Revisiting Co-Occurring Directions: Sharper Analysis and Efficient Algorithm for Sparse Matrices

Luo Luo1, Cheng Chen2*, Guangzeng Xie3, Haishan Ye4

1 Department of Mathematics, The Hong Kong University of Science and Technology
2 Department of Computer Science and Engineering, Shanghai Jiao Tong University
3 Academy for Advanced Interdisciplinary Studies, Peking University
4 Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen
luoluo@ust.hk, jack.chen1990@sjtu.edu.cn, smsxgz@pku.edu.cn, huye_cs@outlook.com

Abstract

We study the streaming model for approximate matrix multiplication (AMM). We are interested in the scenario that the algorithm can only take one pass over the data with limited memory. The state-of-the-art deterministic sketching algorithm for streaming AMM is the co-occurring directions (COD), which has much smaller approximation errors than randomized algorithms and outperforms other deterministic sketching methods empirically. In this paper, we provide a tighter error bound for COD whose leading term considers the potential approximate low-rank structure and the correlation of input matrices. We prove COD is space optimal with respect to our improved error bound. We also propose a variant of COD for sparse matrices with theoretical guarantees. The experiments on real-world sparse datasets show that the proposed algorithm is more efficient than baseline methods.

1 Introduction

A large scale machine learning system usually receives data sequentially and it is often impossible to exactly store the entire data set. Thus, the approximate matrix multiplication (AMM) in the streaming fashion is an important and fundamental task for scientific computation and big data analysis. For example, the product of matrices from multi-modal datasets captures the correlation between different modalities. In addition, many classical algorithms including canonical correlation analysis (Hotelling 1992), generalized eigenvector decomposition (Golub and Loan 1996), partial least squares (Wegelin 2000), spectral co-clustering (Dhillon 2001) require to perform approximate matrix multiplication when the data set is very large. On the other hand, data matrices from real-world are usually low-rank and sparse, which motivated us to design efficient and effective sparse algorithms.

This paper considers streaming AMM problem as follows. Give two large matrices \( X \in \mathbb{R}^{n \times d_x} \) and \( Y \in \mathbb{R}^{n \times d_y} \), we are interested in finding a low-rank estimator \( A^\top B \) to approximate \( X^\top Y \), where \( A \in \mathbb{R}^{m \times d_x}, B \in \mathbb{R}^{m \times d_y} \) and \( m \) is much smaller than \( n, d_x \) and \( d_y \). We focus on the row update model, that is, the algorithm receives rows of \( X \) and \( Y \) sequentially and it only takes one pass over input matrices with limited memory. The key challenge for this problem is to reduce the space/time complexity while maintaining the approximation error.

In this paper, we provide a sharper analysis for co-occurrence directions (COD) for streaming AMM. COD shrinks the singular values of input matrices sequentially and it only takes one pass over input matrices. The key challenge for this problem is to reduce the space/time complexity while maintaining the approximation error.
occuring directions (COD). We give a new spectral norm error bound which considers the potential low-rank structure of the target matrix. Our bound could be much tighter than [Mroueh, Marcheret, and Goel]’s (2017) results when the spectrum of the exact matrix product is dominated by its top singular values. In addition, we prove that the space complexity of COD is optimal to attain our improved error bound. Furthermore, in the case of $X = Y$, our result matches the error bound of FD.

We further propose sparse co-occurring directions (SCOD) and provide an error bound which matches our improved analysis on standard COD while the running time of the algorithm mainly depends on the non-zero entries of input matrices. We conduct numerical experiments on cross-language datasets to show that SCOD has better performance than state-of-the-art algorithms empirically. Concurrent to our work, [Wan and Zhang] (2020) have also proposed a similar COD based algorithm to address streaming AMM with sparse inputs but their error bound does not consider the potential low-rankness.

The rest of the paper is organized as follows. In Section 2, we define the notation used in this paper and introduce the background of related algorithms for streaming AMM. In Section 3, we provide our new error bound for COD algorithm and show the corresponding space lower bound. In Section 4, we propose SCOD and give its theoretical guarantees. In Section 5, we conduct the numerical experiments to show the superiority of SCOD. We defer detailed proof of some lemmas and theorems into supplementary materials. We conclude our work in Section 6.

2 Notations and Preliminaries

In this section, we first introduce the notation will be used in this paper. Then we give the backgrounds of frequent directions and related algorithms for AMM.

2.1 Notations

We let $I_p$ be the $p \times p$ identity matrix and $0_{p \times q}$ be the $p \times q$ matrix of all zeros. For an $p \times q$ matrix $A = [A_{ij}]$, we denote $(a^{(i)})^\top$ its $i$-th row, $\pi_2(A)$ be the number of non-zero entries of $A$. The condensed singular value decomposition (SVD) of $A$ is defined as $U \Sigma V^\top$, where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{m \times r}$ are column orthogonal, $\Sigma = \text{diag}(\sigma_1(A), \sigma_2(A), \ldots, \sigma_r(A))$ with $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_r(A) > 0$ places the nonzero singular values on its diagonal entries and $r$ is the rank of $A$. We have $\sigma_i(A) = 0$ for any $i > r$. Additionally, we let $\|A\|_p = \sqrt{\sum_{i,j} A_{ij}^2}$ be the Frobenius norm, $\|A\|_2 = \sigma_1(A)$ be the spectral norm, $\|A\|_* = \sum_{i=1}^r \sigma_i(A)$ be the nuclear norm and $\|A\|_k = \sum_{i=1}^k \sigma_i(A)$ be the Ky Fan $k$-norm. We also denote $A_k$ as the best rank-$k$ approximation to $A$ for any unitary invariant norms, that is, $A_k = \sum_{i=1}^k \sigma_i(A) u_i v_i^\top$, where $u_i$ and $v_i$ are the $i$-th column of $U$ and $V$ respectively.

2.2 Frequent Directions

Frequent directions [Liberty 2013, Ghashami et al. 2016] is a deterministic algorithm for covariance sketching. Given any matrix $X \in \mathbb{R}^{n \times d}$ and sketch size $m$ that is much smaller than $n$ and $d$, FD processes the rows of $X$ one by one and produces a sketch matrix $A \in \mathbb{R}^{2m \times d}$ to approximate $X^\top X$ by $A^\top A$. We present the details of FD in Algorithm 1 which requires $O(md)$ space and $O(mnd)$ time complexity. The algorithm has the following theoretical guarantees.

Lemma 1 (Ghashami and Phillips 2014, Ghashami et al. 2016). The output $A$ of Algorithm 1 satisfies

$$\|X^\top X - A^\top A\|_2 \leq \frac{1}{m-k} \left( \|X\|_F^2 - \|X_k\|_F^2 \right)$$

for any $k < m$.

Ghashami et al. (2016) also prove FD is space optimal with respect to the guaranteed accuracy in Lemma 1. Note that the shrinking step in line 8 of the algorithm is necessary because the output could be extremely worse without this operation [Desai, Ghashami, and Phillips 2016, Luo et al. 2019].

