On a Connection Between Fast and Sparse Oblivious Subspace Embeddings

Rui Wang **Wangli Xu**
Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing 100872, China

Abstract

Fast Johnson-Lindenstrauss Transform (FJLT) and Sparse Johnson-Lindenstrauss Transform (SJLT) are two important oblivious subspace embeddings. So far, the developments of these two methods are almost orthogonal. In this work, we propose an iterative algorithm for oblivious subspace embedding which makes a connection between these two methods. The proposed method is built upon an iterative implementation of FJLT and is equipped with several theoretically motivated modifications. One important strategy we adopt is the early stopping strategy. On the one hand, the early stopping strategy makes our algorithm fast. On the other hand, it results in a sparse embedding matrix. As a result, the proposed algorithm is not only faster than the FJLT, but also faster than the SJLT with the same degree of sparsity. We present a general theoretical framework to analyze the embedding property of sparse embedding methods, which is used to prove the embedding property of the proposed method. This framework is also of independent interest. Lastly, we conduct numerical experiments to verify the good performance of the proposed algorithm.

1 INTRODUCTION

In the practice of statistics and machine learning, it is often necessary to deal with datasets with extremely large volume. When handling large dataset, exact algorithms often cost prohibitive computing time. In recent years, approximate algorithms based on sketching techniques have been actively researched for various tasks. Sketching methods use random subspace embeddings to reduce the data volume, yielding a fast computing time for subsequent procedures; see, e.g., Woodruff (2014); Kannan and Vempala (2017); Martinsson and Tropp (2020) for reviews of sketching methods. Take the linear regression problem as an example, suppose we have a data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and a response vector $\mathbf{y} \in \mathbb{R}^n$. And we would like to solve the least-squares problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2.$$

Using sketching method, we generate a random matrix $\mathbf{\Pi} \in \mathbb{R}^{m\times n}$ with $m \ll n$ and consider the sketched least-squares problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{\Pi}\mathbf{A}\mathbf{x}-\mathbf{\Pi}\mathbf{y}\|^2.$$

Note that the sketched least-squares problem only relies on the the sketched data $(\mathbf{\Pi}\mathbf{A}, \mathbf{\Pi}\mathbf{y})$. If $(\mathbf{\Pi}\mathbf{A}, \mathbf{\Pi}\mathbf{y})$ can well preserve the information of the original data and can be obtained efficiently, then one can obtain an approximate solution to the least-squares problem efficiently. Furthermore, when used in conjunction with iteration algorithms, sketching methods can also yield high-precision approximation to the least-squares problem; see, e.g., Rokhlin and Tygert (2008); Avron et al. (2010); Pilanci and Wainwright (2016); Lacotte et al. (2020); Lacotte and Pilanci (2020b). For these sketching methods, the subspace embedding step is one of the computational bottlenecks. Also, the precision of sketching methods is often largely affected by the subspace embedding property of $\mathbf{\Pi}$.

In this paper, we consider subspace embeddings that are oblivious to the input data. Following Cohen (2016), for $\epsilon, \delta \in (0,1)$, an $m \times n$ random matrix $\mathbf{\Pi}$ is called an Oblivious Subspace Embedding (OSE) with parameter (d, ϵ, δ) if for any non-random $n \times d$ column orthogonal matrix $\mathbf{U}$,

$$\Pr(\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d\| > \epsilon) < \delta. \quad (1)$$

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

If $d = 1$, an OSE with property (1) reduces to a Johnson-Lindenstrauss transform (Johnson and Lindenstrauss, 1984). The celebrated Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984) implies that there exists a $(1, \epsilon, \delta)$-OSE with $m = \Theta(\epsilon^{-2} \log(1/\delta))$. Standard proofs of Johnson-Lindenstrauss lemma consider unstructured dense $\mathbf{\Pi}$. For example, one can choose $\mathbf{\Pi}$ to be a random matrix whose elements are independent and identically distributed (i.i.d.) sub-Gaussian random variables. However, it is not efficient to apply such unstructured $\mathbf{\Pi}$ to input data.

Researchers have made much efforts to achieve fast embedding time while preserving good embedding property. In the seminal work of Ailon and Chazelle (2006, 2009), a highly structured embedding matrix, known as Fast Johnson-Lindenstrauss Transform (FJLT), is constructed. Since FJLT is constructed via Walsh-Hadamard matrix, it is also known as Subsampled Randomized Hadamard Transform (SRHT). For FJLT, the embedding time for an input data $\mathbf{A} \in \mathbb{R}^{n \times d}$ is $O(nd \log(m))$; see, e.g., Ailon and Liberty (2009). The embedding property of FJLT has been investigated by many researchers; see, e.g., Sarlós (2006), Tropp (2011), Boutsidis and Gittens (2013), Lu et al. (2013), Cohen et al. (2016), Lacotte et al. (2020), Lacotte and Pilanci (2020a). In Cohen et al. (2016), it was shown that FJLT is (d, ϵ, δ)-OSE for $m = \Omega(\epsilon^{-2}(d + \log(1/(\epsilon\delta))) \log(d/\delta))$.

In another line of work, researchers sought for fast embedding time by making $\mathbf{\Pi}$ sparse. This direction was initiated by Achlioptas (2003) who considered an embedding matrix whose elements are i.i.d. and equal to 0 with probability 2/3. After some subsequent developments (Dasgupta et al., 2010; Braverman et al., 2010; Clarkson and Woodruff, 2013; Meng and Mahoney, 2013), Kane and Nelson (2014) introduced Sparse Johnson-Lindenstrauss Transform (SJLT). For SJLT, each column of $\mathbf{\Pi}$ contains exactly s nonzero entries. For input data $\mathbf{A} \in \mathbb{R}^{n \times d}$, the application of SJLT can be completed within $O(nds)$ time. Furthermore, SJLT can achieve input sparsity time for sparse input data. The subspace embedding property of SJLT for general $d \geq 1$ was investigated by Nelson and Nguyen (2013); Bourgain et al. (2015); Cohen (2016). In Cohen (2016), it was shown that some constructions of SJLT are (d, ϵ, δ)-OSE with $m = \Omega(\epsilon^{-2} d \log(d/\delta))$ and $s = \Theta(\epsilon^{-1} \log(d/\delta))$.

FJLT and SJLT have their own advantages. The recursive structure of FJLT allows for fast embedding time for arbitrary input data. However, the FJLT embedding matrix is dense and does not utilize sparsity. On the other hand, while SJLT does not have a recursive structure, its sparsity yields fast embedding time. So far, the developments of FJLT and SJLT are almost orthogonal. The goal of the present work is to propose a new OSE method which utilizes the advantages of both FJLT and SJLT. The widely used algorithm of FJLT is a recursive algorithm. We propose an iterative implementation of FJLT. The iterative algorithm allows us to access the intermediate results of FJLT. We find that the intermediate results of FJLT share certain key properties of SJLT and are highly structured. This motivates us to adopt the early stopping strategy to obtain a sparse OSE. While this idea can not be directly applied, it can indeed work after some modifications. We investigate the embedding property of the proposed embedding method. It shows that the proposed embedding method has an embedding property which is similar to that of SJLT. Also, the proposed embedding method is significantly faster than both FJLT and SJLT.

Our main contributions are as follows:

- We present a general theoretical framework to analyze the subspace embedding property of sparse OSEs. It generalizes the result of Nelson and Nguyen (2013) in several ways. This framework is interesting in its own right. As a special case, our theorem gives the subspace embedding property of Allen-Zhu et al. (2014).
- We propose an iterative implementation of FJLT. This implementation allows us to access the intermediate results of FJLT. It turns out that the intermediate result of FJLT is closely related to SJLT.
- We consider some theoretically motivated modifications of the iterative implementation of FJLT and propose a new sparse OSE. We derive the embedding property of the proposed sparse OSE via our general framework. For any input data $\mathbf{A} \in \mathbb{R}^{n \times d}$, the embedding time of the proposed method is $O(nd(\log(1/\epsilon) + \log(\log(d/\delta))))$, which improves the computing time of FJLT and SJLT significantly.

2 PRELIMINARIES

First we introduce some notations that will be used throughout the paper. For elements $a_1, \ldots, a_n$, we use $\{a_1, \cdots, a_n\}$ to denote the set of $a_1, \ldots, a_n$, which is unordered and has distinct elements; we use $(a_1, \cdots, a_n)$ to denote the tuple of $a_1, \ldots, a_n$, which is ordered. For a set A, let $\mathrm{Card}(A)$ denote the cardinality of A. For sets $A_1, \ldots, A_n$, let $A_1 \times \cdots \times A_n$ denote their product $\{(x_1, \ldots, x_n) : x_i \in A_i, i = 1, \ldots, n\}$. A function f is understood as its set-theoretic definition.

That is, a function f is a subset of $A \times B$ for some sets A and B such that if $(x, y_1) \in f$ and $(x, y_2) \in f$ then $y_1 = y_2$. The domain and range of f are denoted as $\mathrm{dom}(f)$ and $\mathrm{range}(f)$, respectively. The set of all functions from A to B is denoted as B^A. For positive integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. For real number x, denote by $\lfloor x \rfloor$ the floor of x. For two non-negative sequence $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if there exists a constant $c > 0$ such that $a_n \le c b_n$ for all n, we write $a_n = \Omega(b_n)$ if $b_n = O(a_n)$, and we write $a_n = \Theta(b_n)$ if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$. For two matrices $\mathbf{A}, \mathbf{B}$, let $\mathbf{A} \otimes \mathbf{B} := (a_{i,j}\mathbf{B})$ denote their Kronecker product, and we denote

$$\mathrm{diag}(\mathbf{A}, \mathbf{B}) := \begin{pmatrix} \mathbf{A} & \mathbf{O} \\ \mathbf{O} & \mathbf{B} \end{pmatrix}.$$

If $\mathbf{A}$ and $\mathbf{B}$ have the same dimension, let $\mathbf{A} \circ \mathbf{B} := (a_{i,j} b_{i,j})$ denote their Hadamard product. Throughout the paper, we assume that m, n and s are powers of 2.

Following Tropp (2011); Lu et al. (2013); Boutsidis and Gittens (2013); Lacotte and Pilanci (2020a), FJLT is defined as

$$\mathbf{\Pi}_{\mathrm{F}} := \frac{1}{\sqrt{m}} \mathbf{P} \mathbf{H}_n \mathbf{D},$$

where $\mathbf{D}$ is an $n \times n$ diagonal matrix whose diagonal elements are independent Rademacher random variables, that is, random variables taking values in -1 an 1 with equal probability $1/2$, $\mathbf{H}_n$ is Walsh-Hadamard matrix defined recursively as

$$\mathbf{H}_n := \begin{pmatrix} \mathbf{H}_{\frac{n}{2}} & \mathbf{H}_{\frac{n}{2}} \\ \mathbf{H}_{\frac{n}{2}} & -\mathbf{H}_{\frac{n}{2}} \end{pmatrix}$$

with $\mathbf{H}_1 = 1$, and $\mathbf{P}$ is an $m \times n$ matrix whose rows are m uniform samples (without replacement) from the standard bases of $\mathbb{R}^n$, $\mathbf{P}$ and $\mathbf{D}$ are independent.

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a data matrix. Using a recursive algorithm, one can compute $\mathbf{\Pi}_{\mathrm{F}} \mathbf{X}$ in $O(nd \log(n))$ time. This recursive algorithm is similar to Cooley-Tukey algorithm of fast Fourier transform (Cooley and Tukey, 1965), and is summarized in Algorithm 1. The fastest known algorithm of FJLT is proposed by Ailon and Liberty (2009), which is based on a careful pruning of the execution tree of Algorithm 1. The algorithm of Ailon and Liberty (2009) can compute $\mathbf{\Pi}_{\mathrm{F}} \mathbf{X}$ in $O(nd \log(m))$ time; see Ailon and Liberty (2009), Theorem 2.1.

Kane and Nelson (2014) proposed SJLT which is defined as

$$\mathbf{\Pi}_{\mathrm{S}} = \frac{1}{\sqrt{s}} \mathbf{\Delta} \circ \mathbf{\Sigma},$$

where $\mathbf{\Sigma} = (\sigma_{i,j})$ and $\mathbf{\Delta} = (\delta_{i,j})$ are independent $m \times n$ random matrices, $\{\sigma_{i,j}\}$ are independent Rademacher

Algorithm 1: A recursive implementation of FJLT

```
// Compute H_n X
Function RecursiveAlgorithm(n, m, d, X):
    X_1 ← X[1 : n/2, 1 : d]; X_2 ← X[(n/2 + 1) : n, 1 : d]
    Y_1 ← RecursiveAlgorithm(n/2, m, d,
     X_1 + X_2)
    Y_2 ← RecursiveAlgorithm(n/2, m, d,
     X_1 − X_2)
    return (Y_1; Y_2)
// Compute Π_F X
Function FJLT(n, m, d, X):
    Generate matrices P and D
    Y ← RecursiveAlgorithm(n, m, d, DX)
    return PY
```

random variables, the n columns of $\mathbf{\Delta}$ are independent, $\delta_{i,j} \in \{0, 1\}$ and $\sum_{i=1}^{m} \delta_{i,j} = s$. Kane and Nelson (2014) gave two concrete constructions of $\mathbf{\Delta}$. One is the *graph construction*, that is, for each column of $\mathbf{\Delta}$, the positions of nonzeros are uniformly sampled from $[m]$ without replacement. The other one is the *block construction*, that is, the m rows of $\mathbf{\Delta}$ are divided into s blocks with equal numbers and for each column of $\mathbf{\Delta}$, each block contains exactly one nonzero element whose position is uniformly selected within the block. Nelson and Nguyen (2013) investigated the embedding property of SJLT with more general $\mathbf{\Delta}$. They considered the following Oblivious Sparse Norm-Approximating Projections (OSNAP) properties:

(1) The elements $\sigma_{i,j}$ of $\mathbf{\Sigma}$ are i.i.d. Rademacher random variables.

(2) The elements $\delta_{i,j}$ of $\mathbf{\Delta}$ take value in $\{0, 1\}$.

(3) For any $j \in [n]$, $\sum_{i=1}^{m} \delta_{i,j} = s$.

(4) For any $\mathcal{S} \subset [m] \times [n]$,

$$\mathrm{E}\left(\prod_{(i,j) \in \mathcal{S}} \delta_{i,j} \right) \le \left(\frac{s}{m} \right)^{\mathrm{Card}(\mathcal{S})}.$$

(5) The columns of $\mathbf{\Pi}$ are i.i.d.

Nelson and Nguyen (2013) proved that if $\mathbf{\Pi}_{\mathrm{S}}$ satisfies OSNAP properties, then $\mathbf{\Pi}_{\mathrm{S}}$ satisfies (d, ϵ, δ)-OSE property provided that $s = \Theta(\epsilon^{-1} \log^3(d/\delta))$ and $m = \Omega(\epsilon^{-2} d \log^6(d/\delta))$. Later, Cohen (2016) proved that for SJLT with graph construction, one can take $m = \Omega(\epsilon^{-2} d \log(d/\delta))$ and $s = \Theta(\epsilon^{-1} \log(d/\delta))$ to achieve (d, ϵ, δ)-OSE property, which nearly matches the lower bound derived in Nelson and Nguyên (2014).

As Cohen (2016) noted, however, it is not clear if their analysis can be applied to the general SJLT with OSNAP properties.

