A. Appendix

Well-Conditioned Basis: (Dasgupta et al., 2009) Let $A \in \mathbb{R}^{n \times d}$ a rank $d$ matrix. For $p \geq 1$, it has a dual $q = p/(p-1)$. A matrix $U$ is said to be $(\alpha, \beta, p)$ well-conditioned basis of $A$, if $U$ spans the column space of $A$, $\sum_{i=1}^{d} \|u_i\|_p \leq \alpha$, $\forall x \in \mathbb{R}^d$, $\|UX\|_p \leq \beta$ and $(\alpha, \beta)$ are $O(1)$ and also independent of $n$.

Theorem A.1. Bernstein (Dubhashi & Panconesi, 2009) Let the scalar random variables $x_1, x_2, \ldots, x_n$ be independent that satisfy $\forall i \in [n], |x_i - E[x_i]| \leq b$. Let $X = \sum_{i=1}^{n} x_i$ and let $\sigma^2 = \sum_{i=1}^{n} \sigma_i^2$ be the variance of $X$. Then for any $t > 0$,

$$Pr(\{X > E[X] + t\}) \leq \exp\left(-\frac{t^2}{2\sigma^2 + bt/3}\right)$$

Theorem A.2. Matrix Bernstein (Tropp et al., 2015) Let $X_1, \ldots, X_n$ are independent $d \times d$ random matrices such that $\forall i \in [n], \|X_i\|_2 \leq b$ and $\text{var}(\|X_i\|) \leq \sigma^2$ where $X = \sum_{i=1}^{n} X_i$, then for some $t > 0$,

$$Pr(\{\|X\| - E[\|X\|] \geq t\}) \leq d \exp\left(-\frac{t^2/2}{\sigma^2 + bt/3}\right)$$

$\epsilon$-net argument: Here we discuss the $\epsilon$-net argument, which we use to ensure a guarantee all query vector $x$ from a fixed dimensional query space $Q$ using union bound argument. Similar use of the argument is discussed in various applications (Woodruff et al., 2014).

Definition A.1. $\epsilon$-net (Haussler & Welzl, 1987) Given some metric space $Q$ its subset $P \subseteq Q$ is an $\epsilon$-net of $Q$ on $\ell_p$ norm if, $\forall x \in Q, \exists y \in P$ such that $\|x-y\|_p \leq \epsilon$.

Note that $\|PAX\|_p^p = \sum_{i=1}^{m} |a_i^T x|^p$, where $P$ is a sampling matrix which samples $m$ rows from $A$ with proper scaling. Now we argue this $\forall x \in Q$, i.e. $\|PAX\|_p^p = (1+\epsilon) \|AX\|_p^p$ which is same as $\|PUy\|_p^p = (1+\epsilon) \|UY\|_p^p$ where $UY = AX$. Now with an $\epsilon$-net, we argue $\forall x \in Q, Q \subseteq R$. Let $B = \{z \in R^n | z = Uy \text{ for some } y \in R^k \text{ and } \|z\|_p = 1\}$. From this set we intend to find a finite subset $N$ which is an $\epsilon$-net to the set. Now here we argue that if we can ensure $\|Pw\|_p^p = (1+\epsilon) \|w\|_p^p, \forall w \in N$ then we can claim that $\|Pz\|_p^p = (1+\epsilon) \|z\|_p^p, \forall w \in B$ which further imply that $\|PAx\|_p^p = (1+\epsilon) \|Ax\|_p^p, \forall x \in Q$.

Let $v \in B$ whose closest $\epsilon$-net point is $w_1 \in N$ such that $\|v-w_1\|_p \leq \epsilon$. Now note that,

$$\|Pv\|_p^p = \|P(w_1 + \Pi(v-w_1))\|_p^p \leq (\|Pw_1\|_p^p + \|\Pi(v-w_1)\|_p^p) \leq (1+\epsilon + \|\Pi(v-w_1)\|_p^p) \leq (1+\epsilon + \|\Pi(w_2/\alpha + v - w_1 - w_2/\alpha)\|_p^p \leq (1+\epsilon + (1+\epsilon) + \|\Pi(v-w_1 - w_2/\alpha)\|_p^p$$

Repeated application of this argument yields

$$\|Pv\|_p^p \leq \left(\sum_{i\geq 0}(1+\epsilon)^i\right)^p \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^p \leq 1 + O(\epsilon)$$

By similar argument one can show that $\|Pv\|_p^p \geq 1 - O(\epsilon)$. Finally by rescaling $\epsilon$ by some constant factor we achieve $\|Pz\|_p^p \leq 1 + \epsilon, \forall z \in B$.

Lemma A.1. There is an $\epsilon$-net $N$, with $|N| \leq (2/\epsilon)^k$.

Proof. Let $N$ be the maximal subset of $y \in R^n$ in the column space of $A$ such that $\|y\|_p = 1$, $\forall y \neq y' \in N$, $\|y-y'\|_p > \epsilon$. Now as $N$ is a maximal set, hence $\forall y \in B, \exists \nu \in N$ for which $\|w-y\|_p \leq \epsilon$. Further $\forall y \neq y' \in N$ two balls centered at $y$ and $y'$ with radius $\epsilon / 2$ are disjoint otherwise by triangle inequality, $\|y-y'\|_p \leq \epsilon$, is a contradiction. So it follows that in a unit sphere in $R^k$ there could be at most $(2/\epsilon)^k$ such balls, i.e. the number of points in $N$.

Now we state the modified version of Sherman Morrison which we use in the function Score().

Lemma A.2. Given a rank-k positive semi-definite matrix $M \in R^{d \times d}$ and a vector $x$ such that it completely lies in the column space of $M$. Then we have,

$$(M + xx^T) = M + Mxx^TM$$

Proof. The proof is in the similar spirit to lemma 5.3. Consider $[V, \Sigma, U] = \text{SVD}(M)$ and since $x$ lies completely in the column space of $M$, hence $\exists y \in R^k$ such that $V^Ty = x$. Note that $V \in R^{d \times k}$.

$$(M + xx^T) = (V^T \Sigma V + yy^T V^T) = V^T (\Sigma + yy^T)^{-1} V^T = V^T (\Sigma^{-1} - \Sigma^{-1} yy^T \Sigma^{-1}) V = V^T (\Sigma^{-1} - \Sigma^{-1} V yy^T V^T \Sigma^{-1} V^T) V = M + Mxx^TM$$

In the above analysis, the first couple of inequalities are by substitution. In the third equality, we use Sherman Morrison formula on the smaller $k \times k$ matrix $\Sigma$ and the rank-1 update $yy^T$.