Algorithm 1 Frequent Directions (FD)

1: Input: $X \in \mathbb{R}^{n \times d}$ and sketch size $m$
2: $A \leftarrow 0_{2m \times d}$
3: for $t = 1, 2, \ldots, n$
4: insert $x^{(t)} \top$ into a zero valued row of $A$
5: if $A$ has no zero valued rows then
6: $[U, \Sigma, V] \leftarrow \text{SVD}(A)$
7: $\delta \leftarrow \sigma_2^2(A)$
8: $\Sigma \leftarrow \sqrt{\max(\Sigma^2 - \delta I_{2m}, 0_{2m \times 2m})}$
9: $A \leftarrow \Sigma V^\top$
10: end if
11: end for
12: Output: $A$

2.3 Sketching Algorithms for AMM

It is natural to exploit the idea of FD to solve general AMM problem [Ye, Luo, and Zhang 2016]. We can concatenate the input matrix $X \in \mathbb{R}^{n \times d_x}$ and $Y \in \mathbb{R}^{n \times d_y}$ to construct a larger matrix $Z = [X, Y] \in \mathbb{R}^{n \times (d_x + d_y)}$, and then apply FD on $Z$ to approximate $Z^\top Z$ by $C^\top C$, where $C = [A, B], A \in \mathbb{R}^{n \times d_x}, B \in \mathbb{R}^{n \times d_y}$. The top right sub-matrix of $C^\top C$, i.e., the matrix $A^\top B$ is an approximation of $X^\top Y$. Intuitively, this algorithm wastes a large proportion of cost to approximate $X^\top X$ and $Y^\top Y$ (the other sub-matrices of $Z^\top Z$), which is unnecessary for the AMM task.

[Mroueh, Marcheret, and Goel] (2017) proposed the co-occurring directions (COD) for AMM. We present its detailed procedure in Algorithm 2. Each iteration of COD constructs the column basis of $A$ and $B$ by QR factorization independently and executes the shrinkage step on the small interaction matrix $R_x R_y^\top$. We point out that both COD and FD-AMM requires $O(m(d_x + d_y))$ space and $O(mn(d_x + d_y))$ time complexity. However, COD looks more reasonable than
We can prove that COD holds the guaranteed accuracy as Woodruff 2017; Sarlos 2006) when input matrices is dense. The real-world dataset typically enjoys some approximate and loop has been executed for $t$ times. Additionally, we use $A^{(t)}$ and $B^{(t)}$ to represent the matrices after insert operations (line 3 Sharper Analysis for COD

In this section, we provide a tighter error bound for COD. We let $\delta^{(t)}$ be the value of $\delta$ at time step $t$. If the algorithm does not enter the “then” section in the $t$-th step, then we have $\delta^{(t)} = 0$. Similarly, let $A^{(t)}$, $B^{(t)}$, $Q^{(t)}$, $Q^{(t)}$, $U^{(t)}$, $V^{(t)}$ and $\tilde{\Sigma}^{(t)}$ be the corresponding variables after the main loop has been executed for $t$ times. Additionally, we use $A^{(t)}$ and $B^{(t)}$ to represent the matrices after insert operations (line 5-6) have been executed at the $t$-th iteration. We need the following two lemmas for proving our main results.

Lemma 3 [Mroueh, Marcheret, and Goel 2017]. The output matrices $A$ and $B$ of Algorithm 2 satisfy

$$\|X^T Y - A^T B\|_2 \leq \frac{1}{m-k} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_F \right),$$

for any $k < m$.

Proof. Let $\Delta = \sum_{i=1}^{n} \delta^{(i)}$. Connecting inequality (4) in Lemma 3 and the result of Lemma 4, we have

$$m \Delta + \|X^T Y\|_F - \|A\|_F \|Y\|_F \leq \|X^T Y\|_F - \|B\|_F \|Y\|_F \leq \sum_{i=k+1}^{d} \sigma_i(X^T Y) + k \Delta,$$

that is $\Delta \leq \frac{1}{m-k} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_F \right)$. Substituting above bound of $\Delta$ into inequality (4) of Lemma 3, we finish the proof of this theorem.

To achieve the accuracy that $\|X^T Y - A^T B\|_2 \leq \varepsilon$, the previous error bound (Lemma 2) requires the sketch size to be at least $m_1 = \frac{1}{\varepsilon} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_F \right)$, while Theorem 1 only requires the sketch size $m_2 = k + \frac{1}{\varepsilon} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_F \right)$. When input matrices $X$ and $Y$ have strongly correlation and approximate low-rank structure, $m_2$ could be much smaller than $m_1$.

In addition, the error bound of Theorem 1 matches that of FD (Lemma 1) when $X = Y$:

$$\|X^T X - A^T A\|_2 \leq \frac{1}{m-k} \left( \|X\|_F \|X\|_F - \sum_{i=1}^{k} \sigma_i(X^2 X) \right) = \frac{1}{m-k} \left( \|X\|_F^2 - \sum_{i=1}^{k} \sigma_i^2(X) \right) = \frac{1}{m-k} \left( \|X\|_F^2 - \|X_k\|_F^2 \right).$$

On the other hand, the previous error bound (Lemma 2) is worse than that of FD (Lemma 1) in the symmetric case of $X = Y$.  

Algorithm 2 Co-Occurring Directions (COD)

1: **Input:** $X \in \mathbb{R}^{n \times d_x}$, $Y \in \mathbb{R}^{n \times d_y}$ and sketch size $m$
2: $A \leftarrow 0_{2m \times d_x}$
3: $B \leftarrow 0_{2m \times d_y}$
4: for $t = 1, 2, \ldots, n$
5: \hspace{1em} insert $(x^{(t)})^T$ into a zero valued row of $A$
6: \hspace{1em} insert $(y^{(t)})^T$ into a zero valued row of $B$
7: if $A$ or $B$ has no zero valued rows then
8: \hspace{1em} $(Q_x, R_x) \leftarrow QR (A^T)$
9: \hspace{1em} $(Q_y, R_y) \leftarrow QR (B^T)$
10: \hspace{1em} $[U, \Sigma, V] \leftarrow \text{SVD}(R_x R_y^T)$
11: \hspace{1em} $\delta \leftarrow \frac{\sigma_m(R_x R_y^T)}{m}$
12: \hspace{1em} $\tilde{\Sigma} \leftarrow \max(\Sigma - \delta I_{2m}, 0_{2m \times 2m})$
13: $A \leftarrow \tilde{\Sigma}^{1/2} U^T Q_x$
14: $B \leftarrow \tilde{\Sigma}^{1/2} V^T Q_y$
15: end if
16: end for
17: **Output:** $A$ and $B$
4 Space Lower Bounds Analysis

In this section, we show that COD is space optimal with respect to our new error bound in Theorem 1. We first introduce the following lemma for low-rank matrices.

**Lemma 5 (Kapralov and Talwar [2013])**. For each \( \delta > 0 \) there exists a set of matrices \( Q = \{Q_1, \ldots, Q_N\} \) and \( N = 2^{\Omega((d_x - \delta \log(1/\delta)))} \), where \( Q_i \in \mathbb{R}^{d_x \times d_x} \) with \( Q_iQ_i^\top = I_d \) such that \( \|Q_iQ_j^\top\|_2 < 1 - \delta \).

By using Lemma 5, we can construct a sets contains exponential number of matrices that each pair of them are not “too close”. The formalized result is shown in Lemma 6.