3 SUBSPACE EMBEDDING PROPERTY OF GENERAL SPARSE OSES

While the framework of Nelson and Nguyen (2013) is fairly general, it does not include our proposed algorithm in Section 5. Also, it does not include some variants of SJLT, such as sign-consistent SJLT (Allen-Zhu et al., 2014). In this section, we extend the analysis of Nelson and Nguyen (2013) to a general setting. Our general framework will be used to investigate the proposed algorithm. It also gives theoretical properties of some variants of SJLT.

We make the following assumption which extends OSNAP properties.

Assumption 1. *Suppose* $\mathbf{\Pi} = s^{-1/2}\mathbf{\Delta} \circ \mathbf{\Sigma}$, *where* $\mathbf{\Sigma} = (\sigma_{i,j})$ *and* $\mathbf{\Delta} = (\delta_{i,j})$ *are independent* $m \times n$ *random matrices and satisfy the following conditions:*

(1) Suppose the elements $\sigma_{i,j}$ *of* $\mathbf{\Sigma}$ *are generated according to one of the two following schemes: (a)* $\{\sigma_{i,j}\}$ *are i.i.d. Rademacher random variables; or (b)* $\sigma_{i,j} = \sigma_{1,j}$ *and* $\{\sigma_{1,j}\}_{j=1}^{n}$ *are i.i.d. Rademacher random variables.*

(2) The elements $\delta_{i,j}$ *of* $\mathbf{\Delta}$ *take value in* $\{-1, 0, 1\}$.

(3) For any $j \in [n]$, $\sum_{i=1}^{m} \delta_{i,j} = s$.

(4) There is an absolute constant $c > 0$ *such that for any* $\mathcal{S} \subset [m] \times [n]$,

$$\mathrm{E} \prod_{(i,j)\in\mathcal{S}} |\delta_{i,j}| \le \Big(c\frac{s}{m}\Big)^{\mathrm{Card}(\mathcal{S})}.$$

(5) The distribution of $\mathbf{\Delta}$ *is invariant under the permutation of columns.*

Assumption 1 generalizes OSNAP properties in several aspects. First, we allow two generation schemes of $\mathbf{\Sigma}$. The first one is the standard scheme used in SJLT. And the second one uses a single Rademacher random variable in each column of $\mathbf{\Sigma}$. This scheme is used in sign-consistent SJLT (Allen-Zhu et al., 2014). Compared with the first scheme, the second one requires less bits of random seeds. Second, we allow $\delta_{i,j}$ taking on three values -1, 0, and 1 while OSNAP property requires that $\delta_{i,j}$ takes value in $\{0, 1\}$. Third, we only require that the columns of $\mathbf{\Delta}$ are exchangeable while OSNAP property requires that the columns of $\mathbf{\Pi}$ are independent. Hence Assumption 1 greatly generalizes OSNAP. We have the following theorem.

Theorem 1. *Let* $\mathbf{\Pi}$ *be an* $m \times n$ *random matrix satisfying Assumption 1. Suppose* $\epsilon, \delta \in (0, 1)$ *and*

$$m \ge C\log^2(d/\delta)s^2 d, \quad s \ge C\frac{\log^2(d/\delta)}{\epsilon}, \qquad (2)$$

where C *is an absolute constant. Then* $\mathbf{\Pi}$ *is* (d, ϵ, δ)*-OSE.*

To meet the condition (2), we can take $m = \Omega(d\log^6(d/\delta)/\epsilon^2)$ and $s = \Theta(\log^2(d/\delta)/\epsilon)$. Therefore, compared with Theorem 5 of Nelson and Nguyen (2013), our result improves the order of s by a logarithm factor. In the meanwhile, the conditions of Theorem 1 is much weaker than theirs. For example, while sign-consistent SJLT of Allen-Zhu et al. (2014) does not satisfy OSNAP properties, it satisfies the condition of Theorem 1. Hence Theorem 1 gives the subspace embedding property of sign-consistent SJLT.

Following Kane and Nelson (2014); Nelson and Nguyen (2013); Allen-Zhu et al. (2014), we use the moment method to prove Theorem 1. The basic idea of this approach is to use the Markov's inequality to obtain the bound

$$\begin{aligned}&\Pr(\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\| > \epsilon)\\ \le&\frac{1}{\epsilon^\ell}\,\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d)^\ell),\end{aligned}$$

where $\ell > 0$ is an even integer. We give a fine-grained analysis to bound the moment in the right hand side. This fine-grained analysis is based on graph theory.

4 AN ITERATIVE IMPLEMENTATION OF FJLT

In this section, we present a new algorithm of FJLT. The proposed algorithm can be regarded as an iterative version of Algorithm 1. While the proposed algorithm has the same order of computing time as Algorithm 1, it provides an insight on the connection between FJLT and SJLT, and will motivate a new algorithm which is faster than both FJLT and SJLT.

We note that Algorithm 1 is a recursive algorithm. It is also possible to use iterative algorithm to implement FJLT; see Yarlagadda and Hershey (1997), Chapter 7. Now we present a simple iterative algorithm of FJLT with computing time $O(n\log(n))$. For any $k \in [\log_2(n)]$, define

$$\mathbf{B}_k := \mathbf{I}_{2^{k-1}} \otimes \mathbf{H}_2 \otimes \mathbf{I}_{\frac{n}{2^k}}.$$

Since Kronecker product is associative, $\mathbf{B}_k$ is well-defined. We have the following lemma.

Lemma 1. *For* $k \in [\log_2(n)]$, *we have*

$$\mathbf{B}_k\mathbf{B}_{k-1}\cdots\mathbf{B}_1 = \mathbf{H}_{2^k} \otimes \mathbf{I}_{\frac{n}{2^k}}.$$

Algorithm 2: An iterative implementation of FJLT

Function `IterativeFJLT`(n, m, d, $\mathbf{X}$):

Generate matrices $\mathbf{P}$ and $\mathbf{D}$

$\mathbf{Y}_1^{(0)} \leftarrow \mathbf{DX}$

for $k = 1$ **to** $\log_2(n)$ **do**

for $p = 1$ **to** 2^{k-1} **do**

$$\mathbf{Y}_{2p-1}^{(k)} \leftarrow \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & \mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{Y}_p^{(k-1)}$$

$$\mathbf{Y}_{2p}^{(k)} \leftarrow \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & -\mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{Y}_p^{(k-1)}$$

$$\mathbf{Y} \leftarrow \begin{pmatrix} \mathbf{Y}_1^{(\log_2(n))} \\ \vdots \\ \mathbf{Y}_n^{(\log_2(n))} \end{pmatrix}$$

return $\mathbf{PY}$

From Lemma 1, we have

$$\mathbf{H}_n := \mathbf{B}_{\log_2(n)} \mathbf{B}_{\log_2(n)-1} \cdots \mathbf{B}_1.$$

Based on the above expression, the matrix multiplication by $\mathbf{H}_n$ can be decomposed into $\log_2(n)$ subsequent multiplications. Note that for any $k \in [\log_2(n)]$, each column of $\mathbf{B}_k$ contains exactly two nonzero elements. Hence $\mathbf{B}_k$ allows for fast matrix multiplication. This allows us to use an iterative algorithm to compute FJLT. Below we give an explicit form of this iteration algorithm.

Let $\mathbf{C}_1^{(0)} := \mathbf{I}_n$. For $k \in [\log_2(n)]$ and $p \in [2^{k-1}]$, we define

$$\mathbf{C}_{2p-1}^{(k)} := \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & \mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{C}_p^{(k-1)},$$

$$\mathbf{C}_{2p}^{(k)} := \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & -\mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{C}_p^{(k-1)}.$$

For $k \in \{0, 1, \ldots, \log_2(n)\}$, define

$$\mathbf{C}^{(k)} := \begin{pmatrix} \mathbf{C}_1^{(k)} \\ \vdots \\ \mathbf{C}_{2^k}^{(k)} \end{pmatrix}.$$

From the above definition, we have $\mathbf{C}^{(0)} = \mathbf{C}_1^{(0)}$ and $\mathbf{C}^{(k)} = \mathbf{B}_k \mathbf{C}^{(k-1)}$, $k \in [\log_2(n)]$. It follows that $\mathbf{C}^{(k)} = \mathbf{B}_k \mathbf{B}_{k-1} \cdots \mathbf{B}_1$, $k \in [\log_2(n)]$. In particular, we have $\mathbf{C}^{(\log_2(n))} = \mathbf{H}_n$. Based on the above derivations, we can formulate an iterative implementation of FJLT, as summarized in Algorithm 2. In Algorithm 2, the matrix $\mathbf{Y}_p^{(k)}$ is equal to $\mathbf{C}_p^{(k)} \mathbf{DX}$. Hence Algorithm 2 returns $\mathbf{PC}^{(\log_2(n))} \mathbf{DX}$ which is exactly $\mathbf{\Pi}_{\mathrm{F}} \mathbf{X}$.

Now we consider the computing time of Algorithm 2. In Algorithm 2, $\mathbf{Y}_p^{(k)}$ is a $(2^{-k}n) \times d$ matrix. Hence given $\mathbf{Y}_p^{(k)}$, the computation of $\mathbf{Y}_{2p-1}^{(k)}$ and $\mathbf{Y}_{2p}^{(k)}$ costs $O(2^{-k}nd)$ time. Hence for each iteration of $k \in [\log_2(n)]$, the computation costs $O(nd)$ time. Consequently, the total computing time of Algorithm 2 is $O(nd \log(n))$, which equals the computing time of Algorithm 1.

While Algorithm 2 is not faster than Algorithm 1, it is an iterative algorithm and hence allows us to access intermediate results of FJLT. Initially, $\mathbf{C}^{(0)}$ is the identity matrix which has exactly one nonzero element in each column. In general, it can be proved by mathematical induction that for any $k \in [\log_2(n)]$ and $p \in [2^k]$, $\mathbf{C}_p^{(k)}$ has exactly one nonzero element in each column. Consequently, $\mathbf{C}^{(k)}$ has exactly 2^k nonzero element in each column. Thus, although $\mathbf{H}_n = \mathbf{C}^{(\log_2(n))}$ is a dense matrix, the intermediate matrices $\{\mathbf{C}^{(k)}\}_{k=0}^{n-1}$ are sparse and satisfy the properties (2) and (3) in Assumption 1.

We have presented an evolution process which starts with the identity matrix which is a very sparse matrix, and gradually processes sparse operations and finally reaches the dense matrix of FJLT. This evolution process reveals an interesting connection between FJLT and SJLT. In view of the theory of SJLT, one may expect that the intermediate matrices of FJLT are enough to yield a good OSE method. We note that for SJLT matrix $\mathbf{\Pi}_{\mathrm{S}}$ with sparsity parameter s, the computation of $\mathbf{\Pi}_{\mathrm{S}} \mathbf{X}$ requires $O(sNd)$ time. On the other hand, for the iterative implementation of FJLT, the matrix $\mathbf{C}^{(\log_2(s))}$ has s nonzero elements in each column and the computation of $\mathbf{C}^{(\log_2(s))} \mathbf{DX}$ only requires $O(\log(s)Nd)$ time. This advantage of FJLT in computing time implies that it may be possible to utilize the structural property of FJLT to obtain a faster sparse OSE method.

5 A FASTER SPARSE OSE METHOD

In this section, we propose a faster sparse OSE method with good subspace embedding property. Based on our previous analysis, a natural idea to obtain a faster OSE method is to adopt the early stopping strategy in Algorithm 2 and use the intermediate matrix $\mathbf{C}^{(\log_2(s))}$ instead of $\mathbf{H}_n$ in the definition of FJLT to obtain a sparse embedding:

$$\frac{1}{\sqrt{m}} \mathbf{PC}^{(\log_2(s))} \mathbf{D}. \tag{3}$$

However, the direct application of this idea meets some difficulties.

First, the expression (3) involves a sampling matrix $\mathbf{P}$. While $\mathbf{C}^{(\log_2(s))}$ satisfies the condition (3) in Assumption 1, after the action of $\mathbf{P}$, the columns of the matrix

$\mathbf{PC}^{(\log_2(s))}$ are not guaranteed to have constant numbers of nonzero elements. We note that for FJLT, the sampling matrix $\mathbf{P}$ is used to reduce the row number of data from n to m. To avoid the difficulty caused by $\mathbf{P}$, we use another matrix to reduce the row number. For $k \in [s]$, let $\tilde{\mathbf{P}}_k \in \mathbb{R}^{(m/s)\times(n/s)}$ be a random matrix whose columns are independent and are uniformly sampled from the standard basis of $\mathbb{R}^{m/s}$. Then for $k \in [\log_2(s)]$, the matrix $\tilde{\mathbf{P}}_k \mathbf{C}_k^{(\log_2(s))}$ has exactly one nonzero element in each column. Define $\tilde{\mathbf{P}} = \mathrm{diag}(\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_s)$. Then $\tilde{\mathbf{P}}$ is an $m \times n$ matrix and we have

$$\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))} = \begin{pmatrix} \tilde{\mathbf{P}}_1 \mathbf{C}_1^{(\log_2(s))} \\ \vdots \\ \tilde{\mathbf{P}}_s \mathbf{C}_s^{(\log_2(s))} \end{pmatrix}.$$

Thus, $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$ has exactly s nonzero elements in each column and satisfies the condition (3) in Assumption 1.

Before we proceed, we would like to give a further understanding of the structure of $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$. From Lemma 1, we have $\mathbf{C}^{(\log_2(s))} = \mathbf{H}_s \otimes \mathbf{I}_{n/s}$. Therefore, the matrix $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$ has the following form:

$$\begin{pmatrix} \pm\tilde{\mathbf{P}}_1 & \cdots & \pm\tilde{\mathbf{P}}_1 \\ \pm\tilde{\mathbf{P}}_2 & \cdots & \pm\tilde{\mathbf{P}}_2 \\ \vdots & \ddots & \vdots \\ \pm\tilde{\mathbf{P}}_s & \cdots & \pm\tilde{\mathbf{P}}_s \end{pmatrix}.$$

From the above expression, we can see that the matrix $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$ is highly structured with some repeated blocks. We note that such a structured matrix can not satisfy the condition (4) in Assumption 1. To see this, consider the index set

$$\mathcal{S} = \{(1,1), (1, n/s+1), \ldots, (1, n(s-1)/s+1)\}.$$

For this $\mathcal{S}$, we have

$$\mathrm{E} \prod_{(i,j)\in\mathcal{S}} \left|(\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))})_{i,j}\right| = \mathrm{E}\left|(\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))})_{1,1}\right| = \frac{s}{m},$$

which violates the condition (4) in Assumption 1.

The above difficulty is caused by the repeated blocks in the matrix $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$. To ease this difficulty, we resort to an additional random permutation. Specifically, let $\mathbf{G} \in \mathbb{R}^{n\times n}$ be a uniformly distributed permutation matrix. We define

$$\tilde{\mathbf{\Delta}} := \tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}\mathbf{G}.$$

The columns of $\tilde{\mathbf{\Delta}}$ are permuted, and hence does not have repeated blocks. It can be expected that $\tilde{\mathbf{\Delta}}$ is more likely to satisfy the condition (4) in Assumption 1. On the other hand, we note that the columns of the original matrix $\tilde{\mathbf{P}}\mathbf{C}^{(\log_2(s))}$ are not exchangeable which violates the condition (5) of Assumption 1. In comparison, for the permuted matrix $\tilde{\mathbf{\Delta}}$, the distribution of its columns is invariant under the permutation of columns, which satisfies the condition (5) of Assumption 1. Here we should emphsis that the above permutation technique is not new. In fact, Lacotte et al. (2020) considered a similar permutation trick for FJLT to break the non-uniformity in the data.

