. However, the naive approach to these computations incurs exponential time complexity with respect to the number of modes. The work of [Rakhshan and Rabusseau, 2020] introduced an extension of the random projection tailored for tensor datasets, which compresses large tensors into compact vectors (*a.k.a* sketches). Their approach provides unbiased estimates of the original pairwise Euclidean distance. However, the variance of their estimates grows exponentially with the number of modes, making their estimates less reliable for small sketch sizes. This work propose improved sketching algorithms that provide unbiased estimates for pairwise inner products, with significantly lower variance - independent of the number of modes—compared to that of [Rakhshan and Rabusseau, 2020]. Moreover, our approach offers asymptotically improved time complexity. Our sketching algorithm builds on the framework of [Ahle et al., 2020], which proposed a sketching algorithm for high-degree *polynomial kernels*.

1 INTRODUCTION

Many real-world datasets are inherently multi-modal and can be naturally represented as higher-order tensors $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_p}$, capturing interactions across p modes (e.g., time, space, frequency) [Kolda and Bader, 2009]. However, for notational simplicity and without loss of generality, we assume throughout the paper that all mode dimensions are equal, i.e., $d_1 = d_2 = \cdots = d_p = d$. The number of parameters in an unconstrained tensor grows exponentially with p due to the "curse of dimensionality", leading to prohibitive storage and computation costs. Tensor decomposition techniques alleviate this by approximating $\mathcal{X}$ in compact factorized forms. The CANDECOMP/PARAFAC (CP) decomposition expresses $\mathcal{X}$ as a sum of R rank-1 outer products [Hitchcock, 1927], while the Tucker decomposition generalizes this to a core tensor multiplied along each mode by factor matrices [Tucker, 1966]. More recently, the Tensor Train (TT) format represents $\mathcal{X}$ as a chain of third-order core tensors [Oseledets, 2011]. These low-rank models facilitate time and space efficient training/algorithms for large-scale tensor data.

The computation of pairwise inner product/Euclidean distance is a core subroutine across multiple domains, including collaborative filtering for recommendation systems [Bachrach et al., 2014, Li et al., 2017], document similarity and vector-based search in information retrieval [Xiang et al., 2019, DataStax, 2025], as well as similarity evaluation in clustering algorithms and feature representation in neural networks [Memgraph Team, 2023, Luo et al., 2018]. Beyond these applications, inner products also play a central role in tensor-based learning and analysis [Kolda and Bader, 2009, Ballard and Kolda, 2025]. The time complexity of exact computation of these quantities for mode-p tensors $\mathcal{X} \in \mathbb{R}^{d \times \cdots \times d}$ (p times) grow *exponentially* in the number of modes p. Dimensionality reduction techniques are well-suited for addressing such scenarios. However, a naive approach of applying random projection [Johnson and Lindenstrauss, 1984] requires generating a random tensor $\mathcal{R} \in \mathbb{R}^{d \times \cdots \times d}$ (p times) whose each entries are from $\mathcal{N}(0,1)$, which has time complexity $O(d^p)$, making it impractical for large d, p.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

* Equal Contribution

The work of [Rakhshan and Rabusseau, 2020] introduces the notion of random projections in the context of tensors, and provides significant improvements over the naive approach. They propose two random projection maps, namely $f_{\mathrm{CP}(R)}(\mathcal{X}), f_{\mathrm{TT}(R)}(\mathcal{X})$ by leveraging CP and TT decomposition (Definitions 1, 2 of [Rakhshan and Rabusseau, 2020]), respectively. The first map $f_{\mathrm{TT}(R)} : \mathbb{R}^{d_1 \times \cdots \times d_p} \to \mathbb{R}^m$ with $\left[f_{\mathrm{TT}(R)}(\mathcal{X})\right]_i := \frac{1}{\sqrt{m}} \langle\langle \mathcal{G}_i^1, \mathcal{G}_i^2, \ldots, \mathcal{G}_i^p \rangle\rangle, \mathcal{X} \rangle$ for $i \in [m]$, where $\mathcal{G}_i^1 \in \mathbb{R}^{1 \times d_1 \times R}$, $\mathcal{G}_i^2 \in \mathbb{R}^{R \times d_2 \times R}$, ..., $\mathcal{G}_i^{p-1} \in \mathbb{R}^{R \times d_{p-1} \times R}$, $\mathcal{G}_i^p \in \mathbb{R}^{R \times d_p \times 1}$, with entries drawn independently from Gaussian distributions with mean 0 and variance $\frac{1}{\sqrt{R}}$ if $n \in \{1, p\}$ and variance $\frac{1}{R}$ if $1 < n < p$. The second map $f_{\mathrm{CP}(R)} : \mathbb{R}^{d_1 \times \cdots \times d_p} \to \mathbb{R}^m$ with $\left[f_{\mathrm{CP}(R)}(\mathcal{X})\right]_i := \frac{1}{\sqrt{m}} \langle [\![\mathbf{A}_i^1, \mathbf{A}_i^2, \ldots, \mathbf{A}_i^p]\!], \mathcal{X} \rangle$ for $i \in [m]$, where each $\mathbf{A}_i^n \in \mathbb{R}^{d_n \times R}$ has entries drawn independently from a Gaussian distribution with mean 0 and variance $(1/R)^{\frac{1}{p}}$. For both random projection maps, the inner product between the input tensor and random tensor provides an unbiased estimation of the ℓ_2 norm of the original tensor. However, variances of their estimators grow exponentially with the number of modes p [Rakhshan and Rabusseau, 2020], restricting their applicability for tensors with large modes.

In this work, we address this challenge and propose dimensionality reduction for the tensor datasets that overcomes the above-mentioned limitations. Our sketching methodology draws inspiration from the work of [Ahle et al., 2020]*, who introduced an efficient method for sketching high-dimensional polynomial kernels by combining `CountSketch` [Charikar et al., 2004] with `TensorSketch` [Pham and Pagh, 2013]. Their sketching has shown a significant impact in large-scale applications such as leverage score computation [Bharadwaj et al., 2023, Woodruff and Zandieh, 2022], and efficient polynomial kernel sketching for LLM optimization [Kacham et al., 2024]. To the best of our knowledge, while Recursive TensorSketch has been explored for tensor compression (e.g., [Malik, 2022]), our work is the first to employ a recursive sketch specifically for oblivious random projection purposes. Our key contribution is showing the adaptation of the `Recursive TensorSketch` [Ahle et al., 2020] and proposing compression algorithms for tensor datasets, by exploiting the inherent Kronecker structure present in both CP and TT decompositions. Through careful theoretical analysis and algorithmic design, we show that our proposals yield an asymptotically better time complexity and variance bounds.

*The construction presented in [Ahle et al., 2020] was originally referred to as Oblivious Sketch, in our work, we refer it as the `Recursive TensorSketch`.

Our Contributions. We propose two compression techniques, stated in Definitions 4.1, 4.2, for tensor datasets that take as input an mode p tensor $\mathcal{Z} \in \mathbb{R}^{d \times \cdots \times d \ (p \text{ times})}$ and output a one-dimensional vector $\Phi(\mathcal{Z}) \in \mathbb{R}^m$ satisfying $\mathbb{E}\left[\|\Phi(\mathcal{Z})\|_2^2\right] = \|\mathcal{Z}\|_F^2$ and $\mathrm{Var}\left(\|\Phi(\mathcal{Z})\|_2^2\right) \leq \frac{3}{m}\|\mathcal{Z}\|_F^4$. The first technique applies when $\mathcal{Z}$ is given in CP format of rank R, sketching it in low dimension with total time complexity of $O\left(Rp(\Delta_{CP} + m \log m)\right)$, where Δ_{CP} denotes the maximum number of non-zero entries in any column across all factor matrices (Definition 1). The second technique applies when $\mathcal{Z}$ is given in Tensor-Train format with TT rank $(R_0, R_1, \ldots, R_p)$, sketching it in low dimension with total time complexity of $O\left(\widetilde{R}p(\Delta_{TT} + m \log m)\right)$ where $\widetilde{R} = \prod_{k=0}^{p} R_k$, and Δ_{TT} stated in Definition 2. Both methods require sketching dimension $m \geq 3\varepsilon^{-2}/\delta$, independent of p and $R/\widetilde{R}$.

Compared to work of [Rakhshan and Rabusseau, 2020], who embed $\mathcal{Z}$ in $O(m\,p\,d\,\max(R, R')^2)$ time for CP and $O(m\,p\,d\,\max(R, R')^3)$ for TT, and achieve variance bounds $\mathrm{Var}\|f_{CP(R')}(\mathcal{Z})\|^2 \leq m^{-1}\left(3^{p-1}(1+2/R')-1\right)\|\mathcal{Z}\|_F^4$ and $\mathrm{Var}\|f_{TT(R')}(\mathcal{Z})\|^2 \leq m^{-1}\left(3(1+2/R')^{p-1}-1\right)\|\mathcal{Z}\|_F^4$. Our proposals remove the exponential dependence on mode p in the variance expression and simultaneously provide faster compression algorithms (Table 1, Theorem 4).

Organization of the paper: Section 2 reviews related work and positions our method relative to existing baselines. Section 3 introduces the notation and background required for our analysis. In Section 4, we present the proposed sketching algorithm and its theoretical guarantees, with proof sketches provided in Section 5. Experimental results and comparisons are reported in Section 6. Finally, we conclude and outline open questions in the Section 7.

2 RELATED WORK

Classical dimensionality reduction for vectors relies on the JL lemma (*a.k.a. Random projection*) [Johnson and Lindenstrauss, 1984], which guarantees that any set of n points in $\mathbb{R}^D$ can be sketched into $\mathbb{R}^k$ with distortion $1 \pm \varepsilon$ provided $k = O(\varepsilon^{-2} \log n)$ in $O(kD)$ time. There are improved variants of random projections, such as very sparse random projections [Achlioptas, 2003, Li et al., 2006], Fast JL Transform [Ailon and Chazelle, 2009], and `CountSketch` algorithm [Charikar et al., 2004]. However, for mode p tensor $\mathcal{X} \in \mathbb{R}^{d \times \cdots \times d \ (p \text{ times})}$, the time complexity of these approaches is $O(d^p)$ in the worst case, making them computationally expensive for large p.

Kar and Karnick [Kar and Karnick, 2012] introduced

Table 1: Comparison of algorithms in terms of variance and sketching time. Here, $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d\ (p \text{ times})}$ denotes a mode p tensor whose CP rank is R, and TT rank is $(R_0, R_1, \ldots, R_p)$, we denote $\widetilde{R} = \prod_{k=0}^{p} R_k$. $m \in \mathbb{N}$ is the sketching dimension. The parameter R' specifies the mappings $f_{\mathrm{CP}(R')}(\mathcal{Z})$ and $f_{\mathrm{TT}(R')}(\mathcal{Z})$. $\Delta_{CP} \leq d$ and $\Delta_{TT} \leq d$ are defined in Equation (1), and (2), respectively.

Algorithm	Variance	Time Complexity
`CountSketch` [Charikar et al., 2004]	$\frac{2}{m}\left(\|\mathcal{Z}\|_F^4 - \sum_{k=1}^{p}\sum_{i_k=1}^{d} \mathcal{Z}_{i_1,\ldots,i_p}^4\right)$	$O(\mathrm{nnz}(\mathcal{Z}))$
$f_{\mathrm{CP}(R')}(\mathcal{Z})$ [Rakhshan and Rabusseau, 2020]	$\frac{1}{m}\left(3^{p-1}\left(1+\frac{2}{R'}\right)-1\right)\|\mathcal{Z}\|_F^4$	$O(mpd\max(R,R')^2)$
$f_{\mathrm{TT}(R')}(\mathcal{Z})$ [Rakhshan and Rabusseau, 2020]	$\frac{1}{m}\left(3\left(1+\frac{2}{R'}\right)^{p-1}-1\right)\|\mathcal{Z}\|_F^4$	$O(mpd\max(R,R')^3)$
Our work (CP)	$\frac{3}{m}\|\mathcal{Z}\|_F^4$	$O\left(Rp\left(\Delta_{CP}+m\log m\right)\right)$
Our work (TT)	$\frac{3}{m}\|\mathcal{Z}\|_F^4$	$O\left(\widetilde{R}p\left(\Delta_{TT}+m\log m\right)\right)$

random feature mappings for dot-product kernels. Building on this direction, Pham and Pagh proposed `TensorSketch` [Pham and Pagh, 2013], a sketching method for high-degree polynomial kernels, and demonstrated its effectiveness in scaling kernel support vector machines. Their construction extends Pagh's compressed matrix multiplication framework [Pagh, 2013]. Since then, `TensorSketch` has been employed in a variety of linear-algebraic kernel approximation tasks, including PCA, principal component regression, and CCA [Avron et al., 2014]. Let $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^d, i \in [p]$, and let us denote $\mathbf{x} := \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \ldots \mathbf{x}_p$ and $\mathbf{y} := \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \ldots \mathbf{y}_p$. Then `TensorSketch` (stated in Algorithm 1 of [Pham and Pagh, 2025]) $\mathrm{TS} : \mathbb{R}^{d^p} \mapsto \mathbb{R}^m$ with the guarantee that inner product between $\mathrm{TS}(\mathbf{x})$ and $\mathrm{TS}(\mathbf{y})$ gives an unbiased estimation to the inner product between $\mathbf{x}$ and $\mathbf{y}$. With appropriate reshaping, the vector $\mathbf{x} \in \mathbb{R}^{d^p}$ can be considered as a rank one tensor (say) $\mathcal{X} \in \mathbb{R}^{d\times\cdots\times d\ (p \text{ times})}$. As `TensorSketch` is a linear sketch, it can also be used for compressing a tensor $\mathcal{Z}$ which is given as the sum of R rank-one tensors. However, due to Theorem 9 of [Pham and Pagh, 2025], the variance of the corresponding estimator obtained from `TensorSketch` grows exponentially with the number of modes p, making them impractical for tensors with large modes. Moreover, it remains unclear how the `TensorSketch` can be used to give a sketch for tensors given in TT decomposition format.

As mentioned above, Tensorized random projections [Rakhshan and Rabusseau, 2020] give compression algorithms that compress an input tensor $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d\ (p \text{ times})}$ that closely approximates the ℓ_2 norm of the input tensor and the pairwise inner product. However, to compute m-sized sketch of the input tensor, the running of their algorithms $f_{CP(R')}$ and $f_{TT(R')}$ are $O(mp\,d\,\max(R,R')^2)$ and $O(mp\,d\,\max(R,R')^3)$ when input is given in CP and TT decomposition format, respectively. Moreover, the variances of their estimates grow exponentially in the number of modes p. In contrast, the variances of our proposed methods are asymptotically smaller and, importantly, independent of p. For the CP case, our algorithm achieves an asymptotic improvement in time complexity, which is further supported by our experimental results. For the TT case, we achieve an asymptotic improvement in time complexity under the condition $\widetilde{R} = o(R^3)$, where R denotes the CP rank of input tensor $\mathcal{X}$, which also has TT rank $(R_0, R_1, \ldots, R_p)$, and $\widetilde{R} := \prod_{k=0}^{p} R_k$. Furthermore, the two rank formats are related by $R \leq \widetilde{R}$, as established in Lemma 2. Empirically, we also observed improvements in time complexity for the TT case that are comparable to those in our CP setting.