**Lemma 6.** For each \( \delta > 0 \) and \( d_z \leq d_y \) there exists a set of matrices \( \tilde{Z}_\ell = \{(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), \ldots, (\tilde{X}^{(N)}, \tilde{Y}^{(N)})\} \), where \( N = 2^{\Omega((d_z - \delta \log(1/\delta)))} \) and \( \tilde{Y}^{(i)} \in \mathbb{R}^{d_x \times d_z} \) satisfies \( \tilde{X}^{(i)}\tilde{X}^{(i)^\top} = I_d \) and \( \|\tilde{Y}^{(i)}\tilde{Y}^{(i)^\top}\|_2 < 1 - \delta \) for any \( i = 1, \ldots, n \) and

\[
\|\tilde{X}^{(i)}\tilde{Y}^{(i)^\top} - \tilde{X}^{(j)}\tilde{Y}^{(j)^\top}\|_2 > \sqrt{2\delta}
\]

for any \( j \neq i \).

**Proof.** Based on Lemma 5 there exist a set of matrices \( Y = \{Y^{(1)}, \ldots, Y^{(N)}\} \), where \( N = 2^{\Omega((d_z - \delta \log(1/\delta)))} \), and \( \tilde{Y}^{(i)} \in \mathbb{R}^{d_x \times d_z} \) satisfies \( \tilde{Y}^{(i)}\tilde{Y}^{(i)^\top} = I_d \) and \( \|\tilde{Y}^{(i)}\tilde{Y}^{(i)^\top}\|_2 < 1 - \delta \). We further set \( \tilde{X}^{(i)} = [I_d, 0_{m \times (d_z - d_y)}] \). We have

\[
\|\tilde{X}^{(i)}\tilde{Y}^{(i)} - \tilde{X}^{(j)}\tilde{Y}^{(j)}\|_2 = \|\tilde{Y}^{(i)} - \tilde{Y}^{(j)}\|_2 \geq 2 - \|\tilde{Y}^{(j)}\tilde{Y}^{(j)^\top} + \tilde{Y}^{(i)}\tilde{Y}^{(i)^\top}\|_2 \geq 2\delta,
\]

where we use the definition of \( \tilde{X}^{(i)}, \tilde{Y}^{(i)} \) and the fact \( \|A^\top A\|_2 = \|A\|_2 \).

Then we present a lower bound of space complexity for approximate matrix multiplication, which matches the memory cost of COD. Hence, we can conclude that COD is space optimal with respect to the guaranteed accuracy in Theorem 1.

**Theorem 2.** We consider any matrix sketching algorithm with inputs as \( X \in \mathbb{R}^{n \times d_x} \) and \( Y \in \mathbb{R}^{n \times d_y} \) and outputs \( A \in \mathbb{R}^{m \times d_x} \) and \( B \in \mathbb{R}^{m \times d_y} \) with guarantee

\[
\|X^\top Y - A^\top B\|_2 \leq \frac{1}{m - k} \left( \|X\|_F \|Y\|_F - \|X^\top Y\|_k \right)
\]

for any \( k \leq m \). Assuming that a constant number of bits is required to describe a word (i.e., a unit of memory), then the algorithm requires at least \( \Omega\left(m(d_x + d_y)\right) \) bits of space.

**Proof.** Without loss of generality, we suppose that \( d_y \geq d_x \). Let \( \tilde{Z}_\ell = \{(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), \ldots, (\tilde{X}^{(N)}, \tilde{Y}^{(N)})\} \) be the set of matrices defined in Lemma 6 with \( \ell = m/4 \), \( \delta = 1/8 \) and \( N = 2^{\Omega((d_x - \delta \log(1/\delta)))} \). We construct matrices \( X^{(i)} = [\tilde{X}^{(i)}, 0_{(m - m/4) \times d_x}] \in \mathbb{R}^{n \times d_x} \) and \( Y^{(i)} = [\tilde{Y}^{(i)}, 0_{n \times d_y}] \in \mathbb{R}^{n \times d_y} \) for \( i = 1, \ldots, N \). Then we have \( \tilde{Z}_\ell = \{(X^{(i)}, Y^{(i)})\}_{i=1}^N \) which satisfies \( \|X^{(i)^\top} Y^{(i)} - X^{(j)^\top} Y^{(j)}\|_2 > 1/2 \) for each \( i \neq j \). Let (A, B) be the output of the matrix sketching algorithm with input \( (X^{(i)}, Y^{(i)}) \). The guarantee of the algorithm indicates

\[
\|X^{(i)^\top} Y^{(i)} - A^\top B\|_2 \leq \frac{1}{m - k} \left( \frac{m}{4} - k \right) \leq \frac{1}{4}.
\]

Hence, each (A, B) only encodes one matrix pencil in \( Z_\ell \) (the product of the matrices), which means that the lower bound of space complexity to attach the desired accuracy is \( \log_2 N = \Omega(md_y) = \Omega(m(d_x + d_y)) \) bits.

5 Sparse Co-Occurring Directions

In this section, we proposed a variant of COD for sparse AMM. We also prove its error bound is similar to our improved result of COD.

5.1 The Algorithm

We describe details of our sparse co-occuring directions (SCOD) in Algorithm 3. The procedure of SCOD maintains the sparse data in two buffer matrices \( X' \) and \( Y' \). The algorithm restricts the non-zero entries in buffers to be less than \( m(d_x + d_y) \) and the number of row of each buffer is at most \( d_z + d_y \). When the buffers are full, we perform subspace power method (SPM) (Woodruff [2014b], Musco and Musco [2015]) to approximate the data in the buffers by low-rank matrices \( X \in \mathbb{R}^{m \times d_x} \) and \( Y \in \mathbb{R}^{m \times d_y} \) such that \( X^\top Y' \approx X' Y' \). We present the procedure of SPM in Algorithm 3.

Let \( X^{(i)} \) and \( Y^{(i)} \) be the results of \( X' \) and \( Y' \) after Algorithm 3 has executed “then” section for \( i \)-times. Define \( C = [X^{(1)}, \ldots, X^{(T)}] \) and \( D = [Y^{(1)}, \ldots, Y^{(T)}] \) where \( T \) is the number of total times we enter “then” section of the algorithm. Then \( C \) and \( D \) are the estimators of \( X \) and \( Y \) respectively and the procedure of SCOD can be regarded as as running standard COD on input matrices \( C \) and \( D \) in streaming fashion. Since the row numbers of buffers \( X' \) and \( Y' \) could be much larger than \( m \), the operations on dense matrices (line 14-20) will not be executed frequently. Hence, SCOD is much more efficient than COD for sparse inputs.