A.1. LineFilter

Here we provide the proofs of the lemmas used to prove the guarantee claimed in theorem 4.1 by LineFilter.
A.1.1. PROOF OF LEMMA 5.1

Proof. We define the restricted streaming (online) sensitivity scores \( \tilde{s}_i \) for each row \( i \) as follows,

\[
\tilde{s}_i = \sup_x \frac{|a_i^T x|^p}{\sum_{j=1}^n |a_j^T x|^p} = \sup_y \frac{|u_i^T y|^p}{\sum_{j=1}^n |u_j^T y|^p}
\]

Here \( y = \Sigma V^T x \) where \( U, \Sigma, V = \text{svd}(A) \) and \( u_i^T \) is the \( i^{th} \) row of \( U \). Now at this \( i^{th} \) step we also define \( |U_i, \Sigma_i, V_i| = \text{svd}(A_i) \). So with \( y = \Sigma_i V_i^T x \) and \( u_i^T \) is the \( i^{th} \) row of \( U_i \) we rewrite the above optimization function as follows,

\[
\tilde{s}_i = \sup_y \frac{|\tilde{u}_i^T y|^p}{\sum_{j=1}^n |\tilde{u}_j^T y|^p} = \sup_y \frac{1}{\|U_i y\|_p} |\tilde{u}_i^T y|^p
\]

Let there be an \( x^* \) which maximizes \( \tilde{s}_i \). Corresponding to it we have \( y^* = \Sigma_i V_i^T x^* \). For a fixed \( x \), let \( f(x) = \frac{|\tilde{a}_i^T x|^p}{\sum_{j=1}^n |\tilde{a}_j^T x|^p} = \frac{\|\tilde{a}_i x\|_p^p}{\|A_i x\|_p} \) and \( g(y) = \frac{|\tilde{u}_i^T y|^p}{\|U_i y\|_p} \). By assumption we have \( f(x^*) \geq f(x), \forall x \).

We prove this by contradiction that \( \forall y, g(y^*) \geq g(y) \), where \( y = \Sigma_i V_i^T x \). Let \( \exists y' \) such that \( g(y') \geq g(y^*) \). Then we get \( x^* = V_i \Sigma_i^{-1} y' \) for which \( f(x^*) \geq f(x^*) \), by definition we have \( f(x) = g(y) \) for \( y = \Sigma_i V_i^T x \). This contradicts our assumption, unless \( x^* = x^* \).

Now to maximize the score, \( \tilde{s}_i, x \) is chosen from the row space of \( A_i \). Next, without loss of generality we assume that \( \|y\|_p = 1 \) as we know that if \( x \) is in the row space of \( A_i \) then \( y \) is in the row space of \( U_i \). Hence we get \( \|U_i y\| = \|y\| = 1 \).

We break denominator into sum of numerator and the rest, i.e. \( \|U_i y\|_p = |\tilde{u}_i^T y|^p + \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^p \). Consider the denominator term as \( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^p \geq f(n) \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right) \).

From this we estimate \( f(n) \) as follows,

\[
\sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \leq \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{1/2} \cdot 1 = \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{1/2}
\]

\[
\leq \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{2/p/2} = \left( \sum_{j=1}^{i-1} 1 \right)^{2/p} \cdot \left( \sum_{j=1}^{i-1} 1 \right)^{1/2/p} = \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{1/2/p} = \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{1/2/p}
\]

Here equation (i) is by holder’s inequality, where we have \( 2/p + 1 - 2/p = 1 \). So we rewrite the above term as \( \left( \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 \right)^{2/p} \cdot \left( i \right)^{1/2/p} = \sum_{j=1}^{i-1} |\tilde{u}_j^T y|^2 = 1 - |\tilde{u}_i^T y|^2 \). Now substituting this in equation (ii) we get,

\[
\sum_{j=1}^{i-1} |\tilde{u}_j^T y|^p \geq \left( \frac{1}{i} \right)^{1-2/p} (1 - |\tilde{u}_i^T y|^2)
\]

\[
\sum_{j=1}^{i-1} |\tilde{u}_j^T y|^p \geq \left( \frac{1}{i} \right)^{p/2-1} (1 - |\tilde{u}_i^T y|^2)^{p/2}
\]

So we get \( \tilde{s}_i \leq \sup_y \frac{|\tilde{u}_i^T y|^p}{\|U_i y\|^p + (1/i)^{p/2-1} (1 - |\tilde{u}_i|^2)^{p/2}} \).

As we know that a function \( \frac{a+b}{a+b} \leq \min(1, a/b) \), so we get \( \tilde{I}_i = \min(1, (i/p)^{p/2-1} |\tilde{u}_i|^p) \). Note that \( \tilde{I}_i = (i/p)^{p/2-1} |\tilde{u}_i|^p \) when \( \|U_i y\|^p \leq (1/i)^{p/2-1} \).

Here the scores are similar to leverage scores (Woodruff et al., 2014) but due to \( p \) order and data point coming in online manner LineFilter charges an extra factor of \( (i/p)^{p/2-1} \) for every row. Although we have bound on the \( \sum_{i=1}^n \tilde{I}_i \) from lemma 5.3, but this factor can be very huge as \( i \) increases which eventually sets many \( \tilde{I}_i = 1 \).

A.1.2. PROOF OF LEMMA 5.2

Proof. For simplicity, we prove this lemma at the last timestamp \( n \). But it can also be proved for any timestamp \( t_i \), which is why the LineFilter can also be used in the restricted streaming (online) setting.