Sun *et al.* [Sun et al., 2021] introduce Tensor Random Projection (TRP), which constructs sketches for mode-2 tensors (matrices) via row-wise Kronecker products of independent `CountSketch` matrices. Their approach can be seen as a special case of the proposal of [Rakhshan and Rabusseau, 2020] for mode two tensors (*i.e.* matrices). Even for the $p = 2$, their compression bounds are weaker than that of [Rakhshan and Rabusseau, 2020]. Jin *et al.* [Jin et al., 2019] extend the fast JL transform [Ailon and Chazelle, 2009] to compress vectors that have Kronecker structure (which can be viewed as a rank one tensor). They show that their algorithm requires $m = \Omega\left(\varepsilon^{-2}\log^{2p-1} n \log d^p\right)$ to achieve $(1\pm\varepsilon)$ approximation to the ℓ_2 norm of input (rank-one) tensor. In contrast, our approach generalizes for rank R tensors, and the compression bound is independent of mode p. However, in terms of probability parameter δ, our compression bound depends on $1/\delta$, whereas theirs depends on $\log(1/\delta)$. We summarize the comparison between baselines and our work in Table 1.

3 PRELIMINARIES

Notations: We use the following notation throughout the paper, vectors are represented by bold lowercase letters (*e.g.* $\mathbf{x}, \mathbf{y}$), matrices by bold uppercase

letters (*e.g.* $\mathbf{X}, \mathbf{Y}$), and higher mode tensors by bold calligraphic letters (*e.g.* $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}$). For a positive integer m, we denote by $[m] := \{1, 2, \ldots, m\}$ the corresponding index set. For any input vector $\mathbf{x}$, we denote by $\mathrm{nnz}(\mathbf{x})$ the number of its nonzero entries. The symbol $\otimes$ denotes the Kronecker (tensor) product, while the dot "$\cdot$" indicates standard matrix multiplication. The outer product of vectors is denoted by "$\circ$". We use the shorthand notation $\langle\!\langle \cdot \rangle\!\rangle$ to represent a tensor expressed in *Tensor Train (TT)* decomposition form, and $[\![\cdot]\!]$ to represent a tensor expressed in *CANDECOMP/PARAFAC (CP)* decomposition form. The indicator function is denoted by $\mathbb{1}[\cdot]$.

Definition 3.1 (JL Transform [Johnson and Lindenstrauss, 1984])**.** *A JL transform is a randomized linear map $\mathbf{S} : \mathbb{R}^d \to \mathbb{R}^k$ with $k \ll d$ such that, for any $0 < \varepsilon < 1$ and any finite set $\mathcal{X} \subset \mathbb{R}^d$, it holds with high probability that $(1-\varepsilon)\|\mathbf{x}-\mathbf{y}\|_2^2 \le \|\mathbf{S}\mathbf{x}-\mathbf{S}\mathbf{y}\|_2^2 \le (1+\varepsilon)\|\mathbf{x}-\mathbf{y}\|_2^2$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, provided that $k = \mathcal{O}(\varepsilon^{-2}\log|\mathcal{X}|)$.*

Definition 3.2 (*CANDECOMP/PARAFAC* (CP) Decomposition [Ballard, 2025])**.** *Let $\mathcal{Z} \in \mathbb{R}^{d\times d\times\cdots\times d\ (p\ times)}$ be a mode p tensor. A rank-R CP decomposition of $\mathcal{Z}$ consists of p factor matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(p)}$, where each $\mathbf{A}^{(k)} \in \mathbb{R}^{d\times R}$, $\forall\ k \in [p]$. The r-th column of $\mathbf{A}^{(k)}$ is denoted by $\mathbf{a}_{:,r}^{(k)} \in \mathbb{R}^d$. Using these factor matrices, the tensor $\mathcal{Z}$ is expressed as a sum of R rank-one tensors*

$$\mathcal{Z} = \sum_{r=1}^{R} \mathbf{a}_{:,r}^{(1)} \circ \mathbf{a}_{:,r}^{(2)} \circ \cdots \circ \mathbf{a}_{:,r}^{(p)},$$

where $\circ$ denotes the outer product. In compact notation, $\mathcal{Z}$ can be written as $\mathcal{Z} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(p)}]\!]$.

Equivalently, each entry of $\mathcal{Z}$ can be written as

$$\mathcal{Z}(i_1, \ldots, i_p) = \sum_{r=1}^{R} \prod_{k=1}^{p} a_{i_k r}^{(k)}, \quad \forall\ i_k \in [d],\ k \in [p].$$

The smallest integer R for which such a decomposition exists is called the CP rank *of $\mathcal{Z}$. We further denote*

$$\Delta_{\mathrm{CP}} := \max_{k\in[p],\ r\in[R]} \mathrm{nnz}\big(\mathbf{a}_{:,r}^{(k)}\big) \ \le\ d, \tag{1}$$

which denotes the maximum number of non-zero entries in any column across all factor matrices.

Definition 3.3 (Tensor Train (TT) Decomposition [Oseledets, 2011])**.** *Let $\mathcal{Z} \in \mathbb{R}^{d\times d\times\cdots\times d\ (p\ times)}$ be a mode p tensor. A* Tensor Train (TT) decomposition *of $\mathcal{Z}$ represents it as a product of mode 3 core tensors,*

$$\mathcal{Z} = \langle\!\langle \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(p)} \rangle\!\rangle,$$

where each core tensor is given by $\mathcal{G}^{(k)} \in \mathbb{R}^{R_{k-1}\times d\times R_k}\ \forall\ k \in [p]$, and $R_0 = R_p = 1$. The tuple $(R_0, R_1, \ldots, R_p)$ is referred as TT rank of tensor $\mathcal{Z}$. Further, each entry of tensor $\mathcal{Z}$ is computed as

$$\mathcal{Z}(i_1, i_2, \ldots, i_p) = \mathcal{G}^{(1)}(:, i_1, :)\mathcal{G}^{(2)}(:, i_2, :)\cdots\mathcal{G}^{(p)}(:, i_p, :)$$

where the product is a sequence of matrix multiplications along the TT-rank dimensions. The term $\mathcal{G}^{(k)}(:, i_k, :) \in \mathbb{R}^{R_{k-1}\times R_k}$ denotes the matrix obtained by fixing the second mode of the k-th core tensor to index i_k, and slicing over the first and third modes. We also denote

$$\Delta_{TT} := \max_{\substack{k\in[p],\ a\in[R_{k-1}],\\ b\in[R_k]}} \mathrm{nnz}\Big(\mathcal{G}_{a,:,b}^{(k)}\Big) \le d, \tag{2}$$

i.e., the maximum number of non-zero entries across all $\mathcal{G}_{a,:,b}^{(k)}$ vectors, which corresponds to the densest vector among the core tensors. $\mathcal{G}_{a,:,b}^{(k)} \in \mathbb{R}^d$ denotes the corresponding vector of $\mathcal{G}^{(k)}$, where first and third indices are fixed and take values a and b, respectively.

Definition 3.4 (`CountSketch` [Charikar et al., 2004])**.** *Given an input vector $\mathbf{y} \in \mathbb{R}^d$, the* ***`CountSketch`*** *is a randomized linear map $\mathbf{T} \in \mathbb{R}^{m\times d}$ that maps $\mathbf{y}$ to a lower-dimensional vector $\mathbf{z} = \mathbf{T}\mathbf{y} \in \mathbb{R}^m$. The* ***`CountSketch`*** *matrix $\mathbf{T}$ is constructed by two hash functions: (a) h: $[d] \to [m]$ a 3-wise independent hash function, and (b) $s : [d] \to \{1, -1\}$ a 4-wise independent random sign function. The j^{th} entry of vector $\mathbf{z} \in \mathbb{R}^m$ is computed as, $z_j = \sum_{h(i)=j} s(i)\, y_i$, $\forall j = \{1, \ldots, m\}$. The time complexity of computing the* ***`CountSketch`*** *is $O(\mathrm{nnz}(\mathbf{y}))$, which in the worst case can be $O(d)$.*

Definition 3.5 (`TensorSketch` of Degree Two [Pham and Pagh, 2013, Pham and Pagh, 2025])**.** *Let $h_1, h_2 : [d] \to [m]$ be 3-wise independent hash functions, and $\sigma_1, \sigma_2 : [d] \to \{-1, +1\}$ be 4-wise independent random sign functions. Then the* ***`TensorSketch`*** *of degree two $\mathbf{S} \in \mathbb{R}^{m\times d^2}$ is defined $\forall\ r \in [m]$, $i_1, i_2 \in [d]$, as follows*

$$S_{r,(i_1,i_2)} = \sigma_1(i_1)\cdot\sigma_2(i_2) \cdot \mathbb{1}\left[h_1(i_1) + h_2(i_2) \equiv r \pmod m\right]. \tag{3}$$

For all $\mathbf{x} \in \mathbb{R}^d$, Let $\mathbf{S}(\mathbf{x}\otimes\mathbf{x})$ can be computed in time $\mathcal{O}(m\log m + \mathrm{nnz}(\mathbf{x}))$ using the Fast Fourier Transform.

Definition 3.6 (`Recursive TensorSketch` [Ahle et al., 2020])**.** *Given a vector $\mathbf{z} \in \mathbb{R}^{d^p}$ where p takes values that are powers of two, then the* ***`Recursive TensorSketch`*** *is a randomized linear map*

$\Pi^p : \mathbb{R}^{d^p} \to \mathbb{R}^m$, *defined as,* $\Pi^p := Q^p \cdot T^p$, *where*

- $T^p = T_1 \otimes T_2 \otimes \cdots \otimes T_p$, *with each* $T_i \in \mathbb{R}^{m\times d}$ $\forall i \in [p]$ *a* ***`CountSketch`*** *matrix (Definition 3.4),*

- $Q^p = S^2 \cdot S^4 \cdots S^{p/2} \cdot S^p$, *with each* $S^\ell \in \mathbb{R}^{m^{\ell/2} \times m^\ell}$ *a Kronecker product of matrices* $S^\ell_j \in \mathbb{R}^{m \times m^2}$,
- *Each* S^ℓ_j *is a* `TensorSketch` *matrix of degree* 2 *(Definition 3.5), and* $S^\ell = S^\ell_1 \otimes S^\ell_2 \otimes \cdots \otimes S^\ell_{\ell/2}$.

The time complexity required to compute the `Recursive TensorSketch` *for* $\mathbf{z} \in \mathbb{R}^{d^p}$ *is* $O(pm \log m + p\Delta_{CP})$ *where* Δ_{CP} *is stated in Equation (1).*

4 SKETCHING ALGORITHMS FOR CP AND TT TENSOR DECOMPOSITIONS

In this section, we give sketching algorithms for tensor datasets corresponding to CP and TT decompositions, when p is a power of two. In Subsection 4.3, we state the theoretical guarantee of the sketching algorithms, and their proofs are discussed in Section 5.

4.1 Sketching algorithm for CP Decomposition

Definition 4.1. *Let* $\mathcal{Y}_r \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})}$ *be a CP-rank-1 mode* p *tensor which is constructed such as*

$$\mathcal{Y}_r = \mathbf{a}^{(1)}_{:,r} \circ \mathbf{a}^{(2)}_{:,r} \circ \cdots \circ \mathbf{a}^{(p)}_{:,r} \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})},$$

where $\circ$ *denotes the outer product and* $\mathbf{a}^{(i)}_{:,r} \in \mathbb{R}^d \ \forall i \in [p]$ *as stated in Definition 3.2. Let* $\mathcal{Z} \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})}$ *be a tensor of CP rank* R*. Suppose it is constructed as a sum of* R *rank-one tensors* $\mathcal{Y}_r$*, i.e.,*

$$\text{vec}(\mathcal{Z}) = \sum_{r=1}^{R} \mathbf{y}_r, \tag{4}$$

where $\mathbf{y}_r := \text{vec}(\mathcal{Y}_r)$ *denotes the vectorization of the tensor* $\mathcal{Y}_r$*. Then we define the sketching map* $\Phi : \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})} \to \mathbb{R}^m$ *such that*

$$\Phi(\mathcal{Z}) := \Pi^p \, \text{vec}(\mathcal{Z}), \tag{5}$$

where, $\Pi^p \in \mathbb{R}^{m \times d^p}$ *denote the* `Recursive TensorSketch` *of order* p *as stated in Definition 3.6.*

Remark 1. *For a CP rank-*1 *tensor, the time complexity of sketching map stated in Definition 4.1 is* $O(pm \log m + p\Delta_{\text{CP}})$ *time, where* Δ_{CP} *is stated in Equation (1). Extending to a CP rank-*R *tensor requires applying the sketch to each of the* R *components, resulting in a total time complexity of* $O\left(Rp(\Delta_{\text{CP}} + m \log m)\right)$.

4.2 Sketching algorithm for TT Decomposition

For a mode p tensor $\mathcal{Z}$, the standard approach is to represent its TT decomposition as a product of p core tensors, each of mode 3 (Definition 3.3). The lemma below facilitates expressing $\mathcal{Z}$ as a sum of Kronecker products of p vectors. Each vector in the Kronecker product is obtained by taking suitable fibres of the mode 3 core tensors obtained in the TT composition of $\mathcal{Z}$. This result enables applying the `Recursive TensorSketch` for the tensor represented in the TT decomposition. See Appendix A.1 for the proof.

Lemma 2. *Let* $\mathcal{Z} \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})}$ *be a mode* p *tensor, represented in the TT decomposition with TT-rank* $(R_0, R_1, \ldots, R_p)$*, where* $R_0 = R_p = 1$*. The decomposition is given by core tensors* $\mathcal{G}^{(k)} \in \mathbb{R}^{R_{k-1} \times d \times R_k}$ *for* $k = 1, \ldots, p$ *as stated in Definition 3.3. Then the vectorized form* $\mathbf{z} := \text{vec}(\mathcal{Z}) \in \mathbb{R}^{d^p}$ *can be written as*

$$\mathbf{z} = \sum_{r=1}^{\widetilde{R}} \mathcal{G}^{(1)}_{1,:,r_1} \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \cdots \cdots \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \mathcal{G}^{(p)}_{r_{p-1},:,1}, \tag{6}$$

where $\widetilde{R} := \prod_{k=0}^{p} R_k$*,* $\mathcal{G}^{(k)}_{a,:,b} \in \mathbb{R}^d$ *denotes the vector obtained by fixing the first index to* a *and the third index to* b*, and* $r = 1 + \sum_{k=1}^{p-1}(r_k - 1)\prod_{j=1}^{k-1} R_j$.