**Algorithm 3 Subspace Power Method (SPM)**

1: Input: \( M \in \mathbb{R}^{d_1 \times d_2} \), target rank \( m \) and integer \( q > 0 \)
2: \( G = [G_{ij}] \in \mathbb{R}^{d_2 \times m} \), where \( G_{ij} \sim \mathcal{N}(0, 1) \) i.i.d
3: \( K = (MM^\top)^q MG \in \mathbb{R}^{d_1 \times m} \)
4: \( Z \leftarrow \) orthonormal column basis of \( K \)
5: Output: \( Z \)

5.2 Analysis of Error Bound

The analysis of SCOD is more challenging than sparse frequent directions (SFD) (Ghashami, Liberty, and Phillips [2016]) which only addresses the case of \( X = Y \). The reason is the “mergeability property” of FD (Ghashami et al. [2016], Desai, Ghashami, and Phillips [2016], Ghashami, Liberty, and Phillips [2016]) only works for Frobenius norm and it is not applicable to COD.
Algorithm 4 Sparse Co-Occurring Directions (SCOD)

1: **Input:** $X \in \mathbb{R}^{n \times d_x}$, $Y \in \mathbb{R}^{n \times d_y}$, sketch size $m$, failure probability $\delta$ and sequence $\{q_i\}_{i=1,2,\ldots}$

2: $i = 0$

3: $A \leftarrow 0_{m \times d_x}$, $B \leftarrow 0_{m \times d_y}$

4: $X' \leftarrow$ empty, $Y' \leftarrow$ empty

5: for $t = 1, 2, \ldots , n$

6: $X' \leftarrow [X'; (x^{(t)}_i)^T]$, $Y' \leftarrow [Y'; (y^{(t)}_i)^T]$

7: if $\text{nnz}(X') + \text{nnz}(Y') > m(d_x + d_y)$ or $t = n$

or $\text{rows}(X') = d_x + d_y$

or $\text{rows}(Y') = d_x + d_y$

then

8: $Z = \text{SubspacePowerMethod}(X'^T Y', m, q_i)$

9: $[\bar{U}, \Sigma, \bar{V}] = \text{SVD}(Z^T X' Y')$

10: $\bar{X} \leftarrow \Sigma_{1/2} \bar{U}^T Z^T$

11: $\bar{Y} \leftarrow \Sigma_{1/2} \bar{V}^T$

12: $A \leftarrow [A; \bar{X}]$

13: $B \leftarrow [B; \bar{Y}]$

14: $(Q_x, R_x) \leftarrow \text{QR}(A^T)$

15: $(Q_y, R_y) \leftarrow \text{QR}(B^T)$

16: $[U, \Sigma, V] \leftarrow \text{SVD}(R_x R_y^T)$

17: $\delta \gets \sigma_m(R_x R_y^T)$

18: $\bar{\Sigma} \leftarrow \max(\Sigma - \delta I_m, 0_{m \times m})$

19: $A \leftarrow \bar{\Sigma}^{1/2} U^T Q_x$

20: $B \leftarrow \bar{\Sigma}^{1/2} V^T Q_y$

21: $X' \leftarrow$ empty, $Y' \leftarrow$ empty

22: $i \leftarrow i + 1$

end if

23: end for

24: **Output:** $A$ and $B$

Proof. Let $X^{(i)}$ and $Y^{(i)}$ be the value of $X'$ and $Y'$ when we execute subspace power methods at $i$-th time in line 8 of Algorithm 4; then we have

$$X = [X^{(1)}; \ldots ; X^{(T)}] \text{ and } Y = [Y^{(1)}; \ldots ; Y^{(T)}].$$

Using Lemma 7 with $M = \bar{X}^{(i)} Y^{(i)}$ and $q_i$, then with probability $1 - p_i$, we have

$$\|X^{(i)^T Y^{(i)}} - X^{(i)^T} Y^{(i)}\|_2 \leq (1 + \varepsilon)\sigma_{m+1}(X^{(i)^T} Y^{(i)})$$

for $\varepsilon > 0$ and $p_i = \delta / 2i^2$, with probability $1 - \delta$, the outputs $A$ and $B$ of Algorithm 4 hold that

$$\|X^T Y - A^T B\|_2 \leq \frac{2 + \varepsilon}{m - k} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_k \right)$$

for all $k < m$.

Proof. Consider that $A$ and $B$ can be viewed as the output of running Algorithm 4 with input matrices

$$C = [\bar{X}^{(1)}; \ldots ; \bar{X}^{(T)}] \text{ and } D = [\bar{Y}^{(1)}; \ldots ; \bar{Y}^{(T)}].$$

The approximation error of SCOD comes from two parts: the compressing error from sub-routine SPM and the merge error from estimating $C^T D$ by $A^T B$. We first consider a single call of SPM, which approximation error can be bounded as follows.

**Lemma 7.** Let $q = \hat{O}(\log(md_1/p)/\varepsilon)$ for Algorithm 4, then the output $Z$ satisfies $\|M - ZZ^T M\|_2 \leq (1 + \varepsilon)\sigma_{m+1}(M)$ with probability at least $1 - p$.

Based on Lemma 7, we can bound the total compressing error of SCOD by the following lemma.

**Lemma 8.** Setting $q_i = \hat{O}(\log(md_1/p_i)/\varepsilon)$ and $p_i = \delta / 2i^2$, then we have then Algorithm 4 holds that

$$\|X^T Y - C^T D\|_2 \leq \frac{2 + \varepsilon}{m - k} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_k \right),$$

for any $k < m$ and $\varepsilon > 0$ with probability $1 - \delta$.

Since the analysis of merging error is similar to standard COD, we can establish the error bound of SCOD by using above lemmas.

**Theorem 3.** Setting $q_i = \hat{O}(\log(md_1/p_i)/\varepsilon)$ with constant $\varepsilon > 0$ and $p_i = \delta / 2i^2$, with probability $1 - \delta$, the outputs $A$ and $B$ of Algorithm 4 hold that

$$\|X^T Y - A^T B\|_2 \leq \frac{2 + \varepsilon}{m - k} \left( \|X\|_F \|Y\|_F - \|X^T Y\|_k \right)$$

for all $k < m$.

Proof. Consider that $A$ and $B$ can be viewed as the output of running Algorithm 4 with input matrices

$$C = [\bar{X}^{(1)}; \ldots ; \bar{X}^{(T)}] \text{ and } D = [\bar{Y}^{(1)}; \ldots ; \bar{Y}^{(T)}].$$
Following the proof of Theorem 1, we have
\[
\begin{align*}
\| C^T D - A^T B \|_2 & \leq \frac{1}{m-k} \left( \sum_{i=1}^{T} \| \tilde{X}^{(i)} \|_F \| \tilde{Y}^{(i)} \|_F - \| X^T Y \|_k + k \| X^T Y - C^T D \|_2 \right) \\
& \leq \sum_{i=1}^{T} \frac{\| \tilde{X}^{(i)} \|_F \| \tilde{Y}^{(i)} \|_F - \| X^T Y \|_k + \| X^T Y - C^T D \|_2}{m-k} \\
& \leq \frac{1}{m-k} \left( \sum_{i=1}^{T} \| \tilde{X}^{(i)} \|_F \| \tilde{Y}^{(i)} \|_F - \| X^T Y \|_k \right) \\
& \quad + \frac{(1+\epsilon)k}{(m-k)^2} \left( \| X \|_F \| Y \|_F - \| X^T Y \|_k \right) \\
& \leq \left( \frac{1}{m-k} + \frac{(1+\epsilon)k}{(m-k)^2} \right) \left( \| X \|_F \| Y \|_F - \| X^T Y \|_k \right)
\end{align*}
\]
where we use Lemma 8 and triangle inequality. Combining above results and Lemma 8, we have
\[
\begin{align*}
\| X^T Y - A^T B \|_2 & \leq \| X^T Y - C^T D \|_2 + \| C^T D - A^T B \|_2 \\
& \leq \frac{2+\epsilon}{m-k} \left( \| X \|_F \| Y \|_F - \| X^T Y \|_k \right),
\end{align*}
\]
with probability at least 1 − δ. \qed

5.3 Complexity Analysis

We use the constant-word-size model for our analysis like that of sparse FD (Ghashami et al. 2016). We suppose floating point numbers are represented by a constant number of bits, random access into memory requires O(1) time and multiplying a sparse matrix M by a dense vector requires O(1)(M) time and storing M requires O(1)(M) space.