Algorithm 3: Faster sparse OSE method

Function `IterativeFJLT(`n, m, d, s, $\mathbf{X}$`):`

Generate matrices $\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_s$, $\mathbf{G}$ and $\mathbf{D}$

$\mathbf{Y}_1^{(0)} \leftarrow \mathbf{GDX}$

for $k = 1$ **to** $\log_2(s)$ **do**

  for $p = 1$ **to** 2^{k-1} **do**

$$\mathbf{Y}_{2p-1}^{(k)} \leftarrow \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & \mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{Y}_p^{(k-1)}$$

$$\mathbf{Y}_{2p}^{(k)} \leftarrow \begin{pmatrix} \mathbf{I}_{\frac{n}{2^k}} & -\mathbf{I}_{\frac{n}{2^k}} \end{pmatrix} \mathbf{Y}_p^{(k-1)}$$

return $\begin{pmatrix} \tilde{\mathbf{P}}_1 \mathbf{Y}_1^{(\log_2(s))} \\ \vdots \\ \tilde{\mathbf{P}}_s \mathbf{Y}_s^{(\log_2(s))} \end{pmatrix}$

Having introduced the above modifications of (3), now we are ready to define the proposed subspace embedding method:

$$\mathbf{\Pi}_{\mathrm{NEW}} := s^{-1/2}\tilde{\mathbf{\Delta}}\mathbf{D}.$$

We summarize the fast computation algorithm of $\mathbf{\Pi}_{\mathrm{NEW}}$ in Algorithm 3.

Now we analyze the randomness of $\tilde{\mathbf{\Delta}}$. Let $\tilde{p}_j^{(k)}$ denote the position of the nonzero element in the jth column of $\tilde{\mathbf{P}}_k$. That is, the $(\tilde{p}_j^{(k)}, j)$th element of $\tilde{\mathbf{P}}_k$ is 1. Then by the construction of $\tilde{\mathbf{P}}$, the random variables $\{\tilde{p}_j^{(k)}\}_{k\in[s], j\in[n/s]}$ are i.i.d. uniformly distributed on $[m/s]$. For $j \in [n]$, let $\tau(j) \in [n]$ denote the index such that $\mathbf{G}_{\tau(j),j} = 1$. Then τ is a uniform permutation of $[n]$. For $j \in [n]$, the jth column of $\tilde{\mathbf{\Delta}}$ has s nonzeros, and the positions of these s nonzeros are

$$w_j^{(k)} := (k-1)\frac{m}{s} + \tilde{p}^{(k)}_{\tau(j)-\left\lfloor \frac{\tau(j)}{n/s} \right\rfloor \frac{n}{s}}, \quad k \in [s].$$

For any $j \in [n]$, the s positions $w_j^{(1)}, \ldots, w_j^{(s)}$ are independent. However, we emphasis that different columns of $\tilde{\mathbf{\Delta}}$ may not be independent. In fact, if two column indices j and j' satisfy

$$\tau(j) - \left\lfloor \frac{\tau(j)}{n/s} \right\rfloor \frac{n}{s} = \tau(j') - \left\lfloor \frac{\tau(j')}{n/s} \right\rfloor \frac{n}{s},$$

then the jth column and j'th column of $\tilde{\mathbf{\Delta}}$ share the same positions of nonzeros.

With careful analysis, we can prove that the condition (4) in Assumption 1 holds under certain conditions. Formally, we have the following proposition.

Proposition 1. *The $m \times n$ embedding matrix $\mathbf{\Pi}_{\text{NEW}}$ is (d, ϵ, δ)-OSE for*

$$\begin{aligned} m =& \Theta(d \log^6(d/\delta)/\epsilon^2), \\ s =& \Theta(\log^2(d/\delta)/\epsilon), \\ n =& \exp\left\{\Omega\left\{\log(d/\delta)\{\log(d/\epsilon) + |\log(\log(d/\delta))|\}\right\}\right\}. \end{aligned}$$

In Proposition 1, the choice of m and s achieves the same order as Theorem 1, and the choice of s improves Theorem 5 of Nelson and Nguyen (2013) by a logarithm factor. However, in Proposition 1, we make an additional assumption on n. Intuitively, this assumption avoids the case that n is small where the permutation of n rows of data matrix may not provide sufficient randomness for our purpose. In practice, subspace embedding is often applied when n is extremely large. In this view, this condition on n is reasonable. Moreover, we conjecture that this condition on n can be relaxed.

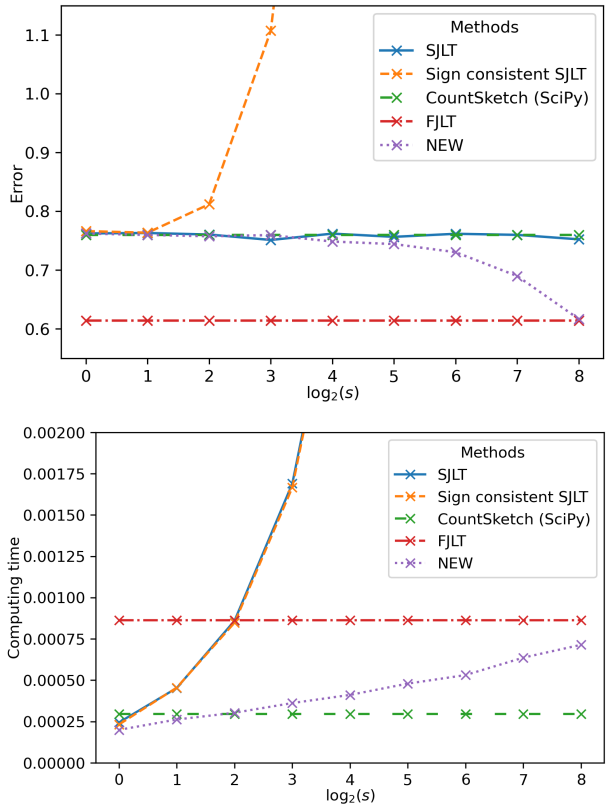

Figure 1: Error and computing time for various methods. Dense case. $n = 2^{10}$, $d = 2^5$, $m = 2^8$.

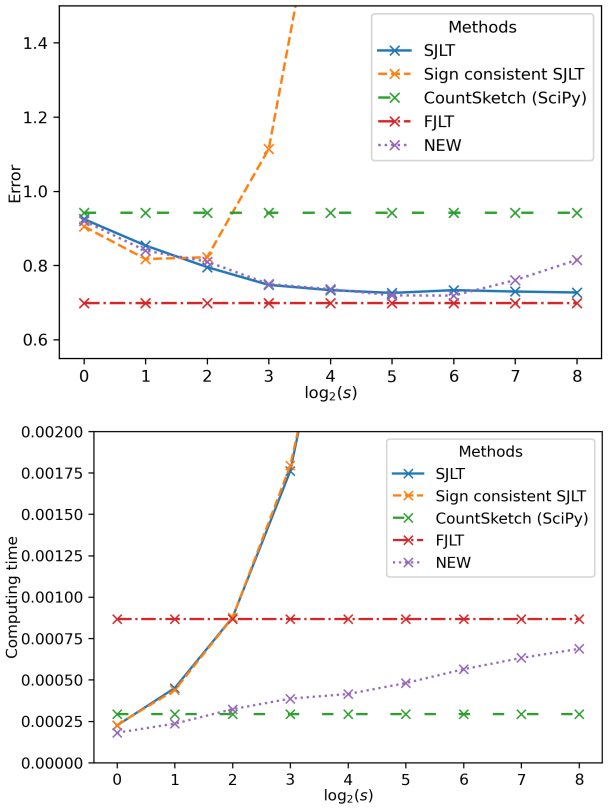

Figure 2: Error and computing time for various methods. Sparse case. $n = 2^{10}$, $d = 2^5$, $m = 2^8$.

Now we consider the computing time of the proposed OSE method. Suppose $\mathbf{X} \in \mathbb{R}^{n \times d}$. The permutation of the data costs $O(nd)$ time. The $\log_2(s)$ iterations costs $O(nd \log(s))$ time. Finally, the action of $\tilde{\mathbf{P}}$ costs $O(nd)$ time. Hence the total computing time is $O(nd \log(s))$. We note that $s = \Theta(\log^2(d/\delta)/\epsilon)$. In summary, the computation of $\mathbf{\Pi}_{\text{NEW}}\mathbf{X}$ can be completed in time

$$O\left(nd\left(\log\left(\frac{1}{\epsilon}\right) + \left|\log\left(\log\left(\frac{d}{\delta}\right)\right)\right|\right)\right).$$

This computing time is very close to $O(nd)$, and is faster than FJLT and SJLT for dense input matrice.

6 NUMERICAL EXPERIMENTS

In this section, we examine the performance of the proposed OSE method and compare it with FJLT, SJLT and sign-consistent SJLT. These methods are implemented in Python and run on a CPU with 3.00 GHz. For FJLT, Algorithm 2 is used. For SJLT and sign-consistent SJLT, we adopt the graph construction of $\mathbf{\Delta}$. For SJLT and sign-consistent SJLT, if $s = 1$, then they are equivalent to CountSketch in Clarkson and Woodruff (2013). For comparison, we also report

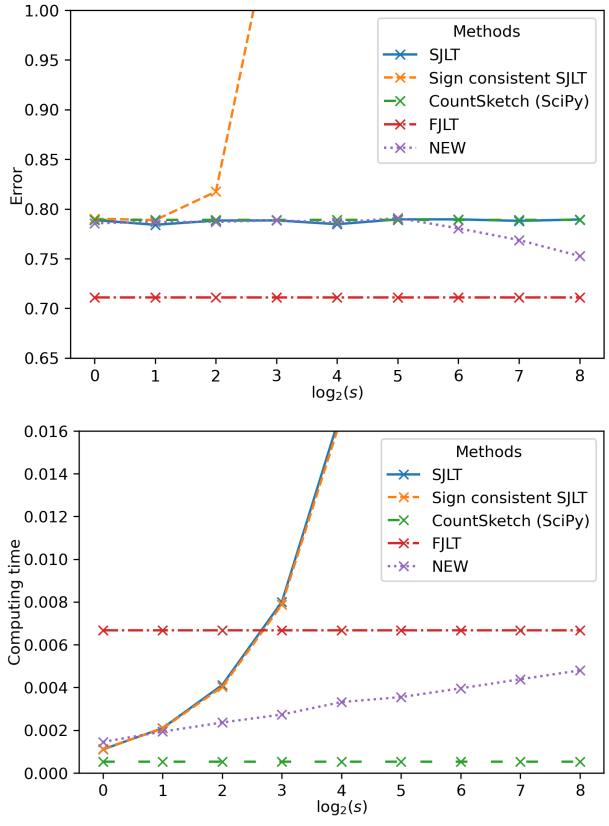

Figure 3: Error and computing time for various methods. Dense case. $n = 2^{12}$, $d = 2^6$, $m = 2^9$.

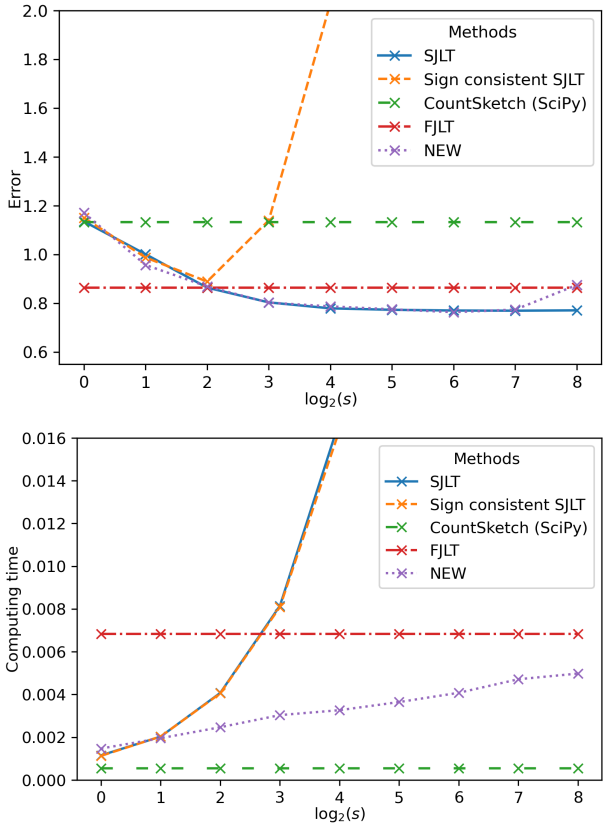

Figure 4: Error and computing time for various methods. Sparse case. $n = 2^{12}$, $d = 2^6$, $m = 2^9$.

the performance of CountSketch implemented in the library SciPy (Virtanen et al., 2020).

First we consider experiments for synthetic data. We consider two different $\mathbf{U}$. In the first case, we first generate an $n \times d$ random matrix whose elements are i.i.d. standard normal random variables. And we take $\mathbf{U}$ to be the left singular vector of this random matrix. We refer to this case the dense case. In the second case, we take $\mathbf{U} = (\mathbf{I}_d, \mathbf{O}_{d\times(n-d)})^\top$. We refer to this case the sparse case.

We use $\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\|$ to measure the errors of OSE methods. We illustrate the performance of various methods in Figures 1-4. The reported results are the average result of 200 replications. When $s = 1$, the proposed OSE method have similar error performance as SJLT, sign-consistent SJLT and CountSketch. As s increases, the proposed OSE can achieve smaller error than sign-consistent SJLT and CountSketch. Also, as s increases, the proposed OSE has a slowly increasing computing time and tends to be much faster than SJLT and sign-consistent SJLT. This phenomenon is well predicted by our theory. In fact, our theory implies that the computing time of the proposed OSE relies on s with the order $\log(s)$. In some cases, FJLT has smaller error than the proposed OSE method. Nevertheless, the proposed OSE is much faster than FJLT. In summary, the proposed method has attractive performance in both error and computing time.

Now we consider the experimental results on the MiniBooNE particle identification dataset in UCI Machine Learning Repository (Dua and Graff, 2017). This dataset contains $n = 130,064$ instances and $d = 50$ attributes. We add zero rows to this dataset such that $n = 2^{17}$ is a power of 2. Then we standardize each attribute to form the standardized data matrix $\mathbf{A}$. Finally we take the matrix $\mathbf{U} \in \mathbb{R}^{2^{17}\times 50}$ as the left singular vectors of $\mathbf{A}$. We take $m = 2^{10}$. The results are reported in Figure 5. For this dataset, the proposed OSE method has similar error performance as that of the FJLT and SJLT, but is much faster than these two methods.