Now for a fixed \( x \in \mathbb{R}^d \) and its corresponding \( y \), we define a random variables as follows, i.e. the choice LineFilter has every for incoming row \( i \).

\[
w_i = \begin{cases} \frac{1}{p_i} |u_i^T y|^p & \text{with probability } p_i \\ 0 & \text{with probability } (1 - p_i) \end{cases}
\]

where \( u_i^T \) is the \( i^{th} \) row of \( U \) for \( U, \Sigma, V = \text{svd}(A) \) and \( y = \Sigma V^T x \). Here we get \( E[w_i] = (|u_i^T y|^p) \). In our online algorithm we have defined \( p_i = \min\{r\tilde{l}_i / \sum_j \tilde{l}_j, 1\} \) where \( r \) is some constant. When \( p_i \leq 1 \), we have

\[
p_i = \frac{r \tilde{l}_i / \sum_j \tilde{l}_j \geq \frac{r |u_i^T y|^p}{\sum_j \tilde{l}_j \sum_j |u_j^T y|^p}}{\sum_j \tilde{l}_j \sum_j |u_j^T y|^p} \geq \frac{r |u_i^T y|^p}{\sum_j \tilde{l}_j \sum_j |u_j^T y|^p}
\]

As we are analysing a lower bound on \( p_i \), and both the terms in the denominator are positive so we extend the sum of first \( i \) terms to all the \( n \) terms. Now to apply Bernstein inequality
A.1 we bound the term $|w_i - \mathbb{E}[w_i]| \leq b$. Consider the two possible cases,

Case 1: When $w_i$ is non-zero, then $|w_i - \mathbb{E}[w_i]| \leq \frac{u_T^T y}{p_i} \leq \frac{\|u_T^T y\|^p}{r}$. Note for $p_i = 1$, $|w_i - \mathbb{E}[w_i]| = 0$.

Case 2: When $w_i$ is 0 then $p_i < 1$. So by setting $b = \frac{\|u_T^T y\|^p}{r} \sum_{j=1}^n |u_j^T y|^p$ we can bound the term $|w_i - \mathbb{E}[w_i]| \leq \frac{\|u_T^T y\|^p}{r}$. Note that $\sum_{i=1}^n u_i = \sum_{i=1}^n \sigma_i^2$, since every incoming rows are independent of each other and here we consider $\sigma_i^2 = \text{var}(w_i)$.

$$\sigma^2 = \sum_{i=1}^n \sigma_i^2 = \sum_{i=1}^n \mathbb{E}[u_i^2] - (\mathbb{E}[w_i])^2$$

$$\leq \sum_{i=1}^n \frac{|u_T^T y|^2}{p_i} \leq \sum_{i=1}^n \frac{|u_T^T y|^2}{r} \sum_{j=1}^n |u_j^T y|^p$$

$$\leq \frac{(\sum_{k=1}^n \hat{t}_k)(\sum_{j=1}^n |u_j^T y|^p)^2}{r}$$

Note that $\|Uy\|_p^2 = \sum_{j=1}^n |u_j^T y|^p$. Now in Bernstein inequality we set $t = \epsilon \sum_{j=1}^n |u_j^T y|^p$, let

$$ \mathbb{P} = \text{Pr}\left( |W - \sum_{j=1}^n (u_j^T y)^p| \geq \epsilon \sum_{j=1}^n |u_j^T y|^p \right)$$

which we bound as follows,

$$ \mathbb{P} \leq \exp\left( \frac{(\epsilon \sum_{j=1}^n |u_j^T y|^p)^2}{2 \sigma^2 + bt/3} \right)$$

$$ \leq \exp\left( \frac{-r \epsilon^2 (\|Uy\|_p^2)^2}{(\|Uy\|_p^2)^2 \sum_{j=1}^n \hat{t}_j (2 + \epsilon/3)} \right)$$

$$ \leq \exp\left( \frac{-r \epsilon^2 (2 + \epsilon/3) \sum_{j=1}^n \hat{t}_j}{(2 + \epsilon/3) \sum_{j=1}^n \hat{t}_j} \right)$$

Now to ensure that the above probability at most 0.01, $\forall x \in Q$ we use $\epsilon$-net argument as in $A$ where we take a union bound over $(2/\epsilon)^k$, $x$ from the net. Note that for our purpose $1/2$-net also suffices. Hence with the union bound over all $x$ in $1/2$-net we need to set $r$ as $O(2k^\sum_{i=1}^n \hat{t}_i)$.

Now to ensure the guarantee for $\ell_p$ subspace embedding for any $p \geq 2$ as in equation (2) one can consider the following form of the random variable,

$$w_i = \begin{cases} \frac{1}{p_i} |u_T^T y|^p & \text{with probability } p_i \\ 0 & \text{with probability } (1 - p_i) \end{cases}$$

and follow the above proof. Finally by setting $r$ as $O(\frac{2k^\sum_{i=1}^n \hat{t}_i}{\epsilon^2})$ one can get

$$ \mathbb{P} = \text{Pr}\left( |W - \|Ax\|_p^p| \geq \epsilon \|Ax\|_p^p \right) \leq 0.01$$

Since both the guarantees of equation (1) and (2) the sampling probability of every incoming row is the same, just the random variables are different, hence for integer valued $p \geq 2$ the same sampled rows preserves both tensor contraction as in equation (1) and $\ell_p$ subspace embedding as in equation (2).

Now we give the detail proof of sum of upper bounds of sensitivity scores, $\sum_{i=1}^n \hat{t}_i$. The proof is novel because of the way we use matrix determinant lemma for a rank deficient matrix, which is further used to get a telescopic sum for all the terms.

A.1.3. Proof of Lemma 5.3

Proof. Recall that $A_i$ denotes the $i \times d$ matrix of the first $i$ incoming rows. LineFilter maintains the covariance matrix $M$. At the $(i - 1)^{th}$ step we have $M = A_{i-1}^T A_{i-1}$. This is then used to define the score $\hat{t}_i$ for the next step, i.e.,

$$\hat{t}_i = \min\{i^{p/2-1}2^{p/2-1}, 1\}, \quad \hat{e}_i = a_i^T (M + a_i a_i^T)^{-1} a_i = a_i^T (A_i^T A_i)^{-1} a_i$$

where $a_i$ is the $i^{th}$ row. The scores $\hat{e}_i$ is also called online leverage scores. We first give a bound on $\sum_{i=1}^n a_i$. A similar bound is given in the online matrix row sampling by (Cohen et al., 2016), albeit for a regularized version of the scores $\hat{e}_i$. As the rows are coming, the rank of $M$ increases from 1 to at most $d$. We say that the algorithm is in phase-$k$ if the rank of $M$ equals $k$. For each phase $k \in [1, d - 1]$, let $k_i$ denote the index where row $a_{i_k}$ caused a phase-change in $M$ i.e. rank of $(A_{i_k}^T A_{i_k})$ is $k - 1$, while rank of $(A_{i_k}^T A_{i_k})$ is $k$. For each such $k_i$, the online leverage score $\hat{e}_{k_i} = 1$, since row $a_{k_i}$ does not lie in the row space of $A_{i_k}$. There are at most $d$ such indices $k_i$.