The procedure of SCOD (Algorithm 4) implies the buffer \( \mathbf{X}' \) and \( \mathbf{Y}' \) is sparse and contains at most \( m(d_x + d_y) \) non-zero entries and it is not difficult to verify that all dense matrices in the algorithm cost no more than \( O(m(d_x + d_y)) \) space. Hence, the space complexity of SCOD is \( O(m(d_x + d_y)) \) in total which is the same as COD (Algorithm 2).

Then we analyze the time complexity of SCOD. The constraints on buffer size means we have
\[
T \leq \frac{\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})}{m(d_x + d_y)} + \frac{n}{d_x + d_y}.
\]

Since each QR factorization or SVD on \( m \times d \) matrix cost \( O(m^2 d) \) time, the operation on dense matrices of Algorithm 4 from line 20 requires at most
\[
O(m^2(d_x + d_y)T) = O(m(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + m^2 n).
\]

Note that SCOD calls SPM with input \( \mathbf{M} = \mathbf{X}'^T \mathbf{Y}' \). Since both \( \mathbf{X}' \) and \( \mathbf{Y}' \) are sparse, it is unnecessary to construct \( \mathbf{M} \) explicitly and we can multiply \( \mathbf{X}' \) and \( \mathbf{Y}' \) on \( \mathbf{G} \) separately in line 3 of Algorithm 3. Then the time complexity of executing SPM needs \( O(q, \text{nnz}(\mathbf{X}^{(i)}) + \text{nnz}(\mathbf{Y}^{(i)}) + m^2 d_x) \) when the algorithm enters “then” section at the \( i \)-th time. Following the upper bound of \( T \) and the setting of \( q_i \) in Theorem 3 the calls of SPM in Algorithm 3 entirely takes at most
\[
O \left( \sum_{i=1}^{T} \left( m q_i (\text{nnz}(\mathbf{X}^{(i)}) + \text{nnz}(\mathbf{Y}^{(i)}) + m^2 d_x) \right) \right)
\]
\[
\leq O \left( m q_T (\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + T m^2 d_x \right)
\]
\[
= O \left( m \text{nnz}(\mathbf{X}) (1 + \text{nnz}(\mathbf{Y})) + T m^2 d_x \right).
\]
Hence, the total time complexity of proposed SCOD is \( O(m \text{nnz}(\mathbf{X}) (1 + \text{nnz}(\mathbf{Y}))) + n^2 m \).

6 Numerical Experiments

In this section, we empirically compare the proposed sparse co-occurring directions (SCOD) with frequent direction based AMM (FD-AMM) (Ye, Luo, and Zhang 2016), co-occurring directions (COD) (Mroueh, Marcheret, and Goel 2017) and sparse frequent direction based AMM algorithm (SFD-AMM) instead of increasing \( q_i \) logarithmically as the analysis of Theorem 2 we fix \( q_i = 5 \) in our experiment since the empirical error arise from subspace power method is very small in practice.

We evaluate performance of all algorithms on cross-language datasets: Amazon Product Reviews (APR), PAN-PC-11 (PAN), JRC Acquis (JRC) and Europarl (EURO) which contain millions of English (EN), French (FR) and Spanish (ES) sentences (Prentenhofer and Stein 2010; Pottikhast et al. 2010; Potthast et al. 2010; Koehn 2005). We use bag-of-words feature for our experiments. All of input matrices are large but very sparse and we summary the parameters in Table 1.

We demonstrate sketch-error and time-error comparisons in Figure 1 and 2 respectively. It is apparently that SCOD always performs better than all baseline algorithms. We do not include the curve of FD-AMM and COD in time-error comparison because these two algorithms take much more time than others. Due to the limit of space, we defer the result of sketch-time comparison and detailed computing infrastructure in appendix.

7 Conclusion

In this paper, we first improved the error bound of a deterministic sketching algorithm COD for streaming AMM problem. In symmetric case, our result matches the error bound of classical algorithm FD. We also proved COD matches the space lower bound complexity to achieve our error bound. In addition, we proposed a sparse variant of COD with a reasonable error bound. The experimental results show that the proposed algorithm has better performance than baseline methods in practice.

It would be interesting to borrow the idea of this paper to establish better theoretical guarantees and streaming algorithms for more classical machine learning and statistical models such as canonical correlation analysis (Hotelling 1992; Avron et al. 2013; Ye, Luo, and Zhang 2016), generalized eigenvector decomposition (Bhatia et al. 2018; Golub and Loan 1996) and spectral co-clustering (Dhillon 2001).

SFD-AMM refers to the method simply replacing FD step in FD-AMM with sparse frequent directions (Ghashami et al. 2016). We provide more detailed discussion about SFD-AMM in appendix.
| Dataset       | $n$    | $d_x$    | $d_y$    | $\text{density}(X)$ | $\text{density}(Y)$ |
|---------------|--------|----------|----------|----------------------|----------------------|
| APR (EN-FR)   | $2.32 \times 10^4$ | $2.80 \times 10^4$ | $4.28 \times 10^4$ | $6.31 \times 10^{-4}$ | $4.53 \times 10^{-4}$ |
| PAN (EN-FR)   | $8.90 \times 10^4$ | $5.12 \times 10^4$ | $9.96 \times 10^4$ | $4.38 \times 10^{-4}$ | $2.43 \times 10^{-4}$ |
| JRC (EN-FR)   | $1.50 \times 10^5$ | $1.72 \times 10^5$ | $1.87 \times 10^5$ | $1.65 \times 10^{-4}$ | $1.64 \times 10^{-4}$ |
| JRC (EN-ES)   | $1.50 \times 10^5$ | $1.72 \times 10^5$ | $1.92 \times 10^5$ | $1.65 \times 10^{-4}$ | $1.64 \times 10^{-4}$ |
| JRC (FR-ES)   | $1.50 \times 10^5$ | $1.87 \times 10^5$ | $1.92 \times 10^5$ | $1.64 \times 10^{-4}$ | $1.64 \times 10^{-4}$ |
| EURO (EN-FR)  | $4.76 \times 10^5$ | $7.25 \times 10^4$ | $8.77 \times 10^4$ | $3.46 \times 10^{-4}$ | $3.65 \times 10^{-4}$ |
| EURO (EN-ES)  | $4.76 \times 10^5$ | $7.25 \times 10^4$ | $8.80 \times 10^4$ | $3.46 \times 10^{-4}$ | $3.47 \times 10^{-4}$ |
| EURO (FR-ES)  | $4.76 \times 10^5$ | $8.77 \times 10^4$ | $8.80 \times 10^4$ | $3.65 \times 10^{-4}$ | $3.47 \times 10^{-4}$ |

Table 1: We present the size and density of datasets used in our experiments, where $\text{density}(X) = \text{nnz}(X)/nd_x$ and $\text{density}(Y) = \text{nnz}(Y)/nd_y$. All of these datasets are publicly available [Ferrero et al. 2016].
Acknowledgements
Luo Luo is supported by GRF 16201320. Haishan Ye is supported by Shenzhen Research Institute of Big Data (named “Automated Machine Learning”).