Definition 4.2. *Let* $\mathcal{Z} \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})}$ *be a mode* p *tensor represented in TT decomposition rank* $(R_0, R_1, \ldots, R_p)$ *with TT core tensors* $\mathcal{G}^{(k)} \in \mathbb{R}^{R_{k-1} \times d \times R_k}$ *for* $k = 1, \ldots, p$*, and* $R_0 = R_p = 1$*. Let* $\mathbf{y}_r \in \mathbb{R}^{d^p}$ *be a vector which is constructed such as*

$$\mathbf{y}_r = \mathcal{G}^{(1)}_{1,:,r_1} \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \cdots \cdots \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \mathcal{G}^{(p)}_{r_{p-1},:,1} \in \mathbb{R}^{d^p}, \tag{7}$$

where $\mathcal{G}^{(k)}_{r_{k-1},:,r_k} \in \mathbb{R}^d, \forall\ k \in [p]$ *as stated in Lemma 2. We construct* $\text{vec}(\mathcal{Z})$ *using Lemma 2 as,*

$$\text{vec}(\mathcal{Z}) = \sum_{r=1}^{\widetilde{R}} \mathbf{y}_r. \tag{8}$$

where $\widetilde{R} := \prod_{k=0}^{p} R_k$*. Then we define the sketching map* $\Phi : \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})} \to \mathbb{R}^m$ *such that*

$$\Phi(\mathcal{Z}) := \Pi^p \, \text{vec}(\mathcal{Z}), \tag{9}$$

where, $\Pi^p \in \mathbb{R}^{m \times d^p}$ *denote the* `Recursive TensorSketch` *of mode* p *as stated in Definition 3.6.*

Remark 3. *For a vector* $\mathbf{y}_r \in \mathbb{R}^{d^p}$*, the running time of the sketching map stated in Definition 4.2 is* $O(pm \log m + p\Delta_{\text{TT}})$ *time, where* Δ_{TT} *is stated in Equation (2). For a tensor* $\mathcal{Z}$ *with TT rank* $(R_0, R_1, \ldots, R_p)$*, it requires applying the sketch to each of the* $\widetilde{R}$ *components, resulting in a total time complexity of* $O\left(\widetilde{R}p(\Delta_{\text{TT}} + m \log m)\right)$.

4.3 Theoretical Guarantee

Theorem 4. *Let tensor $\mathcal{Z} \in \mathbb{R}^{d\times d\times\cdots\times d \ (p \text{ times})}$ be a mode p tensor (in CP/TT Decomposition), then the map $\Phi : \mathbb{R}^{d\times d\times\cdots\times d \ (p \text{ times})} \to \mathbb{R}^m$ such that,*

$$\Phi(\mathcal{Z}) = \Pi^p \operatorname{vec}(\mathcal{Z}), \tag{10}$$

where, Φ is stated in Definition 4.1, 4.2 for tensor in CP and TT decomposition, respectively. Then, for any tensor $\mathcal{Z} \in \mathbb{R}^{d\times d\times\cdots\times d \ (p \text{ times})}$, its sketched vector $\Phi(\mathcal{Z}) \in \mathbb{R}^m$ satisfies the following properties

$$\mathbb{E}\left[\|\Phi(\mathcal{Z})\|_2^2\right] = \|\mathcal{Z}\|_F^2, \tag{11}$$

$$\operatorname{Var}\left[\|\Phi(\mathcal{Z})\|_2^2\right] \le \frac{3}{m}\|\mathcal{Z}\|_F^4. \tag{12}$$

Using these bounds, we establish concentration guarantees for Φ under CP and TT decompositions. See Appendix A.2 for the proof.

Theorem 5. *Let $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d \ (p \text{ times})}$ be a mode p tensor, and let Φ denote the sketching map stated in Definitions 4.1 and 4.2 for the CP and TT decompositions, respectively. For any $\varepsilon, \delta \in (0,1)$, if $m \ge \frac{3}{\varepsilon^2\,\delta}$, then, with probability at least $1-\delta$, the following concentration guarantee holds*

$$\left\|\Phi(\mathcal{Z})\right\|_2^2 \;=\; (1\pm\varepsilon)\,\|\mathcal{Z}\|_F^2,$$

irrespective of whether $\mathcal{Z}$ is in CP or TT representation.

Remark 6. *Our sketching algorithms described in Definitions 4.1, 4.2 remain applicable even when the input tensor $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d \ (p \text{ times})}$ is not given in CP/TT decomposition form. In that case, to apply sketching map Φ, we require explicitly constructing the mapping Π^p (Definition 3.6) as a matrix of size $m \times d^p$. Although the guarantee stated in Theorem 4 still holds, the time complexity of sketching becomes $O(md^p)$. The advantage in time complexity is achieved when the input tensor is represented in CP or TT format (Remark 1, 3).*

5 PROOFS OF THEORETICAL GUARANTEES AND SUPPORTING LEMMAS

We present the detailed proofs of the theoretical guarantees stated in Section 4.3, together with the supporting lemmas required for the analysis. Furthermore, we extend our sketching map Φ to arbitrary tensor of mode p (including non-power-of-two cases) in Section 5.3.

5.1 Technical Lemmas

In this subsection, we present a set of auxiliary lemmas that formalize the moment properties of the `Recursive TensorSketch` matrix as stated in Definition 3.6. Specifically, we establish bounds on its first, second, and fourth moments and unbiased estimate of the inner product. These results further allow us to bound the covariance of sketched inner products, which will be crucial for proving the main approximation guarantees in the subsequent sections. See Appendix B.1 for the proofs.

Lemma 7. *[Moments of the `Recursive TensorSketch`] Let $\Pi^p \in \mathbb{R}^{m\times d^p}$ denote the `Recursive TensorSketch` matrix as stated in Definition 3.6. Then, for all $i \in [m]$ and $j \in [d^p]$,*

$$\mathbb{E}\left[\Pi_{ij}^p\right] = 0, \tag{13}$$

$$\mathbb{E}\left[\left(\Pi_{ij}^p\right)^2\right] = \frac{1}{m}, \tag{14}$$

$$\mathbb{E}\left[(\Pi_{ij}^p)^4\right] = \frac{1}{m} + \frac{3}{m^{2p}}\left(\prod_{i=1}^{p/2}(m^{2i}-1)\right). \tag{15}$$

We now show that `Recursive TensorSketch` preserves inner products in expectation, and the subsequent lemma analyze the covariance between different sketched inner products.

Lemma 8. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{d^p}$ and let $\Pi^p \in \mathbb{R}^{m\times d^p}$ denote the `Recursive TensorSketch` matrix as stated in Definition 3.6 Then,*

$$\mathbb{E}\left[\langle \Pi^p\mathbf{x}, \Pi^p\mathbf{y}\rangle\right] = \langle \mathbf{x}, \mathbf{y}\rangle. \tag{16}$$

Lemma 9. *Let $\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^{d^p}$ and let $\Pi^p \in \mathbb{R}^{m\times d^p}$ be `Recursive TensorSketch` as stated in Definition 3.6. Then,*

$$\begin{aligned}
&\operatorname{Cov}\left[\langle \Pi^p\mathbf{x}, \Pi^p\mathbf{y}\rangle, \langle \Pi^p\mathbf{w}, \Pi^p\mathbf{z}\rangle\right] \\
&= \left[\frac{3}{m^{2p-1}}\left(\prod_{i=1}^{p/2}(m^{2i}-1)\right) - \frac{2}{m}\right]\sum_{j=1}^{d^p} x_j y_j w_j z_j + \\
&\qquad + \frac{1}{m}\left[\langle \mathbf{x}, \mathbf{w}\rangle\langle \mathbf{y}, \mathbf{z}\rangle + \langle \mathbf{x}, \mathbf{z}\rangle\langle \mathbf{y}, \mathbf{w}\rangle\right], \\
&\le \frac{1}{m}\left[\langle \mathbf{x}, \mathbf{w}\rangle\langle \mathbf{y}, \mathbf{z}\rangle + \langle \mathbf{x}, \mathbf{z}\rangle\langle \mathbf{y}, \mathbf{w}\rangle + \sum_{j=1}^{d^p} x_j y_j w_j z_j\right].
\end{aligned} \tag{17}$$

5.2 Proof of Theorem 4

Proof. We now derive bounds on the expectation and variance of the squared ℓ_2-norm of the sketched vector $\Phi(\mathcal{Z})$, where $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d \ (p \text{ times})}$ is a mode p tensor (given in CP/TT Decomposition). The proof proceeds by expanding the $\operatorname{vec}(\mathcal{Z})$ into its components $\mathbf{y}_r$ as stated in Definition 4.1, 4.2. Then by applying the unbiasedness and covariance properties of the `Recursive`

`TensorSketch`, using Lemma 8 and Lemma 9, respectively. Applying the sketching map Φ on $\mathcal{Z}$,

$$\Phi(\mathcal{Z}) := \Pi^p \operatorname{vec}(\mathcal{Z}) \tag{18}$$

The vectorization of $\mathcal{Z}$ can be written as $\operatorname{vec}(\mathcal{Z}) = \sum_r \mathbf{y}_r$, with the parameters depending on the decomposition: for CP, $r \in [R]$ where R is the CP rank and $\mathbf{y}_r := \operatorname{vec}(\mathcal{Y}_r)$ as in Definition 4.1; for TT, $r \in [\widetilde{R}]$ where $\widetilde{R}$ and $\mathbf{y}_r$ are given in Definition 4.2.

To generalize both case suppose $\mathbf{z} := \operatorname{vec}(\mathcal{Z}) = \sum_{r=1}^{\bar{R}} \mathbf{y}_r$,where $\bar{R} = R$ for CP and $\bar{R} = \widetilde{R}$ for TT. We first analyze the expectation of $\|\Phi(\mathcal{Z})|_2^2$

$$\mathbb{E}\left[\|\Phi(\mathcal{Z})\|_2^2\right] = \mathbb{E}[\langle \Phi(\mathcal{Z}), \Phi(\mathcal{Z})\rangle], \tag{19}$$

$$= \sum_{r_1,r_2=1}^{\bar{R}} \mathbb{E}[\langle \Pi^p \mathbf{y}_{r_1}, \Pi^p \mathbf{y}_{r_2}\rangle], \tag{20}$$

$$= \sum_{r_1,r_2=1}^{\bar{R}} \langle \mathbf{y}_{r_1}, \mathbf{y}_{r_2}\rangle = \|\mathbf{z}\|_2^2 = \|\mathcal{Z}\|_F^2. \tag{21}$$

Equation (19) expands $\Phi(\mathcal{Z})$ via Equation (18) and $\mathbf{z} = \sum_{r=1}^{\bar{R}} \mathbf{y}_r$. Equation (20) follows from bilinearity and linearity of expectation. Then by using Lemma 8 we get Equation (21).We now analyze the variance

$$\operatorname{Var}\left[\|\Phi(\mathcal{Z})\|_2^2\right] = \operatorname{Var}[\langle \Phi(\mathcal{Z}), \Phi(\mathcal{Z})\rangle], \tag{22}$$

$$= \operatorname{Var}\left[\sum_{r_1,r_2=1}^{\bar{R}} \langle \Pi^p \mathbf{y}_{r_1}, \Pi^p \mathbf{y}_{r_2}\rangle\right],$$

$$= \sum_{\substack{r_1,r_2,\\ r_3,r_4=1}}^{\bar{R}} \operatorname{Cov}\Big[\langle \Pi^p \mathbf{y}_{r_1}, \Pi^p \mathbf{y}_{r_2}\rangle, \langle \Pi^p \mathbf{y}_{r_3}, \Pi^p \mathbf{y}_{r_4}\rangle\Big]. \tag{23}$$

Applying Lemma 9 to Equation (23), and letting $[y_r]_q$ denote the q-th entry of $\mathbf{y}_r$, we obtain

$$\leq \frac{1}{m} \sum_{\substack{r_1,r_2,\\ r_3,r_4=1}}^{\bar{R}} \Big[\langle \mathbf{y}_{r_1}, \mathbf{y}_{r_2}\rangle\langle \mathbf{y}_{r_3}, \mathbf{y}_{r_4}\rangle + \langle \mathbf{y}_{r_1}, \mathbf{y}_{r_4}\rangle\langle \mathbf{y}_{r_3}, \mathbf{y}_{r_2}\rangle\Big]$$

$$+ \frac{1}{m} \sum_{\substack{r_1,r_2,\\ r_3,r_4=1}}^{\bar{R}} \sum_{q=1}^{d^p} [y_{r_1}]_q [y_{r_2}]_q [y_{r_3}]_q [y_{r_4}]_q, \tag{24}$$

$$= \frac{2}{m}\left(\sum_{r_1,r_2=1}^{\bar{R}} \langle \mathbf{y}_{r_1}, \mathbf{y}_{r_2}\rangle\right)^2 + \frac{1}{m}\sum_{q=1}^{d^p}\left(\sum_{r=1}^{\bar{R}} [y_r]_q\right)^4,$$

$$= \frac{2}{m}\left\langle \sum_{r=1}^{\bar{R}} \mathbf{y}_r, \sum_{r=1}^{\bar{R}} \mathbf{y}_r\right\rangle^2 + \frac{1}{m}\sum_{q=1}^{d^p} (z_q)^4, \tag{25}$$

$$= \frac{2}{m}\|\mathbf{z}\|_2^4 + \frac{1}{m}\sum_{q=1}^{d^p} (z_q)^4. \tag{26}$$

$$\leq \frac{3}{m}\|\mathbf{z}\|_2^4 = \frac{3}{m}\|\mathcal{Z}\|_F^4. \tag{27}$$

Using $\sum_{r=1}^{\bar{R}} [y_r]_q = z_q$ gives Equation (25). Finally, applying Cauchy–Schwarz to the second term in Equation (26) yields the bound in Equation (27).

□

5.3 Generalized sketching map Φ for any mode p Tensor

For non-power-of-two p, below Equation (28) defines the sketching map Φ, which is shown to satisfy both unbiasedness and the covariance bound. The proof is deferred to Appendix B.2.

Theorem 10. *[Unbiasedness and Covariance for Non-Power-of-Two p] Let $\mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d}$ (p times) be a mode p tensor with $p \geq 2$, where p is not a power of two. Set $q := 2^{\lceil \log_2 p\rceil}$ and define*

$$\Phi(\mathcal{Z}) := \Pi^q\Big(\operatorname{vec}(\mathcal{Z}) \otimes \mathbf{e}_1^{\otimes(q-p)}\Big), \tag{28}$$

where $\mathbf{e}_1 \in \mathbb{R}^d$ is the first standard basis vector and Π^q is the power-of-two `Recursive TensorSketch` from Definition 3.6.

Then, for all $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{d\times\cdots\times d}$ (p times),

$$\mathbb{E}[\langle \Phi(\mathcal{X}), \Phi(\mathcal{Y})\rangle] = \langle \mathcal{X}, \mathcal{Y}\rangle,$$

and for all $\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathcal{Z} \in \mathbb{R}^{d\times\cdots\times d}$ (p times),

$$\operatorname{Cov}(\langle \Phi(\mathcal{X}), \Phi(\mathcal{Y})\rangle, \langle \Phi(\mathcal{W}), \Phi(\mathcal{Z})\rangle)$$

$$\leq \tfrac{1}{m}\Big(\langle \mathcal{X}, \mathcal{W}\rangle\langle \mathcal{Y}, \mathcal{Z}\rangle + \langle \mathcal{X}, \mathcal{Z}\rangle\langle \mathcal{Y}, \mathcal{W}\rangle + \ldots$$

$$+ \sum_{k=1}^{p}\sum_{i_k=1}^{d} \mathcal{X}_{i_1,\ldots,i_p}\mathcal{Y}_{i_1,\ldots,i_p}\mathcal{W}_{i_1,\ldots,i_p}\mathcal{Z}_{i_1,\ldots,i_p}\Big). \tag{29}$$

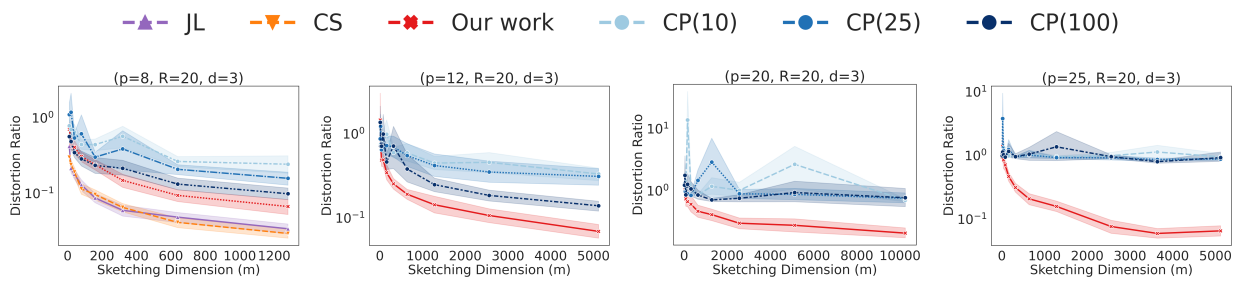

Figure 1: Distortion ratio comparison between our work and baselines for varying tensor mode (p) in the CP representation. CP(R') refers to tensorized random projection, while CS refers to `CountSketch`. The tuple (p, R, d) denote mode, CP rank, and dimension along each mode of the input tensor.