7 DISCUSSIONS

In this work, we investigated the connection between FJLT and SJLT. We proposed an iterative algorithm for FJLT. This reveals an interesting connection between FJLT and SJLT. We modified this algorithm

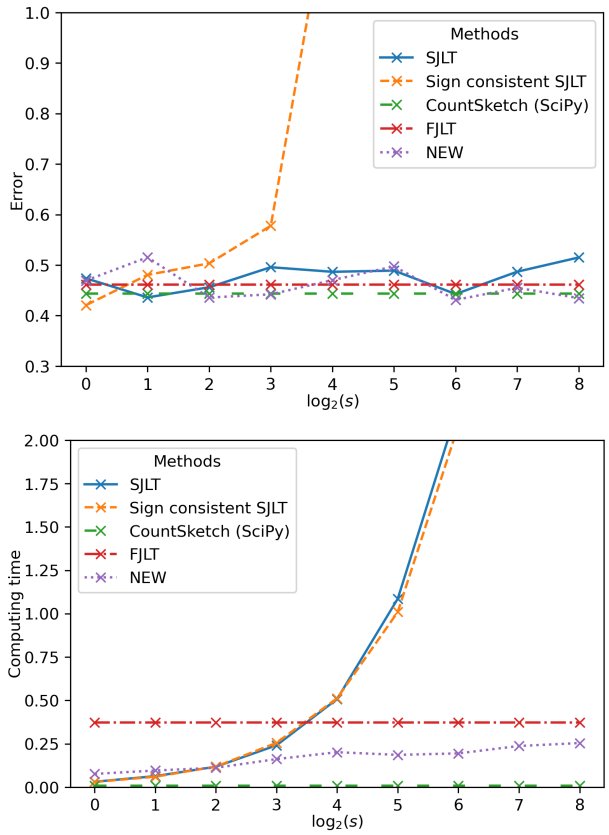

Figure 5: Error and computing time for various methods for MiniBooNE particle identification dataset.

and obtain a new subspace embedding method. The new subspace embedding method takes both advantages of FJLT and SJLT and is faster to apply than both FJLT and SJLT. We investigated the subspace embedding property of the proposed method. It shows that the proposed subspace embedding method is OSE with the same sparsity parameter as SJLT.

In the theoretical analysis of the OSE property of the proposed method, we impose the condition $n = \exp\{\Omega\{\log(d/\delta)\{\log(d/\epsilon) + |\log(\log(d/\delta))|\}\}\}$. We conjecture that this condition can be relaxed. We leave open the problem of establishing a better dependence on n.

Acknowledgements

The authors thank anonymous reviewers for their valuable comments and suggestions. This work was supported by National Natural Science Foundation of China (No 11971478) and Beijing Natural Science Foundation (No Z200001). Wangli Xu serves as the corresponding author of the present paper.

References

Achlioptas, D. (2003). Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687. Special issue on PODS 2001.

Ailon, N. and Chazelle, B. (2006). Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 557–563.

Ailon, N. and Chazelle, B. (2009). The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322.

Ailon, N. and Liberty, E. (2009). Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615–630.

Allen-Zhu, Z., Gelashvili, R., Micali, S., and Shavit, N. (2014). Sparse sign-consistent johnson–lindenstrauss matrices: Compression with neuroscience-based constraints. *Proceedings of the National Academy of Sciences*, 111(47):16872–16876.

Avron, H., Maymounkov, P., and Toledo, S. (2010). Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236.

Bourgain, J., Dirksen, S., and Nelson, J. (2015). Toward a unified theory of sparse dimensionality reduction in Euclidean space. *Geometric and Functional Analysis*, 25(4):1009–1088.

Boutsidis, C. and Gittens, A. (2013). Improved matrix algorithms via the subsampled randomized Hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1301–1340.

Braverman, V., Ostrovsky, R., and Rabani, Y. (2010). Rademacher chaos, random Eulerian graphs and the sparse Johnson-Lindenstrauss transform. arXiv:1011.2590.

Clarkson, K. L. and Woodruff, D. P. (2013). Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing Conference*, pages 81–90.

Cohen, M. B. (2016). Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–287.

Cohen, M. B., Nelson, J., and Woodruff, D. P. (2016). Optimal Approximate Matrix Product in Terms of Stable Rank. In *43rd International Colloquium*

on Automata, Languages, and Programming, volume 55, pages 11:1–11:14.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301.

Dasgupta, A., Kumar, R., and Sarlós, T. (2010). A sparse Johnson-Lindenstrauss transform. In Schulman, L. J., editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 341–350.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability*, volume 26 of *Contemp. Math.*, pages 189–206.

Kane, D. M. and Nelson, J. (2014). Sparser johnson-lindenstrauss transforms. *Journal of the ACM*, 61(1).

Kannan, R. and Vempala, S. (2017). Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135.

Lacotte, J., Liu, S., Dobriban, E., and Pilanci, M. (2020). Optimal iterative sketching methods with the subsampled randomized hadamard transform. In *Advances in Neural Information Processing Systems*, volume 33, pages 9725–9735.

Lacotte, J. and Pilanci, M. (2020a). Effective dimension adaptive sketching methods for faster regularized least-squares optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 19377–19387.

Lacotte, J. and Pilanci, M. (2020b). Optimal randomized first-order methods for least-squares problems. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5587–5597. PMLR.

Lu, Y., Dhillon, P. S., Foster, D., and Ungar, L. (2013). Faster ridge regression via the subsampled randomized hadamard transform. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, page 369–377.

Martinsson, P.-G. and Tropp, J. A. (2020). Randomized numerical linear algebra: foundations and algorithms. *Acta Numerica*, 29:403–572.

Meng, X. and Mahoney, M. W. (2013). Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Symposium on Theory of Computing Conference*, pages 91–100. ACM.

Nash-Williams, C. S. J. A. (1961). Edge-disjoint spanning trees of finite graphs. *The Journal of the London Mathematical Society*, 36:445–450.

Nelson, J. and Nguyen, H. L. (2013). OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings. In *54th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126.

Nelson, J. and Nguyên, H. L. (2014). Lower bounds for oblivious subspace embeddings. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP*, volume 8572 of *Lecture Notes in Computer Science*, pages 883–894. Springer.

Pilanci, M. and Wainwright, M. J. (2016). Iterative Hessian sketch: fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research (JMLR)*, 17:Paper No. 53, 38.

Rokhlin, V. and Tygert, M. (2008). A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences of the United States of America*, 105(36):13212–13217.

Sarlós, T. (2006). Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 143–152.

Tropp, J. A. (2011). Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis. Theory and Applications*, 3(1-2):115–126.

Tutte, W. T. (1961). On the problem of decomposing a graph into n connected factors. *The Journal of the London Mathematical Society*, 36:221–230.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Woodruff, D. P. (2014). Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):iv+157.

Yarlagadda, R. K. R. and Hershey, J. E. (1997). *Hadamard matrix analysis and synthesis*, volume 383. Kluwer Academic Publishers, Boston, MA.

Supplementary Material: On a Connection Between Fast and Sparse Oblivious Subspace Embeddings

A PROOF OF THEOREM 1

Let $\mathbf{U} \in \mathbb{R}^{n\times d}$ be a column orthogonal matrix. Let $\mathbf{u}_i \in \mathbb{R}^d$ be the ith row of $\mathbf{U}$, $j = 1, \ldots, n$. We need to prove that

$$\Pr(\|\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\| > \epsilon) \leq \delta.$$

From Markov's inequality, we have

$$\Pr(\|\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\| > \epsilon) = \Pr(\|\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\|^\ell > \epsilon^\ell) \leq \frac{1}{\epsilon^\ell} \operatorname{E} \operatorname{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d)^\ell),$$

where ℓ is a positive even integer that will be specified latter. Hence to prove the conclusion, a major task is to derive a good upper bound of $\operatorname{E} \operatorname{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d)^\ell)$. To compute this expectation, we need some notations. Define the index set

$$\Psi := \{(\mathbf{i}, \mathbf{j}, \mathbf{r}) : i_1, \ldots, i_\ell \in [n],\, j_1, \ldots, j_\ell \in [n],\, i_t \neq j_t,\, t \in [\ell],\, r_1, \ldots, r_\ell \in [m]\},$$

where $\mathbf{i} := (i_1, \ldots, i_\ell)$, $\mathbf{j} := (j_1, \ldots, j_\ell)$ and $\mathbf{r} := (r_1, \ldots, r_\ell)$ are vectors of indices. For $(\mathbf{i}, \mathbf{j}, \mathbf{r}) \in \Psi$, define

$$h_1(\mathbf{i}, \mathbf{j}, \mathbf{r}) = \prod_{t\in[\ell]} \delta_{r_t,i_t} \delta_{r_t,j_t} \sigma_{r_t,i_t} \sigma_{r_t,j_t}, \quad h_2(\mathbf{i}, \mathbf{j}, \mathbf{r}) = \prod_{t\in[\ell]} \langle \mathbf{u}_{j_t}, \mathbf{u}_{i_{t+1}} \rangle.$$

In the above expression, $i_{\ell+1}$ is understood as i_1. This convention will be used throughout our proof. Here we emphasis that $h_2(\mathbf{i}, \mathbf{j}, \mathbf{r})$ does not rely on $\mathbf{r}$. We add $\mathbf{r}$ as an argument of h_2 just for convenience.

Lemma 2. *For any positive integer ℓ, we have*

$$\operatorname{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d)^\ell) = \frac{1}{s^\ell} \sum_{(\mathbf{i},\mathbf{j},\mathbf{r})\in\Psi} h_1(\mathbf{i}, \mathbf{j}, \mathbf{r}) h_2(\mathbf{i}, \mathbf{j}, \mathbf{r}).$$

Proof. We have

$$\operatorname{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi}\mathbf{U} - \mathbf{I}_d)^\ell) = \operatorname{tr}\left\{(\mathbf{U}^\top (\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}_n)\mathbf{U})^\ell\right\} = \operatorname{tr}\left\{((\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}_n)\mathbf{U}\mathbf{U}^\top)^\ell\right\}.$$

It can be seen that

$$(\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}_n)_{i,j} = \begin{cases} \frac{1}{s} \sum_{r\in[m]} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j} & \text{if } i \neq j, \\ 0 & \text{if } i = j. \end{cases}$$

On the other hand, $(\mathbf{U}\mathbf{U}^\top)_{i,j} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle$. Thus,

$$((\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}_n)\mathbf{U}\mathbf{U}^\top)_{i,k} = \sum_{j\in[n]} (\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}_n)_{i,j} (\mathbf{U}\mathbf{U}^\top)_{j,k} = \frac{1}{s} \sum_{\substack{j\in[n] \\ j\neq i \\ r\in[m]}} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j} \langle \mathbf{u}_j, \mathbf{u}_k \rangle.$$

It follows that

$$\begin{aligned}\operatorname{tr}((\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d)^\ell) =& \sum_{i_1,\ldots,i_\ell\in[n]}\prod_{t\in[\ell]}((\mathbf{\Pi}^\top\mathbf{\Pi}-\mathbf{I}_n)\mathbf{U}\mathbf{U}^\top)_{i_t,i_{t+1}}\\ =&\frac{1}{s^\ell}\sum_{\substack{i_1,\ldots,i_\ell\in[n]\\ j_1,\ldots,j_\ell\in[n]\\ j_t\neq i_t,\,t\in[\ell]\\ r_1,\ldots,r_\ell\in[m]}}\prod_{t\in[\ell]}\left(\delta_{r_t,i_t}\delta_{r_t,j_t}\sigma_{r_t,i_t}\sigma_{r_t,j_t}\langle\mathbf{u}_{j_t},\mathbf{u}_{i_{t+1}}\rangle\right)\\ =&\frac{1}{s^\ell}\sum_{(\mathbf{i},\mathbf{j},\mathbf{r})\in\Psi}h_1(\mathbf{i},\mathbf{j},\mathbf{r})h_2(\mathbf{i},\mathbf{j},\mathbf{r}).\end{aligned}$$

This completes the proof. □

The behavior of $\mathrm{E}(h_1(\mathbf{i},\mathbf{j},\mathbf{r})h_2(\mathbf{i},\mathbf{j},\mathbf{r}))$ is determined by certain graph structures of $(\mathbf{i},\mathbf{j},\mathbf{r})$. We collect essential definitions and results in graph theory in Section B. For any $(\mathbf{i},\mathbf{j},\mathbf{r})\in\Psi$, we associate a bipartite multigraph with labeled edges, denoted as $\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$. The vertex numbers of the two parts of $\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$ are

$$v_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) :=\mathrm{Card}(\{i_1,\ldots,i_\ell,j_1,\ldots,j_\ell\}),\quad v_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) := \mathrm{Card}(\{r_1,\ldots,r_\ell\}).$$

The two vertex sets are defined as

$$V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) :=\{(1,i): i\in[v_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})]\},\quad V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) := \{(2,i): i\in[v_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})]\}.$$

The above definitions make sure that the elements of $V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ and $V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ have different first coordinates and consequently $V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})\cap V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})=\emptyset$. The edge set $E(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ of $\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$ is defined by Algorithm 4. Roughly speaking, e_{2t-1} connecting the vertices corresponding to i_t and r_t, and e_{2t} connecting the vertices corresponding to j_t and r_t. Here we note that by definition in Algorithm 4, the tth edge e_t is a function with domain $\{1,2\}$, and $e_t(1)$ and $e_t(2)$ are its endvertices in $V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ and $V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$, respectively. Algorithm 4 also returns two functions f_1 and f_2. The function f_1 records the correspondence between the vertex set $V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ and the index set $\{i_1,\ldots,i_\ell,j_1,\ldots,j_\ell\}$, and the function f_2 records the correspondence between the vertex set $V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$ and the index set $\{r_1,\ldots,r_\ell\}$.

We emphasis that the function $(\mathbf{i},\mathbf{j},\mathbf{r})\mapsto\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$ is not injective. Nevertheless, the function $\mathcal{A}_1:(\mathbf{i},\mathbf{j},\mathbf{r})\mapsto(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}},f_1,f_2)$ defined in Algorithm 4 is injective. In fact, the function $\mathcal{A}_2$ defined in Algorithm 5 is the inverse of $\mathcal{A}_1$. Precisely, if $(\mathbf{i},\mathbf{j},\mathbf{r})\in\Psi$, then $\mathcal{A}_2(\mathcal{A}_1(\mathbf{i},\mathbf{j},\mathbf{r}))=(\mathbf{i},\mathbf{j},\mathbf{r})$. It follows that $\mathcal{A}_1$ is injective on Ψ. Let $\mathscr{G}$ denote the range of the function $(\mathbf{i},\mathbf{j},\mathbf{r})\mapsto\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$ with domain Ψ. Furthermore, it follows from Algorithms 4 and 5 that for any $\mathcal{G}\in\mathscr{G}$,

$$\{(\mathbf{i},\mathbf{j},\mathbf{r})\in\Psi:\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}=\mathcal{G}\}=\left\{\mathcal{A}_2(\mathcal{G},f_1^\dagger,f_2^\dagger):\ f_1^\dagger\in[n]^{[v_1(\mathcal{G})]},\ f_1^\dagger\text{ is injective},\ f_2^\dagger\in[m]^{[v_2(\mathcal{G})]},\ f_2^\dagger\text{ is injective}\right\}. \quad (4)$$

Let $\mathscr{G}_2$ be the collection of graph G in $\mathscr{G}$ such that each vetex in $V_1(\mathcal{G})$ has even edge-degree. Using (4), we can derive the following proposition.

Proposition 2. *For any positive integer ℓ, we have*

$$\mathrm{E}\operatorname{tr}((\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d)^\ell)\le\frac{1}{s^\ell}\sum_{\mathcal{G}\in\mathscr{G}_2}\left|\sum_{\substack{f_2\in[m]^{[v_2(\mathcal{G})]}\\ f_2\text{ is injective}}}\mathrm{E}\prod_{b\in B(\mathcal{G})}\delta_{f_2(b(2)),b(1)}^{\alpha(b)}\right|\left|\sum_{\substack{f_1\in[n]^{[v_1(\mathcal{G})]}\\ f_1\text{ is injective}}}\prod_{t\in[\ell]}\langle\mathbf{u}_{f_1(e_{2t}(1))},\mathbf{u}_{f_1(e_{2t+1}(1))}\rangle\right|,$$

where e_t is the abbreviation of $(E(\mathcal{G}))(t)$ and relies on $\mathcal{G}$.