References
Avron, H.; Boutsidis, C.; Toledo, S.; and Zouzias, A. 2013. Efficient dimensionality reduction for canonical correlation analysis. In ICML.
Bhatia, K.; Pacchiano, A.; Flammarion, N.; Bartlett, P. L.; and Jordan, M. I. 2018. Gen-Oja: Simple & efficient algorithm for streaming generalized eigenvector computation. In NIPS.
Chen, C.; Luo, L.; Zhang, W.; Yu, Y.; and Lian, Y. 2020. Efficient and Robust High-Dimensional Linear Contextual Bandits. In IJCAI.
Clarkson, K. L.; and Woodruff, D. P. 2017. Low-rank approximation and regression in input sparsity time. Journal of the ACM 63(6): 1–45.
Desai, A.; Ghashami, M.; and Phillips, J. M. 2016. Improved practical matrix sketching with guarantees. IEEE Transactions on Knowledge and Data Engineering 28(7): 1678–1690.
Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In SIGKDD.
Dickens, C. 2020. Ridge Regression with Frequent Directions: Statistical and Optimization Perspectives. arXiv preprint:2011.03607.
Drineas, P.; Kannan, R.; and Mahoney, M. W. 2006. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. SIAM Journal on Computing 36(1): 132–157.
Ferrero, J.; Agnes, F.; Besacier, L.; and Schwab, D. 2016. A multilingual, multi-style and multi-granularity dataset for cross-language textual similarity detection. In LREC.
Ghashami, M.; Liberty, E.; and Phillips, J. M. 2016. Efficient frequent direction algorithm for sparse matrices. In SIGKDD.
Ghashami, M.; Liberty, E.; Phillips, J. M.; and Woodruff, D. P. 2016. Frequent directions: Simple and deterministic matrix sketching. SIAM Journal on Computing 45(5): 1762–1792.
Ghashami, M.; and Phillips, J. M. 2014. Relative errors for deterministic low-rank matrix approximations. In SODA.
Golub, G. H.; and Loan, C. F. V. 1996. Matrix computations. Johns Hopkins University Press, 3rd edition.
Horn, R. A.; and Johnson, C. R. 1994. Topics in matrix analysis. Cambridge university press.
Hotelling, H. 1992. Relations between two sets of variates. In Breakthroughs in statistics, 162–190. Springer.
Huang, Z. 2019. Near optimal frequent directions for sketching dense and sparse matrices. Journal of Machine Learning Research 20(56): 1–23.
Kapralov, M.; and Talwar, K. 2013. On differentially private low rank approximation. In SODA.
Woodruff, D. P. 2014b. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends in Theoretical Computer Science* 10(1-2): 1–157.

Ye, Q.; Luo, L.; and Zhang, Z. 2016. Frequent direction algorithms for approximate matrix multiplication with applications in CCA. In *IJCAI*. 
In this supplementary materials, Section [A] provide detailed proofs of lemmas we used in main text. Section [E] gives more details of experiments. We also provide additional discussion on algorithm SFD-AMM in Section [F].

### A The Proof of Lemma 3

This lemma can be proved by the analysis of Mroueh, Marcheret, and Goel’s (2017) Theorem 2. We reformulate the details by our notations here for completeness.

**Proof.** Let $\hat{A}^{(t)} = (\Sigma^{(t)})^{1/2}U^{(t)\top}Q_x^{(t)}$ and $\hat{B}^{(t)} = (\Sigma^{(t)})^{1/2}V^{(t)\top}Q_y^{(t)}$. The first inequality can be proved as follows:

$$
\|X^\top Y - A^\top B\|_2
= \sum_{t=1}^n x^{(t)}(y^{(t)})^\top - \sum_{t=1}^n (A^{(t)})^\top B^{(t)} - (A^{(t-1)})^\top B^{(t-1)}
= \sum_{t=1}^n (\hat{A}^{(t)})^\top \hat{B}^{(t)} - (A^{(t)})^\top B^{(t)}
\leq \sum_{t=1}^n \|\hat{A}^{(t)}\|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t-1)}\|_\ast \|B^{(t-1)}\|_\ast - \sum_{t=1}^n \|\hat{A}^{(t)}\|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t-1)}\|_\ast \|B^{(t-1)}\|_\ast.
$$

Then we show the second inequality. Similar to above analysis, we have

$$
\|A^\top B\|_\ast
= \sum_{t=1}^n \|\hat{A}^{(t)}\|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t-1)}\|_\ast \|B^{(t-1)}\|_\ast - \sum_{t=1}^n \|\hat{A}^{(t)}\|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t-1)}\|_\ast \|B^{(t-1)}\|_\ast.
$$

The QR steps means $(\hat{A}^{(t)})^\top \hat{B}^{(t)}$ and $(A^{(t)})^\top B^{(t)}$ can be written as

$$(\hat{A}^{(t)})^\top \hat{B}^{(t)} = Q_x^{(t)} U^{(t)} \Sigma^{(t)} V^{(t)\top} Q_y^{(t)}$$
and

$$(A^{(t)})^\top B^{(t)} = Q_x^{(t)} U^{(t)} (\Sigma - \sigma_m^{(t)} I_{m\ast})^{(t)} V^{(t)\top} Q_y^{(t)}.$$

which implies

$$
\|\hat{A}^{(t)}\|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t)}\|_\ast \|B^{(t)}\|_\ast = m\sigma_m^{(t)}.
$$

Using triangle inequality, we have

$$
\|\hat{A}^{(t)} \|_\ast \|\hat{B}^{(t)}\|_\ast - \|A^{(t-1)}\|_\ast \|B^{(t-1)}\|_\ast
\leq \|\hat{A}^{(t)} \|_\ast \|\hat{B}^{(t)} - (A^{(t-1)})^\top B^{(t-1)}\|_\ast
= \|x^{(t)}(y^{(t)})^\top\|_\ast = \|x^{(t)}\|_2 \|y^{(t)}\|_2
$$

Combing all above results, we have

$$
\|A^\top B\|_\ast \leq \sum_{t=1}^n \|x^{(t)}\|_2 \|y^{(t)}\|_2 - m \sum_{t=1}^n \sigma_m^{(t)} \leq \|X\|_F \|Y\|_F - m \sum_{t=1}^n \sigma_m^{(t)}.
$$

\[\square\]
The Proof of Lemma 4

This lemma is crucial to establish the tighter bound of COD. We first introduce the following property of Ky Fan $k$-norm.

**Lemma 10** (Horn and Johnson 1994, Theorem 3.4.1). *Given matrix $M$, we have*

$$
\|M\|_k = \max \left\{ \|\text{tr} \left( P^T MQ \right)\| : P^T P = I_k, Q^T Q = I_k \right\},
$$

where $\text{tr}(\cdot)$ is the trace of the matrix.

Then we prove Lemma 4 by using Lemma 10.