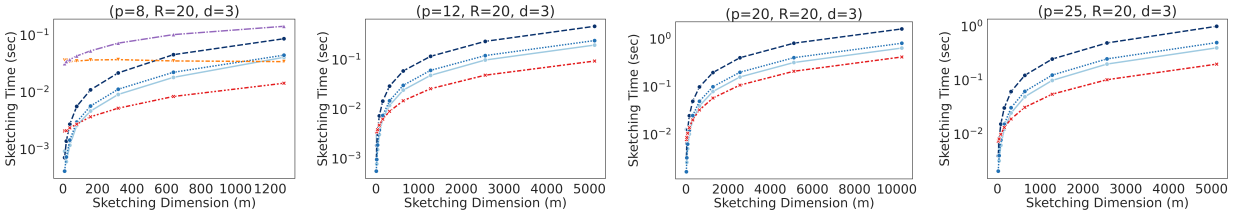

Figure 2: Sketching time comparison of our work with baselines for varying tensor mode (p) in CP representation.

6 EXPERIMENTS

We compare our map (Φ) with TRP [Rakhshan and Rabusseau, 2020] (using the authors' public implementation [†]), JL [Johnson and Lindenstrauss, 1984], and `CountSketch` [Charikar et al., 2004] (both re-implemented by us following the original algorithmic descriptions). We further study how the input tensor's rank (R) and mode (p) affect the results. All experiments were performed on Ubuntu 22.04.4 with an Intel® Core™ i9-14900K processor (24 cores, 32 threads) and 32 GB RAM.

Datasets. We consider both synthetic and real-world datasets. For the synthetic setting, we generate a random $p = 8, 12, 20, 25$ mode, $d = 3$ dimensional tensor $\mathcal{Z}$ (*i.e.*, a vector of size d^p) with CP-rank R, normalized to have unit Frobenius norm. For real-world benchmarks, we use: (i) **CIFAR-10** [Krizhevsky, 2009], a fourth-order image dataset of size $(60000, 32, 32, 3)$; (ii) **COIL-100** [Nene et al., 1996], a fourth-order image tensor of size $(7200, 128, 128, 3)$.

Metrics. For a tensor $\mathcal{Z}$ and a sketching map $f()$, we report the **distortion ratio**, $D(f, \mathcal{Z}) = \left\| \frac{\|f(\mathcal{Z})\|^2}{\|\mathcal{Z}\|^2} - 1 \right\|$, measuring relative norm error. A smaller distortion ratio indicates higher accuracy. We also capture **sketching time**, which denotes the time (in seconds) required to compute the sketch of the input tensor. All results are averaged over 100 independent runs.

Insights. In the main paper, we present results for input tensors provided in CP decomposition format. We begin by analyzing the trade-off between distortion ratio and sketching time across tensors with varying modes, CP ranks, and dimensions. Figure 1 highlights the robustness of our method: it consistently achieves low distortion across different tensor modes (p). In contrast, competing methods fail to maintain this stability. For instance, Tensorized projections show errors that grow rapidly—exponentially with p and inversely with the CP-rank (R') of the auxiliary random tensor used in sketching. Classical approaches such as JL and `CountSketch` perform better for very low-mode tensors (e.g., $p = 8$). However, since both require explicit construction of the full tensor, they become infeasible for larger modes ($p = 12, 20, 25$) and cannot be directly applied to compress the input tensor. In contrast, our compression algorithm is applicable to higher mode tensors represented in CP decomposition form, while simultaneously providing a desired accuracy.

The runtime analysis in Figure 2 demonstrates the efficiency of our method. It delivers both lower distortion ratio and fast sketching time scaling linearly in mode p tensor and CP-rank R, and only superlinearly in sketch dimension (m). No baseline achieves similar scalability. Tensorized projections, in contrast, show quadratic to cubic growth with their parameter R' and R, making them increasingly inefficient for larger ten-

[†] https://github.com/brakhshan/Tensorized_Random_Projection

sor ranks. JL and `CountSketch` exhibit exponential dependence on p (see Table 1). While they can handle small tensor modes (e.g., $p = 8$), they fail for higher modes ($p = 12, 20, 25$) because they require explicit tensor materialization. Overall, these findings establish our method as the most practical, offering the best balance of distortion, runtime, and scalability, while directly supporting high-order CP/TT tensor formats. We obtained a similar performance on both metrics - distortion ratio and sketching time for the tensors given in TT decomposition format as well as real-world datasets. We discuss these results in Appendix C .

7 CONCLUSION

We propose two sketching algorithms for tensors represented in CP and TT decomposition formats that closely approximate the ℓ_2 norm and pairwise inner product of the input tensor. Our sketching algorithms use techniques from [Ahle et al., 2020], which proposed an efficient sketching algorithm for high-dimensional polynomial kernels. The variance bounds of existing baselines [Rakhshan and Rabusseau, 2020, Pham and Pagh, 2025] grow exponentially with the number of modes p, whereas the variance of our approach is independent of p. Further, our algorithm is asymptotically faster than these baselines. This study leaves several open questions for future research. To achieve $(1 \pm \varepsilon)$ approximation, the dependency of our approach on the probability parameter δ is proportional to $1/\delta$. Improving this dependency δ by employing better concentration analysis is a major open question of this work. Further, integrating recent advances of improved recursive sketching variants [Ahle et al., 2020], particularly from [Song et al., 2021], could further reduce randomness and computational complexities of the sketch, which could be another interesting direction to pursue on.

Acknowledgements

This research was supported by the ANRF under Project No. ANRF/ECRG/2024/001063/ENS. We gratefully acknowledge ANRF for this support.

References

[Achlioptas, 2003] Achlioptas, D. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687.

[Ahle et al., 2020] Ahle, T. D., Kapralov, M., Knudsen, J. B. T., Pagh, R., Velingker, A., Woodruff, D. P., and Zandieh, A. (2020). Oblivious sketching of high-degree polynomial kernels. In *31st Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, January 5-8, 2020, Salt Lake City, Utah, USA, Proceedings*, pages 141–160. SIAM.

[Ailon and Chazelle, 2009] Ailon, N. and Chazelle, B. (2009). The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322.

[Avron et al., 2014] Avron, H., Nguyen, H. L., and Woodruff, D. P. (2014). Subspace embeddings for the polynomial kernel. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2258–2266.

[Bachrach et al., 2014] Bachrach, Y., Finkelstein, Y., Gilad-Bachrach, R., Katzir, L., Koenigstein, N., Nice, N., and Paquet, U. (2014). Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 257–264.

[Ballard, 2025] Ballard, G. (2025). Tensor methods textbook (draft). Accessed: September 7, 2025.

[Ballard and Kolda, 2025] Ballard, G. and Kolda, T. G. (2025). *Tensor Decompositions for Data Science*. Cambridge University Press.

[Bharadwaj et al., 2023] Bharadwaj, V., Malik, O. A., Murray, R., Grigori, L., Buluç, A., and Demmel, J. (2023). Fast exact leverage score sampling from khatri-rao products with applications to tensor decomposition. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

[Charikar et al., 2004] Charikar, M., Chen, K. C., and Farach-Colton, M. (2004). Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15.

[DataStax, 2025] DataStax (2025). Exploring the real-world applications of cosine similarity. https://www.datastax.com/guides/real-world-applications-of-cosine-similarity.

[Hitchcock, 1927] Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

[Jin et al., 2019] Jin, R., Kolda, T. G., and Ward, R. A. (2019). Faster johnson-lindenstrauss transforms via kronecker products. *CoRR*, abs/1909.04801.

[Johnson and Lindenstrauss, 1984] Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206.

[Kacham et al., 2024] Kacham, P., Mirrokni, V., and Zhong, P. (2024). Polysketchformer: Fast transformers via sketching polynomial kernels. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

[Kar and Karnick, 2012] Kar, P. and Karnick, H. (2012). Random feature maps for dot product kernels. In Lawrence, N. D. and Girolami, M. A., editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 583–591. JMLR.org.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

[Krizhevsky, 2009] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. pages 32–33.

[Li et al., 2017] Li, H., Nam, C., Yiu, M., and Mamoulis, N. (2017). Fexipro: Fast and exact inner product retrieval in recommender systems. pages 835–850.

[Li et al., 2006] Li, P., Hastie, T., and Church, K. W. (2006). Very sparse random projections. In Eliassi-Rad, T., Ungar, L. H., Craven, M., and Gunopulos, D., editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 287–296. ACM.

[Luo et al., 2018] Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., and Yang, Q. (2018). Cosine normalization: Using cosine similarity instead of dot product in neural networks. In Kurková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. S., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I*, volume 11139 of *Lecture Notes in Computer Science*, pages 382–391. Springer.

[Malik, 2022] Malik, O. A. (2022). More efficient sampling for tensor decomposition with worst-case guarantees. 162:14887–14917.

[Memgraph Team, 2023] Memgraph Team (2023). Understanding cosine similarity in python with scikit-learn. https://memgraph.com/blog/cosine-similarity-python-scikit-learn.

[Nene et al., 1996] Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (coil-100). Technical report, Technical report CUCS-006-96.

[Nickel and Rol, 2013] Nickel, M. and Rol, E. (2013). Python scikit-tensor toolbox: a python module for multilinear algebra and tensor factorizations. https://github.com/evertrol/scikit-tensor-py3. Accessed: 2025-10-10.

[Oseledets, 2011] Oseledets, I. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317.

[Oseledets et al., 2014] Oseledets, I., Kazeev, V., et al. (2014). Python tt-toolbox2.2: Fast multidimensional array operations in tt-format. https://github.com/oseledets/ttpy. Available online, June 2014. Accessed: 2025-10-10.

[Pagh, 2013] Pagh, R. (2013). Compressed matrix multiplication. *ACM Trans. Comput. Theory*, 5(3):9:1–9:17.

[Pham and Pagh, 2013] Pham, N. and Pagh, R. (2013). Fast and scalable polynomial kernels via explicit feature maps. In Dhillon, I. S., Koren, Y., Ghani, R., Senator, T. E., Bradley, P., Parekh, R., He, J., Grossman, R. L., and Uthurusamy, R., editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 239–247. ACM.

[Pham and Pagh, 2025] Pham, N. and Pagh, R. (2025). Tensor sketch: Fast and scalable polynomial kernel approximation. *CoRR*, abs/2505.08146.

[Rakhshan and Rabusseau, 2020] Rakhshan, B. T. and Rabusseau, G. (2020). Tensorized random projections. In Chiappa, S. and Calandra, R., editors,

The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy], volume 108 of *Proceedings of Machine Learning Research*, pages 3306–3316. PMLR.

[Song et al., 2021] Song, Z., Woodruff, D., Yu, Z., and Zhang, L. (2021). Fast sketching of polynomial kernels of polynomial degree. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9812–9823. PMLR.

[Sun et al., 2021] Sun, Y., Guo, Y., Tropp, J. A., and Udell, M. (2021). Tensor random projection for low memory dimension reduction. *CoRR*, abs/2105.00105.

[Tucker, 1966] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[Woodruff and Zandieh, 2022] Woodruff, D. P. and Zandieh, A. (2022). Leverage score sampling for tensor product matrices in input sparsity time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24218–24244. PMLR.

[Xiang et al., 2019] Xiang, L., Tang, B., and Yang, C. (2019). Accelerating exact inner product retrieval by cpu-gpu systems. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1277–1280, New York, NY, USA. Association for Computing Machinery.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   Yes. Please refer Subsections 4.1 and 4.2.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Yes. Please refer Subsections 4.1, 4.2 and 4.3.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes. A zip file containing anonymized source code is included with the supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Yes. Please refer Section - 4
   (b) Complete proofs of all theoretical results.
   Yes. Please refer Section - 5
   (c) Clear explanations of any assumptions.
   Yes. We have stated our assumptions regarding the input tensor format, specifying it should be either CP or TT, in Remark 6.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   Yes. It is provided in the supplemental material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Not Applicable. As our experiments involve randomized sketching methods, not model training.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   Yes. Evaluation metrics are defined in Section 6 with results averaged over 100 runs and error bars showing standard deviation.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   Yes. It is mentioned in Section 6, experiments ran on Ubuntu 22.04.4 with Intel® Core™ i9-14900K (24 cores, 32 threads) and 32 GB RAM.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets.
   Yes. We cite the implementation of the baseline [Rakhshan and Rabusseau, 2020], and mention all dataset sources in Section 6.
   (b) The license information of the assets, if applicable.
   Not Applicable. All assets are publicly available for research without license restrictions.
   (c) New assets either in the supplemental material or as a URL, if applicable.
   Not Applicable. We do not release new assets beyond our implementation.

(d) Information about consent from data providers/curators.
Not Applicable. All datasets are public benchmarks that do not require consent.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
No. The datasets used contain no sensitive or offensive content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots.
Not Applicable.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
Not Applicable.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
Not Applicable.

Supplementary Material for: Efficient and Accurate Tensor Compression via Recursive Sketching

A Missing Proofs from Section 4

A.1 Missing proof of Subsection 4.2 (Proof of Lemma 2)

In this subsection, we present a detailed proof of characterization of the vectorization of a tensor represented in TT Decomposition as a summation of Kronecker products of vectors derived from the TT cores. The following lemma formalizes this representation.

Lemma 2. *Let $\mathcal{Z} \in \mathbb{R}^{d \times d \times \cdots \times d \ (p \text{ times})}$ be a mode p tensor, represented in the TT decomposition with TT-rank $(R_0, R_1, \ldots, R_p)$, where $R_0 = R_p = 1$. The decomposition is given by core tensors $\mathcal{G}^{(k)} \in \mathbb{R}^{R_{k-1} \times d \times R_k}$ for $k = 1, \ldots, p$ as stated in Definition 3.3. Then the vectorized form $\mathbf{z} := \mathrm{vec}(\mathcal{Z}) \in \mathbb{R}^{d^p}$ can be written as*

$$\mathbf{z} = \sum_{r=1}^{\widetilde{R}} \mathcal{G}^{(1)}_{1,:,r_1} \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \cdots\cdots \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \mathcal{G}^{(p)}_{r_{p-1},:,1}, \tag{30}$$

where $\widetilde{R} := \prod_{k=0}^{p} R_k$, $\mathcal{G}^{(k)}_{a,:,b} \in \mathbb{R}^d$ denotes the vector obtained by fixing the first index to a and the third index to b, and $r = 1 + \sum_{k=1}^{p-1}(r_k - 1)\prod_{j=1}^{k-1} R_j$.