Algorithm 4: Construction of $\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}$

Function $\mathcal{A}_1(\mathbf{i},\mathbf{j},\mathbf{r})$**:**

$v_1 \leftarrow 0,\ f_1 \leftarrow \emptyset$
$v_2 \leftarrow 0,\ f_2 \leftarrow \emptyset$
for $t \leftarrow 1$ **to** ℓ **do**
  if $i_t \notin \mathrm{range}(f_1)$ **then**
    $v_1 \leftarrow v_1 + 1$
    $f_1 \leftarrow f_1 \cup \{(v_1, i_t)\}$
  if $r_t \notin \mathrm{range}(f_2)$ **then**
    $v_2 \leftarrow v_2 + 1$
    $f_2 \leftarrow f_2 \cup \{(v_2, r_t)\}$
  if $j_t \notin \mathrm{range}(f_1)$ **then**
    $v_1 \leftarrow v_1 + 1$
    $f_1 \leftarrow f_1 \cup \{(v_1, j_t)\}$
for $t \leftarrow 1$ **to** ℓ **do**
  $e_{2t-1} \leftarrow \{(1, f_1^{-1}(i_t)), (2, f_2^{-1}(r_t))\}$
  $e_{2t} \leftarrow \{(1, f_1^{-1}(j_t)), (2, f_2^{-1}(r_t))\}$
$V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) \leftarrow \{(1,i) : i \in [v_1]\}$
$V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) \leftarrow \{(2,i) : i \in [v_2]\}$
$V(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) \leftarrow V_1(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) \cup V_2(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}})$
$E(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}) \leftarrow \{(1,e_1), \ldots, (2\ell, e_{2\ell})\}$
$\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}} \leftarrow (V(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}), E(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}}))$
return $(\mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}},\ f_1,\ f_2)$

Proof. We have

$$\begin{aligned}
\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) =& \frac{1}{s^\ell} \sum_{(\mathbf{i},\mathbf{j},\mathbf{r}) \in \Psi} \{\mathrm{E}\, h_1(\mathbf{i},\mathbf{j},\mathbf{r})\} h_2(\mathbf{i},\mathbf{j},\mathbf{r}) \\
=& \frac{1}{s^\ell} \sum_{\mathcal{G} \in \mathscr{G}} \sum_{(\mathbf{i},\mathbf{j},\mathbf{r}) : \mathcal{G}_{\mathbf{i},\mathbf{j},\mathbf{r}} = \mathcal{G}} \{\mathrm{E}\, h_1(\mathbf{i},\mathbf{j},\mathbf{r})\} h_2(\mathbf{i},\mathbf{j},\mathbf{r}) \\
=& \frac{1}{s^\ell} \sum_{\mathcal{G} \in \mathscr{G}} \sum_{\substack{f_1 \in [n]^{[v_1(\mathcal{G})]} \\ f_2 \in [m]^{[v_2(\mathcal{G})]} \\ f_1, f_2 \text{ are injective}}} \{\mathrm{E}\, h_1(\mathcal{A}_2(\mathcal{G}, f_1, f_2))\} h_2(\mathcal{A}_2(\mathcal{G}, f_1, f_2)), \qquad (5)
\end{aligned}$$

where the first equality follows from Lemma 2 and the last equality follows from (4). We have

$$\begin{aligned}
\mathrm{E}\, h_1(\mathcal{A}_2(\mathcal{G}, f_1, f_2)) =& \mathrm{E} \prod_{(i,e) \in E(\mathcal{G})} \left(\delta_{f_2(e(2)), f_1(e(1))} \sigma_{f_2(e(2)), f_1(e(1))}\right) \\
=& \left(\mathrm{E} \prod_{b \in B(\mathcal{G})} \delta^{\alpha(b)}_{f_2(b(2)), f_1(b(1))}\right) \left(\mathrm{E} \prod_{b \in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)), f_1(b(1))}\right).
\end{aligned}$$

We have assumed that the distribution of the n columns $(\delta_{1,1}, \ldots, \delta_{m,1})^\top, \ldots, (\delta_{1,n}, \ldots, \delta_{m,n})^\top$ of $\mathbf{\Delta}$ is invariant under permutation. It follows that

$$\mathrm{E} \prod_{b \in B(\mathcal{G})} \delta^{\alpha(b)}_{f_2(b(2)), f_1(b(1))} = \mathrm{E} \prod_{p \in [v_1(\mathcal{G})]} \prod_{\substack{b \in B(\mathcal{G}) \\ b(1) = p}} \delta^{\alpha(b)}_{f_2(b(2)), f_1(p)} = \mathrm{E} \prod_{p \in [v_1(\mathcal{G})]} \prod_{\substack{b \in B(\mathcal{G}) \\ b(1) = p}} \delta^{\alpha(b)}_{f_2(b(2)), p} = \mathrm{E} \prod_{b \in B(\mathcal{G})} \delta^{\alpha(b)}_{f_2(b(2)), b(1)}.$$

Similarly, since the columns of $\mathbf{\Sigma}$ are independent, we have

$$\mathrm{E} \prod_{b \in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)), f_1(b(1))} = \mathrm{E} \prod_{b \in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)), b(1)}.$$

Algorithm 5: Converse of Algorithm 4

Function $\mathcal{A}_2(\mathcal{G},\, f_1,\, f_2)$**:**
  for $t \leftarrow 1$ **to** ℓ **do**
    $e_{2t-1} \leftarrow (E(\mathcal{G}))(2t-1)$
    $e_{2t} \leftarrow (E(\mathcal{G}))(2t)$
    $i_t \leftarrow f_1(e_{2t-1}(1))$
    $r_t \leftarrow f_2(e_{2t-1}(2))$
    $j_t \leftarrow f_1(e_{2t}(1))$
  $\mathbf{i} \leftarrow (i_1, \ldots, i_\ell)$
  $\mathbf{j} \leftarrow (j_1, \ldots, j_\ell)$
  $\mathbf{r} \leftarrow (r_1, \ldots, r_\ell)$
  return $(\mathbf{i}, \mathbf{j}, \mathbf{r})$

Therefore, $h_1(\mathcal{A}_2(\mathcal{G}, f_1, f_2))$ does not rely on the function f_1. We claim that if some vertex in $V_1(\mathcal{G})$ has odd edge-degree, then $\mathrm{E}(\prod_{b\in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)),f_1(b(1))}) = 0$. To prove this claim, first we consider the case that $\{\sigma_{i,j}\}$ are sign-consistent. In this case, we have

$$\mathrm{E} \prod_{b\in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)),f_1(b(1))} = \mathrm{E} \prod_{b\in B(\mathcal{G})} \sigma^{\alpha(b)}_{1,f_1(b(1))} = \prod_{v\in V_1(\mathcal{G})} \mathrm{E}\, \sigma^{\beta(v)}_{1,f_1(v)},$$

where $\beta(v)$ is the edge-degree of the vertex v. If $\beta(v)$ is odd for some $v \in V_1(\mathcal{G})$, then $\mathrm{E}\, \sigma^{\beta(v)}_{1,f_1(v)} = \mathrm{E}\, \sigma_{1,f_1(v)} = 0$, and the claim holds. Now we consider the case that $\sigma_{i,j}$ are i.i.d. If some vertex in $V_1(\mathcal{G})$ has odd edge-degree, then there exists a bond b incident on this vertex that has odd multiplicity. Then we have $\mathrm{E}\, \sigma^{\alpha(b)}_{f_2(b(2)),f_1(b(1))} = 0$, and the claim holds.

It follows from (5) and the above arguments that

$$\begin{aligned}
&\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \\
=&\frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} \sum_{\substack{f_2\in[m]^{[v_2(\mathcal{G})]} \\ f_2 \text{ is injective}}} \left\{ \left(\mathrm{E} \prod_{b\in B(\mathcal{G})} \delta^{\alpha(b)}_{f_2(b(2)),b(1)} \right) \left(\mathrm{E} \prod_{b\in B(\mathcal{G})} \sigma^{\alpha(b)}_{f_2(b(2)),b(1)} \right) \right\} \sum_{\substack{f_1\in[n]^{[v_1(\mathcal{G})]} \\ f_1 \text{ is injective}}} \prod_{t\in[\ell]} \langle \mathbf{u}_{f_1(e_{2t}(1))}, \mathbf{u}_{f_1(e_{2t+1}(1))} \rangle \\
\le&\frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} \left| \sum_{\substack{f_2\in[m]^{[v_2(\mathcal{G})]} \\ f_2 \text{ is injective}}} \mathrm{E} \prod_{b\in B(\mathcal{G})} \delta^{\alpha(b)}_{f_2(b(2)),b(1)} \right| \left| \sum_{\substack{f_1\in[n]^{[v_1(\mathcal{G})]} \\ f_1 \text{ is injective}}} \prod_{t\in[\ell]} \langle \mathbf{u}_{f_1(e_{2t}(1))}, \mathbf{u}_{f_1(e_{2t+1}(1))} \rangle \right|.
\end{aligned}$$

This completes the proof.

□

Now we deal with the quantity

$$\sum_{\substack{f_1\in[n]^{[v_1(\mathcal{G})]} \\ f_1 \text{ is injective}}} \prod_{t\in[\ell]} \langle \mathbf{u}_{f_1(e_{2t}(1))}, \mathbf{u}_{f_1(e_{2t+1}(1))} \rangle.$$

For any $\mathcal{G} \in \mathscr{G}_2$ with edges $(1, e_1), \ldots, (2\ell, e_{2\ell})$, we define a multigraph $\hat{\mathcal{G}}$ with labeled edges as follows:

$$\begin{aligned}
V(\hat{\mathcal{G}}) :=& \mathrm{range}(V_1(\mathcal{G})) = [v_1(\mathcal{G})], \\
E(\hat{\mathcal{G}}) :=& \{(1, \{e_2(1), e_3(1)\}), (2, \{e_4(1), e_5(1)\}), \ldots, (\ell, \{e_{2\ell}(1), e_1(1)\})\},
\end{aligned}$$

We have the following lemma.

Lemma 3. *Suppose $\mathcal{G} \in \mathscr{G}_2$. Then every vertex in $\hat{\mathcal{G}}$ has even edge-degree.*

Proof. By construction of $\mathcal{G}$ and $\hat{\mathcal{G}}$, for any $v \in V_1(\mathcal{G})$, the edge-degree of v in $\mathcal{G}$ is equal to the edge-degree of v in $\hat{\mathcal{G}}$. The conclusion follows. □

It can be seen that

$$\prod_{t=1}^{\ell} \langle \mathbf{u}_{f_1(e_{2t}(1))}, \mathbf{u}_{f_1(e_{2t+1}(1))} \rangle = \prod_{\substack{(p,e)\in E(\hat{\mathcal{G}}) \\ e=\{i,j\}}} \langle \mathbf{u}_{f_1(i)}, \mathbf{u}_{f_1(j)} \rangle.$$

With the above graph representation, we have the following proposition.

Proposition 3. *Suppose $\mathcal{G}$ is a multigraph with labeled edges whose vertex set is $V(\mathcal{G}) = [v(\mathcal{G})]$. Suppose $E(\mathcal{G}) \neq \emptyset$ and every vertex of $\mathcal{G}$ has even edge-degree. Then*

$$\Big| \sum_{\substack{f\in[n]^{[v(\mathcal{G})]} \\ f \text{ is injective}}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{u}_{f(j)} \rangle \Big| \leq v(\mathcal{G})! d^{w(\mathcal{G})}.$$

Remark 1. The proof of Proposition 3 is similar to the proof of Lemma 7 in Nelson and Nguyen (2013). For completeness, we provide its proof in Section A.1. Compared with the proof of Nelson and Nguyen (2013), our proof is a little simplified and some small gaps are fixed.

From Propositions 2, 3 and Lemma 3, we have

$$\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq \frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} m^{v_2(\mathcal{G})} v_1(\mathcal{G})! d^{w(\hat{\mathcal{G}})} \left(\max_{\substack{f_2\in[m]^{[v_2(\mathcal{G})]} \\ f_2 \text{ is injective}}} \mathrm{E} \prod_{b\in B(\mathcal{G})} |\delta_{f_2(b(2)),b(1)}| \right).$$

By assumption, we have

$$0 \leq \mathrm{E} \prod_{b\in B(\mathcal{G})} |\delta_{f_2(b(2)),b(1)}| \leq \left(c\frac{s}{m}\right)^{b(\mathcal{G})}.$$

It follows that

$$\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq \frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} m^{v_2(\mathcal{G})} v_1(\mathcal{G})! d^{w(\hat{\mathcal{G}})} \left(c\frac{s}{m}\right)^{b(\mathcal{G})}. \tag{6}$$

Lemma 4. *Suppose $\mathcal{G} \in \mathscr{G}_2$. Then $w(\hat{\mathcal{G}}) \leq b(\mathcal{G}) - v_2(\mathcal{G}) + 1$.*

Proof. Note that $\mathcal{G} \cup \hat{\mathcal{G}}$ is connected. Hence $\hat{\mathcal{G}}$ must have at least $w(\mathcal{G}) - 1$ bonds to connect the $w(\mathcal{G})$ connected components of $\mathcal{G}$. These $w(\mathcal{G}) - 1$ bonds each reduces the number of connected component of $\hat{\mathcal{G}}$ by 1. Thus, $w(\hat{\mathcal{G}}) \leq v_1(\mathcal{G}) - w(\mathcal{G}) + 1$. Note that for $\mathcal{G} \in \mathscr{G}$, we have $v_1(\mathcal{G}) + v_2(\mathcal{G}) \leq b(\mathcal{G}) + w(\mathcal{G})$. Summing the above two inequalities yields the conclusion. □

From (6) and Lemma 4, we have

$$\begin{aligned}
\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq & \frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} v_1(\mathcal{G})! (cs)^{b(\mathcal{G})} \left(\frac{d}{m}\right)^{b(\mathcal{G})-v_2(\mathcal{G})} \frac{1}{d^{b(\mathcal{G})-v_2(\mathcal{G})-w(\hat{\mathcal{G}})}} \\
\leq & d\frac{1}{s^\ell} \sum_{\mathcal{G}\in\mathscr{G}_2} v_1(\mathcal{G})! (cs)^{b(\mathcal{G})} \left(\frac{d}{m}\right)^{b(\mathcal{G})-v_2(\mathcal{G})}.
\end{aligned} \tag{7}$$

Now we consider some basic combinatorics. Since each vertex in $V_2(\mathcal{G})$ associates with at least two distinct bonds, we have $b(\mathcal{G}) \geq 2v_2(\mathcal{G})$. Also, we have $v_1(\mathcal{G}) \leq b(\mathcal{G}) \leq e(\mathcal{G}) = 2\ell$.

Lemma 5. *The number of different $\mathcal{G} \in \mathscr{G}_2$ with given bond number b, left vertex number v_1 and right vertex number v_2 is no more than $(ev_2)^b b^{2\ell}/(v_1!)$.*

Proof. Each bond has $v_1 v_2$ choices. Hence there are at most $\binom{v_1 v_2}{b}$ choices of b bonds. After we choose the bonds, the 2ℓ edges are all picked from these b bonds, and there are at most $b^{2\ell}$ choices of edges. The above choice method can pick all $\mathcal{G}$ with given v_1, v_2 and b. However, as illustrated in Algorithms 4 and 5, this procedure counts each $\mathcal{G}$ in $\mathscr{G}_2$ such graph for at least $v_1! v_2!$ times. In summary, the number of different $\mathcal{G} \in \mathscr{G}_2$ with given v_1, v_2 and b is at most

$$\frac{\binom{v_1 v_2}{b} b^{2\ell}}{v_1! v_2!} \leq \frac{(v_1 v_2)^b b^{2\ell}}{b! v_1! v_2!} \leq \frac{(v_1 v_2)^b b^{2\ell}}{b! v_1!} \leq \frac{(v_1 v_2)^b b^{2\ell}}{v_1! \sqrt{2\pi b} (b/e)^b} \leq \frac{(e v_2)^b b^{2\ell}}{v_1!},$$

where the second last inequality follows from Stirling's formula. This completes the proof.