**Proof.** We let $(\hat{P}, \hat{Q}) = \arg \max \left\{ \|\text{tr} \left( P^T AQ \right)\| : P^T P = I, Q^T Q = I \right\}$. Then we have

$$
\|X^T Y\|_* - \|A^T B\|_*
\leq \sum_{i=1}^d \sigma_i(X^T Y) + \sum_{i=k+1}^d \sigma_i(X^T Y) - \|A^T B\|_*
= \max \left\{ \|\text{tr} \left( P^T (X^T Y)Q \right)\| : P^T P = I_k, Q^T Q = I_k \right\} + \sum_{i=k+1}^d \sigma_i(X^T Y)
\geq \max \left\{ \|\text{tr} \left( P^T (A^T B)Q \right)\| : P^T P = I_k, Q^T Q = I_k \right\} - \sum_{i=k+1}^d \sigma_i(A^T B)
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \|\text{tr} \left( \hat{P}^T (X^T Y) \hat{Q} \right)\| - \|\text{tr} \left( \hat{P}^T (A^T B) \hat{Q} \right)\|
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \max \left\{ \|\text{tr} \left( P^T (X^T Y - A^T B) Q \right)\| : P^T P = I_k, Q^T Q = I_k \right\}
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \max \left\{ \left\| \text{tr} \left( P^T \left( \sum_{t=1}^n \sigma_m^{(t)} Q_{x}^{(t)} U^{(t)} V^{(t)} Q_{y}^{(t)^T} \right) \right) \right\| : P^T P = I_k, Q^T Q = I_k \right\}
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \sum_{i=k+1}^d \sigma_i(X^T Y) \sum_{t=1}^n \sigma_i \left( Q_{x}^{(t)} U^{(t)} V^{(t)} Q_{y}^{(t)^T} \right)
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \sum_{i=1}^n \delta^(t) \sum_{t=1}^n \sigma_i \left( Q_{x}^{(t)} U^{(t)} V^{(t)} Q_{y}^{(t)^T} \right)
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \sum_{i=1}^n \delta^(t) \sum_{t=1}^n \sigma_i \left( Q_{x}^{(t)} U^{(t)} V^{(t)} Q_{y}^{(t)^T} \right)
\leq \sum_{i=k+1}^d \sigma_i(X^T Y) + \sum_{i=1}^n \delta^(t),
$$

where the first inequality is due to the definition of $\hat{P}$ and $\hat{Q}$; the second and the third one use triangle inequality; the last two inequality is based on the procedure of the algorithm; all equalities come from Lemma 10 and the procedure of COD.

The Proof of Lemma 7

We can prove Lemma 7 by modifying the analysis in Section 4.3 of Woodruff’s (2014) survey. We present the details for completeness. The proof is based on the following lemma.

**Lemma 11** (Rudelson and Vershynin 2010, Proposition 2.4 and (3.2)). *Let $\Omega \in \mathbb{R}^{d_1 \times d_2}$ be a random matrix whose entries are independent mean zero sub-gaussian random variables whose subgaussian moments are bounded by 1. Then we have*

1. $\mathbb{P} \left( \|\Omega\|_2 > C \sqrt{d_1 + \sqrt{d_2}} + t \right) \leq 2 \exp(-ct^2)$ for any $t > 0$;
2. $\mathbb{P}(\sigma_{min}(\Omega) \leq \zeta d^{-1/2}) \leq \zeta$ when $d_1 = d_2 = d$ for any $\zeta > 0$;

where $c > 0$ and $C > 0$ are some constants.
Then we provide the proof of Lemma \ref{lem:zhang:2014:technical}.

**Proof.** Let $N = (MM^T)^qM$. By Woodruff’s (2014b) Lemma 4.14, $ZZ^TM$ is the best rank-\(m\) approximation of $M$ in the column space of $Z$ with respect to the spectral norm. Hence, we have
\[
\|M - ZZ^TM\|_2 \leq \|M - (ZZ^TN)(ZZ^TN)^\dagger M\|_2 \leq \|(I_d_1 - (ZZ^TN)(ZZ^TN)^\dagger M\|_2 ,
\]
where the notation $(\cdot)^\dagger$ presents pseudo-inverse; the inequality follows $ZZ^TN$ is of rank-\(m\) and in the column space of $Z$. Since $I_d_1 - (ZZ^TN)(ZZ^TN)^\dagger$ is a projection matrix, we can apply Woodruff’s (2014b) Lemma 4.15 to infer that
\[
\|I_d_1 - (ZZ^TN)(ZZ^TN)^\dagger M\|_2 \leq \|I_d_1 - (ZZ^TN)(ZZ^TN)^\dagger M\|_2^{1/(2q+1)}
\]
and the statement follows.

We now apply Woodruff’s (2014b) Lemma 4.4 with the $G$ of that lemma equal to $Z$ above, the $Z$ of that lemma equal to $V_m$, and the $A$ of that lemma equal to $N$ above. This implies the $E$ of that lemma is equal to $N - N_m$. Note that to apply the lemma we need $V_m^TG$ to have full rank, which holds with probability 1 since it is a $m \times m$ matrix of i.i.d. $N(0, 1)$ random variables. We thus have
\[
\|M - ZZ^TM\|_2 \leq \|N - ZZ^TN\|_2^{1/(2q+1)} .
\]

Let $USV^T$ be the SVD of $N$, $V_m^TG \in \mathbb{R}^{m \times m}$ and $\Omega_L = V_{d_1 - m}^TG \in \mathbb{R}^{(d_1 - m) \times m}$, where $V_m^T$ denotes the top $m$ rows of $V$ and $V_{d_1 - m}^T$ the remaining rows. Since $V^T$ are column orthonormal, by rotational invariance of the Gaussian distribution, both $\Omega_L$ and $\Omega_U$ are independent matrices of i.i.d. $\mathcal{N}(0, 1)$ entries.

We now apply Woodruff’s (2014b) Lemma 4.4 with the $C$ of that lemma equal to $Z$ above, the $Z$ of that lemma equal to $V_m$, and the $A$ of that lemma equal to $N$ above. This implies the $E$ of that lemma is equal to $N - N_m$. Note that to apply the lemma we need $V_m^TG$ to have full rank, which holds with probability 1 since it is a $m \times m$ matrix of i.i.d. $N(0, 1)$ random variables. We thus have
\[
\|N - ZZ^TN\|_2^2
\]
\begin{align*}
= & \|N - N_m\|_2^2 + \|(N - N_m)G(V_m^TG)^\dagger\|_2^2 \\
= & \|N - N_m\|_2^2 + \|U_{d_1 - m}^T\Sigma_{d_1 - m}V_{d_1 - m}^TG(V_m^TG)^\dagger\|_2^2 \\
= & \|N - N_m\|_2^2 + \|\Sigma_{d_1 - m}V_{d_1 - m}^TG(V_m^TG)^\dagger\|_2^2 \\
\leq & \|N - N_m\|_2^2 \left(1 + \|\Omega_L\|_2^2 \|\Omega_L\|_2^2\right)
\end{align*}

where $\Sigma_{d_1 - m}$ denotes the $(d_1 - m) \times (d_1 - m)$ diagonal matrix whose entries are the bottom $d_1 - m$ diagonal entries of $\Sigma$, and $U_{d_1 - m}$ denotes the rightmost $d_1 - m$ columns of $U$. Here in the second equality we use unitary invariance of $U_{d_1 - m}$, while in the inequality we use sub-multiplicativity of the spectral norm.