Proof. We first express the i-th entry of the vector $\mathbf{z}$ as

$$z_i = \mathcal{Z}(i_1, i_2, \ldots, i_p), \tag{31}$$

where the linear index i corresponds to the multi-index $(i_1, \ldots, i_p)$ via the standard column-major mapping

$$i = 1 + \sum_{k=1}^{p} (i_k - 1) d^{k-1}. \tag{32}$$

According to the TT decomposition, the entry $\mathcal{Z}(i_1, i_2, \ldots, i_p)$ can be expressed as

$$z_i = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{p-1}=1}^{R_{p-1}} \left[\mathcal{G}^{(1)}(1, i_1, r_1) \cdot \mathcal{G}^{(2)}(r_1, i_2, r_2) \cdot \cdots \cdot \mathcal{G}^{(p)}(r_{p-1}, i_p, 1) \right]. \tag{33}$$

On the other hand, the i^{th} element of the following Kronecker product

$$\mathcal{G}^{(p)}_{r_{p-1},:,1} \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \cdots \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \mathcal{G}^{(1)}_{1,:,r_1},$$

is given by

$$\mathcal{G}^{(p)}(r_{p-1}, i_p, 1) \cdot \mathcal{G}^{(p-1)}(r_{p-2}, i_{p-1}, r_{p-1}) \cdot \cdots \cdot \mathcal{G}^{(2)}(r_1, i_2, r_2) \cdot \mathcal{G}^{(1)}(1, i_1, r_1), \tag{34}$$

Now take summation over all TT-rank indices then

$$\sum_{r_1=1}^{R_1} \cdots \sum_{r_{p-1}=1}^{R_{p-1}} \left[\mathcal{G}^{(p)}(r_{p-1}, i_p, 1) \cdots \mathcal{G}^{(k)}(r_{k-1}, i_k, r_k) \cdots \mathcal{G}^{(1)}(1, i_1, r_1) \right]. \tag{35}$$

Comparing Equations (33) and (35), we conclude

$$\mathbf{z} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{p-1}=1}^{R_{p-1}} \Big(\mathcal{G}^{(p)}_{r_{p-1},:,1} \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \cdots \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \mathcal{G}^{(1)}_{1,:,r_1} \Big). \tag{36}$$

Above equation can also be expressed by combining the $p-1$ index tuple as the single index r, then

$$\mathbf{z} = \sum_{r=1}^{\widetilde{R}} \mathcal{G}^{(1)}_{1,:,r_1} \otimes \mathcal{G}^{(2)}_{r_1,:,r_2} \otimes \cdots \otimes \mathcal{G}^{(p-1)}_{r_{p-2},:,r_{p-1}} \otimes \mathcal{G}^{(p)}_{r_{p-1},:,1}, \tag{37}$$

where,

$$r = 1 + \sum_{k=1}^{p-1} (r_k - 1) \prod_{j=1}^{k-1} R_j.$$

□

A.2 Missing Proof from Subsection 4.3 (Proof of Theorem 5)

Building on the expectation and variance bounds of $\Phi(\mathcal{Z})$ established in Theorem 4, we derive concentration guarantees for the `Recursive TensorSketch` (RTS). Our analysis applies to tensors represented in both CP and TT decompositions.

Theorem 5. *Let $\mathcal{Z} \in \mathbb{R}^{d \times \cdots \times d \ (p \text{ times})}$ be a mode p tensor, and let Φ denote the sketching map stated in Definitions 4.1 and 4.2 for the CP and TT decompositions, respectively. For any $\varepsilon, \delta \in (0,1)$, if $m \geq \frac{3}{\varepsilon^2 \delta}$, then, with probability at least $1-\delta$, the following concentration guarantee holds*

$$\big\|\Phi(\mathcal{Z})\big\|_2^2 \;=\; (1 \pm \varepsilon)\, \|\mathcal{Z}\|_F^2,$$

irrespective of whether $\mathcal{Z}$ is in CP or TT representation.

Proof. We first prove the result for $\mathcal{Z}$ represented in CP decomposition. From Theorem 4, we have

$$\mathbb{E}\big[\|\Phi(\mathcal{Z})\|_2^2\big] = \|\mathcal{Z}\|_F^2, \qquad \mathrm{Var}\big(\|\Phi(\mathcal{Z})\|_2^2\big) \leq \frac{3}{m} \|\mathcal{Z}\|_F^4.$$

Applying Chebyshev's inequality gives

$$\Pr\Big[\big|\|\Phi(\mathcal{Z})\|_2^2 - \mathbb{E}[\|\Phi(\mathcal{Z})\|_2^2]\big| \geq t\Big] \leq \frac{\mathrm{Var}(\|\Phi(\mathcal{Z})\|_2^2)}{t^2}. \tag{38}$$

Choosing $t = \varepsilon \|\mathcal{Z}\|_F^2$ yields

$$\Pr\Big[\big|\|\Phi(\mathcal{Z})\|_2^2 - \|\mathcal{Z}\|_F^2\big| \geq \varepsilon \|\mathcal{Z}\|_F^2\Big] \leq \frac{\frac{3}{m}\|\mathcal{Z}\|_F^4}{\varepsilon^2 \|\mathcal{Z}\|_F^4} = \frac{3}{m\varepsilon^2}. \tag{39}$$

Requiring the above probability to be at most δ gives

$$\frac{3}{m\varepsilon^2} \leq \delta \quad \Longrightarrow \quad m \geq \frac{3}{\varepsilon^2\, \delta}. \tag{40}$$

Thus, with $m \geq 3/(\varepsilon^2\delta)$, we have

$$\Pr\Big[\big|\|\Phi(\mathcal{Z})\|_2^2 - \|\mathcal{Z}\|_F^2\big| < \varepsilon \|\mathcal{Z}\|_F^2\Big] \geq 1 - \delta. \tag{41}$$

Equivalently,

$$\Pr\Big[(1-\varepsilon)\|\mathcal{Z}\|_F^2 \leq \|\Phi(\mathcal{Z})\|_2^2 \leq (1+\varepsilon)\|\mathcal{Z}\|_F^2\Big] \geq 1 - \delta. \tag{42}$$

The proof for $\mathcal{Z}$ represented in TT decomposition follows analogously, since the same expectation and variance bounds hold for TT representation as well. □

B Missing Proofs from Section 5

B.1 Missing Proof of Technical Lemmas from Subsection 5.1

In this section, we provide detailed proofs of the auxiliary lemmas stated in Subsection 5.1. These lemmas characterize the moment properties of the `Recursive TensorSketch` matrix introduced in Definition 3.6. In particular, we derive bounds on the first, second, and fourth moments, and establish that the sketch provides an unbiased estimate of the inner product. These results are subsequently used to bound the covariance of sketched inner products, which is a key step in proving the main approximation guarantees presented in Subsection 5.2. We define the multi-index

$$\mathbf{l} := \left(l_1^2, l_2^2, \ldots, l_1^4, \ldots, l_4^4, \ldots, l_1^{p/2}, \ldots, l_{p/2}^{p/2}, \ldots, l_1^p, \ldots, l_p^p\right) \in [m]^p,$$

where m denotes the embedding dimension, p is the order of the `Recursive TensorSketch`, and $[m]^p$ denotes the p-fold cartesian product of $[m] := \{1, 2, \ldots, m\}$. Thus, $\mathbf{l}$ represents a tuple enumerating all index combinations appearing in the construction of the `Recursive TensorSketch`.

Using this notation, the summation over all index tuples can be written compactly as

$$\sum_{\substack{l_1^2, l_2^2 \\ l_1^4, \ldots, l_4^4 \\ \vdots \\ l_1^{p/2}, \ldots, l_{p/2}^{p/2} \\ l_1^p, \ldots, l_p^p}} = \sum_{\mathbf{l} \in [m]^p}. \tag{43}$$

This convention will be used throughout the subsequent lemmas to keep the notation concise.

Lemma 7. *[Moments of the `Recursive TensorSketch`] Let $\Pi^p \in \mathbb{R}^{m \times d^p}$ denote the `Recursive TensorSketch` matrix as stated in Definition 3.6. Then, for all $i \in [m]$ and $j \in [d^p]$,*

$$\mathbb{E}\left[\Pi_{ij}^p\right] = 0, \tag{13}$$

$$\mathbb{E}\left[\left(\Pi_{ij}^p\right)^2\right] = \frac{1}{m}, \tag{14}$$

$$\mathbb{E}\left[(\Pi_{ij}^p)^4\right] = \frac{1}{m} + \frac{3}{m^{2p}}\left(\prod_{i=1}^{p/2}(m^{2i} - 1)\right). \tag{15}$$

Proof. The entries of `Recursive TensorSketch` can be constructed as follows

$$\Pi^p = Q^p \cdot T^p \tag{44}$$

By Definition 3.6 we can write

$$\begin{aligned}
\Pi^p &= Q^{p/2}\left(S_1^p \otimes S_2^p \otimes \cdots \otimes S_{p/2-1}^p \otimes S_{p/2}^p\right)(T_1 \otimes T_2 \otimes \cdots \otimes T_{p-1} \otimes T_p) \\
&\vdots \\
&= S_1^2\Bigg[S_1^4\Big(\cdots S_1^{p/2}\left(S_1^p(T_1 \otimes T_2) \otimes S_2^p(T_3 \otimes T_4)\right)\Big) \otimes \cdots \\
&\qquad \cdots \otimes S_2^4\Big(\cdots S_{p/4}^{p/2}\left(S_{p/2-1}^p(T_{p-3} \otimes T_{p-2}) \otimes S_{p/2}^p(T_{p-1} \otimes T_p)\right)\Big)\Bigg]
\end{aligned} \tag{45}$$

We know that each T_p corresponds to a `CountSketch` matrix and each S_a^ℓ represents a `TensorSketch` transformation. Consequently, we may express the entry as follows

For each `CountSketch` matrix T_p, the j-th entry is

$$(T_p)_j = \sum_{i=1}^{d} \sigma_{T_p}(i) z_{ij} \tag{46}$$

Where, Let mapping $h : [d] \to [m]$ to map features into k buckets and also let mapping $\sigma_{T_p} : [d] \to \{-1, +1\}$ for each feature. To capture feature contributions, we introduce an indicator random variable z_{ij}

$$z_{ij} = \begin{cases} 1, & \text{if } h(i) = j, \\ 0, & \text{otherwise.} \end{cases}$$

For each `TensorSketch` matrix S_a^ℓ, the k-th entry is

$$(S_a^\ell)_k = \sum_{i=1}^{m} \sum_{j=1}^{m} \phi_{S_a^\ell}(i,j) \Upsilon_{ij}^k \tag{47}$$

Where, Let mapping $h_1, h_2 : [m] \to [m]$ and also let mapping $\phi_1, \phi_2 : [m] \to \{-1, 1\}$, so we can write

- $H(i,j) = [h_1(i) + h_2(j)] \bmod \mathrm{m}$
- $\phi_{S_a^\ell}(i,j) = \phi_1(i)\phi_2(j)$

we can define indicator variable Υ_{ij}^k as :-

$$\Upsilon_{ij}^k = \begin{cases} 1, & \text{if } H(i,j) = k, \\ 0, & \text{otherwise.} \end{cases}$$

By substituting the expressions from Equations (46) and (47) into the recursive formulation (45), the (i,j)-th entry of Π^p can be written explicitly as follows

$$\Pi_{ij}^p = \sum_{\mathbf{l} \in [m]^p}^{m} \left(\sigma_{T_1}(j_1) \cdots \sigma_{T_p}(j_p) \prod_{k=1}^{p} z_{j_k l_k^p} \right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q} \left(l_{2a-1}^q, l_{2a}^q \right) \Upsilon_{l_{2a-1}^q l_{2a}^q}^{l_a^{q/2}} \right) \phi_{S_1^2}(l_1^2, l_2^2) \Upsilon_{l_1^2 l_2^2}^i, \tag{48}$$

where the lexicographic index j is given by

$$j = 1 + \sum_{i=1}^{p} (j_i - 1) d^{i-1},$$

First Moment:

$$\mathbb{E}\left[\Pi_{ij}^p\right] = \mathbb{E}\left[\sum_{\mathbf{l} \in [m]^p} \left(\sigma_{T_1}(j_1) \cdots \sigma_{T_p}(j_p) \prod_{k=1}^{p} z_{j_k l_k^p} \right) \prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q} \left(l_{2a-1}^q, l_{2a}^q \right) \Upsilon_{l_{2a-1}^q l_{2a}^q}^{l_a^{q/2}} \phi_{S_1^2}(l_1^2, l_2^2) \Upsilon_{l_1^2 l_2^2}^i \right], \tag{49}$$

We observe that for every term in the sum, the product $\sigma_{T_1}(j_1) \cdots \sigma_{T_p}(p)$ involves the random signs generated independently with $\mathbb{E}[\sigma_{T_i}(i)] = 0$ for each $i \in [p]$. Therefore, we can write

$$\mathbb{E}\left[\Pi_{ij}^p\right] = 0. \tag{50}$$

Second Moment:

Let $Z = \left(\Pi_{ij}^p\right)^2$. We can decompose Z as $Z = Z_1 + Z_2$, where Z_1 is the diagonal part and Z_2 is the cross (non-diagonal) part, as detailed below:

$$Z = \left(\Pi_{ij}^p\right)^2 \tag{51}$$

$$= \left(\sum_{\mathbf{l} \in [m]^p} \left(\sigma_{T_1}(j_1) \cdots \sigma_{T_p}(j_p) \prod_{k=1}^{p} z_{j_k l_k^p} \right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}(l_{2a-1}^q, l_{2a}^q) \Upsilon_{l_{2a-1}^q l_{2a}^q}^{l_a^{q/2}} \right) \phi_{S_1^2}(l_1^2, l_2^2) \Upsilon_{l_1^2 l_2^2}^i \right)^2 \tag{52}$$

$$= Z_1 + Z_2, \tag{53}$$

where

$$Z_1 = \sum_{\mathbf{l}\in[m]^p} \left(\sigma_{T_1}(j_1)\cdots\sigma_{T_p}(j_p)\prod_{k=1}^{p} z_{j_k l_k^p}\right)^2 \left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)^2 \phi_{S_1^2}(l_1^2, l_2^2)^2(\Upsilon^i_{l_1^2 l_2^2})^2, \quad (54)$$

$$Z_2 = \sum_{\mathbf{l}\neq\mathbf{l}'} \left(\prod_{k=1}^{p} z_{j_k l_k^p}\right)\left(\prod_{k=1}^{p} z_{j_k l_k'^p}\right)\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)\times\cdots$$

$$\cdots\times\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}'^q, l_{2a}'^q\right)\Upsilon^{l_a'^{q/2}}_{l_{2a-1}'^q l_{2a}'^q}\right)\phi_{S_1^2}(l_1^2, l_2^2)\phi_{S_1^2}(l_1'^2, l_2'^2)(\Upsilon^i_{l_1^2 l_2^2})(\Upsilon^i_{l_1'^2 l_2'^2}). \quad (55)$$