□

From (7) and Lemma 5, we have

$$\begin{aligned}
\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq & d \frac{1}{s^\ell} \sum_{b \in [2\ell]} \sum_{v_1 \in [2\ell]} \sum_{v_2 \in [\ell]} (e v_2)^b b^{2\ell} (cs)^b \left(\frac{d}{m}\right)^{b - v_2} \\
\leq & d \frac{1}{s^\ell} \sum_{b \in [2\ell]} \sum_{v_1 \in [2\ell]} \sum_{v_2 \in [\ell]} (e\ell)^b b^{2\ell} (cs)^b \left(\frac{d}{m}\right)^{\frac{b}{2}} \\
= & d \frac{1}{s^\ell} \sum_{b \in [2\ell]} \sum_{v_1 \in [2\ell]} \sum_{v_2 \in [\ell]} b^{2\ell} \left(\frac{c^2 e^2 \ell^2 s^2 d}{m}\right)^{\frac{b}{2}}.
\end{aligned}$$

Pick m such that $(c^2 e^2 \ell^2 s^2 d)/m \leq 1$. Then we have

$$\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq \frac{4 d \ell^3 (2\ell)^{2\ell}}{s^\ell}.$$

Then for $s \geq \epsilon^{-1} (2\ell)^2 e$, we have

$$\mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq 4 d \ell^3 \left(\frac{\epsilon}{e}\right)^\ell.$$

Then by Markov's inequality,

$$\Pr(\|\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d\| > \epsilon) \leq \frac{1}{\epsilon^\ell} \mathrm{E}\,\mathrm{tr}((\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d)^\ell) \leq \frac{4 d \ell^3}{e^\ell}.$$

The above probability is bounded by δ if we pick $\ell = \Theta(\log(d/\delta))$. This completes the proof.

A.1 Proof of Proposition 3

We would like to prove a more general conclusion. For an edge $(p, \{i, j\}) \in \mathcal{G}$, we associate two matrices $\mathbf{M}_{p,i,j}, \mathbf{M}_{p,j,i} \in \mathbb{R}^{d \times d}$ such that $\mathbf{M}_{p,i,j} = \mathbf{M}_{p,j,i}^\top$ and $\|\mathbf{M}_{p,i,j}\| \leq 1$. We shall prove that

$$\left| \sum_{\substack{f \in [n]^{[v(\mathcal{G})]} \\ f \text{ is injective}}} \prod_{(p,\{i,j\}) \in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \right| \leq v(\mathcal{G})! d^{w(\mathcal{G})}. \tag{8}$$

It can be seen that the original conclusion corresponds to the case $\mathbf{M}_{p,i,j} = \mathbf{I}_d$. The condition $\mathbf{M}_{p,i,j} = \mathbf{M}_{p,j,i}^\top$ implies that

$$\langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle = \langle \mathbf{u}_{f(j)}, \mathbf{M}_{p,i,j}^\top \mathbf{u}_{f(i)} \rangle = \langle \mathbf{u}_{f(j)}, \mathbf{M}_{p,j,i} \mathbf{u}_{f(i)} \rangle.$$

That is, the term $\langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{(j)} \rangle$ does not rely on the order of i and j. Thus, the left hand side of (8) is well-defined.

Lemma 6. *Let $\mathcal{G}$ be a multigraph with labeled edges whose vertex set is a finite subset of $\{1,2,\ldots\}$. Suppose $\mathcal{G}$ is connected, $E(\mathcal{G})\neq\emptyset$. Let $\hat{v}$ be any fixed vertex of $\mathcal{G}$, k be any fixed integer in $[n]$, and $\mathbf{c}$ be any fixed vector in $\mathbb{R}^d$ with $\|\mathbf{c}\|\leq 1$. For $f\in[n]^{V(\mathcal{G})\setminus\{\hat{v}\}}$, let $\mathbf{v}_{f,i}=\mathbf{u}_{f(i)}$, $i\in V(\mathcal{G})\setminus\{\hat{v}\}$ and $\mathbf{v}_{f,\hat{v}}=\mathbf{c}\in\mathbb{R}^d$. Then we have*

$$\sum_{f\in[n]^{V(\mathcal{G})\setminus\{\hat{v}\}}}\prod_{(p,\{i,j\})\in E(\mathcal{G})}\langle\mathbf{v}_{f,i},\mathbf{M}_{p,i,j}\mathbf{v}_{f,j}\rangle^2\leq\|\mathbf{c}\|^2.$$

Proof. We have assumed that $\mathcal{G}$ is connected. From Lemma 10, there is a bijection π from $[v(\mathcal{G})]$ onto $V(\mathcal{G})$ such that $\pi(v(\mathcal{G}))=\hat{v}$ and for any $i\in[v(\mathcal{G})-1]$, the edge set

$$E_i:=\{(p,\{\pi(i),\pi(j)\})\in E(\mathcal{G}):j\in\{i+1,\ldots,v(\mathcal{G})\}\}$$

is not empty. For $i\in[v(\mathcal{G})-1]$, we pick a $\tau(i)>i$ such that $(p_i,\{\pi(i),\pi(\tau(i))\})\in E_i$. Note that $\tau(v(\mathcal{G})-1)>v(\mathcal{G})$. Hence $\tau(v(\mathcal{G})-1)=v(\mathcal{G})$ and $\pi(\tau(v(\mathcal{G})-1))=\hat{v}$. For any $f\in[n]^{V(\mathcal{G})\setminus\{\hat{v}\}}$, we have

$$\prod_{(p,\{i,j\})\in E(\mathcal{G})}\langle\mathbf{v}_{f,i},\mathbf{M}_{p,i,j}\mathbf{v}_{f,j}\rangle^2\leq\prod_{i=1}^{v(\mathcal{G})-1}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2.$$

It follows that

$$\sum_{f\in[n]^{V(\mathcal{G})\setminus\{\hat{v}\}}}\prod_{(p,\{i,j\})\in E(\mathcal{G})}\langle\mathbf{v}_{f,i},\mathbf{M}_{p,i,j}\mathbf{v}_{f,j}\rangle^2\leq\sum_{\substack{f(\pi(1))\in[n]\\ \cdots\\ f(\pi(v(\mathcal{G})-1))\in[n]}}\prod_{i=1}^{v(\mathcal{G})-1}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2.$$

For $i\in[v(\mathcal{G})-1]$, we have

$$\begin{aligned}
&\sum_{f(\pi(i))\in[n]}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2\\
=&\mathbf{v}_{f,\pi(\tau(i))}^\top\mathbf{M}_{p_i,\pi(\tau(i)),\pi(i)}\left(\sum_{f(\pi(i))\in[n]}\mathbf{v}_{f,\pi(i)}\mathbf{v}_{f,\pi(i)}^\top\right)\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\\
=&\|\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\|^2\\
\leq&1.
\end{aligned}$$

Applying the above inequality recursively yields

$$\begin{aligned}
&\sum_{\substack{f(\pi(1))\in[n]\\ \cdots\\ f(\pi(v(\mathcal{G})-1))\in[n]}}\prod_{i=1}^{v(\mathcal{G})-1}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2\\
=&\sum_{\substack{f(\pi(2))\in[n]\\ \cdots\\ f(\pi(v(\mathcal{G})-1))\in[n]}}\left(\sum_{f(\pi(1))\in[n]}\langle\mathbf{v}_{f,\pi(1)},\mathbf{M}_{p_1,\pi(1),\pi(\tau(1))}\mathbf{v}_{f,\pi(\tau(1))}\rangle^2\right)\prod_{i=2}^{v(\mathcal{G})-1}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2\\
\leq&\sum_{\substack{f(\pi(2))\in[n]\\ \cdots\\ f(\pi(v(\mathcal{G})-1))\in[n]}}\prod_{i=2}^{v(\mathcal{G})-1}\langle\mathbf{v}_{f,\pi(i)},\mathbf{M}_{p_i,\pi(i),\pi(\tau(i))}\mathbf{v}_{f,\pi(\tau(i))}\rangle^2\\
&\vdots\\
\leq&\sum_{f(\pi(v(\mathcal{G})-1))\in[n]}\langle\mathbf{v}_{f,\pi(v(\mathcal{G})-1)},\mathbf{M}_{p_{v(\mathcal{G})-1},\pi(v(\mathcal{G})-1),\hat{v}}\mathbf{v}_{f,\hat{v}}\rangle^2\\
\leq&\|\mathbf{v}_{f,\hat{v}}\|^2.
\end{aligned}$$

This completes the proof. □

Lemma 7. *Suppose $\mathcal{G}$ is a multigraph with labeled edges whose vertex set is a finite subset of $\{1,2,\dots\}$. Suppose $\mathcal{G}$ is Eulerian and $E(\mathcal{G}) \neq \emptyset$. Then*

$$\Big| \sum_{f\in[n]^{V(\mathcal{G})}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big| \leq d.$$

Proof. We prove the conclusion by induction on $v(\mathcal{G})$. If $v(\mathcal{G}) = 1$, then

$$\Big| \sum_{f\in[n]^{V(\mathcal{G})}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big| \leq \sum_{i=1}^{n} \|\mathbf{u}_i\|^2 = \|\mathbf{U}\|_F^2 = d.$$

And the conclusion holds. Below we consider the general case of $v(\mathcal{G}) \geq 2$.

First we consider the case that $\mathcal{G}$ has two edge-disjoint spanning trees, denoted as $\mathcal{T}_1$ and $\mathcal{T}_2$. Then the graph $(V(\mathcal{G}), E(\mathcal{T}_1))$ is connected. On the other hand, since $E(\mathcal{T}_2) \subset E(\mathcal{G})\backslash E(\mathcal{T}_1)$, the graph $(V(\mathcal{G}), E(\mathcal{G})\backslash E(\mathcal{T}_1))$ is also connected. We have

$$\begin{aligned}
&\Big| \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big| \\
=&\Big| \prod_{(p,\{i,j\})\in E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big| \Big| \prod_{(p,\{i,j\})\in E(\mathcal{G})\backslash E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big| \\
\leq&\frac{1}{2} \prod_{(p,\{i,j\})\in E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle^2 + \frac{1}{2} \prod_{(p,\{i,j\})\in E(\mathcal{G})\backslash E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle^2.
\end{aligned}$$

Applying Lemma 6 to the graphs $(V(\mathcal{G}), E(\mathcal{T}_1))$ and $(V(\mathcal{G}), E(\mathcal{G})\backslash E(\mathcal{T}_1))$ leads to

$$\left| \sum_{f\in[n]^{V(\mathcal{G})}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \right| \leq \sum_{i=1}^{n} \|\mathbf{u}_i\|^2 = \|\mathbf{U}\|_F^2 = d.$$

Hence the conclusion holds.

Now we consider the case that $\mathcal{G}$ does not have two edge-disjoint spanning trees. Suppose $S^* \subset V(\mathcal{G})$ satisfies the two properties of Lemma 12. Let $(\tilde{p}, \{g, h\})$ and $(\tilde{p}', \{g', h'\})$ be the only two edges connecting S^* and $V(\mathcal{G})\backslash S^*$ where $g, g' \in V(\mathcal{G})\backslash S^*$ and $h, h' \in S^*$. Then

$$\begin{aligned}
&\sum_{f\in[n]^{V(\mathcal{G})}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \\
=&\sum_{f\in[n]^{V(\mathcal{G})}} \left(\prod_{(p,\{i,j\})\in E(\mathcal{G}(V(\mathcal{G})\backslash S^*))} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \right) \left(\prod_{(p,\{i,j\})\in E(\mathcal{G}(S^*))} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \right) \\
&\cdot \langle \mathbf{u}_{f(g)}, \mathbf{M}_{\tilde{p},g,h}\mathbf{u}_{f(h)}\rangle \langle \mathbf{u}_{f(g')}, \mathbf{M}_{\tilde{p}',g',h'}\mathbf{u}_{f(h')}\rangle \\
=&\sum_{f\in[n]^{V(\mathcal{G})\backslash S^*}} \Big(\prod_{(p,\{i,j\})\in E(\mathcal{G}(V(\mathcal{G})\backslash S^*))} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \Big) \langle \mathbf{u}_{f(g)}, \tilde{\mathbf{M}}\mathbf{u}_{f(g')}\rangle,
\end{aligned}$$

where

$$\tilde{\mathbf{M}} = \sum_{f\in[n]^{S^*}} \mathbf{M}_{\tilde{p},g,h}\mathbf{u}_{f(h)} \left(\prod_{(p,\{i,j\})\in E(\mathcal{G}(S^*))} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j}\mathbf{u}_{f(j)}\rangle \right) \mathbf{u}_{f(h')}^{\top} \mathbf{M}_{\tilde{p}',h',g'}.$$

If $\|\tilde{\mathbf{M}}\| \leq 1$, then we add an edge $(\hat{p}, \{g, g'\})$ to the graph $V(\mathcal{G})\backslash S^*$ and define $\mathbf{M}_{\hat{p},g,g'} := \tilde{\mathbf{M}}$. Then the above sum reduces to the sum for this new graph. Note that this new graph has fewer vertices than $\mathcal{G}$ and is still Eulerian. Then the conclusion follows from the induction hypothesis.

It remains to prove that $\|\tilde{\mathbf{M}}\| \le 1$. The matrix $\tilde{\mathbf{M}}$ is associated with the graph $\mathcal{G}(S^*)$. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be edge-disjoint spanning trees of S^*. For any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ such that $\|\mathbf{x}_1\| = \|\mathbf{x}_2\| = 1$, we have

$$\begin{aligned}
\mathbf{x}_1^\top \tilde{\mathbf{M}} \mathbf{x}_2 =& \sum_{f\in[n]^{S^*}} \langle \mathbf{x}_1, \mathbf{M}_{\tilde{p},g,h} \mathbf{u}_{f(h)} \rangle \prod_{(p,\{i,j\})\in E(\mathcal{G}(S^*))} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \langle \mathbf{u}_{f(h')}, \mathbf{M}_{\tilde{p}',h',g'} \mathbf{x}_2 \rangle \\
=& \sum_{f\in[n]^{S^*}} \Big(\langle \mathbf{x}_1, \mathbf{M}_{\tilde{p},g,h} \mathbf{u}_{f(h)} \rangle \prod_{(p,\{i,j\})\in E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \Big) \\
& \cdot \Big(\langle \mathbf{u}_{f(h')}, \mathbf{M}_{\tilde{p}',h',g'} \mathbf{x}_2 \rangle \prod_{(p,\{i,j\})\in E(\mathcal{G}(S^*))\setminus E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \Big) \\
\le& \frac{1}{2} \sum_{f\in[n]^{S^*}} \langle \mathbf{x}_1, \mathbf{M}_{\tilde{p},g,h} \mathbf{u}_{f(h)} \rangle^2 \prod_{(p,\{i,j\})\in E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle^2 \\
& + \frac{1}{2} \sum_{f\in[n]^{S^*}} \langle \mathbf{u}_{f(h')}, \mathbf{M}_{\tilde{p}',h',g'} \mathbf{x}_2 \rangle^2 \prod_{(p,\{i,j\})\in E(\mathcal{G}(S^*))\setminus E(\mathcal{T}_1)} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle^2 \\
\le& \frac{1}{2}\|\mathbf{x}_1\|^2 + \frac{1}{2}\|\mathbf{x}_2\|^2 \\
=& 1,
\end{aligned}$$

where the second last line follows from Lemma 6. This completes the proof.