We now bound the $\sum_{i=1}^{k_i} \hat{e}_{k_i}$. Suppose the thin-SVD$(A_i^T A_i) = V \Sigma_i V^T$, all entries in $\Sigma_i$ being positive. Furthermore, for any $i$ in this phase, i.e. for $i \in [k, k + 1 - 1]$, then $V$ forms the basis of the row space of $A_i$. Define $X_i = V^T (A_i^T A_i) V$ and the $i^{th}$ row $a_i = V b_i$. Notice that each $X_i \in \mathbb{R}^{k \times k}$, and $X_{i_k} = \Sigma_{i_k}$. Also, $X_i$ is positive definite. Now for each $i \in [k, k + 1 - 1]$, we have $X_i = X_{i-1} + b_i b_i^T$.

So we have $\hat{e}_i = a_i^T (A_i^T A_i)^{-1} a_i = b_i^T V^T (V (X_{i-1} + b_i b_i^T) V^T) V b_i = b_i^T (X_{i-1} + b_i b_i^T)^{-1} b_i$.
conditions the product of terms from phase 1, we get the following,
\[
\begin{align*}
\det(\mathbf{X}_{i-1} + \mathbf{b}_i^T \mathbf{b}_i) &= \det(\mathbf{X}_{i-1})(1 + \mathbf{b}_i^T \mathbf{X}_{i-1}^{-1} \mathbf{b}_i) \\
&\geq \det(\mathbf{X}_{i-1})(1 + \mathbf{b}_i^T (\mathbf{X}_{i-1} + \mathbf{b}_i \mathbf{b}_i^T)^{-1} \mathbf{b}_i) \\
&= \det(\mathbf{X}_{i-1})(1 + \tilde{e}_i) \\
&\geq \det(\mathbf{X}_{i-1}) \exp(\tilde{e}_i/2) \\
\exp(\tilde{e}_i/2) &\leq \frac{\det(\mathbf{X}_{i-1} + \mathbf{b}_i \mathbf{b}_i^T)}{\det(\mathbf{X}_{i-1})}
\end{align*}
\]

Inequality (i) follows as \( \mathbf{X}_{i-1}^{-1} - (\mathbf{X}_{i-1} + \mathbf{b} \mathbf{b}^T)^{-1} \geq 0 \) (i.e. p.s.d.). Inequality (ii) follows from the fact that \( 1 + x \geq \exp(x/2) \) for \( x \leq 1 \). Now with \( \tilde{e}_{ik} = 1 \), we analyze the product of the remaining terms of the phase \( k \) i.e.,
\[
\prod_{i \in [k+1,n+1]} \exp(\tilde{e}_i/2)
\]
which is,
\[
\prod_{i \in [k+1,n+1]} \frac{\det(\mathbf{X}_i)}{\det(\mathbf{X}_{i-1})} \leq \frac{\det(\mathbf{X}_{k+1+1})}{\det(\mathbf{X}_{k+1})}.
\]
Now by taking the product over all phases the term \( \exp\left(\sum_{k \in [1,d-1]} \tilde{e}_i/2\right) \) gets,
\[
\exp((d-1)/2) \left( \prod_{k \in [1,d-1]} \prod_{i \in [k+1,n+1]} \exp(\tilde{e}_i/2) \right)
= \exp((d-1)/2) \left( \prod_{k \in [1,d-1]} \frac{\det(\mathbf{X}_{k+1})}{\det(\mathbf{X}_{k+1})} \right)
= \exp((d-1)/2) \left( \frac{\det(\mathbf{X}_{d-1})}{\det(\mathbf{X}_{d-1})} \prod_{k \in [2,d-1]} \frac{\det(\mathbf{X}_{k+1})}{\det(\mathbf{X}_{k+1})} \right)
\]
Because we know that for any phase \( k \) we have \( \mathbf{A}_{i+1}^T \mathbf{A}_{i+1} \geq (\mathbf{A}_i^T \mathbf{A}_{i+1}) \) so we get, \( \det(\mathbf{X}_{k+1}) \geq \det(\mathbf{X}_{k+1}) \). Further between inter phases terms, i.e. between the last term of phase \( k-1 \) and the second term of phase \( k \) we have \( \det(\mathbf{X}_{k-1}) \leq \det(\mathbf{X}_{k+1}) \). Note that we independently handle the first term of phase \( k \), i.e. phase change term. Hence we get \( \exp((d-1)/2) \) as there are \( d-1 \) many \( i \) such that \( \tilde{e}_i = 1 \). Due to these conditions the product of terms from 1 to \( i_d \) yields a telescopic product, which gives,
\[
\exp\left(\sum_{i \in [1,i_d]} \tilde{e}_i/2\right) \leq \exp((d-1)/2) \frac{\det(\mathbf{X}_{d-1})}{\det(\mathbf{X}_{d-1})} \leq \exp((d-1)/2) \frac{\det(\mathbf{X}_{d} \mathbf{A}_{d}^T)}{\det(\mathbf{X}_{d+1})}
\]
Furthermore, we know \( \tilde{e}_i = 1 \), so for \( i \in [i_d,n] \), the matrix \( \mathbf{M} \) is full rank. We follow the same argument as above, and obtain the following,
\[
\text{exp}\left(\sum_{i \in [i_d,n]} \tilde{e}_i/2\right) \leq \frac{\exp(1/2) \det(\mathbf{A}_i^T \mathbf{A}_i)}{\det(\mathbf{A}_{i+1}^T \mathbf{A}_{i+1})} \leq \frac{\exp(1/2)\|\mathbf{A}\|^d}{\det(\mathbf{A}_{i+1}^T \mathbf{A}_{i+1})}
\]
Let \( a_{i+1} \) be the first non independent incoming row. Now multiplying the above two expressions and taking logarithm of both sides, and accounting for the indices \( i_k \) for \( k \in [2,d] \),
\[
\sum_{i \leq n} \tilde{e}_i \leq d/2 + 2d \log \|\mathbf{A}\| - 2 \log \|a_{i+1}\|
\]
Now, we give a bound on \( \sum_{i=1}^{n} \tilde{t}_i \) where \( \tilde{t}_i = \min(1, \sqrt{p/2 - 1} \tilde{e}_i^2) \) \leq \min(1, np/2 \tilde{e}_i^2). \) We consider two cases. When \( \tilde{e}_i^2 \geq n^{1-p/2} \) then \( \tilde{t}_i = 1 \), this implies that \( \tilde{e}_i \geq n^{2/p-1} \). But we know \( \sum_{i=1}^{n} \tilde{e}_i \leq O(d + d \log \|\mathbf{A}\| - \min_{i \in [n]} \log \|a_i\|) \) and hence there are at most \( O(n^{1-p/2}d + d \log \|\mathbf{A}\| - \min_{i \in [n]} \log \|a_i\|) \) indices with \( \tilde{t}_i = 1 \). Now for the case where \( \tilde{e}_i^2 < n^{1-p/2} \), we get \( \tilde{e}_i^2 \leq \min(1, np/2 \tilde{e}_i^2) \). Then \( \sum_{i=1}^{n} n^{p/2-1} \tilde{e}_i^2 \leq \sum_{i=1}^{n} n^{p/2-1} \tilde{e}_i = O(n^{1-p/2}d + d \log \|\mathbf{A}\| - \min_{i \in [n]} \log \|a_i\|) \).