Applying expectation, we have,

$$\mathbb{E}[Z_1] = \mathbb{E}\left[\sum_{\mathbf{l}\in[m]^p}\left(\prod_{k=1}^{p}(z_{j_k l_k^p})^2\right)\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}(\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q})^2\right)(\Upsilon^i_{l_1^2 l_2^2})^2\right], \quad (56)$$

$$= \sum_{\mathbf{l}\in[m]^p}\frac{1}{m^p}\times\frac{1}{m^{p-2}}\times\frac{1}{m}, \quad (57)$$

$$= \frac{1}{m}. \quad (58)$$

and

$$\mathbb{E}[Z_2] = \mathbb{E}\Bigg[\sum_{\mathbf{l}\neq\mathbf{l}'}\left(\prod_{k=1}^{p} z_{j_k l_k^p}\right)\left(\prod_{k=1}^{p} z_{j_k l_k'^p}\right)\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)\times\cdots$$

$$\cdots\times\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}'^q, l_{2a}'^q\right)\Upsilon^{l_a'^{q/2}}_{l_{2a-1}'^q l_{2a}'^q}\right)\phi_{S_1^2}(l_1^2, l_2^2)\phi_{S_1^2}(l_1'^2, l_2'^2)(\Upsilon^i_{l_1^2 l_2^2})(\Upsilon^i_{l_1'^2 l_2'^2})\Bigg], \quad (59)$$

$$= 0 \qquad \left(\because\ \mathbb{E}[\phi_{S_a^\ell}(i,j)] = \mathbb{E}[\phi_1(i)\phi_2(j)] = 0 \quad \forall i,j\in[p]\right). \quad (60)$$

Therefore, using Equation (58) for Z_1 and (60) for Z_2,

$$\mathbb{E}\left[\left(\Pi_{ij}^p\right)^2\right] = \mathbb{E}[Z] = \mathbb{E}[Z_1] + \mathbb{E}[Z_2] = \frac{1}{m}. \quad (61)$$

Fourth Moment:

Calculating expectation of 4th moment of Π_{ij}^p,

$$\left(\Pi_{ij}^p\right)^4 = \left(\sum_{\mathbf{l}\in[m]^p}\left(\sigma_{T_1}(j_1)\cdots\sigma_{T_p}(j_p)\prod_{k=1}^{p} z_{j_k l_k^p}\right)\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)\phi_{S_1^2}(l_1^2, l_2^2)\Upsilon^i_{l_1^2 l_2^2}\right)^4, \quad (62)$$

$$= \left(\sum_{\mathbf{l}\in[m]^p}\left(\prod_{k=1}^{p} z_{j_k l_k^p}\right)\left(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)\phi_{S_1^2}(l_1^2, l_2^2)\Upsilon^i_{l_1^2 l_2^2}\right)^4, \quad (63)$$

We know that

$$\left(\sum_n X_n\right)^4 = \sum_n X_n^4 + 4\sum_{n\neq n_1} X_n^3 X_{n_1} + 3\sum_{n\neq n_1} X_n^2 X_{n_1}^2$$
$$+ 6\sum_{n\neq n_1\neq n_2} X_n^2 X_{n_1} X_{n_2} + \sum_{n\neq n_1\neq n_2\neq n_3} X_n X_{n_1} X_{n_2} X_{n_3}. \quad (64)$$

Here, we have

$$X_n = \sum_{\mathbf{l}\in[m]^p} \left(\prod_{k=1}^{p} z_{j_k l_k^p}\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right) \Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right) \phi_{S_1^2}(l_1^2, l_2^2) \Upsilon^i_{l_1^2 l_2^2}, \tag{65}$$

$$X_{n_1} = \sum_{\mathbf{l}'\in[m]^p} \left(\prod_{k=1}^{p} z_{j_k l_k'^p}\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}'^q, l_{2a}'^q\right) \Upsilon^{l_a'^{q/2}}_{l_{2a-1}'^q l_{2a}'^q}\right) \phi_{S_1^2}(l_1'^2, l_2'^2) \Upsilon^i_{l_1'^2 l_2'^2}, \tag{66}$$

$$X_{n_2} = \sum_{\mathbf{l}''\in[m]^p} \left(\prod_{k=1}^{p} z_{j_k l_k''^p}\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}''^q, l_{2a}''^q\right) \Upsilon^{l_a''^{q/2}}_{l_{2a-1}''^q l_{2a}''^q}\right) \phi_{S_1^2}(l_1''^2, l_2''^2) \Upsilon^i_{l_1''^2 l_2''^2}, \tag{67}$$

$$X_{n_3} = \sum_{\mathbf{l}'''\in[m]^p} \left(\prod_{k=1}^{p} z_{j_k l_k'''^p}\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}'''^q, l_{2a}'''^q\right) \Upsilon^{l_a'''^{q/2}}_{l_{2a-1}'''^q l_{2a}'''^q}\right) \phi_{S_1^2}(l_1'''^2, l_2'''^2) \Upsilon^i_{l_1'''^2 l_2'''^2}. \tag{68}$$

Therefore,

$$\mathbb{E}\left[(\Pi_{ij}^p)^4\right] = \mathbb{E}\left[\sum_n X_n^4 + 4\sum_{n\neq n_1} X_n^3 X_{n_1} + 3\sum_{n\neq n_1} X_n^2 X_{n_1}^2 + 6\sum_{n\neq n_1\neq n_2} X_n^2 X_{n_1} X_{n_2} + \sum_{n\neq n_1\neq n_2\neq n_3} X_n X_{n_1} X_{n_2} X_{n_2}\right], \tag{69}$$

$$= \sum_n \mathbb{E}[X_n^4] + 4\sum_{n\neq n_1} \mathbb{E}[X_n^3]\mathbb{E}[X_{n_1}] + 3\sum_{n\neq n_1} \mathbb{E}[X_n^2]\mathbb{E}[X_{n_1}^2] + 6\sum_{n\neq n_1\neq n_2} \mathbb{E}[X_n^2]\mathbb{E}[X_{n_1}]\mathbb{E}[X_{n_2}] + \cdots$$

$$\cdots + \sum_{n\neq n_1\neq n_2\neq n_3} \mathbb{E}[X_n]\mathbb{E}[X_{n_1}]\mathbb{E}[X_{n_2}]\mathbb{E}[X_{n_2}], \tag{70}$$

Odd power X_n has expectation zero because $\mathbb{E}[\sigma_{T_i}(j_i)] = 0$, then we get

$$= \mathbb{E}\left[\sum_n X_n^4\right] + 3\mathbb{E}\left[\sum_{n\neq n_1} X_n^2 X_{n_1}^2\right]. \tag{71}$$

We analyze each term separately as follows:

$$\mathbb{E}\left[\sum_n X_n^4\right] = \mathbb{E}\left[\left(\sum_{\mathbf{l}\in[m]^p} \left(\prod_{k=1}^{p} z_{j_k l_k^p}\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right) \Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right) \phi_{S_1^2}(l_1^2, l_2^2) \Upsilon^i_{l_1^2 l_2^2}\right)^4\right], \tag{72}$$

$$= \mathbb{E}\left[\sum_{\mathbf{l}\in[m]^p} \left(\prod_{k=1}^{p} (z_{j_k l_k^p})^4\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right)^4 (\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q})^4\right) \phi_{S_1^2}(l_1^2, l_2^2)^4 (\Upsilon^i_{l_1^2 l_2^2})^4\right], \tag{73}$$

$$= \sum_{\mathbf{l}\in[m]^p} \left(\prod_{k=1}^{p} \mathbb{E}[z_{j_k l_k^p}]\right) \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \mathbb{E}[\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}]\right) \mathbb{E}[\Upsilon^i_{l_1^2 l_2^2}], \tag{74}$$

$$= \sum_{\mathbf{l}\in[m]^p} \frac{1}{m^p} \times \frac{1}{m^{p-2}} \times \frac{1}{m}, \tag{75}$$

$$= \frac{1}{m}. \tag{76}$$

$$3\mathbb{E}\left[\sum_{n\neq n_1} X_n^2 X_{n_1}^2\right] = 3\mathbb{E}\left[\sum_{\mathbf{l}\neq\mathbf{l}'} \left(\prod_{k=1}^{p} z_{j_k l_k^p}\right)^2 \left(\prod_{k=1}^{p} z_{j_k l_k'^p}\right)^2 \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}^q, l_{2a}^q\right) \Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}\right)^2 \times \cdots\right.$$

$$\left. \cdots \times \left(\prod_{q=4,8,\ldots,p} \prod_{a=1}^{q/2} \phi_{S_a^q}\left(l_{2a-1}'^q, l_{2a}'^q\right) \Upsilon^{l_a'^{q/2}}_{l_{2a-1}'^q l_{2a}'^q}\right)^2 \phi_{S_1^2}(l_1^2, l_2^2)^2 \phi_{S_1^2}(l_1'^2, l_2'^2)^2 (\Upsilon^i_{l_1^2 l_2^2})^2 (\Upsilon^i_{l_1'^2 l_2'^2})^2\right], \tag{77}$$

$$= 3\mathbb{E}\Bigg[\sum_{\mathbf{l}\neq\mathbf{l}'}\Bigg(\prod_{k=1}^{p}(z_{j_k l_k^p})^2\Bigg)\Bigg(\prod_{k=1}^{p}(z_{j_k l_k'^p})^2\Bigg)\Bigg(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}(\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q})^2\Bigg)\times\cdots$$

$$\cdots\times\Bigg(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}(\Upsilon^{l'^{q/2}_a}_{l'^q_{2a-1}l'^q_{2a}})^2\Bigg)(\Upsilon^i_{l_1^2 l_2^2})^2(\Upsilon^i_{l_1'^2 l_2'^2})^2\Bigg], \tag{78}$$

$$= 3\sum_{\mathbf{l}\neq\mathbf{l}'}\Bigg(\prod_{k=1}^{p}\mathbb{E}[z_{j_k l_k^p}]\Bigg)\Bigg(\prod_{k=1}^{p}\mathbb{E}[z_{j_k l_k'^p}]\Bigg)\Bigg(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\mathbb{E}[\Upsilon^{l_a^{q/2}}_{l_{2a-1}^q l_{2a}^q}]\Bigg)\times\cdots$$

$$\cdots\times\Bigg(\prod_{q=4,8,\ldots,p}\prod_{a=1}^{q/2}\mathbb{E}[\Upsilon^{l'^{q/2}_a}_{l'^q_{2a-1}l'^q_{2a}}]\Bigg)\mathbb{E}[\Upsilon^i_{l_1^2 l_2^2}]\mathbb{E}[\Upsilon^i_{l_1'^2 l_2'^2}], \tag{79}$$

$$= 3\sum_{\mathbf{l}\neq\mathbf{l}'}\frac{1}{m^{2p}}\times\frac{1}{m^{2(p-1)}}, \tag{80}$$

$$= 3\left(\prod_{i=1}^{p/2}m^{2i}(m^{2i}-1)\right)\times\frac{1}{m^{4p-2}}, \tag{81}$$

$$= \frac{3}{m^{2p}}\left(\prod_{i=1}^{p/2}(m^{2i}-1)\right). \tag{82}$$

Therefore,

$$\mathbb{E}\left[(\Pi_{ij}^p)^4\right] = \frac{1}{m}+\frac{3}{m^{2p}}\left(\prod_{i=1}^{p/2}(m^{2i}-1)\right). \tag{83}$$

□

After analyzing the moment properties of the `Recursive TensorSketch`, we now show that it preserves inner products in expectation.

Lemma 8. *Let* $\mathbf{x},\mathbf{y}\in\mathbb{R}^{d^p}$ *and let* $\Pi^p\in\mathbb{R}^{m\times d^p}$ *denote the* ***Recursive TensorSketch*** *matrix as stated in Definition 3.6 Then,*

$$\mathbb{E}\left[\langle\Pi^p\mathbf{x},\Pi^p\mathbf{y}\rangle\right] = \langle\mathbf{x},\mathbf{y}\rangle. \tag{16}$$

Proof. Let $\mathbf{x},\mathbf{y}\in\mathbb{R}^{d^p}$ and compute the inner product of $\Pi^p\mathbf{x}$ and $\Pi^p\mathbf{y}$,

$$\langle\Pi^p\mathbf{x},\Pi^p\mathbf{y}\rangle = \sum_{i=1}^{m}\sum_{j=1}^{d^p}\sum_{k=1}^{d^p}(\Pi_{ij}^p x_j)(\Pi_{ik}^p y_k), \tag{84}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{d^p}(\Pi_{ij}^p)^2 x_j y_j + \sum_{i=1}^{m}\sum_{j\neq k}^{d^p}\Pi_{ij}^p x_j \Pi_{ik}^p y_k, \tag{85}$$

Taking the expectation on the both sides,

$$\mathbb{E}\left[\langle\Pi^p\mathbf{x},\Pi^p\mathbf{y}\rangle\right] = \sum_{i=1}^{m}\sum_{j=1}^{d^p}\mathbb{E}\left[(\Pi_{ij}^p)^2\right]x_j y_j + \sum_{i=1}^{m}\sum_{j\neq k}^{d^p}\mathbb{E}[\Pi_{ij}^p]\mathbb{E}[\Pi_{ik}^p]x_j y_k, \tag{86}$$

Using Equation (13), (14) we have $\mathbb{E}\left[\Pi_{ij}^p\right] = 0$, $\mathbb{E}\left[(\Pi_{ij}^p)^2\right] = \frac{1}{m}$ respectively,

$$\mathbb{E}\left[\langle \Pi^p \mathbf{x}, \Pi^p \mathbf{y}\rangle\right] = \sum_{i=1}^{m} \frac{1}{m} \sum_{j=1}^{d^p} x_j y_j, \tag{87}$$

$$= \sum_{j=1}^{d^p} x_j y_j, \tag{88}$$

$$= \langle \mathbf{x}, \mathbf{y}\rangle. \tag{89}$$

□

While Lemma 8 shows that the `Recursive TensorSketch` preserves inner products in expectation, the following result analyze the covariance between different sketched inner products.