□

Lemma 8. *Suppose $\mathcal{G}$ is a multigraph with labeled edges whose vertex set is a finite subset of $\{1, 2, \ldots\}$. Suppose $E(\mathcal{G}) \neq \emptyset$ and every vertex has even edge-degree. Then*

$$\Big| \sum_{f\in[n]^{V(\mathcal{G})}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \Big| \le d^{w(\mathcal{G})}.$$

Proof. Since every vertex of $\mathcal{G}$ has even edge-degree, we can decompose $\mathcal{G}$ into $w(\mathcal{G})$ disjoint Eulerian subgraphs. Then the conclusion follows by applying Lemma 7 to each of the subgraphs. □

We are now ready to prove (8). For $1 \le j \le i \le v(\mathcal{G})$, let $\kappa_{i,j} = \mathbf{1}_{\{f(i)=f(j)\}}$. Then

$$\mathbf{1}_{\{f(1),\ldots,f(v(\mathcal{G})) \text{ are distinct}\}} = \prod_{i=2}^{v(\mathcal{G})} \left(1 - \sum_{j=1}^{i-1} \mathbf{1}_{\{f(i)=f(j)\}} \right) = \sum_{\substack{q\in[v(\mathcal{G})]^{[v(\mathcal{G})]} \\ q(1)\le 1 \\ q(2)\le 2 \\ \cdots \\ q(v(\mathcal{G}))\le v(\mathcal{G})}} (-1)^{\gamma(q)} \prod_{i=1}^{v(\mathcal{G})} \mathbf{1}_{\{f(i)=f(q(i))\}},$$

where $\gamma(q) = \sum_{i=1}^{v(\mathcal{G})} \mathbf{1}_{\{q(i)<i\}}$. Hence we have

$$\begin{aligned}
& \Big| \sum_{\substack{f\in[n]^{[v(\mathcal{G})]} \\ f \text{ is injective}}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \Big| \\
=& \Big| \sum_{f\in[n]^{[v(\mathcal{G})]}} \sum_{\substack{q\in[v(\mathcal{G})]^{[v(\mathcal{G})]} \\ q(1)\le 1 \\ q(2)\le 2 \\ \cdots \\ q(v(\mathcal{G}))\le v(\mathcal{G})}} (-1)^{\gamma(q)} \prod_{\ell=1}^{v(\mathcal{G})} \mathbf{1}_{\{f(\ell)=f(q(\ell))\}} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \Big| \\
\le& \sum_{\substack{q\in[v(\mathcal{G})]^{[v(\mathcal{G})]} \\ q(1)\in[1] \\ q(2)\in[2] \\ \cdots \\ q(v(\mathcal{G}))\in[v(\mathcal{G})]}} \left| \sum_{f\in[n]^{[v(\mathcal{G})]}} \left\{ \prod_{\ell=1}^{v(\mathcal{G})} \mathbf{1}_{\{f(\ell)=f(q(\ell))\}} \right\} \prod_{(p,\{i,j\})\in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \right|.
\end{aligned}$$

Given a function q, we can obtain an induced graph of $\mathcal{G}$ by combining the vertices ℓ and $q(\ell)$ as a single vertex, $i \in [v(\mathcal{G})]$. For the induced graph, every vertex also has even edge-degree. We apply Lemma 8 to the induced graph. Then

$$\left| \sum_{f \in [n]^{[v(\mathcal{G})]}} \left\{ \prod_{\ell=1}^{v(\mathcal{G})} \mathbf{1}_{\{f(\ell)=f(q(\ell))\}} \right\} \prod_{(p,\{i,j\}) \in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \right|$$

is upper bounded by d^w where w is the number of connected components of the graph induced by q. But the number of connected components of the induced graph is smaller than that of the original graph $\mathcal{G}$. Consequently, the above quantity is also upper bounded by $d^{w(\mathcal{G})}$. It follows that

$$\left| \sum_{\substack{f \in [n]^{[v(\mathcal{G})]} \\ f \text{ is injective}}} \prod_{(p,\{i,j\}) \in E(\mathcal{G})} \langle \mathbf{u}_{f(i)}, \mathbf{M}_{p,i,j} \mathbf{u}_{f(j)} \rangle \right| \leq \sum_{\substack{q(1) \in [1] \\ q(2) \in [2] \\ \vdots \\ q(v(\mathcal{G})) \in [v(\mathcal{G})]}} d^{w(\mathcal{G})} = v(\mathcal{G})! d^{w(\mathcal{G})}.$$

This completes the proof.

B RESULTS IN GRAPH THEORY

Our proofs of main results involve some arguments of graph theory. In order for our proofs of main results to be understood correctly, we collect definitions and results in graph theory that are used in our proofs of main results. Throughout our proofs, graphs are understood as *multigraph with labeled edges* whose rigorous definition is as follows.

Definition 1. A multigraph $\mathcal{G}$ with labeled edges is an ordered pair $(V(\mathcal{G}), E(\mathcal{G}))$ where $V(\mathcal{G})$ is the finite set of vertices and $E(\mathcal{G})$ is a function from a finite subset of $\{1, 2, \ldots\}$ to the set

$$\{\{v_1, v_2\} : v_1, v_2 \in V(\mathcal{G})\}.$$

Suppose $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ is a multigraph with labeled edges. Let $v(\mathcal{G}) := \mathrm{Card}(V(\mathcal{G}))$ denote the vertex number of $\mathcal{G}$. By definition, $E(\mathcal{G})$ is a function. Recall that a function is understood as its set-theoretic definition. That is, the function $E(\mathcal{G})$ is a set, and its elements are something like $(1, \{v_1, v_2\})$, $(2, \{v_3, v_4\})$, etc. We call $E(\mathcal{G})$ the edge set of $\mathcal{G}$. Let $e(\mathcal{G}) := \mathrm{Card}(E(\mathcal{G}))$ denote the edge number of $\mathcal{G}$. The domain of $E(\mathcal{G})$ is the set of labels of edges. From the property of function, different edges in $E(\mathcal{G})$ have different labels. We note that $\mathrm{range}(E(\mathcal{G})) \subset V(\mathcal{G}) \times V(\mathcal{G})$ is the unlabeled edges. Define the *bond set* of $\mathcal{G}$ as

$$B(\mathcal{G}) := \mathrm{range}(E(\mathcal{G})) \backslash \{\{i\} : i \in V(\mathcal{G})\}.$$

That is, the bond set of $\mathcal{G}$ contains unlabeled edges and excludes self loops. The elements in $B(\mathcal{G})$ are called *bonds*. Let $b(\mathcal{G}) := \mathrm{Card}(B(\mathcal{G}))$ denote the number of bonds of $\mathcal{G}$. For $b \in B(\mathcal{G})$, let $\alpha(b)$ denote the multiplicity of b. It can be seen that $(V(\mathcal{G}), B(\mathcal{G}))$ is a simple graph in the usual sense. We refer to the number of bonds a vertex is incident upon as its *bond-degree*, and the number of edges it is incident upon as its *edge-degree*. Let $w(\mathcal{G})$ be the number of connected components in the simple graph $(V(\mathcal{G}), B(\mathcal{G}))$.

Lemma 9. *For any multigraph $\mathcal{G}$ with labeled edges,*

$$v(\mathcal{G}) \leq b(\mathcal{G}) + w(\mathcal{G}).$$

Proof. For each of w connected components, the vertex number is not larger than the bond number plus 1. Summing w such inequalities yields the conclusion. □

Definition 2. A multigraph $\mathcal{G}$ with labeled edges is *bipartite* if there are disjoint sets of vertices $V_1(\mathcal{G})$ and $V_2(\mathcal{G})$ such that $V(\mathcal{G}) = V_1(\mathcal{G}) \cup V_2(\mathcal{G})$ and every edge joins a vertex of $V_1(\mathcal{G})$ to a vertex of $V_2(\mathcal{G})$. Let $v_i(\mathcal{G}) := \mathrm{Card}(V_i(\mathcal{G}))$ denote the vertex number of $V_i(\mathcal{G})$, $i = 1, 2$.

A multigraph $\mathcal{G}$ with labeled edges is *Eulerian* if the edges of $\mathcal{G}$ can be rearranged to a closed trail $(p_1, \{i_1, i_2\})$, $(p_2, \{i_2, i_3\})$, $\ldots$, $(p_{e(\mathcal{G})}, \{i_{e(\mathcal{G})}, i_1\})$. It is known that $\mathcal{G}$ is Eulerian if and only if every vertex of $\mathcal{G}$ has even edge-degree.

Suppose $\mathcal{G}$ is a multigraph with labeled edges. A *subgraph* $\mathcal{G}^\dagger = (V^\dagger, E^\dagger)$ of $\mathcal{G}$ is a multigraph with labeled edges such that $V^\dagger \subset V(\mathcal{G})$ and $E^\dagger \subset E(\mathcal{G})$. If $E^\dagger$ contains all edges in $E(\mathcal{G})$ that join two vertices in $V^\dagger$, then $\mathcal{G}^\dagger$ is said to be the subgraph *induced* by $V^\dagger$ and is denoted by $\mathcal{G}(V^\dagger)$. A set of subgraphs of $\mathcal{G}$ are called *edge-disjoint* if no two of them have an edge in common. A *tree* is a minimal connected graph. A *spanning tree* of $\mathcal{G}$ is a subgraph of $\mathcal{G}$ which includes all vertices of $\mathcal{G}$ and is a tree.

Lemma 10. *Let $\mathcal{G}$ be a connected graph. Let v^* be any vertex of $\mathcal{G}$. Then there is a bijection π from $[v(\mathcal{G})]$ onto $V(\mathcal{G})$ such that $\pi(v(\mathcal{G})) = v^*$ and for any $i \in [v(\mathcal{G}) - 1]$, the edge set*

$$\{(p, \{\pi(i), \pi(j)\}) \in E(\mathcal{G}) : j \in \{i+1, \ldots, v(\mathcal{G})\}\}$$

is not empty.

Proof. We prove the conclusion by induction on $v(\mathcal{G})$. If $v(\mathcal{G}) = 1$, the conclusion holds trivially. Now we consider the general case. Let $\mathcal{T}$ be a spanning tree of $\mathcal{G}$. Note that any tree with at least two vertices must have at least 2 leaves, that is, vertices with edge-degree 1. Hence $\mathcal{T}$ has a leaf other than v^*. We denote this leaf by v_1. We remove v_1 from $\mathcal{T}$. After removing v_1, $\mathcal{G}$ is still a connected graph, but only has $v(\mathcal{G}) - 1$ vertices. By induction hypothesis, there is a bijection π^* from $[v(\mathcal{G}) - 1]$ onto $V(\mathcal{G})\backslash\{v_1\}$ such that $\pi^*(v(\mathcal{G}) - 1) = v^*$ and for any $i \in [v(\mathcal{G}) - 2]$, the edge set

$$\{(p, \{\pi^*(i), \pi^*(j)\}) \in E(\mathcal{G}) : j \in \{i+1, \ldots, v(\mathcal{G}) - 1\}\}$$

is not empty. Define $\pi(1) = v_1$ and $\pi(i) = \pi^*(i-1)$, $i = 2, \ldots, v(\mathcal{G})$. It can be seen that the function π satisfies our requirement. This completes the proof. □

For a partition $\mathscr{P}$ of $V(\mathcal{G})$, we use $E_{\mathscr{P}}(\mathcal{G})$ to denote the set of edges of $\mathcal{G}$ which join vertices belonging to different members of $\mathscr{P}$. The following result was proved by Tutte (1961) and Nash-Williams (1961).

Lemma 11 (Tutte (1961) and Nash-Williams (1961))**.** *Suppose $\mathcal{G}$ is a multigraph with labeled edges and k is a positive integer. Then $\mathcal{G}$ has k edge-disjoint spanning trees if and only if*

$$\mathrm{Card}(E_{\mathscr{P}}(\mathcal{G})) \geq k(\mathrm{Card}(\mathscr{P}) - 1)$$

for every partition $\mathscr{P}$ of $V(\mathcal{G})$.

Lemma 11 can be used to prove the following result.

Lemma 12. *Suppose $\mathcal{G}$ is a multigraph with labeled edges. Suppose $\mathcal{G}$ is Eulerian and $E(\mathcal{G}) \neq \emptyset$. Then either $\mathcal{G}$ itself has 2 edge-disjoint spanning trees or $V(\mathcal{G})$ has a nonempty subset S^* satisfying the following conditions:*

- *There are exact two edges in $E(\mathcal{G})$ joining the induced subgraphs $\mathcal{G}(S^*)$ and $\mathcal{G}(V(\mathcal{G})\backslash S^*)$.*
- *The induced subgraph $\mathcal{G}(S^*)$ has 2 edge-disjoint spanning trees.*

Proof. Suppose that $\mathcal{G}$ itself does not have 2 edge-disjoint spanning trees. By Lemma 11, there is a partition $\mathscr{P}$ of $V(\mathcal{G})$ such that $\mathrm{Card}(E_{\mathscr{P}}(\mathcal{G})) < 2(\mathrm{Card}(\mathscr{P}) - 1)$. Note that

$$2\mathrm{Card}(E_{\mathscr{P}}(\mathcal{G})) = \sum_{S \in \mathscr{P}} \mathrm{Card}(\{\text{Edges in } E(\mathcal{G}) \text{ joining } S \text{ and } V(\mathcal{G})\backslash S\}).$$

Hence the collection

$$\{S \subset V(\mathcal{G}) : S \neq \emptyset, S \neq V(\mathcal{G}), \text{ and there are less than 4 edges in } E(\mathcal{G}) \text{ joining } S \text{ and } V(\mathcal{G})\backslash S\}$$

is not empty. Let S^* be a vertex set in the above collection with minimum vertex number. Since $\mathcal{G}$ is Eulerian, there are even edges connecting S^* and $V(\mathcal{G})\backslash S^*$. Hence there are exactly two edges connecting S^* and $V(\mathcal{G})\backslash S^*$. Suppose these two edges are $(\tilde{p}, \{g, h\})$ and $(\tilde{p}', \{g', h'\})$ where $g, g' \in V(\mathcal{G})\backslash S^*$ and $h, h' \in S^*$. Let $\mathcal{C}$ be the

induced subgraph $\mathcal{G}(S^*)$ with an additional edge joining h and h'. Then all vertices of $\mathcal{C}$ have even edge-degrees and hence $\mathcal{C}$ is Eulerian.

We claim that for any nonempty proper subset S' of S^*, there are at least 4 edges in $\mathcal{C}$ joining S' and $S^*\backslash S'$.