### A.2. LineFilter + StreamingLW

As we know that any offline algorithm can be converted into a streaming algorithm by using merge and reduce method (Har-Peled & Mazumdar, 2004), so we apply merge and reduce on (Cohen & Peng, 2015). The results in (Cohen & Peng, 2015) is better than the results of (Dasgupta et al., 2009; Woodruff & Zhang, 2013; Clarkson et al., 2016) in terms of sampling complexity, ignoring the \( c \) factor in it. Now we discuss the guarantee that we get from the streaming version of (Cohen & Peng, 2015).

#### A.2.1. PROOF OF LEMMA 4.1

**Proof.** Here the data is coming in streaming sense and it is need to the streaming version of the algorithm in (Cohen & Peng, 2015), i.e. StreamingLW for \( \ell_p \) subspace embedding. We use merge and reduce from (Har-Peled & Mazumdar, 2004) for streaming data. From the results of (Cohen & Peng, 2015) we know that for a set \( P \) of size \( n \) takes \( O(nd^{p/2}) \) time to return a coreset \( Q \) of size \( O(d^{p/2} \log d) \). Note that for the StreamingLW in section 7 of (Har-Peled & Mazumdar, 2004) we set \( M = O(d^{p/2} \log d) \). The method returns \( Q \) as the \( (1 + \delta) \) coreset for the partition \( P \), where
Thus we have $|P_i|$ is either $2^i M$ or 0, here $\rho_j = \epsilon/(c(j + 1)^2)$ such that $1 + \delta_j = \prod_{j=0}^{i}(1 + \rho_j) \leq 1 + \epsilon/2, \forall j \in [\log n]$. Thus we have $|Q_i|$ is $O(d^{p/2}(\log d)(i + 1)^{10}\epsilon^{-5})$. In StreamingLW the method reduce sees at max log $n$ many coresets at any point of time. Hence the total working space is $O(d^{p/2}(\log n)^2(\log d)\epsilon^{-5})$. Now the amortized time spent per update is,

$$\sum_{i=1}^{\log(n/M)} \frac{1}{2^i M}(|Q_i|d^{p/2}) = \sum_{i=1}^{\log(n/M)} \frac{1}{2^i M}(M(i + 1)^4d^{p/2}) \leq d^{p/2}$$

So the finally the algorithm return $Q$ as the final coreset of $O(d^{p/2}(\log n)^2(\log d)\epsilon^{-5})$ rows and uses $O(d^{p/2})$ amortized update time.

Next we discuss the proof of the guarantee of the improved streaming algorithm i.e. LineFilter+StreamingLW. Here we do not pass an incoming row directly to StreamingLW, instead first we feed it to LineFilter, if it samples then the row is further passed on to StreamingLW.

### A.2.2. PROOF OF LEMMA 4.2

**Proof.** Here the data is coming in streaming sense. The first method LineFilter filters out the rows with small sensitivity scores and only the sampled rows (high sensitivity score) are passed to StreamingLW. Here the LineFilter ensures that StreamingLW only gets $O(n^{1-2/p}d)$, hence the amortized update time is same as that of LineFilter, i.e. $O(d^2)$. Now similar to the above proof A.2.2, by the StreamingLW from section 7 of (Har-Peled & Mazumdar, 2004) we set $M = O(d^{p/2}(\log d)^{5\epsilon^{-5}})$. The method returns $Q_i$ as the $(1 + \delta_j)$ coreset for the partition $P_i$, where $|P_i|$ is either $2^i M$ or 0, here $\rho_j = \epsilon/(c(j + 1)^2)$ such that $1 + \delta_i = \prod_{j=0}^{i}(1 + \rho_j) \leq 1 + \epsilon/2, \forall j \in [\log n]$. Thus we have $|Q_i|$ is $O(d^{p/2}(\log d)(i + 1)^{10}\epsilon^{-5})$. Hence the total working space is $O((1 - 2/p)^{11}d^{p/2}(\log n)^2(\log d)\epsilon^{-5})$. So finally LineFilter+StreamingLW returns a coreset $Q$ of $O((1 - 2/p)^{10}d^{p/2}(\log 10 n)(\log d)\epsilon^{-5})$ rows.

Note that LineFilter+StreamingLW also returns a slightly improved sampling complexity compare to StreamingLW. We get this benefit due to the sublinear size sample, which LineFilter returns.

### A.3. KernelFilter

In this section we discuss the supporting lemma for proving the theorem 4.3. First we show the reduction from $p$ order operation to $q$ order operation, where $q \leq 2$. While doing that we go from $d$ dimensional vectors to its corresponding higher dimensional vector. The higher dimension depends on the value of the $p$.

### A.4. Proof of Lemma 4.2

**Proof.** The term $|x^T y|^p = |x^T y|^{p/2}|x^T y|^{p/2}$. We define $|x^T y|^{p/2} = \|x\otimes y\|_2$ and $|x^T y|^{p/2} = \|x^T y\|_2$. Hence we get $|x^T y|^p = \|x\otimes y\|_2^2 = \|x^T y\|_2^2$ which is same as in (Schechtman, 2011). Here the vector $\tilde{x}$ is the higher dimensional vector, where $\tilde{x} = \text{vec}(x\otimes y) \in \mathbb{R}^{p/2}$ and similarly $\tilde{y}$ is also defined from $y$. For even valued $p$ we know $|p/2| = [p/2]$, so for simplicity we write as $|x^T y|^{p/2} = |\tilde{x}^T \tilde{y}|$. Hence we get $|x^T y|^p = \|x^T y\|_2^2$ which is same in (Schechtman, 2011). Here the vector $\tilde{x}$ is the higher dimensional vector, where $\tilde{x} = \text{vec}(x\otimes y) \in \mathbb{R}^{p/2}$ and similarly $\tilde{y}$ is also defined from $y$. Now for odd value of $p$ we have $\tilde{x} = \text{vec}(x\otimes y^{(p-1)/2}) \in \mathbb{R}^{p/2}$ and $\tilde{y} = \text{vec}(y\otimes x^{(p+1)/2}) \in \mathbb{R}^{p+1}/2$. Similarly $\tilde{x}$ and $\tilde{y}$ are defined from $y$. Further note that $|x^T y| = |\tilde{x}^T \tilde{y}| - |x^T y|^{(p-1)/2}$ which gives $|x^T y| = \|x\otimes(y^{(p-1)/2})\|_2 - \|x\otimes(y^{(p+1)/2})\|_2 = |\tilde{x}^T \tilde{y}| - |\tilde{x}^T \tilde{y}|^{(p+1)/2}$. It completes the proof.