Lemma 9. *Let $\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^{d^p}$ and let $\Pi^p \in \mathbb{R}^{m \times d^p}$ be `Recursive TensorSketch` as stated in Definition 3.6. Then,*

$$\mathrm{Cov}\left[\langle \Pi^p \mathbf{x}, \Pi^p \mathbf{y}\rangle, \langle \Pi^p \mathbf{w}, \Pi^p \mathbf{z}\rangle\right] = \left[\frac{3}{m^{2p-1}} \left(\prod_{i=1}^{p/2} (m^{2i} - 1)\right) - \frac{2}{m}\right] \sum_{j=1}^{d^p} x_j y_j w_j z_j \tag{90}$$

$$+ \frac{1}{m}\left[\langle \mathbf{x}, \mathbf{w}\rangle\langle \mathbf{y}, \mathbf{z}\rangle + \langle \mathbf{x}, \mathbf{z}\rangle\langle \mathbf{y}, \mathbf{w}\rangle\right], \tag{91}$$

$$\leq \frac{1}{m}\left[\langle \mathbf{x}, \mathbf{w}\rangle\langle \mathbf{y}, \mathbf{z}\rangle + \langle \mathbf{x}, \mathbf{z}\rangle\langle \mathbf{y}, \mathbf{w}\rangle + \sum_{j=1}^{d^p} x_j y_j w_j z_j\right]. \tag{92}$$

Proof. Let $\Pi^p \in \mathbb{R}^{m \times d^p}$ be `Recursive TensorSketch` , and let $\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^{d^p}$ be fixed vectors. Define

$$Z_1 := \langle \Pi^p \mathbf{x}, \Pi^p \mathbf{y}\rangle, \tag{93}$$

$$Z_2 := \langle \Pi^p \mathbf{w}, \Pi^p \mathbf{z}\rangle. \tag{94}$$

We aim to compute

$$\mathrm{Cov}\left[Z_1, Z_2\right] = \mathbb{E}[Z_1 Z_2] - \mathbb{E}[Z_1]\mathbb{E}[Z_2]. \tag{95}$$

$$\tag{96}$$

We expand Z_1 and Z_2 as,

$$Z_1 = \sum_{i=1}^{m} \left(\sum_{j=1}^{d^p} \Pi_{ij}^p x_j\right) \left(\sum_{k=1}^{d^p} \Pi_{ik}^p y_k\right), \tag{97}$$

$$= \sum_{i=1}^{m} \sum_{j,k=1}^{d^p} \Pi_{ij}^p \Pi_{ik}^p x_j y_k. \tag{98}$$

Similarly,

$$Z_2 = \sum_{i=1}^{m} \sum_{j',k'=1}^{d^p} \Pi_{ij'}^p \Pi_{ik'}^p w_{j'} z_{k'}. \tag{99}$$

Thus,

$$\mathbb{E}[Z_1 Z_2] = \sum_{i,i'=1}^{m} \sum_{j,k,j',k'=1}^{d^p} \mathbb{E}[\Pi_{ij}^p \Pi_{ik}^p \Pi_{i'j'}^p \Pi_{i'k'}^p] x_j y_k w_{j'} z_{k'}. \tag{100}$$

The non-diagonal term of $\mathbb{E}[Z_1 Z_2]$ from Equation (100), corresponding to $i \neq i'$, is non-zero when $j = k$, $j' = k'$, and $j \neq j'$, then,

$$\mathbb{E}[\Pi^p_{ij}\Pi^p_{ik}\Pi^p_{i'j'}\Pi^p_{i'k'}] = \mathbb{E}[(\Pi^p_{ij})^2] \cdot \mathbb{E}[(\Pi^p_{i'j'})^2], \tag{101}$$

$$= \frac{1}{m^2} \ . \tag{102}$$

So, contribution of non-diagonal part is given by,

$$\sum_{i\neq i'}^{m} \sum_{j=k, j'=k', j\neq j'} \mathbb{E}[\Pi^p_{ij}\Pi^p_{ik}\Pi^p_{i'j'}\Pi^p_{i'k'}] x_j y_k w_{j'} z_{k'} = \frac{1}{m^2} \sum_{i\neq i'} \sum_{j\neq j'} x_j y_j w_{j'} z_{j'}, \tag{103}$$

$$= \frac{m(m-1)}{m^2} \sum_{j\neq j'} x_j y_j w_{j'} z_{j'} \ . \tag{104}$$

Now for diagonal term of $\mathbb{E}[Z_1 Z_2]$ from Equation (100), corresponding to $i = i'$ is given by,

$$\sum_{i=1}^{m} \sum_{j,k,j',k'=1}^{d^p} \mathbb{E}[\Pi^p_{ij}\Pi^p_{ik}\Pi^p_{ij'}\Pi^p_{ik'}] x_j y_k w_{j'} z_{k'}. \tag{105}$$

Non-zero contributions occur when

$$\text{(i) } j = k = j' = k' : \quad \mathbb{E}[(\Pi^p_{ij})^4] x_j y_j w_j z_j, \tag{106}$$

$$\text{(ii) } j = j', k = k', j \neq k : \quad \mathbb{E}[(\Pi^p_{ij})^2]\mathbb{E}[(\Pi^p_{ik})^2] x_j w_j y_k z_k, \tag{107}$$

$$\text{(iii) } j = k, j' = k', j \neq j' : \quad \mathbb{E}[(\Pi^p_{ij})^2]\mathbb{E}[(\Pi^p_{ij'})^2] x_j y_j w_{j'} z_{j'}, \tag{108}$$

$$\text{(iv) } j = k', j' = k, j \neq j' : \quad \mathbb{E}[(\Pi^p_{ij})^2]\mathbb{E}[(\Pi^p_{ij'})^2] x_j z_j w_{j'} y_{j'} \ . \tag{109}$$

By using Equation (13) & (14) we have contributions by diagonal term of $\mathbb{E}[Z_1 Z_2]$ as,

$$\sum_{i=1}^{m} \left[\left[\frac{1}{m} + \frac{3}{m^{2p}} \left(\prod_{i=1}^{p/2} (m^{2i} - 1) \right) \right] \sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m^2} \left[\sum_{j\neq k} x_j y_k w_j z_k + \sum_{j\neq j'} (x_j y_j w_{j'} z_{j'} + x_j y_{j'} w_{j'} z_j) \right] \right], \tag{110}$$

$$= \left[1 + \frac{3}{m^{2p-1}} \left(\prod_{i=1}^{p/2} (m^{2i} - 1) \right) \right] \sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m} \left[\sum_{j\neq k} x_j y_k w_j z_k + \sum_{j\neq j'} (x_j y_j w_{j'} z_{j'} + x_j y_{j'} w_{j'} z_j) \right]. \tag{111}$$

By combining the diagonal and non-diagonal terms using Equation (111) and Equation (104), respectively, we obtain

$$\mathbb{E}[Z_1 Z_2] = \langle \mathbf{x}, \mathbf{y} \rangle \langle \mathbf{w}, \mathbf{z} \rangle + \left[\frac{3}{m^{2p-1}} \left(\prod_{i=1}^{p/2} (m^{2i} - 1) \right) \right] \sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m} \left[\sum_{j\neq k} x_j y_k w_j z_k + \sum_{j\neq j'} x_j y_{j'} w_{j'} z_j \right]. \tag{112}$$

Now by using Lemma 8, we get

$$\mathbb{E}[Z_1]\mathbb{E}[Z_2] = \mathbb{E}[\langle \Pi^p \mathbf{x}, \Pi^p \mathbf{y} \rangle] \cdot \mathbb{E}[\langle \Pi^p \mathbf{w}, \Pi^p \mathbf{z} \rangle], \tag{113}$$

$$= \langle \mathbf{x}, \mathbf{y} \rangle \langle \mathbf{w}, \mathbf{z} \rangle \ . \tag{114}$$

Now substitute expressions of $\mathbb{E}[Z_1 Z_2]$ and $\mathbb{E}[Z_1]\mathbb{E}[Z_2]$ using Equation (112) and Equation (114) respectively in

Equation (95), we get

$$\mathrm{Cov}\left[Z_1, Z_2\right] = \mathbb{E}[Z_1 Z_2] - \mathbb{E}[Z_1]\mathbb{E}[Z_2], \tag{115}$$

$$= \left[\frac{3}{m^{2p-1}}\left(\prod_{i=1}^{p/2}(m^{2i}-1)\right) - \frac{2}{m}\right]\sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m}\left[\langle \mathbf{x},\mathbf{w}\rangle\langle \mathbf{y},\mathbf{z}\rangle + \langle \mathbf{x},\mathbf{z}\rangle\langle \mathbf{y},\mathbf{w}\rangle\right], \tag{116}$$

$$\leq \left[\frac{3}{m^{2p-1}}\left(\prod_{i=1}^{p/2}(m^{2i})\right) - \frac{2}{m}\right]\sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m}\left[\langle \mathbf{x},\mathbf{w}\rangle\langle \mathbf{y},\mathbf{z}\rangle + \langle \mathbf{x},\mathbf{z}\rangle\langle \mathbf{y},\mathbf{w}\rangle\right], \tag{117}$$

$$\leq \left(\frac{3}{m} - \frac{2}{m}\right)\sum_{j=1}^{d^p} x_j y_j w_j z_j + \frac{1}{m}\left[\langle \mathbf{x},\mathbf{w}\rangle\langle \mathbf{y},\mathbf{z}\rangle + \langle \mathbf{x},\mathbf{z}\rangle\langle \mathbf{y},\mathbf{w}\rangle\right], \tag{118}$$

$$= \frac{1}{m}\left[\langle \mathbf{x},\mathbf{w}\rangle\langle \mathbf{y},\mathbf{z}\rangle + \langle \mathbf{x},\mathbf{z}\rangle\langle \mathbf{y},\mathbf{w}\rangle + \sum_{j=1}^{d^p} x_j y_j w_j z_j\right]. \tag{119}$$

□

B.2 Missing Proofs from Subsection 5.3

In this subsection, we present the proofs of Theorem 10 together with the auxiliary lemmas required for its derivation. These results establish that our sketch Φ remains unbiased and satisfies the covariance bound for arbitrary tensor modes p, not just powers of two. Here, Lemma 10 shows that padding by e_1 does not alter the inner products, allowing the unbiasedness of the power-of-two sketch Π^q transfer directly to Π^p after padding.And, Lemma 11 characterizes the coordinate mapping under padding, ensuring that sums over d^q entries collapse to the original d^p entries, which is essential to translate the covariance bound from the padded to the unpadded vectors.

Lemma 10. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{d^p}$ for some positive integers $d \geq 1$ and $p \geq 2$. Let $e_1 \in \mathbb{R}^d$ be the first standard basis column vector, i.e.,*

$$\mathbf{e_1} = (1, 0, \ldots, 0)^\top.$$

Set

$$q = \left\lceil 2^{\log_2 p}\right\rceil,$$

and note that $q \geq p$. Then

$$\langle \mathbf{x} \otimes \mathbf{e_1}^{\otimes(q-p)},\ \mathbf{y} \otimes \mathbf{e_1}^{\otimes(q-p)}\rangle \ = \ \langle \mathbf{x}, \mathbf{y}\rangle. \tag{120}$$

Proof. We use the standard property of the Kronecker product, for any vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ of compatible dimensions,

$$\langle \mathbf{a} \otimes \mathbf{b},\ \mathbf{c} \otimes \mathbf{d}\rangle = \langle \mathbf{a}, \mathbf{c}\rangle\,\langle \mathbf{b}, \mathbf{d}\rangle. \tag{121}$$

Applying this iteratively, we get

$$\langle \mathbf{x} \otimes \mathbf{e_1}^{\otimes(q-p)},\ \mathbf{y} \otimes \mathbf{e_1}^{\otimes(q-p)}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle\,\left\langle \mathbf{e_1}^{\otimes(q-p)},\ \mathbf{e_1}^{\otimes(q-p)}\right\rangle. \tag{122}$$

Since $\mathbf{e_1}$ is a unit vector, $\langle \mathbf{e_1}, \mathbf{e_1}\rangle = 1$, and by the tensor product norm property,

$$\left\langle \mathbf{e_1}^{\otimes(q-p)},\ \mathbf{e_1}^{\otimes(q-p)}\right\rangle = (\langle \mathbf{e_1}, \mathbf{e_1}\rangle)^{q-p} = 1^{\,q-p} = 1. \tag{123}$$

Therefore,

$$\langle \mathbf{x} \otimes \mathbf{e_1}^{\otimes(q-p)},\ \mathbf{y} \otimes \mathbf{e_1}^{\otimes(q-p)}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle \cdot 1 = \langle \mathbf{x}, \mathbf{y}\rangle. \tag{124}$$

□

Lemma 11 (Index Relation under Padding using e_1)**.** *Let $\mathbf{x} \in \mathbb{R}^{d^p}$ for integers $d \geq 1$, $p \geq 2$. Let $q = \lceil 2^{\log_2 p} \rceil$ and $\mathbf{e}_1 \in \mathbb{R}^d$ the first standard basis vector. Consider the padded vector*

$$\widetilde{\mathbf{x}} := \mathbf{x} \otimes \mathbf{e}_1^{\otimes(q-p)} \in \mathbb{R}^{d^q}. \tag{125}$$

Let the j'-th coordinate of $\widetilde{\mathbf{x}}$ be indexed in lexicographic form via

$$j' = 1 + \sum_{t=1}^{q} (j'_t - 1)\, d^{\,t-1}, \quad \textit{with} \quad j'_t \in [d].$$

Then

$$\widetilde{x}_{j'} = \begin{cases} x_j & \textit{if } j'_{p+1} = j'_{p+2} = \cdots = j'_q = 1, \\ 0 & \textit{otherwise}, \end{cases}$$

where

$$j = 1 + \sum_{t=1}^{p} (j'_t - 1)\, d^{\,t-1}.$$

Proof. By definition of the Kronecker product between vectors, for any $\mathbf{u} \in \mathbb{R}^{d^a}$ and $\mathbf{v} \in \mathbb{R}^{d^b}$,

$$\mathbf{u} \otimes \mathbf{v} \in \mathbb{R}^{d^{ab}},$$

is given entrywise by

$$(\mathbf{u} \otimes \mathbf{v})_{(k,l)} = u_k \cdot v_l,$$

where (k, l) denotes the lexicographic pair of indices.

Apply this iteratively to,

$$\widetilde{\mathbf{x}} = \mathbf{x} \otimes \underbrace{\mathbf{e}_1 \otimes \cdots \otimes \mathbf{e}_1}_{q-p \text{ times}}.$$

If in the multi-index $(j_1, \ldots, j_q)$ any of the last $(q-p)$ coordinates $j'_{p+1}, \ldots, j'_q$ is not 1, then at least one factor from $\mathbf{e}_1$ will be zero (since $\mathbf{e}_1$ has 1 at position 1 and 0 elsewhere), and hence $\widetilde{x}_j = 0$.

If instead $j_{p+1} = j_{p+2} = \cdots = j_q = 1$, then each $\mathbf{e}_1$ factor contributes 1 in the product, so

$$\widetilde{\mathbf{x}}_{j'} = \mathbf{x}_j,$$

with j as in the statement, corresponding to the first p indices of the q-tuple.