By using Lemma \ref{lem:zhang:2014:technical} with $\Omega = \Omega_U$ and $t = \sqrt{c^{-1}\log(4/p)}$, we have
\[
\mathbb{P}\left(\|\Omega_U\|_2^2 \leq \left(\sqrt{d_1 - m} + \sqrt{m} + \sqrt{c^{-1}\log(4/p)}\right)^2\right) \geq 1 - \frac{p}{2}.
\]
(9)

By using Lemma \ref{lem:zhang:2014:technical} with $\Omega = \Omega_U$ and $\zeta = p/2$, we have
\[
\mathbb{P}\left(\sigma_{\min}^2(\Omega_U) \geq \frac{p^2}{4m}\right) \geq 1 - \frac{p}{2} .
\]
(10)

Since $\Omega_L$ and $\Omega_U$ are independent, combing inequalities (9) and (10), we have
\[
1 + \|\Omega_L\|_2^2 \|\Omega_U\|_2^2 \leq 1 + \left(\sqrt{d_1 - m} + \sqrt{m} + \sqrt{c^{-1}\log(4/p)}\right)^2 \cdot \frac{4m}{p^2} \leq \frac{c_0(d_1 + \log(1/p))m}{p^2}
\]
(11)
for some constant $c_0 > 0$ with probability at least $(1 - p/2)^2 > 1 - p$. 

Combining results of (7), (8) and (11), we have
\[ \| M - ZZ^T M \|_2 \leq \| N - N_m \|_2^{1/(2q+1)} \cdot \left( \frac{c_0(d_1 + \log(1/p))m}{p^2} \right)^{1/(4q+2)}. \]
Noting that \( \| N - N_m \|_2 = \| M - M_m \|_2^{2q+1} \) and setting
\[ q = \frac{1}{4} \left( \frac{1}{\varepsilon} \log \left( \frac{c_0(d_1 + \log(1/p))m}{p^2} \right) - 2 \right) \]
we have
\[ \| M - ZZ^T M \|_2 \leq (1 + \varepsilon) \| M - M_m \|_2 = (1 + \varepsilon)\sigma_{m+1}(M) \]
with probability at least \( 1 - p \).

\[ \square \]

**D The Proof of Lemma 9**

**Proof.** The procedure of Algorithm 4 means \( \sigma^2_i(\tilde{X}^{(i)}) = \sigma^2_i(\tilde{Y}^{(i)}) = \sigma_i(\tilde{X}^{(i)}\tilde{Y}^{(i)}) \). Consider that the output \( Z^{(i)} \) of SPM
(Algorithm 3) is column orthonormal, then we have
\[
\| X^{(i)} \|_F^2 \| Y^{(i)} \|_F^2 = \| \bar{X}^{(i)} \|_F^2 \| \bar{Y}^{(i)} \|_F^2 = \| Z^{(i)} \|_F^2 \| Z^{(i)} \|_F^2 = \| Z^{(i)} \|_F^2 \| Z^{(i)} \|_F^2 \leq \| X^{(i)} \|_F^2 \| Y^{(i)} \|_F^2 \]
\[
\leq \| X^{(i)} \|_F^2 \| Y^{(i)} \|_F^2 .
\]

\[ \square \]

**E More Details of Numerical Experiments**

Our experiments are conducted on a desktop computer with Intel(R) Core(TM) i5-4570 CPU and 24GB memory. We use MATLAB 2019a to run the experiments and the operating system is Windows 10. In the implementation of subspace power method (Algorithm 3), powering \( MM^\top \) makes \( (MM^\top)^q MG \) could be ill-conditioned. We include an additionally orthonormalization step after each round of multiplications to improve the stability (Martinsson et al. 2010; Musco and Musco 2015). This operation does not change the column span, so it gives an equivalent algorithm in exact arithmetic, but improves empirical performance significantly. Since \( q \) is typical a small constant in practice, the additionally cost of orthonormalization is limited.

We use cross-language datasets as we mentioned in Section 6. Each of dataset has alignment information of two languages at sentence-level and there are \( n \) sentences in total. We let \( t \)-th row of \( X \) be the bag-of-words feature of \( t \)-th sentence with respect to one language and \( t \)-th row of \( Y \) be the bag-of-words feature of the same sentence respect to the other language.

We present sketch-time comparison in Figure 3. The algorithms SFD-AMM and SCOD are much more faster than FD-AMM and COD, since FD-AMM and COD ignore the sparse structure of the input matrices. The running time of SFD-AMM and SCOD are comparable which satisfies our complexity analysis.

**F Additional Discussion on SFD-AMM**

Ghashami et al. (2016) proposed a variant of FD for sketching sparse matrices called sparse frequent directions (SFD). Given input matrix \( Z \in \mathbb{R}^{m \times d} \), the output algorithm \( C \in \mathbb{R}^{m \times d} \) such that
\[
\| Z^\top Z - C^\top C \|_2 \leq \frac{1}{\alpha m - k} \left( \| C \|_F^2 - \| Z \|_F^2 \right)
\]
with high probability for any \( k < \alpha m \), where \( \alpha \) is a constant depends on the accuracy of SPM. SFD requires \( \tilde{O}(m\cdot m_{nz}(Z)+m^2n) \) time complexity and \( O(md) \) space. The procedure of SFD is similar to SCOD in the case of \( X = Y \), but includes additional

\[ \text{The code is publicly available at: http://luoluo.people.ust.hk/code/SCOD.zip} \]
shrinking operation on the output of SPM (Ghashami, Liberty, and Phillips 2016) to apply the “mergeability property” of FD (Ghashami et al. 2016) in their analysis.

For streaming AMM with sparse input, it is natural to combine the idea of SFD with FD-AMM directly which leads to the algorithm sparse FD-AMM (SFD-AMM). Similar to FD-AMM, SFD-AMM applies SFD on concatenated matrix $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}] \in \mathbb{R}^{n \times (d_1 + d_2)}$ and its output $\mathbf{C}$ which can be written as $\mathbf{C} = [\mathbf{A}, \mathbf{B}]$, where $\mathbf{A} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{B} \in \mathbb{R}^{n \times d_y}$. Then we use $\mathbf{A}^\top \mathbf{B}$ to approximate $\mathbf{X}^\top \mathbf{Y}$ that satisfies

$$
\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \|\mathbf{Z}^\top \mathbf{Z} - \mathbf{C}^\top \mathbf{C}\|_2 \leq \frac{1}{\alpha m - k} \left( \|\mathbf{Z}\|_F^2 - \|\mathbf{Z}_k\|_F^2 \right).
$$

The time complexity of SFD-AMM has the same order as SCOD since $\text{nnz}(\mathbf{Z}) = \text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})$. It is not easy to compare the error bound (12) with SCOD (Theorem 3) in general because there does not exist simple relationship between the singular values of $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ and $\mathbf{X}^\top \mathbf{Y}$. However, SCOD always performs better than SFD-AMM empirically as we observed in Section 5.

In theoretical, we can improve the time complexity of SFD-AMM to achieve the error bound of (12) by integrating random sampling (Huang 2019). However, the implementation of this strategy requires the value of $k$ is given. Unfortunately, it is difficult to select a suitable $k$ for streaming setting in general. In contrast, the value of $k$ in SFD-AMM or SCOD is only for theoretical analysis and it is no related to the implementation of algorithms.