Here the novelty is in the kernelization for the odd value $p$. In the following supporting lemmas, we will see the benefit for our above kernelization method.

### A.4.1. PROOF OF LEMMA 5.4

**Proof.** We define the online sensitivity scores $\hat{s}_i$ for each point $i$ as follows,

$$\hat{s}_i = \sup_{\{x:||x||_1=1\}} \frac{\{a_i^T x\}_p}{\|A_i x\|_p^p}$$

Let $\hat{A}$ be the matrix where its $j^{th}$ row $\hat{a}_j = \text{vec}(a_j \otimes \tilde{x}^{\otimes (p/2)}) \in \mathbb{R}^{p/2}$. Further let $\hat{A}_i$ are the corresponding matrices $\hat{A}_i \in \mathbb{R}^{p/d}$ which represents first $i$ streaming rows. We define $[U_i, \Sigma_i, \hat{V}_i] = \text{svd}(A_i)$ such that $a_i^T = \hat{u}_i^T \Sigma_i \hat{V}_i^T$. Now for a fixed $x \in \mathbb{R}^{d}$ its corresponding $\tilde{x}$ is also fixed in its corresponding higher dimension. Here $\Sigma_i^2 \hat{V}_i^T \tilde{x} = \tilde{z}$ from which we define unit vector $\tilde{y} = \tilde{z}/||\tilde{z}||$. Now for even value $p$, similar to (Schechtman, 2011) can easily upper bound the terms $\hat{s}_i$ as follows,

$$\hat{s}_i = \sup_{\{x:||x||_1=1\}} \frac{\{\hat{a}_i^T x\}_p}{\|\hat{A}_i x\|_p^p} = \sup_{\{\tilde{y}:||\tilde{y}||=1\}} \frac{\{\hat{a}_i^T \tilde{y}\}_2}{\|\hat{U}_i \tilde{y}\|_2^2} \leq \frac{\|\hat{u}_i\|_2^2}{\|\hat{U}_i \tilde{y}\|_2^2}$$

Here every equality is by substitution from our above mentioned assumptions and the final inequality is well known from (Woodruff et al., 2014; Cohen et al., 2015). Hence
finally we get $\hat{s}_i \leq ||\hat{u}_i||^2$ for even value $p$ as defined in KernelFilter.

Now for odd value $p$ we analyze $\hat{s}_i$ as follows,

$$\hat{s}_i = \sup_{\{x||x||=1\}} |\hat{u}_i^T x|^p \leq \sup_{\{x||x||=1\}} \|\hat{u}_i^T x\|^{2p/(p+1)}$$

$$= \sup_{\{x||x||=1\}} \|\hat{u}_i^T x\|^{2p/(p+1)} \leq \|\hat{u}_i\|^{2p/(p+1)}$$

The equality (i) is by lemma 4.2. Next with similar assumption as above let $\{\hat{U}_i, \hat{\Sigma}_i, \hat{V}_i\} = svd(\hat{A}_i)$. The inequality (ii) is because $||\hat{U}_i\hat{y}_i||_{2p/(p+1)} \geq ||\hat{U}_i\hat{y}_i||$ and finally we get $\hat{s}_i \leq \hat{l}_i$ as defined in KernelFilter for odd $p$ value.

A.4.2. PROOF OF LEMMA 5.5

Proof. For simplicity we prove this lemma at the last timestamp $n$. But it can also be proved for any timestamp $t_i$ which is why the KernelFilter can also be used in restricted streaming (online) setting. Also for a change we show this for $\ell_p$ subspace embedding. Now for some fixed $x \in \mathbb{R}^d$ consider the following random variable for every row $i$:

$$w_i = \begin{cases} (1/p_i - 1)|\hat{u}_i^T x|^p & \text{w.p. } p_i \\ -|\hat{u}_i^T x|^p & \text{w.p. } (1 - p_i) \end{cases}$$

Note that $\mathbb{E}[w_i] = 0$. Now to show the concentration of the expected term we will apply Bernstein’s inequality A.1 on $W = \sum_{i=1}^n w_i$. For this first we bound $|w_i - \mathbb{E}[w_i]| = |w_i| \leq b$ and then we give a bound on $\mathbb{E}[W]$ by $\sigma^2$.

Now for the $i$th timestamp KernelFilter defines $p_i = \min\{1, r_i/\sum_{j=1}^n \hat{l}_j\}$ where $r$ is some constant. If $p_i = 1$ then $|w_i| = 0$, else if $p_i < 1$ and KernelFilter samples $x$ then $|w_i| \leq |\hat{u}_i^T x|^p/p_i = |\hat{u}_i^T x|^p \sum_{j=1}^n \hat{l}_j/(r_i \hat{l}_j) \leq \|A_i x\|^p ||\hat{u}_i^T x||^p \sum_{j=1}^n \hat{l}_j/(r_i \hat{l}_j) \leq \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/(r_i \hat{l}_j) / \|A_i x\|^p \sum_{j=1}^n \hat{l}_j$. Next when KernelFilter does not sample the $i$th row, it means that $p_i < 1$, then we have $1 > r_i/\sum_{j=1}^n \hat{l}_j \geq r_i |\hat{u}_i^T x|^p/(\|A_i x\|^p \sum_{j=1}^n \hat{l}_j) \geq r_i |\hat{u}_i^T x|^p/(\|A_i x\|^p \sum_{j=1}^n \hat{l}_j)$. Finally we get $|\hat{u}_i^T x|^p \leq \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/r$. So for each $i$ we get $|w_i| \leq \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/r$. Now to show the concentration of $w_i$ we use Bernstein A.1 to bound the probability $\mathbb{P} = \text{Pr}(\mathbb{E}[W] \geq \epsilon \|A_i x\|^p/p_i)$. Here we have $b = \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/r$, $\sigma^2 = \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/r$ and we set $t = \epsilon \|A_i x\|^p/p_i$, then we get

$$\mathbb{P} \leq \exp\left(\frac{-(\epsilon \|A_i x\|^p)^2}{2 \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/r + \epsilon \|A_i x\|^p \sum_{j=1}^n \hat{l}_j/3r}\right)$$

$$= \exp\left(\frac{-(\epsilon \|A_i x\|^p)^2}{(2 + \epsilon/3) \|A_i x\|^p \sum_{j=1}^n \hat{l}_j}\right)$$