To prove the above claim, we decompose the vertex set $V(\mathcal{G})$ into three vertex sets S', $S^*\backslash S'$ and $V(\mathcal{G})\backslash S^*$. By the mininum property of S^*, there are at least 4 edges in $E(\mathcal{G})$ joining S' and $V(\mathcal{G})\backslash S' = (V(\mathcal{G})\backslash S^*) \cup (S^*\backslash S')$. If $\mathcal{S}' \cap \{h, h'\} = \emptyset$, then there is no edge in $E(\mathcal{G})$ joining S' and $V(\mathcal{G})\backslash S^*$, and hence there are at least 4 edges in $E(\mathcal{G})$ joining S' and $S^*\backslash S'$. If exactly one of h and h' belongs to $\mathcal{S}'$ and the other one belong to $S^*\backslash \mathcal{S}'$, then there is exactly one edge in $E(\mathcal{G})$ joining S' and $V(\mathcal{G})\backslash S^*$, and hence there are at least 3 edges in $E(\mathcal{G})$ joining S' and $S^*\backslash S'$. Also, the additional edge in $E(\mathcal{C})$ joining h and h' also joins S' and $S^*\backslash S'$. Hence in this case, there are at least 4 edges in $E(\mathcal{C})$ joining S' and $S^*\backslash S'$. If h and h' are both in S', then there is no edge in $E(\mathcal{G})$ joining $S^*\backslash S'$ and $V(\mathcal{G})\backslash S^*$. Hence the number of edges in $E(\mathcal{G})$ joining $S^*\backslash S'$ and S' is equal to the number of edges in $E(\mathcal{G})$ joining $S^*\backslash S'$ and $V(\mathcal{G})\backslash(S^*\backslash S')$ which, by the minimality of S^*, is at least 4. Hence our claim holds.

Let $\mathscr{P}$ be any partition of $V(S^*)$. Then the above claim implies that

$$\mathrm{Card}(E_{\mathscr{P}}(\mathcal{C})) = \frac{1}{2}\sum_{S'\in\mathscr{P}} \mathrm{Card}(\{\text{Edges of } \mathcal{C} \text{ joining } S' \text{ and } S^*\backslash S'\}) \geq 2\mathrm{Card}(\mathscr{P}).$$

Compared with $\mathcal{G}(S^*)$, the graph $\mathcal{C}$ has only one additional edge. Consequently,

$$\mathrm{Card}(E_{\mathscr{P}}(\mathcal{G}(S^*))) \geq \mathrm{Card}(E_{\mathscr{P}}(\mathcal{C})) - 1 \geq 2\mathrm{Card}(\mathscr{P}) - 1.$$

Hence from Lemma 11, the graph $\mathcal{G}(S^*)$ has 2 edge-disjoint spanning trees. This completes the proof.

□

C PROOF OF PROPOSITION 1

We claim that for $\mathcal{S} \subset [m] \times [n]$ such that $\mathrm{Card}(\mathcal{S}) = O(\log(d/\delta))$, there exists an absolute constant $c > 1$ such that

$$\mathrm{E}\prod_{(i,j)\in\mathcal{S}} |\tilde{\delta}_{i,j}| \leq \Big(c\frac{s}{m}\Big)^{\mathrm{Card}(\mathcal{S})}. \tag{9}$$

For the kth row block of $\tilde{\mathbf{\Delta}}$, its each column contains exactly one nonzero element. Consequently, if there are two indices (i_1, j_1) and (i_2, j_2) in $\mathcal{S}$ with $i_1 \neq i_2$, $j_1 = j_2$ such that they are in the same row block, that is, $\lfloor i_1/(m/s)\rfloor = \lfloor i_2/(m/s)\rfloor$, then we have $\prod_{(i,j)\in\mathcal{S}} |\tilde{\delta}_{i,j}| = 0$. Hence to prove (9), we can without loss of generality and assume that for each $j \in [n]$, the indices i such that $(i, j) \in \mathcal{S}$ are in distinct row blocks.

Define

$$t := \max\{j \in [n] : \text{ there exists } i \in [m] \text{ such that } (i, j) \in \mathcal{S}\}.$$

We prove (9) by induction on t. First we consider the case of $t = 1$. In this case, suppose $\mathcal{S} = \{(i_1, 1), \ldots, (i_c, 1)\}$. From the independence of $w_1^{(1)}, \ldots, w_1^{(s)}$, we have

$$\begin{aligned}
\mathrm{E}\prod_{(i,j)\in\mathcal{S}} |\tilde{\delta}_{i,j}| &= \Pr\Big(|\tilde{\delta}_{i,1}| = 1 \text{ for all } (i,1) \in \mathcal{S}\Big) \\
&= \Pr\Big(w_1^{(\lfloor \frac{i}{m/s}\rfloor+1)} = i \text{ for all } (i,1) \in \mathcal{S}\Big) \\
&= \prod_{(i,1)\in\mathcal{S}} \Pr\Big(w_1^{(\lfloor \frac{i}{m/s}\rfloor+1)} = i\Big) \\
&= \Big(\frac{s}{m}\Big)^{\mathrm{Card}(\mathcal{S})}.
\end{aligned}$$

Suppose the conclusion holds for $t < T$. We consider the case of $t = T$. Define

$$t' := \max\{j \in [T-1] : \text{ there exists } i \in [m] \text{ such that } (i,j) \in \mathcal{S}\}.$$

If the above set is empty, then the indices of $\mathcal{S}$ are all in one column and we can prove the conclusion in a similar way as in the case of $t = 1$. Below we assume the above set is not empty. Then t' is well defined. Define

$$\begin{aligned}
\mathcal{S}_T :=& \{(i,j) \in \mathcal{S} : j = T\},\\
\mathcal{S}_{t'} :=& \{(i,j) \in \mathcal{S} : j \le t'\},\\
\mathcal{I}_{t'} :=& \{i : (i,j) \in \mathcal{S}_{t'}\},\\
\mathcal{I}_T^* :=& \{i : (i,T) \in \mathcal{S}_T,\, i \notin \mathcal{I}_{t'}\},\\
\mathcal{J}_{t'} :=& \{j : (i,j) \in \mathcal{S}_{t'}\}.
\end{aligned}$$

Let $\mathcal{F}_{t'}$ denote the σ-algebra generated by the random variables $\tau(j)$ and $w_j^{(\lfloor \frac{i}{m/s} \rfloor)}$, $(i,j) \in \mathcal{S}_{t'}$. Then we have

$$\mathrm{E} \prod_{(i,j)\in\mathcal{S}} |\tilde{\delta}_{i,j}| = \mathrm{E}\Big\{ \big(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}} |\tilde{\delta}_{i^\dagger,j^\dagger}| \big) \Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i,\, (i,T) \in \mathcal{S}_T \mid \mathcal{F}_{t'} \big) \Big\}. \tag{10}$$

We define the event $\mathcal{A}$ as

$$\mathcal{A} = \left\{ \tau(T) - \left\lfloor \frac{\tau(T)}{n/s} \right\rfloor \frac{n}{s} \in \left\{ \tau(j) - \left\lfloor \frac{\tau(j)}{n/s} \right\rfloor \frac{n}{s} : j \in \mathcal{J}_{t'} \right\} \right\}.$$

Conditioning on $\mathcal{F}_{t'}$, the randomness of the event $\mathcal{A}$ comes entirely from $\tau(T)$. Hence we have

$$\Pr(\mathcal{A} \mid \mathcal{F}_{t'}) \le \frac{s\mathrm{Card}(\mathcal{S})}{n - \mathrm{Card}(\mathcal{S})}. \tag{11}$$

Conditioning on $\mathcal{F}_{t'}$ and the event $\mathcal{A}^{\complement}$, the random variables $w_T^{(1)}, \ldots, w_T^{(s)}$ are independent of $\mathcal{F}_{t'}$. Hence

$$\Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i,\, (i,T) \in \mathcal{S}_T \mid \mathcal{F}_{t'}, \mathcal{A}^{\complement} \big) = \Big(\frac{s}{m} \Big)^{\mathrm{Card}(\mathcal{S}_T)}. \tag{12}$$

Now we consider the case of conditioning on $\mathcal{F}_{t'}$ and the event $\mathcal{A}$. We have

$$\begin{aligned}
\Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i,\, (i,T) \in \mathcal{S}_T \mid \mathcal{F}_{t'}, \mathcal{A}, \tau(T) \big) &\le \Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i,\, i \in \mathcal{I}_T^* \mid \mathcal{F}_{t'}, \mathcal{A}, \tau(T) \big)\\
&= \prod_{i\in\mathcal{I}_T^*} \Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i \mid \mathcal{F}_{t'}, \mathcal{A}, \tau(T) \big).
\end{aligned} \tag{13}$$

Let i^* be any index in the set $\mathcal{I}_T^*$. Then there are two possibilities. The first possibility is that there exists an index $(i^\dagger, j^\dagger) \in \mathcal{S}_{t'}$ such that

$$\left\lfloor \frac{i^*}{m/s} \right\rfloor = \left\lfloor \frac{i^\dagger}{m/s} \right\rfloor \quad \text{and} \quad \tau(T) - \left\lfloor \frac{\tau(T)}{n/s} \right\rfloor \frac{n}{s} = \tau(j^\dagger) - \left\lfloor \frac{\tau(j^\dagger)}{n/s} \right\rfloor \frac{n}{s}.$$

In this case, by the first condition, $i^\dagger$ falls in the same row block as i^*. By the second condition, we have $|\tilde{\delta}_{i^*,T}| = |\tilde{\delta}_{i^*,j^\dagger}|$. But by the definition of i^*, we must have $i^\dagger \neq i^*$. Hence it must be that $|\tilde{\delta}_{i^*,j^\dagger}| \neq |\tilde{\delta}_{i^\dagger,j^\dagger}|$. Consequently, $|\tilde{\delta}_{i^*,T}| \neq |\tilde{\delta}_{i^\dagger,j^\dagger}|$. That is, these two quantities can not be nonzero simultaneously. It follows that

$$\big(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}} |\tilde{\delta}_{i^\dagger,j^\dagger}| \big) \prod_{i\in\mathcal{I}_T^*} \Pr\big(w_T^{(\lfloor \frac{i}{m/s} \rfloor)} = i \mid \mathcal{F}_{t'}, \mathcal{A}, \tau(T) \big) = 0. \tag{14}$$

The second possibility is that for any index $(i^\dagger, j^\dagger) \in \mathcal{S}_{t'}$,

$$\left\lfloor \frac{i^*}{m/s} \right\rfloor \neq \left\lfloor \frac{i^\dagger}{m/s} \right\rfloor \quad \text{or} \quad \tau(T) - \left\lfloor \frac{\tau(T)}{n/s} \right\rfloor \frac{n}{s} \neq \tau(j^\dagger) - \left\lfloor \frac{\tau(j^\dagger)}{n/s} \right\rfloor \frac{n}{s}.$$

In this case, we have

$$\prod_{i\in\mathcal{I}_T^*}\Pr\left(w_T^{(\lfloor\frac{i}{m/s}\rfloor)}=i\mid\mathcal{F}_{t'},\mathcal{A},\tau(T)\right)=\left(\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{I}_T^*)}. \tag{15}$$

It follows from (13), (14) and (15) that

$$(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|)\prod_{i\in\mathcal{I}_T^*}\Pr\left(w_T^{(\lfloor\frac{i}{m/s}\rfloor)}=i\mid\mathcal{F}_{t'},\mathcal{A}\right)\leq(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|)\left(\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{I}_T^*)}. \tag{16}$$

Combining (11), (12) and (16) leads to

$$\begin{aligned}&\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|\Pr\left(w_T^{(\lfloor\frac{i}{m/s}\rfloor)}=i,\,(i,T)\in\mathcal{S}_T\mid\mathcal{F}_{t'}\right)\\&\leq\left(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|\right)\left(\left(\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{I}_T^*)}\frac{s\mathrm{Card}(\mathcal{S})}{n-\mathrm{Card}(\mathcal{S})}+\left(\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{S}_T)}\right).\end{aligned}$$

It follows from the condition $\mathrm{Card}(\mathcal{S})=O(\log(d/\delta))$ and the choice of s, m and n that

$$\begin{aligned}\frac{s\mathrm{Card}(\mathcal{S})}{n-\mathrm{Card}(\mathcal{S})}=&\frac{O\left(\frac{\log^3(d/\delta)}{\epsilon}\right)}{\exp\left\{\Omega(\log(d/\delta)(\log(d/\epsilon)+|\log(\log(d/\delta))|)\right\}}\\=&\frac{1}{\exp\left\{\Omega(\log(d/\delta)(\log(d/\epsilon)+|\log(\log(d/\delta))|)\right\}}\\=&\left(\frac{1}{\exp\left\{\Omega(\log(d/\epsilon)+|\log(\log(d/\delta))|)\right\}}\right)^{\mathrm{Card}(\mathcal{S}_T)}\\=&O\left(\left(\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{S}_T)}\right).\end{aligned}$$

Hence there exists an absolute constant $c>1$ such that

$$\left(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|\right)\Pr\left(w_T^{(\lfloor\frac{i}{m/s}\rfloor)}=i,\,(i,T)\in\mathcal{S}_T\mid\mathcal{F}_{t'}\right)\leq\left(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|\right)\left(c\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{S}_T)}.$$

Then from (10),

$$\mathrm{E}\prod_{(i,j)\in\mathcal{S}}|\tilde{\delta}_{i,j}|\leq\mathrm{E}\left(\prod_{(i^\dagger,j^\dagger)\in\mathcal{S}_{t'}}|\tilde{\delta}_{i^\dagger,j^\dagger}|\right)\left(c\frac{s}{m}\right)^{\mathrm{Card}(\mathcal{S}_T)}.$$

Then by induction, (9) holds. Thus, Assumption 1 holds when $\mathrm{Card}(\mathcal{S})=O(\log(d/\delta))$. Note that in the proof of Theorem 1, we take $\ell=\Theta(\log(d/\delta))$. Hence the restriction $\mathrm{Card}(\mathcal{S})=O(\log(d/\delta))$ does not cause problems. Then the conclusion follows from Theorem 1.

D PROOF OF LEMMA 1

Lemma 13. *Suppose n,m,k are powers of 2 satisfying $n=mk$. Then we have*

$$\mathbf{H}_n=\mathbf{H}_m\otimes\mathbf{H}_k.$$

Proof. We prove the conclusion by induction on n. The conclusion clearly holds for $n=1$. Suppose the conclusion holds for any n^* such that n^* is a power of 2 and $n^*<n$. We prove that the conclusion holds for n. By the definition of $\mathbf{H}_n$, we have $\mathbf{H}_n=\mathbf{H}_2\otimes\mathbf{H}_{\frac{n}{2}}$. Hence we only consider the case that $m>2$. In this case, we have

$$\mathbf{H}_n=\mathbf{H}_2\otimes\mathbf{H}_{\frac{n}{2}}=\mathbf{H}_2\otimes\mathbf{H}_{\frac{m}{2}}\otimes\mathbf{H}_k=\mathbf{H}_m\otimes\mathbf{H}_k,$$

where the second equality follows from the induction hypothesis. This completes the proof. □

Now we prove Lemma 1 by induction on k. The claim clearly holds for $k = 1$. Now suppose the claim holds for $k - 1$. Then by the definition of $\mathbf{B}_k$ and the induction hypothesis, we have

$$\begin{aligned}\mathbf{B}_k\mathbf{B}_{k-1}\cdots\mathbf{B}_1 =& \left(\mathbf{I}_{2^{k-1}} \otimes \left(\mathbf{H}_2 \otimes \mathbf{I}_{\frac{n}{2^k}}\right)\right)\left(\mathbf{H}_{2^{k-1}} \otimes \mathbf{I}_{\frac{n}{2^{k-1}}}\right)\\ =&\mathbf{H}_{2^{k-1}} \otimes \mathbf{H}_2 \otimes \mathbf{I}_{\frac{n}{2^k}}\\ =&\mathbf{H}_{2^k} \otimes \mathbf{I}_{\frac{n}{2^k}},\end{aligned}$$

where the last equality follows from Lemma 13. This completes the proof of Lemma 1.