□

Theorem 10. *[Unbiasedness and Covariance for Non-Power-of-Two p] Let $\mathcal{Z} \in \mathbb{R}^{d \times \cdots \times d \ (p \text{ times})}$ be a mode p tensor with $p \geq 2$, where p is not a power of two. Set $q := 2^{\lceil \log_2 p \rceil}$ and define*

$$\Phi(\mathcal{Z}) := \Pi^q\left(\mathrm{vec}(\mathcal{Z}) \otimes \mathbf{e}_1^{\otimes(q-p)}\right), \tag{126}$$

where $\mathbf{e}_1 \in \mathbb{R}^d$ is the first standard basis vector and Π^q is the power-of-two `Recursive TensorSketch` *from Definition 3.6.*

Then, for all $\mathcal{X}, \mathcal{Y}$,

$$\mathbb{E}[\langle \Phi(\mathcal{X}), \Phi(\mathcal{Y}) \rangle] = \langle \mathcal{X}, \mathcal{Y} \rangle,$$

and for all $\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathcal{Z}$,

$$\begin{aligned} &\mathrm{Cov}(\langle \Phi(\mathcal{X}), \Phi(\mathcal{Y}) \rangle, \langle \Phi(\mathcal{W}), \Phi(\mathcal{Z}) \rangle) \\ &\quad \leq \tfrac{1}{m}\Big(\langle \mathcal{X}, \mathcal{W} \rangle \langle \mathcal{Y}, \mathcal{Z} \rangle + \langle \mathcal{X}, \mathcal{Z} \rangle \langle \mathcal{Y}, \mathcal{W} \rangle \\ &\quad + \sum_{k=1}^{p} \sum_{i_k=1}^{d} \mathcal{X}_{i_1,\ldots,i_p} \mathcal{Y}_{i_1,\ldots,i_p} \mathcal{W}_{i_1,\ldots,i_p} \mathcal{Z}_{i_1,\ldots,i_p} \Big). \end{aligned} \tag{127}$$

Proof. We analyze `Recursive TensorSketch` for sketch mode $p \neq 2^t$ $(t \geq 0)$, proving *unbiasedness* and deriving the *covariance* of the estimator. These results establish that our method provides a valid and efficient sketching scheme for tensors beyond the powers-of-two case.
Let $p \geq 2$ be an integer that is not a power of two, and set

$$q := 2^{\lceil \log_2 p \rceil}.$$

We denote the vectorized forms

$$\mathbf{x} := \mathrm{vec}(\mathcal{X}), \quad \mathbf{y} := \mathrm{vec}(\mathcal{Y}), \quad \mathbf{w} := \mathrm{vec}(\mathcal{W}), \quad \mathbf{z} := \mathrm{vec}(\mathcal{Z}),$$

and introduce the corresponding *padded vectors* as

$$\widetilde{\mathbf{x}} := \mathbf{x} \otimes \mathbf{e}_1^{\otimes(q-p)}, \quad \widetilde{\mathbf{y}} := \mathbf{y} \otimes \mathbf{e}_1^{\otimes(q-p)}, \quad \widetilde{\mathbf{w}} := \mathbf{w} \otimes \mathbf{e}_1^{\otimes(q-p)}, \quad \widetilde{\mathbf{z}} := \mathbf{z} \otimes \mathbf{e}_1^{\otimes(q-p)}. \tag{128}$$

We then define their corresponding *sketches* via

$$\Phi(\mathcal{X}) := \Pi^q \widetilde{\mathbf{x}}, \quad \Phi(\mathcal{Y}) := \Pi^q \widetilde{\mathbf{y}}, \quad \Phi(\mathcal{W}) := \Pi^q \widetilde{\mathbf{w}}, \quad \Phi(\mathcal{Z}) := \Pi^q \widetilde{\mathbf{z}}. \tag{129}$$

We first analyze unbiased estmation of inner product of sketched vectors,

$$\begin{aligned}
\mathbb{E}[\langle \Phi(\mathcal{X}),\, \Phi(\mathcal{Y}) \rangle] =& \mathbb{E}[\langle \Pi^q \widetilde{\mathbf{x}},\, \Pi^q \widetilde{\mathbf{y}} \rangle], && (130)\\
=& \mathbb{E}\Big[\langle \Pi^q (\mathbf{x} \otimes \mathbf{e_1}^{\otimes(q-p)}),\, \Pi^q (\mathbf{y} \otimes \mathbf{e_1}^{\otimes(q-p)}) \rangle\Big], && (131)\\
=& \langle \mathbf{x} \otimes \mathbf{e_1}^{\otimes(q-p)},\, \mathbf{y} \otimes \mathbf{e_1}^{\otimes(q-p)} \rangle, && (132)\\
=& \langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathcal{X}, \mathcal{Y} \rangle. && (133)
\end{aligned}$$

Using the Equation (129) and (128) , we first get Equations (130) and (131). Then, by applying Lemma 8 for $q = 2^t$ $(t \geq 0)$ and Lemma 10, we arrive at Equation (133). Next, we analyze covariance of inner product of sketched vectors,

$$\begin{aligned}
\mathrm{Cov}[\langle \Phi(\mathcal{X}), \Phi(\mathcal{Y}) \rangle,\ \langle \Phi(\mathcal{W}), \Phi(\mathcal{Z}) \rangle] &= \mathrm{Cov}[\langle \Pi^q \widetilde{\mathbf{x}}, \Pi^q \widetilde{\mathbf{y}} \rangle,\ \langle \Pi^q \widetilde{\mathbf{w}}, \Pi^q \widetilde{\mathbf{z}} \rangle], && (134)\\
&\leq \frac{1}{m} \left[\langle \widetilde{\mathbf{x}}, \widetilde{\mathbf{w}} \rangle \langle \widetilde{\mathbf{y}}, \widetilde{\mathbf{z}} \rangle + \langle \widetilde{\mathbf{x}}, \widetilde{\mathbf{z}} \rangle \langle \widetilde{\mathbf{y}}, \widetilde{\mathbf{w}} \rangle + \sum_{j=1}^{d^q} \widetilde{x}_j \widetilde{y}_j \widetilde{w}_j \widetilde{z}_j \right], && (135)\\
&\leq \frac{1}{m} \left[\langle \mathbf{x}, \mathbf{w} \rangle \langle \mathbf{y}, \mathbf{z} \rangle + \langle \mathbf{x}, \mathbf{z} \rangle \langle \mathbf{y}, \mathbf{w} \rangle + \sum_{j=1}^{d^p} x_j y_j w_j z_j \right]\\
&\leq \frac{1}{m} \Bigg[\langle \mathcal{X}, \mathcal{W} \rangle \langle \mathcal{Y}, \mathcal{Z} \rangle + \langle \mathcal{X}, \mathcal{Z} \rangle \langle \mathcal{Y}, \mathcal{W} \rangle\\
&\qquad + \sum_{k}^{p} \sum_{i_k}^{d} \mathcal{X}_{i_1,\ldots,i_p} \mathcal{Y}_{i_1,\ldots,i_p} \mathcal{W}_{i_1,\ldots,i_p} \mathcal{Z}_{i_1,\ldots,i_p} \Bigg]. && (136)
\end{aligned}$$

Using the Equation (129) we get Equation (134). Then by Lemma 9 we arrived at Equation (135) and by Lemma 10 and 11 we get expression for covariance in Equation (136). □

Using Theorem 10, we establish that the sketching map Φ satisfies both unbiasedness and the covariance bound for any p, not only when $p = 2^l$ $\forall l \in \mathbb{N}$.

C Extended Experimental Results

C.1 Experimental Evaluation of Sketching Methods for Tensors in Tensor-Train (TT) Format

We also perform experiments to understand the tradeoff between distortion ratio and sketching time when the input tensors are given in the TT format. The experimental setup and evaluation metric remain the same as

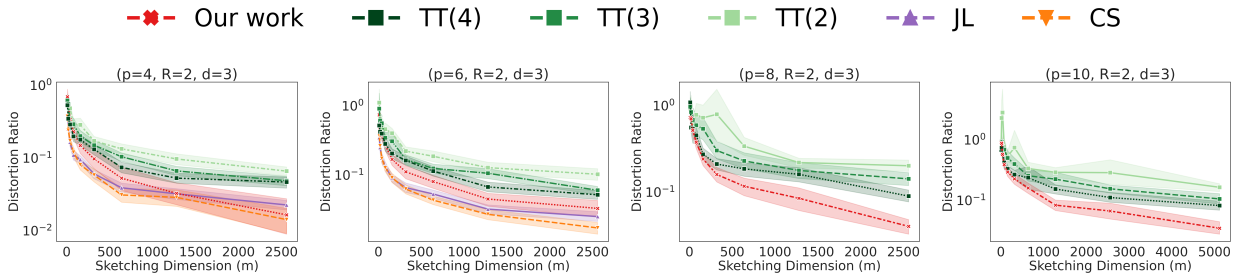

Figure 3: Distortion ratio comparison between our work and baselines for varying tensor mode (p) in the TT representation. TT(R') refers to tensorized random projection, while CS refers to `CountSketch`. The tuple (p, R, d) denote mode, TT rank, and dimension along each mode of the input tensor. In the plots, lines indicate the mean distortion over multiple trials, and the shaded regions illustrate the variation across runs.

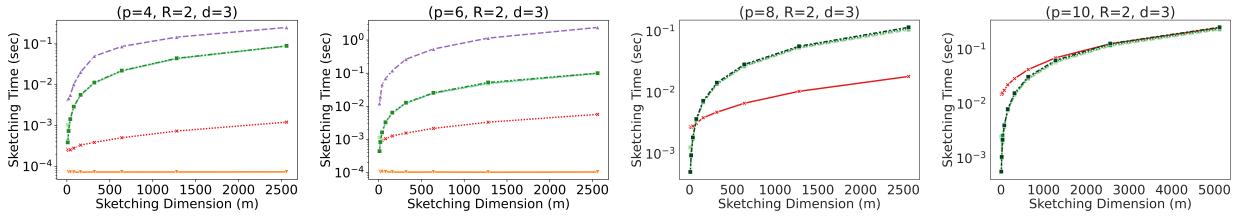

Figure 4: Sketching time comparison of our work with baselines for varying tensor mode (p) in TT representation.

in Section 6 of the main paper. We summarize our results in Figures 3 and 4. As illustrated in Figure 3, the proposed method consistently achieves a lower distortion ratio across varying tensor modes (p). The shaded region in the plots capture the standard deviation of the baseline methods. Further, as the value of p increases, the variance among the baselines also increases, indicating reduced stability in their performance under higher tensor modes. Specifically, the Tensorized Random Projection method shows distortion that worsens exponentially with p and inversely with the CP-rank (R') of the auxiliary random tensor used in its sketching process. Other classical projection methods, such as Johnson–Lindenstrauss (JL) and `CountSketch`, necessitate explicit reconstruction of the full tensor from its compact TT representation. Consequently, their total sketching time comprises both (i) the cost of tensor reconstruction and (ii) the subsequent compression of the reconstructed tensor. To ensure a fair comparison, we employed standard tensor toolboxes for converting TT/CP representations into full tensors [Oseledets et al., 2014, Nickel and Rol, 2013]. Empirically, we found that while these methods perform competitively for low-order tensors (e.g., $p = 4, 6$), their computational overhead becomes prohibitive for higher-order cases (e.g., $p = 8, 10$), with the tensor reconstruction step failing to complete within 2 hours. In contrast, our proposed approach operates directly on the TT representation, scales efficiently to high-order tensors, and consistently achieves substantially lower distortion ratios.

Figure 4 presents the corresponding sketching time analysis. Our method achieves lower or comparable sketching times relative to Tensorized Random Projection and JL across all tested tensor modes (p), while simultaneously maintaining superior distortion performance. `CountSketch` demonstrates lower sketching time for small, low-order tensors (e.g., $p = 4, 6$). However, it requires explicit construction of the input tensor, which becomes infeasible for higher-order tensors (e.g., $p = 8, 10$) due to memory constraints. As a result, we could not run `CountSketch` in these cases. In contrast, our method scales efficiently with tensor order, achieving a favourable trade-off between sketching time and distortion quality.

As mentioned in Section 2, when input tensor is given in TT format, our proposal, achieve an asymptotic improvement in time complexity *w.r.t.* $f_{TT(R')}$ [Rakhshan and Rabusseau, 2020] under the condition $\widetilde{R} = o(R^3)$, where R denotes the CP rank of the input tensor $\mathcal{X}$, which also admits TT ranks $(R_0, R_1, \ldots, R_p)$ and $\widetilde{R} := \prod_{k=0}^{p} R_k$. As established in Lemma 2, the two rank formats satisfy $R \leq \widetilde{R}$. Our experiments established that this condition is not restrictive in practice—most real-world tensor data exhibit moderate TT-ranks relative to their CP-ranks, ensuring that the asymptotic advantage of our method over baselines.

C.2 Pairwise Distance Estimation in real world datasets

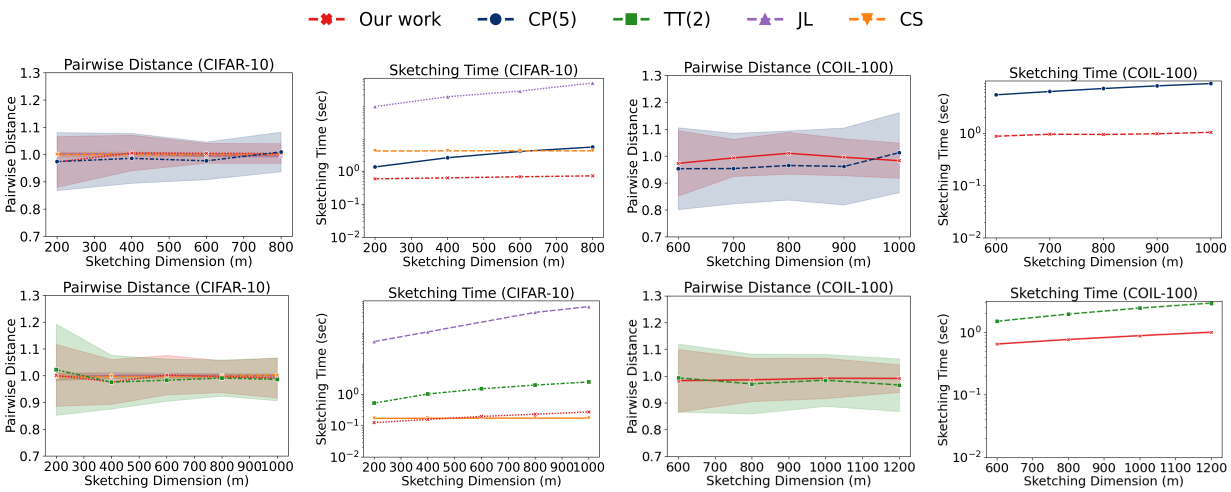

Figure 5: Pairwise distance preservation and sketching time comparison between our proposed method and baseline approaches on real-world datasets. CP(R') and TT(R') denote the tensorized random projection methods, while CS represents `CountSketch`. In the plots, lines indicate the mean over trials and the shaded regions represent one standard deviation. The first four sets of plots correspond to CP input tensors, and the subsequent four sets correspond to TT input tensors.

We also perform experiments on some real-life datasets - CIFAR-10 and COIL-100. We compare our method against tensorized random projections, the Johnson–Lindenstrauss (JL) sketch, and `CountSketch`. For the CIFAR-10 dataset, we reshape the first $n = 100$ image vectors (originally of size $32 \times 32 \times 3$) into 6^{th}-order tensors of size $4 \times 4 \times 4 \times 4 \times 4 \times 3$. Similarly, for the COIL-100 dataset, we reshape the first $n = 100$ images (of size $128 \times 128 \times 3$) into 8^{th}-order tensors of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$. All tensors are normalized prior to projection.

To evaluate geometric consistency, we compute the mean pairwise distance preservation ratio

$$\frac{1}{n(n-1)} \sum_{1 \le i \ne j \le n} \frac{\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2}{\|\mathbf{x}_i - \mathbf{x}_j\|_2},$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the input tensors, along with its standard deviation over 100 independent trials for varying projection dimensions k. As illustrated in Figure 5, the lines denote the mean pairwise distance ratios, while the shaded regions represent the standard deviation.

For the CIFAR-10 dataset, our method consistently achieves superior distance preservation compared to tensorized random projections, as indicated by the smaller shaded region representing standard deviation in the estimated pairwise distances. While classical algorithms such as JL and `CountSketch` achieve comparable standard deviation in pairwise distance estimation, they incur substantially higher sketching times due to the explicit construction of large tensor representations.

For the COIL-100 dataset, both JL and `CountSketch` require explicit reconstruction of the full tensor from its CP representation. As a result, sketching did not complete within 2 hours. Consequently, we focus our comparison primarily against tensorized random projections for this dataset. As illustrated, the shaded region denoting the standard deviation of the pairwise distance estimates is markedly larger for CP(5) than for our approach, underscoring the superior robustness and efficiency of our method in preserving geometric structure under compression.