Now to ensure that the above probability at most $0.01$, $\forall \mathbb{x} \in \mathbb{Q}$ we use $\epsilon$-net argument as in A where we take a union bound over $(2/\epsilon)^k$, $\mathbb{x}$ from the net. Note that for our purpose $1/2$-net also suffices. Hence with the union bound over all $\mathbb{x}$ in $1/2$-net we need to set $r = O(ke^{-2} \sum_{j=1}^n \hat{l}_j)$. Now to ensure the guarantee for tensor contraction as equation (1) one can define

$$w_i = \begin{cases} (1/p_i - 1)(\hat{u}_i^T x)^p & \text{w.p. } p_i \\ -(\hat{u}_i^T x)^p & \text{w.p. } (1 - p_i) \end{cases}$$

and follow the above proof. By setting the $r = O(ke^{-2} \sum_{j=1}^n \hat{l}_j)$ one can get

$$\mathbb{P} = \text{Pr}\left(|W - \sum_{j=1}^n (\mathbb{u}_j^T \mathbb{y})^p| \geq \epsilon \sum_{j=1}^n |\mathbb{u}_j^T \mathbb{y}|^p\right) \leq 0.01$$

One may follow the above proof to claim the final guarantee as in equation 1 using the same sampling complexity. Again similar to LineFilter as the sampling probability of the rows are same for both tensor contraction and $\ell_p$ subspace embedding, hence the same subsampled rows preserves both the properties as in equation (1) and (2).
A.4.3. Proof of Lemma 5.6

Proof. Let \( \hat{c}_i = \| \hat{u}_i \| \). Now for even value \( p \) we have \( \sum_{i=1}^{n} \hat{l}_i = \sum_{i=1}^{n} c_i^2 \). From lemma 5.3 we get \( \sum_{i=1}^{n} c_i^2 = O(d^{p/2}(1 + \log \| A \| - d^{-p/2} \min_{i} \log \| a_i \|)) \). Now with \( \| u, \Sigma, V \| = \text{svd}(A) \) we have \( \hat{A}^T = \text{vec}(a_i^T \otimes p^2) = \text{vec}(u_i^T \Sigma V^T)^{p/2}) \). So we get \( \| \hat{A} \| \leq \sigma_i^{p/2} \). Hence \( \sum_{i=1}^{n} \hat{l}_i = O(d^{p/2}(1 + \log \| A \| - d^{-p/2} \min_{i} \log \| a_i \|)) \). Now with \( \| u, \Sigma, V \| = \text{svd}(A) \) we have \( \hat{A}^T = \text{vec}(a_i^T \otimes p^2) = \text{vec}(u_i^T \Sigma V^T)^{p/2}) \). So we get \( \| \hat{A} \| \leq \sigma_i^{p/2} \). Hence \( \sum_{i=1}^{n} c_i^2 = O(d^{p/2}(1 + (p + 1)(\log \| A \| - d^{-p/2} \min_{i} \log \| a_i \|))) \). Now let \( \hat{c} \) be a vector with each index \( \hat{c}_i \) is defined as above. Then in this case we have \( \sum_{i=1}^{n} \hat{l}_i = \| \hat{c} \|^{2p/(p+1)} \leq n^{1/(p+1)}\| \hat{c} \|^{p/(p+1)} \) which is \( O(n^{1/(p+1)}d^{p/(p+1)}) \) (1 + \( p + 1)(\log \| A \| - d^{-p/2} \min_{i} \log \| a_i \|)) \). Now we have, \( \hat{l}_i = a_i^T(A_{i-1}^T A_{i-1} + a_i a_i^T) a_i \geq a_i^T(A^2) a_i = u_i^T u_i \).

For \( p_i \geq \min \{ r u_i^T u_i, 1 \}, \) if \( p_i = 1, \) then \( \| w_i \| = 0, \) else \( p_i = r u_i^T u_i \). So we get \( \| w_i \| \leq 1/r \). Next we bound \( \mathbb{E}(\| w_i \|^2) \).

Let \( W = \sum_{i=1}^{n} w_i \), then variance of \( \| w \| \)

\[
\begin{align*}
\text{var}(\| w \|) &= \sum_{i=1}^{n} \text{var}(\| w_i \|) & \leq n \mathbb{E}(\| w_i \|^2) \\
& \leq \left\| \sum_{j=1}^{n} u_j u_j^T \right\| / r & \leq 1/r
\end{align*}
\]

Next by applying matrix Bernstein theorem A.2 with appropriate \( r \) we get,

\[
\Pr(\| X \| \geq \epsilon) \leq d \exp \left( \frac{-\epsilon^2/2}{2r + \epsilon/(3r)} \right) \leq 0.01
\]

This implies that our algorithm preserves spectral approximation with at least 0.99 probability by setting \( r = O(\log d/\epsilon^2) \).

Then the expected number of samples to preserve \( \ell_2 \) subspace embedding is \( O(\sum_{i=1}^{n} \hat{l}_i(\log d)/\epsilon^2) \). Now from lemma 5.3 we know that for \( p = 2, \sum_{i=1}^{n} \hat{l}_i = O(d(1 + \log \| A \|) - \min_{i} \log \| a_i \|)) \). Finally to get \( \Pr(\| W \| \geq \epsilon) \leq 0.01 \) the algorithm samples \( O(d \log d (1 + \log \| A \|) - d^{-1} \min_{i} \log \| a_i \|)) \) rows.


Under the assumption that the data is generated by some generative model such as Gaussian Mixture model, Topic model, Hidden Markov model etc, one can represent the data in terms of higher order (say 3) moments as \( T^3 \) to realize the latent variables (Anandkumar et al., 2014). Next the tensor is reduced to an orthogonally decomposable
tensor by multiplying a matrix called whitening matrix \( W \in \mathbb{R}^{d \times k} \), such that \( W^T M_2 W = I_k \). Here \( k \) is the number of number of latent variables we are interested and \( M_2 \in \mathbb{R}^{d \times d} \) is the 2nd order moment. Now the reduced tensor \( \bar{T}_r = \bar{T}_j(W,W,W) \) is a \( k \times k \times k \) orthogonally decomposable tensor. Next by running robust tensor power iteration (RTPI) on \( T_r \) we get the eigenvalue/eigenvector pair on which upon applying inverse whitening transformation we get the estimated latent factors and its corresponding weights (Anandkumar et al., 2014).

Note that we give guarantee over the \( d \times d \times d \) tensor where as the main theorem 5.3 (Anandkumar et al., 2014) has conditioned over the smaller orthogonally reducible tensor \( T_r \in \mathbb{R}^{k \times k \times k} \). Now reprhasing the main theorem 5.1 of (Anandkumar et al., 2014) we get that the \( \| M_3 - \bar{T}_3 \| \leq \varepsilon \| W \|^{-3} \) where \( M_3 \) is the true 3-order tensor with no noise and \( \bar{T}_3 \) is the empirical tensor that we get from the dataset. Now we state the guarantees that one gets by applying the RTPI on our sampled data.

**Corollary A.2.** For a dataset \( A \in \mathbb{R}^{n \times d} \) with rows coming in streaming fashion and the algorithm LineFilter+KernelFilter returns acoreset \( C \) which guarantees (1) such that if for all unit vector \( x \in Q \), it ensures \( \varepsilon \sum_{i \leq n} |a_i x|^3 \leq \varepsilon \| W \|^{-3} \). Then applying the RTPI on the sampled coreset \( C \) returns \( k \) eigenpairs \( \{ v_i, \lambda_i \} \) of the reduced (orthogonally decomposable) tensor, such that it ensures \( \forall i \in [k] \),

\[
|v_{(i)} - v_i| \leq 8\varepsilon/\lambda_i \quad \text{and} \quad |\lambda_{(i)} - \lambda_i| \leq 5\varepsilon
\]

Here precisely we have \( Q \) as the column space of the \( W^\dagger \), where \( W \) is the whitening matrix as defined above.

### A.6.1. Tensor Contraction

Now we show empirically that how coreset from LineFilter+KernelFilter preserves 4-order tensor contraction. We compare our method with two other sampling schemes, namely - uniform and LineFilter(2). Here LineFilter(2) is the LineFilter with \( p = 2 \).

**Dataset:** We generated a dataset with 200K rows in \( \mathbb{R}^{30} \). Each coordinate of the row was set with a uniformly generated scalar in \( [0, 1] \). Further, each row were normalized to have \( L_2 \) norm as 1. So we get a matrix of size \( 200K \times 30 \), but we ensured that it had rank 12. Furthermore, 99.99% of the rows in the matrix spanned only an 8-dimensional subspace in \( \mathbb{R}^{30} \) and its orthogonal 4 dimensional subspace was spanned by the remaining 0.01% of the rows. We simulated these rows to come in the online fashion and applied the three sampling strategies. From the coreset returned from these sampling strategies we generated 4-mode tensors \( T \) and we also create the tensor \( T \) using the entire dataset. The three sampling strategies are Uniform, LineFilter(2) and LineFilter+KernelFilter.

**Uniform:** Here, we sample rows uniformly at random. It means that every row has a chance of getting sampled with a probability of \( 1/n \). Intuitively it is highly unlikely to pick a representative row from a subspace spanned by fewer rows. Hence the coreset from this sampling method might not preserve tensor contraction \( \forall x \in Q \).

**LineFilter(2):** Here we sample rows based on online leverage scores \( c_i = \alpha_i^T (A_i^T A_i)^{-1} \alpha_i \). We define a sampling probability for an incoming row \( i \) as \( p_i = c_i / (\sum_{j=1}^n c_j) \). Rows with high leverage scores have higher chance of getting sampled. Though leverage score sampling preserved rank of the the data, but it is not known to preserve higher order moments of the data.

**LineFilter+KernelFilter:** Here every incoming row is first feed to LineFilter. If it samples the row then it further passed to KernelFilter, which decides whether to sample the row in the final coreset or not.

Now we compare the relative error approximation \( |T(\bar{x}, x, x, x) - \bar{T}(\bar{x}, x, x, x)|/T(\bar{x}, x, x, x) \), between three sampling schemes mentioned above. Here \( \bar{T}(\bar{x}, x, x, x) = \sum_{i=1}^n (\alpha_i x)^4 \) and \( \bar{T}(\bar{x}, x, x, x) = \sum_{c \in C} (c_i x)^4 \). In table (3), \( Q \) is set of right singular vectors of \( A \) corresponding to the 5 smallest singular values. This table reports the relative error approximation \( |\sum_{x \in Q} T(\bar{x}, x, x, x) - \sum_{x \in Q} \bar{T}(\bar{x}, x, x, x)|/\sum_{x \in Q} T(\bar{x}, x, x, x) \). The table (4) reports for \( x \) as the right singular vector of the smallest singular value of \( A \). Here we choose this \( x \) because this direction captures the worst direction, as in the direction which has the highest variance in the sampled data. For each sampling technique and each sample size, we ran 5 random experiments and reported the mean of the experiments. Here, the sample size are in expectation.

#### Table 3. Error with \( x \in Q \)

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Uniform</th>
<th>LineFilter(2)</th>
<th>LineFilter + KernelFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1.1663</td>
<td>0.2286</td>
<td>0.1576</td>
</tr>
<tr>
<td>250</td>
<td>0.4187</td>
<td>0.1169</td>
<td>0.0855</td>
</tr>
<tr>
<td>300</td>
<td>0.6998</td>
<td>0.1195</td>
<td>0.0611</td>
</tr>
<tr>
<td>350</td>
<td>0.5704</td>
<td>0.0470</td>
<td>0.0436</td>
</tr>
</tbody>
</table>

#### Table 4. Error with \( x \) as right singular vector of the smallest singular value

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Uniform</th>
<th>LineFilter(2)</th>
<th>LineFilter + KernelFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.3584</td>
<td>0.8842</td>
<td>0.6879</td>
</tr>
<tr>
<td>200</td>
<td>0.8886</td>
<td>0.5003</td>
<td>0.3952</td>
</tr>
<tr>
<td>300</td>
<td>0.8742</td>
<td>0.4195</td>
<td>0.3666</td>
</tr>
<tr>
<td>350</td>
<td>0.9187</td>
<td>0.3574</td>
<td>0.2000</td>
</tr>
</tbody>
</table>